Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.045807

ARTICLE

Network Configuration Entity Extraction Method Based on Transformer with Multi-Head Attention Mechanism

Yang Yang[1], Zhenying Qu[1], Zefan Yan[1], Zhipeng Gao[1,*] and Ti Wang[2]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2]Product Development Department, China Unicom Smart City Research Institute, Beijing, 100044, China

*Corresponding Author: Zhipeng Gao. Email: gaozhipeng@bupt.edu.cn

Received: 08 September 2023 Accepted: 01 November 2023 Published: 30 January 2024

ABSTRACT

Nowadays, ensuring the quality of network services has become increasingly vital. Experts are turning to knowledge graph technology, with a significant emphasis on entity extraction in the identification of device configurations. This research paper presents a novel entity extraction method that leverages a combination of active learning and attention mechanisms. Initially, an improved active learning approach is employed to select the most valuable unlabeled samples, which are subsequently submitted for expert labeling. This approach successfully addresses the problems of isolated points and sample redundancy within the network configuration sample set. Then the labeled samples are utilized to train the model for network configuration entity extraction. Furthermore, the multi-head self-attention of the transformer model is enhanced by introducing the Adaptive Weighting method based on the Laplace mixture distribution. This enhancement enables the transformer model to dynamically adapt its focus to words in various positions, displaying exceptional adaptability to abnormal data and further elevating the accuracy of the proposed model. Through comparisons with Random Sampling (RANDOM), Maximum Normalized Log-Probability (MNLP), Least Confidence (LC), Token Entrop (TE), and Entropy Query by Bagging (EQB), the proposed method, Entropy Query by Bagging and Maximum Influence Active Learning (EQBMIAL), achieves comparable performance with only 40% of the samples on both datasets, while other algorithms require 50% of the samples. Furthermore, the entity extraction algorithm with the Adaptive Weighted Multi-head Attention mechanism (AW-MHA) is compared with BILSTM-CRF, Mutil_Attention-Bilstm-Crf, Deep_Neural_Model_NER and BERT_Transformer, achieving precision rates of 75.98% and 98.32% on the two datasets, respectively. Statistical tests demonstrate the statistical significance and effectiveness of the proposed algorithms in this paper.

KEYWORDS

Entity extraction; network configuration; knowledge graph; active learning; transformer

1 Introduction

As information and communication technology advances, the Internet's role in society has become crucial, and closely tied to economies and social development. The sudden outbreak of the epidemic has further emphasized the significance of the Internet, elevating the importance of network infrastructure and services. The increasing variety of devices and access methods has led to greater complexity, requiring efficient network management, resource allocation, and diverse services for various businesses.

A knowledge graph [1] is a structured knowledge base that employs graphs to represent entities, including concepts, individuals, and objects, along with their interconnected relationships in the real world. Knowledge graphs are utilized to transform various data forms into a graphical knowledge representation, where entities and relationships are depicted as nodes and edges. Once entities, such as devices, services, parameters, etc., are extracted from network configuration data, they are used to construct a graphical knowledge graph. The approach involves tracing the network's logical structure, mapping association relationships among network devices, and understanding the agreements within each device. Then, it analyzes business function relationships, updates the knowledge graph database in real-time based on network status changes, and supports keyword searches. Finally, knowledge interactions are identified, and data from various equipment configurations are constructed into a contactable, traceable, and extensible map. This aids in comprehending the structure and semantics of network configurations, as well as in standardizing and adding semantic meaning to network configuration data, ultimately enhancing network maintainability and scalability. The network configuration knowledge graph, derived after extracting network configuration entities, can provide support for various domains such as network configuration, fault diagnosis [2], performance analysis [3], security detection [4], and more.

Entity extraction constitutes a pivotal component in the creation of knowledge graphs. Its primary objective is to identify and extract entities from different data origins and structures, then converting them into structured data suitable for storage within a knowledge graph. Presently, entity extraction technologies fall into three fundamental categories. The first category relies on manual modification rules, frequently constructed using dictionaries and knowledge bases. However, this method necessitates lots of human effort to develop language models, leading to prolonged system cycles, sluggish information updates, and limited portability. The second category is based on statistical principles. Nevertheless, these approaches suffer from several drawbacks, including the need for many artificial features, expensive costs, and limited migration and generalization capabilities due to considerable human intervention. The third category is based on deep learning techniques. These approaches incorporate neural network models combined with attention mechanisms, which consider different features and influence levels. They yield diverse entity extraction results while minimizing attention to redundant information.

This paper focuses on the following key issues in existing entity extraction methods: (1) Both supervised and semi-supervised learning methods require a significant amount of labeled data, and the quality of these labeled samples directly impacts the performance of the classification model. However, data acquisition can be time-consuming and labor-intensive, often resulting in a substantial number of redundant samples in the training set. (2) Current neural network models that utilize the attention mechanism face a challenge when increasing the number of attention heads. This can lead to an expression bottleneck, diminishing the model's capability to effectively express context vectors and potentially resulting in decreased accuracy.

Our contributions are as follows:

- This paper presents an entity extraction method on the basis of active learning and a transformer with an attention mechanism. The approach involves utilizing an enhanced active learning method to label the training set, which is then utilized to train the deep learning model.

- An enhanced active learning method is presented to improve the quality of sample selection. Initially, the active learning technique is employed to select the most informative unlabeled samples using a specific strategy algorithm. These queried samples are then utilized to train the classification model, aiming to enhance its accuracy. Unlike existing query strategies, this paper introduces novel improvements that effectively address issues of outliers and redundancy within the training set examples.
- This paper proposes an adaptive weighting method on the basis of the Laplace mixture distribution idea, aiming at overcoming the bottleneck problem in the expression of the multi-head attention mechanism. The Transformer model is employed to capture word semantics and context, as well as to handle long-distance dependencies through the multi-head self-attention mechanism. By combining the idea of the Laplace mixture distribution, the weight matrix is mixed and superimposed to heighten the expression ability of the distribution of attention, thus improving the performance of the multi-head self-attention mechanism of the Transformer.
- This paper conducted simulation experiments on two datasets, demonstrating the effectiveness of the proposed method. Furthermore, when compared to other algorithms, the presented framework exhibits better performance.

The following sections are organized as follows. Section 2 introduces existing entity extraction and active learning methods. Section 3 elaborates on the specific algorithmic improvements. In Section 4, the simulation results of the algorithm are given and compared with other methods. Finally, Section 5 provides the conclusion, which summarizes the research results.

2 Related Work

Expert annotation for network configuration data is often expensive and time-consuming. Active learning offers a solution to reduce labeling expenses and improve data utilization. Additionally, Transformers are renowned for their powerful self-attention mechanism, allowing them to capture long-range dependencies within input sequences. This is highly beneficial for handling complex relationships among entities in network configuration data. Therefore, this article is designed and improved based on Transformers and active learning. During the composition of this section, several literature searches were conducted on Web of Science, Engineering Village, and Google Scholar, employing keywords like “entity extraction”, “active learning”, and “Transformer”. These works encompassed the primary research achievements in this domain. Table 1 provides a succinct comparison of related studies. This section will provide a detailed discussion of both entity extraction and active learning.

Table 1: A comparative analysis of the literature

	Reference	Year	Models	Statistical-based methods	Deep learning-based methods	Active learning
1	Bikel et al. [5]	1999	Hidden Markov	Yes	No	No
2	Borthwick et al. [6]	1999	MEM	Yes	No	No
3	Chieu et al. [7]	2002	MEM	Yes	No	No

(Continued)

Table 1 (continued)

	Reference	Year	Models	Statistical-based methods	Deep learning-based methods	Active learning
4	Mayfield et al. [8]	2003	SVM	Yes	No	No
5	McCallum et al. [9]	2003	CRF	Yes	No	No
6	Settles [10]	2004	CRF	Yes	No	No
7	Lample et al. [11]	2016	LSTM	No	Yes	No
8	Luo et al. [12]	2018	BILSTM	No	Yes	No
9	Xu et al. [13]	2019	BILSTM	No	Yes	No
10	Singh et al. [14]	2022	CNN	No	Yes	No
11	Parsaeimehr et al. [15]	2023	CNN	No	Yes	No
12	He et al. [16]	2023	Transformer	No	Yes	No
13	Alissa et al. [17]	2023	Transformer	No	Yes	No
14	Jia et al. [18]	2020	Transformer	No	Yes	No
15	Beygelzimer et al. [19]	2009	Importance weighted	No	Yes	Yes
16	Mahapatra et al. [20]	2018	GAN	No	Yes	Yes
17	**Our paper**	**2023**	**Transformer**	**No**	**Yes**	**Yes**

2.1 Entity Extraction

As a highly effective method for managing text data, entity extraction has long been a focus of research in the areas of artificial intelligence and computer science. Its main goal is to automatically recognize particular entities from non-formatted textual data and classify them into predetermined groups, such as names of people, places, and organizations. The current technical methods for entity extraction may be loosely categorized into three groups: methods on the basis of manual customization rules, methods on the basis of traditional machine learning, and deep learning-based approaches.

In the early stages of entity extraction research, the reliance was primarily on hand-crafted rules. Experts and academics from numerous domains built knowledge extraction frameworks manually using their subject knowledge. These frameworks included a variety of strategies, including headwords, demonstrative words, directional words, positional words, punctuation analysis, and keyword detection. Pattern and string matching served as the main strategy, delivering admirable accuracy. However, this approach required extensive human labor to develop language models, because it mainly relied on the development of knowledge repositories and lexicons. As a result, it caused protracted system cycles, slowly updated information, and restricted portability of the extraction system.

With the advent of machine learning, statistical-based methods have been introduced for entity extraction. These methods leverage statistical machine learning to learn knowledge from a vast amount of labeled corpora, eliminating the need for manually defined rules. They treat Named Entity Recognition (NER) and word segmentation problems as sequence labeling tasks, where the predicted label depends not only on the current predicted sequence label but also on the preceding predicted label, displaying a strong interdependence between labels. Prominent statistical-based methods include the hidden Markov model (HMM) [5], maximum entropy model (MEM) [6,7], support vector machine

(SVM) [8], and conditional random field (CRF) [9,10]. While these methods have yielded favorable outcomes in entity extraction, they present particular challenges additionally. The hidden Markov model relies solely on each state and its corresponding observation without considering the length of the observation sequence or word context. The maximum entropy model's constraint function relationship is tied to the number of samples, leading to extensive calculations in the iterative process, and making practical application more challenging. Support vector machines rely on quadratic programming to solve support vectors, involving calculations of m-order matrices that can be difficult to implement for large-scale training samples. Furthermore, the conditional random field model exhibits slow convergence speed, leading to elevated training costs and increased complexity.

In recent years, deep learning methods like Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) have gained widespread adoption in the area of natural language processing, demonstrating impressive results across various tasks. Techniques like LSTM-CRF [11], BILSTM-CRF [12,13], BERT-CNN [14], and CNN-CRF [15], among others, have been successfully applied. In comparison to earlier statistical machine learning methods, deep learning approaches offer distinct advantages in automatic feature learning, leveraging deep semantic knowledge, and addressing data sparsity issues. These methods utilize neural networks to automatically learn features and train sequence labeling models, surpassing traditional methods on the basis of handcrafted features, thus positioning them as current research hotspots.

However, RNNs demand sequential processing, rendering both training and inference time-consuming. Conversely, CNNs, originally tailored for visual tasks, excel at extracting local information due to their inherent bias but struggle to capture global context, resulting in suboptimal performance in entity recognition tasks. In recent years, Transformer models, powered by attention mechanisms, have achieved state-of-the-art results in natural language processing and computer vision. Transformers have effectively addressed the constraints of sequential processing in RNNs and the dependency on local features in CNNs, enabling them to adeptly capture global context. In [18], authors leveraged the Transformer for entity extraction and constructed a pre-trained model on the ClueNER dataset, achieving remarkable performance. Nonetheless, the pre-trained models established in [18] are grounded in standard Chinese corpora and exhibit inferior performance when applied to entity extraction tasks within specific network contexts. Furthermore, many researchers have delved into the utilization of Transformers in network-related scenarios. For instance, reference [16] employed the Transformer architecture for anomaly detection, while reference [17] leveraged Transformers for text simplification. Nevertheless, within the domain of entity extraction in network contexts, the exploration of the Transformer architecture has been relatively limited.

2.2 Active Learning in Model Training

Deep learning-based methods heavily rely on annotated corpora, and there is a shortage of large-scale, general-purpose corpora available for constructing and evaluating entity extraction systems. This limitation has emerged as a significant obstacle to the widespread adoption of these methods.

Both supervised learning and semi-supervised learning require a certain amount of labeled data, and the effectiveness of the classification model depends on the quality of these labeled samples. However, acquiring training samples is not only time-consuming and labor-intensive but also leads to a considerable number of redundant samples within the training set. To address these challenges and reduce training set size and labeling costs, active learning methods [19,20] have been proposed to optimize classification models. Active learning employs specialized algorithms to select the most informative unlabeled samples, which are then annotated by experts. These selected samples are subsequently integrated into the training process of the classification model, enhancing its accuracy. The key to successful active learning lies in selecting an appropriate query strategy. Two commonly used types of active learning models are stream-based active learning and pool-based active learning. Different situations may require various implementation solutions, and the choice of query strategy can be based on a single machine learning model or multiple models. Currently, widely adopted query strategies include uncertainty sampling, committee-based sampling, model change expectations, query error reduction, variance reduction, and density weighting, among others.

Compared to traditional supervised methods, active learning demonstrates improved capabilities in handling larger training datasets, distinguishing diverse sample points, and reducing the volume of training data and manual labeling costs. However, traditional active learning may prove insufficient when dealing with challenges such as multiclass classification, outliers, sample redundancy within the training set, and imbalanced data. This paper introduces a pool-based active learning method that incorporates innovative enhancements designed to effectively address the challenges posed by outliers and redundancy in the training set examples.

3 Entity Extraction Method

This section describes the network configuration entity recognition method. Section 3.1 provides an overview of the overall model flow and structure. Section 3.2 elaborates on the improved active learning algorithm, EQBMIAL. Section 3.3 provides a detailed explanation of the specific structure of the entity extraction stage and the AW-MHA designed to enhance the transformer.

3.1 Overall Flow and Framework

In this section, an enhanced entity extraction method is proposed, integrating active learning and an attention mechanism [21,22]. Leveraging entropy query-by-bagging (EQB) and maximum influence (MI) active learning strategies, samples with high uncertainty and low redundancy are selected and manually labeled, expanding the labeled sample set. Through iterative expansion and model training, the method improves the model's generalization ability. In addition, an improved adaptive weighting mechanism is introduced into the multi-head self-attention mechanism of the transformer model. This allows the model to achieve more flexible weight allocation by appropriately setting the mean and variance parameters. Moreover, this improvement can weigh information from different modes to better capture multimodal information. This fusion of plural attention heads leads to an effective amalgamation of information, which may also help the model deal with noise and uncertainty in the data better, thus improving the robustness and generalization performance of the whole model. The overall flow chart of the algorithm is shown in Fig. 1.

The entity extraction process is as follows:

1. In the sample extraction stage, the selected sample set is selected from the unlabelled sample pool utilizing the entropy bagging query and the active learning screening strategy on the basis of the maximum influence. These selected samples are included in the labeled dataset after expert labeling, while the remaining unselected samples expand the unlabeled pool.
2. During the entity extraction stage, the labeled dataset serves as the training set for training the entity extraction model, which comprises the input layer, hidden layers, including embedding, and an improved Transformer with adaptive weighting for multi-head self-attention mechanisms, as well as sequence labeling. Then, the output layer is composed.

3. The performance of the current model relies on its output. If the model meets the performance requirements, the model and labeled samples are obtained, and the whole process is ended. Otherwise, steps 1 and 2 are repeated until the performance requirements are satisfied.

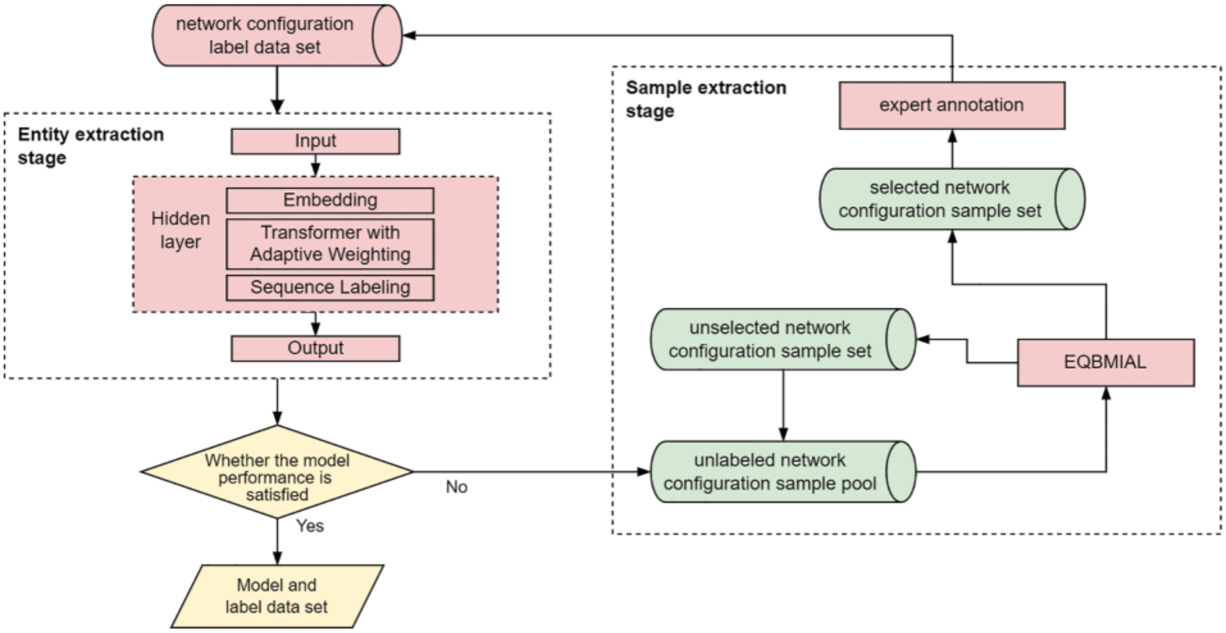
network configuration label data set
Entity extraction stage
Input
Hidden layer
Embedding
Transformer with Adaptive Weighting
Sequence Labeling
Output
Whether the model performance is satisfied
No
Yes
Model and label data set
Sample extraction stage
expert annotation
selected network configuration sample set
unselected network configuration sample set
EQBMIAL
unlabeled network configuration sample pool

Figure 1: Entity extraction flow chart

3.2 Sample Extraction Method on the Basis of Improved Active Learning

Active learning involves selecting a subset of unlabelled samples for manual labeling and iteratively expanding the labeled dataset to enhance the model's generalization capability. In the active learning process, the key challenge is selecting unlabelled samples for labeling and efficiently improving the model's generalization after learning from these selected samples. The research at present is primarily on the basis of the pool-based sample selection strategies, aiming to develop sample selection strategies suitable for named entity recognition tasks while ensuring the model achieves a certain level of performance and minimizes labeling costs. One such active learning approach is the Entropy Query Bagging (EQB) algorithm, which utilizes information entropy to measure the uncertainty and information gain of samples. Therefore, it tends to select samples that provide maximum information during model training. In the context of network configuration entity extraction tasks, EQB significantly reduces the manual data labeling workload, thereby improving annotation efficiency. Furthermore, EQB can expedite model convergence by focusing more attention on the critical aspects of entity extraction tasks.

This research focuses on maintaining a future annotation of the sample pool and training the model after selecting the sample label through the active learning sample selection strategy to rapidly improve the model's generalization ability. Therefore, the active learning approach proposed in this article, EQBMIAL, builds upon EQB and enhances it with the maximization of influence to optimize the algorithm. The query strategy in EQB primarily relies on measuring sample categories based on entropy values, which provides valuable information. However, this method does not solve two problems. The first problem concerns isolated points. A sample with a large entropy value has only one sample in the dataset. Choosing this sample does not boost the model's accuracy or other indicators. A sample with a low entropy value is not an isolated point. Adding this sample may boost the model's accuracy and other indicators. The second problem is related to sample redundancy in the training set. If a sample with a large entropy value is already in the training set, adding this sample to the model does not improve accuracy. If a sample with a small entropy value is not in the training set, adding this sample may greatly improve the model. Therefore, to address these two problems, this paper puts forward to optimize the algorithm by combining entropy bagging and maximum influence. The maximum influence is used to identify the most influential samples, while entropy bagging encompasses entropy-based queries and maximum influence. The force index comprises two components: a representative model and a different model.

The first index is EQB, which is a query committee method. EQB begins by selecting k training subsets from the initial training set using bagging. These k subsets are then used to train k classification models, forming a committee. Each classifier in the committee predicts the category for each sample in the unlabelled sample set, resulting in each sample having k labels based on the predicted categories. EQB formally uses these tags to calculate the sample's entropy value, and the query's formula is as follows:

$$f\left(x_i\right) = \sum_{w=1}^{N_i} p\left(y_i^* = w \middle| x_i\right) \log[p(\left(y_i^* = w \middle| x_i\right)] \tag{1}$$

where $\mathrm{p}(\left(\mathrm{y}_i^* = \mathrm{w} \middle| \mathrm{x}_i\right)$ denotes that the sample x_i's probability is predicted to belong to class w by k training models. In other words, the predicted label of the sample x_i is determined on the basis of the number of votes for class w divided by k, where N_i represents the total number of categories.

When all classifiers in the committee make identical predictions for a sample, f(x) equals 0. This suggests that the category of this sample is highly certain for the current classification model, and adding it to the training set would offer little improvement to the model. On the other hand, when the predictions of the sample's label by the classifiers in the committee vary, f(x) increases, indicating that this sample provides a substantial amount of information that can be beneficial in enhancing the model.

The second part is the maximization of influence, which consists of two components: the representative model and the difference model.

The representative model is used to solve the outlier problem of the sample. It primarily relies on k-means clustering and category prior probability. To find a representative sample from the labeled sample set, k-means is utilized to cluster the data. Initially, k samples are selected as the initial cluster centers. The distance from each sample in the dataset to each of the k cluster centers is calculated, and each sample is assigned to the cluster corresponding to the center with the least distance. The formula used to determine each cluster's center is as follows:

$$a_j = \frac{1}{|c_i|} \sum_{x \in c_i} x \tag{2}$$

The calculation is repeated until the termination condition is reached so that k representative samples are obtained. Then the category prior probability b_k formula in the unlabelled sample set is as follows:

$$p\left(k|x_i\right) = \frac{b_k \exp\left(-\frac{1}{2\sigma^2} ||x_i - a_k||^2\right)}{\sum_{k=1}^{K} b_k \exp\left(-\frac{1}{2\sigma^2} ||x_i - a_k||^2\right)} \tag{3}$$

$$b_k = \frac{1}{n}\sum\nolimits_{i=1}^{n} p\left(k|x_i\right) \tag{4}$$

According to Eq. (4), the representative model formula is as follows:

$$w\left(x_i\right) = \sum\nolimits_{k=1}^{K} b_k \exp\left(-\frac{1}{2\sigma^2}\,||x_i - a_k||^2\right) \tag{5}$$

The larger the value $\mathrm{w}\left(\mathrm{x_i}\right)$, the more representative the unlabelled sample with $\mathrm{x_i}$ value.

The difference model is used to settle the sample redundancy problem in the training set. The node similarity algorithm is mainly used. The node similarity algorithm uses Jaccard similarity. The similarity formula is as follows:

$$\mathrm{Sim}\left(\mathrm{X,Y}\right) = \frac{\left|\Gamma\left(\mathrm{X}\right) \cap \Gamma\left(\mathrm{Y}\right)\right|}{\left|\Gamma\left(\mathrm{X}\right) \cup \Gamma\left(\mathrm{Y}\right)\right|} \tag{6}$$

The Jaccard distance formula is as follows:

$$\mathrm{d_J}\left(\mathrm{X,Y}\right) = 1 - \mathrm{Sim}\left(\mathrm{X,Y}\right) \tag{7}$$

For an unlabelled sample $\mathrm{x_i}$, the minimum Euclidean distance $\mathrm{d(x_i)}$ between it and all currently labeled samples are used to measure the difference between the samples:

$$\mathrm{d}\left(\mathrm{x_i}\right) = \min_{\mathrm{x_i \in U, x_j \in L}} \mathrm{d_J}\left(\mathrm{x_i, x_j}\right) \tag{8}$$

The larger $\mathrm{d}\left(\mathrm{x_i}\right)$ is, the farther the currently unlabelled sample is from the currently labeled sample and the greater the difference in the sample.

The maximum influence model's formula is as follows:

$$\mathrm{z}\left(\mathrm{x_i}\right) = \mathrm{a} * \mathrm{w}\left(\mathrm{x_i}\right) + \mathrm{b} * \mathrm{d}\left(\mathrm{x_i}\right) \tag{9}$$

where $\mathrm{a} \in [0, 1]$, $\mathrm{b} \in [0, 1]$, $\mathrm{a} + \mathrm{b} = 1$, and a and b are the weights.

Finally, EQBMIAL is composed of the maximization of influence and EQB. The maximization of influence is composed of representative models and different models. The final EQBMIAL model is $\mathrm{g(x_i)}$, and the formula is as follows:

$$\mathrm{g}\left(\mathrm{x_i}\right) = \mathrm{c} * \mathrm{z}\left(\mathrm{x_i}\right) + \mathrm{d} * \mathrm{H^{BAG}}\left(\mathrm{x_i}\right) \tag{10}$$

where $\mathrm{c} \in [0, 1]$, $\mathrm{d} \in [0, 1]$, $\mathrm{c} + \mathrm{d} = 1$, and c and d are the weights.

The EQB flow is as follows:

Algorithm 1: EQBMIAL

Input: entity extraction model M, unlabelled dataset U, labeled dataset L, algorithm iteration step K, the dataset to be labeled Q, maximum iteration number T, current iteration number t.

Output: selected sample NK

```
Step:
1. Initialize the entity extraction model M
2. While t < T
3.         calculate EQB$f\left(x_i\right)$
4.         calculate MI z$\left(x_i\right)$
```

(Continued)

Algorithm 1 (continued)

```
        calculate EQBMIAL $g(x_i) = a * z(x_i) + b * f(x_i)$
        obtain the sample set to be labeled with the number of samples K from the candidate set Q
        = R($g(x_i)$,K)
        L = L + Q;
        U = U − Q;
        M = Train (M, L)
end While
return NK, M
```

3.3 Transformer with Improved Multi-Head Self-Attention Based on Adaptive Weighting Idea

This section is divided into four subparts. In Section 3.3.1, the structure of the entity extraction model and the functions of each layer are introduced. Section 3.3.2 explains the operation of the existing multi-head attention mechanism. Section 3.3.3 proposes the Adaptive Weighting Mechanism to enhance the multi-head attention mechanism further. Section 3.3.4 provides a detailed description of how the Adaptive Weighting Mechanism is utilized to enhance the multi-head attention of the transformer.

3.3.1 Entity Extraction Model

The essence of entity extraction lies in constructing a model. This article adopts a deep learning approach to identify entities. The Fig. 2 below visually presents the entity extraction method devised within this study.

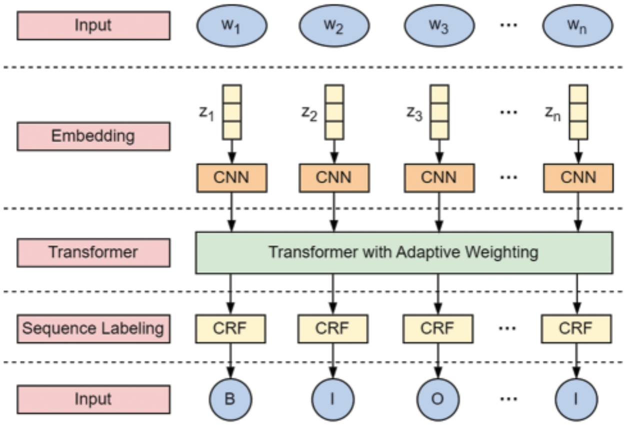

Figure 2: The model structure diagram of entity extraction

Input layer: This layer serves as the entry point and is responsible for receiving incoming text sequences $O = [w_1, w_2, \ldots, w_n]$ as the input.

Embedding layer: This layer converts textual text into a format recognizable and computable by the model, providing computer models with enhanced text representation and processing capabilities. It does this by representing each word with a low-dimensional vector, where words with similar meanings occupy nearby positions in the vector space. In this paper, CNN is used to process the embedding layer, and the local receptive field and feature extraction capabilities of CNN are utilized to transform text data into meaningful embedding representations, providing richer features for subsequent tasks. After entering the text sequence O, each word is converted into a pre-trained word vector, which forms a matrix $E = [z_1, z_2, \ldots, z_n]$, with each row representing a word's embedding. Next, the embedded matrix E serves as the input for CNN, where convolution operations with different filter sizes extract local features from various positions. These features extracted from each convolution kernel are then pooled to generate the feature matrix $K = [x_1, x_2, \ldots, x_n]$.

Transformer layer: To create a thorough feature representation, the transformer layer is used to capture global context information in the input text sequence. The embedding layer provides the input for this layer $K = [x_1, x_2, \ldots, x_n]$. The multi-layer Transformer encoder uses a multi-head self-attention mechanism and a feedforward neural network for feature extraction and context modeling of text sequences. The decoder then classifies the features at each location, predicting the corresponding named entity category. The multi-head self-attention is a core component of the transformer that helps capture contextual information and entity relationships within the input sequence in the model. The adaptive Weighting mechanism is introduced in this paper to improve the multi-head self-attention mechanism of the transformer. This strategy uses Laplace mixture principles to refine how the distribution is expressed within the attention mechanism. Additionally, it considers the entire text sequence and the significance of every word in the phrase. This enhancement empowers the model to achieve improved performance. The output of the transformer layer is $ST = [st_1, st_2, \ldots, st_n]$, where st_i stands for the attention vector for the present word.

Sequence Labeling layer: The sequence labeling layer uses CRF to address sequence labeling and serialization challenges by modeling the target sequence while considering the observation sequence. This layer forecasts the probabilities for character labels inside the context before and after using $ST = [st_1, st_2, \ldots, st_n]$, the output of the Transformer layer, as input. With training, the CRF model enhances its ability to accurately predict BIO labels for word sequences based on context and label dependencies. The label sequences $F = [y_1, y_2, \ldots, y_n]$ are then produced and y_i denotes the BIO label of the present word.

Output layer: In this layer, the label created by the sequence labeling layer is produced by the output layer.

3.3.2 Multi-Head Self-Attention Mechanism

In this model, the transformer layer utilizes a multi-head self-attention mechanism as a crucial component. Due to the fact that not every word carries equal importance for correctly identifying entities in named entity recognition, the attention mechanism plays a crucial role in configuring the network for entity recognition. The attention mechanism functions as an addressing process, determining the attention value by distributing attention across keys and associating it with corresponding values, all based on a task-specific query vector Q. This process can be deemed as a form of attention-based neural network, which reduces complexity by inputting only relevant task-related information into the neural network, rather than processing all N inputs.

The MHA operation process is shown in the blue box in Fig. 3. The formula for the original MHA process has been elaborated in [23], so it will not be reiterated here. To begin, the input sequence undergoes linear transformations using three trainable weight matrices, yielding vector representations for query (Q), key (K), and value (V). Subsequently, for each attention head, the dot product of Q and K across all positions is computed, followed by scaling and applying the softmax function to derive position weight distributions. These position weights are then applied to the V vectors, resulting in a weighted average at each position and producing the output for each head. Lastly, the outputs generated by multiple attention heads are concatenated and undergo an additional linear transformation to yield the final multi-head attention output.

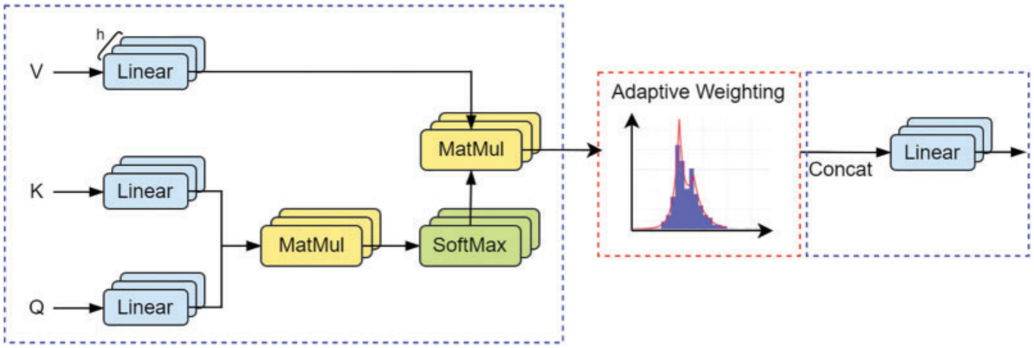

Figure 3: Adaptive weighted multi-attention mechanism structure diagram

3.3.3 Adaptive Weighting Mechanism

In order to further improve the ability of the multi-head self-attention mechanism to capture context information and the robustness of the model in the transformer, this paper proposes an adaptive weighting method based on the idea of the Laplace mixture model to realize the adaptive weight allocation.

The Laplacian distribution can be viewed as a combination of two exponential distributions symmetrically arranged back-to-back, often referred to as the double exponential distribution. Unlike the normal distribution, the Laplacian distribution shares a similar overall shape but is distinguished by heavier tails, making it more sensitive to outliers and exceptional observations. A single random variable, denoted as x, adheres to the Laplace distribution with a mean of u and a scale parameter of b. In the context of a D-dimensional vector x, if its individual elements conform to the Laplace distribution with u as the mean vector, the probability density density can be described as follows:

$$p(x|u,b) = \frac{1}{2b}\exp\left\{-\frac{|x-u|}{b}\right\} \tag{11}$$

where u is the D-dimensional mean vector, and b is the scale parameter of this distribution.

The linear superposition of Laplace distributions can fit very complex density functions. Complex continuous density can be fitted by superposing enough Laplace distributions and adjusting their mean, scale parameters, and linear combination coefficients. The Laplace mixture model can be viewed as a model comprised of K individual Laplace models, with these K sub-models representing the mixture model's latent variables or hidden variables. This study considers the linear superposition of K Laplace distributions. The probability density function for this Laplace mixture distribution is as follows:

$$p(x) = \sum\nolimits_{k=1}^{K} \Pi_k P_k\left(x_i|u_k,b_k\right) \tag{12}$$

where $p_k(x_i|u_k,b_k)$ represents the probability density of the Laplace distribution with the parameter mean u_k and the scale parameter b_k, and $p(x) \geq 0$, $p_k(x_i|u_k,b_k) \geq 0$.

3.3.4 AW-MHA

In the Multi-Head Attention (MHA) mechanism, the model is divided into plural attention heads, to compose plural sub-spaces, allowing it to focus on information from various directions. While this benefits overall model training, increasing the number of attention heads can lead to expression bottlenecks, limiting the model's ability to capture context. To address this, this article draws inspiration from the Laplace mixture distribution and mixes and superimposes the attention weight matrices generated by plural attention heads to improve the distribution expression in attention. The improved operation process of MHA is shown in the red box in Fig. 3. After the output for each head, the attention weight matrices generated by each attention head are combined with the characteristics of a mixture Laplace distribution.

The introduction of Adaptive weighting to the transformer's multi-head self-attention mechanism allows for the prioritization of words in different positions, enabling the model to focus on positions relevant to entities and thereby improving named entity recognition accuracy. The adaptive weighting mechanism, based on a Laplacian mixed distribution, has a stronger response to outliers due to the fat-tailed nature of the Laplacian distribution. This enhancement makes the model more resilient and adaptable to variations in different samples and contexts.

The weight matrix A_i calculated for each attention head is as follows:

$$A_i = Att(Q_i, K_i, V_i), i = 1, \ldots\ldots, h \tag{13}$$

Combining the idea of the Laplace mixture distribution and taking the weight matrix of each attention head as a base distribution, the AW-MHA formula is as follows:

$$AW_{MHA(Q,K,V)} = W^o \sum_{k=1}^{K} \Pi_k A_k \tag{14}$$

4 Experiments

This section begins by introducing the dataset, comparative algorithms, and evaluation indices employed in this paper. Subsequently, Section 4.3 provides a detailed analysis of the performance of the EQBMIAL and AW-MHA algorithms. In Section 4.4, statistical tests are conducted on the experimental results, further illustrating the significance of this algorithm. Finally, Section 4.5 delves into the analysis of the algorithm's time complexity.

4.1 Dataset and Comparison Algorithm

To assess the proposed algorithm's performance, this study utilizes two datasets: the fine-grained named entity recognition dataset ClueNER and the network device configuration dataset NetDevConfNER. ClueNER is a Chinese dataset, which is used to identify named entities such as person names, place names, and organization names, with a training-to-test ratio set of 10:1. NetDevConfNER contains network configuration files from two vendors, provided by an internet service provider. These configuration files contain all parameter information that the devices adhere to during runtime. For this dataset, the training-to-test set ratio is approximately 2:1, encompassing 66 different data types.

To verify the improved sample sampling method of active learning proposed in this paper, comparison methods such as RANDOM, TE [24], MNLP [25], LC [26], and EQB [27] are employed. Among them, RANDOM selects sentences randomly for labeling, TE selects sentences with the highest entropy values, LC selects samples with the lowest confidence, MNLP is based on LC but uses regularized logarithmic probability to express uncertainty and solves the tendency of long sentences in the LC score, and EQB trains a committee of classifiers using labeled samples to select the "most inconsistent" unlabeled samples based on voting entropy.

To validate the entity extraction method, a comparison is conducted with three existing algorithms: BILSTM-CRF [28], Mutil_Attention-Bilstm-Crf [29], Deep_Neural_Model_NER [30] and BERT_Transformer [18]. The BILSTM-CRF efficiently captures two-way semantic interdependence when performing contextual sequence labeling tasks using a BiLSTM network. Additionally, it makes extensive use of conditional random fields to take into account the limitations and reliance between neighboring character label criteria. The Multi-Attention-BiLSTM-CRF algorithm includes both a conditional random field and a BiLSTM network. By using a multi-head attention mechanism, it considerably improves named entity recognition performance And captures plural semantic features from the character, word, and sentence levels. The Deep_Neural_Model_NER utilizes both a conditional random field and a BiLSTM network. It also incorporates a CNN to obtain local features from the current phrase, enhancing named entity identification performance. The BERT_Transformer first utilizes a semi-supervised entity-enhanced BERT method for pre-training. It then integrates entity information into BERT using the CharEntity-Transformer. Finally, it performs entity classification for Chinese entities. The suggested approach will be evaluated in comparison to these recognized.

4.2 Evaluation Index

In this part, the effectiveness of entity extraction is assessed using several measures. For the sake of evaluating the predictive ability of the presented algorithm, precision, recall, and F1-score are utilized while considering entity extraction to be a multi-classification problem. Out of all projected positive instances, precision is the percentage of correctly predicted positive events. The proportion of actual positive instances that were correctly anticipated, on the other hand, out of all positive instances is known as recall. The F1-score evaluates precision and Recall thoroughly while weighing their trade-offs. The following are the calculation formulas for the aforementioned indicators:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

TP stands for genuine positives, which are occurrences of positivity that were accurately expected to be positive; TN, often known as cases that were accurately expected to be negative; FP stands for cases of negative data that were mistakenly forecasted as positive, and FN for instances of positive data that were mistakenly projected as negative. These metrics help assess how accurately the program can extract items.

4.3 Result Analysis

In this section, the results of the sample training set sampling technique and entity extraction are analyzed. Comparisons are made between the suggested algorithm and various other algorithms to demonstrate its superior performance.

The parameter values for the algorithms in this paper were determined through a combination of empirical knowledge and multiple iterations to select the best-performing parameter configurations. During the process of parameter selection, this paper relied on prior research and domain expertise to initially define the parameter ranges. Subsequently, a series of experiments were conducted, testing various combinations of parameters, and ultimately selecting the parameter settings that exhibited the best performance under the experimental conditions.

During the experiment, a sensitivity analysis was conducted on the critical parameters of the network configuration entity extraction model. Initially, various word embedding dimensions, including 64, 128, and 256, were tested. The results revealed that higher dimensions led to improved performance but also incurred higher computational costs. Consequently, this paper opted for a word embedding dimension of 128. Subsequently, parameters of the CNN, such as kernel size and the number of convolutional layers, were explored. Ultimately, a kernel size of 3 × 3 and the inclusion of 2 convolutional layers yielded the best results. Following that, we examined the transformer's layer count and the number of attention heads. Experiments were conducted with layer counts of 1, 2, and 4, demonstrating that higher layer counts facilitated the model in learning more information, resulting in enhanced performance at the cost of increased training time. Regarding the number of attention heads, we conducted experiments by incrementally increasing it from 4 to 20. The results showed an initial performance improvement followed by a decline. Without enhancements to the transformer, the performance plateaued at around 12 attention heads. However, due to the improvements made to the multi-head attention mechanism in this paper, the model exhibited superior performance and greater stability even when the number of attention heads reached 12, with a smaller decline in performance.

The enhanced active learning sample sampling approach is compared with several benchmark methods, including RANDOM, LC, MNLP, TE, and EQB, to validate it. The AW-MHA model is used in the entity extraction stage model. Both the ClueNER dataset and the NetDevConfNER dataset are used to test the sampling technique. In this research, the ordinate indicates changes in the three assessment metrics—precision, Recall, and F1-score—while the abscissa represents the proportion of samples chosen for manual labeling to the training set. Each technique is run ten times for each termination circumstance, and the average of the outcomes is used to get the final value.

Figs. 4 to 6 show the outcomes of the approaches used on the ClueNER dataset, while Table 2 summarizes the detailed indicator values for the NetDevConfNER dataset. These contrasts and studies show how well the suggested algorithm performs and how well it extracts entities.

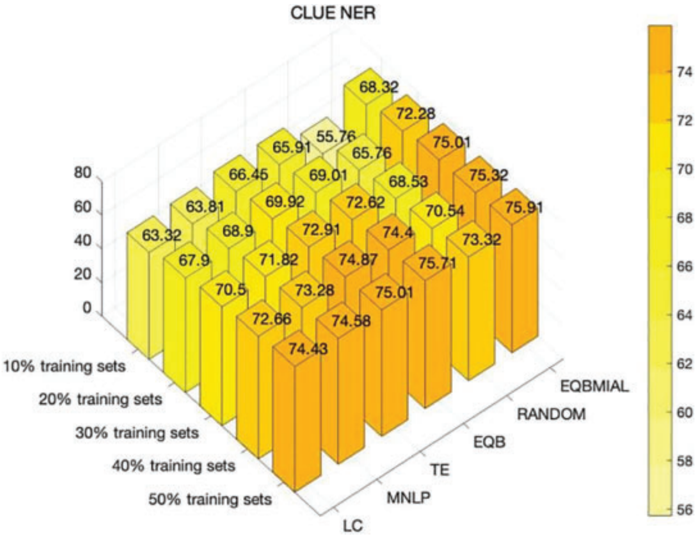

Figure 4: ClueNER dataset precision index comparison

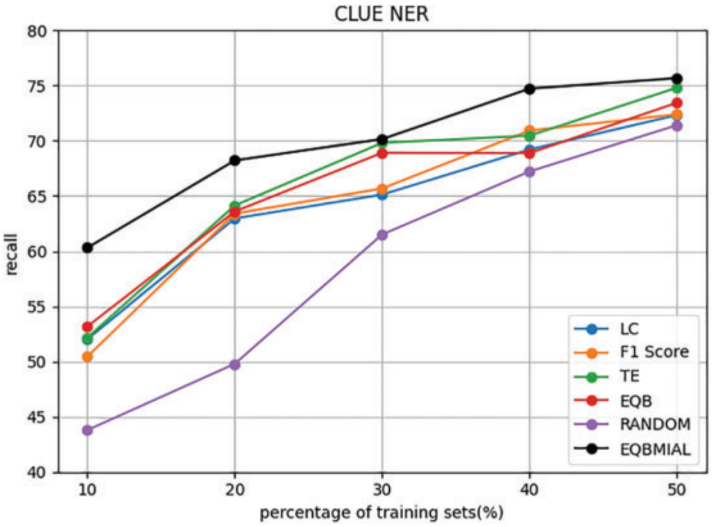

Figure 5: ClueNER dataset recall index comparison

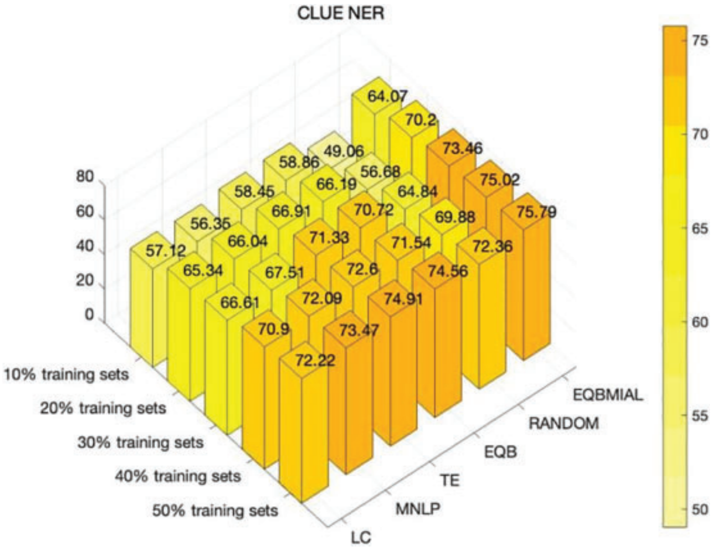

Figure 6: ClueNER dataset F1-score index comparison

Table 2: Comparison of evaluation indicators of active learning methods on NetDevConfNER dataset (%)

Percentage of training set\ active learning methods		LC	MNLP	TE	EQB	RANDOM	EQBMIAL
	Precision	76.04	77.82	78.87	77.80	72.65	79.17
10%	Recall	79.62	80.14	80.83	79.31	73.67	82.55
	F1-score	77.79	78.96	79.84	78.55	73.16	80.82

(Continued)

Table 2 (continued)

Percentage of training set\ active learning methods		LC	MNLP	TE	EQB	RANDOM	EQBMIAL
20%	Precision	79.55	80.17	84.73	84.54	76.07	87.06
	Recall	80.78	81.88	84.21	83.91	78.59	87.56
	F1-score	80.16	81.02	84.47	84.22	77.31	87.31
30%	Precision	83.02	84.75	86.03	85.91	81.80	89.14
	Recall	82.97	83.42	86.73	85.55	80.12	90.93
	F1-score	82.99	84.08	86.38	85.73	80.95	90.03
40%	Precision	85.07	86.07	90.56	89.12	85.19	93.74
	Recall	84.24	85.56	88.13	87.02	83.09	92.68
	F1-score	84.65	85.81	89.33	88.06	84.13	93.21
50%	Precision	90.83	91.06	93.96	92.23	88.80	95.75
	Recall	88.69	90.35	92.55	89.48	86.15	94.91
	F1-score	89.75	90.70	93.25	90.83	87.45	95.30

Fig. 4 shows that the EQBMIAL method has better results than several other comparison methods as far as precision is concerned. When the percentage of the training set reaches 10%, 20%, 30%, 40%, and 50%, the EQBMIAL method in this paper is improved by 1.87%, 2.36%, 2.1%, 0.45%, and 0.2%, respectively, compared to the optimal TE method. Using the samples required for precision as a comparison, the optimal TE method in the comparison algorithm requires at least 50% of the samples to achieve 75% precision, while the EQBMIAL method proposed in this paper only needs 40% of the samples can achieve 75% precision.

Fig. 5 shows that the EQBMIAL method has better results than several other comparison algorithms as far as recall is concerned. When the percentage of the training set reaches 10%, 20%, 30%, 40%, and 50%, the EQBMIAL method proposed in this paper is improved by 7.14%, 4.08%, 0.34%, 3.5%, and 0.87%, respectively, compared to the optimal TE method. Using the samples required for the recall rate as a comparison, the optimal TE method in the comparison algorithm requires at least 50% of the samples to achieve a recall rate of 74%, while the EQBMIAL method put forward in this article only needs 40% of the samples can achieve 74%.

Fig. 6 demonstrates that in the F1-score index, the EQBMIAL approach is preferable to the other comparison methods. When the percentage of the training set reaches 10%, 20%, 30%, 40%, and 50%, the EQBMIAL method proposed in this paper is improved by 5.21%, 3.29%, 2.13%, 2.42%, and 0.88% compared to the optimal TE method. Using the samples required by the F1-score as a comparison, the optimal TE method in the comparison algorithm requires at least 50% of the samples to achieve 74% of the F1-score, while the EQBMIAL method proposed in this paper only needs 40% of the samples can achieve a 74% F1-score.

The indicators of the method on the NetDevConfNER dataset are shown in Table 2.

In terms of recall, the EQBMIAL methodology outperforms the other comparative techniques. When the training set's percentage hits 10%, 20%, 30%, 40%, and 50%, the EQBMIAL method proposed in this paper is improved by 1.72%, 3.35%, 4.2%, 4.55%, and 2.36%, respectively, when compared to the optimal TE method. Using the samples required for the recall rate as a comparison, the optimal TE method in the comparison algorithm requires at least 50% of the samples to achieve a recall rate of 92%, while the EQBMIAL method proposed in this paper only needs 40% of the samples can reach a recall rate of 92%.

Table 2 shows that the EQBMIAL method is greater than the other comparison technologies in the F1-score index. When the percentage of the training set reaches 10%, 20%, 30%, 40%, and 50%, the EQBMIAL method proposed in this paper is improved by 0.98%, 2.84%, 3.65%, 3.88%, and 2.05%, respectively, compared to the optimal TE method. Taking the samples required by the F1-score as a comparison, the optimal TE method in the comparison algorithm requires at least 50% of the samples to achieve an F1-score of 93%, while the EQBMIAL method proposed in this paper only needs 40% of the samples can achieve a 93% F1-score.

It is evident that the results of using the EQBMIAL approach suggested in this research for the ClueNER and NetDevConfNER datasets greatly surpass those of previous comparison methods. The entity extraction approach suggested in this paper is then further validated through comparison with the comparison algorithms BILSTM-CRF, Mutil_Attention-Bilstm-Crf, Deep_Neural_Model_NER, and BERT_Transformer. The evaluation compares the precision, recall, and F1-score on both the ClueNER dataset and the NetDevConfNER dataset.

Next, the performance metrics of the algorithm on the ClueNER and NetDevConfNER datasets are summarized. When compared to the chosen benchmark algorithms, these findings attest to how effective and better the proposed entity extraction algorithm is.

Figs. 7 and 8 show the comparison of indicators of various entity extraction on the ClueNER dataset and the NetDevConfNER dataset.

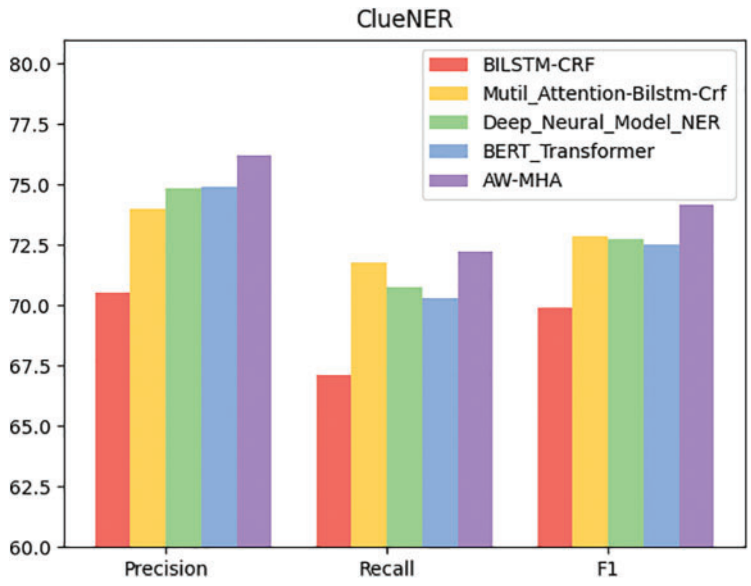

Figure 7: Comparison of several ClueNER dataset algorithm indications

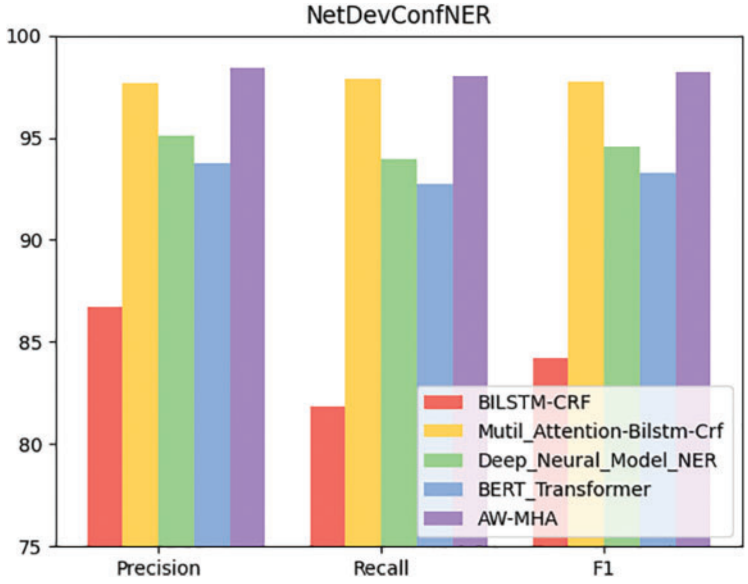

Figure 8: Comparison of several NetDevConfNER dataset algorithm indications

According to Fig. 7, on the ClueNER dataset, the precision index values for BILSTM-CRF, Mutil_Attention-Bilstm-Crf, Deep_Neural_Model_NER, and BERT_Transformer are 70.5%, 74.01%, 74.82%, and 74.89%, respectively. In contrast, the precision value achieved by the method put forward in this article is 76.21%, surpassing the optimal comparison algorithm, BERT_Transformer by 1.32%. Regarding recall, the method put forward in this article reached a recall value of 72.25%, outperforming the optimal comparison algorithm, Mutil_Attention-Bilstm-Crf, by 0.49%. Furthermore, the F1-score of the comparison algorithms BILSTM-CRF, Mutil_Attention-Bilstm-Crf, Deep_Neural_Model_NER, and BERT_Transformer achieved index values of 69.87%, 72.86%, 72.74%, and 72.51%, respectively. In contrast, the value of the F1-score achieved by the proposed method is 74.18%, exhibiting a remarkable increase of 4.31% and outperforming the BILSTM-CRF algorithm. These results indicate that the algorithm put forward in this paper achieves superior outcomes regarding precision, recall, and F1-score compared with those selected comparison algorithms, making it a more effective solution for entity extraction tasks on the ClueNER dataset.

Fig. 8 illustrates that the overall performance of various algorithms on the NetDevConfNER dataset is notably superior to that on the ClueNER dataset. The precision indicators of different algorithms show a slight variation, with BILSTM-CRF having the lowest precision index value of 86.74%, while AW-MHA achieves the highest precision with an index value of 98.43%. The suggested approach in this study outperforms the foremost Mutil_Attention-Bilstm-Crf algorithm by a small margin, with an increase of 0.74%. Regarding recall, the value of Recall achieved by the proposed algorithm is 98.01%, outperforming the Mutil_Attention-Bilstm-Crf algorithm by 0.14%. Regarding the F1-score, the BILSTM-CRF, Mutil_Attention-Bilstm-Crf, Deep_Neural_Model_NER, BERT_Transformer algorithms severally achieved index values of 84.21%, 97.76%, 94.55%, and 93.25%. In comparison, the value of the F1-score achieved by the proposed method is 98.22%, showing an improvement of 0.46% and outperforming the Mutil_Attention-Bilstm-Crf algorithm. Due to its focus on Chinese entity recognition, the BERT_Transformer algorithm is pre-trained using a Chinese character-based BERT model, which may result in lower performance on the network configuration dataset. These results indicate that the method suggested in this paper achieves superior outcomes of precision, recall, and F1-score compared with those selected comparison algorithms, demonstrating its effectiveness for entity extraction tasks on the NetDevConfNER dataset.

4.4 Statistical Results

In order to validate that the algorithm proposed in this paper exhibits significant differences compared to other algorithms in the task of network configuration entity extraction, the Kruskal-Wallis test is used for verification. Before conducting the statistical test, performance metrics for different algorithms are first grouped according to the algorithms. A significance level of 0.05 is set. In this test, the null hypothesis states that there is no significant difference in performance between the algorithm proposed in this paper and the other algorithms.

First, a Kruskal-Wallis test was conducted on the EQBMIAL algorithm proposed in this paper with a training set percentage of 50%. The calculated *p*-value for the CLUENER dataset was 0.0215, and for the NetDevConfNER dataset, it was 0.0114. In both of these datasets, the *p*-value is less than 0.05, leading to the rejection of the null hypothesis. Therefore, it can be concluded that the performance of the EQBMIAL algorithm is statistically highly significant on these two datasets.

Next, a Kruskal-Wallis test was conducted on the AW-MHA method. For the CLUENER dataset, the calculated *p*-value was 0.1586, indicating that this algorithm does not exhibit a significant difference from other algorithms. However, for the NetDevConfNER dataset, the calculated *p*-value was 0.0090, leading to the rejection of the null hypothesis. While its performance in the CLUENER dataset may not significantly outperform other algorithms, its excellent performance on the network configuration dataset demonstrates that the algorithm excels and is highly significant for the task of network configuration entity extraction.

4.5 Time Complexity Analysis

This section will delve into the time complexity analysis of the entity extraction stage. Among the elements within the entity extraction component, the most computationally demanding aspect is the enhanced transformer model. Consequently, our primary focus will be on the time complexity of this particular component. In this paper, the enhanced transformer layer incorporates an adaptive weighting mechanism based on the Laplace mixture distribution into the transformer's multi-head attention mechanism. Although this approach may introduce some additional computational overhead, the core time complexity of this section remains predominantly determined by the multi-head attention mechanism itself. As a result, we can represent the time complexity of this section of the model as $\mathrm{O}(\mathrm{n}^2\mathrm{d} + \mathrm{nd}^2)$, where n represents the length of the input sequence, and d represents the dimension of word embeddings.

5 Conclusion

This paper focuses on entity extraction using active learning and a multi-head self-attention mechanism with adaptive weighting. It starts with a review of relevant literature and introduces the concept of Laplace mixture distribution to address issues in the extraction model. The paper also presents EQBMIAL, an active learning method designed to handle outliers and redundancy in the training set. Additionally, AW-MHA is proposed to overcome challenges arising from the increasing number of attention heads.

Simulation experiments were conducted to compare the proposed algorithm with other models such as RANDOM, LC, MNLP, TE, and EQB. The results demonstrate significant improvements across various evaluation metrics, particularly achieving a precision of 98.32% on the NetDevConfNER dataset. This algorithm outperforms other models in network configuration entity recognition, highlighting its superior performance in current entity recognition tasks.

However, it is crucial to consider adversarial attacks when developing network configuration entity recognition models. These models are vulnerable to intentional manipulation of input data by attackers, leading to incorrect outputs. Therefore, enhancing the model's resilience against such attacks becomes essential. One approach to achieve this is through adversarial training, which can improve the model's robustness.

Acknowledgement: The authors thank the anonymous reviewers for their careful reading and valuable suggestions. Their professional opinions and suggestions have enabled me to explore and improve the research more comprehensively. In addition, we sincerely thank all the members involved in this paper for their support and valuable comments. Our collaboration has made this research richer and more meaningful.

Funding Statement: This work is supported by the National Key R&D Program of China (2019YFB2103202).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yang Yang, Zhenying Qu, Zefan Yan; data collection: Zefan Yan, Zhenying Qu, Ti Wang; analysis and interpretation of results: Yang Yang, Zhenying Qu, Zefan Yan, Zhipeng Gao; draft manuscript preparation: Zefan Yan, Zhenying Qu, Yang Yang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The ClueNER dataset can be accessed through "https://github.com/CLUEbenchmark/CLUENER2020". The NetDevConfNER dataset cannot be made publicly accessible due to proprietary restrictions.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. X. Ji, S. R. Pan, E. Cambria, P. Marttinen and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2021.

[2] S. Yang, L. Dong, G. Deng and Y. Liu, "Design and implementation of fault diagnosis system for power communication network based on CNN," in *Proc. of the 2021 13th Int. Conf. on Communication Software and Networks*, Chongqing, China, pp. 69–74, 2021.

[3] S. Dezfulian, Y. Ghaedsharaf and N. Motee, "Performance analysis and optimal design of time-delay directed consensus networks," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 197–209, 2021.

[4] K. Shaukat, S. Luo and V. Varadharajan, "A novel deep learning-based approach for malware detection," *Engineering Applications of Artificial Intelligence*, vol. 122, pp. 106030, 2023.

[5] D. M. Bikel, R. Schwartz and R. M. Weischedel, "An algorithm that learns what's in a name," *Machine learning*, vol. 34, pp. 211–231, 1999.

[6] A. E. Borthwick, "A maximum entropy approach to named entity recognition," Ph.D. dissertation, New York University, USA, 1999.

[7] H. L. Chieu and H. T. Ng, "Named entity recognition: A maximum entropy approach using global information," in *Proc. of the 19th Int. Conf. on Computational Linguistics*, Taipei, Taiwan, pp. 190–196, 2002.

[8] J. Mayfield, P. McNamee and C. Piatko, "Named entity recognition using hundreds of thousands of features," in *Proc. of the Seventh Conf. on Natural Language Learning at HLT-NAACL 2003*, Edmonton, Canada, pp. 184–187, 2003.

[9] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction, and web-enhanced lexicons," in *Proc. of the Seventh Conf. on Natural Language Learning*, Edmonton, Canada, pp. 188–191, 2003.

[10] B. Settles, "Biomedical named entity recognition using conditional random fields and rich feature sets," in *Proc. of the Int. Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Geneva, Switzerland, pp. 104–107, 2004.

[11] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural architectures for named entity recognition," in *Proc. of the 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, USA, pp. 260–270, 2016.

[12] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang *et al.,* "An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition," *Bioinformatics*, vol. 34, no. 8, pp. 1381–1388, 2018.

[13] K. Xu, Z. Yang, P. Kang, Q. Wang and W. Liu, "Document-level attention-based BiLSTM-CRF incorporating disease dictionary for disease named entity recognition," *Computers in Biology and Medicine*, vol. 108, pp. 122–132, 2019.

[14] A. K. Singh, I. R. Khan, S. Khan, K. Pant, S. Debnath *et al.,* "Multichannel CNN model for biomedical entity reorganization," *BioMed Research International*, vol. 2022, pp. 1–11, 2022.

[15] E. Parsaeimehr, M. Fartash and J. Akbari Torkestani, "Improving feature extraction using a hybrid of CNN and LSTM for entity identification," *Neural Processing Letters*, vol. 55, no. 5, pp. 5979–5994, 2023.

[16] S. He, T. Deng, B. Chen, R. S. Sherratt and J. Wang, "Unsupervised log anomaly detection method based on multi-feature," *Computers, Materials & Continua*, vol. 76, no. 1, pp. 517–541, 2023.

[17] S. Alissa and M. Wald, "Text simplification using transformer and BERT," *Computers, Materials & Continua*, vol. 75, no. 2, pp. 3479–3495, 2023.

[18] C. Jia, Y. Shi, Q. Yang and Y. Zhang, "Entity enhanced BERT pre-training for Chinese NER," in *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6384–6396, 2020.

[19] A. Beygelzimer, S. Dasgupta and J. Langford, "Importance weighted active learning," in *Proc. of the 26th Annual Int. Conf. on Machine Learning*, Montreal, Canada, pp. 49–56, 2009.

[20] D. Mahapatra, B. Bozorgtabar, J. P. Thiran and M. Reyes, "Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network," in *Proc. of the Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, Granada, Spain, vol. 11071, pp. 580–588, 2018.

[21] C. Wu, G. Luo, C. Guo, Y. Ren, A. Zheng *et al.,* "An attention-based multi-task model for named entity recognition and intent analysis of chinese online medical questions," *Journal of Biomedical Informatics*, vol. 108, pp. 103511, 2020.

[22] R. A. Hallyal, C. Sujatha, D. Padmashree and S. M. Meena, "Optimized recognition of CAPTCHA through attention models," in *2023 IEEE 8th Int. Conf. for Convergence in Technology (I2CT)*, Lonavla, India, pp. 1–7, 2023.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Proc. of the 31st Int. Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, pp. 6000–6010, 2017.

[24] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. of the 2008 Conf. on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, USA, pp. 1070–1079, 2008.

[25] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod and A. Anandkumar, "Deep active learning for named entity recognition," in *Proc. of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada, pp. 252–256, 2017.

[26] P. Radmard, Y. Fathullah and A. Lipani, "Subsequence based deep active learning for named entity recognition," in *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. on Natural Language Processing*, vol. 1, pp. 4310–4321, 2021.

[27] L. Copa, D. Tuia, M. Volpi and M. Kanevski, "Unbiased query-by-bagging active learning for VHR image classification," in *Image and Signal Processing for Remote Sensing XVI*, Toulouse, France, vol. 7830, pp. 176–183, 2010.

[28] Z. Huang, W. Xu and K. Yu, "Bidirectional LSTM-CRF models for sequence taggin," arXiv:1508.01991, 2015.

[29] G. Wu, G. Tang, Z. Wang, Z. Zhang and Z. Wang, "An attention-based BiLSTM-CRF model for Chinese clinic named entity recognition," *IEEE Access*, vol. 7, pp. 113942–113949, 2019.

[30] T. Lê and M. S. Burtsev, "A deep neural network model for the task of named entity recognition," *International Journal of Machine Learning and Computing*, vol. 9, no. 1, pp. 8–13, 2019.