Reducing Class Imbalance during Active Learning for Named Entity Annotation

Katrin Tomanek **Udo Hahn**
Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena, Germany
{katrin.tomanek|udo.hahn}@uni-jena.de

ABSTRACT
In lots of natural language processing tasks, the classes to be dealt with often occur heavily imbalanced in the underlying data set and classifiers trained on such skewed data tend to exhibit poor performance for low-frequency classes. We introduce and compare different approaches to reduce class imbalance by design within the context of active learning (AL). Our goal is to compile more balanced data sets up front during annotation time when AL is used as a strategy to acquire training material. We situate our approach in the context of named entity recognition. Our experiments reveal that we can indeed reduce class imbalance and increase the performance of classifiers on minority classes while preserving a good overall performance in terms of macro F-score.

Categories and Subject Descriptors
I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; I.2.7 [**Computing Methodologies**]: Artificial Intelligence—*Natural Language Processing*

General Terms
Algorithms, Design, Experimentation, Performance

1. INTRODUCTION
The use of supervised machine learning has become a standard technique for many tasks in natural language processing (NLP). One of the main problems with this technique is its greediness for *a priori* supplied annotation metadata. The active learning (AL) paradigm [4] offers a promising solution to deal with this demand efficiently. Unlike random selection of training instances, AL biases the selection of examples which have to be manually annotated such that the human labeling effort be minimized. This is achieved by selecting examples with (presumably) high utility for the classifier training. AL has already been shown to meet these expectations in a variety of NLP tasks [6, 10, 15, 19].

Machine learning (ML) approaches, however, often face a problem with skewed training data, in particular, when it is drawn from already imbalanced ground data. This primary bias can be observed for many NLP tasks such as named entity recognition (NER). Here, imbalance between the different entity classes occurs especially when semantically general classes (such as person names) are split into more fine-grained and specific ones (actors, politicians, sportsmen, etc.). Since rare information carries the potential to be particularly useful and interesting, performance might then be tuned – to a certain extent – in favor of minority classes at the danger of penalizing the overall outcome.

Class imbalance and the resulting effects in learning classifiers from skewed data have been intensively studied in recent years. Common ways to cope with skewed data include different re-sampling strategies and cost-sensitive learning [11, 3, 5]. It has been argued that AL can also be used to leverage class imbalance: The class imbalance ratio of data points close to the decision boundaries is typically lower than the imbalance ratio in the complete data set [7] so that AL provides the learner with more balanced classes. The focus of this paper is whether this natural characteristic of AL can be intensified to obtain even more balanced data sets.

We compare four approaches to reduce the class imbalance up front during AL-driven data acquisition for NER. Section 2 contains a brief sketch of our approach to AL for NER and in Section 3 we present four alternative ways to reduce class imbalance during AL. Related work is discussed in Section 4. We experimentally evaluate the methods under scrutiny on two data sets from the biomedical domain (Section 5) and discuss the results in Section 6.

2. AL FOR NER
AL is a selective sampling technique where the learning protocol is in control of the data to be used. The goal of AL is to learn a good classifier with minimal human labeling effort. The class labels for examples which are considered most useful for the classifier training are queried iteratively from an oracle – typically a human annotator. In our sce-

K-CAP'09, September 1–4, 2009, Redondo Beach, California, USA.

Algorithm 1 Committee-based Active Learning

Given:

L: set of labeled examples, P: set of unlabeled examples

Algorithm:

loop until stopping condition is met

1. generate a committee C of classifiers $c_1, ..., c_k$: sample subset $L_i \subset L$ and train classifier c_i on it, $i \in [1, k]$
2. let each classifier c_i predict labels for all $e \in P$, yielding classifications $c_{e_1}, \ldots, c_{e_k}$
3. calculate disagreement $D_e(c_{e_1}, \ldots, c_{e_k})$ for all $e \in P$
4. select n examples $e \in P$ with highest D_e for annotation
5. move the annotated examples from P to L

nario, the usefulness of examples is computed by a committee of classifiers [8]. Algorithm 1 describes this selection routine in a generalized form.

NER is often considered a sequence-labeling task where each unit of a sequence is assigned a class label. Such units are typically tokens in a sentence. If a token is not part of an entity, the OUTSIDE class is assigned, otherwise the respective entity class label. Being a reasonable linguistic level for annotation, we consider sentences as proper example granularity so that AL selects complete sentences.

Our approach to AL for NER [19] saved between 50% and 70% of annotation effort (measured by the number of annotated tokens) when evaluated on well-known named entity corpora from the news-paper and the biomedical domain. A committee of size $|C| = 3$ is employed and each committee member c_i is trained on a randomly drawn subset L_i, with $|L_i| = \frac{2}{3}|L|$. The vote entropy (VE) is used as disagreement metric [6] to estimate the usefulness of an example. On the token-level it is defined as

$$VE_{tok}(t) = -\sum_{j=1}^{|K|} \frac{V(k_j, t)}{|C|} \log \frac{V(k_j, t)}{|C|}$$

where $V(k_j, t)$ denotes the number of committee members voting for class $k_j \in K$ on token t. The usefulness of a complete sentence is defined as the arithmetic mean over the single token-level VE values.

The final goal classifier for which we want to create training material is based on Conditional Random Fields (CRFs) [13]. CRFs exhibit a high performance on the NER task. We employ default features for NER including orthographic and morphological patterns and lexical, part-of-speech and context information (cf. [19] for more details on the feature set.). As CRFs suffer from high training complexity we instead use Maximum Entropy (ME) classifiers [1] in the AL committee. Thus, AL selection time is significantly reduced. We have shown before, that examples selected by a committee of ME classifiers is well suited to train a CRF [19].

3. REDUCING CLASS IMBALANCE

For simplicity, we develop and test our approaches in the context of a two-entity-class scenario where the less frequent entity class is called *minority class* and the more frequent one *majority class*.[1] We here focus exclusively on the distribution of the entity classes in terms of class imbalance rather than on the OUTSIDE class, which, of course, is also learned and predicted by the NE tagger. The *entity class ratio* is defined as the number of majority class entity mentions divided by the number of minority class entity mentions. As our AL approach to NER selects whole sentences it can be considered a multi-instance selection scenario in which a selection example consists of several tokens each requiring its own class label. Thus, unlike most class imbalance studies where "pure" minority or majority class examples were considered, our examples simultaneously contain mentions of minority *and* majority class entities. In the following, we describe four methods to balance the entity class ratio up front during the AL selection.

3.1 Selection Focus on Minority Class

To focus on a specific entity class during the AL selection, AL is run so that it is only aware of the minority entity class and the OUTSIDE class. Hence, the usefulness of a sentence is estimated using binary classifiers for these two classes only. Once selected, sentences are still annotated with respect to both the minority and the majority entity class. *Minority class-focused AL* (AL-MINOR) can be expected to work well for the minority class, but a penalty is likely to occur for the majority class. Yet, sentences containing minority class entity mentions, in many cases, also contain majority class entity mentions so that training material for the majority class will also be available as a side effect – a phenomenon we already studied as the *co-selection effect* [18].

3.2 Re-Sampling

Re-sampling strategies, including over- and under-sampling, are common practice to tackle the class imbalance problem when passively learning from skewed data sets [16]. Both methods can also be applied in a straightforward manner during AL selection: After the manual annotation step in each AL iteration, either examples for the minority class are over-sampled (e.g., by simple replication), or examples of the majority class are discarded to achieve a more balanced entity class ratio in the sample. Under-sampling is apparently disadvantageous in the AL annotation scenario: After having spent human labeling effort on the selected sentences in an AL iteration, some of these are immediately discarded in the next step. Over-sampling, on the other hand, comes at no extra costs. For word sense disambiguation, Zhu and Hovy [20] have shown that AL combined with over-sampling can significantly increase the performance.

We over-sample on the sentence-level. In each AL iteration, all selected sentences which contain at least one minor-

[1] Although discussed in a two-entity-class scenario, all approaches can be generalized to scenarios with more than two entity classes.

ity class entity mention are duplicated. In a multi-instance scenario it is hard to achieve a specific, predefined class ratio between the minority and the majority class as sentences which we over-sample might also contain entity mentions of the majority class. Still, experiments show that the entity class ratio is shifted in favor of the minority class. This approach is called *oversampling during AL* (AL-OVER).

3.3 Altering the AL Selection Scheme

AL is started from a seed set of labeled examples (see example set L outside the loop in Algorithm 1). A seed set is usually a small random sample. Given high entity class imbalance, a randomly sampled seed set might contain not a single minority class entity mention. However, a seed set containing also information on the minority entity class might help guide the AL selection towards the less frequent class by "announcing" this class properly. We performed preliminary experiments with a dense (containing many entity mentions) and balanced (entity class ratio of 1) seed set. These experiments showed that AL with such a seed set performed only well in early iterations, but then fell back to the performance of AL with a random seed set. So, seed set modification alone is not a sustained remedy to class imbalance during AL data acquisition. Instead, we propose two ways to modify the AL selection scheme directly: *Balanced-batch AL* (AL-BAB) and *AL with boosted disagreement* (AL-BOOD).

Our unmodified AL approach selects a batch of n sentences with the highest disagreement in each AL iteration. Such a selection might contain no or only few entity mentions of the minority class. We alter the AL selection procedure so that the batch of selected sentences is as balanced as possible. With the sentences sorted in descending order by their usefulness score, balanced-batch AL then greedily chooses n sentences from a window W of the $n * f$ most useful sentences so that the entity class ratio is kept close to 1. Obviously, not every AL iteration can achieve a completely balanced batch. The maximum level of balance achievable depends on the size of the window W and on the number of minority/majority class examples contained therein. Choosing a large window results in largely ignoring the committee's disagreement, while an overly narrow window size can be too restrictive. We set $f = 5$ after experimental validation.

A disadvantage of balanced-batch AL is that it does not take into account the disagreement values within the window: Any example within this window which helps optimizing the entity class ratio of the batch can be selected, irrespective of its actual disagreement. To explicitly consider disagreement values of single examples, AL with boosted disagreement makes use of a modified disagreement function VE'_{tok} having a class-specific boosting factor $b_j \geq 1$ for each class k_j:

$$VE'_{tok}(t) = -\sum_{j=1}^{|K|} b_j \frac{V(k_j,t)}{|C|} \log \frac{V(k_j,t)}{|C|}$$

The votes on a specific class k_j are given more importance by setting $b_j > 1$. A boosting factor of $b_j = 1$ does not affect the disagreement, while a higher value of b for the minority class accounts for our intuition that a token where at least one committee member predicted the minority class should be considered more useful and thus result in a higher disagreement value. Moreover, the less certain the committee is that a token should be labeled as minority class (i.e., fewer committee members voting for this class) the more this boosting factor affects the disagreement value on that token. If all committee members agree on the class of a token, the disagreement is 0, irrespective of the chosen boosting factor.

As a default boosting factor b_{min} for the minority class, we choose the entity class ratio divided by the average number of tokens per minority class entity mention. The normalization by the entity length is reasonable, since the boosting factor applies to the token level, while the entity class ratio is calculated with reference to the entity mention level. In our two-entity-class scenario, the majority and the OUTSIDE class are not boosted, i.e., $b_{maj} = b_{outside} = 1$.

4. RELATED WORK

Approaches to AL are mostly based on variants of uncertainty sampling [14] or the Query-by-Committee framework [4]. AL is increasingly gaining attention by the NLP community[2] and has already been applied to a wide range of NLP tasks including, amongst others, part-of-speech tagging, chunking, statistical parsing, word sense disambiguation, and named entity recognition [17, 15, 10, 20, 19].

There is a vast body of literature on the class imbalance problem in the ML community. Common ways to cope with skewed data include re-sampling strategies such as under-/over-sampling or generative approaches [11, 3] and cost-sensitive learning [5]. AL could be shown to be capable of reducing class imbalance because it selects data points near the decision boundaries and so provides the learner with more balanced classes [7]. Class imbalance is typically addressed in scenarios where (large) amounts of fully labeled examples are readily available. Our scenario is different in that we start from unlabeled data and use AL to select the examples to be labeled by a human annotator. Hence, we aim at acquiring preferably balanced data sets by addressing class imbalance up front during the process of selecting and annotating training data.

There is little work on the combination of AL and remedies to class imbalance at annotation time. Zhu and Hovy [20] combine AL and re-sampling strategies, including under- and over-sampling, for word sense disambiguation. In each AL iteration, examples are selected on the basis of a default AL scheme. Accordingly, either examples of the majority class are discarded, or examples of the minority class are replicated. While the authors report that under-sampling is inefficient because some examples are directly discarded after they were manually labeled beforehand, positive evi-

[2] As evidenced most recently by the NAACL-HLT 2009 Workshop on "Active Learning for Natural Language Processing" (`http://nlp.cs.byu.edu/alnlp/`).

	MAL	TF
sentences	11,164	4,629
tokens	277,053	139,600
OUTSIDE tokens	257,173	136,266
majority class tokens	18,962	3,152
minority class tokens	918	182
majority class entities	9,321	2,776
minority class entities	604	179
entity class ratio	15.43	15.51

Table 1: Quantitative data of the simulation corpora

dence for over-sampling combined with AL was found. While Zhu and Hovy only consider re-sampling techniques, we study different approaches to address class imbalance during the AL selection process. Most importantly, we modify the AL selection mechanism so that examples of the minority class are given a higher chance to be selected.

Recently, Bloodgood and Shanker [2] proposed a method to address class imbalance in AL with cost-weighted SVMs. The general idea is similar to AL with boosted disagreement. Class-specific cost factors are derived from the class imbalance ratio observed on a small random data sample. Similarly, we derive the boosting factors from the entity ratio divided by the average length of minority class entity mentions. While the approach of Bloodgood and Shanker requires a cost-sensitive classifier, ours is a wrapper approach which can be applied to any classifier.

5. EXPERIMENTAL SETTINGS

We performed experiments on corpora from the biomedical domain where entity classes often are more fine-grained and class imbalance between entities occurs in a more pronounced way than in the newspaper material. We focus on scenarios with *two* entity classes only, namely one majority and one minority entity class. Our first data set (MAL) is based on the annotations of the PENNBIOIE corpus [12]. We regrouped PENNBIOIE's original annotations with malignancy entity classes into two entity classes: The majority class combines the original classes 'malignancy-type' and 'malignancy', the minority class comprises all entity classes of malignancy stages. All other entity annotations of this corpus (e.g., genes and variation events) were removed. Our second data set (TF) is based on the GENEREG corpus which is annotated with genes involved in the regulation of gene expression [9]. We removed all entity mentions except those labeled as 'transcription factor' (majority class) and 'transcription cofactor' (minority class). Table 1 summarizes the characteristics of both data sets. While the MAL corpus is significantly larger than TF, both data sets have approximately the same entity class imbalance ratio of 15.5.

In each AL iteration, $n = 25$ sentences are selected according to the chosen AL protocol variant (cf. Section 3). AL is started with a randomly drawn seed set of 25 sentences. We performed 30 independent runs for each protocol on each data set. In each run, we randomly split the data set into a pool from which AL selects and a gold standard for evaluation. On the MAL data set, 90% of the sentences are used as the AL pool and 10% for evaluation. Due to the smaller size of the TF data set, 30% of the sentences were used for evaluation to obtain a reasonable coverage of minority class entity mentions in the gold standard. Our result figures show averages over all 30 runs. As cost metric for annotation effort, we consider the number of tokens being annotated.

The performance of AL protocols is usually measured by the F-score in relation to the annotation costs. We compute the F-scores for each entity class separately and then calculate the arithmetic mean over the F-scores. This *macro F-score* shows how well a classifier performs across all classes. In contrast, the *micro F-score* is an average where each single class F-score is weighted proportionally to the number of entity mentions of this class in the gold standard. This F-score is dominated by the classification performance of highly frequent classes and therefore implies that frequent classes are considered to be of higher importance. Since we here assume the minority class to be equally important, we optimize for the class-centric macro F-score.

6. RESULTS

Our experiments compare the protocols introduced in Section 3: Minority class-focused AL (AL-MINOR), balanced-batch AL (AL-BAB), AL with boosted disagreement (AL-BOOD, using the default boosting factor from Section 3.3), and over-sampling during AL (AL-OVER). Random selection (RAND) and the unmodified AL approach (AL-def) presented in Section 2 serve as baselines.

6.1 Re-balancing during Data Acquisition

To determine whether, and if so, to what extent these protocols are suitable to guide the AL process towards the minority class we analyzed the effect of the protocols with respect to the entity class ratio, the number of minority class entities selected, and performance in terms of different F-scores. While our primary goal is to increase the macro F-score, we also show the F-scores for both classes separately as a more detailed picture of the effects of the alternative protocols.

Figures 1 and 2 (left plots) show the entity class ratio yielded by each protocol at different token positions. The ratio for random sampling roughly corresponds to the data sets' overall entity class ratio of approximately 15.5. As already shown before in a different context [7], AL-def in our framework also shifts the ratio in favor of the minority class to values of about 11 on both data sets. While AL-BOOD achieves a very low ratio in early AL iterations, the ratio increases in later iteration rounds. Only AL-MINOR and AL-OVER keep a low ratio over many AL rounds.

Figures 1 and 2 also depict the absolute numbers of entity mentions of the minority and majority class. Using AL-MINOR or AL-BOOD on the MAL corpus, after 30,000 tokens most entity mentions of the minority class have been found and annotated. As only few sentences containing mi-

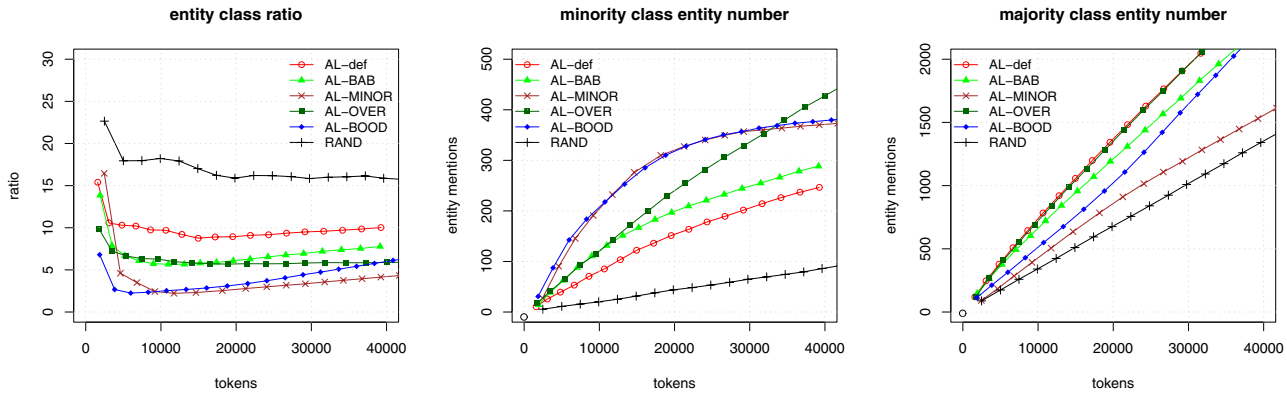


Figure 1: Entity mention statistics on MAL data set

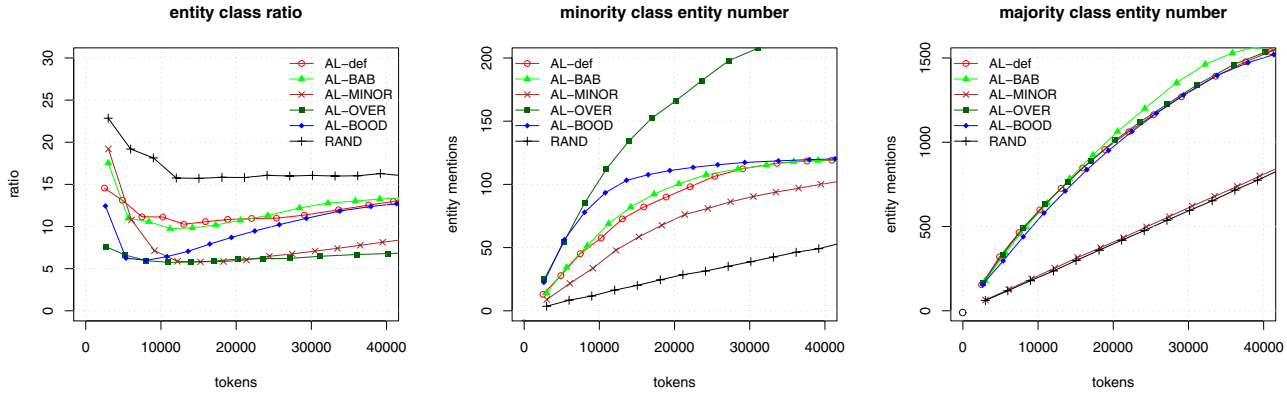


Figure 2: Entity mention statistics on TF data set

nority class entity mentions remain in the unlabeled rest of the corpus only a minor performance increase on this class can be expected from that point onwards. At the time when AL-BOOD cannot find many more sentences with minority class entities, the number of majority class entity mentions selected increases. The same pattern, although much more pronounced due to the smaller size of the corpus, holds for the TF data set where for AL-BOOD most entity mentions from the minority class have been found after 15,000 tokens.[3] On MAL, AL-OVER exceeds the number of minority class entities selected by AL-BOOD after 30,000 tokens as then for AL-BOOD the pool is exhausted with respect to the minority class entity mentions. On TF, this happens in very early AL iterations. While AL-OVER significantly increases the amount of minority class entity mentions on both data sets, it does not affect the number of majority class entity mentions. AL-MINOR results in high numbers on minority class entities on MAL, comparable with that of AL-BOOD; on TF, however, this protocol performs even worse than AL-def. Overall, we see that the different protocols indeed have an effect on the number of minority class entity mentions and, by this, on the entity class ratio.

[3]After splitting TF into the AL pool and the gold set, only about 100-110 entity mentions of the minority class remain in the pool.

Figures 3 and 4 show learning curves for the minority class, the majority class, and the macro F-score. On the MAL data set, AL-OVER does not perform significantly different from AL-def in terms of minority class or macro F-score but slightly deteriorates the majority class. AL-BAB increases the minority and macro F-score a bit with minor losses on the majority class F-score. While both AL-BOOD and AL-MINOR result in a steep increase on the minority class F-score, AL-BOOD performs better than AL-MINOR on the macro F-score. This is because AL-MINOR harms the majority class F-score more than AL-BOOD does – the majority class F-score for AL-MINOR is almost as low as RAND.

On the TF corpus, until about 15,000 tokens AL-BOOD outperforms AL-OVER in terms of minority class (and slightly also macro) F-score. After that point AL-OVER takes over. This can be explained by the phenomenon we have already observed in Figures 1 and 2: Almost all of the minority class entity mentions of the AL pool have been found and selected by AL-BOOD. Obviously, AL-OVER can and does outperform the other protocols as it is less restricted by the small overall amount of minority class entity mentions. AL-BAB does not have any significant effect on the TF data set. AL-MINOR performs very poorly here – even on the minority

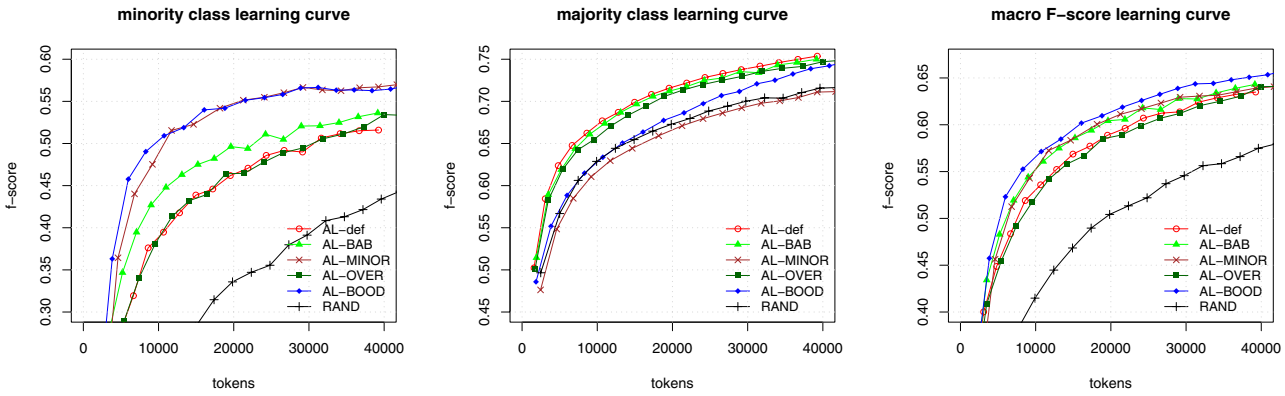


Figure 3: Learning curves on MAL data set

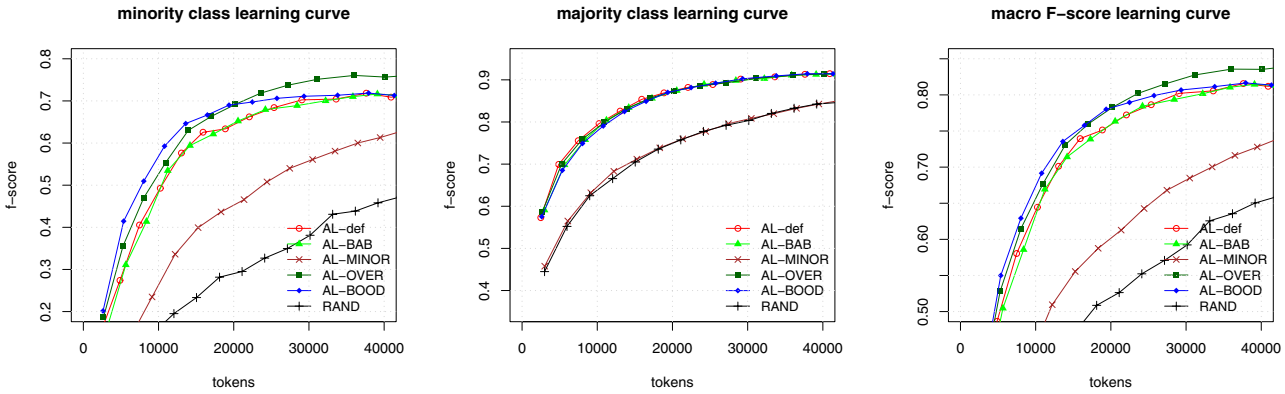


Figure 4: Learning curves on TF data set

class F-score worse than AL-def. We observed that a randomly compiled seed set is quite disadvantageous here as it hardly contains any minority class entity mentions. Thus, in early AL iterations, the binary classifiers employed in the committee for AL-MINOR mostly predict the OUTSIDE class and do thus hardly disagree so that AL-MINOR rather resembles a random selection mechanism in early AL iterations until – by chance – some more sentences with minority class entity mentions are selected. Only AL-MINOR had a significant (negative) effect on the majority class, all other protocols did not affect the majority class F-score.

In all our experiments, we have applied AL-BOOD with the default boosting factor for the minority class determined by the heuristic described in Section 3.3 (i.e., $b_{min} = 10.15$ for MAL, and $b_{min} = 15.25$ for TF). Further investigation in the effect of different factor values showed that the heuristically found value was among the best ones on the TF corpus. On the MAL corpus, however, lower values around $b_{min} = 6$ would have resulted in a better performance (cf. Section 6.3 for a brief discussion on ways to improve b).

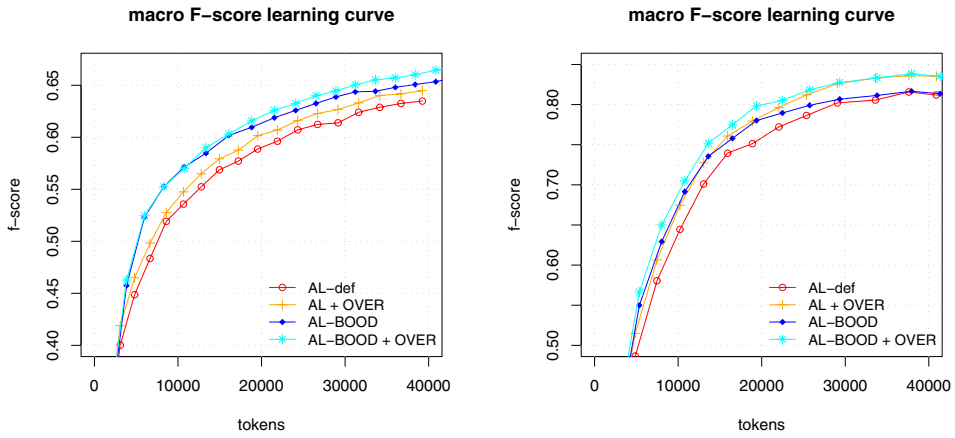
6.2 Re-balancing after Data Acquisition

Instead of addressing class imbalance *during* data acquisition, we could also have applied default AL to select the data to be annotated and once the data is available apply re-sampling techniques to address class imbalance. To study whether in our scenario re-sampling within the AL loop is more appropriate than re-sampling applied in a post-processing step, we performed further experiments.

We ran the selection by AL-def and then over-sampled during evaluation time in the same manner as we did in AL-OVER, i.e., at each evaluation position all sentences containing the minority class were duplicated. A direct comparison of over-sampling during AL selection (AL-OVER) and delayed over-sampling after AL selection (AL + OVER) reveals (see Table 2) that on both data sets AL-OVER performs worse than AL + OVER. On the MAL corpus, AL-OVER is clearly inferior to AL + OVER (with the exception of the 10,000 tokens measurement point), on the TF corpus both protocols yield roughly the same performance. We con-

corpus	scenario	tokens 10,000	20,000	30,000
MAL	AL-OVER	0.529	0.587	0.615
	AL + OVER	0.523	0.602	0.630
TF	AL-OVER	0.660	0.783	0.825
	AL + OVER	0.677	0.788	0.826

Table 2: Simultaneous and delayed over-sampling (compared on macro F-score)



(a) MAL (b) TF

Figure 5: Combination of AL-BOOD and delayed oversampling (OVER)

clude, that AL-OVER does not enforce a special "guidance" effect to the AL selection process and thus – if over-sampling is applied at all – one should better do so once the labeled data is available and not during annotation time.

In addition, our experiments showed that AL-BOOD does outperform AL-OVER on the MAL corpus and up to about 15,000 tokens also slightly on the TF corpus. However, to profit from both AL-BOOD's ability to select sentences containing many minority class entities and from the fact that over-sampling can help out when the overall number of minority class entities in a corpus is extremely limited or even exhausted, we combined both protocols (AL-BOOD + OVER). Figure 5 depicts macro F-score learning curves on MAL and TF comparing AL-def, AL-BOOD, AL + OVER, and AL-BOOD + OVER. On both corpora, AL-BOOD + OVER outperforms both "pure" AL + OVER and "pure" AL-BOOD by combining the respective protocols' strengths. The beneficial effects of the combination on MAL come into play only in later AL iterations (20,000 to 40,000 tokens), whereas on the TF corpus, the combination especially improves the performance on early to medium AL iterations (up to 20,000 tokens), while in later AL rounds the performance of AL + OVER is not exceeded.

With AL-BOOD + OVER, the macro F-score improved on both corpora compared to AL-def. On MAL, the macro F-score after 40,000 tokens was increased from 63.74 (AL-def) to 66.3 (AL-BOOD + OVER) and in order to reach AL-def's macro F-score of 63.74, we need only approximately 25,000 tokens using AL-BOOD + OVER, which comes to a saving of over 40%. Similarly, on TF where we get a macro F-score of 83.7 instead of 81.4 and save also about 40%.

6.3 Discussion

While improvements on the minority class clearly come at the cost of diminished performance on the majority class, our experiments showed overall gains in terms of improved macro F-score using AL-BOOD in a two-entity-class setting. AL-BOOD works only reasonably when suffiently many examples for the minority class are available. Since this was not the case for the TF data set, AL-BOOD could only improve upon the minority class F-score in early iterations. In real-world annotation settings, however, large amounts of unlabeled examples are typically available so that it is quite unlikely that the AL pool could be exhausted with respect to one entity class. While both AL-MINOR and AL-OVER only performed well on one data set, AL-BOOD was always amongst the best-performing protocols.

Our experiments indicate that our heuristic to determine the boosting factor for the minority class was a good start. However, more sophisticated ways to determine such a factor are necessary to yield optimal results. A good boosting factor depends on several influencing factors, including corpus-specific characteristics (average sentence length, number of entities per sentence, number of tokens per entity, difficulty to learn each class, whether sentences, on the average, contain exclusively one entity class, etc.) as well as application-specific considerations (misclassification costs and acceptable trade-off between gains on minority class and losses on majority class). In real-world annotation projects, the value of the boosting factor might also change over time. When a severe class imbalance is ascertained after several AL rounds, one might adjust the factor more in favor of the minority class, or vice versa.

7. CONCLUSIONS

In our research, two streams of work on increasing annotation economy converge. On the one hand, we deal with minimizing the efforts it takes to supply reasonable amounts of annotation meta data for (semi-)supervised learning from natural language corpora without compromising on the quality of this annotation data. We have shown before [19] that this requirement can be fulfilled using active learning as a sampling strategy that purposefully biases the learning data

to be annotated in favor of 'hard', i.e., particularly controversial decision cases, while skipping the 'easy' cases. On the other hand, in this paper we used active learning to further bias the selection process on already skewed data sets by balancing high-frequency and low-frequency classes up front *during* of the annotation process again avoiding unnecessary extra efforts. The need for this second bias comes from the assumption that low-frequency entity classes are as informative as or even more informative than high-frequency ones but suffer from extremely sparse training data and, correspondingly, low classification accuracy. We also claim that due to the Zipfian nature of natural language, class imbalance is a ubiquitous problem for NLP and by no means limited to named entity recognition in the biomedical application domain which we have deliberately chosen as our experimental framework. Our proposal, *viz.* altering the AL selection scheme by boosted disagreement with over-sampling once the data is available, turned out to be most effective among several alternatives to balance skewed data and improve the overall performance in terms of macro F-score.

Acknowledgements

This work was funded by the EC within the BOOTStrep (FP6-028099) and the CALBC (FP7-231727) project.

8. REFERENCES

[1] A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] M. Bloodgood and V. Shanker. Taking into account the differences between actively and passively acquired data: The case of active learning with support vector machines for imbalanced datasets. In *Proc. of the NAACL-HLT '09*, pages 137–140, 2009.

[3] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[4] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[5] C. Elkan. The foundations of cost-sensitive learning. In *Proc. of the IJCAI '01*, pages 973–978, 2001.

[6] S. Engelson and I. Dagan. Minimizing manual annotation cost in supervised training from corpora. In *Proc. of the ACL '96*, pages 319–326, 1996.

[7] S. Ertekin, J. Huang, L. Bottou, and L. Giles. Learning on the border: Active learning in imbalanced data classification. In *Proc. of the CIKM '07*, pages 127–136, 2007.

[8] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[9] U. Hahn, E. Beisswanger, E. Buyko, M. Poprat, K. Tomanek, and J. Wermter. Semantic annotations for biology: A corpus development initiative at the Jena University Language & Information Engineering Lab. In *Proc. of the LREC '08*, 2008.

[10] R. Hwa. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276, 2004.

[11] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[12] S. Kulick, A. Bies, M. Liberman, M. Mandel, R. T. McDonald, M. S. Palmer, and A. I. Schein. Integrated annotation for biomedical information extraction. In *Proc. of the HLT-NAACL '04 Workshop 'Linking Biological Literature, Ontologies and Databases: Tools for Users'*, pages 61–68, 2004.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the ICML '01*, pages 282–289, 2001.

[14] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proc. of the SIGIR '94*, pages 3–12, 1994.

[15] G. Ngai and D. Yarowsky. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proc. of the ACL '00*, pages 117–125, 2000.

[16] F. Provost. Machine learning from imbalanced data sets 101 (extended abstract). In *Proc. of the AAAI '00 Workshop on Learning from Imbalanced Data Sets*, 2000.

[17] E. Ringger, P. McClanahan, R. Haertel, G. Busby, M. Carmen, J. Carroll, K. Seppi, and D. Lonsdale. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proc. of the Linguistic Annotation Workshop at ACL '07*, pages 101–108, 2007.

[18] K. Tomanek, F. Laws, U. Hahn, and H. Schütze. On proper unit selection in Active Learning: Co-selection effects for named entity recognition. In *Proc. of the NAACL-HLT '09 Workshop 'Active Learning for NLP'*, pages 9–17, 2009.

[19] K. Tomanek, J. Wermter, and U. Hahn. An approach to text corpus construction which cuts annotation costs and maintains corpus reusability of annotated data. In *Proc. of the EMNLP-CoNLL '07*, pages 486–495, 2007.

[20] J. Zhu and E. Hovy. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proc. of the EMNLP-CoNLL '07*, pages 783–790, 2007.