Bootstrapping Coreference Resolution Using Word Associations

Hamidreza Kobdani, Hinrich Schütze, Michael Schiehlen and **Hans Kamp**
Institute for Natural Language Processing
University of Stuttgart
kobdani@ims.uni-stuttgart.de

Abstract

In this paper, we present an unsupervised framework that bootstraps a complete coreference resolution (CoRe) system from *word associations* mined from a large unlabeled corpus. We show that word associations are useful for CoRe – e.g., the strong association between *Obama* and *President* is an indicator of likely coreference. Association information has so far not been used in CoRe because it is sparse and difficult to learn from small labeled corpora. Since unlabeled text is readily available, our unsupervised approach addresses the sparseness problem. In a self-training framework, we train a decision tree on a corpus that is automatically labeled using word associations. We show that this unsupervised system has better CoRe performance than other learning approaches that do not use manually labeled data.

1 Introduction

Coreference resolution (CoRe) is the process of finding markables (noun phrases) referring to the same real world entity or concept. Until recently, most approaches tried to solve the problem by binary classification, where the probability of a pair of markables being coreferent is estimated from labeled data. Alternatively, a model that determines whether a markable is coreferent with a preceding cluster can be used. For both pair-based and cluster-based models, a well established feature model plays an important role. Typical systems use a *rich feature space* based on lexical, syntactic and semantic knowledge. Most commonly used features are described by Soon et al. (2001).

Most existing systems are supervised systems, trained on human-labeled benchmark data sets for English. These systems use linguistic features based on number, gender, person etc. It is a challenge to adapt these systems to new domains, genres and languages because a significant human labeling effort is usually necessary to get good performance.

To address this challenge, we pursue an *unsupervised self-training approach.* We train a classifier on a corpus that is automatically labeled using association information. Self-training approaches usually include the use of some manually labeled data. In contrast, our self-trained system is not trained on any manually labeled data and is therefore a completely unsupervised system. Although training on automatically labeled data can be viewed as a form of supervision, we reserve the term *supervised system* for systems that are trained on manually labeled data.

The key novelty of our approach is that we bootstrap a competitive CoRe system from association information that is mined from an unlabeled corpus in a completely unsupervised fashion. While this method is shallow, it provides valuable information for CoRe because it considers the actual identity of the words in question. Consider the pair of markables *(Obama, President)*. It is a likely coreference pair, but this information is not accessible to standard CoRe systems because they only use string-based features (often called lexical features), named entity features and semantic word class features (e.g., from WordNet) that do not distinguish,

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 783–792,
Portland, Oregon, June 19-24, 2011.

say, *Obama* from *Hawking*.

In our approach, word association information is used for *clustering markables* in unsupervised learning. Association information is calculated as *association scores between heads of markables* as described below. We view association information as an example of a *shallow feature space* which contrasts with the rich feature space that is generally used in CoRe.

Our experiments are conducted using the MCORE system ("Modular COreference REsolution").[1] MCORE can operate in three different settings: unsupervised (subsystem *A-INF*), supervised (subsystem *SUCRE* (Kobdani and Schütze, 2010)), and self-trained (subsystem *UNSEL*). The unsupervised subsystem A-INF ("Association INFormation") uses the association scores between heads as the distance measure when clustering markables. SUCRE ("SUpervised Coreference REsolution") is trained on a labeled corpus (manually or automatically labeled) similar to standard CoRe systems. Finally, the unsupervised self-trained subsystem UNSEL ("UNsupervised SELf-trained") uses the unsupervised subsystem A-INF to automatically label an unlabeled corpus that is then used as a training set for SUCRE.

Our main contributions in this paper are as follows:

1. We demonstrate that word association information can be used to develop an unsupervised model for shallow coreference resolution (subsystem A-INF).

2. We introduce an unsupervised self-trained method (UNSEL) that takes a two-learner two-feature-space approach. The two learners are A-INF and SUCRE. The feature spaces are the shallow and rich feature spaces.

3. We show that the performance of UNSEL is better than the performance of other unsupervised systems when it is self-trained on the automatically labeled corpus and uses the leveraging effect of a rich feature space.

4. MCORE is a flexible and modular framework that is able to learn from data with different quality and domain. Not only is it able to deal with shallow information spaces (A-INF), but it can also deliver competitive results for rich feature spaces (SUCRE and UNSEL).

This paper is organized as follows. Related work is discussed in Section 2. In Section 3, we present our system architecture. Section 4 describes the experiments and Section 5 presents and discusses our results. The final section presents our conclusions.

2 Related Work

There are three main approaches to CoRe: supervised, semi-supervised (or weakly supervised) and unsupervised. We use the term *semi-supervised* for approaches that use some amount of human-labeled coreference pairs.

Müller et al. (2002) used co-training for coreference resolution, a semi-supervised method. Co-training puts features into disjoint subsets when learning from labeled and unlabeled data and tries to leverage this split for better performance. Ng and Cardie (2003) use self-training in a multiple-learner framework and report performance superior to co-training. They argue that the multiple learner approach is a better choice for CoRe than the multiple view approach of co-training. Our self-trained model combines multiple learners (A-INF and SUCRE) and multiple views (shallow/rich information). A key difference to the work by Müller et al. (2002) and Ng and Cardie (2003) is that we do not use any human-labeled coreference pairs.

Our basic idea of self-training without human labels is similar to (Kehler et al., 2004), but we address the general CoRe problem, not just pronoun interpretation.

Turning to unsupervised CoRe, Haghighi and Klein (2007) proposed a generative Bayesian model with good performance. Poon and Domingos (2008) introduced an unsupervised system in the framework of Markov logic. Ng (2008) presented a generative model that views coreference as an EM clustering process. We will show that our system, which is simpler than prior work, outperforms these systems.

Haghighi and Klein (2010) present an "almost unsupervised" CoRe system. In this paper, we only compare with completely unsupervised approaches,

[1] MCORE can be downloaded from `ifnlp.org/~schuetze/mcore`.

not with approaches that make some limited use of labeled data.

Recent work by Haghighi and Klein (2009), Klenner and Ailloud (2009) and Raghunathan et al. (2010) challenges the appropriateness of machine learning methods for CoRe. These researchers show that a "deterministic" system (essentially a rule-based system) that uses a rich feature space including lexical, syntactic and semantic features can improve CoRe performance. Almost all CoRe systems, including ours, use a limited number of rules or filters, e.g., to implement binding condition A that reflexives must have a close antecedent in some sense of "close". In our view, systems that use a few basic filters are fundamentally different from carefully tuned systems with a large number of complex rules, some of which use specific lexical information. A limitation of complex rule-based systems is that they require substantial effort to encode the large number of deterministic constraints that guarantee good performance. Moreover, these systems are not adaptable (since they are not machine-learned) and may have to be rewritten for each new domain, genre and language. Consequently, we do not compare our performance with deterministic systems.

Ponzetto (2010) extracts metadata from Wikipedia for supervised CoRe. Using such additional resources in our unsupervised system should further improve CoRe performance. Elsner et al. (2009) present an unsupervised algorithm for identifying clusters of entities that belong to the same named entity (NE) class. Determining common membership in an NE class like person is an easier task than determining coreference of two NEs.

3 System Architecture

Figure 1 illustrates the system architecture of our unsupervised self-trained CoRe system (UNSEL). Oval nodes are data, box nodes are processes. We take a self-training approach to coreference resolution: We first label the corpus using the unsupervised model A-INF and then train the supervised model SUCRE on this automatically labeled training corpus. Even though we train on a labeled corpus, the labeling of the corpus is produced in a completely automatic fashion, without recourse to human labeling. Thus, it is an unsupervised approach.

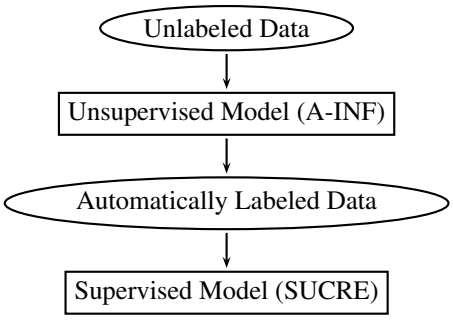

Figure 1: System Architecture of UNSEL (Unsupervised Self-Trained Model).

The MCORE architecture is very flexible; in particular, as will be explained presently, it can be easily adapted for supervised as well as unsupervised settings.

The unsupervised and supervised models have an identical top level architecture; we illustrate this in Figure 2. In *preprocessing*, tokens (words), markables and their attributes are extracted from the input text. The key difference between the unsupervised and supervised approaches is in how *pair estimation* is accomplished — see Sections 3.1 & 3.2 for details.

The main task in *chain estimation* is clustering. Figure 3 presents our clustering method, which is used for both supervised and unsupervised CoRe. We search for the best predicted antecedent (with coreference probability $p \geq 0.5$) from right to left starting from the end of the document. McEnery et al. (1997) showed that in 98.68% of cases the antecedent is within a 10-sentence window; hence we use a window of 10 sentences for search. We have found that limiting the search to a window increases both efficiency and effectiveness.

Filtering. We use a feature definition language to define the templates according to which the filters and features are calculated. These templates are hard constraints that filter out all cases that are clearly disreferent, e.g., *(he, she)* or *(he, they)*. We use the following filters: (i) the antecedent of a reflexive pronoun must be in the same sentence; (ii) the antecedent of a pronoun must occur at a distance of at most 3 sentences; (iii) a coreferent pair of a noun and a pronoun or of two pronouns must not

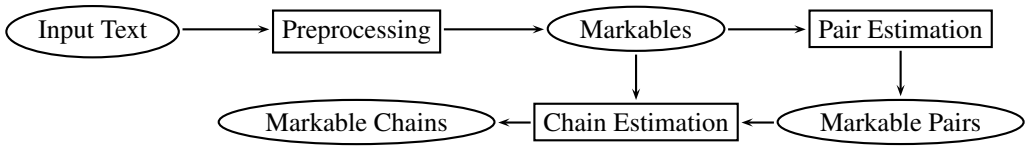

Figure 2: Common architecture of unsupervised (A-INF) and supervised (SUCRE) models.

Chain_Estimation $(M_1, M_2, \ldots, M_n)$
1. $t \leftarrow 1$
2. For each markable M_i: $C_i \leftarrow \{M_i\}$
3. Proceed through the markables from the end of the document. For each M_j, consider each preceding M_i within 10 sentences:
If Pair_Estimation(M_i, M_j)>=t: $C_i \leftarrow C_i \cup C_j$
4. $t \leftarrow t - 0.01$
5. If $t >= 0.5$: go to step 3

Pair_Estimation (M_i, M_j):
If Filtering(M_i, M_j)==FALSE then return 0; else return the probability p (or association score N) of markable pair (M_i, M_j) being coreferent.

Filtering (M_i, M_j):
return TRUE if all filters for (M_i, M_j) are TRUE else FALSE

Figure 3: MCORE chain estimation (clustering) algorithm (test). t is the clustering threshold. C_i refers to the cluster that M_i is a member of.

disagree in number; (iv) a coreferent pair of two pronouns must not disagree in gender. These four filters are used in supervised and unsupervised modes of MCORE.

3.1 Unsupervised Model (A-INF)

Figure 4 (top) shows how A-INF performs pair estimation. First, in the pair generation step, all possible pairs inside 10 sentences are generated. Other steps are separately explained for train and test as follows.

Train. In addition to the filters (i)–(iv) described above, we use the following filter: *(v) If the head of markable M_2 matches the head of the preceding markable M_1, then we ignore all other pairs for M_2 in the calculation of association scores.*

This additional filter is necessary because an approach without some kind of string matching constraint yields poor results, given the importance of string matching for CoRe. As we will show below, even the simple filters (i)–(v) are sufficient to learn high-quality association scores; this means that we do not need the complex features of "deterministic" systems. However, if such complex features are available, then we can use them to improve performance in our self-trained setting.

To learn word association information from an unlabeled corpus (see Section 4), we compute mutual information (MI) scores between heads of markables. We define MI as follows: (Cover and Thomas, 1991)

$$\text{MI}(a,b) = \sum_{i \in \{\bar{a}, a\}} \sum_{j \in \{\bar{b}, b\}} P(i,j) \log_2 \frac{P(i,j)}{P(i)P(j)}$$

E.g., $P(a, \bar{b})$ is the probability of a pair whose two elements are a and a word not equal to b.

Test. A key virtue of our approach is that in the classification of pairs as coreferent/disreferent, the coreference probability p estimated in supervised learning plays exactly the same role as the association information score N (defined below). For p, it is important that we only consider pairs with $p \geq 0.5$ as potentially coreferent (see Figure 3). To be able to impose the same constraint on N, we normalize the MI scores by the maximum values of the two words and take the average:

$$N(a,b) = \frac{1}{2}\left(\frac{\text{MI}(a,b)}{\text{argmax}_x \text{MI}(a,x)} + \frac{\text{MI}(a,b)}{\text{argmax}_x \text{MI}(x,b)}\right)$$

In the above equation, the value of N indicates how strongly two words are associated. N is normalized to ensure $0 \leq N \leq 1$. If a or b did not occur, then we set $N = 0$.

In filtering for test, we use filters (i)–(iv). We then fetch the MI values and calculate N values. The clustering algorithm described in Figure 3 uses these N values in exactly the same way as p: we search for the antecedent with the maximum association score

N greater than 0.5 from right to left starting from the end of the document.

As we will see below, using N scores acquired from an unlabeled corpus as the only source of information for CoRe performs surprising well. However, the weaknesses of this approach are (i) the failure to cover pairs that do not occur in the unlabeled corpus (negatively affecting recall) and (ii) the generation of pairs that are not plausible candidates for coreference (negatively affecting precision). To address these problems, we train a model on a corpus labeled by A-INF in a self-training approach.

3.2 Supervised Model (SUCRE)

Figure 4 (bottom) presents the architecture of pair estimation for the supervised approach (SUCRE).

In the pair generation step for train, we take each coreferent markable pair (M_i, M_j) without intervening coreferent markables and use (M_i, M_j) as a positive training instance and (M_i, M_k), $i < k < j$, as negative training instances. For test, we generate all possible pairs within 10 sentences. After filtering, we then calculate a feature vector for each generated pair that survived filters (i)–(iv).

Our basic features are similar to those described by Soon et al. (2001): string-based features, distance features, span features, part-of-speech features, grammatical features, semantic features, and agreement features. These basic features are engineered with the goal of creating a feature set that will result in good performance. For this purpose we used the relational feature engineering framework which has been presented in (Kobdani et al., 2010). It includes powerful and flexible methods for implementing and extracting new features. It allows systematic and fast search of the space of features and thereby reduces the time and effort needed for defining optimal features. We believe that the good performance of our supervised system SUCRE (tables 1 and 2) is the result of our feature engineering approach.[2]

As our classification method, we use a decision tree[3] (Quinlan, 1993) that is trained on the training set to estimate the coreference probability p for a pair and then applied to the test set. Note that, as is standard in CoRe, filtering and feature calculation are exactly the same for training and test, but that pair generation is different as described above.

4 Experimental Setup

4.1 Data Sets

For computing word association, we used a corpus of about 63,000 documents from the 2009 English *Wikipedia* (the articles that were larger than 200 bytes). This corpus consists of more than 33.8 million tokens; the average document length is 500 tokens. The corpus was parsed using the Berkeley parser (Petrov and Klein, 2007). We ignored all sentences that had no parse output. The number of detected markables (all noun phrases extracted from parse trees) is about 9 million.

We evaluate unsupervised, supervised and self-trained models on *ACE (Phase 2)* (Mitchell et al., 2003).[4] This data set is one of the most widely used CoRe benchmarks and was used by the systems that are most comparable to our approach; in particular, it was used in most prior work on unsupervised CoRe. The corpus is composed of three data sets from three different news sources. We give the number of test documents for each: (i) Broadcast News (BNEWS): 51. (ii) Newspaper (NPAPER): 17. (iii) Newswire (NWIRE): 29. We report results for *true markables* (markables extracted from the answer keys) to be able to compare with other systems that use true markables.

In addition, we use the recently published *OntoNotes* benchmark (Recasens et al., 2010). OntoNotes is an excerpt of news from the OntoNotes Corpus Release 2.0 (Pradhan et al., 2007). The advantage of OntoNotes is that it contains two parallel annotations: (i) a *gold setting*, gold standard manual annotations of the preprocessing information and (ii) an *automatic setting*, automatically predicted annotations of the preprocessing information. The automatic setting reflects the situation a CoRe system

[2] While this is not the focus of this paper, SUCRE has performance comparable to other state-of-the-art supervised systems. E.g., B^3/MUC F_1 are 75.6/72.4 on ACE-2 and 69.4/70.6 on MUC-6 compared to 78.3/66.0 on ACE-2 and 70.9/68.5 on MUC-6 for Reconcile (Stoyanov et al., 2010)

[3] We also tried support vector machines and maximum entropy models, but they did not perform better.

[4] We used two variants of ACE (Phase 2): ACE-2 and ACE2003

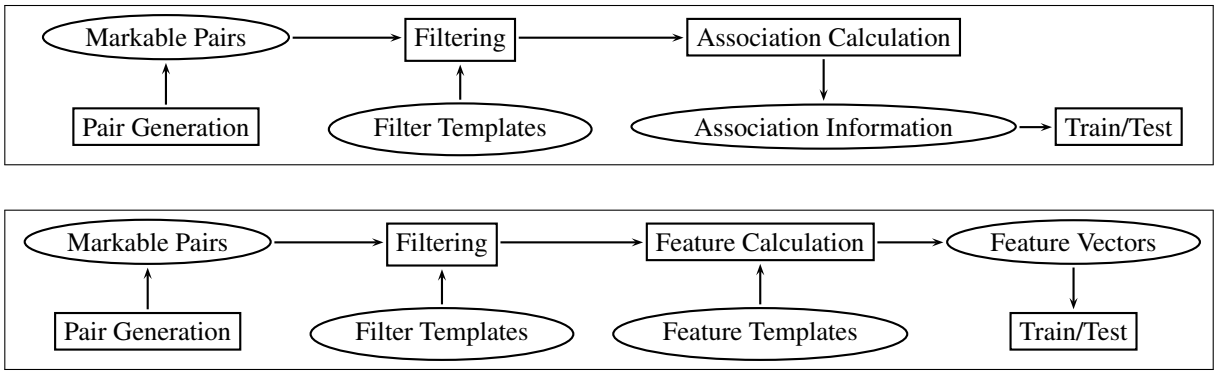

Figure 4: Pair estimation in the unsupervised model A-INF (top) and in the supervised model SUCRE (bottom).

faces in reality; in contrast, the gold setting should be considered less realistic.

The issue of gold vs. automatic setting is directly related to a second important evaluation issue: the influence of markable detection on CoRe evaluation measures. In a real application, we do not have access to true markables, so an evaluation on *system markables* (markables automatically detected by the system) reflects actual expected performance better. However, reporting only CoRe numbers (even for system markables) is not sufficient either since accuracy of markable detection is necessary to interpret CoRe scores. Thus, we need (i) measures of the quality of system markables (i.e., an evaluation of the markable detection subtask) and CoRe performance on system markables as well as (ii) a measure of CoRe performance on true markables. We use OntoNotes in this paper to perform such a, in our view, complete and realistic evaluation of CoRe. The two evaluations correspond to the two evaluations performed at SemEval-2010 (Recasens et al., 2010): the automatic setting with system markables and the gold setting with true markables. Test set size is 85 documents.

In the experiments with A-INF we use Wikipedia to compute association information and then evaluate the model on the test sets of ACE and OntoNotes. For the experiments with UNSEL, we use its unsupervised subsystem A-INF (which uses Wikipedia association scores) to automatically label the training sets of ACE and OntoNotes. Then for each data set, the supervised subsystem of UNSEL (i.e., SUCRE) is trained on its automatically labeled training set and then evaluated on its test set. Finally, for the supervised experiments, we use the manually labeled training sets and evaluate on the corresponding test sets.

4.2 Evaluation Metrics

We report recall, precision, and F_1 for MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF (Luo, 2005). We selected these three metrics because a single metric is often misleading and because we need to use metrics that were used in previous unsupervised work.

It is well known that MUC by itself is insufficient because it gives misleadingly high scores to the "single-chain" system that puts all markables into one chain (Luo et al., 2004; Finkel and Manning, 2008). However, B^3 and CEAF have a different bias: they give high scores to the "all-singletons" system that puts each markable in a separate chain. On OntoNotes test, we get $B^3 = 83.2$ and CEAF $= 71.2$ for all-singletons, which incorrectly suggests that performance is good; but MUC F_1 is 0 in this case, demonstrating that all-singletons performs poorly. With the goal of performing a complete evaluation, one that punishes all-singletons as well as single-chain, we use one of the following two combinations: (i) MUC and B^3 or (ii) MUC and CEAF. Recasens et al. (2010) showed that B^3 and CEAF are highly correlated (Pearson's $r = 0.91$). Therefore, either combination (i) or combination (ii) fairly characterizes CoRe performance.

5 Results and Discussion

Table 1 compares our unsupervised self-trained model UNSEL and unsupervised model A-INF to

		MUC			B^3			CEAF		
	BNEWS-ACE-2	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
1	P&D	68.3	66.6	67.4	70.3	65.3	67.7	–	–	–
2	A-INF	60.8	61.4	61.1	55.5	69.0	61.5	52.6	52.0	52.3
3	UNSEL	72.5	65.6	68.9	72.5	66.4	69.3	56.7	64.8	60.5
4	SUCRE	86.6	60.3	71.0	87.6	64.6	74.4	56.1	81.6	66.5
	NWIRE-ACE-2	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
5	P&D	67.7	67.3	67.4	74.7	68.8	71.6	–	–	–
6	A-INF	62.4	57.4	59.8	59.2	62.4	60.7	46.8	52.5	49.5
7	UNSEL	76.2	61.5	68.1	81.5	67.6	73.9	61.5	77.1	68.4
8	SUCRE	82.5	65.7	73.1	85.4	72.3	78.3	63.5	80.6	71.0
	NPAPER-ACE-2	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
9	P&D	69.2	71.7	70.4	70.0	66.5	68.2	–	–	–
10	A-INF	60.6	56.0	58.2	52.4	60.3	56.0	38.9	44.0	41.3
11	UNSEL	78.6	65.7	71.6	74.0	68.0	70.9	57.6	73.2	64.5
12	SUCRE	82.5	67.0	73.9	80.7	69.5	74.6	58.8	77.1	66.7
	BNEWS-ACE2003	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
13	H&K	68.3	56.8	62.0	–	–	–	59.9	53.9	56.7
14	Ng	71.4	56.1	62.8	–	–	–	60.5	53.3	56.7
15	A-INF	60.9	64.9	62.8	50.9	72.5	59.8	53.8	49.4	51.5
16	UNSEL	69.5	65.0	67.1	70.2	65.9	68.0	58.5	64.2	61.2
17	SUCRE	73.9	68.5	71.1	75.4	69.6	72.4	60.1	66.6	63.2
	NWIRE-ACE2003	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
18	H&K	66.2	46.8	54.8	–	–	–	62.8	49.6	55.4
19	Ng	68.3	47.0	55.7	–	–	–	60.7	49.2	54.4
20	A-INF	62.7	60.5	61.6	54.8	66.1	59.9	47.7	50.2	49.0
21	UNSEL	64.8	68.6	66.6	61.5	73.6	67.0	59.8	55.1	57.3
22	SUCRE	77.6	69.3	73.2	78.8	75.2	76.9	65.1	74.4	69.5

Table 1: Scores for MCORE (A-INF, SUCRE and UNSEL) and three comparable systems on ACE-2 and ACE2003.

P&D (Poon and Domingos, 2008) on ACE-2; and to Ng (Ng, 2008) and H&K[5] (Haghighi and Klein, 2007) on ACE2003. To our knowledge, these three papers are the best and most recent evaluation results for unsupervised learning and they all report results on ACE-2 and ACE-2003. Results on SUCRE will be discussed later in this section.

A-INF scores are below some of the earlier unsupervised work reported in the literature (lines 2, 6, 10) although they are close to competitive on two of the datasets (lines 15 and 20: MUC scores are equal or better, CEAF scores are worse). Given the simplicity of A-INF, which uses nothing but associations mined from a large unannotated corpus, its performance is surprisingly good.

Turning to UNSEL, we see that F_1 is always better for UNSEL than for A-INF, for all three measures (lines 3 vs 2, 7 vs 6, 11 vs 10, 16 vs 15, 21 vs 20). This demonstrates that the self-training step of UNSEL is able to correct many of the errors that A-INF commits. Both precision and recall are improved with two exceptions: recall of B^3 decreases from line 2 to 3 and from 15 to 16.

When comparing the unsupervised system UNSEL to previous unsupervised results, we find that UNSEL's F_1 is higher in all runs (lines 3 vs 1, 7 vs 5, 11 vs 9, 16 vs 13&14, 21 vs 18&19). The differences are large (up to 11%) compared to H&K and

[5]We report numbers for the better performing Pronoun-only Salience variant of H&K proposed by Ng (2008).

Ng. The difference to P&D is smaller, ranging from 2.7% (B^3, lines 11 vs 9) to 0.7% (MUC, lines 7 vs 5). Given that MCORE is a simpler and more efficient system than this prior work on unsupervised CoRe, these results are promising.

In contrast to F_1, there is no consistent trend for precision and recall. For example, P&D is better than UNSEL on MUC recall for BNEWS-ACE-2 (lines 1 vs 3) and H&K is better than UNSEL on CEAF precision for NWIRE-ACE2003 (lines 18 vs 21). But this higher variability for precision and recall is to be expected since every system trades the two measures off differently.

These results show that the application of self-training significantly improves performance. As discussed in Section 3.1, self-training has positive effects on both recall and precision. We now present two simplified examples that illustrate this point.

Example for recall. Consider the markable pair (*Novoselov*[6]*,he*) in the test set. Its N score is 0 because our subset of 2009 Wikipedia sentences has no occurrence of *Novoselov.* However, A-INF finds many similar pairs like (*Einstein,he*) and (*Hawking,he*), pairs that have high N scores. Suppose we represent pairs using the following five features: <sentence distance, string match, type of first markable, type of second markable, number agreement>. Then (*Einstein,he*), (*Hawking,he*) and (*Novoselov,he*) will all be assigned the feature vector <1, No, Proper Noun, Personal Pronoun, Yes>. We can now automatically label Wikipedia using A-INF – this will label (*Einstein,he*) and (*Hawking,he*) as coreferent – and train SUCRE on the resulting training set. SUCRE can then resolve the coreference (*Novoselov,he*) correctly. We call this the *better recall effect*.

Example for precision. Using the same representation of pairs, suppose that for the sequence of markables *Biden, Obama, President* the markable pairs (*Biden,President*) and (*Obama,President*) are assigned the feature vectors <8, No, Proper Noun, Proper Noun, Yes> and <1, No, Proper Noun, Proper Noun, Yes>, respectively. Since both pairs have N scores > 0.5, A-INF incorrectly puts the three markables into one cluster. But as we would expect, A-INF labels many more markable pairs with the second feature vector (distance=1) as coreferent than with the first one (distance=8) in the entire automatically labeled training set. If we now train SUCRE on this training set, it can resolve such cases in the test set correctly even though they are so similar: (*Biden,President*) is classified as disreferent and (*Obama,President*) as coreferent. We call this the *better precision effect*.

[6]The 2010 physics Nobel laureate.

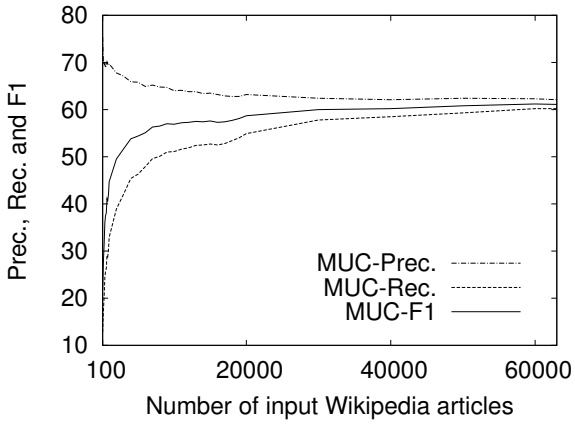

Figure 5: MUC learning curve for A-INF.

Recall that UNSEL has better recall and precision than A-INF in almost all cases (discussion of Table 1). This result shows that better precision and better recall effects do indeed benefit UNSEL.

To summarize, the advantages of our self-training approach are: (i) We cover cases that do not occur in the unlabeled corpus (better recall effect); and (ii) we use the leveraging effect of a rich feature space including distance, person, number, gender etc. to improve precision (better precision effect).

Learning curve. Figure 5 presents MUC scores of A-INF as a function of the number of Wikipedia articles used in unsupervised learning. We can see that a small number of input articles (e.g., 100) results in low recall and high precision. When we increase the number of input articles, recall rapidly increases and precision rapidly decreases up to about 10,000 articles. Increase and decrease continue more slowly after that. F_1 increases throughout because lower precision is compensated by higher recall. This learning curve demonstrates the importance of the size of the corpus for A-INF.

Comparison of UNSEL with SUCRE

Table 2 compares our unsupervised self-trained (UNSEL) and supervised (SUCRE) models with the recently published SemEval-2010 OntoNotes re-

Gold setting + True markables				
System	MD	MUC	B^3	CEAF
Relax	100	33.7	84.5	75.6
$SUCRE_{2010}$	100	60.8	82.4	74.3
SUCRE	100	64.3	87.0	80.1
UNSEL	100	63.0	86.9	79.7
Automatic setting + System markables				
System	MD	MUC	B^3	CEAF
$SUCRE_{2010}$	80.7	52.5	67.1	62.7
Tanl-1	73.9	24.6	61.3	57.3
SUCRE	80.9	55.7	69.7	66.6
UNSEL	80.9	55.0	69.8	66.3

Table 2: F_1 scores for MCORE (SUCRE and UNSEL) and the best comparable systems in SemEval-2010. MD: Markable Detection F_1 (Recasens et al., 2010).

sults (gold and automatic settings). We compare with the scores of the two best systems, Relax and $SUCRE_{2010}$[7] (for the gold setting with true markables) and $SUCRE_{2010}$ and Tanl-1 (for the automatic setting with system markables, 89.9% markable detection (MD) F_1). It is apparent from this table that our supervised and unsupervised self-trained models outperform Relax, $SUCRE_{2010}$ and Tanl-1. We should make clear that we did not use the test set for development to ensure a fair comparison with the participant systems at SemEval-2010.

Table 1 shows that the unsupervised self-trained system (UNSEL) does a lot worse than the supervised system (SUCRE) on ACE.[8] In contrast, UNSEL performs almost as well as SUCRE on OntoNotes (Table 2), for both gold and automatic settings: F_1 differences range from $+.1$ (Automatic, B^3) to -1.3 (Gold, MUC). We suspect that this is partly due to the much higher proportion of singletons in OntoNotes than in ACE-2: 85.2% (OntoNotes) vs. 60.2% (ACE-2). The low recall of the automatic labeling by A-INF introduces a bias for singletons when UNSEL is self-trained. Another reason is that the OntoNotes training set is about 4 times larger than each of BNEWS, NWIRE and NPAPER training sets. With more training data, UNSEL can correct more of its precision and recall errors. For an unsupervised approach, which only needs unlabeled data, there is little cost to creating large training sets. Thus, this comparison of ACE-2/Ontonotes results is evidence that in a realistic scenario using association information in an unsupervised self-trained system is almost as good as a system trained on manually labeled data.

It is important to note that the comparison of SUCRE to UNSEL is the most direct comparison of supervised and unsupervised CoRe learning we are aware of. The two systems are identical with the single exception that they are trained on manual vs. automatic coreference labels.

6 Conclusion

In this paper, we have demonstrated the utility of association information for coreference resolution. We first developed a simple unsupervised model for shallow CoRe that only uses association information for finding coreference chains. We then introduced an unsupervised self-trained approach where a supervised model is trained on a corpus that was automatically labeled by the unsupervised model based on the association information. The results of the experiments indicate that the performance of the unsupervised self-trained approach is better than the performance of other unsupervised learning systems. In addition, we showed that our system is a flexible and modular framework that is able to learn from data with different quality (perfect vs noisy markable detection) and domain; and is able to deliver good results for shallow information spaces and competitive results for rich feature spaces. Finally, our framework is the first CoRe system that is designed to support three major modes of machine learning equally well: supervised, self-trained and unsupervised.

Acknowledgments

This research was funded by DFG (grant SCHU 2246/4).

We thank Aoife Cahill, Alexander Fraser, Thomas Müller and the anonymous reviewers for their helpful comments.

[7] It is the first version of our supervised system that took part in SemEval-2010. We call it $SUCRE_{2010}$.

[8] A reviewer observes that SUCRE's performance is better than the supervised system of Ng (2008). This may indicate that part of our improved unsupervised performance in Table 1 is due to better feature engineering implemented in SUCRE.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *LREC Workshop on Linguistics Coreference '98*, pages 563–566.

Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. Wiley.

Micha Elsner, Eugene Charniak, and Mark Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *HLT-NAACL '09*, pages 164–172.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *HLT '08*, pages 45–48.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. In *ACL '07*, pages 848–855.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *EMNLP '09*, pages 1152–1161.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *NAACL-HLT '10*, pages 385–393.

Andrew Kehler, Douglas E. Appelt, Lara Taylor, and Aleksandr Simma. 2004. Competitive Self-Trained Pronoun Interpretation. In *HLT-NAACL '04*, pages 33–36.

Manfred Klenner and Étienne Ailloud. 2009. Optimization in coreference resolution is not needed: A nearly-optimal algorithm with intensional constraints. In *EACL*, pages 442–450.

Hamidreza Kobdani and Hinrich Schütze. 2010. Sucre: A modular system for coreference resolution. In *SemEval '10*, pages 92–95.

Hamidreza Kobdani, Hinrich Schütze, Andre Burkovski, Wiltrud Kessler, and Gunther Heidemann. 2010. Relational feature engineering of natural language processing. In *CIKM '10*. ACM.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. In *ACL '04*, pages 135–142.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *HLT '05*, pages 25–32.

A. McEnery, I. Tanaka, and S. Botley. 1997. Corpus annotation and reference resolution. In *ANARESOLUTION '97*, pages 67–74.

Alexis Mitchell, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstein, Lisa Ferro, and Beth Sundheim. 2003. ACE-2 version 1.0. Linguistic Data Consortium, Philadelphia.

Christoph Müller, Stefan Rapp, and Michael Strube. 2002. Applying co-training to reference resolution. In *ACL '02*, pages 352–359.

Vincent Ng and Claire Cardie. 2003. Bootstrapping coreference classifiers with multiple machine learning algorithms. In *EMNLP '03*, pages 113–120.

Vincent Ng. 2008. Unsupervised models for coreference resolution. In *EMNLP '08*, pages 640–649.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *HLT-NAACL '07*, pages 404–411.

Simone Paolo Ponzetto. 2010. *Knowledge Acquisition from a Collaboratively Generated Encyclopedia*, volume 327 of *Dissertations in Artificial Intelligence*. Amsterdam, The Netherlands: IOS Press & Heidelberg, Germany: AKA Verlag.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *EMNLP '08*, pages 650–659.

Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. In *ICSC '07*, pages 517–526.

J. Ross Quinlan. 1993. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *EMNLP '10*, pages 492–501.

Marta Recasens, Lluís Màrquez, Emili Sapena, M.Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. SemEval-2010 Task 1: Coreference resolution in multiple languages. In *SemEval '10*, pages 70–75.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. In *CL '01*, pages 521–544.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Coreference resolution with reconcile. In *ACL '10*, pages 156–161.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *MUC6 '95*, pages 45–52.