mathematics

Article

KCB-FLAT: Enhancing Chinese Named Entity Recognition with Syntactic Information and Boundary Smoothing Techniques

Zhenrong Deng [1,2], Zheng Huang [1], Shiwei Wei [3,*] and Jinglin Zhang [1]

[1] Guangxi Key Laboratory of Images and Graphics Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China; zhrdeng@guet.edu.cn (Z.D.); 22032303154@mails.guet.edu.cn (Z.H.); 22032303126@mails.guet.edu.cn (J.Z.)
[2] Nanning Research Institute, Guilin University of Electronic Technology, Guilin 541004, China
[3] School of Computer Science and Engineering, Guilin University of Aerospace Technology, Guilin 541004, China
* Correspondence: swwei@guat.edu.cn

Abstract: Named entity recognition (NER) is a fundamental task in Natural Language Processing (NLP). During the training process, NER models suffer from over-confidence, and especially for the Chinese NER task, it involves word segmentation and introduces erroneous entity boundary segmentation, exacerbating over-confidence and reducing the model's overall performance. These issues limit further enhancement of NER models. To tackle these problems, we proposes a new model named KCB-FLAT, designed to enhance Chinese NER performance by integrating enriched semantic information with the word-Boundary Smoothing technique. Particularly, we first extract various types of syntactic data and utilize a network named Key-Value Memory Network, based on syntactic information to functionalize this, integrating it through an attention mechanism to generate syntactic feature embeddings for Chinese characters. Subsequently, we employed an encoder named Cross-Transformer to thoroughly combine syntactic and lexical information to address the entity boundary segmentation errors caused by lexical information. Finally, we introduce a Boundary Smoothing module, combined with a regularity-conscious function, to capture the internal regularity of per entity, reducing the model's overconfidence in entity probabilities through smoothing. Experimental results demonstrate that the proposed model achieves exceptional performance on the MSRA, Resume, Weibo, and self-built ZJ datasets, as verified by the F1 score.

Keywords: named entity recognition; Chinese NER; syntactic information; word-boundary smoothing

MSC: 68T99

check for updates

Citation: Deng, Z.; Huang, Z.; Wei, S.; Zhang, J. KCB-FLAT: Enhancing Chinese Named Entity Recognition with Syntactic Information and Boundary Smoothing Techniques. *Mathematics* **2024**, *12*, 2714. https://doi.org/10.3390/math12172714

Academic Editor: Jonathan Blackledge

Received: 2 August 2024
Revised: 27 August 2024
Accepted: 27 August 2024
Published: 30 August 2024

1. Introduction

Recently, the study of named entity recognition (NER) in Chinese has become a significant area of interest in natural language processing research. Due to the swift advancement of big data analytics and artificial intelligence technology, the precise recognition and categorization of named entities from vast amounts of textual information has gained significant importance. As smart mobile devices gain popularity, the Internet user base in developing nations is experiencing a surge. Recently, on 28 August 2023, the China Internet Network Information Centre unveiled its 52nd Statistical Report on China's Internet Development Status in Beijing. The report reveals that by June 2023, China boasted 1.079 billion Internet users, marking an escalation of 11.09 million since December 2022. Notably, the Internet penetration stood at 76.4%. The growing access to information, content searches, and online interactions has resulted in a massive accumulation of Chinese text data. In the realm of Chinese NLP, NER is a foundational and vital task, playing a pivotal role in diverse downstream applications including one-question-one-answer dialogue system [1], intelligent personalized recommendation [2], text generation [3], and content understanding [4,5]. An important task of Chinese NER is to identify all possible entities from a text and classify them based on their attributes, such as names, places, institutions, and other similar groups [6]. However, Chinese NER presents unique challenges compared with its English counterpart. First of all, Chinese NER has been developed relatively late, and there are fewer available corpora, which limits its development [7,8]. Secondly, Chinese does not have a corpus like English [9]. Furthermore, Chinese does not have obvious word spacing like English [10]. The fact that Chinese has no obvious word spacing like English makes word segmentation a necessary preprocessing step. Moreover, the flexible nature of Chinese word formation allows the same character sequences to be segmented differently depending on the context, adding to the complexity of NER.

To accomplish the objective of Chinese NER, the model undergoes learning by analyzing a substantial volume of annotated training data and fine-tunes its parameters through a likelihood function aimed at optimizing the fit to the training dataset. However, over-confidence is a potential problem in Chinese NER research. If there are specific patterns in the training data, the model may over-learn these patterns, resulting in over-confidence in these patterns when making predictions [11]. This situation may degrade the performance and calibration of the model, as the model may be too biased towards predicting specific entity classes while ignoring other situations. In other words, it is possible that the model does not really understand the concepts or syntax behind these patterns, which is akin to memorizing the answers rather than really understanding the meaning of the question. In addition, some studies have identified incorrect entity boundaries as one of the main reasons for incorrect entity identification [12]. Therefore, addressing the problem of over-confidence in NER models is crucial to improving their performance and usefulness.

Empirical evidence shows that positive samples (i.e., real entities) are sparsely distributed across the candidate ranges in a dataset. For example, in the Resume dataset, entities make up only 11% of the candidate ranges. To address this, assigning explicit probabilities to the adjacent ranges helps reduce overconfidence by blurring the boundaries, preventing the model from concentrating all its probability mass on areas with few or no positive samples. For example, in the Chinese context, “莲花清瘟胶囊” (“Lotus Clear Fever capsule”) and “莲花清瘟” (“Lotus Clear Fever”) are not fully distinguishable and refer to the same medicine. In previous NER work, the model may not recognize an example like “莲花清瘟”as an entity. With the introduction of the Boundary Smoothing (BS) module, the model will assign the probability of belonging to the “Lotus Clear Fever capsule” to the entity boundary, thus alleviating the problem of over-dependence on the probability of some entities in the model, allowing the model to recognize “Lotus Clear Fever “as an entity in the learning process. In response to this issue, inspired by Zhu et al. [13], this project proposes the KCB-FLAT model based on the complex semantic structure of the Chinese language and our previous work [14]. This model is built on the Flat-Lattice Transformer (FLAT) [15] architecture, chosen for its lightweight design and parallel computing capabilities. The KCB-FLAT model enhances this foundation by extracting three distinct types of syntactic data along with their contextual features. These features are encoded using a Key-Value Memory Network (KVMN) and are subsequently fused through an attention mechanism. The model then integrates lexical and syntactic information via a Cross-Transformer (CT). Additionally, the Boundary Smoothing module, operating in synergy with an internal regularity perception function, further boosts the model’s capability to precisely detect named entities. This combination explores the internal consistency of each entity and applies smoothing to refine the entity boundaries.

In summary, incorporating Boundary Smoothing (BS) can assist NER models in tackling the issue of overconfidence. Through the redistribution of probabilities in the vicinity of labeled entities, BS enhances the model’s prediction caution and diminishes excessive confidence on specific boundaries. This adjustment facilitates the model’s adaptability to novel domains and evolving data distributions, ultimately elevating the precision and reliability of named entity recognition. Our specific contributions are outlined below:

1. This article proposes a method KCB-FLAT that achieves good results on four Chinese datasets: Weibo, MSRA, Resume, and a self-built dataset ZJ.
2. We combine the smoothing process with syntactic information extraction, which discretizes the entity probabilities to the entity boundaries, effectively alleviating the problem of over-confidence of the model in entity labelling.
3. We control the smoothing factor on the Boundary Smoothing method to control the degree to which the model differentiates the probability of entities and analyze how these adjustments impact overall model performance.
4. We conducted experiments on our self-built dataset, ZJ, by integrating it, and the results show that our model is well-suited for Chinese named entity recognition tasks within the judicial domain.

Through experiments with the KCB-FLAT model, it was observed that the model significantly boosts performance in complex and fuzzy boundary cases. This integration method not only improves the model's ability to process Chinese text, but also enhances its ability to respond to specific problems.

2. Related Work

2.1. Current Status of Named Entity Identification

Named entity recognition (NER) has been a long-standing and elemental research topic within the realm of natural language processing. Traditional methods can generally be divided into two main categories: token-based and span-based approaches. Token-based methods (e.g., Tourani et al., 2024 [16]; Wang et al., 2020 [17]) typically perform sequence labelling at the token level, which is then converted into span-level predictions. Meanwhile, span-based methods (e.g., Sohrab and Miwa 2018, [18]; Eberts and Ulges, 2020 [19]; Shen et al., 2021 [20]; Li et al., 2022 [21]) directly perform entity classification of potential spans for prediction. In addition to this, some approaches attempt to formalize NER as a sequence to set (Tan et al., 2021 [22]; Ma et al., 2016 [23]), and even more methods attempt reading comprehension (Li et al., 2019 [24]; Yu et al., 2020 [25]; Yu et al., 2019 [26]) tasks for prediction. In addition, some have used autoregressive generative NER methods (e.g., Yang et al., 2019 [27]; Athiwaratkun et al., 2020 [28]), i.e., relying on sequence-to-sequence language models (e.g., BART [29], Sentence-T5 [30] etc.) that linearize structured named entities into sequences to decode entities. Recent breakthroughs in the realm of text classification, exemplified by the refinement of feature selection via an improved Discrete Egg Laying Chicken Algorithm (DELCA) [31] have yielded substantial improvements not only in classification efficacy but also in the domain of named entity recognition (NER). Enhanced precision in feature extraction has been instrumental in bolstering the accuracy of NER systems' recognition and categorization capabilities. These works have designed various translation models to unify NER to text generation tasks and have achieved good performance and generalization.

2.2. Current Status of Syntactic Information Extraction in NLP

Syntactic information extraction work usually includes several categories such as dependent syntactic analysis, phrase structure syntactic analysis, and syntactic information extraction and semantic annotation. The focus of dependency syntactic analysis lies in determining the dependency relationships among words within a sentence, and resolving these relationships aids in comprehending the sentence's structure and meaning, as exemplified in Liang et al., 2020. [32], which utilizes the idea of graph-based dependency syntactic analysis by introducing a bi-affine model that provides the model with a global view of the inputs, thus enabling accurate prediction of named entities. Phrase structure syntactic analysis strives to ascertain the structure and arrangement of phrases within a sentence, enabling a deeper comprehension of the sentence's syntactic makeup and semantic significance. Lou et al.'s study, 2022 [33], explored the problem of nested named entity recognition and proposed a span-based syntactic analyzer to deal with nested NERs, while exploiting the lexicalized organizational tree structure of headword annotation and a new training strategy. Syntactic information extraction and semantic annotation entail assigning labels to each predicate in a sentence, indicating its corresponding semantic role, thereby facilitating a more precise comprehension of the sentence meaning and enabling inference of semantic information (e.g., Ma et al., 2022 [34]; Nie et al., 2020 [35]). By utilizing these syntactic information extraction methods, the performance and efficacy of the NLP models in different tasks can be enhanced, making them more suitable for practical application in real-world scenarios.

2.3. Application of Smoothing Techniques in Model Performance Enhancement

Boundary smoothing technology plays an important role in NLP tasks, machine learning, and deep learning. It helps to handle zero probability problems, improve model generalization ability, optimize the training process, enhance model robustness, and promote model interpretability. In sequence annotation tasks such as NER, the use of smoothing techniques has significantly improved model performance. These techniques typically reduce over-confidence in individual labels by adjusting the probability distribution of the model's output, thereby bolstering the capacity to manage boundary ambiguity and uncertainty.

Label smoothing [36], for instance, is a widely used technique in deep learning that modifies the probability distribution by assigning weights to both the true labels and a uniform distribution. It has been demonstrated that this approach enhances the generalization performance of models in tasks like image recognition and text classification [37–41]. In the context of the NER task, label smoothing proves effective in mitigating the issue of category imbalance, particularly in datasets where coronal and modifier descriptors of entities are highly frequent.

Additionally, the probabilistic output of the model can be refined through the introduction of a context-driven smoothing approach. As an illustration, the Conditional Random Field (CRF) layer [42], often employed in NER tasks, considers inter-label dependencies. By establishing a globally normalized probability distribution, the CRF layer aids the model in making more precise predictions across the entire sequence. Nevertheless, it is important to note that the CRF does not directly address the smoothing of the initial probability distribution, a distinctive feature offered by the Boundary Smoothing module.

Hence, we propose the Boundary Smoothing module to assess its efficacy and practicality. This module enhances the model's resilience and precision when handling boundary vagueness and contextual intricacies by refining the model's output probabilities. We anticipate that incorporating the smoothed boundary module into the Baseline model will lead to substantial performance improvements in NER tasks, particularly in pinpointing and categorizing entity boundaries.

3. Methodology

3.1. The Overview of the KCB-FLAT

The KCB-FLAT consists of three modules: (1) Key-Value Memory Network (KVMN), which is used to encode syntactic information and contextual features; (2) Cross-Transformer module encoding two different types of information, including lexical information and syntactic information; (3) Boundary smoothing module to study internal regularities. The following chart provides a visual illustration of this structure (Figure 1).

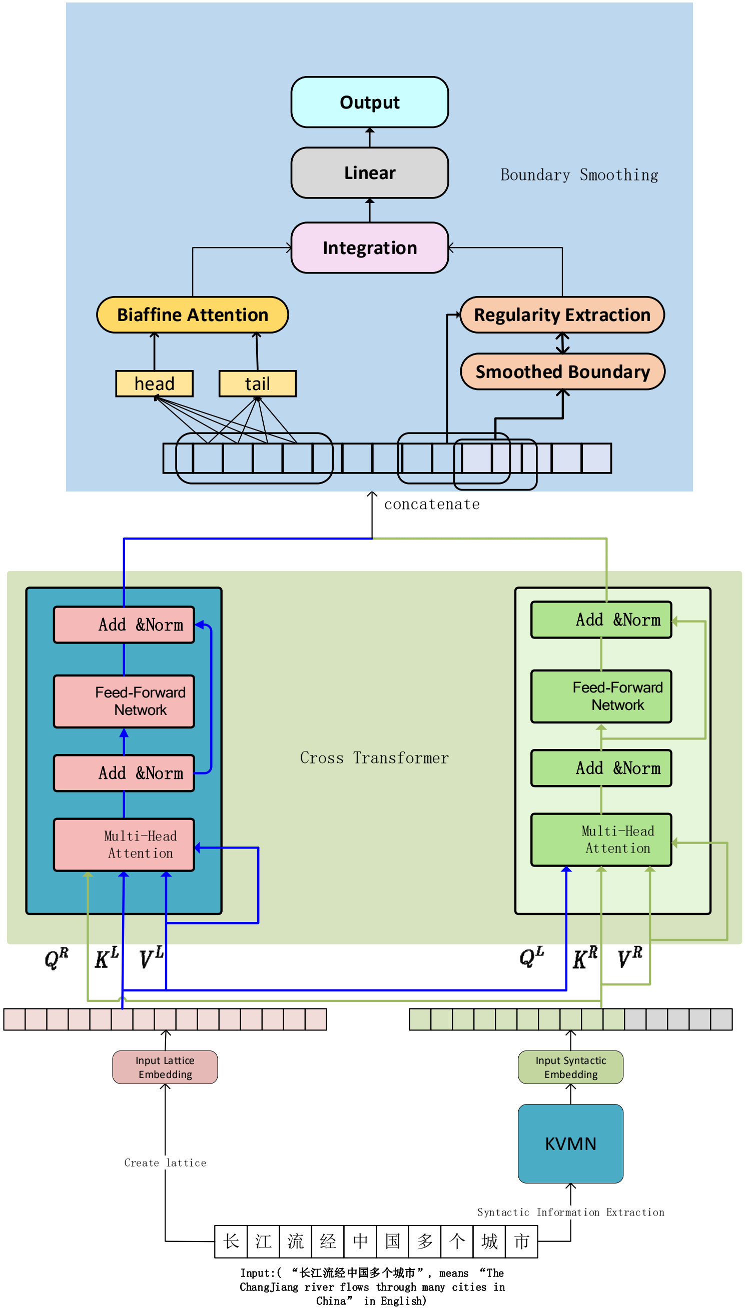

Figure 1. The general structure of KCB-FLAT.

3.2. KVMN for Syntactic Embedding

In the Chinese NER task, accurately identifying entity boundaries is challenging due to the varied meanings Chinese characters can have in different contexts and the lack of clear separation between them. To overcome this challenge, we utilize a character-level NER approach and integrate lexical information as supplementary data, aiming to enrich the input and improve the accuracy of entity boundary recognition. For instance, in the sentence “长江流经中国多个城市” (The Changjiang River flows through multiple cities in China), the ambiguity associated with the character “江” can be resolved through syntactic analysis. By distinguishing between “江流” (flow) and “长江” (The Changjiang River), we can more accurately identify entities such as “长江” (The Changjiang River) and “城市” (city). In this paper, we introduce syntactic information to clarify ambiguities by encoding diverse contextual features along with their corresponding syntactic data, such as POS tags, syntactic constituents, and dependencies. To enhance the model’s accuracy in recognizing named entities within various contexts, we incorporated KVMN. Figure 2 depicts the process of extracting syntactic information.

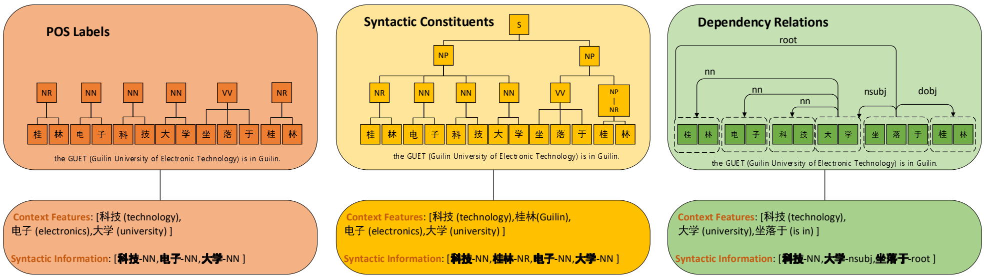
POS Labels
NR
NN
NN
NN
VV
NR
桂 林 电 子 科 技 大 学 坐 落 于 桂 林
the GUET (Guilin University of Electronic Technology) is in Guilin.
Syntactic Constituents
S
NP
NP
NR
NN
NN
NN
VV
NP
NR
桂 林 电 子 科 技 大 学 坐 落 于 桂 林
the GUET (Guilin University of Electronic Technology) is in Guilin.
Dependency Relations
root
nn
nn
nn
nsubj
dobj
桂 林 电 子 科 技 大 学 坐 落 于 桂 林
the GUET (Guilin University of Electronic Technology) is in Guilin.
Context Features: [科技 (technology), 电子 (electronics),大学 (university)]
Syntactic Information: [科技-NN,电子-NN,大学-NN]
Context Features: [科技 (technology),桂林(Guilin), 电子 (electronics),大学 (university)]
Syntactic Information: [科技-NN,桂林-NR,电子-NN,大学-NN]
Context Features: [科技 (technology), 大学 (university),坐落于 (is in)]
Syntactic Information: [科技-NN,大学-nsubj,坐落于-root]

Figure 2. The process of extracting syntactic information: POS Labels, syntactic constituents, and 211 dependencies of “科” (science) in the lexical labels.

In this study, we used Key-Value Pair Storage Network (KVMN) to encode different syntactic information and introduced attention mechanisms into the sentence level vector expression of Chinese characters. By encoding syntactic information with the KVMN and extracting useful features, we then applied an attention mechanism to obtain syntactic-level feature embeddings for Chinese characters. We integrated an attention mechanism to automatically assign weights to different syntactic information. This allowed our model to dynamically adjust and focus on various syntactic data during both the encoding and decoding stages, effectively highlighting essential information while downplaying less critical details. This approach does not necessitate pre-determined weights, as it allows the model to autonomously learn weight allocation strategies during the training process.

Figure 3 illustrates this process, using the example sentence “桂林电子科技大学坐落于桂林”, which means the GUET (Guilin University of Electronic Technology) is in Guilin.

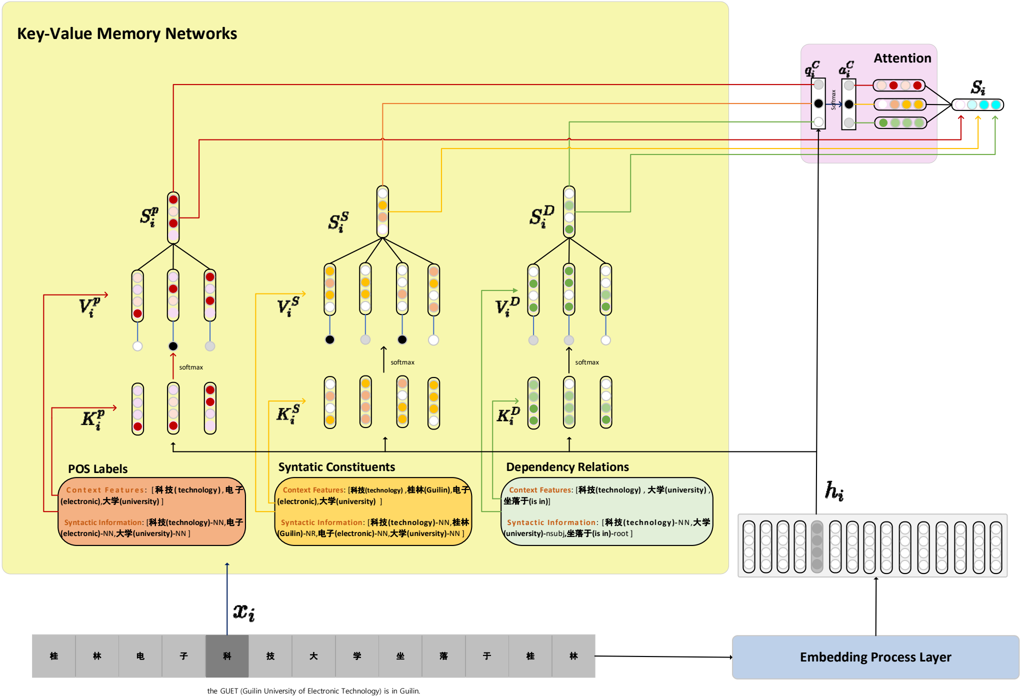
Key-Value Memory Networks
Attention
q_i^C
a_i^C
softmax
S_i
S_i^P
S_i^S
S_i^D
V_i^P
V_i^S
V_i^D
softmax
softmax
softmax
K_i^P
K_i^S
K_i^D
h_i
POS Labels
Context Features: [科技(technology),电子(electronic),大学(university)]
Syntactic Information: [科技(technology)-NN,电子(electronic)-NN,大学(university)-NN]
Syntatic Constituents
Context Features: [科技(technology) ,桂林(Guilin),电子(electronic),大学(university)]
Syntactic Information: [科技(technology)-NN,桂林(Guilin)-NR,电子(electronic)-NN,大学(university)-NN]
Dependency Relations
Context Features: [科技(technology) , 大学(university) , 坐落于(is in)]
Syntactic Information: [科技(technology)-NN,大学(university)-nsubj,坐落于(is in)-root]
x_i
桂 林 电 子 科 技 大 学 坐 落 于 桂 林
the GUET (Guilin University of Electronic Technology) is in Guilin.
Embedding Process Layer

Figure 3. Encoding Syntactic Information with KVMN.

While parsing the input sequence X, every character x_i within X belongs to a word that serves as the focal point for mapping contextual features and relevant information with syntax into the collection of keys and values. These are denoted as $K_i^c = [k_{i,1}^c, \ldots, k_{i,j}^c \ldots,$

k^c_{i,m_i}] and V^c_i = [$v^c_{i,1}, \ldots, v^c_{i,j} \ldots, v^c_{i,m_i}$], where $c \in C = \{P, S, D\}$. Here, P, S, D represent three different syntactic types, respectively, and m_i denotes the count of text context features of information with syntax type c for x_i. The $k^c_{i,j}$ represents the jth contextual feature of syntactic type c, while $v^c_{i,j}$ is the syntactic information corresponding to $k^c_{i,j}$. These keys and values, k^c_i and v^c_i, are mapped into matrix forms, denoted as $e^{k_c}_{i,j}$ and $v^{k_c}_{i,j}$. The syntactic information is computed by the following formula:

$$p^c_{i,j} = \frac{exp\left(h_i \cdot e^{k_c}_{i,j}\right)}{\sum_{j=1}^{m_i} exp\left(h_i \cdot e^{k_c}_{i,j}\right)} \tag{1}$$

we use $p^c_{i,j}$ to express the weighting of syntactic information. Chinese, a pictographic script, differs from English in that the meaning of its grammatical words relies on individual characters. Moreover, within a given word, every character carries an equal weight of syntactic information. For instance, in the word "大学" (The University), both "大" and "学" are encoded by the same vector representing their shared syntactic information. To differentiate between the representations of various characters within the same word h_i is introduced, which is obtained by the following equation:

$$h_i = E(x_i) \tag{2}$$

The weights $p^c_{i,j}$ are applied to the corresponding syntactic data $v^{k_c}_{i,j}$, as calculated by the formula:

$$s^c_i = \sum_{j=1}^{m_i} p^c_{i,j} e^{v_c}_{i,j} \tag{3}$$

The output of the KVMN, denoted as s^c_i, encapsulates the weighted syntactic information of type c. This approach ensures that syntactic data are weighted and encoded in alignment with its contextual features, thereby optimizing the use of the most pertinent information.

Firstly, we perform syntactic analysis on the input to obtain a syntactic information vector s^c_i. Then, the three types of syntactic data are processed uniformly to give each syntactic information a unique weight. Attention mechanism helps with this integration. The weights of the q^c_i are calculated as follows:

$$q^c_i = \sigma\left(W^c_q \cdot (h_i \oplus s^c_i) + b^c_q\right) \tag{4}$$

In this equation, W represents the trainable weight vector, b^c_q represents the bias for implementing residual linking, $\oplus$ represents the concatenation, and σ is the sigmoid function. Next, we use the $softmax$ function to calculate the attention of the input syntactic information a^c_i. The attention score is calculated according to the following formula:

$$a^c_i = \frac{exp\left(q^c_i\right)}{\sum\limits_{c \in C} exp\left(q^c_i\right)} \tag{5}$$

Here, a^c_i represents the attention for syntactic information of type c . Finally, the attention mechanism utilizes different features through calculated weights, effectively resolving the inherent contradictions between syntax and syntax. Different types of syntactic data are combined to form s_i:

$$s_i = \sum_{c \in C} a^c_i s^c_i \tag{6}$$

Rather than simply concatenating three types of syntactic information, then the attention mechanism selectively emphasizes the most relevant features, resolving conflicts among the different syntactic data types. As a result, the various syntactic data are selectively encoded and combined.

3.3. Cross Transformer for Semantic Fusion

Vocabulary information can more readily identify local details like word positions and boundaries because it concentrates on the vocabulary and the relationships among its constituent characters. This focus enhances the sensitivity of vocabulary information to the internal structures and characteristics of words, allowing for more precise determination of word boundaries and positions. In contrast, syntactic information emphasizes the overall structure of sentences, considering the combinations and relationships among words and their functions and roles within sentences. Consequently, syntactic information is more concerned with sentence-level features and constraints than with the internal characteristics of individual words. This gives syntactic information a distinct advantage in delineating sentence structures and relationships, rectifying segmentation errors introduced by vocabulary-based information, and offering a more comprehensive sentence comprehension. To fully capitalize on the benefits of syntactic information, the model employs syntactic constraints to rectify segmentation errors arising from vocabulary-based information. After acquiring the embedded representations of syntactic information, in this study we utilize a Cross-Transformer to synthesize lexical and syntactic data. The Cross-Transformer network effectively fuses syntactic and lexical information through a self-attention mechanism and feed-forward network, augmented by residual connections and normalization, as illustrated in Figure 4 for the specific structure of the Cross-Transformer.

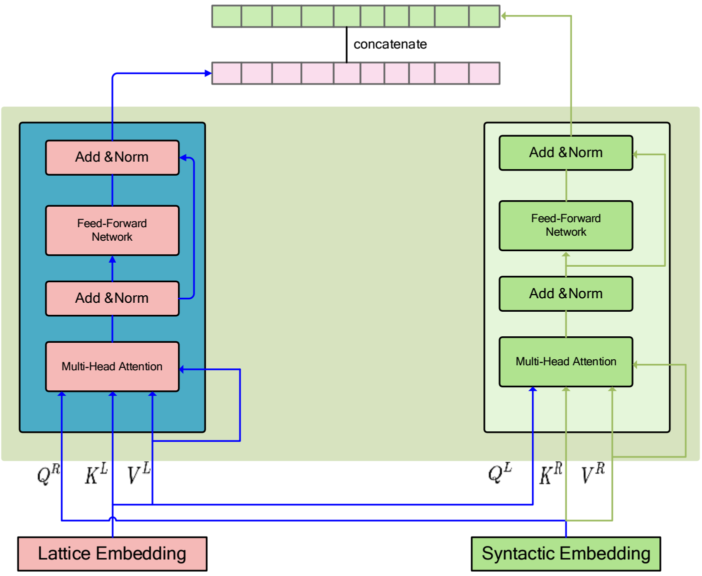

Figure 4. The Cross-Transformer module.

The left Transformer encoder receives an input denoted as (Q_x^L, K_x^L, V_x^L), which is derived from the lattice embedding containing lexical information via a linear transformation. The query (Q), key (K), and value (V) matrices are specifically computed as follows:

$$\left[Q_x^L, K_x^L, V_x^L\right] = E_x^L\left[W_Q^L, W_K^L, W_V^L\right] \tag{7}$$

Here, E_x^L signifies the lattice embedding of input, where $E_x^L \in \{x_1, \ldots..x_i, \ldots.., x_N\}$, each x_i represents the lexical representation of input, to ensure that the length of input sequence X in the cross converter are consistent, we set the length of the sequence as N, and each

W^L is a parameter that can be autonomously learned. Similarly, for the right Transformer encoder, we have

$$\left[Q_S^R, K_S^R, V_S^R\right] = E_S^R\left[W_Q^R, W_K^R, W_V^R\right] \tag{8}$$

Among them, E_S^R represents the syntactic embedding, where $E_S^R \in \{s_1, \ldots, s_i\ , \ \ldots\ , s_N\}$, each s_i denotes a syntactic input. During this period, each W_Q^R value is a learnable parameter. The Cross-Transformer includes 2 encoders, each encoder includes a front network called feed-forward network (FFN), which is a special multi-level perceptron that can perform nonlinear transformations in semantic space and can process positional information, followed by two processes: residual connectivity and hierarchical normalization. The main function of self-attention is to extract semantic-level information from text. Calculate the attention according to following method:

$$Att(A, V) = softmax(A)V \tag{9}$$

$$A_{i,j} = \left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right) \tag{10}$$

where d_k denotes the size of the lattice. Based on syntactic and lexical data obtained from Equations (7) and (8), use relative position encoding in FLAT mode to perform the following operations:

$$Att^L = softmax\left(A^R\right)V^L \tag{11}$$

$$Att^R = softmax\left(A^L\right)V^R \tag{12}$$

In these equations, A^R represents the score of syntax, while A^L denotes the score of lattice attention. The calculation for A^R is according to the following method:

$$A_{i,j}^R = \left(Q_i^R + u^R\right)^T K_j^R + \left(Q_i^R + v^R\right)^T \left(R_{i,j}^R W_r^R\right) \tag{13}$$

where u^R and v^R are attention biases, W_r^R is the learnable parameter $R_{i,j}^R$ is the relative position coding, calculated as the following method:

$$R_{i,j}^R = ReLU\left(W_r\left(p_{h_i - h_j} \oplus p_{t_i - h_j} \oplus p_{h_i - t_j} \oplus p_{t_i - t_j}\right)\right) \tag{14}$$

Relative Position Coding, $R_{i,j}^R$, mitigates directional loss caused by the internal dot product of vectors. Each p represents the relativistic distance. Similarly, the calculation of Att^L and Att^R is generally the same, the specific calculation of Att^R can also be derived.

3.4. Boundary Smoothing Module

In NER tasks, two classic approaches exist. NER is treated as a sequence annotation task by one approach, in which each input character is labeled by category. The other divides the input into multiple spans, each a potential named entity, for recognition and classification. This chapter adopts the span method for handling named entities and uses the pattern perception function to capture the intrinsic features of each span.

In Chinese lexical formation and linguistic structure, specific naming patterns are evident. For instance, terms like "XX Hotel" and "XX Court" typically denote place names, while "XX Mountain" indicates a location. For example, "桂林电子科技大学花江校区" (Huajiang Campus of the University of Electronic Science and Technology of Guilin) refers to the full name "桂林电子科技大学花江校区" rather than separating "桂林" (Guilin) and "电子科技大学" (University of Electronic Science and technology) and "花江校区" (Huajiang Campus). It signifies the Huajiang campus of GUET, not the UESTC's campus in Guilin city.

The inherent complexity of implementing Boundary Smoothing (BS) is primarily due to its elevated computational demands. To mitigate this and efficiently identify naming patterns, we introduce a synergistic application of the Boundary Smoothing function complemented by the Regularity-Aware (RA) function, pioneered by Huawei Cloud in 2022 [43]. This RA function is adept at uncovering inherent regularities within data, concentrating computational efforts on the most critical boundary points. This strategic focus significantly reduces unnecessary computations and enhances overall performance. Furthermore, the integration of state-of-the-art optimization techniques ensures that our model maintains computational tractability, which means that our approach can remain both effective and feasible within the realms of computational resources available.

To prevent overfitting from the model learning the intrinsic rules of named entities, we use the Boundary Smoothing function proposed by Zhu et al. [13] This smooths the entity probability of spans, helping the model maintain calibration during training and ensuring that generated confidence better represents prediction accuracy.

In Figure 5, the regularity representation of each span is derived by enhancing the interaction between the head and tail in the Biaffine layer and incorporating a linear attention mechanism to capture the intrinsic regularity features extracted by the Cross-Transformer. The specific calculation method is as follows:

$$h_{s_{i,j}}^{rec} = \sum_{t=i}^{j} \alpha_t \cdot h_t \tag{15}$$

where h_t is the cascaded output from the Cross-Converter, and $t \in \{i, \ i+1, ., j\}$ represents index of the span $s_{i,j}$. α_t, means the attention score, and is calculated according to the following method:

$$\alpha_t = \frac{exp(a_t)}{\sum_{k=i}^{j} exp(a_k)} \tag{16}$$

$$a_t = W_{reg}^T h_t + b_{reg} \tag{17}$$

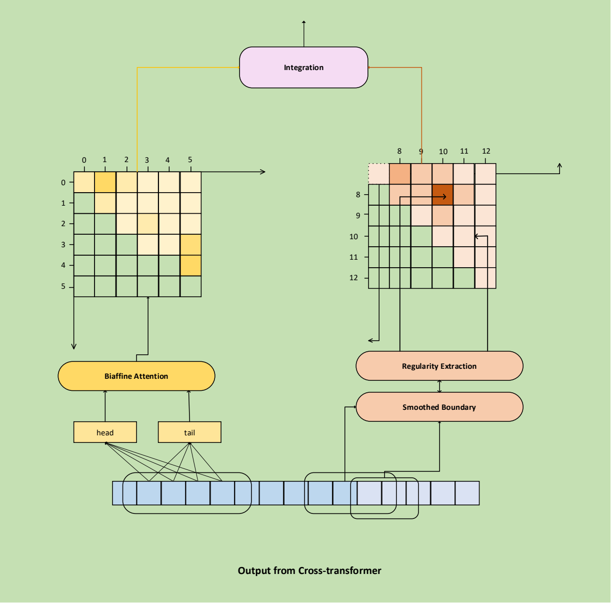

Figure 5. The Boundary Smoothing module.

Here, $W_{reg}^T \in \ R^{d \times 1}$ and $b_{reg} \in R^1$ represent the learnable weights and biases, respectively. To capture the interactions between head and tail features, this model utilizes a biaffine attention mechanism, thereby integrating regularities between entities into a span representation:

$$h_{s_{i,j}}^{(span)} = h_i^T u^{(1)h_j} + \left(h_i \oplus h_j\right) u^{(2)} + b_1 \tag{18}$$

Here, h_i represent the head of span $s_{i,j}$, h_j represent the tail, while $U^{(1)} \in R^{2d \times 2d \times 2d}$ and $U^{(2)} \in R^{4d \times 2d}$. The integration of regularity and span representations is performed through gate networks:

$$g_{s_{i,j}} = \sigma\left(u^{(3)}\left[h_{s_{i,j}}^{(span)}; h_{s_{i,j}}^{(reg)}\right] + b_2\right) \tag{19}$$

$$h_{s_{i,j}} = g_{s_{i,j}} \odot h_{s_{i,j}}^{(span)} + \left(1 - g_{s_{i,j}}\right) \odot h_{s_{i,j}}^{(reg)} \tag{20}$$

where b_2 is the deviation term, σ represents the sigmoid function, $U^{(3)} \in R^{4d \times 1}$ represents learnable weights, and $\odot$ refers to the multiplication of corresponding elements.

Next, we use a $softmax$ linear classifier to predict the entity classification for each span:

$$\tilde{y}_{s_{i,j}} = Softmax\left(W_{type}^T h_{s_{i,j}} + b_3\right) \tag{21}$$

where $W_{type}^T \in R^{2d \times c}$ represents a learnable weight through attention, b_3 is a bias. Then, the loss function we use is calculated by

$$L_{aware} = -\frac{1}{n} \sum_{n=1}^{N} \sum_{i=1}^{l} \sum_{j=1}^{l} y_{s_{i,j}}^{(n)} log\left(\tilde{y}_{s_{i,j}}^{(n)}\right), i \leq j \tag{22}$$

where $\tilde{y}_{s_{i,j}}^{(n)}$ is the predicted value of span, $y_{s_{i,j}}^{(n)}$ is the true value of span, and N is the number of all spans.

To perform smoothing, we use the Manhattan distance d to measure the distance between the surrounding span (span) and the specified original span (Ground-truth span) and to specify the smoothing range (Smoothing size) D ($D \in \{1, 2\}$). Where $d < D < N$, the distribution probability $\tilde{y}_{s_{i,j}}^{(n)'}$ is obtained by the following:

$$\tilde{y}_{s_{i,j}}^{(n)'} = (1 - \alpha\ ) * \tilde{y}_{s_{i,j}}^{(n)} + \alpha / D \tag{23}$$

where α is the smoothing factor. We consider the other spans (spans) that are not assigned to probabilities as non-entity, and the final loss is still expressed in terms of cross-entropy.

The KCB-FLAT model, while enhancing named entity recognition (NER) performance, introduces additional computational demands primarily due to the integration of Key-Value Memory Network (KVMN) and Cross-Transformer module. The KVMN, which encodes syntactic information and contextual features, operates with a time complexity proportional to the number of syntactic features and the dimensionality of the embedding space, approximately $O(m \times d)$, where m is the number of syntactic features and d is the embedding dimension. The Cross-Transformer module, responsible for fusing lexical and syntactic information, involves matrix multiplications that scale with the input size, contributing a complexity of $O\left(n^2 \times d\right)$ per layer, where n represents the input length. Additionally, the Boundary Smoothing module, which refines entity boundaries, has a complexity tied to the number of entities and the smoothing factor applied. Despite these factors, the model is designed with parallel computing capabilities, particularly in the Cross-Transformer module, which mitigates the overall computational load, making it feasible for practical applications while balancing performance and efficiency.

4. Experimental Design

4.1. Datasets and Evaluation Indicators

In this section, we will evaluate KCB-FLAT using four Chinese corpus datasets. To comprehensively evaluate the performance of the model, we use the span method, F1 score (F1), precision (P), and recall (R) as evaluation criteria. On this basis, we will gain a more precise understanding of the performance of the KCB-FLAT on different samples.

To validate our model, we utilized four Chinese NER datasets: MSRA, Resume, Weibo, and our custom dataset, named ZJ, which we manually curated and refined. The MSRA dataset, which was sourced from the 3rd International Chinese Language Processing Contest, primarily consisted of news data and is widely recognized for NER tasks. The Resume dataset encompassed a vast array of personal, place, and company names. Meanwhile, the Weibo dataset covered various Weibo contexts, broadening the application scope of the NER model. Specifically, the Weibo dataset classified entities into person, organization, location, and government wide entry point. The Resume dataset identified eight types of entities: country, education, profession, race, and title, while the MSRA dataset focuses on three main entity types: organization, person, and location.

Our ZJ dataset comprised criminal verdicts issued by the China Judgement Network (CJN) spanning from 2013 to 2019. These comprehensive texts include critical case details in the factual description section, marked by a higher concentration of entities and concise wording, amounting to over 410,000 words. Table 1 outlines our dataset statistics, with a 70–10–20 split for training, validation, and testing. The legal documents in this corpus were manually annotated using the {B, I, O} system. The labeled entities encompassed crime location, place name, person name, organization name, accusation, verdict, laws, and regulations, utilizing a total of 15 distinct labels tailored for various tasks.

Table 1. The ZJ dataset composition.

Entity Type	Amount of Entity	Number of Tags
Crime Location	3356	28,976
Place Name	3360	23,076
Person Name	5047	13,487
Organization Name	1667	6353
Accusation	1846	5837
Laws and regulations	1517	6203
Judgement	2086	7456
Total	19,079	91,348

In this paper, the evaluation index employed is the weighted summed average F1, which is computed using the formula:

$$F1 = \frac{2 * P * R}{P + R} \tag{24}$$

Here, P represents the accuracy rate, and R represents the recall rate; P and R are determined using the following equations:

$$P = \frac{TP}{TP + FP} \tag{25}$$

$$R = \frac{TP}{TP + FN} \tag{26}$$

Here, TP represents the number of accurately detected individuals, FP represents the number of misidentified individuals, and $TP + FP$ represent the total number of predicted entities. FN represents the number of lost entities, and $TP + FN$ represents the total number of real entities.

4.2. Parameter Settings

Our experiments were conducted using PyTorch 1.8.2 and Python 3.6, running on a Windows 10 operating system. The hardware configuration included an Intel (R) Xeon (R) CPU E5-2698 v4 @ 2.20 GHz, equipped with 16 GB RAM, and an NVIDIA Tesla K80 GPU with 16 GB of RAM. The experimental parameters were as follows:

For the Weibo, Resume, ZJ datasets, the Cross-Transformer's FFN layer had 384 hidden layer nodes, a dropout parameter of 0.3, 8 heads in the multi-head attention layer with a dimensionality of 16 per head, resulting in a total of 128 nodes. Learning rate was fixed at 0.0018, and the training was run for 50 epochs.

For the MSRA dataset, the Cross-Transformer's FFN layer featured 480 hidden layer nodes, a dropout of 0.3, 8 heads in the multi-head attention with a dimensionality of 20 per head, totaling 160 nodes. Learning rate was set at 0.0014, and the model was trained for 100 epochs. Additionally, in our experiments, the smoothing factor α was adjusted to 0.1 and 0.2, respectively.

4.3. Results and Discussion

From Table 2, each method has a certain improvement on the performance of the FLAT model. Among them, the performance improvement of the model was the greatest after adding KVMN, with a 2-point increase on MSRA, Resume, and ZJ, respectively, and an improvement of about 6% on the Weibo dataset. This improvement can be attributed to the advanced technologies introduced by KVMN, such as knowledge enhancement and memory mechanisms. KVMN enabled the model to better understand contextual information in text, effectively capturing long-distance dependencies between entities and adapting to various types of text and datasets. The MSRA and Weibo datasets contain more diverse and complex text types, which the Cross-Transformer module can use to better utilize the cross-text information interaction ability in the module, thereby improving the recognition performance of our model. The Resume dataset may mainly consist of highly formatted text, such as personal information and work experience in resumes, which have relatively fixed structures and regular entity distributions. In this case, the Cross-Transformer module may not be able to fully leverage its advantages and may even experience a decrease in effectiveness due to the introduction of unnecessary complexity. This can explain why by adding the Cross-Transformer module, the model's performance on MSRA and Weibo datasets improved by 0.5% but decreased by 0.25% on the Resume dataset. The regularity-aware feature bumps the model's efficacy by an average of 2%. After incorporating the Boundary Smoothing module into the baseline, the model's performance demonstrated improvement across all four datasets, with an approximate increase of 2.57% on the MSRA dataset, 1.05% on the ZJ dataset, 0.79% on the Resume dataset, and 3.41% on the Weibo dataset. The KCB-FLAT outperforms the baseline FLAT model on the MSRA, ZJ, Resume, and Weibo datasets by margins of 3.28%, 2.61%, 2.09%, and 7.95%, respectively, in summary. By tempering the model's predictions and curbing overconfidence on specific boundaries, the introduction of the Boundary Smoothing module enhances the model's adaptability to new domains and changing data distributions, thereby improving the precision and stability of named entity recognition.

Table 2. The results of ablation experiments.

Model	KVMN	CT	RA	BS	MSRA	ZJ	Resume	Weibo
Baseline (FLAT)					93.45	94.27	94.93	63.42
+KVMN	√				95.98	96.04	96.23	69.27
+KVMN&CT	√	√			96.24	95.64	95.98	69.78
+RA			√		96.14	95.66	96.02	66.98
+BS				√	96.02	95.32	95.72	66.83
+BS&RA			√	√	96.30	95.97	96.09	67.15
Ours $(\mathrm{KCB}-\mathrm{FLAT}.\alpha=0.2)$	√	√	√	√	96.59	96.87	97.02	71.37

The improvements indicated in Table 2 are mathematically correlated with the optimization of boundary conditions, which are critical for reducing boundary recognition errors. The Boundary Smoothing technique allows for a more accurate determination of boundary values, which in turn significantly impacts the model's performance by minimizing boundary residuals. Additionally, the enhanced boundary conditions contribute to a more robust convergence, reflected in the decreased variance of our error metrics. These mathematical refinements are essential for enhancing the model's overall efficacy.

In this experiment, we selected NFLAT model [44], Li et al.'s model [21], Xiong et al.'s model [45], DiffusionNER [46], RICON and MECT [47], LSTM + Lexicon augmentation model [48], AESINER [35], Mao et al. [49], and SLK-NER [50]. Comparison experiments were conducted with our KCB-FLAT, and the experimental results are as follows (Table 3).

Table 3. The comparative test results.

Models	MSRA	ZJ	Resume	Weibo
NFLAT [44]	94.55	95.40	95.58	61.94
Li et al. (2022a) [21]	95.97	96.08	96.32	72.18
Xiong et al. (2023) [45]	95.40	95.51	95.97	68.14
DiffusionNER (2023) [46]	/	94.88	94.53	/
RICON [47]	96.12	96.01	95.98	66.82
MECT [47]	96.21	96.13	95.91	70.40
LSTM + Lexicon augment [48]	93.47	96.02	95.58	61.22
AESINER [35]	96.53	96.42	96.40	70.54
Mao et al. (2022) [49]	/	95.98	96.29	70.81
SLK-NER [50]	/	96.01	95.78	64.00
Ours (KCB-FLAT)	96.59	96.87	97.02	71.37

4.4. Analysis and Discussion

In comparison with recent Chinese NER models, the model presented in this study exhibits notable superiority on four datasets: MSRA, Resume, Weibo, and ZJ. When compared with MECT, our proposed model offers a marginal improvement on the MSRA and Resume datasets, and a substantial 0.97% enhancement on the Weibo dataset. This advancement is credited to the integration of syntactic information encoded within KNMN, coupled with the inclusion of the Smoothed Boundary module, thereby validating the augmented model's efficacy. The inclusion of syntactic details elevates the precision of Chinese named entity recognition within context-heavy texts. However, our model is slightly inferior to Li et al.'s model on the Weibo dataset, trailing by 0.81 percentage points. Li et al. [21] adeptly tackled the primary challenge of unified NER by framing it as a classification task for inter-word relationships, particularly those between adjacent entity words, yielding impressive outcomes on the Weibo dataset. Although regularity can sometimes hinder the recognition precision of named entities within diverse contexts, the introduction of Boundary Smoothing allows the generated confidence to more accurately mirror the predicted entities' correctness, bolstering generalizability. Specifically, the Boundary Smoothing module elevates named entity recognition precision in certain scenarios. As an illustration, on both the MSRA and Resume datasets, our model surpasses MECT, RICON, and Mao et al. 2022 [49].

4.5. In-Depth Analyses

To better understand how our ground model leads to improvements, we recorded the LOSS values when training the model on MSRA and Resume, respectively. Figures 6 and 7 visualize the loss on MSRA and Resume for models with smoothing of 0.1 and 0.2. We can conclude that when KCB-FLAT integrates the Boundary Smoothing module, we get lower LOSS values in the training of the model; at $\alpha = 0.2$, our model has a lower LOSS value compared with the case of $\alpha = 0.1$. CE means that without any Boundary Smoothing, we get smaller LOSS values overall. The lower LOSS value means that our improvement makes the model effective in reducing the bias between predicted and actual results during training, and makes our model better at completing the segmentation task first, thus obtaining better results in the named entity recognition.

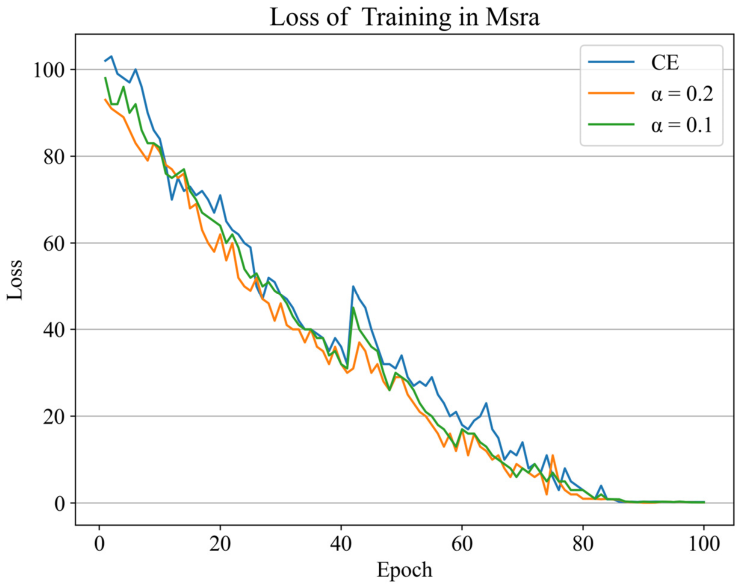

Figure 6. LOSS values of MSRA during training.

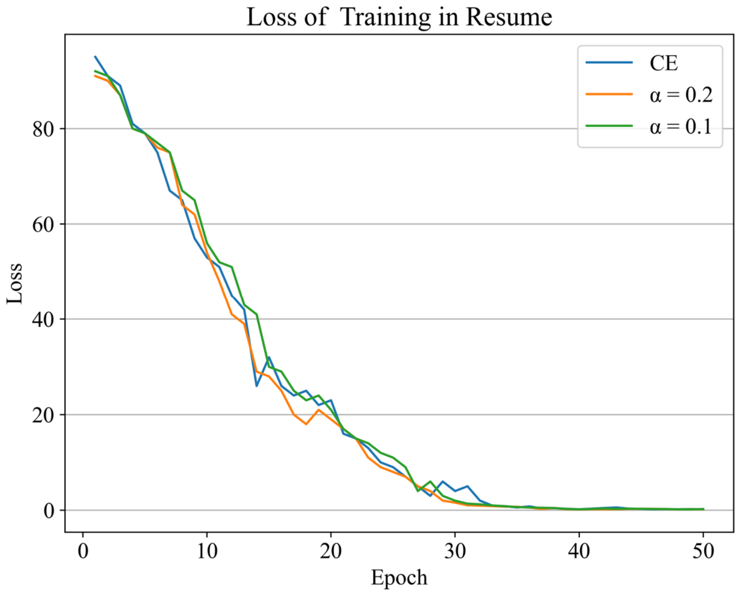

Figure 7. LOSS values of Resume during training.

In this section, we perform a case analysis and use examples from the ZJ and Weibo datasets for our case analysis. We compare the basic FLAT model with our designed KCB-FLAT model, presenting the entity recognition outcomes in a tabular format. In this context, B, I, and E represent the beginning, interior, and the end of an entity, respectively. While O signifies a character that means non-entity. Entity types are represented by ORG, LOC, NAM (unique noun), and NOM (common noun). For instance, B-ORG designates the commencement, category, and position of an organizational entity. The table reveals that our customized Chinese named entity recognition model, KCB-FLAT, enhances recognition precision and boundary detection compared with the baseline FLAT model.

Referring to Table 4, when identifying “Congtai Branch of Handan Public Security Bureau”, FLAT only recognizes “Handan Public Security Bureau,” omitting “Congtai Branch” as it is not perceived as a full entity. However, even though “Handan Public Security Bureau” is identified, it does not align with the contextual entity type. Syntactically, both “Handan Public Security Bureau” and “Congtai Branch” belong to the same syntactic category and should carry equal syntactic weight. KCB-FLAT more precisely identifies “Congtai Branch of Handan Public Security Bureau” as a unified, independent entity.

Table 4. Case analysis of Weibo dataset.

Weibo	**奥林匹克公园 (The Olympic Park)**					
Entity	AO	LIN	PI	KE	GONG	YUAN
Gold label	B-LOC.NAM	I-LOC.NAM	I-LOC.NAM	I-LOC.NAM	I-LOC.NAM	E-LOC.NAM
Baseline (FLAT)	B-LOC.NAM	I-LOC.NAM	I-LOC.NAM	E-LOC.NAM	B-LOC.NOM	E-LOC.NOM
KCB-FLAT(Ours)	B-LOC.NAM	I-LOC.NAM	I-LOC.NAM	I-LOC.NAM	I-LOC.NAM	E-LOC.NAM

In Table 5, FLAT’s recognition splits “Olympics” and “park” into separate entities, failing to recognize them as a cohesive unit. Upon closer inspection, “Olympics” is categorized as a unique noun, whereas “park” is deemed a common noun. Conversely, KCB-FLAT accurately identifies the phrase as a singular address entity, indicating that our enhancements have bolstered the model’s text recognition.

Table 5. Case Analysis of ZJ Dataset.

ZJ	**邯郸市公安局丛台分局 (Congtai Branch of Handan Public Security Bureau)**									
Entity	HAN	DAN	SHI	GONG	AN	JU	CONG	TAI	FEN	JU
Gold label	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	E-ORG
Baseline (FLAT)	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	E-ORG	O	O	O	O
KCB-FLAT(Ours)	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	I-ORG	E-ORG

5. Conclusions

In this study, we present KCB-FLAT, a Chinese NER model incorporating multiple innovations. It specifically employs a Key-Value Memory Network to encode syntactic data, fuses syntactic and lexical data through a Cross-Transformer module, and refines boundary recognition with a Boundary Smoothing module during the decoding stage. Through rigorous experimentation on four datasets, our model effectively mitigates overconfidence issues and significantly boosts Chinese NER performance. Detailed examinations underscore the superiority of our methodology in training efficacy, suggesting that the Boundary Smoothing mechanism significantly bolsters model performance and calibration. Our forthcoming research will endeavor to extend the KCB-FLAT model along several avenues, aiming to augment its generalizability and adaptability, and to integrate the model into practical scenarios. We envision future studies to probe the applicability of this model across diverse linguistic frameworks and to scrutinize the scalability of the Boundary Smoothing technique within expansive datasets. Enhancements in computational efficiency and the fine-tuning of boundary conditions could also be considered. Moreover, the methodologies introduced in this research are poised for adaptation to other languages, entailing the customization of the Boundary Smoothing algorithm to accommodate the distinct linguistic traits of each language. This endeavor may confront unique challenges, such as the adaptation to varied syntactic architectures and the intricacies of diverse phonological systems.

Author Contributions: Conceptualization, Z.D. and Z.H.; methodology, Z.D. and Z.H.; validation Z.D., Z.H. and J.Z.; formal analysis, Z.D., Z.H. and S.W.; investigation Z.D., Z.H., S.W. and J.Z.; writing original draft and visualization, Z.H. and S.W.; supervision S.W. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Guangxi Science and Technology Project (nosAB22035052, AB20238013), the National Natural Science Foundation of China (No. 62362016), and Guangxi Key Laboratory of Image and Graphic Intelligent Processing Project (nos. GIP2211, GIP2308), and the Innovation Project of GUET Graduate Education (No. 2023YCXS062).

Data Availability Statement: There are no restrictions on the sharing of relevant data in this study.

Acknowledgments: The authors thank the Special Issue editors and anonymous reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yin, D.; Cheng, S.; Pan, B.; Qiao, Y.; Zhao, W.; Wang, D. Chinese Named Entity Recognition Based on Knowledge Based Question Answering System. *Appl. Sci.* **2022**, *12*, 5373. [CrossRef]
2. Bose, P.; Srinivasan, S.; Sleeman, W.C., IV; Palta, J.; Kapoor, R.; Ghosh, P. A Survey on Recent Named Entity Recognition and Relationship Extraction Techniques on Clinical Texts. *Appl. Sci.* **2021**, *11*, 8319. [CrossRef]
3. Chen, S.; Pei, Y.; Ke, Z.; Silamu, W. Low-Resource Named Entity Recognition via the Pre-Training Model. *Symmetry* **2021**, *13*, 786. [CrossRef]
4. Ahmad, P.N.; Shah, A.M.; Lee, K. A Review on Electronic Health Record Text-Mining for Biomedical Name Entity Recognition in Healthcare Domain. *Healthcare* **2023**, *11*, 1268. [CrossRef] [PubMed]
5. Huang, C.; Wang, Y.; Yu, Y.; Hao, Y.; Liu, Y.; Zhao, X. Chinese Named Entity Recognition of Geological News Based on BERT Model. *Appl. Sci.* **2022**, *12*, 7708. [CrossRef]
6. Szczepanek, R. A Deep Learning Model of Spatial Distance and Named Entity Recognition (SD-NER) for Flood Mark Text Classification. *Water* **2023**, *15*, 1197. [CrossRef]
7. Yang, J.; Teng, Z.; Zhang, M.; Zhang, Y. Combining discrete and neural features for sequence labeling. In Proceedings of the Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, 3–9 April 2016; pp. 140–154. [CrossRef]
8. He, H.; Sun, X. F-score driven max margin neural network for named entity recognition in Chinese social media. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Part 3: 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2015), Valencia, Spain, 3–7 April 2017; pp. 713–718. [CrossRef]
9. Yao, L.; Huang, H.; Wang, K.-W.; Chen, S.-H.; Xiong, Q. Fine-Grained Mechanical Chinese Named Entity Recognition Based on ALBERT-AttBiLSTM-CRF and Transfer Learning. *Symmetry* **2020**, *12*, 1986. [CrossRef]
10. Chiu, J.P.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
11. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, Australia, 6–11 August 2017; Volume 3. [CrossRef]
12. Wang, Z.; Shang, J.; Liu, L.; Lu, L.; Liu, J.; Han, J. Crossweigh: Training named entity tagger from imperfect annotations. *arXiv* **2019**, arXiv:1909.01441.
13. Enwei, Z.; Jinpeng, L. Boundary Smoothing for Named Entity Recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022*; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; Volume 1, pp. 7096–7108. [CrossRef]
14. Deng, Z.; Tao, Y.; Lan, R.; Yang, R.; Wang, X. Kcr-FLAT: A Chinese-Named Entity Recognition Model with Enhanced Semantic Information. *Sensors* **2023**, *23*, 1771. [CrossRef]
15. Li, X.; Yan, H.; Qiu, X.; Huang, X. FLAT: Chinese NER Using Flat-Lattice Transformer. *arXiv* **2020**, arXiv:2004.11795.
16. Tourani, A.; Bavle, H.; Avşar, D.I.; Sanchez-Lopez, J.L.; Munoz-Salinas, R.; Voos, H. Vision-Based Situational Graphs Exploiting Fiducial Markers for the Integration of Semantic Entities. *Robotics* **2024**, *13*, 106. [CrossRef]
17. Wang, N.; Ge, R.F.; Yuan, C.F.; Wong, K.F.; Li, W.J. Company name identification in Chinese financial domain. *J. Chin. Inf. Pro.* **2002**, *16*, 1–6.
18. Sohrab, M.G.; Miwa, M. Deep exhaustive model for nested named entity recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2843–2849. [CrossRef]
19. Markus, E.; Kevin, P.; Adrian, U. ManyEnt—A Dataset for Few-shot Entity Typing. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, Online, 8–13 December 2020; pp. 5553–5557. [CrossRef]
20. Shen, Y.; Ma, X.; Tan, Z.; Zhang, S.; Lu, W. Locate and label: A two-stage identifier for nested named entity recognition. *arXiv* **2021**, arXiv:2105.06804.
21. Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; Li, F. Unified named entity recognition as word-word relation classification. In Proceedings of the AAAI conference on artificial intelligence, Seattle, WA, USA, 28 June 2022; pp. 10965–10973. [CrossRef]

22. Tan, Z.; Shen, Y.; Zhang, S.; Lu, W.; Zhuang, Y. A sequence-to-set network for nested named entity recognition. *arXiv* **2021**, arXiv:2105.08901.
23. Ma, X.; Hovy, E. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv* **2016**, arXiv:1603.01354.
24. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A unified MRC framework for named entity recognition. *arXiv* **2019**, arXiv:1910.11476.
25. Yu, J.; Bohnet, B.; Poesio, M. Named Entity Recognition as Dependency Parsing. *arXiv* **2020**, arXiv:2005.07150.
26. Yu, B.; Hang, Z.; Shu, X.; Liu, T.; Wang, Y.; Wang, B.; Li, S. Joint extraction of entities and relations based on a novel decomposition strategy. *arXiv* **2019**, arXiv:1909.04273.
27. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.
28. Athiwaratkun, B.; Santos, C.N.D.; Krone, J.; Xiang, B. Augmented natural language for generative sequence labeling. *arXiv* **2020**, arXiv:2009.13272.
29. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1906.08237.
30. Ni, J.; Abrego, G.H.; Constant, N.; Ma, J.; Hall, K.B.; Cer, D.; Yang, Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv* **2021**, arXiv:2108.08877.
31. Daneshfar, F.; Aghajani, M.J. Enhanced text classification through an improved discrete laying chicken algorithm. *Expert Syst.* **2024**, *41*, e13553. [CrossRef]
32. Liang, L.-X.; Lin, L.; Lin, E.; Wen, W.-S.; Huang, G.-Y. A Joint Learning Model to Extract Entities and Relations for Chinese Literature Based on Self-Attention. *Mathematics* **2022**, *10*, 2216. [CrossRef]
33. Lou, C.; Yang, S.; Tu, K. Nested named entity recognition as latent lexicalized constituency parsing. *arXiv* **2022**, arXiv:2203.04665.
34. Ma, J.; Ballesteros, M.; Doss, S.; Anubhai, R.; Mallya, S.; Al-Onaizan, Y.; Roth, D. Label Semantics for Few Shot Named Entity Recognition. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 1956–1971.
35. Nie, Y.; Tian, Y.; Song, Y.; Ao, X.; Wan, X. Improving named entity recognition with attentive ensemble of syntactic information. *arXiv* **2020**, arXiv:2010.15466.
36. Petrovska, B.; Atanasova-Pacemska, T.; Corizzo, R.; Mignone, P.; Lameski, P.; Zdravevski, E. Aerial Scene Classification through Fine-Tuning with Adaptive Learning Rates and Label Smoothing. *Appl. Sci.* **2020**, *10*, 5792. [CrossRef]
37. Wu, P.; Cui, Z.; Gan, Z.; Liu, F. Three-Dimensional ResNeXt Network Using Feature Fusion and Label Smoothing for Hyperspectral Image Classification. *Sensors* **2020**, *20*, 1652. [CrossRef]
38. Mahayossanunt, Y.; Nupairoj, N.; Hemrungrojn, S.; Vateekul, P. Explainable Depression Detection Based on Facial Expression Using LSTM on Attentional Intermediate Feature Fusion with Label Smoothing. *Sensors* **2023**, *23*, 9402. [CrossRef]
39. Ashish, V.; Noam, S.; Niki, P.; Jakob, U.; Llion, J.; Aidan, N.G.; Lukasz, K.; Illia, P. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. [CrossRef]
40. Rafael, M.; Simon, K.; Geoffrey, H. When Does Label Smoothing Help? *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 4696–4705. [CrossRef]
41. Lukasik, M.; Bhojanapalli, S.; Menon, A.; Kumar, S. Does label smoothing mitigate label noise? In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 6448–6458. [CrossRef]
42. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *JML* **2011**, *12*, 2493–2537. [CrossRef]
43. Gu, Y.; Qu, X.; Wang, Z.; Zheng, Y.; Huai, B.; Yuan, N.J. Delving deep into regularity: A simple but effective method for Chinese named entity recognition. *arXiv* **2022**, arXiv:2204.05544.
44. Wu, S.; Song, X.; Feng, Z.; Wu, X. NFLAT: Non-Flat-Lattice Transformer for Chinese Named Entity Recognition. *arXiv* **2022**, arXiv:2205.05832.
45. Xiong, L.; Zhou, J.; Zhu, Q.; Wang, X.; Wu, Y.; Zhang, Q.; Gui, T.; Huang, X.; Ma, J.; Shan, Y. A Confidence-based Partial Label Learning Model for Crowd-Annotated Named Entity Recognition. *arXiv* **2023**, arXiv:2305.12485.
46. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. Diffusionner: Boundary diffusion for named entity recognition. *arXiv* **2023**, arXiv:2305.13298.
47. Wu, S.; Song, X.; Feng, Z. MECT: Multi-Metadata Embedding based Cross-Transformer for Chinese Named Entity Recognition. *arXiv* **2021**, arXiv:2107.05418.
48. Ma, R.; Peng, M.; Zhang, Q.; Huang, X. Simplify the usage of lexicon in Chinese NER. *arXiv* **2019**, arXiv:1908.05969.
49. Mao, Q.; Li, J.; Meng, K. Improving Chinese Named Entity Recognition by Search Engine Augmentation. *arXiv* **2022**, arXiv:2210.12662.
50. Hu, D.; Wei, L. SLK-NER: Exploiting second-order lexicon knowledge for chinese NER. *arXiv* **2020**, arXiv:2007.08416.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.