INTEGRATING BOUNDARY AWARENESS AND DYNAMIC LOSS FOR ENHANCING SEMI-SUPERVISED NER IN POWER COMMUNICATION FAULT TEXTS

Jieke LU[1], Ying LING[2], Xin LI[3,*], Shaofeng MING[4]

As an important data source for troubleshooting, power communication fault text holds significant value in fault diagnosis. However, the effectiveness of generic named entity recognition (NER) techniques is limited due to the irregular nature of domain-specific text, often characterized by small dataset size and nested entities. To address these challenges, this paper proposes an enhanced semi-supervised NER method for power communication fault texts with boundary awareness and dynamic loss. First, the span semantic representation matrix is constructed by capturing the head and tail vectors of entities using BERT with rotary position encoding. Then, a third-order difference operator is introduced to enhance the semantic separation between entities and boundary spans and to better capture potential entity representations. To improve the model's prediction of such potential entities, a length masking (LM) mechanism and dynamic weighted loss strategy are applied. Furthermore, a semi-supervised training strategy incorporating data augmentation is employed to enhance the model's generalization and robustness. Experimental results on a real-world power communication fault corpus demonstrate that the proposed method achieves superior performance compared to baseline models, with precision, recall, and F1-score reaching 95.49%, 93.37%, and 94.42%, respectively. These results indicate that carefully integrating boundary-aware span modeling and semi-supervised training strategies can effectively enhance entity recognition performance in low-resource, domain-specific scenarios.

Keywords: Entity Recognition, Power Communication, Semi-supervised, Negative Sampling, Dynamic Weighting

1. Introduction

The power communication network is crucial for power system operations [1]. TIMS (Telecommunication Information Management System), an essential part of it, maintains network information and ensures normal operation. While handling fault information, TIMS generates defect-overhaul data, but currently,

[1] Eng., Electric Power Research Institute of Guangxi Power Grid Co., Ltd., Nanning 530023, China, 515069043@qq.com
[2] Eng., Electric Power Research Institute of Guangxi Power Grid Co., Ltd., Nanning 530023, China, yinglingly@163.com
[3]* Eng., Electric Power Research Institute of Guangxi Power Grid Co., Ltd., Nanning 530023, China, * Corresponding author: li_xin.sy@gx.csg.cn
[4] Eng., Electric Power Research Institute of Guangxi Power Grid Co., Ltd., Nanning 530023, China, mingsfeng@foxmail.com

operation and maintenance personnel mainly rely on experience, and this data is not effectively utilized. Knowledge graphs [2] can help quickly retrieve and process fault information, so building high - quality graphs is vital for related tasks.

As the foundational step for knowledge graph construction, entity extraction targets key entities from unstructured text [3]. Named entity recognition has three main approaches: rule-based, statistical, and deep learning [4], with the first two facing difficulties in dataset construction and feature selection due to manual feature selection [5]. Deep learning models like LSTM, GRU, and CNN have gained wide attention for extracting deep features. Researchers [6-8] applied sequence annotations (e.g., BIOES, BIO) to domain texts, using BiLSTM and CRF to extract entity information across fields. Bhumireddypalli, Koppula & Koppula [9] adopted ECRF-LSTM for foreign name recognition, optimizing model parameters via CAOA to avoid local optima. He et al. [10] proposed a self-attentive BiGRU-capsule network model for dictionary-independent entity recognition. However, end-to-end training of such word vector-based models hinders textual mining under limited resources. Power communication fault texts present unique challenges—domain specificity, entity diversity, and structural complexity—compared with general-domain texts. They include rare technical terms [11] and domain concepts absent from generic corpora, weakening conventional models. Moreover, entities here (full names, abbreviations, aliases) require models with advanced language understanding to identify equivalent semantics.

To enable efficient and precise entity extraction within power communication fault texts and to establish a foundation for graph construction and downstream applications, this work proposes a novel entity extraction method specifically designed for this domain. The contributions of this work are summarized as follows:

1) We design a boundary-aware span representation framework that integrates BERT-based encoding with rotary position embeddings and higher-order difference operators, tailored for the characteristics of power communication fault texts.
2) We investigate the effect of constraining candidate entity spans using domain-informed length limits and empirically analyze how entity length distributions influence recognition accuracy and computational efficiency.
3) We explore a dynamic loss weighting strategy combined with semi-supervised training and conduct empirical studies to evaluate its impact on handling class imbalance and limited labeled data in a real-world industrial dataset.

2. Power Communication Fault Text Entity Recognition Algorithm

Aiming at the characteristics of power communication fault text and increasing the ability of information extraction, this paper designs a semi-supervised training

strategy and proposes the semi-supervised power communication fault text entity extraction algorithm LM-BRD based on boundary perception. As shown in Fig. 1, the model is mainly divided into 3 parts :(1) BERT-ROPE based head-tail vector encoder, which uses the BERT pre-training model with rotational position encoding to generate the head-tail (q/k) vectors of the entities and construct the entity span matrix M1; (2) Boundary-Aware Based On Third-order Difference Operator, which uses the third-order difference operator to extract the main diagonal semantic features of the entities in the span matrix M1, and to increase the semantic entity and surrounding span distance; (3) Entity Decoder Based On Limiting Entity Length, which eliminates irrelevant spans through a limiting character algorithm (LM), thereby reducing the number of entity predictions and enhancing the model's prediction accuracy.

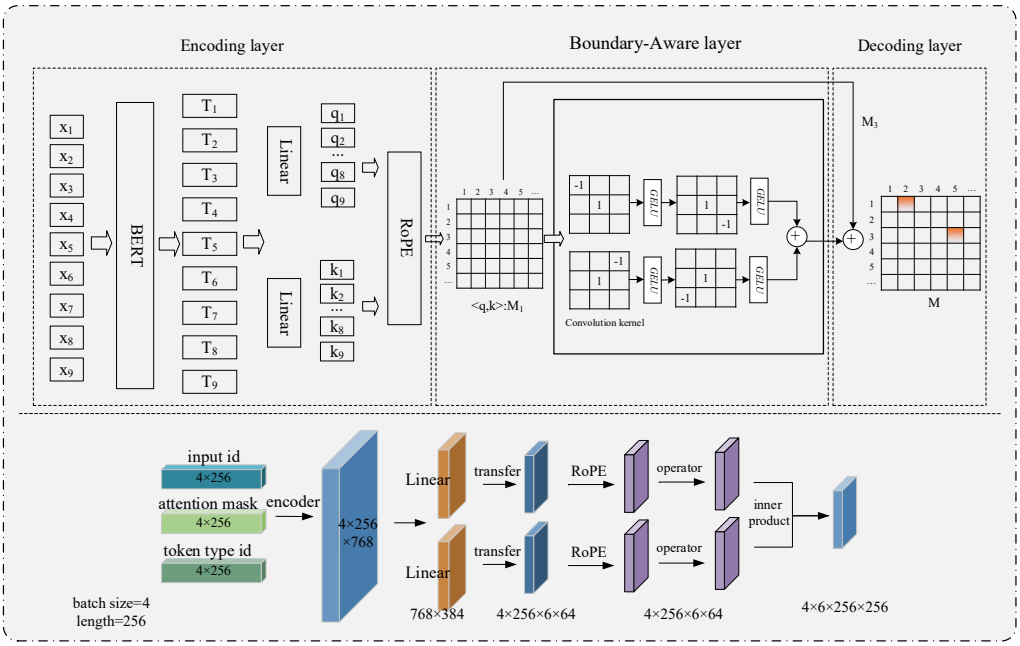

Fig. 1 Modelling framework.

2.1. BERT-ROPE Based Head-tail Vector Encoder

The encoding layer converts text into processable vector representations. Specifically, given an input sequence, the encoder produces contextual token representations $H \in R^{B \times L \times d}$, where B denotes the batch size, L is the input sequence length, and d is the hidden dimension.

In the BERT encoder, each word uses token, segment, and positional embeddings, which are summed and fed into a multi-head self-attention mechanism. Here, representations are refined via Q (query), K (key), V (value)

matrix multiplications to produce output vectors [12], with fused head and tail character $q_i(k_i)$ information generated through two Linear layers (formula below).

$$q_i(k_i) = W_{q/k}(Bert(x_i) + B_{q/k}) \tag{1}$$

where $q_i(k_i) \in \mathrm{R}^{b\times l\times t\times h}$; $W_{q/k} \in \mathrm{R}^{b\times 768\times (t\times h)}$ are trainable parameters; $B_{q/k}$ is the bias term; b is the number of samples in the training batch; l is the maximum text length; t is the predefined entity kind; *Bert* denotes the multi-head self-attention mechanism; q_i , k_i denote the header vector and the footer vector corresponding to position i.

To explicitly encode positional information into the head and tail vectors, we adopt rotary position embedding (RoPE) as introduced in RoFormer [13]. RoPE injects positional information by applying position-dependent rotations to paired dimensions of the query and key vectors, enabling attention scores to depend on relative positional offsets while preserving absolute position information.

Given an even-dimensional head or tail vector, its hidden dimension is decomposed into $h/2$ two-dimensional subspaces. For the $i-$th dimension pair $(2i, 2i+1)$, the RoPE transformation at position mmm is defined as:

$$\mathrm{RoPE}(q_m^{(i)}) = \begin{pmatrix} \cos(m\omega_i) & -\sin(m\omega_i) \\ \sin(m\omega_i) & \cos(m\omega_i) \end{pmatrix} q_m^{(i)}, \quad \mathrm{RoPE}(k_n^{(i)}) = \begin{pmatrix} \cos(n\omega_i) & -\sin(n\omega_i) \\ \sin(n\omega_i) & \cos(n\omega_i) \end{pmatrix} k_n^{(i)} \tag{2}$$

where $q_m^{(i)} = (q_{m,2i}, q_{m,2i+1})^{\cdot}$ and $k_n^{(i)} = (k_{n,2i}, k_{n,2i+1})^{\cdot}$ denote the $i-$th paired subspace of the head and tail vectors, respectively. The angular frequency ω_i is defined as:

$$\omega_i = \theta^{-2i/h}$$

where θ is a predefined constant (set to 10,000 in this work).

After applying RoPE to all paired subspaces, the inner product between the transformed head and tail vectors can be expressed as a summation over dimension pairs, which depends only on the relative position offset $(m-n)$. This property allows the model to capture relative positional relationships between token pairs while maintaining compatibility with standard attention mechanisms. By integrating RoPE into the head–tail vector encoding process, the model enhances its ability to represent span boundaries and contextual dependencies in power communication fault texts.

2.2. Boundary-Aware Module Based On Third-order Difference Operator

In span-based named entity recognition, boundary ambiguity often arises from local noise in token-level representations, especially in power fault texts containing rare terms, aliases, and nested expressions. Such noise may cause sharp

and irregular fluctuations in boundary scores across adjacent spans, leading to unstable span predictions.

Rather than modeling absolute boundary scores, the third-order difference operator is designed to constrain abrupt changes in boundary transition trends. Compared with first- or second-order differences, which mainly capture local score variations, the third-order difference penalizes higher-order instability across neighboring spans. This encourages smoother and more consistent boundary patterns while preserving genuine boundary transitions. By suppressing isolated boundary spikes and enforcing trend-level consistency, the proposed operator effectively reduces spurious span candidates caused bylocal noise, thereby improving boundary localization stability in power fault NER.

In span-based named entity recognition, boundary ambiguity often arises when multiple candidate spans share identical or highly similar head and tail tokens. This phenomenon is particularly common in power communication fault texts, where technical terms, abbreviations, and repeated equipment identifiers frequently appear in similar lexical contexts. As a result, different spans may exhibit highly similar semantic representations, even though they correspond to distinct entity boundaries or non-entity regions. To alleviate this issue, it is insufficient to model absolute boundary scores alone. Instead, capturing relative variations and transition trends between neighboring spans becomes critical for distinguishing true entity boundaries from spurious candidates. Difference-based operators provide a natural mechanism for modeling such local variations. While first-order and second-order differences are effective in capturing immediate or local score changes, they are often sensitive to noise and may fail to suppress higher-order instability across neighboring spans.

In contrast, the third-order difference operator emphasizes trend-level consistency by penalizing abrupt fluctuations in boundary transitions over a wider neighborhood. This property allows the model to suppress isolated boundary spikes caused by local noise, while preserving genuine boundary transitions associated with real entities. Empirically, we find that third-order differences offer a better balance between boundary sensitivity and stability compared to lower-order alternatives in the context of power communication fault texts.

The third order difference operator is calculated as shown in equation (3) below:

$$P_1=\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\ P_2=\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}\ P_3=\begin{bmatrix}0 & 0 & -1\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\ P_4=\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ -1 & 0 & 0\end{bmatrix} \tag{3}$$

$$M_1^1 = GELU(M_1.conv(P_1))$$
$$M_1^2 = GELU(M_1^1.conv(P_2)) \tag{4}$$

where $GELU$ denotes the activation function, P_1 , P_2 represent the two convolution kernels of the main diagonal, and $M.conv$ denotes the convolution operation performed on the matrix M . M_1^2 represents the feature matrix between the span and the main diagonal, and similarly, M_2^2 represents the feature matrix between the span and the sub-diagonal. After obtaining the feature difference matrix M_3 fusing M_1^2 and M_2^2, the original features are lost in the matrix due to the convolution operation. In order to retain the original span and difference feature information and obtain a more complete semantic feature matrix, M_3 is made as the residual term of M_1, and M is obtained after a learnable linear layer, and Eq. (5) is shown below:

$$\begin{aligned} M_3 &= M_1^2 \oplus M_2^2 \\ M &= W \cdot (M_1 \oplus M_3) + B \end{aligned} \tag{5}$$

where $\oplus$ stands for matrix summation; W is the learnable parameter and B is the bias term of the linear layer.

2.3. Entity Decoder Based On Limiting Entity Length

The size of the labeling matrix grows rapidly with the maximum input character length. As illustrated in Fig. 2, even though the lower triangular region of the span matrix is masked to reduce redundant predictions, the number of candidate spans that the model needs to evaluate still increases substantially as the input text becomes longer. This expansion leads to a large number of parameters to be predicted, many of which correspond to extremely long spans that are unlikely to form valid entities in practical scenarios.

From an empirical perspective, only a small fraction of long-span candidates corresponds to true entities in power communication fault texts. Once the span length exceeds a reasonable threshold, it becomes increasingly improbable for such spans to represent correct entity boundaries. Consequently, predicting all possible long spans not only introduces unnecessary computational overhead but also increases the risk of false positives. Motivated by this observation, this study proposes a character-level length constraint mechanism, referred to as Limit Max_entity_length (LM), which operates directly on the span matrix M_1 to restrict the prediction range of candidate spans. As shown in Fig. 2, after applying the character length constraint, the span matrix is effectively transformed, resulting in a significant reduction in the number of spans that need to be predicted. Notably, the longer the input text, the more pronounced the reduction in model parameters, thereby improving both computational efficiency and prediction accuracy.

It should be noted that LM is a heuristic constraint guided by the empirical characteristics of the dataset rather than a theoretically optimal rule. By filtering out semantically implausible long spans while retaining the majority of valid entity candidates, the proposed mechanism enables the model to focus on more informative span regions, laying the foundation for more robust entity decoding.

After limiting the length of the entity, the model needs to predict a reduced number of label lengths as shown in Eq. (6):

$$\frac{(seq\_len - limit)\cdot(seq\_len - limit + 1)}{2} \tag{6}$$

where seq_len represents the sentence length; $limit$ represents the entity limit length.

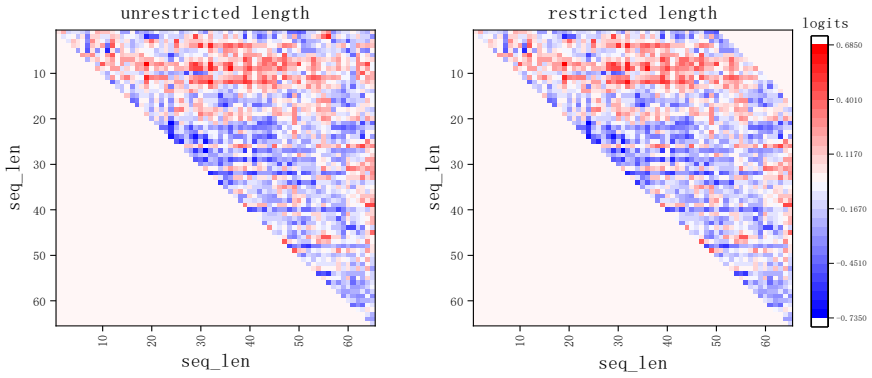

Fig. 2 Label matrix change before and after limiting length=50.

Although the above span length limitation effectively reduces the prediction space, boundary ambiguity may still exist among neighboring candidate spans. Therefore, we further introduce a boundary-aware modeling mechanism based on a third-order difference operator to refine span-level boundary discrimination.

3.Model Training Strategies

3.1. Dynamic Weight Loss Functions that Balance Predictive Entity Difficulty and Sample Balance

In loss computation, the exhaustive span strategy causes inherent positive–negative sample imbalance in labeling [14]. To mitigate this issue, a screening mechanism is adopted to convert the original multi-label multi-classification into a binary classification problem [15], whose mathematical form is given in Eq. (7):

While the screening mechanism effectively reduces the dominance of trivial negatives at the formulation level, in practice the training process can still be biased toward easy samples, and hard or ambiguous spans may receive insufficient optimization. Therefore, we further introduce a dynamic weighted loss to adjust the contribution of samples across training stages, improving robustness under class imbalance and limited labeled data.

The dynamic weighting scheme involves several hyperparameters, namely start, end, penalty, and thresh, each serving a specific role. start and end define the lower and upper bounds of the weighting factor, controlling the overall strength of reweighting. penalty determines the growth rate of the weight during training, enabling a gradual transition from a stable optimization phase to a more

discriminative learning phase. Meanwhile, thresh functions as a confidence-related threshold to prevent overly aggressive reweighting on uncertain predictions, thereby stabilizing training. In this work, a linear schedule is adopted due to its simplicity and stable behavior on small-scale domain-specific datasets, and the detailed formulation is provided in Eq. (8–11).

$$\log(1+\sum_{i\in neg} e^{i}) + \log(1+\sum_{j\in pos} e^{j}) \tag{7}$$

The left term in Eq. (7) uses log sum exp (LSE) on screened positive samples, and the right term is the loss after screening out positives; denotes negative samples and positive samples. Though Eq. (7) alleviates positive-negative imbalance, it ignores labeling difficulty. Thus, this paper proposes a matrix-based dynamic weight loss (DWL) method, which considers both imbalance [16] and labeling difficulty, and increases loss weights via dynamic ways of linear based on training steps (Eq. (8)) to design a refined loss matrix and enhance entity prediction. This formula represents the variation of weight values from two custom parameters.

$$penalty = \frac{step \cdot 5}{global} \cdot (end - start) + start \tag{8}$$

where $step$ denotes the current training step; $global$ denotes the number of global training steps; $start$ and end denote the initial values of weights, respectively; and $penalty$ represents the dynamic weight coefficients.

In the model output matrix M^1, the prediction error set is denoted as: $\{w \mid w > 0, w \in M\}$. to further perceive the impact of the difficult entity loss on the whole model. To enhance the learning of challenging entities, dynamic weight coefficients are applied to the loss associated with the error set, thereby increasing the variability in training. The loss computation is detailed in Eq. (9):

$$L_1 = \log(1+\sum_{m\in M^1, m\notin w} e^{m} + \sum_{n\in M^1, n\in w} e^{n*penalty}) \tag{9}$$

where L_1 represents the loss value; m represents the difference between the M^1 and w sets; and n represents the concatenation of the M^1 and w sets. Consider the loss value function for predicting sample equilibrium

In addition, for the pos set, to maintain stable representation of information that is prone to over- or underestimation in the model's predictions, it is not possible to multiply the correctly predicted part by the dynamic weight coefficients directly in the same way as in Eq. (10). For this reason, the definition of the threshold is used to attenuate the attention of the correctly predicted part. Using the formula $\{t \mid t < thresh, t \in pos\}$ to denote the set t that is predicted correctly, the loss value of the correct part of the model's prediction L_2 be:

$$L_2 = \log(1+\sum_{t\in pos, t<thresh} e^{t\cdot penalty} + \sum_{r\in pos, r>thresh} e^{r}) \tag{10}$$

Where *penalty* denotes the weight of the attenuated loss and *thresh* denotes the threshold information.

Eventually, the loss functions of the 2 approaches are summed up as shown in Eq. (11):

$$L = L_1 + L_2 \tag{11}$$

3.2. Semi-supervised Training Strategies Incorporating Confidence Filtering Mechanisms

To solve overfitting and poor generalization from over-reliance on labeled data, this paper proposes a semi-supervised training strategy with confidence filtering. It uses data augmentation [17] (back-translation, synonym replacement) to get unlabeled data, filters for valuable pseudo-labels via confidence scores, and trains jointly with labeled data—addressing data scarcity, high labeling costs, and long tails to boost robustness and practicality. High-confidence labels are filtered for training, while low-confidence ones are randomly sampled via negative sampling to retain correct low-confidence entities. Random sampling of prediction errors adds noise to prevent overfitting. As Fig. 3 shows, this strategy lets the model learn high-quality labels in the supervised part [18] and more same-entity features via filtered labels in the unsupervised part, boosting generalization and robustness.

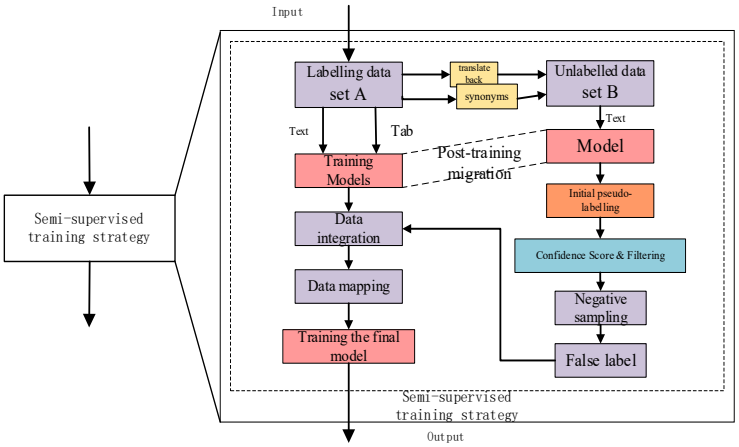

Fig. 3 Training Strategy Diagram.

4. Experimental and Analysis

4.1 Data sets and Experimental parameter setting

Experiments were conducted on a system equipped with an Intel Xeon Gold 6240 CPU @ 2.60 GHz, NVIDIA Tesla T4 GPU, 16 GB RAM, running Ubuntu 18, and utilizing PyTorch 1.8.1 as the runtime framework. The study's power communication fault text corpus is mainly from a State Grid company's O&M department; over 800 faults are excerpted from raw data for annotation (covering equipment alarms, fault names, solutions, troubleshooting). The fault text corpus

was annotated using the brat annotation tool, which supports span-level manual annotation and structured entity labeling. In this study, the annotation work was carried out by a single annotator with relevant domain knowledge, based on a manually developed annotation guideline covering entity definitions, annotation boundaries, and category distinctions. To improve consistency, ambiguous cases were reviewed and discussed with the research team during the annotation process, and the guideline was iteratively refined when necessary. However, because the corpus was not independently annotated by multiple annotators, no inter-annotator agreement metric (e.g., Cohen's kappa) was computed. This should be considered a limitation when assessing the reliability and reproducibility of the dataset. Table 1 presents the statistics of entity types in the initial dataset.

Table 1

Entity Type	**Example Entities**	**Description**	**Count**
Device	Communication power supply, OTN equipment	Power communication devices	647
Method	checking battery air switch or fuse detection line	Fault inspection or troubleshooting methods	381
Name	antenna receiving level low fault	Fault names	585
Phenomenon	AC power outage, network element disconnected	Observable phenomena after faults occur	1229
Reason	municipal construction, internal power line tripping	Possible causes of faults	1164
Solve	modifying configuration parameters, replacing faulty boards	Solutions for fault handling	770

As shown in the table, there exists a significant imbalance in the distribution of different entity types. In particular, the number of Method entities is considerably lower than that of other categories, while Phenomenon and Reason entities appear much more frequently. This imbalance arises mainly because some fault texts are relatively short and do not contain Method-related information. To mitigate the impact of data imbalance during dataset splitting, Method entities were first separated and stored independently. Subsequently, the remaining labeled data were divided into supervised training, validation, and test sets using a 6:2:2 ratio, ensuring a more balanced distribution of low- frequency entities across different subsets. To better understand the limitations of the proposed method, we conduct a qualitative error analysis on the test set. More than 100 misclassified cases are manually inspected and categorized into three main types: boundary errors, entity type confusion, and nested entity errors. Representative error cases are summarized in Table 2.

Table 2

Representative error cases

Sentence excerpt	Gold entity (type)	Predicted entity (type)	Error type
Check whether the battery circuit breaker or fuse	battery circuit breaker or fuse detection line	battery circuit breaker (Method)	Boundary

detection line is abnormal	(Method)		
Municipal construction causes communication interruption	municipal construction (Reason)	communication interruption (Phenomenon)	Type confusion
Antenna receiving signal level low fault	antenna receiving signal level low (Phenomenon) + fault (Name)	antenna receiving signal level low fault (Name)	Nested error

4.2 Results

(1) Comparative tests

Four baseline models (Table 3) are compared to evaluate the proposed method. The BIO-annotated BiLSTM-CRF performs poorly on this dataset, while incorporating BERT significantly improves performance. Compared to the pointer-annotated GlobalPointer [21], the proposed method increases the three evaluation metrics by 1.94%, 2.57%, and 2.43%, respectively. Furthermore, compared to DSpERT [22], the proposed method improves the three evaluation metrics by 2.21%, 2.14%, and 2.07%, respectively.

Table 3

Indicators relative to other models.

Model	ACC **P (%)**	Recall **R (%)**	F1 **F (%)**
BiLSTM-CRF	55.38	61.4	58.2
BERT-BiLSTM-CRF	87.54	90.74	89.11
GlobalPointer	93.55	90.8	91.99
DSpERT	93.28	91.23	92.35
Our	95.49	93.37	94.42

(2) Analysis of the impact of different modules on the model

To verify the Boundary-Aware layer's effectiveness, the dataset tests the m odel with/without the differential operator (Table 4). Introducing the Boundary-A ware module substantially improves potential entity recognition, with F1 increasin g by 1.14%.

Table 4

Impact of Boundary-Aware layers on the model

Parameter	Accuracy **P** (%)	Recall **R** (%)	F1 **F** (%)
No differential operators	92.0	90.3	91.14
Difference operator	93.29	91.59	92.28

To verify entity length restriction's effectiveness, a length constraint mechanism is introduced during training, reducing predictions, computational complexity, and unrealistic entities in results. Fig. 4 shows the span matrix change after character restriction: the decoded matrix transformation significantly reduces predicted spans, with greater parameter reduction for longer texts.

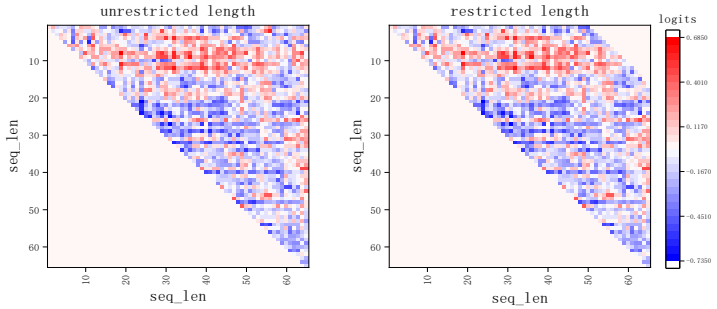

Fig. 4 Label matrix change before and after limiting length=50.

Statistical analysis shows that over 96% of annotated entities contain fewer than 45 characters, and less than 1% exceed 60 characters. Table 5 demonstrates the effect of different entity length restrictions:

Table 5

Effect of different restriction lengths on the model.

Parameter **Limit**	Accuracy **P (%)**	Recall **R (%)**	F1 **F (%)**
None	**93.29%**	91.59%	92.28%
45	92.91%	91.87%	92.25%
50	92.81%	92.3%	92.48%
55	91.93%	**93.36%**	**92.51%**
60	92.37%	92.51%	92.44%

Experiments with varying length restrictions showed the model achieved optimal overall performance at the 55-character restriction, with the highest recall—indicating more correct entities predicted—thus demonstrating the effectiveness of entity length limitation.

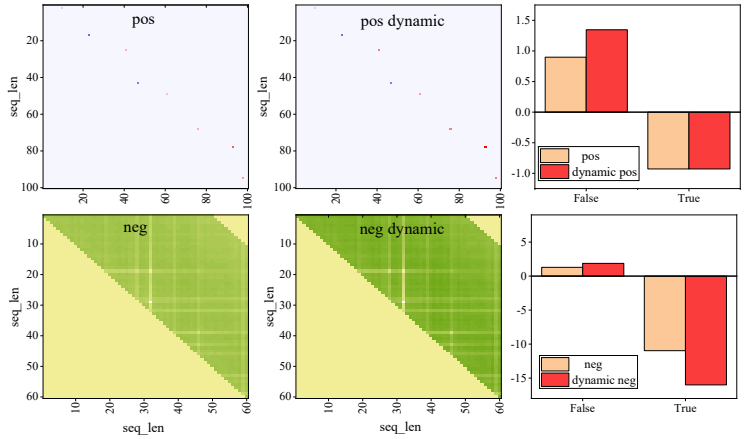

Fig. 5. Change in value of losses

Fig. 5 shows loss changes after masking neg/pos sets and categorized loss averages. The pos set's incorrectly predicted part has higher loss, while the correct predictions show lower loss—overall, the model focuses more on mispredicted parts.

(3) The Impact of Loss Functions on the Model

To verify the effectiveness of the dynamic weight proposed in this paper, we compare the dynamic weight with static fixed weights and three loss weight increment strategies (linear, exponential, logarithmic), as presented in Table 6.

Table 6

Impact of the three additions on the model

parameter	increase method	F1 **F (%)**	increase method	F1 **F (%)**	increase method	F1 **F (%)**
1-1.2	Linear	92.43	exp	92.46	log	92.14
1-1.4		**92.96**		92.54		92.09
1-1.6		92.26		**92.94**		**92.60**
1-1.8		92.93		92.06		92.49
1-2.0		92.57		92.14		92.30

Fig. 6 presents the comparison between loss values and F1-score before and after adjusting the loss function. The results demonstrate that the overall F1-score with dynamic weights is superior to the counterpart without dynamic weights.

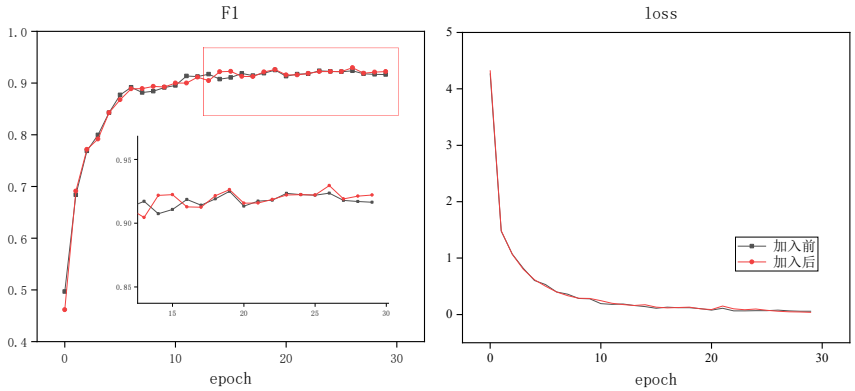

Fig. 6 Comparison of Loss Values and F1-Scores

(4) Supervised and semi-supervised result analysis

Pseudo-labels with 0–2 confidence are randomly sampled at 0.65 ratio and input with other pseudo-labels [19]; metrics are shown in Fig. 7 (upper: training passes vs. indicators; lower: average per 5 epochs vs. indicators). The semi-supervised model shows significantly higher F1, P, and R—due to more entities (including previously rare ones with more learnable features)—with higher and more stable performance than before.

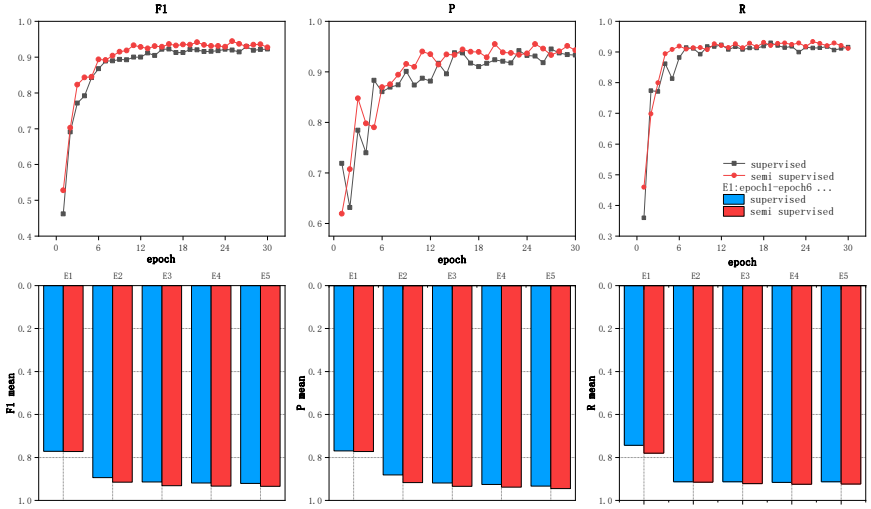

Fig. 7. Comparison of Indicators

Table 7 demonstrates the three metrics for each entity type after the addition of semi-supervision, with both the Method and Reason entities showing improvements of 2.88% and 0.8%, respectively, in their F1 values.

Table 7

Entities and all indicators.

Entity	ACC P (%)	R (%)	F1 F (%)
Device	96.44	94.20	95.31
Method	81.56	83.11	82.33
Name	92.97	93.86	94.77
Phenomenon	92.24	92.13	92.02
Reason	91.71	85.90	88.71
Solve	96.97	96.77	96.57
ALL	95.49	93.37	94.42

To ensure statistical rigor, we repeated experiments for the proposed method and the strongest baseline (GlobalPointer) using three different random seeds (42, 2023, and 2024). The F1-score is reported as mean ± standard deviation. A paired t-test confirms the improvement is statistically significant ($p < 0.01$).

Table 8

Raw F1 results across seeds

Seed	GlobalPointer F1	Our F1	ΔF1
42	91.87	94.21	+2.34
2023	92.05	94.46	+2.41
2024	92.01	94.59	+2.58

As summarized in Table 8 our method achieves an average F1-score of 94.42 ± 0.19 across three random seeds, significantly outperforming GlobalPointer, which

obtains 91.98±0.09. The improvement is statistically significant as confirmed by a paired t-test ($t = 34.3$, $p < 0.01$).

We further investigate the behavior of the semi-supervised component to better understand the source of the observed performance gains. The results indicate that incorporating unlabeled data consistently improves performance, with more pronounced gains in low-resource settings and diminishing returns as the amount of labeled data increases. Higher confidence thresholds produce more accurate pseudo-labels, albeit at the expense of reduced sample size. Furthermore, the combination of back-translation and synonym replacement achieves the best overall performance.

5. Conclusion

In summary, this paper proposes a boundary-aware, span-based NER framework for power fault texts, integrating length masking, dynamic weighted loss, and semi-supervised learning. Extensive experiments demonstrate that the proposed method achieves consistent and significant improvements over strong baselines within the studied power fault domain, thereby validating the effectiveness of each component.

However, the current evaluation is limited to a single domain-specific dataset, and cross-domain generalization as well as performance under extremely low-resource settings have not yet been empirically validated. Extending the proposed framework to multi-domain fault corpora and low-resource transfer scenarios constitutes an important direction for future work.

Acknowledgements:

This work was supported by Electric Power Research Institute of Guangxi Power Grid Co., Ltd. Technology Project (Project Number: GXKJXM20240094).

REFERENCES

[1] *Zhang Y, Pan X, Gao X,* et al. Application of Power Communication Technology in Smart Grid. Applications of IC, 2023, 40 (10): 370-371.

[2] *Guo Q, Zhuang F, Qin C*, et al., A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(8): 3549-3568.

[3] *Jiao K, Li X, Zhu R*. Overview of Chinese Domain Named Entity Recognition. Computer Engineering and Applications, 2021, 57(16): 1-15.

[4] *Li J, Sun A, Han J,* et al. A survey on deep learning for named entity recognition. IEEE transactions on knowledge and data engineering, 2020, 34(1): 50-70.

[5] *Qiu Q, Tian M, Huang Z,* et al., Chinese engineering geological named entity recognition by fusing multi-features and data enhancement using deep learning. Expert Systems with Applications, 2024, 238: 121925.

[6] *Fan H, Chen J, Gao J,* et al., Named Entity Recognition Based on BiLSTM-CRF Discipline Inspection and Supervision Events. Computer Simulation, 2022, 39(06): 464-468.

[7] *Ke J, Wang W, Chen X,* et al., Medical entity recognition and knowledge map relationship analysis of Chinese EMRs based on improved BiLSTM-CRF. Computers and Electrical

Engineering, 2023, 108: 108709.
[8] *Luo L, Yang Z, Yang P,* et al. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. Bioinformatics, 2018, 34(8): 1381-1388.
[9] *Bhumireddypalli V S R, Koppula S R, Koppula N*. Enhanced conditional random field-long short-term memory for name entity recognition in English texts. Concurrency and Computation: Practice and Experience, 2023, 35(9): e7640.
[10] *He Z, Wang L, Sheng T*, et al. MLSAN: Mixed-Lattice Self-Attention Network for Chinese Named Entity Recognition, 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022: 1436-1442.
[11] *Johnson S J, Murty M R, Navakanth I.* A detailed review on word embedding techniques with emphasis on word2vec. Multimedia Tools and Applications, 2023: 1-29.
[12] *Deepa D, Tamilarasi A*. Bidirectional encoder representations from transformers (BERT) language model for sentiment analysis task. Turkish Journal of Computer and Mathematics Education, 2021, 12(7): 1708-1721.
[13] *Su J, Ahmed M, Lu Y*, et al. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 2024, 568: 127063.
[14] *Thabtah F, Hammoud S, Kamalov F*, et al. Data imbalance in classification: Experimental evaluation. Information Sciences, 2020, 513: 429-441.
[15] *Dembczynski K, Jachnik A, Kotlowski W,* et al. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization, International conference on machine learning. PMLR, 2013: 1130-1138.
[16] *Zhang M, Li J.* A commentary of GPT-3 in MIT Technology Review 2021. Fundamental Research, 2021, 1(6): 831-833.
[17] *Sun Y, Wang S, Li Y,* et al. Ernie 2.0: A continual pre-training framework for language understanding, Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 8968-8975.
[18] *Koutsikakis J, Chalkidis I, Malakasiotis P,* et al. Greek-bert: The greeks visiting sesame street, 11th Hellenic conference on artificial intelligence. 2020: 110-117.
[19] *Chen H, Zhang W, Cheng L*, et al. Diverse and High-Quality Data Augmentation Using GPT for Named Entity Recognition, International Conference on Neural Information Processing. Singapore: Springer Nature Singapore, 2022: 272-283.
[20] *Song B, Li F, Liu Y,* et al. Deep learning methods for biomedical named entity recognition: a survey and qualitative comparison[J]. Briefings in Bioinformatics, 2021, 22(6): 1-18.
[21] *Zhai Z, Fan R, Huang J,* et al. A Named Entity Recognition Method Based on Knowledge Distillation and Efficient GlobalPointer for Chinese Medical Texts[J]. IEEE Access, 2024, 12: 83563-83574.
[22] *Zhu E, Liu Y, Li J* Deep span representations for named entity recognition. In: ACL, 2023, pp. 10565–10582.