An Approach to Text Corpus Construction which Cuts Annotation Costs and Maintains Reusability of Annotated Data

Katrin Tomanek **Joachim Wermter** **Udo Hahn**
Jena University Language & Information Engineering (JULIE) Lab
Fürstengraben 30
D-07743 Jena, Germany
{tomanek|wermter|hahn}@coling-uni-jena.de

Abstract

We consider the impact Active Learning (AL) has on effective and efficient text corpus annotation, and report on reduction rates for annotation efforts ranging up until 72%. We also address the issue whether a corpus annotated by means of AL – using a particular classifier and a particular feature set – can be re-used to train classifiers different from the ones employed by AL, supplying alternative feature sets as well. We, finally, report on our experience with the AL paradigm under real-world conditions, i.e., the annotation of large-scale document corpora for the life sciences.

1 Introduction

The annotation of corpora has become a crucial prerequisite for NLP utilities which rely on (semi-) supervised machine learning (ML) techniques. While stability, by and large, has been reached for tagsets up until the syntax layer, semantic annotations in terms of (named) entities, semantic roles, propositions, events, etc. reveal a high degree of variability due to the inherent domain-dependence of the underlying tagsets. This diversity fuels a continuous need for creating semantic annotation data anew.

Accordingly, annotation activities will persist and even increase in number as HLT is expanding on various technical and scientific domains (e.g., the life sciences) outside the classical general-language newspaper genre. Since the provision of annotations is a costly, labor-intensive and error-prone process the amount of work and time this activity requires should be minimized to the extent that corpus data could still be used to effectively train ML-based NLP components on them. The approach we advocate does exactly this and yields reduction gains (compared with standard procedures) ranging between 48% to 72%, without seriously sacrificing annotation quality.

Various techniques to minimize the necessary amount of annotated training material have already been investigated. In co-training (Blum and Mitchell, 1998), e.g., from a small initial set of labeled data multiple learners mutually provide new training material for each other by labeling unseen examples. Pierce and Cardie (2001) have shown, however, that for tasks which require large numbers of labeled examples – such as most NLP tasks – co-training might be inadequate because it tends to generate noisy data. Furthermore, a well compiled initial training set is a crucial prerequisite for successful co-training. As another alternative for minimizing annotation work, active learning (AL) is based on the idea to let the learner have control over the examples to be manually labeled so as to optimize the prediction accuracy. Accordingly, AL aims at selecting those examples with high utility for the model.

AL (as well as semi-supervised methods) is typically considered as a learning protocol, i.e., to train a particular classifier. In contrast, we here propose to employ AL as a corpus annotation method. A corpus built on these premises must, however, still be reusable in a flexible way so that, e.g., training with modified or improved classifiers is feasible and reasonable on AL-generated corpora. Baldridge and Osborne (2004) have already argued that this is a highly critical requirement because the examples selected by AL are tuned to one particular classifier. The second major contribution of this paper ad-

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 486–495, Prague, June 2007.

dresses this issue and provides empirical evidence that corpora built with one type of classifier (based on Maximum Entropy) can reasonably be reused by another, methodologically related type of classifier (based on Conditional Random Fields) without requiring changes of the corpus data. We also show that feature sets being used for training classifiers can be enhanced without invalidating corpus annotations generated on the basis of AL and, hence, with a poorer feature set.

2 Related Work

There are mainly two methodological strands of AL research, *viz.* optimization approaches which aim at selecting those examples that optimize some (algorithm-dependent) objective function, such as prediction variance (Cohn et al., 1996), and heuristic methods with uncertainty sampling (Lewis and Catlett, 1994) and query-by-committee (QBC) (Seung et al., 1992) just to name the most prominent ones. AL has already been applied to several NLP tasks, such as document classification (Schohn and Cohn, 2000), POS tagging (Engelson and Dagan, 1996), chunking (Ngai and Yarowsky, 2000), statistical parsing (Thompson et al., 1999; Hwa, 2000), and information extraction (Lewis and Catlett, 1994; Thompson et al., 1999).

In a more recent study, Shen et al. (2004) consider AL for entity recognition based on Support Vector Machines. Here, the informativeness of an example is estimated by the distance to the hyperplane of the currently learned SVM. It is assumed that an example which lies close to the hyperplane has high chances to have an effect on training. This approach is essentially limited to the SVM learning scheme as it solely relies on SVM-internal selection criteria.

Hachey et al. (2005) propose a committee-based AL approach where the committee's classifiers constitute multiple views on the data by employing different feature subsets. The authors focus on (possible) negative side effects of AL on the annotations. They argue that AL annotations are cognitively more difficult to deal with for the annotators (because of the increased complexity of the selected sentences). Hence, lower annotation quality and higher per-sentence annotation times might be a concern.

There are controversial findings on the reusability of data annotated by means of AL for the problem of parse tree selection. Whereas Hwa (2001) reports positive results, Baldridge and Osborne (2004) argue that AL based on uncertainty sampling may face serious performance degradation when labeled data is reused for training a classifier different from the one employed during AL. For committee-based AL, however, there is a lack of work on reusability. Our experiments of committee-based AL for entity recognition, however, reveal that for this task at least, reusability can be guaranteed to a very large extent.

3 AL for Corpus Annotation - Requirements for Practical Use

AL frameworks for real-world corpus annotation should meet the following requirements:

fast selection time cycles — AL-based corpus annotation is an interactive process in which b sentences are selected by the AL engine for human annotation. Once the annotated data is supplied, the AL engine retrains its underlying classifier(s) on *all* available annotations and then re-classifies all unseen corpus items. After that the most informative (i.e., deviant) b sentences from the set of newly classified data are selected for the next iteration round. In this approach the time needed to select the next examples (which is the idle time of the human annotators) has to be kept at an acceptable limit of a few minutes only. There are various AL strategies which – although they yield theoretically near-optimal sample selection – turn out to be actually impractible for real-world use because of excessively high computation times (cf. Cohn et al. (1996)). Thus, AL-based annotation should be based on a computationally tractable and task-wise feasible and acceptable selection strategy (even if this might imply a suboptimal reduction of annotation costs).

reusability — The examples AL selects for manual annotation are dependent on the model being used, up to a certain extent (Baldridge and Osborne, 2004). During annotation time, however, the best model might not be known and model tuning (especially the choice of features) is typically performed once a training corpus is available. Hence, from a practical point of view, the resulting corpus should be reusable with modified classifiers as well.

adaptive stopping criterion — An explicit and adaptive stopping criterion which is sensitive towards the already achieved level of quality of the annotated corpus is clearly preferred over stopping after an *a priori* fixed number of annotation iterations.

If these requirements, especially the first and the second one, cannot be guaranteed for a specific annotation task one should refrain from using AL. The efficiency of AL-driven annotation (in terms of the time needed to compile high quality training material) might be worse compared to the annotation of randomly (or subjectively) selected examples.

4 Framework for AL-based Named Entity Annotation

For named entity recognition (NER), each change of the application domain requires a more or less profound change of the types of semantic categories (tags) being used for corpus annotation. Hence, one may encounter a lack of training material for various relevant (sub)domains. Once this data is available, however, one might want to modify the features of the final classifier with respect to the specific entity types. Thus, a corpus annotated by means of AL has to provide the flexibility to modify the features of the final classifier.

To meet the requirements from above under the constraints of a real-world annotation task, we decided for QBC-based AL, a *heuristic* AL approach, which is computationally less complex and resource-greedy than *objective function* AL methods (the latter explicitly quantify the differences between the current and an ideal classifier in terms of some objective function). Accordingly, we ruled out uncertainty sampling, another heuristic AL approach, because it was shown before that QBC is more efficient and robust (Freund et al., 1997).

QBC is based on the idea to select those examples for manual annotation on which a committee of classifiers disagree most in their predictions (Engelson and Dagan, 1996). A committee consists of a number of k classifiers of the same type (same learning algorithm, parameters, and features) but trained on different subsets of the training data. QBC-based AL is also iterative. In each AL round the committee's k classifiers are trained on the already annotated data C, then a pool of unannotated data P is predicted with each classifier resulting in n automatically labeled versions of P. These are then compared according to their labels. Those with the highest variance are selected for manual annotation.

4.1 Selection Strategy

In each iteration, a batch of b examples is selected for manual annotation. The informativeness of an example is estimated in terms of the *disagreement*, i.e., the uncertainty among the committee's classifiers on classifying a particular example. This is measured by the *vote entropy* (Engelson and Dagan, 1996), i.e., the entropy of the distribution of classifications assigned to an example by the classifiers. Vote entropy is defined on the token level t as:

$$D_{tok}(t) := -\frac{1}{\log k} \sum_{l_i} \frac{V(l_i, t)}{k} \log \frac{V(l_i, t)}{k}$$

where $\frac{V(l_i,t)}{k}$ is the ratio of k classifiers where the label l_i is assigned to a token t. As (named) entities often span more than a single text token we consider complete sentences as a reasonable example size unit[1] for AL and calculate the disagreement of a sentence D_{sent} as the mean vote entropy of its single tokens. Since the vote entropy is minimal when all classifiers agree in their vote, sentences with high disagreement are preferred for manual annotation. With informed decisions of human annotators made available, the potential for future disagreement of the classifier committee on conflicting instances should decrease. Thus, each AL iteration selects the b sentences with the highest disagreement to focus on the most controversial decision problems.

Besides informativeness, additional criteria can be envisaged for the selection of examples, e.g., *di-*

[1]Sentence-level examples are but one conceivable grain size – lower grains (such as clauses or phrases) as well as higher grains (e.g., paragraphs or abstracts) are equally possible, with different implications for the AL process.

feature class	description
orthographical	based on regular expressions (e.g. *HasDash*, *IsGreek*, ...), token transformation rule: capital letters replaced by "A", lowercase letters by "a", digits by "0", etc. (e.g., *IL2* → *AA0*, *have* → *aaaa*)
lexical and morphological	prefix and suffix of length 3, stemmed version of each token
syntactic	the token's part-of-speech tag
contextual	features of neighboring tokens

Table 1: Features used for AL

versity of a batch and *representativeness* of the respective example (to avoid outliers) (Shen et al., 2004). We experimented with these more sophisticated selection strategies but preliminary experiments did not reveal any significant improvement of the AL performance. Engelson and Dagan (1996) confirm this observation that, in general, different (and even more refined) selection methods still yield similar results. Moreover, strategies incorporating more selection criteria often require more parameters to be set. However, proper parametrization is hard to achieve in real-world applications. Using disagreement exclusively for selection requires only one parameter, *viz.* the batch size b, to be specified.

4.2 Classifier and Features

For our AL framework we decided to employ a Maximum Entropy (ME) classifier (Berger et al., 1996). We employ a rich set of features (see Table 1) which are general enough to be used in most (sub)domains for entity recognition. We intentionally avoided using features such as semantic triggers or external dictionary look-ups because they depend a lot on the specific subdomain and entity types being used. However, one might add them to fin- tune the final classifier, if needed. ME classifiers outperform their generative counterparts (e.g., Naïve Bayesian classifiers) because they can easily handle overlapping, probably dependent features which might be contained in rich feature sets. We also favored an ME classifier over an SVM one because the latter is computationally much more complex on rich feature sets and multiple classes and is thus not so well suited for an interactive process like AL.

It has been shown that *Conditional Random Fields* (CRF) (Lafferty et al., 2001) achieve higher performance on many NLP tasks, such as NER, but on the other hand they are computionally more complex than an ME classifier making them also impractical for the interactive AL process. Thus, in our committee we employ ME classifiers to meet requirement 1 (fast selection time cycles). However, in the end we want to use the annotated corpora to train a CRF and will thus examine the reusability of such an ME-annotated AL corpus for CRFs (cf. Subsection 5.2).

4.3 Stopping Criterion

A question hardly addressed up until now is when to actually terminate the AL process. Usually, it gets stopped when the supervized learning performance of the specific classifier is achieved. The problem with such an approach is, however, that in practice one does not know the performance level which could possibly be achieved on an unannotated corpus.

An apparent way to monitor the progress of the annotation process is to periodically (e.g., after each AL iteration) train a classifier on the data annotated so far and evaluate it against some randomly selected gold standard. When the relative performance growth of each AL iteration falls below a certain threshold this might be a good reason to stop the annotation. Though this is probably the most reliable way, it is impractical for many scenarios since assembling and manually annotating a representative gold standard may already be quite a laborious task. Thus, a measure from which we can *predict* the development of the learning curve would be beneficial.

One way to achieve this goal is to monitor the rate of disagreement among the different classifiers after each iteration. This rate will descend as the classifiers get more and more robust in their predictions on unseen data. Thus, an average disagreement approaching zero can be interpreted as an indication that additional annotations will not render any further improvement. In our experiments, we will show that this is a valid stopping criterion, indeed.

5 Experiments and Results

For our experiments, we specified the following three parameters: the batch size b (i.e., the number of sentences to be selected for each AL iteration), the size and composition of the initial training set, and the number of k classifiers in a committee. The smaller the batch size, the higher the AL performance turns out to be. In the special case of batch size of $b = 1$ only that example with the highest disagreement is selected. This is certainly impractical since after each AL iteration a new committee of classifiers has to be trained causing unwarranted annotation idle time. We found $b = 20$ to be a good compromise between the annotators' idle time and AL performance. The initial training set also contains 20 sentences which are randomly selected though. Our committee consists of $k = 3$ classifiers, which is a good trade-off between computational complexity and diversity. Although the AL iterations were performed on the sentence level, we report on the number of annotated tokens. Since sentences may considerably vary in their length the number of tokens constitutes a better measure for annotation costs.

We ran our experiments on two common entity-annotated corpora from two different domains (see Table 2). From the general-language newspaper domain, we used the English data set of the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003). It consists of a collection of newswire articles from the Reuters Corpus,[2] which comes annotated with three entity types: *persons*, *locations*, and *organizations*. From the sublanguage biology domain we used the oncology part of the PENNBIOIE corpus which consists of some 1150 PubMed abstracts. Originally, this corpus contains gene, variation event, and malignancy entity annotations. Manual annotation after each AL round was simulated by moving the selected sentences from the pool of unannotated sentences P to the training corpus T. For our simulations, we built two subcorpora by filtering out entity annotations: the PENNBIOIE gene corpus (PBgene), including the three gene entity subtypes *generic*, *protein*, and *rna*, and the PENNBIOIE variation events corpus (PBvar) corpus including the variation entity subtypes *type*, *event*, *location*, *state-altered*, *state-generic*, and *state-original*. We split all three corpora into two subsets, *viz.* AL simulation data and gold standard data on which we evaluate[3] a classifier in terms

[2] http://trec.nist.gov/

[3] We use a strict evaluation criterion which only counts exact matches as true positives because annotations having incorrect boundaries are insufficient for manual corpus annotation.

corpus	data set	sentences	tokens
CONLL	AL	14,040	203,617
3 entities	Gold	3,453	46,435
PBGENE	AL	10,050	249,490
3 entities	Gold	1,114	27,563
PBVAR	AL	10,050	249,490
6 entities	Gold	1,114	27,563

Table 2: Corpora used in the Experiments

of f-score trained on the annotated corpus after each AL iteration (learning curve). As far as the CoNLL corpus is concerned, we have used CoNLL's training set for AL and CoNLL's test set as gold standard. As for PBgene and PBvar, we randomly split the corpora into 90% for AL and 10% as gold standard.

In the following experiments we will refer to the classifiers used in the AL committee as *selectors*, and the classifier used for evaluation as the *tester*.

5.1 Efficiency of AL and the Applicability of the Stopping Criterion

In a first series of experiments, we evaluated whether AL-based annotations can significantly reduce the human effort compared to the standard annotation procedure where sentences are selected randomly (or subjectively). We also show that disagreement is an accurate stopping criterion. As described in Section 4.2, we here employed a committee of ME classifiers for AL; a CRF was used as tester for both the AL and the random selection. Figures 1, 2, and 3 depict the learning curves for AL selection and random selection (upper two curves) and the respective disagreement curves (lower curve). The random selection curves contained in these plots are averaged over three random selection runs.

With AL, we get a maximum f-score of $\approx 84.5\%$ on the CoNLL corpus after about 118,000 tokens. At about the same number of tokens the disagreement curve drops down to values of around $D_{sent} = 0$. Comparing AL and random selection, an f-score of $\approx 84\%$ is reached after 86,000 and 165,000 tokens, respectively, which means a reduction of annotation costs of about 48%. On PBgene, the effect of AL is comparable: a maximum value of 83.5% f-score is reached first after about 124,000 tokens, a data point where hardly any disagreement between the committee's classifiers occurs. For, e.g., an f-score of

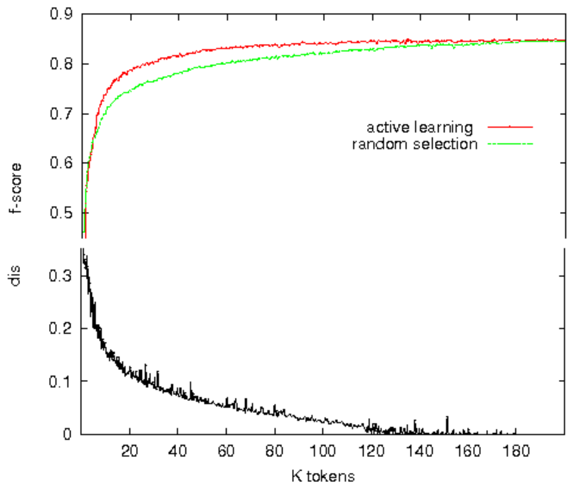

Figure 1: CoNLL Corpus: Learning/Disagreement Curves

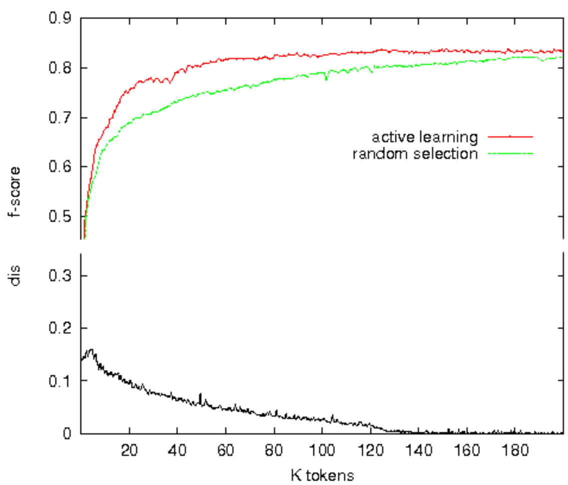

Figure 2: PBgene Corpus: Learning/Disagreement Curves

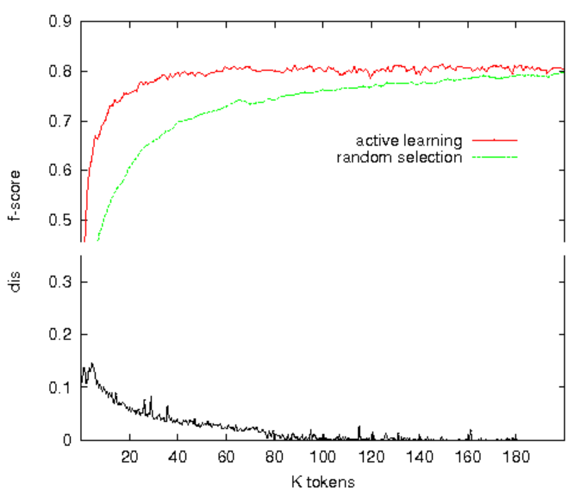

Figure 3: PBvar Corpus: Learning/Disagreement Curves

corpus	selection	F	tokens	reduction
CoNLL	random	84.0	165,000	
	AL	84.0	86,000	$\approx$ 48%
PBGENE	random	83.0	101,000	
	AL	83.0	213,000	$\approx$ 53%
PBVAR	random	80.0	56,000	
	AL	80.0	200,000	$\approx$ 72%

Table 3: Reduction of Annotation Costs Achieved with AL-based Annotation

83%, the annotation effort can be reduced by about 53% using AL. On PBvar, an f-score of about 80% is reached after $\approx$ 56,000 tokens when using AL selection, while 200,000 tokens are needed with random selection. For this task, AL reduces the annotation effort by of 72%. Here, the disagreement curve approaches values of zero after approximately 80,000 tokens. At about this point the learning curve reaches its maximum of about 81% f-score. Table 3 summarizes the reduction of annotation costs achieved on all three corpora.

Comparing both PENNBIOIE simulations, obviously, the reduction of annotation costs through AL is much higher for the variation type entities than for the gene entities. We hypothesize this to be mainly due to incomparable entity densities. Whereas the gene entities are quite frequent (about 1.3 per sentence on average), the variation entities are rather sparse (0.62 per sentence on average) making it an ideal playground for AL-based annotation. Our experiments also reveal that disagreement approaching values of zero is a valid stopping criterion. This is, under all circumstances, definitely the point when AL-based annotation *should* stop because then all classifiers of the committee vote consistently. Any further selection – even though AL selection is used – is then, actually, a *random* selection. If, due to reasons whatsoever, further annotations are wanted, a direct switch to random selection is advisable because this is computationally less expensive than AL-based selection.

5.2 Reusability

To evaluate whether the proposed AL framework for named entity annotation allows for flexible re-use of the annotated data, we performed experiments where we varied both the learning algorithms and the features of the selectors.

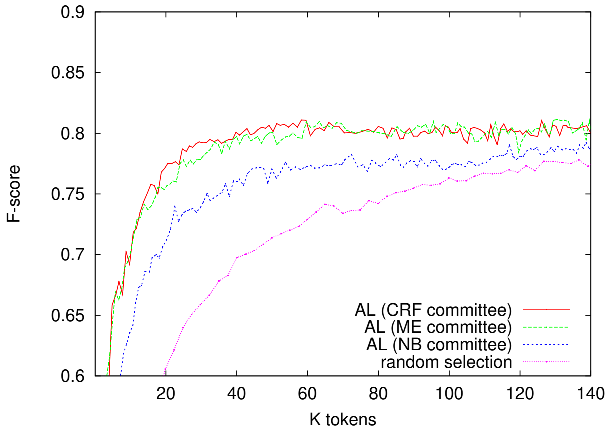

Figure 4: Algorithm Flexibility on PBvar

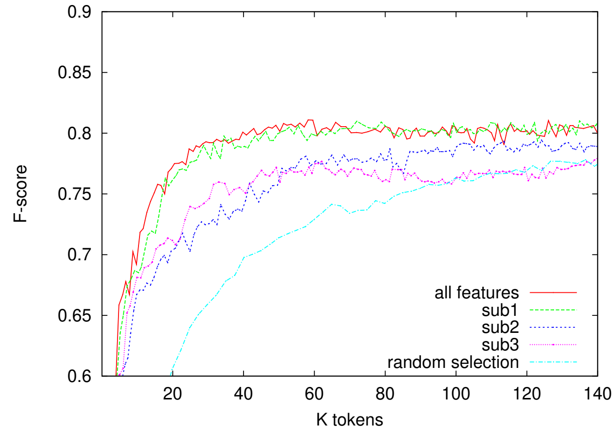

Figure 6: Feature Flexibility on PBvar

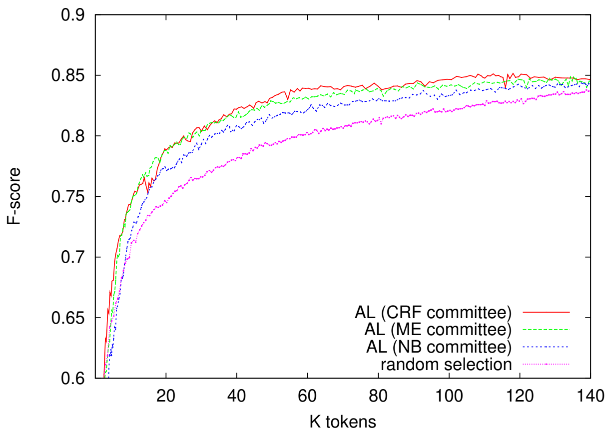

Figure 5: AlgorithmFflexibility on CoNLL

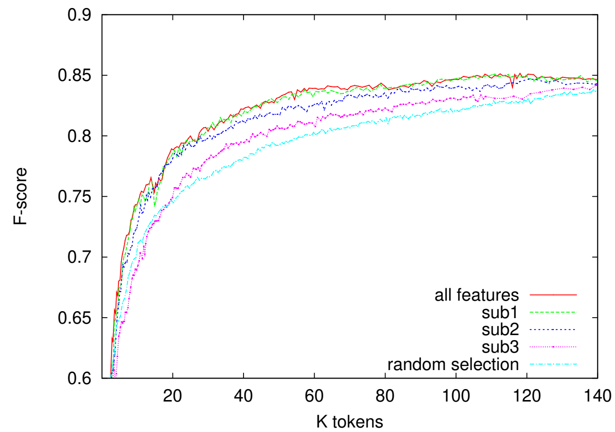

Figure 7: Feature Flexibility on ConLL

First, we analyzed the effect of different probabilistic classifiers as selectors on the resulting learning curve of the CRF tester. Figures 4 and 5 show the learning curves on our original ME committee, a CRF committee, and also a committee of Naïve Bayes (NB) classifiers. It is not surprising that self-reuse (CRF selectors and CRF tester) yields the best results. Switching from CRF selectors to ME selectors has almost no negative effect. Even with a comittee of NB selectors (an ML approach which is essentially less well suited for the NER task), AL-based selection is still substantially more efficient than random selection on both corpora. This shows that our approach to use the less complex ME classifiers for the AL selection process has the positive effect of fast selection cycle times at almost no costs. This is especially interesting as the performance of an ME classifier trained in supervized manner on the complete corpus is significantly worse (several percentage points of f-measure) than a CRF. That means, even though an ME classifier is less well suited as the final classifier, it works well as a selector for CRFs.[4]

Second, we ran experiments on selectors with only some features and our CRF tester with all features (cf. Table 1). Feature subset 1 (*sub1*) contains all but the syntactic features. In the second subset (*sub2*), also morphological and lexical features are missing. The third set (*sub3*) only contains orthographical features. We ran an AL simulation for

[4]We have also conducted experiments where we varied the learning algorithms of the tester (we experimented with NB, ME, MEMM, and CRFs) – with comparable results. In a realistic scenario, however, on would rather choose a CRF as final tester over, e.g., a NB.

each feature subset with a committee of CRF selectors.[5] Figures 6 and 7 show the various learning curves. Here we see that a corpus that was produced with AL on *sub1* can easily be re-used by a tester with little more features. This is probably the most realistic scenario: the core features are kept and only a few specific features (e.g., POS, a dictionary look-up, chunk information, etc.) are added. When adding substantially more features to the tester than were available during AL time, the respective learning curves drop down towards the learning curve for random selection. But even with a selector which has only orthographical features and a tester with many more features – which is actually quite an extreme example and a rather unrealistic scenario for a real-world application – AL is more efficient than random selection. However, the limits of reusability are taking shape: on PBvar, the AL selection with *sub3* converges with the random selection curve after about 100,000 tokens.

5.3 Findings with Real AL Annotation

We currently perform AL entity mention annotations for an information extraction project in the biomedical subdomain of immunogenetics. For this purpose, we retrieved about 200,000 abstracts ($\approx$ 2,000,000 sentences) as our document pool of unlabeled examples from PUBMED. By means of random subsampling, only about 40,000 sentences are considered in each round of AL selection. To regularly monitor classifier performance, we also perform gold standard (GS) annotations on 250 randomly chosen abstracts ($\approx$ 2,200 sentences). In all our annotations of different entity types so far, we found AL learning curves similar to the ones reported in our simulation experiments, with classifier performance levelling off at around 75% - 85% f-score (depending on the entity type).

Our annotations also reveal that AL is especially beneficial when entity mentions are very sparse. Figure 8 shows the cumulated entity density on AL and gold standard annotations of cytokine receptors (specialized proteins for which we annotated six different entity subtypes) – very sparse entity types with less than one entity mention per PUBMED abstract on the average. As can be seen, after 2,000

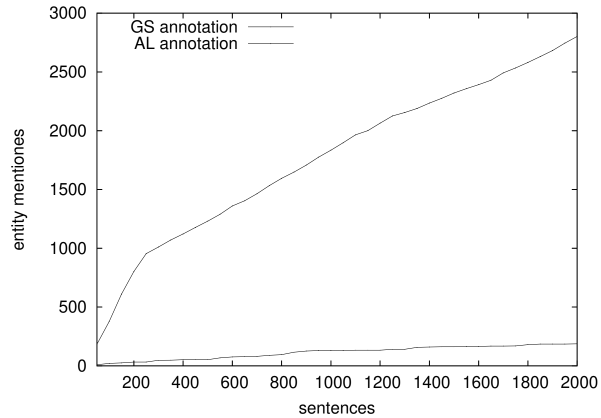

Figure 8: Cumulated Entity Density on AL and GS Annotations of Cytokine Receptors

sentences the entity density in our AL corpus is almost 15 times higher than in our GS corpus. Such a dense corpus may be more appropriate for classifier training than a sparse one yielded by random or sequential annotations, which may just contain lots of negative training examples. We have observed comparable effects with other entity types, too, and thus conclude that the sparser entity mentions of a specific type are in texts, the more beneficial AL-based annotation is. We report on other aspects of AL for real annotation projects in Tomanek et al. (2007).

6 Discussion and Conclusions

We have shown, for the annotation of (named) entities, that AL is well-suited to speed up annotation work under realistic conditions. In our simulations we yielded gains (in the number of tokens) up to 72%. We collected evidence that an average disagreement approaching zero may serve as an adaptive stopping criterion for AL-driven annotation and that a corpus compiled by means of QBC-based AL is to a large extent reusable by modified classifiers.

These findings stand in contrast to those supplied by Baldridge and Osborne (2004) who focused on parse selection. Their research indicates that AL on selectors with different learning algorithms and feature sets then used by the tester can easily get worse than random selection. They conclude that it might not be be advisable to employ AL in environments where the final classifier is not very stable.

Our evidence leads us to a re-assessment of AL-

[5] Here, we employed CRF instead of ME selectors to isolate the effect of feature re-usability.

based annotations. First, we employed a committee-based (QBC) while Baldridge and Osborne performed uncertainty sampling AL. Committee-based approaches calculate the uncertainty on an example in a more implicit way, i.e., by the disagreement among the committee's classifiers. With uncertainty sampling, however, the labeling uncertainty of one classifier is considered directly. In future work we will directly compare QBC and uncertainty sampling with respect to data reusability. Second, whereas Baldridge and Osborne employed AL on a scoring or ranking problem we focused on classification problems. Further research is needed to investigate whether the problem class (classification with a fixed and moderate number of classes vs. ranking large numbers of possible candidates) is responsible for limited data reusability.

On the basis of our experiments we stipulate that the proposed AL approach might be applicable with comparable results to a wider range of corpus annotation tasks, which otherwise would require substantially larger amounts of annotation efforts.

Acknowledgements

This research was funded by the EC within the BOOTStrep project (FP6-028099), and by the German Ministry of Education and Research within the StemNet project (01DS001A to 1C).

References

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In Dekang Lin and Dekai Wu, editors, *EMNLP 2004 – Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 9–16. Barcelona, Spain, July 25-26, 2004. Association for Computational Linguistics.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT'98 – Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100. Madison, Wisconsin, USA, July 24-26, 1998. New York, NY: ACM Press.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *Journal of Artifical Intelligence Research*, 4:129–145.

Sean Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *ACL'96 – Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 319–326. University of California at Santa Cruz, California, U.S.A., 24-27 June 1996. San Francisco, CA: Morgan Kaufmann.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168.

Ben Hachey, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *CoNLL-2005 – Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 144–151. Ann Arbor, MI, USA, June 2005. Association for Computational Linguistics.

Rebecca Hwa. 2000. Sample selection for statistical grammar induction. In *EMNLP/VLC-2000 – Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 45–52. Hong Kong, China, October 7-8, 2000.. Association for Computational Linguistics.

Rebecca Hwa. 2001. On minimizing training corpus for parser acquisition. In Walter Daelemans and Rémi Zajac, editors, *CoNLL-2001 – Proceedings of the 5th Natural Language Learning Workshop*. Toulouse, France, 6-7 July 2001. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML-2001 – Proceedings of the 18th International Conference on Machine Learning*, pages 282–289. Williams College, MA, USA, June 28 - July 1, 2001. San Francisco, CA: Morgan Kaufmann.

David D. Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *ICML '94: Proceedings of the 11th International Conference on Machine Learning*, pages 148–156. San Francisco, CA: Morgan Kaufmann.

Grace Ngai and David Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *ACL'00 – Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 117–125. Hong Kong, China, 1-8 August 2000. San Francisco, CA: Morgan Kaufmann.

David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In Lillian Lee and Donna Harman, editors, *EMNLP 2001 – Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1–9. Pittsburgh, PA, USA, June 3-4, 2001. Association for Computational Linguistics.

Greg Schohn and David Cohn. 2000. Less is more: Active learning with support vector machines. In *ICML '00: Proceedings of the 17th International Conference on Machine Learning*, pages 839–846. San Francisco, CA: Morgan Kaufmann.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *COLT'92 – Proceedings of the 5th Annual Conference on Computational Learning Theory*, pages 287–294. Pittsburgh, PA, USA, July 27-29, 1992. New York, NY: ACM Press.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew Lim Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *ACL'04 – Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 589–596. Barcelona, Spain, July 21-26, 2004. San Francisco, CA: Morgan Kaufmann.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *ICML '99: Proceedings of the 16th International Conference on Machine Learning*, pages 406–414. Bled, Slovenia, June 1999. San Francisco, CA: Morgan Kaufmann.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *CoNLL-2003 – Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 142–147. Edmonton, Canada, 2003. Association for Computational Linguistics.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. 2007. Efficient annotation with the Jena ANnotation Environment (JANE). In *Proceedings of the ACL 2007 'Linguistic Annotation Workshop – A Merger of NLPXML 2007 and FLAC 2007'*. Prague, Czech Republic, June 28-29, 2007. Association for Computational Linguistics (ACL).