A Comparison of Models for Cost-Sensitive Active Learning

Katrin Tomanek and **Udo Hahn**
Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena
http://www.julielab.de

Abstract

Active Learning (AL) is a selective sampling strategy which has been shown to be particularly cost-efficient by drastically reducing the amount of training data to be manually annotated. For the annotation of natural language data, cost efficiency is usually measured in terms of the number of tokens to be considered. This measure, assuming uniform costs for all tokens involved, is, from a linguistic perspective at least, intrinsically inadequate and should be replaced by a more adequate cost indicator, *viz.* the time it takes to manually label selected annotation examples. We here propose three different approaches to incorporate costs into the AL selection mechanism and evaluate them on the MUC7$_\mathcal{T}$ corpus, an extension of the MUC7 newspaper corpus that contains such annotation time information. Our experiments reveal that using a cost-sensitive version of semi-supervised AL, up to 54% of true annotation time can be saved compared to random selection.

1 Introduction

Active Learning (AL) is a selective sampling strategy for determining those annotation examples which are particularly informative for classifier training, while discarding those that are already easily predictable for the classifier given previous training experience. While the efficiency of AL has already been shown for many NLP tasks based on measuring the number of tokens or sentences that are saved in comparison to random sampling (e.g., Engelson and Dagan (1996), Tomanek et al. (2007) or Settles and Craven (2008)), it is obvious that just counting tokens under the assumption of *uniform* annotation costs for each token is empirically questionable, from a linguistic perspective, at least.

As an alternative, we here explore annotation costs that incur for AL based on an empirically more plausible cost metric, *viz.* the time it takes to annotate selected linguistic examples. We investigate three approaches to incorporate costs into the AL selection mechanism by modifying the standard (fully supervised) mode of AL and a non-standard semi-supervised one according to cost considerations. The empirical backbone of this comparison is constituted by MUC7$_\mathcal{T}$, a re-annotation of a part of the MUC7 newspaper corpus that contains annotation time information (Tomanek and Hahn, 2010).

2 Active Learning

Unlike random sampling, AL is a selective sampling technique where the learner is in control of the data to be chosen for training. By design, the intention behind AL is to reduce annotation costs, usually considered as the amount of labeled training material required to achieve a particular target performance of the model. The latter is yielded by querying labels only for those examples which are assumed to have a high training utility. In this section, we introduce different AL frameworks – the default, fully supervised AL approach (Section 2.1), as well as a semi-supervised variant of it (Section 2.2). In Section 2.3 we then propose three methods how these approaches to AL can be made cost-sensitive without further modifications.

Coling 2010: Poster Volume, pages 1247–1255,
Beijing, August 2010

2.1 Fully Supervised AL (FuSAL)

As we consider AL for the NLP task of Named Entity Recognition (NER), some design decisions have to be made. Firstly, the selection granularity is set to complete sentences – a reasonable linguistic annotation unit which still allows for fairly precise selection. Second, a batch of examples instead of a single example is selected per AL iteration to reduce the computational overhead of the sampling process.

We base our approach to AL on Conditional Random Fields (CRFs), which we employ as base learners (Lafferty et al., 2001). For observation sequences $\vec{x} = (x_1, \ldots, x_n)$ and label sequences $\vec{y} = (y_1, \ldots, y_n)$, a linear-chain CRF is defined as

$$P_\theta(\vec{y}|\vec{x}) = \frac{1}{Z_\theta(\vec{x})} \cdot \prod_{i=1}^{n} \exp \sum_{j=1}^{k} \lambda_j f_j\Big(y_{i-1}, y_i, \vec{x}, i\Big)$$

where $Z_\theta(\vec{x})$ is the normalization factor, and k feature functions $f_j(\cdot)$ with feature weights $\theta = (\lambda_1, \ldots, \lambda_k)$ appear.

The core of any AL approach is a utility function $u(p, \theta)$ which estimates the informativeness of each example p, a complete sentence $p = (\vec{x})$, drawn from the pool P of all unlabeled examples, for model induction. For our experiments, we employ two alternative utility functions which have produced the best results in previous experiments (Tomanek, 2010, Chapter 4). The first utility function is based on the confidence of a CRF model θ in the predicted label sequence $\vec{y}^*$ which is given by the probability distribution $P_\theta(\vec{y}^*|\vec{x})$. The utility function based on this probability boils down to

$$u_{\mathrm{LC}}(p, \theta) = 1 - P_\theta(\vec{y}^*|\vec{x})$$

so that sentences for which the predicted label sequence $\vec{y}^*$ has a low probability is granted a high utility. Instead of calculating the model's confidence on the complete sequence, we might alternatively calculate the model's confidence in its predictions on single tokens. To obtain an overall confidence for the complete sequence, the average over the single token-confidence values can be computed by the marginal probability $P_\theta(y_i|\vec{x})$. Now that we are calculating the confidence on the token level, we might also obtain the performance of the second best label and calculate the margin between the first and second best label as a confidence score so that the final utility function is obtained by

$$u_{\mathrm{MA}}(p, \theta) = -\frac{1}{n} \sum_{i=1}^{n} \Big(\max_{y' in \mathcal{Y}} P_\theta(y_i = y'|\vec{x}) - \max_{\substack{y'' in \mathcal{Y} \\ y' \neq y''}} P_\theta(y_i = y''|\vec{x}) \Big)$$

Algorithm 1 formalizes our AL framework. Depending on the utility function, the best b examples are selected per round, manually labeled, and then added to the set of labeled data $\mathcal{L}$ which feeds the classifier for the next training round.

Algorithm 1 NER-specific AL Framework

Given:
b: number of examples to be selected in each iteration
$\mathcal{L}$: set of labeled examples $l = (\vec{x}, \vec{y}) \in \mathcal{X}^n \times \mathcal{Y}^n$
$\mathcal{P}$: set of unlabeled examples $p = (\vec{x}) \in \mathcal{X}^n$
$T(\mathcal{L})$: a learning algorithm
$u(p, \theta)$: utility function

Algorithm:
loop until stopping criterion is met

1. learn model: $\theta \leftarrow T(\mathcal{L})$
2. sort $p \in \mathcal{P}$: let $S \leftarrow (p_1, \ldots, p_m) : u(p_i, \theta) \geq u(p_{i+1}, \theta), i \in [1, m], p \in \mathcal{P}$
3. select b examples p_i with highest utility from S: $\mathcal{B} \leftarrow \{p_1, \ldots, p_b\}, b \leq m, p_i \in \mathcal{S}$
4. query labels for all $p \in \mathcal{B}$: $\mathcal{B}' \leftarrow \{l_1, \ldots, l_b\}$
5. $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}', \mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{B}$

return $\mathcal{L}^* \leftarrow \mathcal{L}$ and $\theta^* \leftarrow T(\mathcal{L}^*)$

The specification is still not cost-sensitive as the selection of examples depends only on the utility function. Using u_{LC} will result in a reduction of the number of examples (i.e., sentences) selected irrespective of the sentence length so that a model learns the most from it. As a result, we observed that the selected sentences are quite long which might even cause higher annotation costs per sentence (Tomanek, 2010, Chapter 4). As for u_{MA} there is at least a slight normalization sensitive to costs since the sum over all token-level utility scores is normalized by the length of the selected sentence.

2.2 Semi-supervised AL (SeSAL)

Tomanek and Hahn (2009) extendeded this standard fully supervised AL framework by a semi-supervised variant (SeSAL). The selection of sentences is performed in a standard manner, i.e., similarly to the procedure in Algorithm 1. However, once selected, rather than manually annotating the complete sentence, only (uncertain) subsequences of each selected sentence are manually labeled, while the remaining (certain) ones are automatically annotated using the current version of the classifier.

After the selection of an informative example $p = (\vec{x})$ with $\vec{x} = (x_1, \ldots, x_n)$, the subsequences $\vec{x}' = (x_a, \ldots, x_b)$, $1 \leq a \leq b \leq n$, with low local uncertainty have to be identified. For reasons of simplicity, only sequences of length 1, i.e., single tokens, are considered. For a token x_i from a selected sequence $\vec{x}$ the model's confidence $C_\theta(y_i^*)$ in label y_i^* is estimated. Token-level confidence for a CRF is calculated as the marginal probability so that

$$C_\theta(y_i^*) = P_\theta(y_i = y_i^* | \vec{x})$$

where y_i^* specifies the label at the respective position of the predicted label sequence $\vec{y}^*$ (the one which is obtained by the Viterbi algorithm). If $C_\theta(y_i^*)$ exceeds a confidence threshold t, y_i^* is assigned as the putatively correct label. Otherwise, manual annotation of this token is required.

Employing SeSAL, savings of over 80 % of the tokens compared to random sampling are reported by Tomanek and Hahn (2009). Even when compared to FuSAL, still 60 % of the number of tokens are eliminated. A crucial question, however, not answered in these experiments, is whether this method actually reduces the overall annotation expenses in time rather than just in the number of tokens. Also SeSAL does not incorporate labeling costs in the selection process.

2.3 Cost-Sensitive AL (CoSAL)

In this section, we turn to an extension of FuSAL and SeSAL which incorporates cost sensitivity into the AL selection process (CoSAL). Three different approaches of CoSAL will be explored. The challenge we now face is that two contradictory criteria – utility and costs – have to be balanced.

2.3.1 Cost-Constrained Sampling

CoSAL can be realized in the most straightforward way by simply constraining the sampling to a particular maximum cost c_{max} per example. Therefore, in a pre-processing step all examples $p \in \mathcal{P}$ for which $\text{cost}(p) > c_{\text{max}}$ are removed from $\mathcal{P}$. The unmodified NER-specific AL framework can then be applied.

An obvious shortcoming of Cost-Constrained Sampling (CCS) is that it precludes any form of compensation between utility and costs. Thus, an exceptionally useful example with a cost factor slightly above c_{max} will be rejected. Another critical issue is how to fix c_{max}. If chosen too low, the pre-filtering of $\mathcal{P}$ results in a much too strong restriction of selection options when only few examples remain inside $\mathcal{P}$. If chosen too high, the cost constraint becomes ineffective.

2.3.2 Linear Rank Combination

A general solution to fit different criteria into a single one is by way of linear combination. If, however, different units of measurement are used, a transformation function for the alignment of benefit, or utility, and costs must be found. This can be difficult to determine. In our scenario, benefits measured by utility scores and costs measured in seconds are clearly incommensurable. As it is not immediately evident how to express utility in monetary terms (or vice versa), we transform utility and cost information into ranks $R\big(u(p,\theta)\big)$ and $R'\big(\text{cost}(p)\big)$ instead. As for utility, higher utility leads to higher ranks. As for costs, lower costs lead to higher ranks. The linear rank combination (LRK) is defined as

$$\phi_{\text{LRK}}(\vec{v}(p)) = \alpha R\big(u(p,\theta)\big) + (1-\alpha) R'\big(\text{cost}(p)\big)$$

where α is a weighting term. In a CoSAL scenario, where utility is the primary criterion, $\alpha > 0.5$ seems a reasonable choice. Alternatively, as costs and utility are contradictory, allowing equal influence for both criteria, as with $\alpha = 0.5$, it may be difficult to find appropriate examples in a medium-sized corpus. Thus, the choice of α depends on size and diversity with respect to combinations of utility and costs within $\mathcal{P}$.

2.3.3 Benefit-Cost Ratio

Our third approach to CoSAL is based on the Benefit-Cost Ratio (BCR). Given equal units of measurement for benefits and costs, the benefit-cost ratio indicates whether a scenario is profitable (ratio > 1). BCR can also be applied when units are incommensurable and a transformation function is available, as is the case for the combination of utility and cost. This holds as long as benefit and costs can be expressed in the same units by a linear transformation function, i.e., $u(p, \theta) = \beta \cdot \text{cost}(p) + b$. If such a transformation function exists, one can refrain from finding proper values for the above variables b and β and instead calculate BCR as

$$\phi_{\text{BCR}}(p) = \frac{u(p, \theta)}{\text{cost}(p)}$$

Since annotation costs are usually expressed on a linear scale, this is also required for utility, if we want to use BCR. But when utility is based on model confidence as we do it here, this property gets lost.[1] Hence a non-linear transformation function is needed to fit the scales of utility and costs. Assuming a linear relationship between utility and costs, BCR has already been applied by Haertel et al. (2008) and Settles et al. (2008). Our approach provides a crucial extension as we explicitly consider scenarios where such a linear relationship is not given and a non-linear transformation function is required instead.

In a direct comparison of LRK with BCR, LRK may be used when such a transformation function would be needed but is unknown and hard to find. Choosing LRK over BCR is also motivated by findings in the context of data fusion in information retrieval where Hsu and Taksa (2005) remark that, given incommensurable units and scales, one would do better when ranks rather than the actual scores or values were combined.

3 Experiments

In the following, we study possible benefits of CoSAL, relative to FuSAL and SeSAL, in the light of real annotation times as a cost measure (instead of the standard, yet inadequate one, *viz.* the number of tokens being selected). Such timing data is available in the MUC7$_\mathcal{T}$ corpus (Tomanek and Hahn, 2010), a re-annotation of the MUC7 corpus containing the ENAMEX types (persons, locations, and organizations) and a time stamp reflecting the time it took annotators to decide on each entity type. The MUC7$_\mathcal{T}$ corpus contains 3,113 sentences (76,900 tokens).

The results we report on are averaged over 20 independent runs. For each run, we split the MUC7$_\mathcal{T}$ corpus randomly into a pool to select from (90%) and an evaluation set (10%). AL was started from a random seed set of 20 sentences. As utility scores to estimate benefits we applied u_{MA} and u_{LC} as defined in Section 2.1.

The plots in the following sections depict costs in terms of annotation time (in seconds) relative to annotation quality (expressed via F1-scores). Learning curves are only shown for early AL iterations. Later on, in the convergence phase, due to the two conflicting criteria now considered simultaneously, selection options become more and more scarce so that CoSAL necessarily performs sub-optimally.

3.1 Parametrization of CoSAL Approaches

Preparatory experiments were run to analyze how different parameters affected different CoSAL settings. For the CCS and LRK experiments, we used the u_{LC} utility function.

For CCS, we tested three c_{max} values, *viz.* 7.5, 10, and 15, to determine the maximum performance attainable on MUC7$_\mathcal{T}$ when only examples below the chosen threshold were included. Our choices of the maximum were based on the distributions of annotation times over the sentences (see Figure 1) where 7.5s marks the 75% quantile and 15s is just above the 90% quantile. For 7.5s, we peaked at $F_{max} = 0.84$, for 10s at $F_{max} = 0.86$, and for 15s at $F_{max} = 0.88$. Figure 2 (top) shows the learning curves of CoSAL with CCS and different c_{max} values. With $c_{max} = 15$, as could be expected from the boxplot in Figure 1, no difference can be observed compared to cost-*in*sensitive FuSAL. CCS with lower values for c_{max} stagnates at the maximum perfor-

[1] Though normalized to $[0, 1]$, confidence estimates, especially for sequence classification, are often not on a linear scale so that confidence values that are twice as high do not necessarily mean that the benefit in training a model on such an example is doubled.

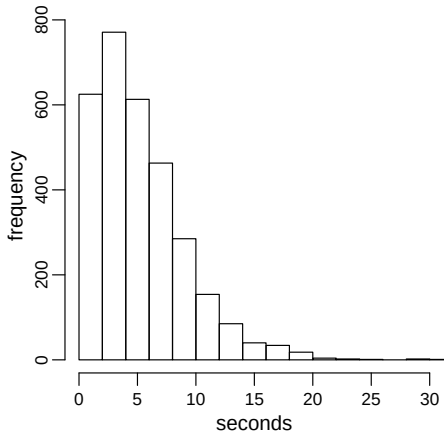


Figure 1: Distribution of annotation times per sentence in MUC7$_\mathcal{T}$.

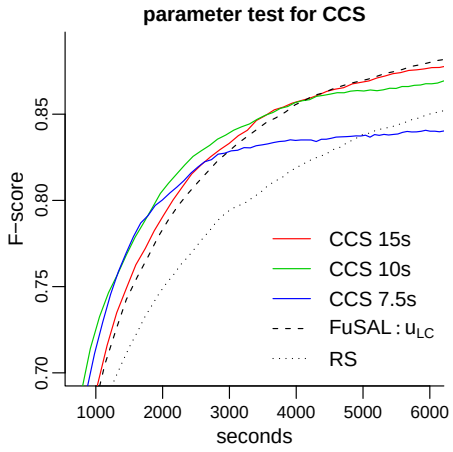
mance reported above, but still improves upon cost-*in*sensitive FuSAL in early AL iterations.

At some point in time all economical examples, with costs below c_{max} but high utility, have been consumed from the corpus. Even in a corpus much larger than MUC7$_\mathcal{T}$ this effect will only occur with some delay. Indeed, any choice of a restrictive value for c_{max} will cause similar exhaustion effects. Unfortunately, it is unclear how to tune c_{max} suitably in a real-life annotation scenario where pretests for maximum performance for a particular c_{max} are not possible. For further experiments, we chose $c_{max} = 10$.

For LRK, we tested three different weights α, *viz.* 0.5, 0.75, and 0.9. Figure 2 (bottom) shows their effects on the learning curves. Similar tendencies as for c_{max} for CCS can be observed. With $\alpha = 0.9$, CoSAL does not fall below default FuSAL, at least in the observed range. A lower weight of $\alpha = 0.75$ results in larger improvements in earlier AL iterations but then falls back to FuSAL and in later AL iterations (not shown here) even below FuSAL. If the time parameter is granted too much influence, as with $\alpha = 0.5$, performance even drops to random selection level. This might also be due to corpus exhaustion. For further experiments, we chose $\alpha = 0.75$ because of its potential to improve upon FuSAL in early iterations.

For BCR with u_{MA}, we change this utility function to $n \cdot u_{\mathrm{MA}}$ to compensate for the normalization by token length which is otherwise already contained in u_{MA}(n is the length of the respective sentence). For u_{LC}, the preparatory experiments already showed that this utility function does not behave on a linear scale. This is so because u_{LC} is based on $P_\theta(\vec{y}|\vec{x})$ for confidence estimation of the complete label sequence $\vec{y}$. Hence, a u_{LC} score twice as high does not indicate doubled benefit for classifier training. Thus, we need a non-linear calibration function to transform u_{LC} into a proper utility estimator on a linear scale so that BCR can be applied.

To determine such a non-linear calibration function, the *true* benefit of an example p would



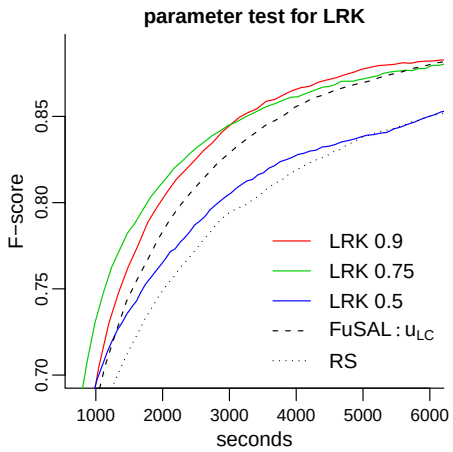


Figure 2: Different parameter settings for CCS and LRK based on FuSAL with u_{LC} as utility function. FuSAL: u_{LC} refers to cost-*in*sensitive FuSAL, CCS and LRK to the cost-sensitive versions of FuSAL with the respective parameters.

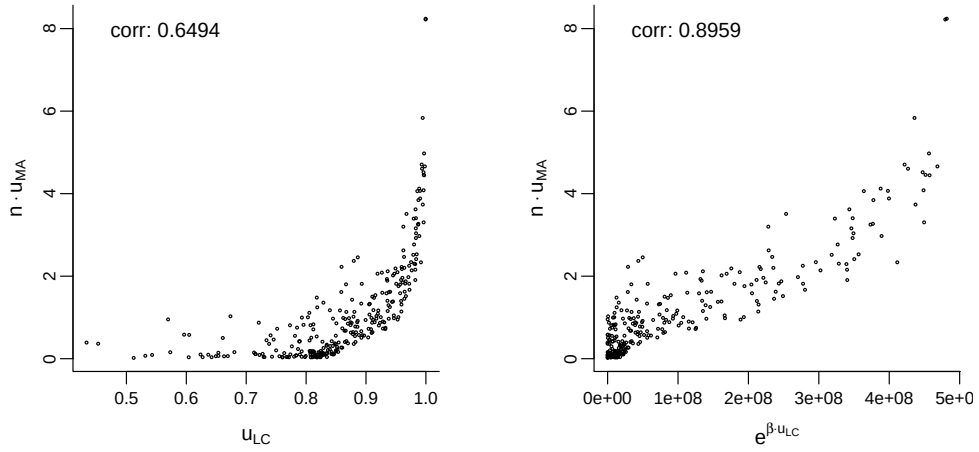


Figure 3: Scatter plots for (a) u_{LC} versus $n \cdot u_{\mathrm{MA}}$ and (b) $e^{\beta \cdot u_{\mathrm{LC}}}$ versus $n \cdot u_{\mathrm{MA}}$

be needed. In the absence of such information, we consider $n \cdot u_{\mathrm{MA}}$ as a good approximation. To identify the relationship between u_{LC} and $n \cdot u_{\mathrm{MA}}$, we trained a model on a random subsample from $P' \subset \mathcal{P}$ and used this model to obtain the scores for u_{LC} and $n \cdot u_{\mathrm{MA}}$ for each example from the test set $\mathcal{T}$.[2] Figure 3 (left) shows a scatter plot of these scores which provides ample evidence that the relationship between u_{LC} and benefit is indeed non-linear. As calibration function for u_{LC} we propose $f(p) = e^{\beta \cdot u_{\mathrm{LC}}(p)}$. Experimentally, we determined $\beta = 20$ as a good value. Figure 3 (right) reveals that $e^{\beta \cdot u_{\mathrm{LC}}(p)}$ is a better utility estimator; the correlation with $n \cdot u_{\mathrm{MA}}$ is now $corr = 0.8959$ and the relationship is close to being linear.

In Figure 4, learning curves for BCR with the utility function u_{LC} and the calibrated function $e^{\beta \cdot u_{\mathrm{LC}}(p)}$ are compared. BCR with the uncalibrated utility function u_{LC} fails miserably (the performance falls even below random selection). This adds credibility to our claim that while u_{LC} may be appropriate for *ranking* examples (as for standard, cost-*in*sensitive AL), it is inappropriate for *estimating* true benefit/utility which is needed when costs are to be incorporated with the BCR method. BCR with the calibrated utility $e^{\beta \cdot u_{\mathrm{LC}}(p)}$, in contrast, outperforms cost-*in*sensitive FuSAL. For further experiments with BCR, we either apply $n \cdot u_{\mathrm{MA}}$ or $e^{\beta \cdot u_{\mathrm{LC}}(p)}$ as utility functions.

[2] We experimented with different sizes for P', with almost identical results.

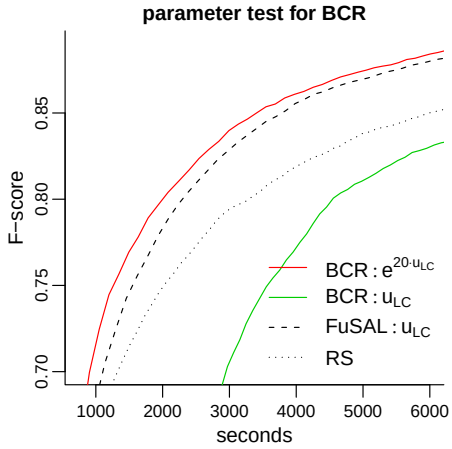


Figure 4: Different parameter settings for BCR

3.2 Comparison of CoSAL Approaches

We compared all three approaches to CoSAL in the parametrization chosen above for the utility functions u_{MA} and u_{LC}. Learning curves are shown in Figure 5. Improvements over cost-*in*sensitive AL are only achieved in early AL iterations up to 2,500s (for CoSAL based on u_{MA}) or 4,000s (for CoSAL based on u_{LC}) of annotation time. This exclusiveness of early improvements can be explained by the size of the corpus and, by this, the limited number of good selection options. Since AL selects with respect to two conflicting criteria, the pool $\mathcal{P}$ should be much larger to increase the chance for examples that are favorable with respect to both criteria.

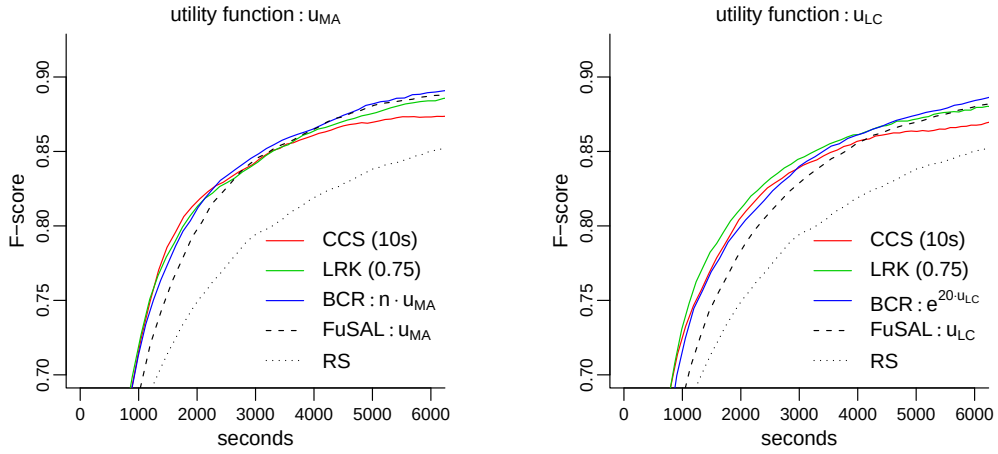


Figure 5: Comparison of CoSAL approaches for the utility functions u_{MA} and u_{LC}. Baseline given by random selection (RS) and standard FuSAL with either u_{MA} or u_{LC}.

Improvements for CoSAL based on u_{LC} are generally higher than for u_{MA}. Moreover, cost-*in*sensitive AL based on u_{LC} does not exhibit any normalization where, in contrast, u_{MA} is normalized at least to the number of tokens per example. In CoSAL, both u_{LC} and u_{MA} are normalized by costs, which is methodologically a more substantial enhancement for u_{LC} than for u_{MA}.

For CoSAL based on u_{MA} we cannot proclaim a clear winner among the different approaches. All three CoSAL approaches improve upon cost-*in*sensitive AL. For CoSAL based on u_{LC}, LRK performs best, while CCS and BCR perform similarly well. Given this result, we might prefer LRK or CCS over BCR. A disadvantage of the first two approaches is that they require corpus-specific parameters which may be difficult to find for a new learning problem for which no data for experimentation is at hand. Though not the best performer, BCR does not require further parametrization and appears more appropriate for real-life annotation projects – as long as utility is an appropriate estimator for benefit. CoSAL with BCR has already been studied by Settles et al. (2008). They also applied a utility function based on sequence-confidence estimation which presumably, as with our u_{LC} utility function, is not a good benefit estimator. The fact that Settles et al. did not explicitly treat this issue might explain why cost-sensitive AL based on BCR often performed worse than cost-*in*sensitive AL in their experiments.

3.3 CoSAL Applied to SeSAL

We looked at a cost-sensitive version of SeSAL by applying the cost-sensitive FuSAL approach together with BCR and the transformation function for the utility as discussed above. On top of this selection, we ran the standard SeSAL approach – only tokens below a confidence threshold were selected for annotation. The following experiments are all based on the u_{LC} utility function (and the transformation function of it).

Figure 6 depicts learning curves for cost-*in*sensitive and cost-sensitive SeSAL and FuSAL which reveal that cost-sensitive SeSAL consid-

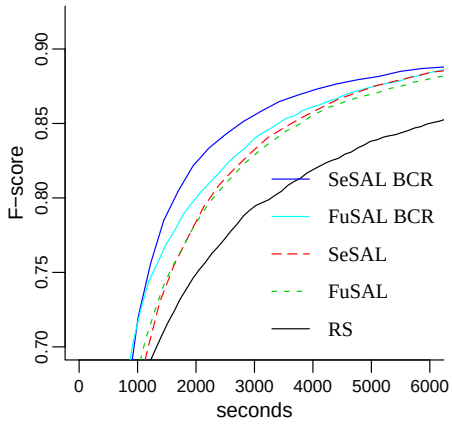


Figure 6: Cost-sensitive (BCR variants) *vs.* cost-*in*sensitive FuSAL and SeSAL with u_{LC} as utility function.

erably outperforms cost-sensitive FuSAL. Cost-sensitive SeSAL attains a target performance of F=0.85 with only 2806s, while cost-sensitive FuSAL needs 3410s, and random selection consumes over 6060s. Thus, cost-sensitive SeSAL here reduces true annotation time by about 54 % compared to random selection, whereas cost-sensitive FuSAL reduces annotation time by only 44 %.

4 Related Work

Although the reduction of data acquisition costs that result from human labeling efforts have always been the main driver for AL studies, *cost-sensitive AL* is a new branch of AL. In an early study on cost metrics for AL, Becker and Osborne (2005) examined whether AL, while decreasing the sample size on the one hand, on the other hand increased annotation efforts. For a real-world AL annotation project, they demonstrated that the actual sampling efficiency measure for an AL approach depends on the cost metric being applied. In a companion paper, Hachey et al. (2005) studied how sentences selected by AL affected the annotators' performance both in terms of the time needed and the annotation accuracy achieved. They found that selectively sampled examples are, on the average, more difficult to annotate than randomly sampled ones. This observation, for the first time, questioned the widespread assumption that all annotation examples can be assigned a uniform cost factor.

Making a standard AL approach cost-sensitive by normalizing utility in terms of annotation time has been proposed before by Haertel et al. (2008), Settles et al. (2008), and Donmez and Carbonell (2008). CoSAL based on the net-benefit (costs subtracted from utility) was proposed by Vijayanarasimhan and Grauman (2009) for object recognition in images and Kapoor et al. (2007) for voice message classification.

5 Conclusions

We investigated three approaches to incorporate the notion of cost into the AL selection mechanism, including a fixed maximal cost budget per example, a linear rank combination to express net-benefit, and a benefit-cost ratio. The cost metric we applied was the *time* needed by human coders for annotating particular annotation examples.

Among the three approaches to cost-sensitive AL, we see a slight advantage for benefit cost ratios in real-world settings because they do not require additional corpus-specific parametrization, once a proper calibration function is found.

Another observation is that advantages of the three cost-sensitive AL models over cost-*in*sensitive ones consistently occur only in early iteration rounds – a result we attribute to corpus exhaustion effects since cost-sensitive AL selects for two criteria (utility and cost) and thus requires a extremely large pool to be able to pick up really advantageous examples. Consequently, applied to real-world annotation settings where the pools may be extremely large, we expect cost-sensitive approaches to be even more effective in terms of the reduction of annotation time.

To be applicable in real-world scenarios, annotation costs which, in our experiments, were directly traceable in the MUC7$_\mathcal{T}$ corpus have to be estimated since they are not known prior to annotation. In Tomanek et al. (2010), we investigated the reading behavior during named entity annotation using eye-tracking technology. With the insights gained from this study on crucial factors influencing annotation time we were able to induce such a much needed *predictive* model of annotation costs. In future work, we plan to incorporate this empirically founded cost model into our approaches to cost-sensitive AL and to investigate whether our positive findings can be reproduced with estimated costs as well.

Acknowledgements

This work was partially funded by the EC within the CALBC (FP7-231727) project.

References

Becker, Markus and Miles Osborne. 2005. A two-stage method for active learning of statistical grammars. In *IJCAI'05 – Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 991–996. Edinburgh, Scotland, UK, July 31 - August 5, 2005.

Donmez, Pinar and Jaime Carbonell. 2008. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *CIKM'08 – Proceeding of the 17th ACM conference on Information and Knowledge Management*, pages 619–628. Napa Valley, CA, USA, October 26-30, 2008.

Engelson, Sean 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. Santa Cruz, CA, USA, June 24-27, 1996.

Hachey, Ben, 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. Ann Arbor, MI, USA, June 29-30, 2005.

Haertel, Robbie, Kevin Seppi, Eric Ringger, and James Carroll. 2008. Return on investment for active learning. In *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*. Whistler, BC, Canada, December 13, 2008.

Hsu, Frank and Isak Taksa. 2005. Comparing rank and score combination methods for data fusion in information retrieval. *Information Retrieval*, 8(3):449–480.

Kapoor, Ashish, Eric Horvitz, and Sumit Basu. 2007. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI'07 – Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 877–882. Hyderabad, India, January 6-12, 2007.

Lafferty, John, 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. Williamstown, MA, USA, June 28 - July 1, 2001.

Settles, Burr 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 1069–1078. Waikiki, Honolulu, Hawaii, USA, October 25-27, 2008.

Settles, Burr, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*. Whistler, BC, Canada, December 13, 2008.

Tomanek, Katrin and Udo Hahn. 2009. Semi-supervised active learning for sequence labeling. In *ACL/IJCNLP'09 – Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 1039–1047. Singapore, August 2-7, 2009.

Tomanek, Katrin and Udo Hahn. 2010. Annotation time stamps: Temporal metadata from the linguistic annotation process. In *LREC'10 – Proceedings of the 7th International Conference on Language Resources and Evaluation*. La Valletta, Malta, May 17-23, 2010.

Tomanek, Katrin, Joachim Wermter, and Udo Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains corpus reusability of annotated data. In *EMNLP-CoNLL'07 – Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 486–495. Prague, Czech Republic, June 28-30, 2007.

Tomanek, Katrin, Udo Hahn, Steffen Lohmann, and Jürgen Ziegler. 2010. A cognitive cost model of annotations based on eye-tracking data. In *ACL'10 – Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, July 11-16, 2010.

Tomanek, Katrin. 2010. *Resource-Aware Annotation through Active Learning*. Ph.D. thesis, Technical University of Dortmund.

Vijayanarasimhan, Sudheendra and Kristen Grauman. 2009. What's it going to cost you? predicting effort vs. informativeness for multi-label image annotations. *CVPR'09 – Proceedings of the 2009 IEEE Computer Vision and Pattern Recognition Conference*.