On Proper Unit Selection in Active Learning: Co-Selection Effects for Named Entity Recognition

Katrin Tomanek[1*] **Florian Laws**[2*] **Udo Hahn**[1] **Hinrich Schütze**[2]

[1]Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena, Germany
{katrin.tomanek|udo.hahn}@uni-jena.de

[2]Institute for Natural Language Processing, Universität Stuttgart, Germany
{fl|hs999}@ifnlp.org

Abstract

Active learning is an effective method for creating training sets cheaply, but it is a biased sampling process and fails to explore large regions of the instance space in many applications. This can result in a missed cluster effect, which signficantly lowers recall and slows down learning for infrequent classes. We show that missed clusters can be avoided in sequence classification tasks by using sentences as natural multi-instance units for labeling. Co-selection of other tokens within sentences provides an implicit exploratory component since we found for the task of named entity recognition on two corpora that entity classes co-occur with sufficient frequency within sentences.

1 Introduction

Active learning (AL) has been shown to be an effective approach to reduce the amount of data needed to train an accurate statistical classifier. AL selects highly informative examples from a pool of unlabeled data and prompts a human annotator for the labels of these examples. The newly labeled examples are added to a training set used to build a statistical classifier. This classifier is in turn used to assess the informativeness of further examples. Thus, a select-label-retrain loop is formed that quickly selects hard to classify examples, honing in on the decision boundary (Cohn et al., 1996).

A fundamental characteristic of AL is the fact that it constitutes a biased sampling process. This is so by design, but the bias can have an undesirable consequence: partial coverage of the instance space. As a result, classes or clusters within classes may be completely missed, resulting in low recall or slow learning progress. This has been called the *missed cluster effect* (Schütze et al., 2006). While AL has been studied for a range of NLP tasks, the missed cluster problem has hardly been addressed.

This paper studies the *missed class effect*, a special case of the missed cluster effect where complete classes are overlooked by an active learner. The missed class effect is the result of insufficient exploration before or during a mainly exploitative AL process. In AL approaches where exploration is only addressed by an initial seed set, poor seed set construction gives rise to the missed class effect.

We focus on the missed class effect in the context of a common NLP task: named entity recognition (NER). We show that for this task the missed class effect is avoided by increasing the sampling granularity from single-instance units (i.e., tokens) to multi-instance units (i.e., sentences). For AL approaches to NER, sentence selection recovers better from unfavorable seed sets than token selection due to what we call the *co-selection effect*. Under this effect, a non-targeted entity class co-occurs in sentences that were originally selected because of uncertainty on tokens of a different entity class.

The rest of the paper is structured as follows: Section 2 introduces the missed class effect in detail. Experiments which demonstrate the co-selection effect achieved by sentence selection for NER are described in Section 3 and their results presented in Section 4. We draw conclusions in Section 5.

[*] Both authors contributed equally to this work.

Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing, pages 9–17,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

2 The Missed Class Effect

This section first describes the missed class effect. Then, we discuss several factors influencing this effect, focusing on co-selection, a natural phenomenon in common NLP applications of AL.

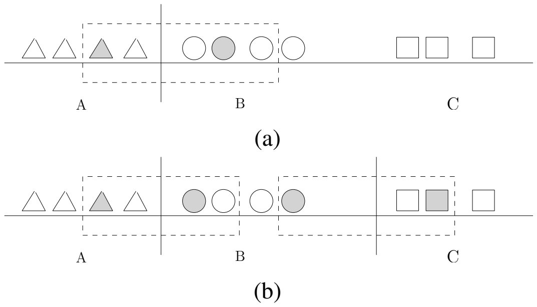
2.1 Sampling bias and misguided AL

The distribution of the labeled data points obtained with an active learner deviates from the true data distribution. While this sampling bias is intended and accounts for the effectiveness of AL, it also poses challenges as it leads to classifiers that perform poorly in some regions, or *clusters*, of the example space. In the literature, this phenomenon has been described as the *missed cluster effect* (Schütze et al., 2006; Dasgupta and Hsu, 2008)

In this context, we must distinguish between exploration and exploitation. By design, AL is a highly exploitative strategy: regions around decision boundaries are inspected thoroughly so that decision boundaries are learned well, but regions far from any of the initial decision boundaries remain unexplored.

An exploitative sampling approach thus has to be combined with some kind of exploratory strategy to make sure the example space is adequately covered. A common approach is to start an AL process with an initial seed set that accounts for the exploration step. However, a seed set which is not representative of the example space may completely misguide AL — at least when no other explorative techniques are applied as a remedy. While approaches to balancing exploration and exploitation (Baram et al., 2003; Dasgupta and Hsu, 2008; Cebron and Berthold, 2009) have been discussed, we here focus on a "pure" AL scenario where exploration takes only place in the beginning by a seed set. In summary, the missed clusters are the result of a scenario where poor exploration is combined with exclusively exploitative sampling.

Why is AL an exploitative sampling strategy? AL selects data points based on the confidence of the active learner. Assume an initial seed set that does not contain examples of a specific cluster. This leads to an initial active learner that is mistakenly overconfident about the class membership of instances in this missed cluster. Far away from the decision boundary, the active learner assumes a high confidence for all instances in that cluster, even if they are in fact misclassified. Consequently, the active learner will fail to select these instances for long until some redirection impulse is received (if at all).



(a)

(b)

Figure 1: Illustration of the missed cluster effect in a 1-d scenario. Shaded points are contained in the seed set, vertical lines are final decision boundaries, and dashed rectangles mark the explored regions

To give an example, let us consider a simple 1-d toy scenario with examples from three clusters A, B, and C as shown in Figure 1. In scenario *(a)*, AL is started from a seed set including one example of clusters A and B only. In subsequent rounds, AL will select examples in these clusters only (shown as the dashed box in the figure). Examples in cluster C are ignored as they are far from the initial decision boundary. Eventually, a decision boundary is fixed as shown by the vertical line which indicates that this AL process has completely overlooked examples from cluster C.

Assuming that the examples fall in two classes $X_1 = \{A \cup C\}$ and $X_2 = \{B\}$ the learned classifier has low recall for class X_1 and relatively low precision for class X_2 as it erroneously assigns examples of cluster C to class X_2. In a related scenario with three classes $X_1 = \{A\}$, $X_2 = \{B\}$, and $X_3 = \{C\}$ this would even mean that the classifier is not at all aware about the third class resulting in the *missed class* problem.

A more representative seed set circumvents this problem. Given a seed set including one example of each cluster, AL might find a second decision boundary[1] between clusters B and C because it is now aware of examples from C. Figure 1(b) shows a possible result of AL on this seed set.

The missed cluster effect can be understood as the generalized problem. A special case of it is the

[1] Assuming a classifier that can learn several boundaries.

missed class effect as shown in the previous example. In general, it has the same causes (insufficient exploration and misguided exploitation), but is easier to test. Often we know (at least the number of) all classes under scrutiny, while we usually cannot assume all clusters in the feature space to be known. In this paper, we focus on the missed class effect, i.e., scenarios where classes are overlooked by a misguided AL process resulting in a slow (active) learning progress.

2.2 Factors influcencing the missed class effect

AL in a practical scenario is subject to several factors which mitigate or intensify the missed class effect described before. In the following, we describe three such factors, with a special focus on the co-selection effect, which we claim to significantly mitigate the missed class effect in a specific type of NLP tasks, sequence learning problems such as NER or POS tagging.

Class imbalance Many studies on AL for NLP tasks assume that AL is started from a randomly drawn seed set. Such a seed set can be problematic when the class distribution in the data is highly skewed. In this case, “rare” classes might not be represented in the seed set, increasing the chance to completely miss out such a class using AL. When classes are relatively frequent, an active learner — even when started from an unfavorable seed set — might still mistake an example of one class for an uncertain example of a different class and consequently select it. Thereby, it can acquire information about the former class “by accident” leading to sudden and rapid discovery of the newly-found class. However, in the case of extreme class imbalance this is very unlikely. Severe class imbalance intensifies the missed cluster effect.

Similarity of considered classes If, e.g., two of the classes to be learned, say X_i and X_j, are harder to discriminate than others, or if the data contains lots of noise, an active learner is more likely to select some instances of X_i if at least its “similar” counterpart X_j was represented in the seed set. Hence, it may mistake the instances of X_i and X_j before it has acquired enough information to discriminate between them. So, under certain situations similarity of classes can mitigate the missed class effect.

The co-selection effect Many NLP tasks are sequence learning problems including, e.g., POS tagging, and named entity recognition. Sequences are consecutive text tokens constituting linguistically plausible chunks, e.g., sentences. Algorithms for sequence learning obviously work on sequence data, so respective AL approaches need to select complete sequences instead of single text tokens (Settles and Craven, 2008). Furthermore, sentence selection has been preferred over token selection in other works with the argument that the manual annotation of single, possibly isolated tokens is almost impossible or at least extremely time-consuming (Ringger et al., 2007; Tomanek et al., 2007).

Within such sequences, instances of different classes often co-occur. Thus, an active learner that selects uncertain examples of one class gets examples of a second class as an unintended, yet positive side effect. We call this the *co-selection effect*. As a result, AL for sequence labeling is not “pure” exploitative AL, but implicitly comprises an exploratory aspect which can substantially reduce the missed class problem. In scenarios where we cannot hope for such a co-selection, we are much more likely to have decreased AL performance due to missed clusters or classes.

3 Experiments

We ran several experiments to investigate how the sampling granularity, i.e. the size of the selection unit, influences the missed class effect. AL based on token selection (T-AL) is compared to AL based on sentence selection (S-AL). Although our experiments are certainly also subject to the other factors mitigating the missed class effect (e.g. similarity of classes), the main focus of the experiments is on the co-selection effect that we expected to observe in S-AL. Several scenarios of initial exploration were simulated by seed sets of different characteristics. The experiments were run on synthetic and real data in the context of named entity recognition (NER).

3.1 Classifiers and active learning setup

The active learning approach used for both S-AL and T-AL is based on uncertainty sampling (Lewis and Gale, 1994) with the *margin* metric (Schein and Ungar, 2007) as uncertainty measure. Let c and c'

be the two most likely classes predicted for token x_j with $\hat{p}_{c,x_j}$ and $\hat{p}_{c',x_j}$ being the associated class probabilities. The per-token margin is calculated as $M = |\hat{p}_{c,x_j} - \hat{p}_{c',x_j}|$.

For T-AL, the sampling granularity is the token, while in S-AL, complete sentences are selected. For S-AL, the margins of all tokens in a sentence are averaged and the aggregate margin is used to select sentences. We chose this uncertainty measure for S-AL for better comparison with T-AL. In either case, examples (tokens or sentences) with a small margin are preferred for selection. In every iteration, a batch of examples is selected: 20 sentences for S-AL, 200 tokens for T-AL.

Bayesian logistic regression as implemented in the BBR classification package (Genkin et al., 2007) with out-of-the-box parameter settings was used as base learner for T-AL. For S-AL, a linear-chain Conditional Random Field (Lafferty et al., 2001) is employed as implemented in MALLET (McCallum, 2002). Both base learners employ standard features for NER including the lexical token itself, various orthographic features such as capitalization, the occurrence of special characters like hyphens, and context information in terms of features of neighboring tokens to the left and right of the current token.

3.2 Data sets

We used three data sets in our experiments. Two of them (ACE and PBIO) are standard data sets. The third (SYN) is a synthetic set constructed to have specific characteristics. For simplicity, we consider only scenarios with two entity classes, a majority class (MAJ) and a minority class (MIN). We discarded all other entity annotations originally contained in the corpus assigning the OUTSIDE class.[2]

The first data set (PBIO) is based on the annotations of the PENNBIOIE corpus for biomedical entity extraction (Kulick et al., 2004). As PENNBIOIE makes fine-grained and subtle distinctions between various subtypes of classes irrelevant for this study, we combined several of the original classes into two entity classes: The majority class consists of the three original classes 'gene-protein', 'gene-generic', and 'gene-rna'. The minority class consists of the original and similar classes 'variation-type' and 'variation-event'. All other entity labels were replaced by the OUTSIDE class.

The second data set (ACE) is based on the newswire section of the ACE 2005 Multilingual Training Corpus (Walker et al., 2006). We chose the 'person' class as majority class and the 'organization' class as the minority class. Again, all other classes are mapped to OUTSIDE.

The synthetic data set (SYN) was constructed by combining the sentences from the original ACE and PENNBIOIE corpora. The 'person' class constitutes the minority class, the very similar classes 'malignancy' and 'malignancy-type' were merged to form the majority class. All other class labels were set to OUTSIDE. SYN's construction was motivated by the following characteristics of the new data set which would make the appearance of the missed class effect very likely for insufficient exploration scenarios:
(i) absence of inner-sentence entity class correlation to ensure that sentences contain either mentions of only a single entity class or no mentions at all.
(ii) marked entity class imbalance between the majority and minority classes
(iii) dissimilar surface patterns of entity mentions of the two entity classes with the rationale that class similarity will be low.

Table 1 summarizes characteristics of the data sets. While SYN exhibits high imbalance (e.g., 1:9.4 on the token level), PBIO and ACE are moderately skewed. In PBIO, the number of sentences containing any entity mention is relatively high compared to ACE or SYN. For our experiments, the corpora were randomly split in a pool for AL and a test set for performance evaluation.

Inner-sentence entity class co-occurrence We have described co-selection as a potential mitigating factor for the missed class effect in Section 2. For this effect to occur, there must be some correlation between the occurrence of entity mentions of the MAJ class with those from MIN.

Table 2 shows correlation statistics based on the χ^2 measure. We found strong correlation in all three corpora[3]: For ACE and PBIO, the correlation is positive; for SYN it is negative so when a sentence in SYN contains a majority class entity mention, it is

[2] The OUTSIDE class marks that a token is not part of an named entity.

[3] All correlations are statistically significant ($p < 0.01$).

	PBio	Ace	Syn
sentences (all)	11,164	2,642	13,804
sentences (MAJ)	7,075	767	5,667
sentences (MIN)	2,156	974	974
MIN-MAJ ratio	1 : 3.3	1 : 1.3	1 : 5.8
tokens (all)	277,053	66,752	343,773
tokens (MAJ)	17,928	2,008	18,959
tokens (MIN)	4,079	1,822	2,008
MIN-MAJ ratio	1 : 4.4	1 : 1.1	1 : 9.4

Table 1: Characteristics of the data sets; "sentences (MAJ)", e.g., specifies the number of sentences containing mentions of the majority class.

	PBio	Ace	Syn
χ^2	132.34	6.07	727
$P(MIN\|MAJ)$	0.26	0.31	0.0

Table 2: Co-occurrence of entity classes in sentences

highly unlikely that it also contains a minority entity. In fact, it is impossible by construction of the data set. Further, this table shows the probability that a sentence containing the majority class also contains the minority class. As expected, this is exactly 0 for Syn, but significantly above 0 for PBio and Ace.

3.3 Seed sets

Selection of an appropriate seed set for the start of an AL process is important to the success of AL. This is especially relevant in the case of imbalanced classes because a typically small random sample will possibly not contain any example of the rare class. We constructed different types of seed sets (whose naming intentionally reflects the use of the entity classes from Section 3.2) to simulate different scenarios of ill-managed initial exploration. All seed sets have a size of 20 sentences. The *RANDOM* set was randomly sampled, the *MAJ* set is made of sentences containing at least one majority class entity, but no minority class entity. Accordingly, *MIN* is densely populated with minority entities. Finally, *OUTSIDE* contains only sentences without entity mentions.

One could think of the OUTSIDE and MAJ seed sets of cases where a random seed set selection has unluckily produced an especially bad seed set. MIN serves to demonstrate the opposite case. For each type of seed set, we sampled ten independent versions to calculate averages over several AL runs.

3.4 Cost measure

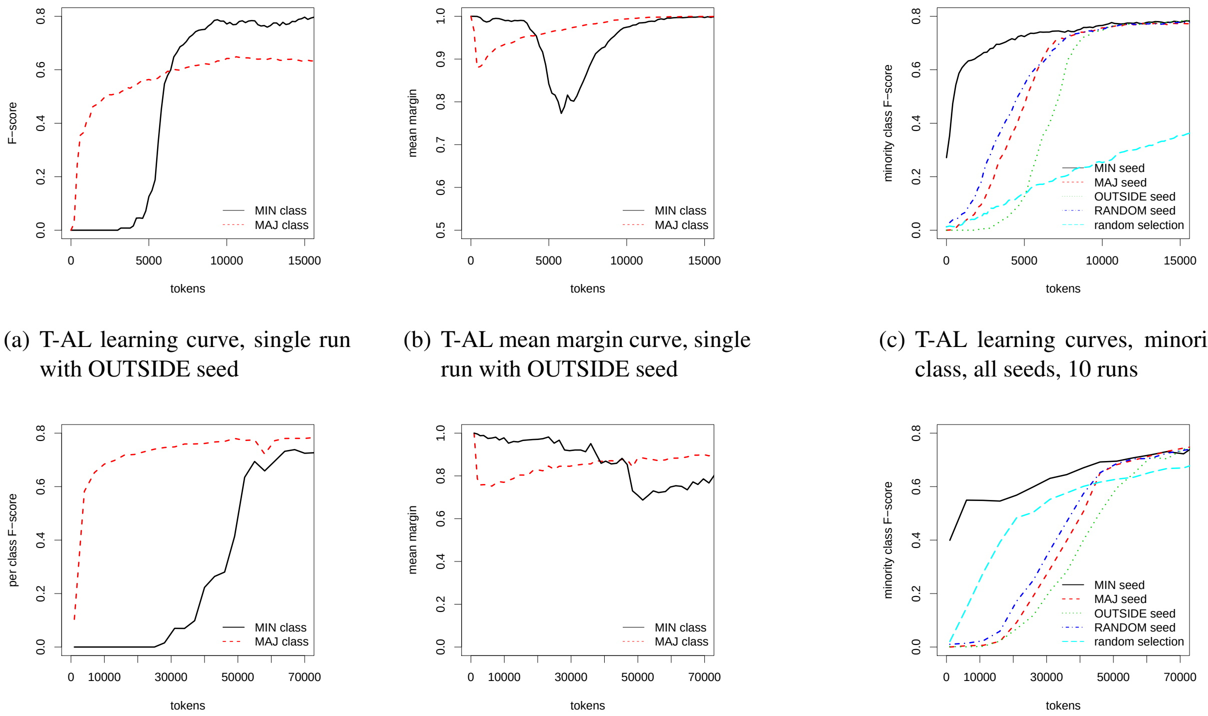
The success of AL is usually measured as reduction of annotation effort according to some cost measure. Traditionally, the most common cost measure considers a unit cost per annotated token, which favors AL systems that select individual tokens. In a real annotation setting, however, it is unnatural, and therefore hard for humans to annotate single, possibly isolated tokens, leading to bad annotation quality (Hachey et al., 2005; Ringger et al., 2007). When providing context, the question arises whether the annotator can label several tokens present in the context (e.g., an entire multi-token entity or even the whole sentence) at little more cost than annotating a single token. Thus, assigning a linear cost of n to a sentence where n is the sentence's length in tokens seems to unfairly disadvantage sentence-selection AL setups.

However, more work is needed to find a more realistic cost measure. At present there is no other generally accepted cost measure than unit cost per token, so we report costs using the token measure.

4 Results

This section presents the results of our experiments on the missed class effect in two different AL scenarios, i.e., sentence selection (S-AL) and token selection (T-AL). The AL runs were stopped when convergence on the minority class F-score was achieved. This was done because early AL iterations before the convergence point are most important and representative for a real-life scenario where the pool is extremely large, so that absolute convergence of the classifier's performance will never be reached.

The learning curves in Figures 2, 3, and 4 reveal general characteristics of S-AL compared to T-AL. For S-AL, the number of tokens on the x-axis is the total number of tokens in the sentences labeled so far. While S-AL generally yields higher F-scores, T-AL converges much earlier when counted in terms of tokens. The reason for this is that T-AL can select uncertain data more specifically. In contrast, S-AL also selects tokens that the classifier can already classify reliably – these tokens are selected because they co-occur in a sentence that also contains an uncertain token. Whether T-AL is really more efficient clearly depends on the cost-metric applied (cf. Sec-



(a) T-AL learning curve, single run with OUTSIDE seed

(b) T-AL mean margin curve, single run with OUTSIDE seed

(c) T-AL learning curves, minority class, all seeds, 10 runs

(d) S-AL learning curve, single run with OUTSIDE seed

(e) S-AL mean margin curve, single run with OUTSIDE seed

(f) S-AL learning curves, minority class, all seeds, 10 runs

Figure 2: Results on SYN corpus for token selection (a,b,c) and sentence selection (d,e,f)

tion 3.4). Since the focus of this paper is on comparing the missed class effect in a sentence and a token selection AL setting (T-AL and S-AL) we apply the straight-forward token measure.

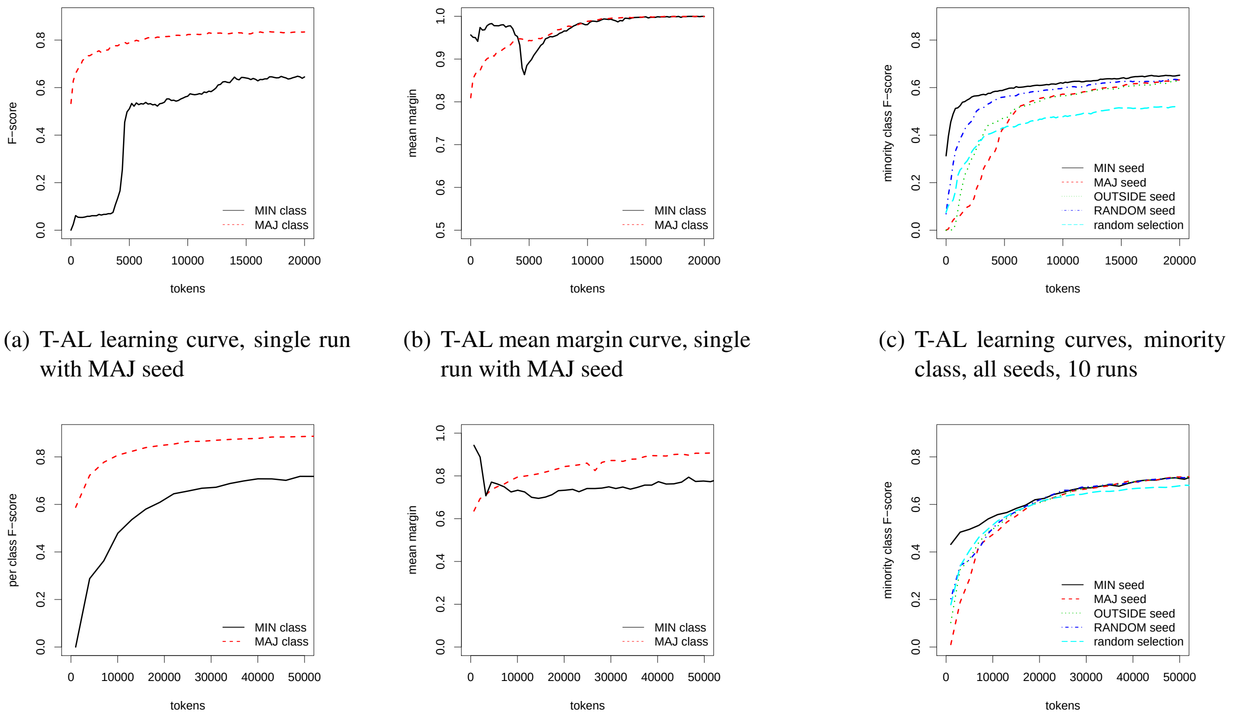
4.1 The pathological case

Figure 2 shows results on the SYN corpus for T-AL (upper row) and S-AL (lower row). Figures 2(a) and 2(d) show the minority and majority class learning curves for a single run starting from the OUTSIDE seed set, which was particularly problematic on SYN. (We show single runs to give a better picture of what happens during the selection process.) The figures show that for both AL scenarios, the OUTSIDE seed set caused the active learner to focus exclusively on the majority class and to completely ignore the minority class for many AL iterations (almost 30,000 tokens for S-AL and over 4,000 tokens for T-AL). Had we stopped the AL process before this turning point, the classifier's performance on the majority entity class would have been reasonably high while the minority class would not have been learned at all — which is precisely the definition of an (initially) missed class.

Figures 2(b) and 2(e) show the corresponding mean margin plots of these AL runs, indicating the confidence of the classifier on each class. The mean margin is calculated as the average margin over tokens in the remaining pool, separately for each true class label.[4] As expected, the active learner is overconfident but wrong on instances of the minority class (assigning them to the OUTSIDE class, we assume). Only after some time, margin scores on minority class tokens start decreasing. This happens because from time to time minority class examples are mistakenly considered as majority class examples with low confidence and thus selected by accident. Lowered minority class confidence then causes the selection of further minority class examples, resulting in a turning point with a steep slope of the minority class learning curve.

Consequences of seed set selection We compare the minority class learning curves for all types of

[4] Note that in a real, non-simulation active learning task, the true class labels would be unknown.



(a) T-AL learning curve, single run with MAJ seed

(b) T-AL mean margin curve, single run with MAJ seed

(c) T-AL learning curves, minority class, all seeds, 10 runs

(d) S-AL learning curve, single run with MAJ seed

(e) S-AL mean margin curve, single run with MAJ seed

(f) S-AL learning curves, minority class, all seeds, 10 runs

Figure 3: Results on PBIO corpus for token selection (a,b,c) and sentence selection (d,e,f)

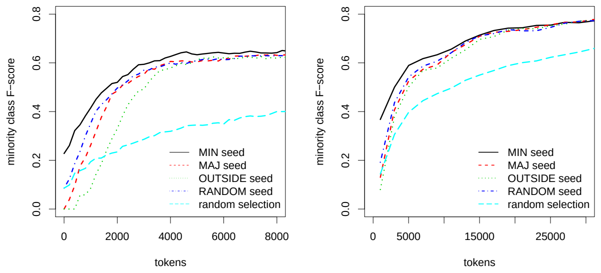
seed sets and for random selection (cf. Figures 2(c) and 2(f)), now averaged over 10 runs. On S-AL all but the MIN seed set were inferior to random selection. Even the commonly used random seed set selection is problematic because the minority class is so rare that there are random seed sets without any example of the minority class.

On T-AL, all seed sets are better than random selection. This, however, is because random selection is an extremely weak baseline for T-AL due to the token distribution (cf. Table 1). Still, the RANDOM, MAJ, and OUTSIDE seed sets are significantly worse than a seed set which covers the minority class well. Note that the majority class learning curves are relatively invariant against different seed sets. The minority class seed set does have some negative impact on initial learning progress on the majority class (not shown here), but the impact is rather small. Because of the higher frequency of the majority class, the classifier soon finds majority class examples to compensate for the seed set by chance or class similarity.

4.2 Missed class effect mitigated by co-selection

Results on PBIO corpus On the PBIO corpus, where minority and majority class entity mentions naturally co-occur on the sentence level, we get a different picture. Figure 3 shows the learning (3(a), 3(d)) and mean margin (3(b), 3(e)) curves for the MAJ seed set. T-AL still exhibits the missed class effect on this seed set. The minority class learning curve again has a delayed slope and high mean margin scores of minority tokens at the beginning, resulting in insufficient selection and slow learning. S-AL, on the other hand, does not really suffer from the missed class effect: minority class entity mentions are co-selected in sentences which were chosen due to uncertainty on majority class tokens. Minority class mean margin scores quickly fall, reinforcing selection for minority class entities. Learning curves for minority and majority classes run approximately in parallel.

Figure 3(f) shows that all seed sets perform quite similar for S-AL. MIN unsurprisingly is a bit better. With the other seed sets, S-AL performance is com-



(a) T-AL

(b) S-AL

Figure 4: Minority class learning curves for all seeds on ACE averaged over 10 runs

parable to random selection. On the PBIO corpus, random selection is a strong baseline as almost every sentence contains an entity mention — which is not the case for SYN and ACE (cf. Table 1). As there is no co-selection effect for T-AL, the MAJ and OUTSIDE seed sets also here are subject to the missed class problem (Figure 3(c)), although not as severely as on the SYN corpus.

Results on ACE corpus Figure 4 shows learning curves averaged over 10 runs on ACE. Overall, the missed class effect is less pronounced on ACE compared to PBIO. Still, co-selection avoids a good portion of the missed class effect on S-AL — all seed sets yield results much better than random selection right from the beginning.

On T-AL, the OUTSIDE seed set has a marked negative effect. However, while different seed sets still have visible differences in learning performance, the magnitude of the effect is smaller than on PBIO. It is difficult to find the exact reasons in a non-synthetic, natural language corpus where a lot of different effects are intermingled. One might assume higher class similarity between the majority ("persons") and the minority ("organizations") classes on the ACE corpus than, e.g., on the PBIO corpus. Moreover, there is hardly any imbalance in frequency between the two entity classes on the ACE corpus. We briefly discussed such influencing factors possibly mitigating the missed class effect in Section 2.2.

4.3 Discussion

To summarize, on a synthetic corpus (SYN) the missed class effect can be well studied in both AL scenarios, i.e., S-AL and T-AL. Moving from a relatively controlled, synthetic corpus (extreme class imbalance, no inner-sentence co-occurrence between entity classes, quite different entity classes) to more realistic corpora, effects generally mix a bit due to different degrees of class imbalance and probably higher similarity between entity classes.

Our experiments unveil that co-selection in S-AL effectively helps avoid dysfunctional classifiers that insufficiently explore the instance space due to a disadvantageous seed set. In contrast, AL based on token-selection (T-AL) cannot recover from insufficient exploration as easy as AL with sentence-selection and is thus more sensitive to the missed class effect.

5 Conclusion

We have shown that insufficient exploration in the initial stages of active learning gives rise to regions of the sample space that contain missed classes that are incorrectly classified. This results in low classification performance and slow learning progress. Comparing two sampling granularities, tokens *vs.* sentences, we found that the missed class effect is more severe when isolated tokens instead of sentences are selected for labeling.

The missed class problem in sequence classification tasks can be avoided using sentences as natural multi-instance units for selection and labeling. Using multi-instance units, co-selection of other tokens within sentences provides an implicit exploratory component. This solution is effective if classes cooccur sufficiently within sentences which is the case for many real-life entity recognition tasks.

While other work has proposed sentence selection in AL for sequence labeling as a means to ease and speed up annotation, we have gathered here additional motivation from the perspective of robustness of learning. Future work will compare the beneficial effect introduced by co-selection with other forms of exploration-enabled active learning.

Acknowledgements

The first and the third author were funded by the German Ministry of Education and Research within the StemNet project (01DS001A-C) and by the EC within the BOOTStrep project (FP6-028099).

References

Yoram Baram, Ran El-Yaniv, and Kobi Luz. 2003. Online choice of active learning algorithms. In *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, pages 19–26.

Nicolas Cebron and Michael R. Berthold. 2009. Active learning for object classification: From exploration to exploitation. *Data Mining and Knowledge Discovery*, 18(2):283–299.

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

Sanjoy Dasgupta and Daniel Hsu. 2008. Hierarchical sampling for active learning. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pages 208–215.

Alexander Genkin, David D. Lewis, and David Madigan. 2007. Large-scale Bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304.

Ben Hachey, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *CoNLL '05: Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 144–151.

Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan T. McDonald, Martha S. Palmer, and Andrew Ian Schein. 2004. Integrated annotation for biomedical information extraction. In *Proceedings of the HLT-NAACL 2004 Workshop 'Linking Biological Literature, Ontologies and Databases: Tools for Users'*, pages 61–68.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12.

Andrew McCallum. 2002. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`.

Eric Ringger, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi, and Deryle Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop at ACL-2007*, pages 101–108.

Andrew Schein and Lyle Ungar. 2007. Active learning for logistic regression: An evaluation. *Machine Learning*, 68(3):235–265.

Hinrich Schütze, Emre Velipasaoglu, and Jan Pedersen. 2006. Performance thresholding in practical text classification. In *CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 662–671.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP '08: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 486–495.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. *ACE 2005 Multilingual Training Corpus*. Linguistic Data Consortium, Philadelphia.