Semi-Supervised Active Learning for Sequence Labeling

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

Abstract

While Active Learning (AL) has already been shown to markedly reduce the annotation efforts for many sequence labeling tasks compared to random selection, AL remains unconcerned about the internal structure of the selected sequences (typically, sentences). We propose a semi-supervised AL approach for sequence labeling where only highly uncertain subsequences are presented to human annotators, while all others in the selected sequences are automatically labeled. For the task of entity recognition, our experiments reveal that this approach reduces annotation efforts in terms of manually labeled tokens by up to 60 % compared to the standard, fully supervised AL scheme.

1 Introduction

Supervised machine learning (ML) approaches are currently the methodological backbone for lots of NLP activities. Despite their success they create a costly follow-up problem, *viz.* the need for human annotators to supply large amounts of "golden" annotation data on which ML systems can be trained. In most annotation campaigns, the language material chosen for manual annotation is selected randomly from some reference corpus.

Active Learning (AL) has recently shaped as a much more efficient alternative for the creation of precious training material. In the AL paradigm, only examples of high training utility are selected for manual annotation in an iterative manner. Different approaches to AL have been successfully applied to a wide range of NLP tasks (Engelson and Dagan, 1996; Ngai and Yarowsky, 2000; Tomanek et al., 2007; Settles and Craven, 2008).

When used for sequence labeling tasks such as POS tagging, chunking, or named entity recognition (NER), the examples selected by AL are sequences of text, typically sentences. Approaches to AL for sequence labeling are usually unconcerned about the internal structure of the selected sequences. Although a high overall training utility might be attributed to a sequence as a whole, the subsequences it is composed of tend to exhibit different degrees of training utility. In the NER scenario, e.g., large portions of the text do not contain any target entity mention at all. To further exploit this observation for annotation purposes, we here propose an approach to AL where human annotators are required to label only uncertain *sub*sequences within the selected sentences, while the remaining subsequences are labeled automatically based on the model available from the previous AL iteration round. The hardness of subsequences is characterized by the classifier's confidence in the predicted labels. Accordingly, our approach is a combination of AL and self-training to which we will refer as *semi-supervised Active Learning* (SeSAL) for sequence labeling.

While self-training and other bootstrapping approaches often fail to produce good results on NLP tasks due to an inherent tendency of deteriorated data quality, SeSAL circumvents this problem and still yields large savings in terms annotation decisions, i.e., tokens to be manually labeled, compared to a standard, fully supervised AL approach.

After a brief overview of the formal underpinnings of Conditional Random Fields, our base classifier for sequence labeling tasks (Section 2), a fully supervised approach to AL for sequence labeling is introduced and complemented by our semi-supervised approach in Section 3. In Section 4, we discuss SeSAL in relation to bootstrapping and existing AL techniques. Our experiments are laid out in Section 5 where we compare fully and semi-supervised AL for NER on two corpora, the newspaper selection of MUC7 and PENNBIOIE, a biological abstracts corpus.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 1039–1047,
Suntec, Singapore, 2-7 August 2009.

2 Conditional Random Fields for Sequence Labeling

Many NLP tasks, such as POS tagging, chunking, or NER, are sequence labeling problems where a sequence of class labels $\vec{y} = (y_1,\ldots,y_n) \in \mathcal{Y}^n$ are assigned to a sequence of input units $\vec{x} = (x_1,\ldots,x_n) \in \mathcal{X}^n$. Input units x_j are usually tokens, class labels y_j can be POS tags or entity classes.

Conditional Random Fields (CRFs) (Lafferty et al., 2001) are a probabilistic framework for labeling structured data and model $P_{\vec{\lambda}}(\vec{y}|\vec{x})$. We focus on first-order linear-chain CRFs, a special form of CRFs for sequential data, where

$$P_{\vec{\lambda}}(\vec{y}|\vec{x}) = \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \exp\Big(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1},y_j,\vec{x},j)\Big) \quad (1)$$

with normalization factor $Z_{\vec{\lambda}}(\vec{x})$, feature functions $f_i(\cdot)$, and feature weights λ_i.

Parameter Estimation. The model parameters λ_i are set to maximize the penalized log-likelihood $\mathcal{L}$ on some training data $\mathcal{T}$:

$$\mathcal{L}(\mathcal{T}) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \log p(\vec{y}|\vec{x}) - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2} \quad (2)$$

The partial derivations of $\mathcal{L}(\mathcal{T})$ are

$$\frac{\partial\mathcal{L}(\mathcal{T})}{\partial\lambda_i} = \tilde{E}(f_i) - E(f_i) - \frac{\lambda_i}{\sigma^2} \quad (3)$$

where $\tilde{E}(f_i)$ is the empirical expectation of feature f_i and can be calculated by counting the occurrences of f_i in $\mathcal{T}$. $E(f_i)$ is the model expectation of f_i and can be written as

$$E(f_i) = \sum_{(\vec{x},\vec{y})\in\mathcal{T}} \sum_{\vec{y}\,'\in\mathcal{Y}^n} P_{\vec{\lambda}}(\vec{y}\,'|\vec{x}) \cdot \sum_{j=1}^{n} f_i(y'_{j-1}, y'_j, \vec{x}, j) \quad (4)$$

Direct computation of $E(f_i)$ is intractable due to the sum over all possible label sequences $\vec{y}\,' \in \mathcal{Y}^n$. The Forward-Backward algorithm (Rabiner, 1989) solves this problem efficiently. Forward (α) and backward (β) scores are defined by

$$\alpha_j(y|\vec{x}) = \sum_{y'\in T_j^{-1}(y)} \alpha_{j-1}(y'|\vec{x}) \cdot \Psi_j(\vec{x},y',y)$$

$$\beta_j(y|\vec{x}) = \sum_{y'\in T_j(y)} \beta_{j+1}(y'|\vec{x}) \cdot \Psi_j(\vec{x},y,y')$$

where $\Psi_j(\vec{x},a,b) = \exp\Big(\sum_{i=1}^{m} \lambda_i f_i(a,b,\vec{x},j)\Big)$, $T_j(y)$ is the set of all successors of a state y at a specified position j, and, accordingly, $T_j^{-1}(y)$ is the set of predecessors.

Normalized forward and backward scores are inserted into Equation (4) to replace $\sum_{\vec{y}\,'\in\mathcal{Y}^n} P_{\vec{\lambda}}(\vec{y}\,'|\vec{x})$ so that $\mathcal{L}(\mathcal{T})$ can be optimized with gradient-based or iterative-scaling methods.

Inference and Probabilities. The marginal probability

$$P_{\vec{\lambda}}(y_j = y'|\vec{x}) = \frac{\alpha_j(y'|\vec{x}) \cdot \beta_j(y'|\vec{x})}{Z_{\vec{\lambda}}(\vec{x})} \quad (5)$$

specifies the model's confidence in label y' at position j of an input sequence $\vec{x}$. The forward and backward scores are obtained by applying the Forward-Backward algorithm on $\vec{x}$. The normalization factor is efficiently calculated by summing over all forward scores:

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{y\in\mathcal{Y}} \alpha_n(y|\vec{x}) \quad (6)$$

The most likely label sequence

$$\vec{y}^{\,*} = \operatorname*{argmax}_{\vec{y}\in\mathcal{Y}^n} \exp\Big(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1},y_j,\vec{x},j)\Big) \quad (7)$$

is computed using the Viterbi algorithm (Rabiner, 1989). See Equation (1) for the conditional probability $P_{\vec{\lambda}}(\vec{y}^{\,*}|\vec{x})$ with $Z_{\vec{\lambda}}$ calculated as in Equation (6). The marginal and conditional probabilities are used by our AL approaches as confidence estimators.

3 Active Learning for Sequence Labeling

AL is a selective sampling technique where the learning protocol is in control of the data to be used for training. The intention with AL is to reduce the amount of labeled training material by querying labels only for examples which are assumed to have a high training utility. This section, first, describes a common approach to AL for sequential data, and then presents our approach to semi-supervised AL.

3.1 Fully Supervised Active Learning

Algorithm 1 describes the general AL framework. A utility function $U_{\mathcal{M}}(p_i)$ is the core of each AL approach – it estimates how useful it would be for

Algorithm 1 General AL framework

Given:
B: number of examples to be selected
L: set of labeled examples
P: set of unlabeled examples
$U_{\mathcal{M}}$: utility function

Algorithm:
loop until stopping criterion is met
1. learn model $\mathcal{M}$ from L
2. for all $p_i \in P : u_{p_i} \leftarrow U_{\mathcal{M}}(p_i)$
3. select B examples $p_i \in P$ with highest utility u_{p_i}
4. query human annotator for labels of all B examples
5. move newly labeled examples from P to L

return L

a specific base learner to have an unlabeled example labeled and, subsequently included in the training set.

In the sequence labeling scenario, such an example is a stream of linguistic items – a sentence is usually considered as proper sequence unit. We apply CRFs as our base learner throughout this paper and employ a utility function which is based on the conditional probability of the most likely label sequence $\vec{y}^*$ for an observation sequence $\vec{x}$ (cf. Equations (1) and (7)):

$$U_{\vec{\lambda}}(\vec{x}) = 1 - P_{\vec{\lambda}}(\vec{y}^*|\vec{x}) \tag{8}$$

Sequences for which the current model is least confident on the most likely label sequence are preferably selected.[1] These selected sentences are *fully* manually labeled. We refer to this AL mode as *fully supervised Active Learning (FuSAL)*.

3.2 Semi-Supervised Active Learning

In the sequence labeling scenario, an example which, as a whole, has a high utility $U_{\vec{\lambda}}(\vec{x})$, can still exhibit subsequences which do not add much to the overall utility and thus are fairly easy for the current model to label correctly. One might therefore doubt whether it is reasonable to manually label the entire sequence. Within many sequences of natural language data, there are probably large subsequences on which the current model already does quite well and thus could automatically generate annotations with high quality. This might, in particular, apply to NER where larger stretches of sentences do not contain any entity mention at all, or merely trivial instances of an entity class easily predictable by the current model.

[1]There are many more sophisticated utility functions for sequence labeling. We have chosen this straightforward one for simplicity and because it has proven to be very effective (Settles and Craven, 2008).

For the sequence labeling scenario, we accordingly modify the fully supervised AL approach from Section 3.1. Only those tokens remain to be manually labeled on which the current model is highly uncertain regarding their class labels, while all other tokens (those on which the model is sufficiently certain how to label them correctly) are automatically tagged.

To select the sequence examples the same utility function as for FuSAL (cf. Equation (8)) is applied. To identify tokens x_j from the selected sequences which still have to be manually labeled, the model's confidence in label y_j^* is estimated by the marginal probability (cf. Equation (5))

$$C_{\vec{\lambda}}(y_j^*) = P_{\vec{\lambda}}(y_j = y_j^*|\vec{x}) \tag{9}$$

where y_j^* specifies the label at the respective position of the most likely label sequence $\vec{y}^*$ (cf. Equation (7)). If $C_{\vec{\lambda}}(y_j^*)$ exceeds a certain *confidence threshold t*, y_j^* is assumed to be the correct label for this token and assigned to it.[2] Otherwise, manual annotation of this token is required. So, compared to FuSAL as described in Algorithm 1 only the third step step is modified.

We call this *semi-supervised Active Learning (SeSAL)* for sequence labeling. SeSAL joins the standard, fully supervised AL schema with a bootstrapping mode, namely self-training, to combine the strengths of both approaches. Examples with high training utility are selected using AL, while self-tagging of certain "safe" regions within such examples additionally reduces annotation effort. Through this combination, SeSAL largely evades the problem of deteriorated data quality, a limiting factor of "pure" bootstrapping approaches.

This approach requires two parameters to be set: Firstly, the *confidence threshold* t which directly influences the portion of tokens to be manually labeled. Using lower thresholds, the self-tagging component of SeSAL has higher impact – presumably leading to larger amounts of tagging errors. Secondly, a *delay factor* d can be specified which channels the amount of manually labeled tokens obtained with FuSAL before SeSAL is to start. Only with $d = 0$, SeSAL will already affect the first AL iteration. Otherwise, several iterations of FuSAL are run until a switch to SeSAL will happen.

[2]*Sequences* of consecutive tokens x_j for which $C_{\vec{\lambda}}(y_j^*) \leq t$ are presented to the human annotator instead of *single*, isolated tokens.

It is well known that the performance of bootstrapping approaches crucially depends on the size of the seed set – the amount of labeled examples available to train the initial model. If class boundaries are poorly defined by choosing the seed set too small, a bootstrapping system cannot learn anything reasonable due to high error rates. If, on the other hand, class boundaries are already too well defined due to an overly large seed set, nothing to be learned is left. Thus, together with low thresholds, a delay rate of $d > 0$ might be crucial to obtain models of high performance.

4 Related Work

Common approaches to AL are variants of the Query-By-Committee approach (Seung et al., 1992) or based on uncertainty sampling (Lewis and Catlett, 1994). Query-by-Committee uses a committee of classifiers, and examples on which the classifiers disagree most regarding their predictions are considered highly informative and thus selected for annotation. Uncertainty sampling selects examples on which a single classifier is least confident. AL has been successfully applied to many NLP tasks; Settles and Craven (2008) compare the effectiveness of several AL approaches for sequence labeling tasks of NLP.

Self-training (Yarowsky, 1995) is a form of semi-supervised learning. From a seed set of labeled examples a weak model is learned which subsequently gets incrementally refined. In each step, unlabeled examples on which the current model is very confident are labeled with their predictions, added to the training set, and a new model is learned. Similar to self-training, co-training (Blum and Mitchell, 1998) augments the training set by automatically labeled examples. It is a multi-learner algorithm where the learners have independent views on the data and mutually produce labeled examples for each other.

Bootstrapping approaches often fail when applied to NLP tasks where large amounts of training material are required to achieve acceptable performance levels. Pierce and Cardie (2001) showed that the quality of the automatically labeled training data is crucial for co-training to perform well because too many tagging errors prevent a high-performing model from being learned. Also, the size of the seed set is an important parameter. When it is chosen too small data quality gets deteriorated quickly, when it is chosen too large no improvement over the initial model can be expected. To address the problem of data pollution by tagging errors, Pierce and Cardie (2001) propose corrected co-training. In this mode, a human is put into the co-training loop to review and, if necessary, to correct the machine-labeled examples. Although this effectively evades the negative side effects of deteriorated data quality, one may find the correction of labeled data to be as time-consuming as annotations from the scratch. Ideally, a human should not get biased by the proposed label but independently examine the example – so that correction eventually becomes annotation.

In contrast, our SeSAL approach which also applies bootstrapping, aims at avoiding to deteriorate data quality by explicitly pointing human annotators to classification-critical regions. While those regions require full annotation, regions of high confidence are automatically labeled and thus do not require any manual inspection. Self-training and co-training, in contradistinction, select examples of high confidence only. Thus, these bootstrapping methods will presumably not find the most useful unlabeled examples but require a human to review data points of limited training utility (Pierce and Cardie, 2001). This shortcoming is also avoided by our SeSAL approach, as we intentionally select informative examples only.

A combination of active and semi-supervised learning has first been proposed by McCallum and Nigam (1998) for text classification. Committee-based AL is used for the example selection. The committee members are first trained on the labeled examples and then augmented by means of Expectation Maximization (EM) (Dempster et al., 1977) including the unlabeled examples. The idea is to avoid manual labeling of examples whose labels can be reliably assigned by EM. Similarly, co-testing (Muslea et al., 2002), a multi-view AL algorithms, selects examples for the multi-view, semi-supervised Co-EM algorithm. In both works, semi-supervision is based on variants of the EM algorithm in combination with *all* unlabeled examples from the pool. Our approach to semi-supervised AL is different as, firstly, we augment the *training data* using a self-tagging mechanism (McCallum and Nigam (1998) and Muslea et al. (2002) performed semi-supervision to augment the *models* using EM), and secondly, we operate in the sequence labeling scenario where an example is made up of several units each requiring

a label – partial labeling of sequence examples is a central characteristic of our approach. Another work also closely related to ours is that of Kristjansson et al. (2004). In an information extraction setting, the confidence per extracted field is calculated by a constrained variant of the Forward-Backward algorithm. Unreliable fields are highlighted so that the automatically annotated corpus can be corrected. In contrast, AL *selection* of examples together with partial manual labeling of the selected examples are the main foci of our work.

5 Experiments and Results

In this section, we turn to the empirical assessment of semi-supervised AL (SeSAL) for sequence labeling on the NLP task of named entity recognition. By the nature of this task, the sequences – in this case, sentences – are only sparsely populated with entity mentions and most of the tokens belong to the OUTSIDE class[3] so that SeSAL can be expected to be very beneficial.

5.1 Experimental Settings

In all experiments, we employ the linear-chain CRF model described in Section 2 as the base learner. A set of common feature functions was employed, including orthographical (regular expression patterns), lexical and morphological (suffixes/prefixes, lemmatized tokens), and contextual (features of neighboring tokens) ones.

All experiments start from a seed set of 20 randomly selected examples and, in each iteration, 50 new examples are selected using AL. The efficiency of the different selection mechanisms is determined by learning curves which relate the annotation costs to the performance achieved by the respective model in terms of F_1-score. The unit of annotation costs are manually labeled tokens. Although the assumption of uniform costs per token has already been subject of legitimate criticism (Settles et al., 2008), we believe that the number of annotated tokens is still a reasonable approximation in the absence of an empirically more adequate task-specific annotation cost model.

We ran the experiments on two entity-annotated corpora. From the general-language newspaper domain, we took the training part of the MUC7 corpus (Linguistic Data Consortium, 2001) which incorporates seven different entity types, *viz.* persons, organizations, locations, times, dates, monetary expressions, and percentages. From the sublanguage biology domain, we used the oncology part of the PENNBIOIE corpus (Kulick et al., 2004) and removed all but three gene entity subtypes (generic, protein, and rna). Table 1 summarizes the quantitative characteristics of both corpora.[4] The results reported below are averages of 20 independent runs. For each run, we randomly split each corpus into a *pool* of unlabeled examples to select from (90 % of the corpus), and a complementary *evaluation set* (10 % of the corpus).

corpus	entity classes	sentences	tokens
MUC7	7	3,020	78,305
PENNBIOIE	3	10,570	267,320

Table 1: Quantitative characteristics of the chosen corpora

5.2 Empirical Evaluation

We compare semi-supervised AL (SeSAL) with its fully supervised counterpart (FuSAL), using a passive learning scheme where examples are randomly selected (RAND) as baseline. SeSAL is first applied in a default configuration with a very high confidence threshold ($t = 0.99$) without any delay ($d = 0$). In further experiments, these parameters are varied to study their impact on SeSAL's performance. All experiments were run on both the newspaper (MUC7) and biological (PENNBIOIE) corpus. When results are similar to each other, only one data set will be discussed.

Distribution of Confidence Scores. The leading assumption for SeSAL is that only a small portion of tokens within the selected sentences constitute really hard decision problems, while the majority of tokens are easy to account for by the current model. To test this stipulation we investigate the distribution of the model's confidence values $C_{\vec{\lambda}}(y_j^*)$ over all tokens of the sentences (cf. Equation (9)) selected within one iteration of FuSAL. Figure 1, as an example, depicts the histogram for an early AL iteration round on the MUC7 corpus. The vast majority of tokens has a confidence score close to 1, the median lies at 0.9966. Histograms of subsequent AL iterations are very similar with an even higher median. This is so because

[3] The OUTSIDE class is assigned to each token that does not denote an entity in the underlying domain of discourse.

[4] We removed sentences of considerable over and under length (beyond +/- 3 standard deviations around the average sentence length) so that the numbers in Table 1 differ from those cited in the original sources.

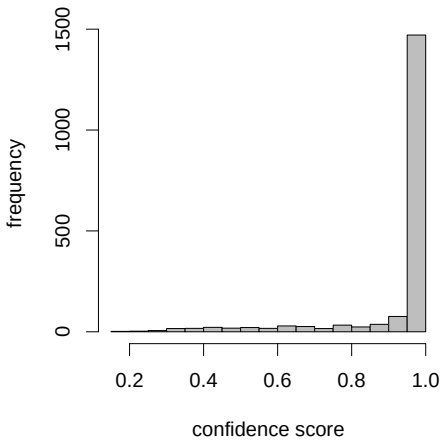

Figure 1: Distribution of token-level confidence scores in the 5th iteration of FuSAL on MUC7 (number of tokens: 1,843)

the model gets continuously more confident when trained on additional data and fewer hard cases remain in the shrinking pool.

Fully Supervised *vs.* Semi-Supervised AL. Figure 2 compares the performance of FuSAL and SeSAL on the two corpora. SeSAL is run with a delay rate of $d = 0$ and a very high confidence threshold of $t = 0.99$ so that only those tokens are automatically labeled on which the current model is almost certain. Figure 2 clearly shows that SeSAL is much more efficient than its fully supervised counterpart. Table 2 depicts the exact numbers of manually labeled tokens to reach the maximal (supervised) F-score on both corpora. FuSAL saves about 50 % compared to RAND, while SeSAL saves about 60 % compared to FuSAL which constitutes an overall saving of over 80 % compared to RAND.

These savings are calculated relative to the number of *tokens* which have to be manually labeled. Yet, consider the following gedanken experiment. Assume that, using SeSAL, every second token in a sequence would have to be labeled. Though this comes to a 'formal' saving of 50 %, the actual annotation effort in terms of the *time* needed would hardly go down. It appears that only when SeSAL splits a sentence into larger

Corpus	$F_{\max}$	RAND	FuSAL	SeSAL
MUC7	87.7	63,020	36,015	11,001
PENNBIOIE	82.3	194,019	83,017	27,201

Table 2: Tokens manually labeled to reach the maximal (supervised) F-score

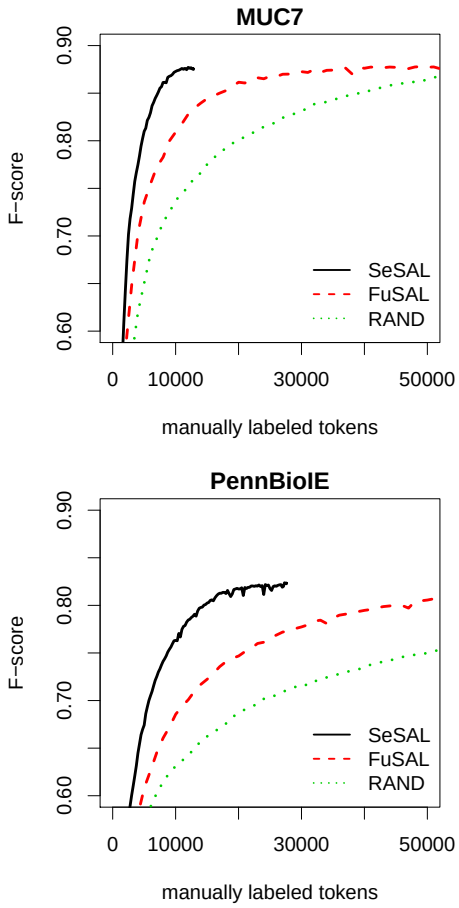

Figure 2: Learning curves for Semi-supervised AL (SeSAL), Fully Supervised AL (FuSAL), and RAND(om) selection

well-packaged, chunk-like subsequences annotation time can really be saved. To demonstrate that SeSAL comes close to this, we counted the number of base noun phrases (NPs) containing one or more tokens to be manually labeled. On the MUC7 corpus, FuSAL requires 7,374 annotated NPs to yield an F-score of 87 %, while SeSAL hit the same F-score with only 4,017 NPs. Thus, also in terms of the number of NPs, SeSAL saves about 45 % of the material to be considered.[5]

Detailed Analysis of SeSAL. As Figure 2 reveals, the learning curves of SeSAL stop early (on MUC7 after 12,800 tokens, on PENNBIOIE after 27,600 tokens) because at that point the whole corpus has been labeled exhaustively – either manually, or automatically. So, using SeSAL the complete corpus can be labeled with only a small fraction of it actually being manually annotated (MUC7: about 18 %, PENNBIOIE: about 13 %).

[5] On PENNBIOIE, SeSAL also saves about 45 % compared to FuSAL to achieve an F-score of 81 %.

Table 3 provides additional analysis results on MUC7. In very early AL rounds, a large ratio of tokens has to be manually labeled (70-80 %). This number decreases increasingly as the classifier improves (and the pool contains fewer informative sentences). The number of tagging errors is quite low, resulting in a high accuracy of the created corpus of constantly over 99 %.

labeled tokens					
manual	automatic	Σ	AR (%)	errors	ACC
1,000	253	1,253	79.82	6	99.51
5,000	6,207	11,207	44.61	82	99.27
10,000	25,506	34,406	28.16	174	99.51
12,800	57,371	70,171	18.24	259	99.63

Table 3: Analysis of SeSAL on MUC7: Manually and automatically labeled tokens, annotation rate (AR) as the portion of manually labeled tokens in the total amount of labeled tokens, errors and accuracy (ACC) of the created corpus.

The majority of the automatically labeled tokens (97-98 %) belong to the OUTSIDE class. This coincides with the assumption that SeSAL works especially well for labeling tasks where some classes occur predominantly and can, in most cases, easily be discriminated from the other classes, as is the case in the NER scenario. An analysis of the errors induced by the self-tagging component reveals that most of the errors (90-100 %) are due to missed entity classes, i.e., while the correct class label for a token is one of the entity classes, the OUTSIDE class was assigned. This effect is more severe in early than in later AL iterations (see Table 4 for the exact numbers).

	labeled		error types (%)		
corpus	tokens	errors	E2O	O2E	E2E
MUC7	10,000	75	100	–	–
	70,000	259	96	1.3	2.7

Table 4: Distribution of errors of the self-tagging component. Error types: OUTSIDE class assigned though an entity class is correct (E2O), entity class assigned but OUTSIDE is correct (O2E), wrong entity class assigned (E2E).

Impact of the Confidence Threshold. We also ran SeSAL with different confidence thresholds t (0.99, 0.95, 0.90, and 0.70) and analyzed the results with respect to tagging errors and the model performance. Figure 3 shows the learning and error curves for different thresholds on the MUC7 corpus. The supervised F-score of 87.7 % is only reached by the highest and most restrictive threshold of $t = 0.99$. With all other thresholds, SeSAL

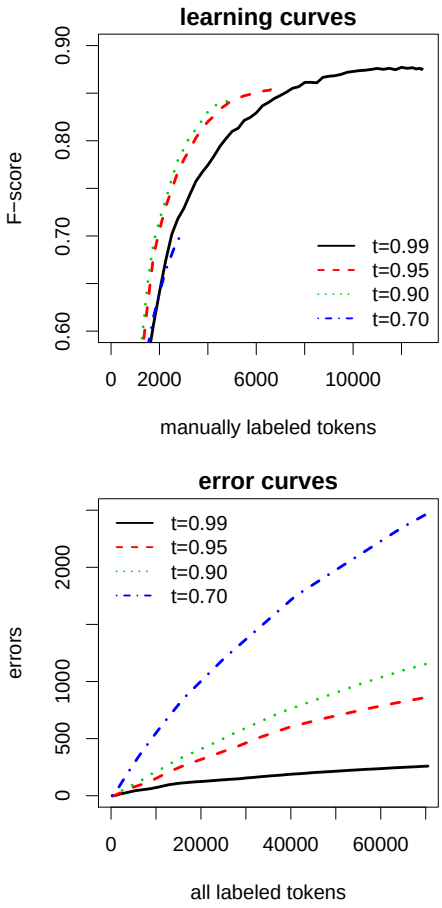

Figure 3: Learning and error curves for SeSAL with different thresholds on the MUC7 corpus

stops at much lower F-scores and produces labeled training data of lower accuracy. Table 5 contains the exact numbers and reveals that the poor model performance of SeSAL with lower thresholds is mainly due to dropping recall values.

threshold	F	R	P	Acc
0.99	87.7	85.9	89.9	99.6
0.95	85.4	82.3	88.7	98.8
0.90	84.3	80.6	88.3	98.1
0.70	69.9	**61.8**	81.1	96.5

Table 5: Maximum model performance on MUC7 in terms of F-score (F), recall (R), precision (P) and accuracy (Acc) – the labeled corpus obtained by SeSAL with different thresholds

Impact of the Delay Rate. We also measured the impact of delay rates on SeSAL's efficiency considering three delay rates (1,000, 5,000, and 10,000 tokens) in combination with three confidence thresholds (0.99, 0.9, and 0.7). Figure 4 depicts the respective learning curves on the MUC7 corpus. For SeSAL with $t = 0.99$, the delay

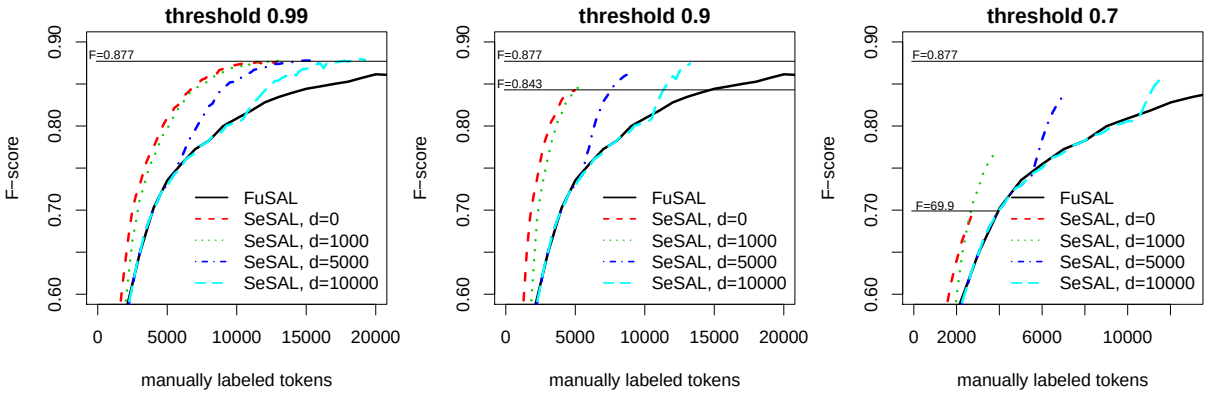

Figure 4: SeSAL with different delay rates and thresholds on MUC7. Horizontal lines mark the supervised F-score (upper line) and the maximal F-score achieved by SeSAL with the respective threshold and $d = 0$ (lower line).

has no particularly beneficial effect. However, in combination with lower thresholds, the delay rates show positive effects as SeSAL yields F-scores closer to the maximal F-score of 87.7 %, thus clearly outperforming undelayed SeSAL.

6 Summary and Discussion

Our experiments in the context of the NER scenario render evidence to the hypothesis that the proposed approach to semi-supervised AL (SeSAL) for sequence labeling indeed strongly reduces the amount of tokens to be *manually* annotated — in terms of numbers, about 60% compared to its fully supervised counterpart (FuSAL), and over 80% compared to a totally passive learning scheme based on random selection.

For SeSAL to work well, a high and, by this, restrictive threshold has been shown to be crucial. Otherwise, large amounts of tagging errors lead to a poorer overall model performance. In our experiments, tagging errors in such a scenario were OUTSIDE labelings, while an entity class would have been correct – with the effect that the resulting models showed low recall rates.

The delay rate is important when SeSAL is run with a low threshold as early tagging errors can be avoided which otherwise reinforce themselves. Finding the right balance between the delay factor and low thresholds requires experimental calibration. For the most restrictive threshold ($t = 0.99$) though such a delay is unimportant so that it can be set to $d = 0$ circumventing this calibration step.

In summary, the self-tagging component of SeSAL gets more influential when the confidence threshold and the delay factor are set to lower values. At the same time though, under these conditions negative side-effects such as deteriorated data quality and, by this, inferior models emerge. These problems are major drawbacks of many bootstrapping approaches. However, our experiments indicate that as long as self-training is cautiously applied (as is done for SeSAL with restrictive parameters), it can definitely outperform an entirely supervised approach.

From an annotation point of view, SeSAL efficiently guides the annotator to regions within the selected sentence which are very useful for the learning task. In our experiments on the NER scenario, those regions were mentions of entity names or linguistic units which had a surface appearance similar to entity mentions but could not yet be correctly distinguished by the model.

While we evaluated SeSAL here in terms of *tokens* to be manually labeled, an open issue remains, namely how much of the real annotation effort – measured by the *time* needed – is saved by this approach. We here hypothesize that human annotators work much more efficiently when pointed to the regions of immediate interest instead of making them skim in a self-paced way through larger passages of (probably) semantically irrelevant but syntactically complex utterances – a tiring and error-prone task. Future research is needed to empirically investigate into this area and quantify the savings in terms of the time achievable with SeSAL in the NER scenario.

Acknowledgements

This work was funded by the EC within the BOOTStrep (FP6-028099) and CALBC (FP7-231727) projects. We want to thank Roman Klinger (Fraunhofer SCAI) for fruitful discussions.

References

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT'98 – Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

S. Engelson and I. 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.

T. Kristjansson, A. Culotta, and P. Viola. 2004. Interactive information extraction with constrained Conditional Random Fields. In *AAAI'04 – Proceedings of 19th National Conference on Artificial Intelligence*, pages 412–418.

S. Kulick, A. Bies, M. Liberman, M. Mandel, R. T. McDonald, M. S. Palmer, and A. I. 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.

J. D. Lafferty, A. McCallum, and F. 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.

D. D. Lewis and J. Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *ICML'94 – Proceedings of the 11th International Conference on Machine Learning*, pages 148–156.

Linguistic Data Consortium. 2001. Message Understanding Conference (MUC) 7. LDC2001T02. FTP FILE. Philadelphia: Linguistic Data Consortium.

A. McCallum and K. Nigam. 1998. Employing EM and pool-based Active Learning for text classification. In *ICML'98 – Proceedings of the 15th International Conference on Machine Learning*, pages 350–358.

I. A. Muslea, S. Minton, and C. A. Knoblock. 2002. Active semi-supervised learning = Robust multi-view learning. In *ICML'02 – Proceedings of the 19th International Conference on Machine Learning*, pages 435–442.

G. Ngai and D. Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *ACL'00 – Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 117–125.

D. Pierce and C. Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *EMNLP'01 – Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1–9.

L. R. Rabiner. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

B. Settles and M. 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.

B. Settles, M. Craven, and L. Friedland. 2008. Active Learning with real annotation costs. In *Proceedings of the NIPS 2008 Workshop on 'Cost-Sensitive Machine Learning'*, pages 1–10.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *COLT'92 – Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 287–294.

K. Tomanek, J. Wermter, and U. 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.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL'95 – Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.