Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 352-359.

Applying Co-Training to Reference Resolution

Christoph Müller
European Media Laboratory GmbH
Villa Bosch
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
mueller@eml.villa-bosch.de

Stefan Rapp
Sony International (Europe) GmbH
Advanced Technology Center Stuttgart
Heinrich-Hertz-Straße 1
70327 Stuttgart, Germany
rapp@sony.de

Michael Strube
European Media Laboratory GmbH
Villa Bosch
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
strube@eml.villa-bosch.de

Abstract

In this paper, we investigate the practical applicability of Co-Training for the task of building a classifie for reference resolution. We are concerned with the question if Co-Training can significantl reduce the amount of manual labeling work and still produce a classifie with an acceptable performance.

1 Introduction

A major obstacle for natural language processing systems which analyze natural language texts or utterances is the need to identify the entities referred to by means of referring expressions. Among referring expressions, pronouns and definit noun phrases (NPs) are the most prominent.

Supervised machine learning algorithms were used for pronoun resolution with good results (Ge et al., 1998), and for definit NPs with fairly good results (Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Soon et al., 2001). However, the defi ciency of supervised machine learning approaches is the need for an unknown amount of annotated training data for optimal performance.

So, researchers in NLP began to experiment with weakly supervised machine learning algorithms such as Co-Training (Blum and Mitchell, 1998). Among others Co-Training was applied to document classificatio (Blum and Mitchell, 1998), named-entity recognition (Collins and Singer, 1999), noun phrase bracketing (Pierce and Cardie, 2001), and statistical parsing (Sarkar, 2001). In this paper we apply Co-Training to the problem of reference resolution in German texts from the tourism domain in order to provide answers to the following questions:

> Does Co-Training work at all for this task (when compared to conventional C4.5 decision tree learning)?
>
> How much labeled training data is required for achieving a reasonable performance?

First, we discuss features that have been found to be relevant for the task of reference resolution, and describe the feature set that we are using (Section 2). Then we briefl introduce the Co-Training paradigm (Section 3), which is followed by a description of the corpus we use, the corpus annotation, and the way we prepared the data for using a binary classifie in the Co-Training algorithm (Section 4). In Section 5 we specify the experimental setup and report on the results.

2 Features for Reference Resolution

2.1 Previous Work

Driven by the necessity to provide robust systems for the MUC system evaluations, researchers began to look for those features which were particular important for the task of reference resolution. While most features for pronoun resolution have been described in the literature for decades, researchers only recently began to look for *robust* and *cheap* features, i.e., those which perform well over several domains and can be annotated (semi-) automatically. Also, the relative quantitative contribution of each of these features came into focus only after the advent of

corpus-based and statistical methods. In the following, we describe a few earlier contributions with respect to the features used.

Decision tree algorithms were used for reference resolution by Aone and Bennett (1995, C4.5), McCarthy and Lehnert (1995, C4.5) and Soon et al. (2001, C5.0). This approach requires the definitio of a set of training features describing pairs of anaphors and their antecedents. Aone and Bennett (1995), working on reference resolution in Japanese newspaper articles, use 66 features. They do not mention all of these explicitly but emphasize the features *POS-tag*, *grammatical role*, *semantic class* and *distance.* The set of semantic classes they use appears to be rather elaborated and highly domain-dependent. Aone and Bennett (1995) report that their best classifie achieved an F-measure of about 77% after training on 250 documents. They mention that it was important for the training data to contain transitive positives, i.e., all possible coreference relations within an anaphoric chain.

McCarthy and Lehnert (1995) describe a reference resolution component which they evaluated on the MUC-5 English Joint Venture corpus. They distinguish between features which focus on individual noun phrases (e.g. *Does noun phrase contain a name?*) and features which focus on the anaphoric relation (e.g. *Do both share a common NP?*). It was criticized (Soon et al., 2001) that the features used by McCarthy and Lehnert (1995) are highly idiosyncratic and applicable only to one particular domain. McCarthy and Lehnert (1995) achieved results of about 86% F-measure (evaluated according to Vilain et al. (1995)) on the MUC-5 data set. However, only a define subset of all possible reference resolution cases was considered relevant in the MUC-5 task description, e.g., only *entity* references. For this case, the domain-dependent features may have been particularly important, making it difficul to compare the results of this approach to others working on less restricted domains.

Soon et al. (2001) use twelve features (see Table 1). They show a part of their decision tree in which the *weak string identity* feature (i.e. identity after determiners have been removed) appears to be the most important one. They also report on the relative contribution of the features where

– distance in sentences between anaphor and antecedent
– antecedent is a pronoun?
– anaphor is a pronoun?
– weak string identity between anaphor and antecedent
– anaphor is a definit noun phrase?
– anaphor is a demonstrative pronoun?
– number agreement between anaphor and antecedent
– semantic class agreement between anaphor and antecedent
– gender agreement between anaphor and antecedent
– anaphor and antecedent are both proper names?
– an alias feature (used for proper names and acronyms)
– an appositive feature

Table 1: Features used by Soon et al.

the three features *weak string identity, alias* (which maps named entities in order to resolve dates, person names, acronyms, etc.) and *appositive* seem to cover most of the cases (the other nine features contribute only 2.3% F-measure for MUC-6 texts and 1% F-measure for MUC-7 texts). Soon et al. (2001) include all noun phrases returned by their NP identifie and report an F-measure of 62.6% for MUC-6 data and 60.4% for MUC-7 data. They only used pairs of anaphors and their *closest* antecedents as positive examples in training, but evaluated according to Vilain et al. (1995).

Cardie and Wagstaff (1999) describe an unsupervised clustering approach to noun phrase coreference resolution in which features are assigned to single noun phrases only. They use the features shown in Table 2, all of which are obtained automatically without any manual tagging.

– position (NPs are numbered sequentially)
– pronoun type (nom., acc., possessive, ambiguous)
– article (indefinite definite none)
– appositive (yes, no)
– number (singular, plural)
– proper name (yes, no)
– semantic class (based on WordNet: time, city, animal, human, object; based on a separate algorithm: number, money, company)
– gender (masculine, feminine, either, neuter)
– animacy (anim, inanim)

Table 2: Features used by Cardie and Wagstaff

The feature *semantic class* used by Cardie and Wagstaff (1999) seems to be a domain-dependent one which can only be used for the MUC domain and similar ones.

Cardie and Wagstaff (1999) report a performance of 53,6% F-measure (evaluated according to Vilain et al. (1995)).

2.2 Our Features

We consider the features we use for our weakly supervised approach to be domain-independent. We distinguish between features assigned to noun phrases and features assigned to the potential coreference relation. They are listed in Table 3 together with their respective possible values. In the literature on reference resolution it is claimed that the antecedent's grammatical function and its realization are important. Hence we introduce the features *ante_gram_func* and *ante_npform*. The identity in grammatical function of a potential anaphor and antecedent is captured in the feature *syn_par*. Since in German the gender and the semantic class do not necessarily coincide (i.e. objects are not necessarily neuter as in English) we also provide a *semantic-class* feature which captures the difference between *human, concrete*, and *abstract objects*. This basically corresponds to the gender attribute in English. The feature *wdist* captures the distance in words between anaphor and antecedent, the feature *ddist* captures the distance in sentences, the feature *mdist* the number of markables (NPs) between anaphor and antecedent. Features like the *string_ident* and *substring_match* features were used by other researchers (Soon et al., 2001), while the features *ante_med* and *ana_med* were used by Strube et al. (2002) in order to improve the performance for definit NPs. The minimum edit distance (MED) computes the similarity of strings by taking into account the minimum number of editing operations (substitutions *s*, insertions *i*, deletions *d*) needed to transform one string into the other (Wagner and Fischer, 1974). The MED is computed from these editing operations and the length of the potential antecedent *m* or the length of the anaphor *n*.

3 Co-Training

Co-Training (Blum and Mitchell, 1998) is a meta-learning algorithm which exploits unlabeled in addition to labeled training data for classifie learning. A Co-Training classifie is complex in the sense that it consists of two simple classifier (most often Naive Bayes, e.g. by Blum and Mitchell (1998) and Pierce and Cardie (2001)). Initially, these classifier are trained in the conventional way using a small set of size L of labeled training data. In this process, each of the two classifier is trained on a different subset of features of the training data. These feature subsets are commonly referred to as different *views* that the classifier have on the data, i.e., each classifie describes a given instance in terms of different features. The Co-Training algorithm is supposed to bootstrap by gradually extending the training data with self-labeled instances. It utilizes the two classifier by letting them in turn label the p best positive and n best negative instances from a set of size P of unlabeled training data (referred to in the literature as the *pool*). Instances labeled by one classifie are then added to the other's training data, and vice versa. After each turn, both classifier are re-trained on their augmented training sets, and the pool is refille with $(p + n) * 2$ unlabeled training instances drawn at random. This process is repeated either for a given number of iterations I or until all the unlabeled data has been labeled. In particular the defi nition of the two data views appears to be a crucial factor which can strongly influenc the behaviour of Co-Training. A number of requirements for these views are mentioned in the literature, e.g., that they have to be disjoint or even conditionally independent (but cf. Nigam and Ghani (2000)). Another important factor is the ratio between p and n, i.e., the number of positive and negative instances added in each iteration. These values are commonly chosen in such a way as to reflec the empirical class distribution of the respective instances.

4 Data

4.1 Text Corpus

Our corpus consists of 250 short German texts (total 36924 tokens, 9399 NPs, 2179 anaphoric NPs) about sights, historic events and persons in Heidelberg. The average length of the texts was 149 tokens. The texts were POS-tagged using *TnT* (Brants, 2000). A basic identificatio of markables (i.e. NPs) was obtained by using the NP-Chunker *Chunkie* (Skut and Brants, 1998). The POS-tagger was also used for assigning attributes to markables (e.g. the NP form). The automatic annotation was followed by a man-

	Document level features	
1.	doc_id	document number (1 . . . 250)
	NP-level features	
2.	ante_gram_func	grammatical function of antecedent (subject, object, other)
3.	ante_npform	form of antecedent (definit NP, indefinit NP, personal pronoun, demonstrative pronoun, possessive pronoun, proper name)
4.	ante_agree	agreement in person, gender, number
5.	ante_semanticclass	semantic class of antecedent (human, concrete object, abstract object)
6.	ana_gram_func	grammatical function of anaphor (subject, object, other)
7.	ana_npform	form of anaphor (definit NP, indefinit NP, personal pronoun, demonstrative pronoun, possessive pronoun, proper name)
8.	ana_agree	agreement in person, gender, number
9.	ana_semanticclass	semantic class of anaphor (human, concrete object, abstract object)
	Coreference-level features	
10.	wdist	distance between anaphor and antecedent in words (1 . . . n)
11.	ddist	distance between anaphor and antecedent in sentences (0, 1, >1)
12.	mdist	distance between anaphor and antecedent in markables (NPs) (1 . . . n)
13.	syn_par	anaphor and antecedent have the same grammatical function (yes, no)
14.	string_ident	anaphor and antecedent consist of identical strings (yes, no)
15.	substring_match	one string contains the other (yes, no)
16.	ante_med	minimum edit distance to anaphor: $ante\_med = 100 \cdot \frac{m-(s+i+d)}{m}$
17.	ana_med	minimum edit distance to antecedent: $ana\_med = 100 \cdot \frac{n-(s+i+d)}{n}$

Table 3: Our Features

ual correction and annotation phase in which further tags were assigned to the markables. In this phase manual coreference annotation was performed as well. In our annotation, coreference is represented in terms of a *member* attribute on markables (i.e., noun phrases). Markables with the same value in this attribute are considered coreferring expressions. The annotation was performed by two students. The reliability of the annotations was checked using the kappa statistic (Carletta, 1996).

4.2 Coreference resolution as binary classificatio

The problem of coreference resolution can easily be formulated in such a way as to be amenable to Co-Training. The most straightforward definitio turns the task into a binary classification Given a pair of potential anaphor and potential antecedent, classify as positive if the antecedent is in fact the closest antecedent, and as negative otherwise. Note that the restriction of this rule to the closest antecedent means that *transitive* antecedents (i.e. those occuring further upwards in the text as the direct antecedent) are treated as negative in the training data. We favour this definitio because it strengthens the predictive power of the word distance between potential anaphor and potential antecedent (as expressed in the *wdist* feature).

4.3 Test and Training Data Generation

From our annotated corpus, we created one initial training and test data set. For each text, a list of noun phrases in document order was generated. This list was then processed from end to beginning, the phrase at the current position being considered as a potential anaphor. Beginning with the directly preceding position, each noun phrase which appeared before was combined with the potential anaphor and both entities were considered a potential antecedent-anaphor pair. If applied to a text with n noun phrases, this algorithm produces a total of $\frac{n*(n-1)}{2}$ noun phrase pairs. However, a number of filter can reasonably be applied at this point. An antecedent-anaphor pair is discarded

- if the anaphor is an indefinit NP,
- if one entity is embedded into the other, e.g., if the potential anaphor is the head of the potential antecedent NP (or vice versa),
- if both entities have different values in their semantic class attributes[1],

[1]This filte applies only if none of the expressions is a pronoun. Otherwise, filterin on semantic class is not possible be-

- if either entity has a value other than 3rd person singular or plural in its agreement feature,
- if both entities have different values in their agreement features[2].

For some texts, these heuristics reduced to up to 50% the potential antecedent-anaphor pairs, all of which would have been negative cases. We regard these cases as irrelevant because they do not contribute any knowledge for the classifie . After application of these filters the remaining candidate pairs were labeled as follows:

- Pairs of anaphors and their direct (i.e. closest) antecedents were labeled P. This means that each anaphoric expression produced exactly *one* positive instance.
- Pairs of anaphors and their indirect (*transitive*) antecedents were labeled TP.
- Pairs of anaphors and those non-antecedents which occurred *before* the direct antecedent were labeled N. The number of negative instances that each expression produced thus depended on the number of non-antecedents occurring before the direct antecedent (if any).
- Pairs of anaphors and non-antecedents were labeled DN (*distant N*) if at least one true antecedent occurred in between.

This produced 250 data sets with a total of 92750 instances of potential antecedent-anaphor pairs (2074 P, 70021 N, 6014 TP and 14641 DN). From this set the last 50 texts were used as a test set. From this set, all instances with class DN and TP were removed, resulting in a test set of 11033 instances. Removing DNs and TPs was motivated by the fact that initial experimentation with C4.5 had indicated that a four way classificatio gives no advantage over a two way classification In addition, this kind of test set approximates the decisions made by a simple resolution algorithm that looks for an antecedent from the current position upwards until it find one or reaches the beginning. Hence, our results are only indirectly comparable with the ones obtained by an evaluation according to Vilain et al. (1995). However, in this paper we only compare results of this direct binary antecedent-anaphor pair decision.

The remaining texts were split in two sets of 50 resp. 150 texts. From the first our labeled training set was produced by removing all instances with class DN and TP. The second set was used as our unlabeled training set. From this set, no instances were removed because no knowledge whatsoever about the data can be assumed in a realistic setting.

5 Experiments and Results

For our experiments we implemented the standard Co-Training algorithm (as described in Section 3) in Java using the Weka machine learning library[3]. In contrast to other Co-Training approaches, we did not use Naive Bayes as base classifiers but J48 decision trees, which are a Weka re-implementation of C4.5. The use of decision tree classifier was motivated by the observation that they appeared to perform better on the task at hand.

We conducted a number of experiments to investigate the question if Co-Training is beneficia for the task of training a classifie for coreference resolution. In previous work (Strube et al., 2002) we obtained quite different results for different types of anaphora, i.e. if we split the data according to the *ana_np* feature into personal and possessive pronouns (*PPER_PPOS*), proper names (*NE*), and definite NPs (*def_NP*). Therefore we performed Co-Training experiments on subsets of our data define by these NP forms, and on the whole data set.

We determined the features for the two different views with the following procedure: We trained classifier on each feature separately and chose the best one, adding the feature which produced it as the firs feature of view 1. We then trained classifier on all remaining features separately, again choosing the best one and adding its feature as the firs feature of view 2. In the next step, we enhanced the firs classifie by combining it with all remaining features separately. The classifie with the best performance was

cause in a real-world setting, information about a pronoun's semantic class obviously is not available prior to its resolution.

[2]This filte applies only if the anaphor *is* a pronoun. This restriction is necessary because German allows for cases where an antecedent is referred back to by a non-pronoun anaphor which has a different grammatical gender.

[3]http://www.cs.waikato.ac.nz/∼ml/weka

then chosen and its new feature added as the second feature of view 1. We then enhanced the second classifie in the same way by selecting from the remaining features the one that most improved it, adding this feature as the second one of view 2. This process was repeated until no features were left or no significan improvement was achieved, resulting in the views shown in Table 4 (features marked *na* were not available for the respective class). This way we determined two views which performed reasonably well separately.

	PPER PPOS		NE		def_NP		all	
features	1	2	1	2	1	2	1	2
2. ante_gram_func		X		X		X	X	
3. ante_npform	X			X	X		X	
4. ante_agree	X		X			X	X	
5. ante_semanticc.	X			X		X		X
6. ana_gram_func			X			X	X	
7. ana_npform			na		na			X
8. ana_agree			X			X		X
9. ana_semanticc.	na			X	X		na	
10. wdist		X	X			X		X
11. ddist	X		X		X			X
12. mdist		X		X	X		X	
13. syn_par				X	X			X
14. string_ident		X	X			X		X
15. string_match	X			X	X		X	
16. ante_med		X		X	X		X	
17. ana_med		X	X			X		X

Table 4: Views used for the experiments

For Co-Training, we committed ourselves to fi ed parameter settings in order to reduce the complexity of the experiments. Settings are given in the relevant subsections, where the following abbreviations are used: L=size of labeled training set, P/N=number of positive/negative instances added per iteration. All reported Co-Training results are averaged over 5 runs utilizing randomized sequences of unlabeled instances.

We compare the results we obtained with Co-Training with the initial result before the Co-Training process started (zero iterations, both views combined; denoted as *XX_0its* in the plots). For this, we used a conventional C4.5 decision tree classifie (J48 implementation, default settings) on labeled training data sets of the same size used for the respective Co-Training experiment. We did this in order to verify the quality of the training data and for obtaining reference values for comparison with the Co-Training classifiers

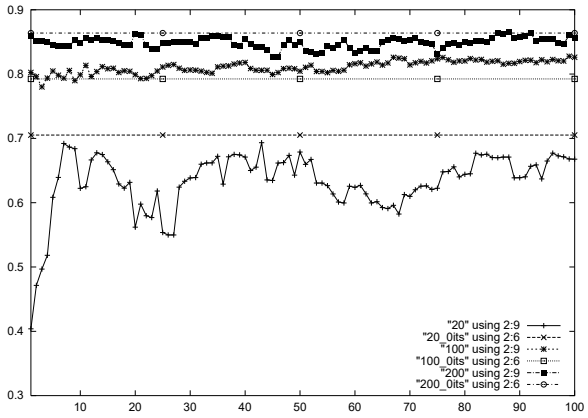

Figure 1: F for *PPER_PPOS* over iterations, baselines

PPER_PPOS. In Figure 1, three curves and three baselines are plotted: For *20* (L=20), *20_0its* is the baseline, i.e. the initial result obtained by just combining the two initial classifiers For *100*, L=100, and for *200*, L=200. The other settings were: P=1, N=1, Pool=10. As can be seen, the baselines slightly outperform the Co-Training curves (except for *100*).

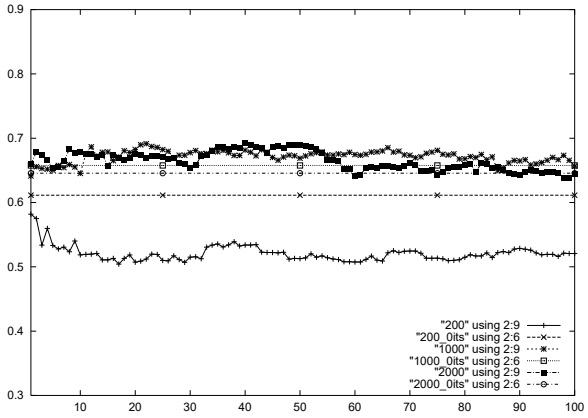

Figure 2: F for *NE* over iterations, baselines

NE. Then we ran the Co-Training experiment with the NP form *NE* (i.e. proper names). Since the distribution of positive and negative examples in the labeled training data was quite different from the previous experiment, we used P=1, N=33, Pool=120. Since all results with L<200 were equally poor, we

started with L=200, where the results were closer to ones of classifier using the whole data set. The resulting Co-Training curve degrades substantially. However, with a training size of 1000 and 2000 the Co-Training curves are above their baselines.

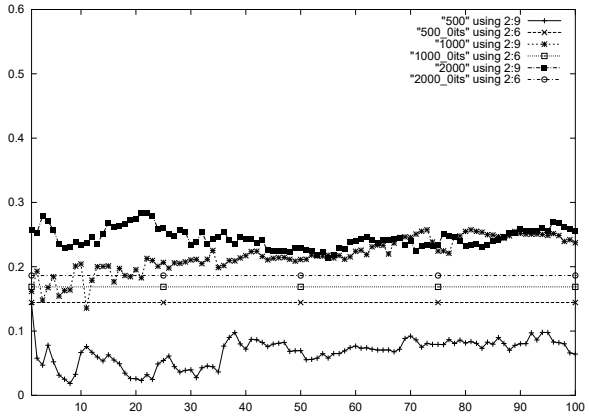

Figure 3: F for *def_NP* over iterations, baselines

def_NP. In the next experiment we tested the NP form *def_NP*, a concept which can be expected to be far more difficul to learn than the previous two NP forms. Used settings were P=1, N=30, Pool=120. For L<500, F-measure was near 0. With L=500 the Co-Training curve is way below the baseline. However, with L=1000 and L=2000 Co-Training does show some improvement.

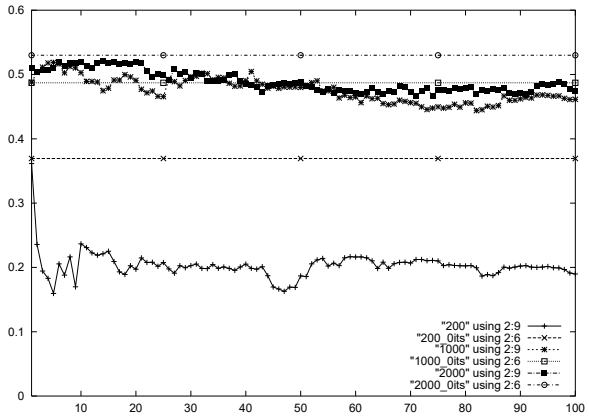

Figure 4: F for *All* over iterations, baselines

All. In the last experiment we trained our classifie on all NP forms, using P=1, N=33, Pool=120. With L=200 the baseline clearly outperforms Co-Training. Co-Training with L=1000 initially rises above the baselines, but then decreases after about 15 to 20 iterations. With L=2000 the Co-Training curve approximates its baseline and then degenerates.

6 Conclusions

Supervised learning of reference resolution classifier is expensive since it needs unknown amounts of annotated data for training. However, reference resolution algorithms based on these classifier achieve reasonable performance of about 60 to 63% F-measure (Soon et al., 2001). Unsupervised learning might be an alternative, since it does not need any annotation at all. However, the cost is the decrease in performance to about 53% F-measure on the same data (Cardie and Wagstaff, 1999) which may be unsuitable for a lot of tasks. In this paper we tried to pioneer a path between the unsupervised and the supervised paradigm by using the Co-Training meta-learning algorithm.

The results, however, are mostly negative. Although we did not try every possible setting for the Co-Training algorithm, we did experiment with different feature views, Pool sizes and positive/negative increments, and we assume the settings we used are reasonable. It seems that Co-Training is useful in rather specialized constellations only. For the classes *PPER_PPOS*, *NE* and *All*, our Co-Training experiments did not yield any benefit worth reporting. Only for *def_NP*, we observed a considerable improvement from about 17% to about 25% F-measure using an initial training set of 1000 labeled instances, and from about 19% to about 28% F-measure using 2000 labeled training instances. In Strube et al. (2002) we report results from other experiments for definit noun phrase reference resolution. Although based on much more labeled training data, these experiments did not yield significantl better results. In this case, therefore, Co-Training seems to be able to save manual annotation work. On the other hand, the definitio of the feature views is non-trivial for the task of training a reference resolution classifie , where no obvious or *natural* feature split suggests itself. In practical terms, therefore, this could outweigh the advantage of annota-

tion work saved.

Another findin of our work is that for personal and possessive pronouns, rather small numbers of labeled training data (about 100) seem to be suffi cient for obtaining classifier with a performance of about 80% F-measure. To our knowledge, this fact has not yet been reported in the literature.

While we restricted ourselves in this work to rather small sets of labeled training data, future work on Co-Training will include further experiments with larger data sets.

Acknowledgments. The work presented here has been partially funded by the German Ministry of Research and Technology as part of the EMBASSI project (01 IL 904 D/2, 01 IL 904 S 8), by Sony International (Europe) GmbH and by the Klaus Tschira Foundation. We would like to thank our annotators Anna Björk Nikulásdôttir, Berenike Loos and Lutz Wind.

References

Chinatsu Aone and Scott W. Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics,* Cambridge, Mass., 26–30 June 1995, pages 122–129.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with Co-Training. In *Proceedings of the 11th Annual Conference on Learning Theory,* Madison, Wisc., 24–26 July, 1998, pages 92–100.

Thorsten Brants. 2000. TnT – A statistical Part-of-Speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing,* Seattle, Wash., 29 April – 4 May 2000, pages 224–231.

Claire Cardie and Kiri Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the 1999 SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora,* College Park, Md., 21–22 June 1999, pages 82–89.

Jean Carletta. 1996. Assessing agreement on classifi cation tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification In *Proceedings of the 1999 SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora,* College Park, Md., 21–22 June 1999, pages 100–110.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora,* Montréal, Canada, pages 161–170.

Joseph F. McCarthy and Wendy G. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificia Intelligence,* Montréal, Canada, 1995, pages 1050–1055.

Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of Co-Training. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, pages pp. 86–93.

David Pierce and Claire Cardie. 2001. Limitations of Co-Training for natural language learning from large datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing,* Pittsburgh, Pa., 3–4 June 2001, pages 1–9.

Anoop Sarkar. 2001. Applying Co-Training methods to statistical parsing. In *Proceedings of the 2nd Conference of the North American Chapter of the Association for Computational Linguistics,* Pittsburgh, Pa., 2–7 June, 2001, pages 175–182.

Wojciech Skut and Thorsten Brants. 1998. A maximum-entropy partial parser for unrestricted text. In *6th Workshop on Very Large Corpora,* Montreal, Canada, pages 143–151.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Michael Strube, Stefan Rapp, and Christoph Müller. 2002. The influenc of minimum edit distance on reference resolution. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing,* Philadelphia, Pa., 6–7 July 2002. To appear.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings fo the 6th Message Understanding Conference (MUC-6)*, pages 45–52, San Mateo, Cal. Morgan Kaufmann.

Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173.