Deep Bayesian Semi-Supervised Active Learning for Sequence Labelling

Tomáš Šabata[1], Juraj Eduard Páll[2], and Martin Holeňa[3]

[1] Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic
tomas.sabata@fit.cvut.cz
[2] Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
palljuraj1@gmail.com
[3] Institute of Computer Science of the Czech Academy of Sciences, Prague, Czech republic
martin@cs.cas.cz

Abstract. In recent years, deep learning has shown supreme results in many sequence labelling tasks, especially in natural language processing. However, it typically requires a large training data set compared with statistical approaches. In areas where collecting of unlabelled data is cheap but labelling expensive, active learning can bring considerable improvement. Sequence learning algorithms require a series of token-level labels for a whole sequence to be available during the training process. Annotators of sequences typically label easily predictable parts of the sequence although such parts could be labelled automatically instead. In this paper, we introduce a combination of active and semi-supervised learning for sequence labelling. Our approach utilizes an approximation of Bayesian inference for neural nets using Monte Carlo dropout. The approximation yields a measure of uncertainty that is needed in many active learning query strategies. We propose Monte Carlo token entropy and Monte Carlo N-best sequence entropy strategies. Furthermore, we use semi-supervised pseudo-labelling to reduce labelling effort.
The approach was experimentally evaluated on multiple sequence labelling tasks. The proposed query strategies outperform other existing techniques for deep neural nets. Moreover, the semi-supervised learning reduced the labelling effort by almost 80% without any incorrectly labelled samples being inserted into the training data set.

Keywords: Active Learning · Semi-supervised Learning · Bayesian Inference · Deep Learning · Sequence Labelling

1 Introduction

Deep learning is achieving state-of-the-art performance in image or video processing, audio processing or natural language processing. However, without using a pretrained model, deep learning typically requires a large amount of data. To

obtain unlabelled input for deep networks in video processing, cameras and other sensors are increasingly available. In natural language processing, a lot of unlabelled inputs can be obtained for almost no cost by gathering them from web sites. Unfortunately, labelling such data is very time consuming and expensive.

In this situation, we can benefit from semi-supervised learning using a large unlabelled dataset along with a small labelled one. Another option is to use active learning wherein each iteration, a part of an annotation budget is spent on labelling the most informative unlabelled samples. The model is retrained including those new samples and the process repeats. The annotation budget is significantly lower than the total number of available unlabelled samples.

Although active learning is a promising way to benefit from unlabelled data, the most common query strategy, uncertainty sampling, requires a measure of uncertainty. In sequence labelling, the measure can be easily defined for statistical models, such as hidden Markov models or conditional random fields (CRF), as they provide a probability of the labelled sequence or a marginal probability distribution for each element of the sequence. For neural networks, defining an uncertainty measure is more complicated since the soft-max activation function, typically used in the last network layer, does not correspond to a real uncertainty of network predictions. To overcome this issue, one can use a Bayesian neural network or include a statistical model, such as CRF, as the last layer of the network.

In sequence labelling, query strategies can be divided into two groups. The first group computes the uncertainty of the sequence predicted by a model. Query strategies of the second group compute uncertainties of separated tokens and then aggregate them to express the uncertainty of the whole sequence.

Querying the most informative sequence means that the annotator has to label every token of the sequence. This is expensive and often not necessary because some tokens can be very reliably annotated automatically. This situation can be found in many natural language processing (NLP) tasks, where some words can be assigned to only one category and we can predict that without knowing the context. A similar situation can be found in a video where two consecutive frames often contain the same or similar information and labelling all frames might be inefficient.

In this paper, we propose an active learning algorithm for sequence labelling with deep neural networks that queries labels of the most informative tokens whereas other labels are labelled automatically.

In the following section, we summarize approaches addressing this topic. In section 3, we define the architecture of our sequence labelling models. In section 4, we describe details of the proposed algorithm. The algorithm is evaluated with experiments on tasks from natural language processing and the results are shown in section 5.

2 Related Work

Sequence labelling models have been used in many areas such as part of speech tagging (POS) or named entity recognition (NER) [25], handwritten recognition [9], protein secondary structure prediction [19], video analysis [39] or facial expression dynamic modeling [4]. In the early years, probabilistic models were the most frequent approach. The most commonly used among them are Hidden Markov models, dynamic Naive Bayesian classifiers, maximum entropy Markov models or Conditional Random fields.

With the increasing amount of data and computational power, and with formulating new network topologies, deep networks are more and more popular in sequence labelling. This is especially true for long short term memory networks (LSTM), which deal well with vanishing gradient problem and are able to incorporate context far from the predicted token. One of the state-of-the-art topologies in sequence labelling is the bi-directional LSTM network (BI-LSTM) [34] or an extended version with a CRF layer on top (BI-LSTM-CRF) [15]. Another interesting topology specific to language processing uses an additional layer (LSTM [23] or CNN [22]) as a character-level embedding for words.

Active Learning in Sequence Labelling was studied intensively for probabilistic models [37]. Query strategies used in AL can be categorized into several groups. Uncertainty Sampling that selects the most uncertain samples, Query by Committee selects samples in which a committee disagree the most, Expected Gradient Length selects samples that would conduct the greatest change to the current model or Fisher information strategy that selects samples that minimize the model variance. These strategies differ in computational complexity and model requirements. The most commonly used strategy, uncertainty sampling, requires the model to return confidence of its predictions. Furthermore, to avoid querying samples that are rather outliers than representative samples, the informativeness of the sample is weighted by its average similarity to all other samples. The technique is called information density [37].

In active learning for sequence labelling, the most informative sequence is labelled. The sequence is then added to the training set, the model is retrained and the process repeats. This requires the whole sequence to be labelled at once. In contrast, Tomanek [40] introduced the *SeSAL* algorithm, where parts of sequences can be labelled automatically. That algorithm was designed for HMMs and CRFs.

Active Learning in Connection with Deep Learning Although active learning has been applied to many ML tasks, application to deep learning is marginal compared to probabilistic modelling. One of the main problems in deep active learning is that many query strategies require some uncertainty estimate, however, most kinds of deep neural networks rarely support it. In literature, we can find several approaches approximating the model posterior: variational inference [8], probabilistic back-propagation [11], Monte-Carlo (MC) dropout [6, 16]

or mixture density networks [3]. With such approximations of uncertainty, active learning has been used in connection with deep learning in image classification [7] or text classification [1]. In the sequence labelling area, active deep learning was successfully used for NER. In [38], a CNN-CNN-LSTM network was used together with active learning in a setup where the whole queried sequence had to be labelled at once.

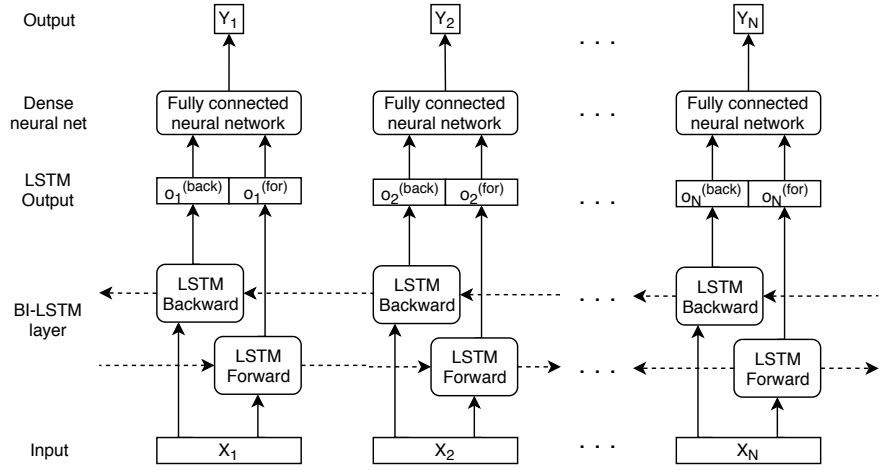
3 Underlying Models

A sequence labelling model assigns categorical labels to all members (tokens) of a sequence of observed values. In general, it considers the optimal label for a given token to be dependent on the choices of nearby tokens. The problem is often simplified through the assumption that the sequence of labels is a Markov chain. With that simplification, the problem can be modelled with a probabilistic graphical model such as a hidden Markov model [29] or a conditional random field. Although the probabilistic models work well on many sequence labelling tasks [21], the Markov property assumption might be too restrictive and unrealistic for problems where a wider context is needed to label tokens correctly. This can be overcome by considering dependencies of higher-order but the computational complexity is growing exponentially with the order which makes these models unusable for real-world problems. Deep learning neural networks can help to overcome the issue of wider context.

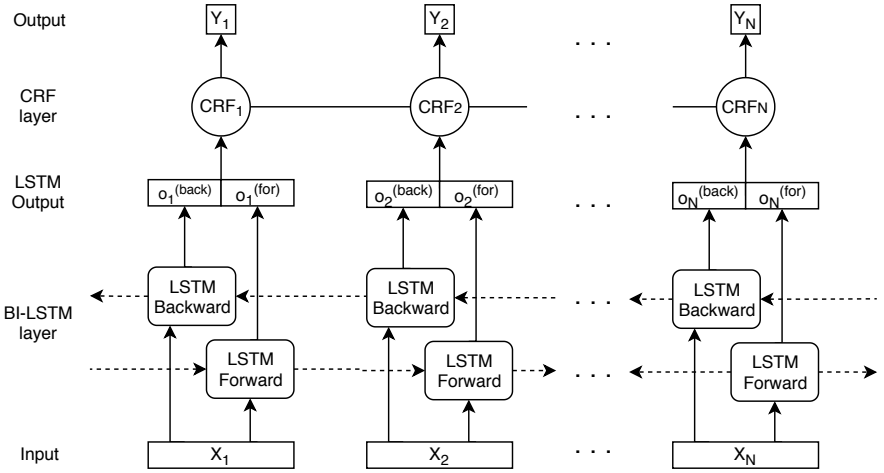
Deep Learning Models. In sequence labelling, various kinds of neural networks are used. These networks are typically designed for a specific task. This is particularly true for their first layers that extract features. In NLP, a character level embedding layer extracts low-level features from the text. In video analysis, feature vectors are extracted using pretrained convolutional networks. After the first layer, a layer that incorporates contextual information from neighbouring elements is plugged in. The most commonly used layers on this level are LSTM cells [13] or gated recurrent unit (GRU) [2]. Last, a sequence decoder layer is used to predict the final sequence. Both context independent layers, for example a fully connected dense layer (BI-LSTM-FCN) (Figure 1a), and contextual layers, for example conditional random fields (BI-LSTM-CRF) (Figure 1b), can be used.

Moreover, to avoid over-fitting, a dropout regularization technique can be used. In our experiments, we use dropout for non-recurrent connections (solid lines in Figure 1). The dropout enabled for each layer allows to estimate prediction uncertainties, as the following section describes.

Bayesian neural networks aim to tackle several drawbacks of neural networks such as overconfidence about their predictions or tendency to overfitting. In classification, the prediction probabilities obtained from the soft-max function are often erroneously interpreted as model confidence [6]. It means, the model



(a) BI-LSTM with fully connected neural network on the top.



(b) BI-LSTM with conditional random fields on the top.

Fig. 1: Schematic representation of deep neural network sequence labelling models used in experiments.

can be uncertain despite high values of the soft-max function and these values require correct calibration [10] before using them as confidences. The main idea of Bayesian neural networks is placing probabilistic distribution over nets' weights [24,26]. However, the approach introduces two issues, intractable inferences and computation costs. Although stochastic variational inference [14,18,27,30] solves the problem with intractable inference, the number of parameters is doubled and it requires more time to converge.

Gal & Ghahramani introduced *Monte Carlo dropout* [6]. They have shown that dropout or various other stochastic regularization techniques can be used to obtain an approximation of Bayesian inference. Consider a sequence of input vectors denoted x to which a sequence of labels denoted y is assigned. A training set containing pairs $\langle x, y \rangle$ is denoted T. Consider a neural net with parameters ω that uses dropout at every layer for the training. Using dropout during testing can be seen as sampling from a model's approximate posterior. This leads to approximate variational inference in which a tractable distribution $q_\theta^*(\omega)$ minimizes the Kullback-Leibler (KL) divergence [20] to the true model posterior $p(\omega|T)$ given a training set T. The prediction uncertainty can be approximated by marginalization over the approximate posterior using Monte Carlo integration:

$$\begin{aligned} p(y = c|x, T) &= \int p(y = c|x, \omega) p(\omega|T) d\omega \\ &\approx \frac{1}{R} \sum_{t=1}^{R} p(y = c|x, \hat{\omega}_t), \end{aligned}$$

where $\hat{\omega}_t \sim q_\theta^*(w)$, R is the number of Monte Carlo runs, and where $q_\theta(w)$ denotes the Dropout distribution [7].

Monte Carlo dropout does not affect the model training complexity, however, each point has to be inferred repeatedly to obtain prediction uncertainty.

4 Active Learning Strategies

Query strategies for sequence labelling models can be divided into several frameworks such as uncertainty sampling (US), query by committee (QbC), expected gradient length (EGL) or information density (ID) [36]. In this section, we describe some of the strategies and propose how they can be used together with the introduced models. The most informative samples are considered to be found by maximising a particular utility function:

$$x^* = \arg\max_x \phi(x).$$

The probability of the sequence y in a by model M given the input sequence x is denoted $P_M(y|x)$. The set of labelled sequences is denoted $\mathcal{L}$ and set of unlabelled sequences is denoted $\mathcal{U}$.

4.1 Query Strategies Utility Functions

Query strategies of the uncertainty sampling framework select the sequences that have the most uncertain label. The uncertainty measure can be expressed in several ways. **Least confidence** query strategy [5] selects the sequence with the lowest probability of the most likely sequence:

$$\phi^{LC}(x) = 1 - P_M(y^*|x),$$

where y^* is the most likely sequence. For CRF, the most likely sequence and its probability can be found using the Viterbi algorithm. For neural nets, the probability of the most likely sequence can be approximated by an empirical probability based on Monte Carlo dropout, which will be denoted $P_M^{\mathrm{MC}}(y|x)$. This empirical distribution is calculated by counting the occurrences of the sequence y for input sequence x in several forward passes through the network, where each forward pass has a different dropout mask. These counts are normalized to sum to 1.

Margin query strategy [33] selects samples where the first and the second most likely sequences have the most similar probabilities. Finding the second most likely sequence in case of probabilistic graphical models requires an updated version of the Viterbi algorithm called N-best Viterbi algorithm [35]. For neural nets, the distribution $P_M^{\mathrm{MC}}(y|x)$ can be used to find the probability of the second most likely path.

The margin query strategy utility function is defined as:

$$\phi^M(x) = -\big(P_M(y_1^*|x) - P_M(y_2^*|x)\big),$$

where y_1^* and y_2^* are first and second the most likely sequences.

Token entropy query strategy [37] uses the Shannon entropy of the model's posteriors:

$$H_M(l) = -\sum_k^K P_M(y_l = k) \log P_M(y_l = k),$$

where y_l is label of the sequence in time l and K is the number of possible labels, over its labellings to define the utility function for selecting the most uncertain sequence:

$$\phi^{TE}(x) = -\frac{1}{L}\sum_{l=1}^{L} H_M(l),$$

where L is the length of the sequence. The utility function is normalized by the length of the sequence. Omitting this normalization, the strategy would lead to querying long sequences as they contain more information. The unnormalized utility function is called **total token entropy**.

Whereas the marginal probability for CRF can be calculated using forward and backward scores, those scores are not available for neural networks. We propose an approximation called **Monte Carlo approximation token entropy**, which uses the idea of Bayesian inference with Monte Carlo [6]:

$$\phi^{TE}_{\mathrm{MC}}(x) = -\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K}\big(P_M^{\mathrm{MC}}(y_l = k) \log P_M^{\mathrm{MC}}(y_l = k)\big).$$

Sequence entropy query strategy computes the entropy of probabilities of all possible sequences. This strategy is unfeasible for long sequences as the number of possible sequences grows exponentially with the length of the sequence. Furthermore, it is not possible to obtain probabilities of a particular sequence

directly in neural network based models. For probabilistic graphical models, the strategy can be approximated with the N-best sequence entropy [17]:

$$\phi^{NSE}(x) = -\frac{1}{C_1}\sum_{\hat{y}\in\mathcal{N}} P_M(\hat{y}|x)\log P_M(\hat{y}|x),$$

where $\mathcal{N} = \{y_1^*, ..., y_N^*\}$ is set of N most likely sequences found by N-best Viterbi algorithm [35] and C_1 normalizes the probabilities to sum to 1.

While the probabilities of N most likely sequences can be obtained directly in probabilistic models, this cannot be done in neural networks. Therefore, we propose a **Monte Carlo approximation of the sequence entropy**:

$$\phi_{\mathrm{MC}}^{NSE}(x) = -\frac{1}{C_2}\sum_{\hat{y}\in\mathcal{N}^{\mathrm{MC}}} P_M^{MC}(\hat{y}|x)\log P_M^{MC}(\hat{y}|x), \tag{1}$$

where $\mathcal{N}^{\mathrm{MC}} = \{y_1, y_2, \dots\}$ is set of all sequences predicted by Monte Carlo sampling and C_2 normalizes the probabilities to sum to 1.

In the query by committee framework, a committee of models $\mathcal{C} = \{M^{(1)}, ..., M^{(C)}\}$, representing different hypotheses, is maintained during the whole process of learning. The committee is used to query the sequence over which the members are most in disagreement about how to label it. The committee is usually trained using bagging. In each round, the labelling set is sampled with replacement to create a unique training set $L^{(C)}$ that is used to train model $M^{(C)}$. The committee prediction is obtained by models voting. In the context of deep neural networks, maintaining a committee is too expensive for practical use. Although dropout can be considered as a form of bagging [12], we do not deal with the framework in the paper.

US and QbC strategies are prone to querying outliers as they are often uncertain for the model and the committee of models often disagrees about them. The framework, called **information density**(ID), can be used to avoid this problem. ID uses a base utility function $\phi_B(x)$ and weights it by samples' representativeness. All above defined utility functions can be used as base utility functions. ID utility function is defined:

$$\phi^{ID}(x) = \phi_B(x) \times \Big(\frac{1}{|\mathcal{U}|}\sum_{u=1}^{|\mathcal{U}|}\mathrm{sim}(x, x^{(u)})\Big)^{\beta}, \tag{2}$$

where $\mathrm{sim}(x, x^{(u)})$ is a chosen similarity function for two sequences and β a parameter that controls a relative importance of the representativeness term. The similarity measure differs from task to task.

4.2 Token-Level Semi-Supervised Active Learning

In the standard AL approach, the annotator has to label the whole sequence although the sequence can contain subsequences that do not add too much value to the utility function. If the model is sufficiently learned, these subsequences can

be easily annotated automatically using model inference. The decision whether a token can be labelled automatically can rely on some kind of model confidence [40] or the disagreement about the most probable paths [31].

We propose to use a combination of active and semi-supervised learning. For models with CRF layer on the top, marginal probability represents the model prediction confidence. Otherwise, the Monte Carlo dropout estimates the model prediction confidence. First, the most informative sequence is found with a chosen query strategy. Tokens in which the model is confident are automatically labelled using semi-supervised learning and the rest is given to an annotator. The labelled sequence is added to the training set and the process repeats. Details of the approach are described in Algorithm 1. The confidence threshold has to be chosen according to the model, problem type and query strategy.

Algorithm 1: Sequential semi-supervised AL framework

Input:
$\mathcal{L}$: labelled set
$\mathcal{U}$: unlabelled set
$\phi(\cdot)$: query strategy utility function
θ: confidence threshold
M: model type
begin
  train model m of type M on data set L
  while *stopping criterion is not met* **do**
    // Find the most informative sequence from $\mathcal{U}$
    $x^* = \text{argmax}_{x \in \mathcal{U}} \phi(x)$
    // label the sequence with the model or query the annotator
    $\hat{y} = m(x^*)$
    for $i = 1$ *to length of* x^* **do**
      if $P_m(y_i = \hat{y}_i | x^*) > \theta$ **then**
        $y_i^* = \hat{y}_i$
      else
        $y_i^* = \text{query}(x_i^*)$
      end
    end
    $\mathcal{L} = \mathcal{L} \cup \langle x^*, y^* \rangle$
    $\mathcal{U} = \mathcal{U} \setminus x^*$
    retrain model m on $\mathcal{L}$
  end
end

5 Experiments

To evaluate the performance of the proposed approach, we have chosen three different sequence labelling problems: named entity recognition(NER), part of

speech tagging (POS) and chunking. Experiments were performed with two sequential models: BI-LSTM-FCN with Monte Carlo dropout and BI-LSTM-CRF, and various query strategies designed for each of those models.

In the paper, we report two experiments. The first experiment tests proposed query strategies against random sampling and least confident query strategies as a baseline. The second experiment is using sequential semi-supervised active learning framework to reduce the labelling effort. Our primary aim was reducing the amount of labelled data required for training, rather than labelling performance. Therefore, we did not extensively optimize hyper-parameters such as learning rate, batch size or momentum.

5.1 Experiment Design

The experiments were performed on the publicly available benchmark dataset CoNLL 2003 [32]. The dataset provides a predefined training set and two testing sets for POS, NER and Chunking. We report performance for the testing set A. The training set was randomly divided into a labelled set and an unlabelled set in the ratio 1:9.

Both models use GloVe embeddings [28] where each word is represented by a vector of length 300. The models contain two LSTM hidden layers with size 100 and dropout with probability 0.4 applied to all layers. The last layer of the BI-LSTM-FCN is linear and uses the soft-max activation function. The number of forward passes for computing P_{MC} was set to 500.

First, each model was trained on the labelled training set with 10% of the original size for 30 epochs. This model was used for experiments with all query strategies. For each query strategy, the model was used to find the most likely paths and their scores together with tokens prediction confidences for all unlabelled sequences. The most informative sequences were selected and annotated until the annotation budget was exhausted. We have defined the annotation budget of one AL cycle in two ways: the number of labelled sequences and the total number of annotated tokens. In the first scenario, 100 sequences were selected and annotated, whereas, in the second scenario, sequences were annotated until the total number of annotated tokens reached 1000. With the updated labelled training set, the model was updated by iterative training for one epoch, then the new score was calculated. This active learning cycle was repeated 20 times. In the second experiment, samples were sorted according to their confidences, and the threshold value was chosen to achieve 0% or 1% of incorrectly labelled samples.

Early results showed that proposed query strategies are prone to select outliers. Therefore, the information density wrapping strategy was used for all of them. Each sequence was represented by the average of embedding vectors. The representativeness of the sequence was computed as an average cosine distance to all other sequences in the unlabelled dataset. The cosine distance is claimed to be an efficient similarity measure of the linguistic or semantic similarity of corresponding words for the chosen embedding [28]. In the results, we use the names of base query strategies for clarity.

5.2 Results

In experiments, we studied the achieved performance in terms of F-measure (specifically F1 score) and accuracy by particular query strategies and the number of tokens that can be labelled automatically by semi-supervised learning. We report the macro-averaged F1 score that is calculated:

$$F1_{\text{macro}} = \frac{1}{|Q|} \sum_{q \in Q} F1_q,$$

where Q is the set of all possible labels and $F1_q$ is the F1 score for the class labeled q considered as the positive class and all remaining classes as the negative class.

These scores were compared with the models learned on the whole labelled dataset and models learned on a small labelled dataset that was later used for active learning.

Query Strategies Comparison Query strategies were compared in two AL scenarios: an unlimited number of tokens and a limited number of tokens. Table 1a shows that the strategies MC total token entropy and MC sequence entropy outperforms other strategies in both F1 score and accuracy. The MC total token entropy, however, required more tokens to be labelled. In NER, it queried almost twice as many tokens. In the scenario with a limited number of tokens, the MC sequence entropy dominates over other strategies except in the Chunking.

Table 1b shows that for the BI-LSTM-CRF model, the least confident and total token entropy query strategies have shown better results compared to the token entropy query strategy. We conclude that total token entropy query strategy dominates in the scenario with an unlimited number of tokens, whereas least confident achieves better results in the scenario with a limited number of tokens. The sequence entropy query strategy is missing as our implementation was lacking n-best Viterbi algorithm.

Moreover, Table 1b shows that the MC sequence entropy query strategy is the best among the compared strategies in the NER and POS tasks during the whole AL loop if the number of annotated tokens is limited in each cycle of the AL loop (Figures 2a and 2b).

Active Learning in Combination with Semi-supervised Learning Last, we studied a possible reduction of the labelling effort using semi-supervised learning. We report how many tokens were automatically labelled if the threshold is set to not allow errors propagate into the training dataset and if 1% of errors are allowed. The results in Table 2 indicate that the BI-LSTM-CRF model has a more reliable uncertainty measure for the marginal distribution than the BI-LSTM-FCN model. It can reduce the labelling effort up to almost 80% without any incorrectly labelled samples being inserted into the training data set. The labelling effort is reduced up to almost 84% with 1% of incorrectly labelled samples being inserted into the training dataset.

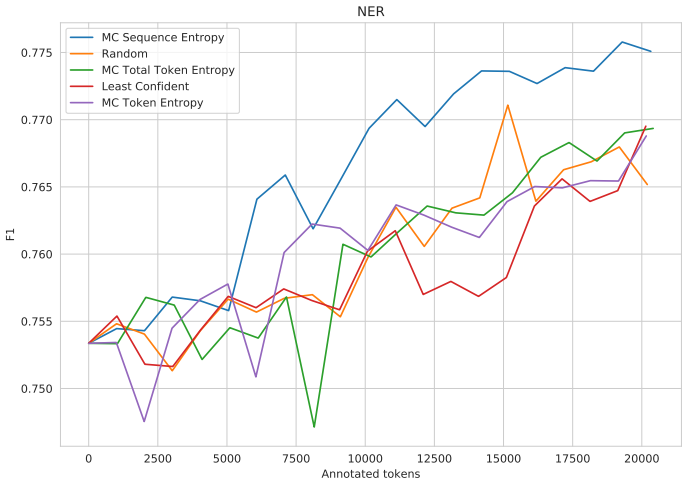
Table 1: Comparison of query strategies for BI-LSTM-FCN. The column 'Tokens' represents the ratio of labelled tokens to the number of all tokens in sequences from the complete dataset. The percentage sign is omitted. The first two lines of the table report performance of the supervised model trained on complete dataset and dataset with only 10% of training data available respectively.

(a) BI-LSTM-FCN

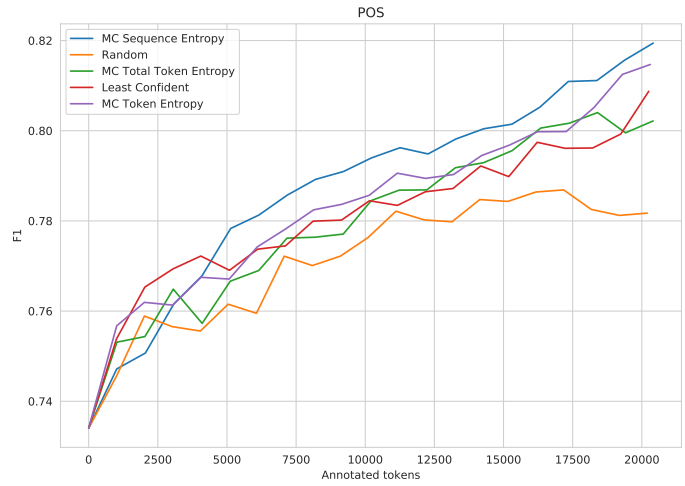
	NER			POS			Chunking		
	F1	Acc	Tokens	F1	Acc	Tokens	F1	Acc	Tokens
No active learning									
BI-LSTM-FCN	85.3	98.6	100	85.3	95.4	100	70.5	96.0	100
	75.4	97.7	10	74.6	92.0	10	53.7	93.7	10
Active learning with an unlimited number of tokens									
Random	73.5	98.1	24	79.6	93.2	24	54.8	94.7	25
Least Confident	76.2	98.0	17	83.2	93.7	39	57.8	95.0	33
MC Token Entropy	77.2	98.2	24	82.7	93.7	40	57.8	94.8	36
MC Total Token Entropy	**82.4**	98.3	40	83.0	93.7	45	**63.4**	95.0	45
MC Sequence Entropy	78.0	**98.4**	26	**83.3**	**94.0**	41	59.7	95.0	37
Active learning with a limited number of tokens									
Random	76.5	98.0	20	78.1	93.3	20	53.3	94.5	20
Least Confident	76.9	98.0	20	80.8	93.1	20	55.2	94.5	20
MC Token Entropy	76.8	98.1	20	81.5	93.0	20	55.3	94.5	20
MC Total Token Entropy	76.9	98.1	20	80.2	93.2	20	**56.7**	94.4	20
MC Sequence Entropy	**77.5**	**98.3**	20	**82.0**	**93.4**	20	55.3	94.4	20

(b) BI-LSTM-CRF

	NER			POS			Chunking		
	F1	Acc	Tokens	F1	Acc	Tokens	F1	Acc	Tokens
No active learning									
BI-LSTM-CRF	85.5	98.7	100	82.3	95.3	100	58.1	95.9	100
	75.9	97.8	10	72.8	92.0	10	50.2	93.7	10
Active learning with unlimited number of tokens									
Random	76.8	98.1	24	75.2	93.3	24	50.7	94.6	24
Least Confident	**78.4**	**98.6**	33	81.4	**94.0**	40	56.0	**95.2**	37
Token Entropy	77.3	98.3	23	75.0	93.0	17	55.9	94.9	20
Total Token Entropy	78.0	98.5	32	**82.0**	**94.0**	39	**56.3**	95.1	40
Active learning with limited number of tokens									
Random	76.4	98.0	20	75.6	93.2	20	55.7	94.4	20
Least Confident	77.4	**98.3**	20	**82.1**	93.3	20	**57.0**	94.5	20
Token Entropy	77.4	**98.3**	20	75.6	**93.6**	20	56.4	**94.7**	20
Total Token Entropy	**77.5**	**98.3**	20	79.3	93.3	20	56.5	94.6	20



(a) NER



(b) POS

Fig. 2: Query strategies comparison for NER and POS for BI-LSTM-FCN in the scenario with fixed number of annotated tokens.

6 Conclusions and Future Work

In this paper, we presented an application of Monte Carlo dropout, an approximation of Bayesian inference for deep neural networks, to active learning strate-

Table 2: Proportion of data labelled automatically by pseudo-labelling.

Task type		NER		POS		CHUNK	
Allowed errors		0%	1%	0%	1 %	0 %	1 %
BI-LSTM-FCN	Least confident	6.9	21.5	0.2	1.7	3.5	10.5
	Sequence entropy	14.2	28.7	0.3	2.0	3.6	10.5
	Total token entropy	7.8	12.8	0.2	1.4	2.0	6.2
	Token entropy	9.0	18.7	0.2	1.6	2.6	8.3
BI-LSTM-CRF	Least confident	77.3	84.3	72.4	82.5	66.6	79.6
	Token entropy	79.5	83.7	72.2	77.9	72.9	79.3
	Total token entropy	75.5	83.5	71.2	81.6	65.3	78.3

gies developed for probabilistic graphical models. We proposed two not yet used adaptations of token entropy and sequence entropy query strategies suitable for LSTM-type deep neural networks. Moreover, we tested a combination of active and semi-supervised learning for sequence labelling for that network.

The proposed query strategies have shown a substantial improvement over the until now used strategy in sequence labelling with deep neural networks, least confident. The proposed strategies outperformed the least confident in all three considered sequence labelling tasks in case of the network without a CRF layer. This is particularly true, if the annotation budget is limited for each active learning batch, which is a typical real-world situation.

The combination of active and semi-supervised learning allows us to achieve up to 80% labelling cost reduction for the BI-LSTM-CRF model. The uncertainty measure based on Monte Carlo dropout, however, still needs improvement to achieve labelling effort reduction comparable with BI-LSTM-CRF. To this end, we would like to study uncertainty measures provided by other approaches to Bayesian recurrent neural networks.

Although uncertainty sampling has shown to be applicable to deep neural networks, other active learning frameworks have not been enough studied. In the future, we would like to study, in the context of sequence labelling and deep neural networks, active learning based on expected gradient length. In addition to this, we would like to apply deep active learning to sequence labelling in video processing, where context is also very important information.

Acknowledgements

The work has been supported by the grant 18-18080S of the Czech Science Foundation (GACR).

References

1. Burkhardt, S., Siekiera, J., Kramer, S.: Semi-supervised bayesian active learning for text classification. In: Bayesian Deep Learning (2018)

2. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
3. Choi, S., Lee, K., Lim, S., Oh, S.: Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6915–6922. IEEE (2018)
4. Cohen, I., Sebe, N., Garg, A., Chen, L.S., Huang, T.S.: Facial expression recognition from video sequences: temporal and static modeling. Computer Vision and image understanding **91**(1-2), 160–187 (2003)
5. Culotta, A., McCallum, A.: Reducing labeling effort for structured prediction tasks. In: AAAI. vol. 5, pp. 746–751 (2005)
6. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
7. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1183–1192. JMLR. org (2017)
8. Graves, A.: Practical variational inference for neural networks. In: Advances in neural information processing systems. pp. 2348–2356 (2011)
9. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Advances in neural information processing systems. pp. 545–552 (2009)
10. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1321–1330. JMLR. org (2017)
11. Hernández-Lobato, J.M., Adams, R.: Probabilistic backpropagation for scalable learning of bayesian neural networks. In: International Conference on Machine Learning. pp. 1861–1869 (2015)
12. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
14. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine Learning Research **14**(1), 1303–1347 (2013)
15. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
16. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. pp. 5574–5584 (2017)
17. Kim, S., Song, Y., Kim, K., Cha, J.W., Lee, G.G.: Mmr-based active machine learning for bio named entity recognition. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. pp. 69–72. Association for Computational Linguistics (2006)
18. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
19. Krogh, A., Larsson, B., Von Heijne, G., Sonnhammer, E.L.: Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. Journal of molecular biology **305**(3), 567–580 (2001)

20. Kullback, S.: Information theory and statistics. Courier Corporation (1997)
21. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
22. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
23. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
24. MacKay, D.J.: A practical bayesian framework for backpropagation networks. Neural computation **4**(3), 448–472 (1992)
25. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
26. Neal, R.M.: Bayesian learning for neural networks, vol. 118. Springer Science & Business Media (2012)
27. Paisley, J., Blei, D., Jordan, M.: Variational bayesian inference with stochastic search. arXiv preprint arXiv:1206.6430 (2012)
28. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), `http://www.aclweb.org/anthology/D14-1162`
29. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
30. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
31. Šabata, T., Borovicka, T., Holena, M.: K-best viterbi semi-supervized active learning in sequence labelling. CEUR workshop proceedings pp. 144–152 (2017)
32. Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
33. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden markov models for information extraction. In: International Symposium on Intelligent Data Analysis. pp. 309–318. Springer (2001)
34. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
35. Seshadri, N., Sundberg, C.E.: List viterbi decoding algorithms with applications. IEEE transactions on communications **42**(234), 313–323 (1994)
36. Settles, B.: Active learning literature survey. Science **10**(3), 237–304 (1995)
37. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing. pp. 1070–1079. Association for Computational Linguistics (2008)
38. Shen, Y., Yun, H., Lipton, Z., Kronrod, Y., Anandkumar, A.: Deep active learning for named entity recognition. Proceedings of the 2nd Workshop on Representation Learning for NLP (2017). https://doi.org/10.18653/v1/w17-2630, `http://dx.doi.org/10.18653/v1/W17-2630`
39. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International conference on machine learning. pp. 843–852 (2015)
40. Tomanek, K., Hahn, U.: Semi-supervised active learning for sequence labeling. In: 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 AFNLP: Volume 2 - Volume 2. pp. 1039–1047. ACL '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)