EASAL: Entity-Aware Subsequence-Based Active Learning for Named Entity Recognition

Yang Liu[1,2], Jinpeng Hu[1,2†],Zhihong Chen[1,2],Xiang Wan[1,3†], Tsung-Hui Chang[1,2]
[1]Shenzhen Research Institute of Big Data
[2]Chinese University of Hong Kong, Shenzhen, China
[3] Pazhou Lab, Guangzhou, 510330, China
{yangliu5, jinpenghu, zhihongchen}@link.cuhk.edu.cn,
wanxiang@sribd.cn, changtsunghui@cuhk.edu.cn

Abstract

Active learning is a critical technique for reducing labelling load by selecting the most informative data. Most previous works applied active learning on Named Entity Recognition (token-level task) similar to the text classification (sentence-level task). They failed to consider the heterogeneity of uncertainty within each sentence and required access to the entire sentence for the annotator when labelling. To overcome the mentioned limitations, in this paper, we allow the active learning algorithm to query subsequences within sentences and propose an **E**ntity-**A**ware **S**ubsequences-based **A**ctive **L**earning (EASAL) that utilizes an effective *Head-Tail* pointer to query one entity-aware subsequence for each sentence based on BERT. For other tokens outside this subsequence, we randomly select 30% of these tokens to be pseudo-labelled for training together where the model directly predicts their pseudo-labels. Experimental results on both news and biomedical datasets demonstrate the effectiveness of our proposed method. The code is released at https://github.com/lylylylylyly/EASAL.

Introduction

Named Entity Recognition (NER) is a fundamental task for text analysis, which aims to identify named entities (NEs), such as person, location, organization, etc., in general text or diseases, chemicals, genes, etc., in biomedical text and is essential in many downstream natural language processing (NLP) tasks. In recent years, there has been a maintained enthusiasm in the NER task (Yoon et al. 2019; Cho and Lee 2019; Hakala and Pyysalo 2019; Lee et al. 2020; Kocaman and Talby 2021; Wang et al. 2021; Hu et al. 2022a,b). Most of these models require a large amount of annotated data to achieve high performance, while constructing such datasets is usually high-costing and time-consuming, which motivates the idea of how to achieve higher model performance with the least amount of data.

As a straightforward solution to the mentioned problem, Active Learning (AL) is a critical technique to effectively select the most informative data for training the model by some specific query functions, e.g., Least Confidence (Felder and Brent 2009) (LC), or Maximum Normalized Log-Probability (Siddhant and Lipton 2018) (MNLP). AL methods have been demonstrated to be effective in classification problems in Computer Vision (CV) and Natural Language Processing (NLP) fields such as image classification or text classification. Thanks to these successes, follow-up work applies AL to the task of sequence labelling paradigm, e.g., NER, by treating one sentence as one query object (Shen et al. 2017; Zhang, Yu, and Zhang 2020; Radmard, Fathullah, and Lipani 2021; Hazra et al. 2021; Naseem et al. 2021).

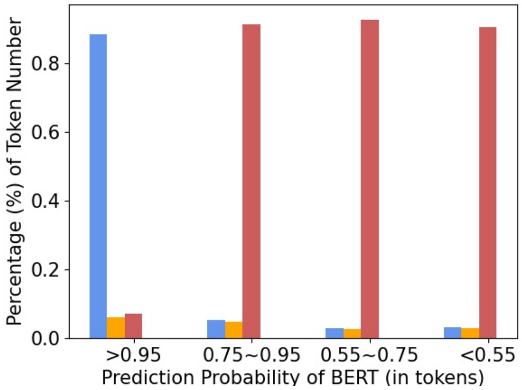


Figure 1: Three percentage values, including the percentage of tokens in each probability segment to the total token number (Blue), the percentage of tokens with entity labels in each probability segment to the total token number (Orange), the percentage of tokens with entity labels in each probability segment to the segment token number (Red).

However, one big difference between the NER task and the image classification or text classification task is that NER is a token-level task, not a sentence-level task since the model has to give each token a label in sentences. Budget wasting may arise from the heterogeneity of uncertainty across each sentence; a sentence can contain multiple subsequences (of tokens), where the model is certain on some and uncertain on others. In this case, if the model still queries the entire sentence, it will not maximize the amount of information that can be queried each time, which is also described in Radmard, Fathullah, and Lipani (2021). In order to prove that budget wasting does exist in full-sentence-based query method, we take an intuitive analysis on Conll2003 (Sang and De Meulder 2003) dataset. Specially, we first randomly select

[†]Corresponding author.

1% full sentences from Conll2003 and train a model with this 1% data. Then, prediction probabilities (i.e., maximum entity class probability) of each token in remaining sentences (i.e., 99% in Conll2003) are obtained on the trained model. We count three distinct percentage values shown in Fig. 1. Denote the total token number of remaining sentences as N; the number of tokens in each probability segment as $N_i, i \in \{> 0.95, 0.75 \sim 0.95, 0.55 \sim 0.75, < 0.55\}$; the number of tokens with entity labels in each probability segment as $N_i^{label}, i \in \{> 0.95, 0.75 \sim 0.95, 0.55 \sim 0.75, < 0.55\}$. In the form of a formula, Blue, Orange, and Red in Fig. 1 represent $N_i/N(\%)$, $N_i^{label}/N(\%)$, and $N_i^{label}/N_i(\%)$, respectively. Nearly 90% of tokens get a high prediction probability (>0.95) with the model, and most of these tokens have the true label "O". Conversely, only about 10% of tokens get a lower prediction probability, and these tokens are always entities. Therefore, it is reasonable to focus on those subsequences containing tokens with low prediction probabilities. We call them "entity-aware subsequences". Moreover, thanks to the characteristics of the token-level task, annotators often do not need to read the entire sentence when annotating entities. In contrast, they only need a subsequence to complete the annotation, which provides the feasibility for selecting subsequences with AL.

To maximize the annotation information under a limited number of tokens, we proposed an **E**ntity-**A**ware **S**ubsequences-based AL, named EASAL, based on BERT. EASAL utilizes an effective *Head-Tail* pointer algorithm to find one entity-aware subsequence for each sentence. The query is performed by sorting these subsequences in a descending order given the subsequence-wise uncertainty score. We select the top ones and train these selected subsequences with their actual labels after each query execution. For the other tokens in each sentence, we randomly select 30% to label them with pseudo-labels, and the model directly predicts the pseudo-labels after the previous training. The remaining tokens are ignored, and no loss calculation and backpropagation are performed. For three biomedical datasets and one news dataset, we conduct comprehensive experiments and analyses on the effect of EASAL.

Contributions of this paper are:

(1)A general AL framework is proposed based on BERT for NER, which is easy to migrate to other domains by replacing the language model in the domain.

(2) A Entity-Aware Subsequences-based AL method (EASAL) is proposed, which utilizes *Head-Tail* pointer to find one entity-aware subsequence for each sentence to maximize the annotation information under a limited number of tokens.

(3) Taking a comprehensive analysis based on BERT, which provides a reasonable explanation of the effectiveness of the subsequences-based AL method on NER.

Background

Active Learning

Define $\mathcal{D}$ as the data pool, $\mathcal{D}_0$ as the initial randomly selected data to be annotated, $\mathcal{D}_s$ as the selected data in each round of AL to be annotated, $\mathcal{M}$ as the model, QF as the query function, $\mathcal{EP}$ as the expert. The standard active learning process can be formulated as: **(i)** Randomly select initial data $\mathcal{D}_0$ and delete them from the data pool $\mathcal{D} = \mathcal{D} - \mathcal{D}_0$, then the expert will annotate the data $\mathcal{D}_0$ for initial training $\mathcal{M} = \mathcal{M}(\mathcal{EP}(\mathcal{D}_0))$; **(ii)** The AL process begins at this step by $\mathcal{D}_s = \mathcal{QF}(\mathcal{M}(\mathcal{D}))$, where $\mathcal{D}_s$ denotes the selected data at the AL process. Next, annotate $\mathcal{D}_s$ and delete the newly selected data $\mathcal{D}_s$ from the data pool $\mathcal{D}$; **(iii)** Fuse the labeled $\mathcal{D}_s$ with the previous training data to obtain a new training set, and perform training with model $\mathcal{M}$; **(iv)** Repeat Step (ii) and Step (iii).

Following the previous active learning researches, we conduct experiments on labeled datasets, where the labels are equivalent to the results of the expert annotations. We utilize the following three different AL query strategies for NER: Let $X = x_1, x_2, x_3, ..., x_n$ be the input sequence, where n represents the length of the input and $Y = y_1, y_2, y_3, .., y_n$ be the labels. Then, $p_{i,c} = P_\theta(y_i = c|X; y_0, y_1, ..., y_{i-1})$ can be the token-wise probability assigned by a model $\mathcal{M}$ with parameters θ to label y for a given input x, where $i = 0, 1, ..., n$, $c \in \mathcal{C}$ and $\mathcal{C}$ is the label set.

Query Functions

Instances in the unlabelled pool are queried using a query function. This function aims to quantify the uncertainty of the model when generating prediction probabilities over possible labels for each instance. Instances with the highest predictive uncertainty are deemed as the most informative for model training. Previously used query functions such as Least Confidence (LC) and Maximum Normalized Log-Probability (MNLP) are generalised for variable length sequences. To better elicit the query function we use, we first give the definition of the token-wise LC score:

$$LC_i = -\max_{c \in \mathcal{C}} \; log\, p_{i,c} \tag{1}$$

The LC query function for sequences is then defined as:

$$LC(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} LC_i, \tag{2}$$

and for MNLP as:

$$MNLP(x_1, x_2, ..., x_n) = \frac{1}{n} \sum_{i=1}^{n} LC_i. \tag{3}$$

Intuitively, MNLP is an extension version of LC, which points out that LC is more inclined to query long sentences. In order to correct this defect, MNLP proposed to utilize the normalized method with respect to the sentence length. Since this work focuses on the querying of subsequences, we generalize the previously defined LC and MNLP to a family of query functions applicable for both full sentences and subsequences(Radmard, Fathullah, and Lipani 2021):

$$LC_\alpha(x_{i+1}, ..., x_{i+l}) = \frac{1}{l^\alpha} \sum_{j=i+1}^{i+l} LC_j, \tag{4}$$

where α is a balance factor considering sequence length. Special cases are when $\alpha = 0$ and $\alpha = 1$ which return the original definitions of LC and MNLP. Sort unlabelled instances in a descending order and select the specified number of instances in the front for the expert annotation.

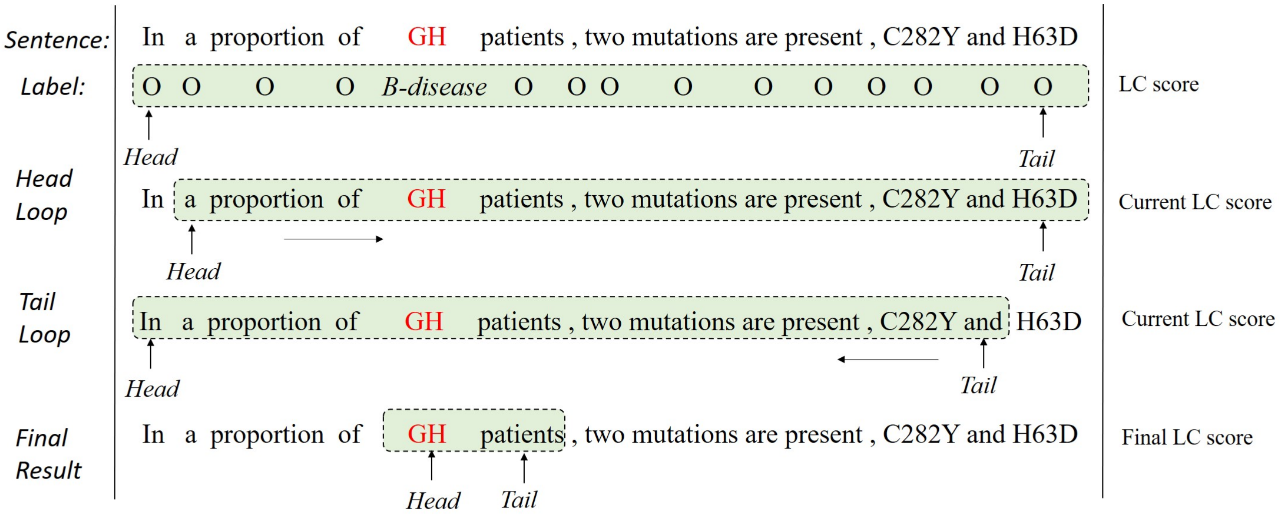

Sentence:
In a proportion of GH patients , two mutations are present , C282Y and H63D
Label:
O O O O B-disease O O O O O O O O O O
LC score
Head
Tail
Head Loop
In a proportion of GH patients , two mutations are present , C282Y and H63D
Current LC score
Head
Tail
Tail Loop
In a proportion of GH patients , two mutations are present , C282Y and H63D
Current LC score
Head
Tail
Final Result
In a proportion of GH patients , two mutations are present , C282Y and H63D
Final LC score
Head
Tail

Figure 2: Illustration of the *Head-Tail* pointer in EASAL, where the LC score represents the uncertainty score.

Proposed Method

In this section, we present the details of our proposed method. To maximize the annotation information under a limited number of tokens, we proposed an Entity-Aware Subsequences-based Active Learning named EASAL. EASAL utilizes an effective *Head-Tail* pointer structure to query one entity-aware subsequence for each sentence. In training, after each execution of the query strategy, tokens in the selected entity-aware subsequence in each sentence are annotated and trained with true labels. For the other tokens in each sentence, we randomly select 30% to label them with pseudo-labels, and the model directly predicts the pseudo-labels after the last training. The remaining tokens are ignored, and no loss calculation and backpropagation are performed on them during current training.

Head-Tail Pointer Algorithm

In this part, we describe the details of how the model selects one entity-aware subsequence for each sentence with the *Head-Tail* pointer. Fig. 2 shows the implementation of *Head-Tail* pointer. Specifically, in each query round, the model, trained with the initial instances, first obtains the LC score (i.e., uncertainty score) for the entire sentence with $\alpha = 1$ in Eq. 4. Note that for the calculation process of the *Head-Tail* pointer, we always set $\alpha = 1$ in Eq. 4 to obtain the LC score, written as LC-score in the following text, for desired sequences or subsequences. Denote the obtained initial LC-score as $\{S_0^i, i = 1, ..., K\}$, where K is the total number of sentences in the data pool. For each sentence, a *Head* pointer $\mathcal{PT}_h$ is initialized to point to the beginning of the sentence, and a *Tail* pointer $\mathcal{PT}_t$ is initialized to point to the end of the sentence, which is shown in the first row in Fig. 2. Then, the *Head* loop begins by moving the *Head* pointer one unit to the right as shown in the second row in Fig. 2. We get a subsequence that does not contain the first token and calculate the LC-score of this subsequence denoting as S_h^i.

Then, the *Head* loop is executed as follows:

- **(i)** If $S_h^i \leq S_0^i$, it means that the token-wise LC-score of the first token is large that represents the uncertainty of the first token is high, and the model can not predict its label well, so withdraw the *Head* pointer one unit to the left and end the *Head* loop.
- **(ii)** If $S_h^i > S_0^i$, it means that the token-wise LC-score of the first token is small that represents the uncertainty of the first token is low, and the model can predict its label well, so update $S_0^i = S_h^i$ and continue to move the *Head* pointer to the right. Repeat this step until $S_h^i \leq S_0^i$.

The *Tail* loop is executed in the same way as the *Head* loop, except that the *Tail* loop begins by moving the *Tail* pointer to the left shown in the third row in Fig. 2. Finally, when the *Head* loop and the *Tail* loop are all over, the subsequence between the *Head* pointer and the *Tail* pointer is the final selected subsequence in the sentence as shown in the fourth row in Fig. 2.

Note that we complete loop processes first. If *Head* and *Tail* pointers stop at the same position, there is no subsequence to query. Otherwise, if the subsequence is less than l_{min}, we move *Head* and *Tail* pointers outward to lengthen the subsequence.

Querying, Deduplication and Sampling

In each query round, we sort these subsequences obtained by *Head-Tail* Pointer based on their LC scores and select a limited number of subsequences with larger LC scores. We construct a subsequence dictionary to prevent information redundancy caused by querying the same subsequence. When a subsequence is found, we first determine whether the subsequence segment is in the dictionary. If not, add the sequence segment to the dictionary; otherwise, this subsequence will not be queried.

Since a subsequence-based query algorithm can result in partially labelled sentences, it raises the question of how unlabelled tokens should be handled. We first ensure that loss computation and backpropagation occur from tokens with true labels in selected subsequences during the model training. For the remaining tokens, we randomly select a certain

Algorithm 1: Entity-Aware Subsequence-Based AL

```
Data: $\mathcal{D}$: task dataset;
Input: $D_0 \leftarrow 1\%$ of dataset $\mathcal{D}$
        $\mathcal{D}, \mathcal{D}_{copy} \leftarrow \mathcal{D} - \mathcal{D}_0$
Output: Labeled data
Initialization: $\mathcal{D}_0^{label} \leftarrow \mathcal{EP}(\mathcal{D}_0),\ \ \mathcal{D}_{train} \leftarrow \mathcal{D}_0^{label}$
                $\mathcal{M} \leftarrow Train(\mathcal{D}_{train}),\ \ N_{stop} \leftarrow 0$
while $N_{stop} < 30$ do
    Head-Tail Pointer process:
    for $X^k$ in $\mathcal{D}$ do
        // Head Loop
        $\mathcal{PT}_h^k = 0;\ \ S_0^k = \mathcal{QF}(\mathcal{M}, X^k)$
        while $1$ do
            $\mathcal{PT}_h^k \leftarrow \mathcal{PT}_h^k + 1$
            $S_h^k = \mathcal{QF}(\mathcal{M}, X^k[\mathcal{PT}_h^k : len(X^k)])$
            if $S_h^k > S_0^k$ do $S_0^k \leftarrow S_h^k$
            if $S_h^k \leq S_0^k$ do $\mathcal{PT}_h^k \leftarrow \mathcal{PT}_h^k - 1$ break
        end while
        // Tail Loop
        $\mathcal{PT}_t^k = len(X^k) - 1;\ \ S_0^k = \mathcal{QF}(\mathcal{M}, X^k)$
        The similar way with Head Loop
    end for
    Querying, Deduplication and Sampling:
    $S_{final}^k = \mathcal{QF}(\mathcal{M}, X^k[\mathcal{PT}_h^k : \mathcal{PT}_t^k])$
    Sort $S_{final}^k$ in descending order and query $\mathcal{D}_s$.
    $\mathcal{D}_s^{label} \leftarrow \mathcal{EP}(\mathcal{D}_s)$
    for $X^k$ in $\mathcal{D}_0$ do
        $\mathcal{EP}'[Sample(X^k[0 : \mathcal{PT}_h^k, \mathcal{PT}_t^k : len(X^k)])]$
    $\mathcal{D}_{train} \leftarrow \mathcal{D}_s^{label}$; $\mathcal{M} \leftarrow Train(\mathcal{D}_{train})$
    $\mathcal{D} \leftarrow \mathcal{D} - \mathcal{D}_{train}$
    $N_{stop} \leftarrow N_{stop} + 1$
end while
```

number of these tokens to label them with pseudo-labels and train them with subsequences with true labels together, thus forming a semi-supervised learning framework. Pseudo-labels are obtained by the direct prediction of the model that has been trained in previous query rounds, which suggests that pseudo-labels may be inaccurate, especially at initial query rounds. Thus, except for the entity-aware subsequence, we randomly sample 30% tokens with a pseudo-label "*O*" in each sentence due to the model's high confidence in the label "*O*". When AL reaches the termination condition, i.e., no data in the data pool or the query round reaches the maximum query number, there will still be a considerable number of tokens neither queried nor pseudo-labelled. Subsequent experiments show that the above fact does not degrade model performance, suggesting that it is reasonable to query the most informative subsequences even ignoring other tokens.

Entity-Aware Subsequence Based AL

We have summarized the proposed EASAL in Algorithm 1. Symbols not mentioned before are explained here. Let $\{X^k, k = 1, .., K\}$ represents sentences in data pool $\mathcal{D}$. Then, the initial S_0^k denotes the LC score for each sentence, which will be updated in loops. S_h^k and S_t^k correspond to the current LC score in the *Head* loop and the *Tail* loop, while $\mathcal{PT}_h^k$ and $\mathcal{PT}_t^k$ correspond to the position of the head pointer and tail pointer. Moreover, N_{stop} is the number of times that there is no improvement on the model result. $\mathcal{EP}, \mathcal{EP}', \mathcal{M}, \mathcal{QS}$ represent the expert, the pseudo labels, the model, the query strategy, respectively.

Experiments

Datasets

We experiment on three biomedical NER datasets and one news NER dataset that are widely used in previous studies, including NCBI-disease (Doğan, Leaman, and Lu 2014), BC5CDR-disease (Li et al. 2016), BC5CDR-chemical (Li et al. 2016), and Conll2003 (Sang and De Meulder 2003).

Biomedical Datasets Biomedical datasets focus on two different biomedical entity types: NCBI-disease and BC5CDR-disease for disease NER and BC5CDR-chemical for chemical NER. To be noted, we merge the training set and validation set together for the data pool generation and training following previous studies (Lee et al. 2020).

News Datasets Conll2003 (Sang and De Meulder 2003) is a dataset, also in BIO format, with only 4 entity types (LOC, MISC, PER, ORG) resulting in K = 9 labels. This dataset is made from a collection of news wire articles from the Reuters Corpus (Lewis et al. 2004). The average sentence length is 12.6 tokens in its train set.

Training and Evaluation

The initial data splits used for training the model $\mathcal{M}$ are set at 1% of randomly sampled data, which are following the splitting techniques used in the existing literature on AL (Shen et al. 2017; Hazra et al. 2021)At each query round, the model selects 1500 tokens, 2250 tokens, and 2000 tokens respectively for NCBI-disease, BC5CDR (disease and chemistry), and Conll2003 that are about 1% of the total number of tokens of NCBI-disease, BC5CDR, and Conll2003.

All results in experiments are evaluated by the same F-1 score (F1) used in previous works, where the F-1 score refers specifically to the entity level rather than the token level. In order to save training time and improve model performance, we adopt incremental training, i.e., model parameters are resumed from the previous query round's parameters for each round. We set the initial training epoch as 10, which uses randomly 1% of the data, and the subsequent training epochs are all 3. The query rounds that stop training are uniformly set to 30 because when the AL method is adopted, the results within 30 query rounds can generally reach 98%∼99% of the one with training based on all data. Note that we do not set a validation set. We use the training loss at each query round to select the best model for testing.

Baselines

We have verified the effectiveness of our method through the following baselines, which highlight the importance of various components.

Random: No active learning is used and random examples are selected at each time.

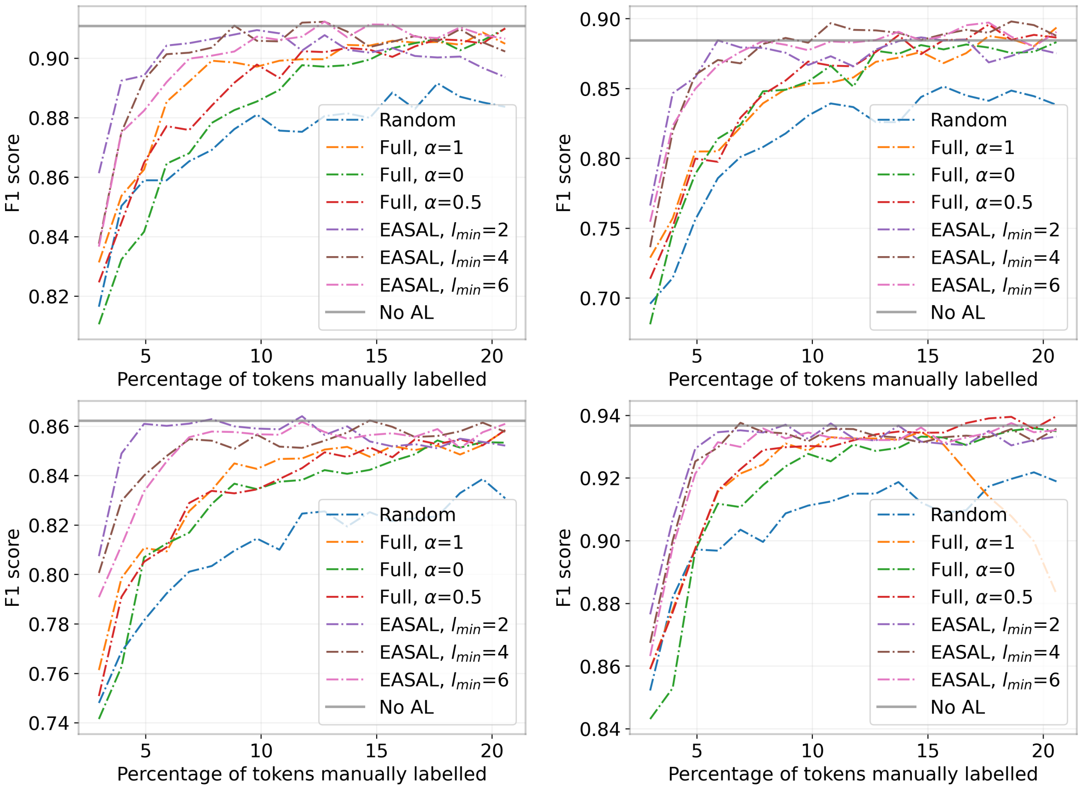


Figure 3: Main Results. The figure shows F-1 scores concerning the percentage of tokens manually labelled on Conll03, NCBI, BC5CDR-D, and BC5CDR-C from left to right and top to bottom, where blue, orange, green, and red represent baselines, namely Random, Full-sentence-based (Full). Other colours represent the EASAL (Ours). Finally, the solid grey line represents the result of using the whole training set directly.

Dataset	**F-1 Score**		**Final Frac. of Dataset**	
	100%	EASAL	Full	EASAL
Conll03	91.08	91.22	24%	10%
NCBI	88.34	89.80	16%	8%
BC5CDR-D	86.22	86.23	22%	12%
BC5CDR-C	93.67	93.60	17%	7%

Table 1: Numerical Results on four datasets, where FRAC represents the simpleness of fractions.

MNLP and its variants: No subsequence-based AL is used. In this baseline, we query the entire sentence each time with query functions described in Section *Background*, Eq. 3 (MNLP), and Eq. 4 (variants). We call this baseline a full-sequence-based AL and set different α for comparison. Since this baseline is performed on full sentences, we abbreviate it as ”Full” in the main results.

No AL: This baseline represents that instead of using any data query function, we train the BERT or BioBERT as the general fine-tuning way. The whole dataset is used for training with an epoch of 30 and a learning rate of 5e-5. We can intuitively see that comparable results can be achieved without using the whole data in this baseline.

One thing to be clear is that in EASAL, like the variant of MNLP, we utilize Eq. 4 to compute subsequence-wise LC-score while the difference is that α is fixed at 1. In the main experimental results, the variable in the EASAL method is l_{min} representing the minimum subsequence length. The setting of l_{min} is to make EASAL have practical application value because it is not reasonable to give the annotator only one token to do the annotation. l_{min} is set as 2, 4, and 6 in the main results.

Results

Main Results

The results of the baselines and the proposed EASAL are presented in Fig. 3. From left to right, from top to bottom are Conll2003, NCBI, BC5CDR-disease, and BC5CDR-chem. In Fig. 3, the solid line with grey colour is the result of using the whole train set for training, which is a powerful reference value to judge the effectiveness of AL. In order to compare different methods intuitively, Fig. 3 shows the results from the first use of Active Learning to select data for annotation. Besides, to justify that EASAL is statistically effective compared to baselines, we also experiment with various random seeds.

Several observations can be drawn from Fig. 3. First, the results of applying the AL query strategy (Full or EASAL) are better than ”Random”. In Fig. 3, the blue dashed line represents the ”Random”. It can be seen that even after many

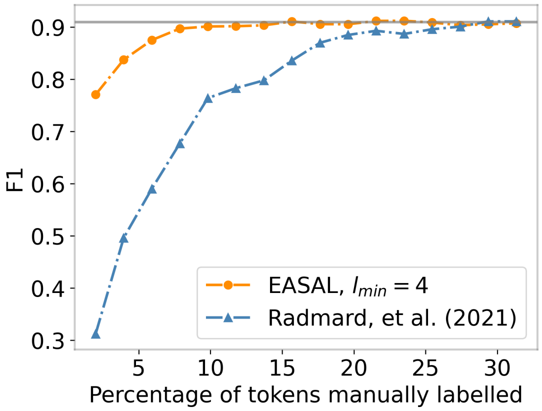


Figure 4: F-1 score Comparison with Radmard, Fathullah, and Lipani (2021).

query rounds, "Random" is still much worse than using the whole training set directly (i.e., the grey horizontal line, "No AL"). Second, the results of EASAL exceed all baselines regardless of the setting of the parameter l_{min}. EASAL reduces the roughly 20% of tokens required for full-sentence based methods to roughly 10% on all datasets. Third, for the full-sentence-based method, different α in Eq. 4 has a slight impact on AL results in most cases except for the top query rounds of Conll2003 and the last query rounds of BC5CDR-chem. Fourth, for EASAL, we fix α to 1 and change the value of l_{min}. In the initial round of the query, a smaller l_{min} value always leads to a higher F1 score because a smaller l_{min} value leads to a shorter subsequence length, which allows us to query more informative tokens. With the increase of query rounds, the F1 scores obtained by different l_{min} gradually tend to be similar. Furthermore, we also compared with the current subsequence-based method (Radmard, Fathullah, and Lipani 2021), which is shown in Fig. 4. In Fig. 4, yellow represents our proposed EASAL method with $l_{min} = 4$, and blue represents the contrasted method. It can be seen that our EASAL can faster reach the results obtained by training with all the data. Because Radmard, Fathullah, and Lipani (2021) is based on CNN-CNN-BiLSTM, and our EASAL is based on BERT, there is a big gap between the two in the early rounds of active learning.

Numerical Results

We also show numerical results in Table 1 for a more intuitive comparison of the subsequence-based (EASAL) and full-sentence-based methods on four datasets, i.e., Conll03, NCBI, BC5CDR-D, and BC5CDR-C. In Table 1, F-1 scores of using the whole train set without the AL method and the proposed EASAL are reported in the first two columns. The last two columns report the percentage of data used when the AL method reaches about 98% of the results of using the whole train set without AL. Note that "Full" means the conventional full-sentence-based method in view of query function Eq. 3 or query function Eq. 4 with α=1 while "EASAL" means the entity-aware subsequence-based AL in view of the same query function Eq. 4 with α=1 and l_{min}=4. It can be seen that "EASAL" can approximate the results trained with the whole train set much faster than "Full".

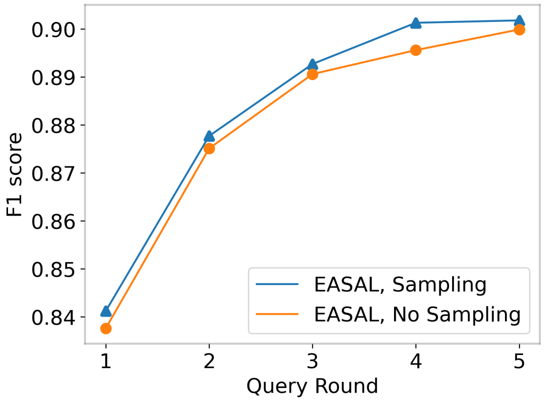


Figure 5: Ablation Study on Sampling.

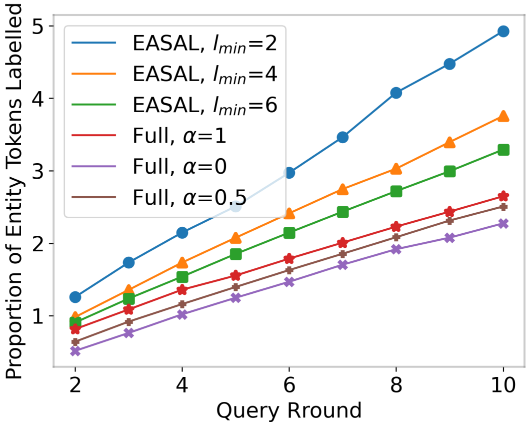


Figure 6: Entity Recall on Full-sentence-based method (Full), and EASAL.

Analysis on Sampling

Each query round in EASAL has an extra sampling operation, which randomly samples 30% of the tokens except the selected entity-aware subsequences to be pseudo-labelled to participate in the training together. In this section, we explore the benefits of performing sampling over no sampling operation with EASAL, $l_{min} = 4$ on Conll2003. Results are shown in Fig. 6. In each query round, sampling is always better than no sampling. Although the improvement is not large, it is still cost-effective to apply the sampling operation because it does not consume any natural resources.

Analysis on Entity Recall

This section aims to understand some of the underlying mechanisms that allow the subsequence querying methods to achieve results substantially better than a full-sentence baseline. Namely, the ability of the different methods to extract the tokens for which the model is the most uncertain. Given that most tokens in both datasets have the same label - "O", signifying no entity - it is likely that tokens belonging to entities, particularly rarer classes, trigger higher model uncertainty. Querying full sentences at a time, the AL algorithm

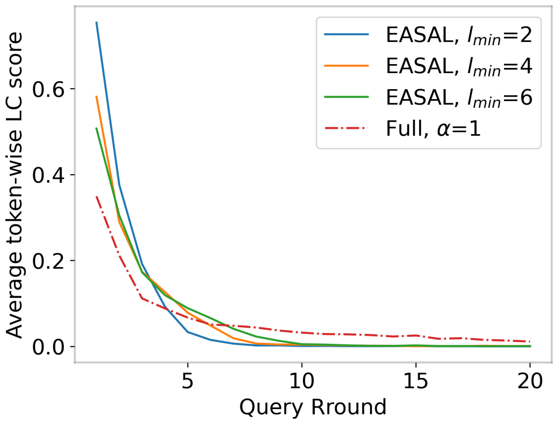


Figure 7: LC score analysis on Full-sentence-based method (Full), and EASAL.

will spend much of its token budget for that round labelling non-entity tokens while attempting to locate the more informative entities. Subsequence querying methods, not faced with this wasteful behaviour, allow the AL algorithm to query entity tokens quicker, locating and labelling the majority of entity tokens faster throughout training.

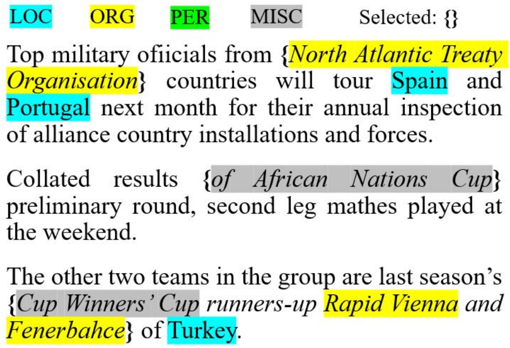
To verify the above statement, we perform the entity recall experiment on Conll2003, which is shown in Fig. 5 with the query round as the horizontal axis and the proportion of entity tokens labelled as the vertical axis. Blue and Orange curves specify full-sentence-based AL (Full) and subsequence-based AL (EASAL). In each query round, the percentage of tokens with entity labelled queried by "EASAL" is always higher than that of "Full", which shows that EASAL can contain more informative tokens and focus on possible entities.

Analysis on Uncertainty Score

This section compares tokens' uncertainty scores (LC scores) in the queried set for each querying method. Fig. 7 shows how the mean of token-wise scores evolve for different querying methods for the Conll2003 dataset until convergence. As seen from the figure, our subsequences-based approach (EASAL) 's average token-wise LC score is high in the initial rounds at the beginning of active learning. However, after several active learning query rounds, the average token-wise LC score of EASAL quickly drops below that of the full-sentence-based query method. In conclusion, Fig. 7 clearly shows that subsequence querying methods converge faster over the full course of the algorithm compared to full sentence querying. This is consistent with Fig. 3 in terms of initial rate and final time of model performance convergence, namely that model performance plateaus alongside the uncertainty score.

Case Study

Fig. 8 shows the selected subsequences of three sentences in Conll2003 in the first query round. Four colours are used to represent different entity classes, while braces ({}) and italics are used to represent the subsequence selected in the sentence. We can see that the selected subsequences in the three instances all contain entities, for example, the MISC entity "Cup Winners' Cup", the ORG entity "Rapid Vienna" and the ORG entity "Fenerbahce" in the last row sample, which intuitively shows that EASAL can preferentially query the subsequences containing entities, maximizing information utilization.



LOC ORG PER MISC Selected: {}

Top military ofiicials from {*North Atlantic Treaty Organisation*} countries will tour Spain and Portugal next month for their annual inspection of alliance country installations and forces.

Collated results {*of African Nations Cup*} preliminary round, second leg mathes played at the weekend.

The other two teams in the group are last season's {*Cup Winners' Cup runners-up Rapid Vienna and Fenerbahce*} of Turkey.

Figure 8: Case Study on Conll2003. Four colours are used to represent different entity classes, while braces ({}) and italics are used to represent the subsequence selected in the sentence.

Related Work

Active Learning (AL) is an important technology to reduce the cost of annotations. Shen et al. (2017) demonstrated that the amount of labeled training data could be drastically reduced when deep learning is combined with active learning based on CNN-CNN-LSTM for NER. Most deep active learning studies for NER focus on three aspects. The first one is exploring the effect of BERT (Kenton and Toutanova 2019) or variations of BERT (e.g., BioBERT, BERT-CRF) combined with active learning (Zhang and Zhang 2019; Shelmanov et al. 2019; Liu et al. 2022; Dor et al. 2020). The second one is exploring the data augmentation method combined with active learning (Zhang, Yu, and Zhang 2020; Quteineh, Samothrakis, and Sutcliffe 2020), where most of these studies make the data augmentation at each query round. The third one aims to solve the data redundancy problem in the AL query. Hazra et al. (2021) proposed Active^2 Learning (A^2L) actively adapts to the deep learning model being trained to eliminate such redundant examples chosen by the query strategy. These methods treat the sentence as a query object, without considering subsequences. Radmard, Fathullah, and Lipani (2021) first proposed subsequences-based deep AL method based on CNN-CNN-BiLSTM. However, Radmard, Fathullah, and Lipani (2021) uses an exhaustive search approach to generate subsequences. Therefore, we propose Entity-Aware Subsequence-based AL with an effective *Head-Tail* pointer on BERT to fill the gap.

Conclusion

In this paper, we proposed an Entity-Aware Subsequence-based Active Learning named EASAL. EASAL utilizes *Head-Tail* pointer to query one entity-aware subsequence for each sentence. For other tokens, we randomly sample 30% to label them with pseudo-labels for training together. Experiments on one news dataset and three biomedical datasets demonstrate the effectiveness of EASAL.

Acknowledgments

This work is supported by Chinese Key-Area Research and Development Program of Guangdong Province (2020B0101350001), the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen and the Shenzhen Fundamental Research Fundunder Grant JCYJ20220818103001002.

References

Cho, H.; and Lee, H. 2019. Biomedical named entity recognition using deep neural networks with contextual information. *BMC bioinformatics*, 20(1): 1–11.

Doğan, R. I.; Leaman, R.; and Lu, Z. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47: 1–10.

Dor, L. E.; Halfon, A.; Gera, A.; Shnarch, E.; Dankin, L.; Choshen, L.; Danilevsky, M.; Aharonov, R.; Katz, Y.; and Slonim, N. 2020. Active learning for BERT: An empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7949–7962.

Felder, R. M.; and Brent, R. 2009. Active learning: An introduction. *ASQ higher education brief*, 2(4): 1–5.

Hakala, K.; and Pyysalo, S. 2019. Biomedical named entity recognition with multilingual BERT. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, 56–61.

Hazra, R.; Dutta, P.; Gupta, S.; Qaathir, M. A.; and Dukkipati, A. 2021. Active^2 Learning: Actively reducing redundancies in Active Learning methods for Sequence Tagging and Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1982–1995. Online: Association for Computational Linguistics.

Hu, J.; Shen, Y.; Liu, Y.; Wan, X.; and Chang, T.-H. 2022a. Hero-Gang Neural Model For Named Entity Recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1924–1936. Seattle, United States: Association for Computational Linguistics.

Hu, J.; Zhao, H.; Guo, D.; Wan, X.; and Chang, T.-H. 2022b. A Label-Aware Autoregressive Framework for Cross-Domain NER. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 2222–2232. Seattle, United States: Association for Computational Linguistics.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Kocaman, V.; and Talby, D. 2021. Biomedical named entity recognition at scale. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I*, 635–646. Springer.

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4): 1234–1240.

Lewis, D. D.; Yang, Y.; Russell-Rose, T.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr): 361–397.

Li, J.; Sun, Y.; Johnson, R. J.; Sciaky, D.; Wei, C.-H.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Wiegers, T. C.; and Lu, Z. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Liu, M.; Tu, Z.; Zhang, T.; Su, T.; Xu, X.; and Wang, Z. 2022. LTP: a new active learning strategy for CRF-based named entity recognition. *Neural Processing Letters*, 54(3): 2433–2454.

Naseem, U.; Khushi, M.; Khan, S. K.; Shaukat, K.; and Moni, M. A. 2021. A comparative analysis of active learning for biomedical text mining. *Applied System Innovation*, 4(1): 23.

Quteineh, H.; Samothrakis, S.; and Sutcliffe, R. 2020. Textual Data Augmentation for Efficient Active Learning on Tiny Datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7400–7410.

Radmard, P.; Fathullah, Y.; and Lipani, A. 2021. Subsequence Based Deep Active Learning for Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4310–4321. Online: Association for Computational Linguistics.

Sang, E. F.; and De Meulder, F. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Shelmanov, A.; Liventsev, V.; Kireev, D.; Khromov, N.; Panchenko, A.; Fedulova, I.; and Dylov, D. V. 2019. Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 482–489. IEEE.

Shen, Y.; Yun, H.; Lipton, Z. C.; Kronrod, Y.; and Anandkumar, A. 2017. Deep Active Learning for Named Entity Recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 252–256.

Siddhant, A.; and Lipton, Z. C. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2904–2909. Brussels, Belgium: Association for Computational Linguistics.

Wang, X.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; and Tu, K. 2021. Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1800–1812.

Yoon, W.; So, C. H.; Lee, J.; and Kang, J. 2019. Collabonet: collaboration of deep neural networks for biomedical named entity recognition. *BMC bioinformatics*, 20(10): 55–65.

Zhang, L.; and Zhang, L. 2019. An ensemble deep active learning method for intent classification. In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, 107–111.

Zhang, R.; Yu, Y.; and Zhang, C. 2020. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8566–8579.