



OPEN

Named entity recognition in aerospace based on multi-feature fusion transformer

Jing Chu¹, Yumeng Liu¹, Qi Yue¹, Zixuan Zheng² & Xiaokai Han¹✉

In recent years, along with the rapid development in the domain of artificial intelligence and aerospace, aerospace combined with artificial intelligence is the future trend. As an important basic tool for Natural Language Processing, Named Entity Recognition technology can help obtain key relevant knowledge from a large number of aerospace data. In this paper, we produced an aerospace domain entity recognition dataset containing 30 k sentences in Chinese and developed a named entity recognition model that is Multi-Feature Fusion Transformer (MFT), which combines features such as words and radicals to enhance the semantic information of the sentences. In our model, the double Feed-forward Neural Network is exploited as well to ensure MFT better performance. We use our aerospace dataset to train MFT. The experimental results show that MFT has great entity recognition performance, and the F_1 score on aerospace dataset is 86.10%.

Unprecedented advances have been made in the domain of aerospace, with manned spaceflight technology being progressively commercialized, such as SpaceX's Dragon spacecraft. In combination with artificial intelligence, some of the complex operational steps in the aerospace domain will become simpler and autonomy can be exploited more in accordance with their prescribed tasks. Similar to humans, artificial intelligence first needs to learn prior knowledge. Text is one of the main storage forms of human knowledge. Therefore, it is particularly important to acquire knowledge accurately and quickly from a large number of aerospace text materials.

Named Entity Recognition (NER) is an essential technology to extract knowledge from documents. Its purpose is to extract words with actual meaning from text, including names of people, places, institutions and proper nouns. Unlike English, which has spaces as natural separators, Chinese entity recognition first needs to perform word segmentation on Chinese sentences, which makes Chinese named entity recognition more challenging. For example, if a sentence containing 12 characters is entity recognized, use $C = \{c_1, c_2, \dots, c_{12}\}$ to express. the result may be w_1 , w_2 or w_3 (w_3 contains both w_1 and w_2), but only w_3 is the correct result.

In order to reduce the impact of word segmentation errors, Zhang et al. proposed a Lattice structure that can consider both characters and words, and this structure was used on the Lattice-LSTM (Lattice Long-Short Term Memory)¹. As shown in Fig. 1, the structure matches the sentence with the lexicon to obtain all potential words contained in the sentence, and then performs feature extraction for each character and the matched words in the sentence. The Lattice will use the contextual information to determine which of w_1 , w_2 and w_3 is the correct word segmentation result, avoiding the recognition failure caused by word segmentation errors. Li et al. modified the Lattice structure to be combined with Transformer-XL², and proposed FLAT (Flat-Lattice-Transformer)³. FLAT uses a Flat-Lattice, which places the words matched from the lexicon at the end of the input sentence and determines the position of these words in the sentence through position encoding. However, this method ignores the radical information of Chinese characters.

As a character evolved from hieroglyphs, the radicals of Chinese characters usually contain a lot of information. This information could be used to further enhance the semantic information. Dong et al. used Bidirectional Long Short-Term Memory (Bi-LSTM) to decompose the Chinese character structure to obtain character-level embeddings and demonstrated the effectiveness of the method⁴. However, the Long Short Term Memory (LSTM) has insufficient parallel computing ability and its performance is lower than that of the Transformer.

To address these issues, we used a NER model based on Multi-Feature Fusion Transformer (MFT). The model is based on FLAT and uses a One-Dimensional Convolutional Neural Network (1D-CNN) to integrate information about the corresponding radicals of Chinese characters. The MFT uses features from Chinese characters, words and radicals to make it computationally efficient and reduce errors in word segmentation by introducing

¹School of Automation, Xi'an University of Posts & Telecommunications, 618 West Chang'an Street, Chang'an District, Xi'an 710121, China. ²School of Astronautics, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi'an 710072, China. ✉email: hxk1256422591@163.com

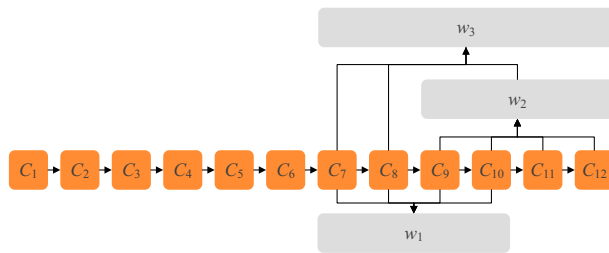


Figure 1. Use the lexicon to match the Chinese sentence to the words w_1 , w_2 and w_3 , these Chinese words make it easier for the NER model to determine entity boundaries. Whether the entity in a sentence is w_1 or w_2 or w_3 can be determined by the NER model using contextual semantics.

radical information to enhance semantic information. The structure of the Transformer has also been adapted to make it perform the task of named entity recognition better.

In order to recognize entities in aerospace contexts, the MFT model needs to be trained on the aerospace domain corpus. At present, there is no publicly available Chinese named entity recognition dataset in the aerospace domain. Therefore, within the scope permitted by law, we used the web crawler technology to obtain relevant data from Baidu Encyclopedia and other websites. To that end, we have produced an aerospace dataset that contains 30 K Chinese sentences to complete the model training and testing.

Related work

Aerospace named entity recognition belongs to the specific field of named entity recognition, but it still belongs to the research in the field of named entity recognition. Deep learning has advanced rapidly in recent years and various named entity recognition methods based on deep learning have appeared. As one of the early deep learning models, LSTM was applied to the named entity recognition task by Hammerton⁵. However, LSTM only extracts features in a sentence from a single direction. To solve this problem, Huang et al. used BiLSTM that combined with Conditional random fields (CRF) for the entity recognition task and had achieved satisfactory results. In addition to temporal models that can be used for semantic modeling⁶, Collobert et al. used Convolutional Neural Networks (CNN) as NER model encoders to model local semantic features of sentences and generate corresponding labels with CRF as decoders⁷. Dos Santos used an improved CNN model for Natural Language Processing (NLP), 1D-CNN, to recognize entities. Experiments show that this improvement is very effective⁸. However, these methods only perform feature extraction on characters or words one by one. For this reason, Vaswani et al. proposed a Transformer model based on a self-attentive mechanism, which provides a new idea for named entity recognition. The method not only improves the recognition accuracy of the model, but also reduces the training time of the model⁹. Dai et al. believe that the modeling ability of long-term dependency is crucial to the language model, which is also the defect of Transformer, so they improved it and proposed Transformer_XL model, which improves the modeling ability of Long-Term dependency by 80%. However, Guo et al. believe that named entity recognition is different from other language models and should pay more attention to the modeling of local semantics¹⁰. They propose a lightweight Star Transformer model. Experiments show that this model is more suitable for NER tasks.

Chinese named entity recognition methods are classified into character-based named entity recognition methods and word-based entity recognition methods. Character-based approaches lose word information in sentences, and word-based approaches are more influenced by the quality of the segmentation. Liu et al. discuss character-based and word-based approaches separately and conclude that character-based approaches are empirically better choices¹¹. However, some researchers have tried to combine the two methods by combining lexicon information on a character-based approach. Gui et al. proposed Lexicon Rethinking Convolutional Neural Network (LR-CNN), which uses a lexicon to assist the model in the determination of entity boundaries¹². Zhang et al. proposed Lattice LSTM, which reinforces semantic and entity boundaries by using a lexicon. Gui et al. proposed a Lexicon-Based Graph Neural Network (LGN), where the graph neural network is used to introduce the latent word information matched by the dictionary into the model to complete the entity recognition task¹³. Li et al. proposed FLAT, which uses relative position encoding to recover lattice structure information. Since Lattice is compatible with Transformer, the performance of the model is further improved.

In terms of structural features of Chinese characters, Dong et al. introduced the structural information of Chinese characters into the NER model for the first time and used Bi-LSTM for the feature extraction of Chinese radicals; this method achieved the best performance on the MSRA dataset. Meng et al. used images of Chinese characters to assist in completing NER by leveraging the image information of Chinese characters to take advantage of the strokes and structural features of Chinese characters¹⁴.

There are also many named entity recognition works in the aerospace domain. Xu et al. crawled relevant texts from NASA's official website to produce a spacecraft named entity recognition dataset and used CRF to complete the entity recognition task¹⁵. Boan Tong et al. used the book World Spacecraft Encyclopedia as the data source for constructing the spacecraft-related dataset and performed migration learning through the Bert-BiGRU-CRF (Bidirectional Gated Recurrent Unit, BiGRU) model to fine-tune the model parameters in the spacecraft domain corpus to accomplish the entity recognition task in the spacecraft domain¹⁶. Tikayat et al. developed an English-language aerospace dataset with which they fine-tuned BERT for better recognition performance in the

aerospace domain¹⁷. In this paper, we will develop a Chinese aerospace dataset and propose a new recognition method based on the characteristics of Chinese.

Aerospace dataset

Since there is no publicly available named entity recognition dataset in the aerospace domain, we use the crawler system to obtain relevant corpus from the data on Internet websites such as Wikipedia to the extent permitted by laws and use Label Studio for manual labelling. A dataset of aerospace domain with 29,953 sentences and 51,482 entities is made. The construction process of the aerospace dataset is shown in Fig. 2.

First, we use a crawler based on the Scrapy framework to obtain aerospace data from Wikipedia and China Aerospace News, then we filtered the corpus to remove contents that are not relevant to the domain. After that, we sliced the corpus in sentences and ensured that each sentence contained at least two aerospace entities. Finally, the corpus was labeled in the BIO format with the help of Label Studio. An example of the BIO Labeling format is shown in Fig. 3, where 'B' stands for 'Begin' and is used to annotate the head of the entity, 'I' stands for 'Inside' and is used to annotate the rest of the entity and 'O' is for 'Outside' and is used to annotate the non-entity.

Entities are categorized into aerospace companies and organizations (ACAO), Airports and spacecraft launch sites (AASLS), Type of aerospace vehicle (TOAV), Constellations and satellites (CAS), Space missions and projects (SMAP), Scientists and astronauts (SAA), aerospace technology and equipment (ATAE). 7 types. In this paper, 80% of the data in the dataset is used to train the model, 10% is used to validate the model, 10% is used to test the model. The main information of the aerospace dataset is shown in the Table 1.

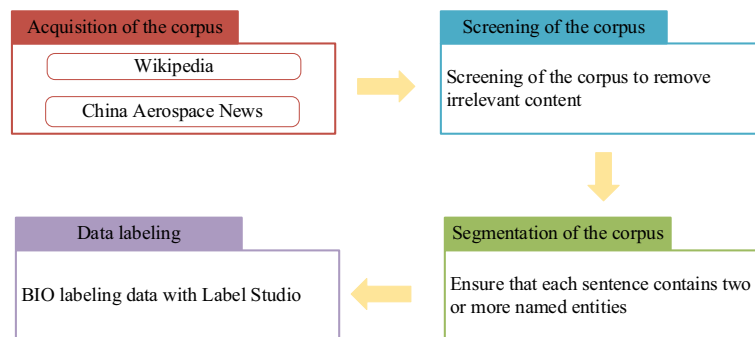


Figure 2. Extracting relevant corpus from Wikipedia and Chinese space news, screening and segmentation of the corpus and labeling it in BIO format using Label Studio.

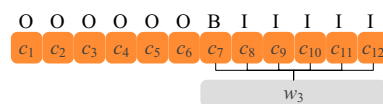


Figure 3. Annotation of sentences containing 12 Chinese characters using the BIO labeling method.

Type	Train	Dev	Test
Sentence	24,000	2976	2977
Entity	41,222	5080	5180
ACAO	2023	254	206
AASLS	453	59	54
TOAV	3467	418	441
CAS	1413	165	153
SMAP	7380	897	930
SAA	550	55	84
ATAE	25,922	3232	3312

Table 1. The main information of aerospace dataset.

Multi-feature fusion transformer

Since the word and radical information are very important features for Chinese characters. So in this paper we use MFT that can fuse these information as a named entity recognition model. The network structure of the MFT model is shown in Fig. 4. The model first extracts the radical embedding of Chinese characters through 1D-CNN, then fuses it with the Lattice sequence embedding output by the FLAT-Lattice model and encodes it as the input of the Flat-Lattice model, which is encoded as inputs to the Double Feed-forward Multi-head Self-attention (DFMS) encoder module, and finally decodes the corresponding label sequences by CRF. In the DFMS encoder module, MFT has exploited the structure of the Transformer by adding a Feed-Forward Neural Network (FFN) before the multi-headed self-attention module. This sandwich structure of the Transformer shows better performance in the NER.

Flat-lattice module

Similar to the Flat-Lattice module in the FLAT model, the Flat-Lattice module in the MFT uses a lexicon to match the input sentence to obtain the potential words contained in the sentence and encodes the positions of characters and these potential words in order to construct the Lattice. The structure of the Flat-Lattice module is shown in Fig. 5. For example, if a sentence containing 12 characters is entity recognized. Match the sentence with the lexicon to get the potential words w_1, w_2 and w_3 . These matched potential words are placed at the end of the sentence as candidates for the entities in the sentence, which together with the sentence form the lattice sequence $LS = \{l_1, \dots, l_n\}$. The tokens in the Flat-Lattice are then located using the head position and tail position to restore the Lattice structure information.

Next, for the Flat-Lattice sequence, we need to convert it to Flat-Lattice sequence embedding and encode it by positions. The Lattice sequence embedding $LE = \{le_1, \dots, le_n\}$ can be obtained by matching LS in a pre-trained embedding table. Their positional embeddings, on the other hand, are calculated respectively by using Eqs. (1)–(7).

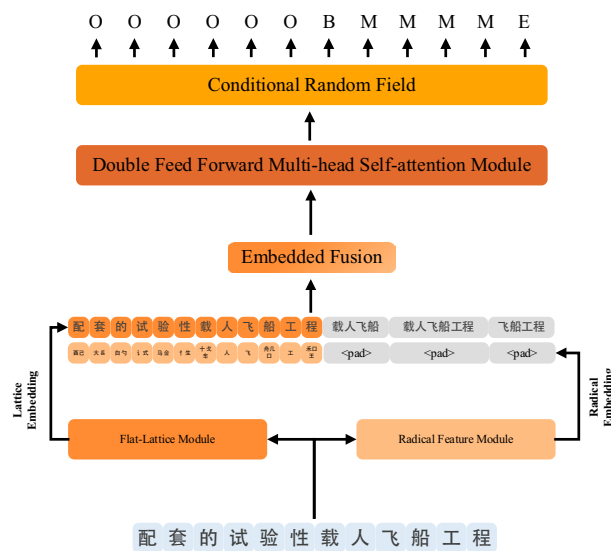


Figure 4. The flat lattice module and the radical feature module represent the embedding of the Chinese sentence respectively, and the double feed-forward multi-self-attention module encodes these embeddings, which are finally decoded by the conditional random fields to obtain the label sequence.

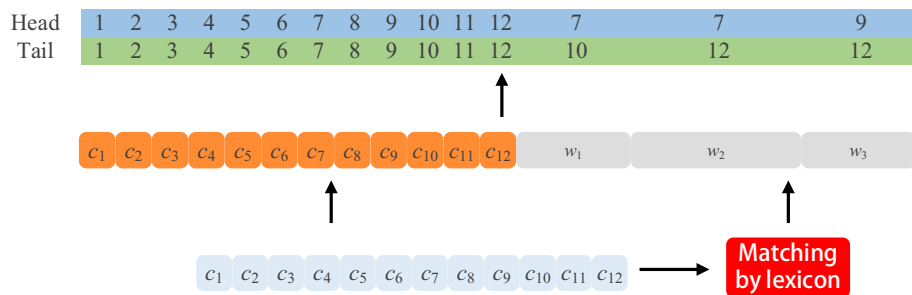


Figure 5. Flatten the lattice by using the head and tail positions of Chinese characters and words to record the position of each token in the lattice structure.

$$R_{ij} = \text{ReLU}((P_{d_{ij}}^{hh} \oplus P_{d_{ij}}^{ht} \oplus P_{d_{ij}}^{th} \oplus P_{d_{ij}}^{tt})W_p) \quad (1)$$

$$P_d^{2k} = \sin(d/1000^{2k/d_{emb}}) \quad (2)$$

$$P_d^{2k+1} = \cos(d/1000^{2k/d_{emb}}) \quad (3)$$

$$d_{ij}^{(hh)} = \text{head}[i] - \text{head}[j] \quad (4)$$

$$d_{ij}^{(ht)} = \text{head}[i] - \text{tail}[j] \quad (5)$$

$$d_{ij}^{(th)} = \text{tail}[i] - \text{head}[j] \quad (6)$$

$$d_{ij}^{(tt)} = \text{tail}[i] - \text{tail}[j] \quad (7)$$

where R_{ij} in Eq. (1) represents the relative position encoding between token i and token j with \oplus representing the concatenation operator and W_p being the learnable parameter. $P_{d_{ij}}^{hh}$ represents the encoding of the relative distance between the head positions of token i and token j , $P_{d_{ij}}^{ht}$, $P_{d_{ij}}^{th}$ and $P_{d_{ij}}^{tt}$ have similar meaning, which is calculated using the same formula as the position code calculation in Transformer, $P_{d_{ij}}^{hh}$ as shown in Eqs. (2)–(3), where k represents the index of the position coding dimension, d_{emb} represents the position coding dimension, d_{ij}^{hh} represents the distance between the head positions of token i and token j , d_{ij}^{ht} , d_{ij}^{th} , and d_{ij}^{tt} have similar meaning, and they are calculated by Eqs. (4)–(7), where $\text{head}[i]$ denotes the head position of token i and $\text{tail}[j]$ denotes the tail position of token j .

Radical feature module

In Chinese, some characters such as "river", "lake" and "sweat" are related to water, so they all contain the same radical. The radicals in Chinese characters are similar to the root affixes in English. As a kind of characters evolved from hieroglyphs, Chinese characters contain a lot of semantic features in their radicals. In order to use these semantic features to enhance the semantic information of sentences, Radical Feature Module splits each Chinese character into multiple radicals by radical dictionary and encodes these radicals by 1D-CNN to obtain the radical encoding of the corresponding Chinese character.

Take the radical encoding of a sentence containing 12 characters as an example. Each character in the sentence is matched in the radical dictionary to obtain the radical group corresponding to each character, where the character with the highest number of radicals contains 3 radicals, then the size of the convolution kernel of 1D-CNN is set to 3 and the step size is also 3. The remaining words with less than 3 radicals are filled in with "<PAD>", a symbol used exclusively for filling in deep learning. The convolution process is shown in Fig. 6. By this method, we can obtain the corresponding radical embedding sequence $RE = \{re_1, \dots, re_n\}$ for the sentence.

Radical Feature Module only extracts the radical feature of the characters in the input sentence, and the radical feature of the potential words in the sentence is not extracted. This results in that the lengths of LE and RE are different. In order to facilitate the subsequent fusion of them, Radical Feature Module also uses "<PAD>" to fill in the radical sequence embedding RE , so that the lengths of LE and RE are consistent with each other.

Finally, the lattice sequence embedding and the radical feature sequence embedding are concatenated to obtain the sequence embedding $E = \{e_1, \dots, e_n\}$, as shown in Eqs. (8) and (9).

$$E = LE \oplus RE \quad (8)$$

$$E = LE + RE \quad (9)$$

DFMS encoder module

There are two kinds of neural networks in DFMS encoder, which are Self-Attention Neural Network and Feedforward Neural Network. The structure is shown in Fig. 7, where the Self-attentive Neural Network is the same as the self-attentive network in Transformer_XL, which uses relative position coding, with the aim of improving the model's ability to model long-term dependencies. DFMS has a Double Feedforward Neural Network, with the self-attentive neural network added between them. This structure has proven to be effective in Conformer¹⁸. Residual connections and normalization are also required between each layer of neural networks.

We use the sequence embedding E , which fuses LE and RE , as the input to the DFMS encoder. As shown in Eqs. (10)–(14). Firstly, the sequence embedding E enters into the first layer of FFN calculation, and the output result enters into the Self-attentive Neural Network for self-attention coding after the layer normalization and residual connection. The coded result also needs the residual connection and layer normalization processing. In the end, after the second FFN calculation, the final encoding of the encoder is obtained.

$$E^* = E + ((\text{ReLU}(EW_1 + b_1))W_2 + b_2) \quad (10)$$

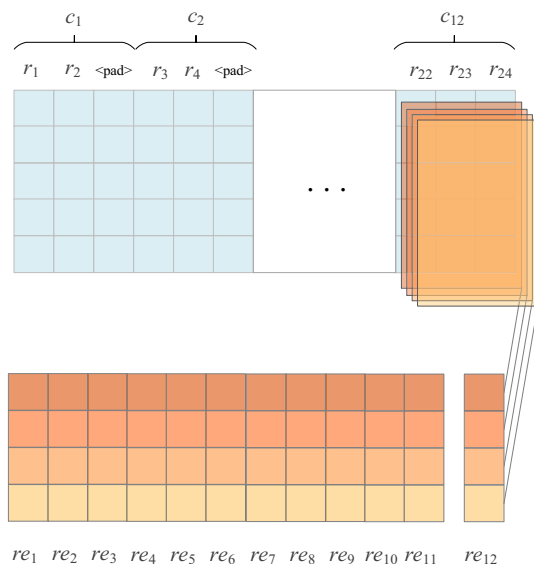


Figure 6. Extraction of the radicals in Chinese characters using 1D convolution to obtain the radical embedding for each character.

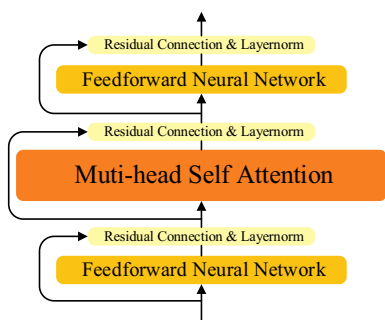


Figure 7. Using double feed-forward neural networks to clip multi-head self-attention modules with residual connections and layer normalization between them.

$$[Q, K, V] = E^*[W_q, W_k, W_v] \tag{11}$$

$$S_{ij} = (Q_i + u)^T K_j + (Q_i + v)^T R_{ij} W_R \tag{12}$$

$$A = E^* + \text{Softmax}(S)V \tag{13}$$

$$X = A + ((\text{ReLU}(AW_3 + b_3))W_4 + b_4) \tag{14}$$

where W_q, W_k, W_v and W_R are the query mapping matrix, the key mapping matrix, the value mapping matrix and the position mapping matrix respectively, all of which are learnable parameters. With W_q, W_k and W_v , the sequence embedding E is mapped to the query matrix Q , the key matrix K and the value matrix V respectively. In Eq. (12) u and v are also learnable parameters that are used to ensure that the attentional bias of the query vector remains constant for different tokens². $W_1, W_2, W_3, W_4, b_1, b_2, b_3$ and b_4 are learnable parameters of the Feedforward Neural Network.

CRF decoder module

Conditional random fields are often used in machine learning-based named entity recognition methods. Benefiting from its excellent performance, CRF is usually used as decoders based on neural network named entity recognition models. CRF is a conditional probability distribution model that can be used to solve prediction problems. Cuong et al. propose that CRF can be used to solve the labeling problem and derive the most sensible label in conjunction with the semantic¹⁹. In the named entity recognition task, CRF takes the input sequence of

observations as the set of random variables X and the output sequence of labels as Y . As shown in Eqs. (15)–(16), for a sequence $X = \{x_1, \dots, x_n\}$, the corresponding sequence of labels is $Y = \{y_1, \dots, y_n\}$. The probability of y is P .

$$P(y|x) = \frac{\exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i))}{Z(x)} \quad (15)$$

$$Z(x) = \sum_y \exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)) \quad (16)$$

where t_k is the transfer eigen-function and s_l is the state eigenfunction, taking values of 1 or 0. λ_k and u_l are the corresponding weight coefficients, which are learnable parameters.

Experiment

In the experiments of this paper, we compare the performance of the MFT model with some mainstream named entity recognition models on our aerospace dataset.

In addition, in order to verify whether the MFT model is only effective on our aerospace dataset, we also conduct performance comparison experiments on some commonly used and public named entity recognition datasets such as Weibo and Resume datasets. Finally, we conduct an effectiveness study on the MFT model to verify the effectiveness of our model structure.

Evaluation indicator

Common evaluation criteria used in Named Entity Recognition tasks are precision (P), recall (R) and F_1 score. (F_1). They are calculated respectively by using the formulas (17)–(19). Precision is the percentage of labels predicted by the model that are correctly predicted. Recall is the number of samples in the sample that are correctly predicted. As precision and recall are mutually exclusive metrics, a combined metric F_1 score is also needed to judge the recognition performance of the model.

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (19)$$

where TP denotes a positive sample with a correct prediction, FP denotes a negative sample with a failed prediction, FN denotes a positive sample with a failed prediction and TN denotes a negative sample with a correct prediction.

Dataset

In this paper, we constructed an Aerospace Named Entity Recognition dataset with data from Wikipedia and the China Aerospace News website. It contains 30 k sentences and 53,788 entities. We predefined seven entity types based on the contents of the data, which were labeled by six annotators dividing the work among themselves, and the results were confirmed and validated by a manager. The whole labeling process took about one month. We divide the dataset in the ratio of 8:1:1 to get the training dataset, developing dataset and testing dataset for training and testing our model. The dataset information is shown in Table 1. We used two mainstream Chinese NER datasets, the Weibo dataset^{20,21} and the Resume dataset¹. The corpus of the Weibo dataset is mainly drawn from social media and contains four types of entities: Person, Location, Organization and Geopolitic. The corpus of the Resume dataset is mainly from Sina Finance. and was made by manually labeling named entities with YEDDA system. Table 2 shows the main information of both datasets.

Dataset	Type	Train (K)	Dev (K)	Test (K)
Weibo	Sentence	1.4	0.27	0.27
	Char	73.8	14.5	14.8
Resume	Sentence	3.8	0.46	0.48
	Char	124.1	13.9	15.1

Table 2. Main information of weibo and resume datasets.

Experimental environment and parameters

In our experiments, we used the same word lexicon and pre-trained character and word embeddings as in the Lattice-LSTM, Radical lexicon from <https://github.com/kfcd/chaizi>. All comparison model codes are provided by the original authors. Our model was trained on an Ubuntu system using an RTX 3060.

The hyperparameters are set differently for different datasets. The hyperparameter setting for MFT are shown in Table 3. The hyperparameters are set differently for different datasets. On the aerospace dataset MFT consists of 9,765,018 trainable parameters. On the resume dataset, MFT consists of 9,319,506 trainable parameters.

Experimental results

In this study, we use the F_1 score as a criterion for judging the performance of the models, so the precision and recall of the models are the results achieved by the model with the highest F_1 score on the test set.

Aerospace dataset

The experimental results of MFT on the aerospace dataset are shown in Table 4. The experimental results indicate that MFT performs well, with a significant performance improvement of 0.97% in F_1 score compared to the baseline model FLAT, the recall rate increased by 0.77%, the precision, is 1.16%. LR-CNN and LGN performed worse on the aerospace dataset than on the other datasets, while the LSTM combined with Lattice achieved an F_1 score of 71.33%, which is 9.88% lower than our MFT model.

The adoption of the pre-training model BERT by MFT results in a substantial improvement in each performance. Although MFT + BERT does not perform as well as FLAT + BERT in terms of recall, both F_1 and P have to perform better.

Figure 8 shows the F_1 curve of each model during training on the aerospace dataset, and the performance improvement of MFT in terms of F_1 score is obvious compared to LGN, Lattice-LSTM and LR-CNN. Compared to FLAT, MFT has a faster improvement in F_1 score in the early stage of training. From the precision curve of each model in Fig. 9, it can be seen that MFT performs much better than FLAT in terms of precision during the training process, and after the 100th Epoch MFT's precision curve is higher than FLAT's precision curve almost everywhere. However, the recall curves of all models in Fig. 10 show that there is not much difference between the performance of MFT and FLAT with respect to the recall criterion, so the improvement in the overall performance metric F_1 score of MFT mainly comes from the improvement in the recognition precision of the model.

Table 5 shows the recognition of MFT for different classes of entities on the aerospace dataset. The best recognized entity type is AEAT with F_1 score of 83.48% followed by TOAV. The worst recognition rate is AASLS with F_1 score of 61.11% and also AASLS has the least number of entities. Thus the recognition effectiveness of the model is directly related to the amount of data.

Weibo dataset

Table 6 shows the experimental results of MFT on the Weibo dataset. Compared with other comparison models, MFT has a greater performance improvement with F_1 score of 64.38%. LR-CNN has the best performance in terms of precision, but the recall rate is 15.03% lower compared to MFT and the F_1 score is 7.84% lower. The comprehensive performance of the model is improved to a higher level when MFT uses BERT to pre-train the model.

Resume dataset

The experimental results of MFT on the Resume dataset are shown in Table 7. The experiments demonstrate that the Double Feed-forward Neural Network and the radical information of Chinese characters do bring

Dataset	N_{head}	D_{head}	D_{FFN}	Epoch	Layer	Batch
Aerospace	16	8	384	200	2	5
Weibo	16	8	384	200	1	5
Resume	8	16	384	200	2	10

Table 3. Hyperparameter setting for MFT.

Models	P	R	F_1
LatticeLSTM ¹	70.58	72.08	71.33
LR-CNN ¹²	70.53	64.35	67.35
LGN ¹³	71.66	72.6	72.13
FLAT ³	80.47	80.01	80.24
MFT	81.63	80.78	81.21
FLAT + BERT ³	85.33	85.49	85.41
MFT + BERT	86.77	85.44	86.10

Table 4. Aerospace NER results. Significant values are in bold.

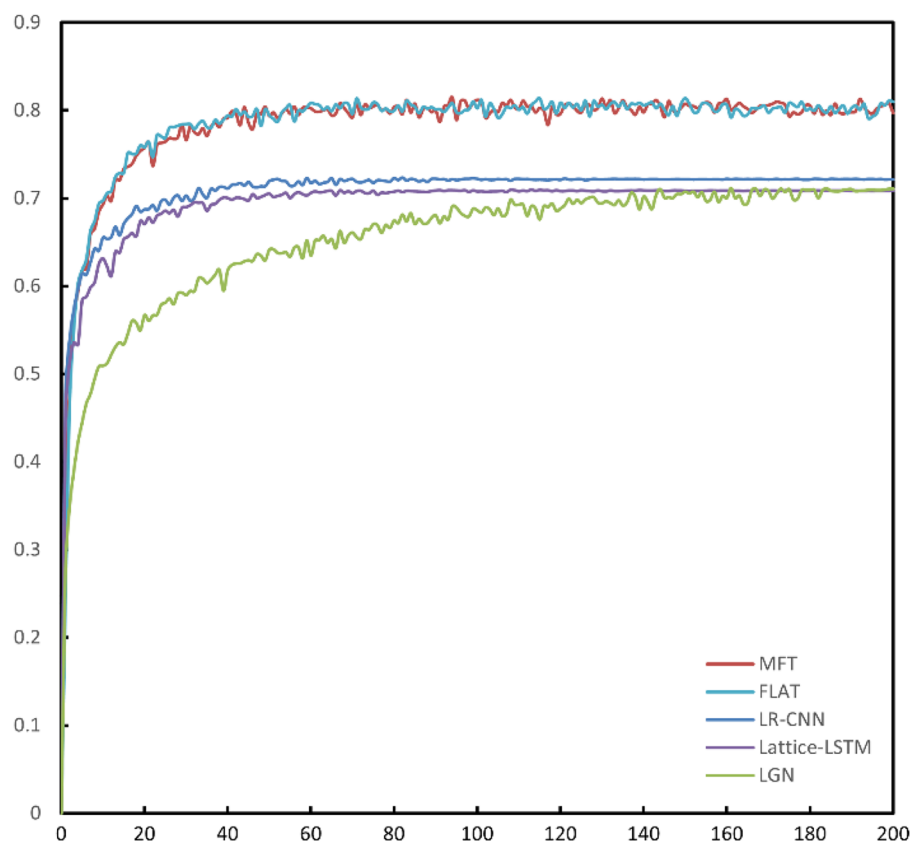


Figure 8. F_1 Curves during training of all models on the aerospace dataset. MFT's F_1 curve is essentially above FLAT.

performance improvements to the model with F_1 score of 95.78%, precision of 96.05% and recall rate of 95.52%, all of which are better than other models.

Experiments of feature fusion method

To study the effect on the MFT model after using different fusion methods on *LE* and *RE*, we conducted experiments on MFT on all three datasets. The experimental results are shown in Table 8. On the Weibo dataset and the Resume dataset, concatenating *LE* and *RE* performed better than adding them together. In contrast, For the aerospace dataset, concatenating *LE* and *RE* together still outperforms FLAT despite a decrease in precision, while the F_1 and recall of MFT are improved, especially the recall by 0.98%.

Experiments of FFN

The Conformer being used to solve the speech recognition problem contains a double half-step FFN, while the MFT contains a double full-step FFN. In order to verify whether double full-step FFN can bring more performance improvement than double half-step FFN in the named entity recognition task, we set up experiments on the impact of different FFN weight connection methods on the model performance. The experimental results are shown in Table 9. Compared to the double half-step FFN, the double full-step FFN is more suitable for the Named Entity Recognition task.

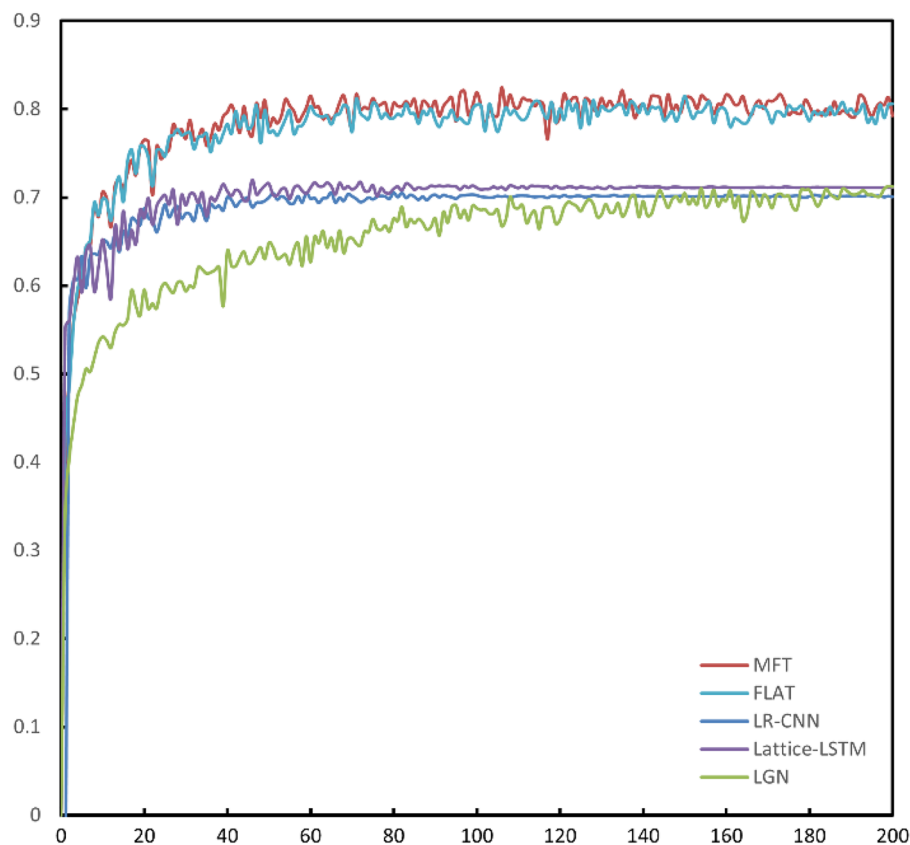


Figure 9. Precision curve during training of the comparison model on the aerospace dataset. MFT has a significantly higher precision rate curve than FLAT.

Effectiveness study

There are two main improvements of the MFT model, namely, the radical information of Chinese characters was added to enhance the semantics, and double FFN was used to improve the feature encoding capability of the model. In order to verify whether all these improvements bring performance benefits to MFT, we disassemble the model structure and conduct experiments on each of the three datasets. As shown in Table 10, we removed the Double FFN of the MFT and the F_1 scores of the MFT dropped by 0.47%, 0.5%, 0.19% on the Aerospace, Weibo, and Resume datasets, respectively, after which we proceeded to remove the Radical Feature Module of MFT and revert to FLAT, the F_1 scores of MFT dropped by 0.5%, 3.56%, 0.14%, respectively. Results in Table show that both improvements on the MFT are effective.

The effect of the radical feature on the attention of the model is intuitive, as can be seen in Fig. 11, where FLAT has a more focused attention score, while MFT adds extra attention to the information of FLAT. In such a way, the attention to key information is ensured not to be distracted. This allows MFT to converge faster than FLAT during the training of the model, and as shown in Fig. 12, where the loss curve of MFT is lower and decreases faster than that of FLAT.

Conclusions

In this paper, we propose an Aerospace Named Entity Recognition method based on multi-feature fusion Transformer. Big data from Wikipedia and China Aerospace News are obtained as corpus by crawlers and the aerospace dataset is produced using a manual labelling method. We train and test the MFT on our dataset and the experimental results demonstrate that our model has excellent performance, due to the fact that the radical features of the Chinese characters and the double Feed-forward Neural Network can provide a boost to the recognition rate of the MFT.

In future work, a wider range of Chinese features, such as the pronunciation and graphics of Chinese characters, could also be incorporated for a multimodal approach. However, incorporating more diverse features may introduce invalid elements or noise, which may lead to an increase in model parameters. To mitigate this problem, future work may also require filtering of features to reduce the model size and save computational costs.

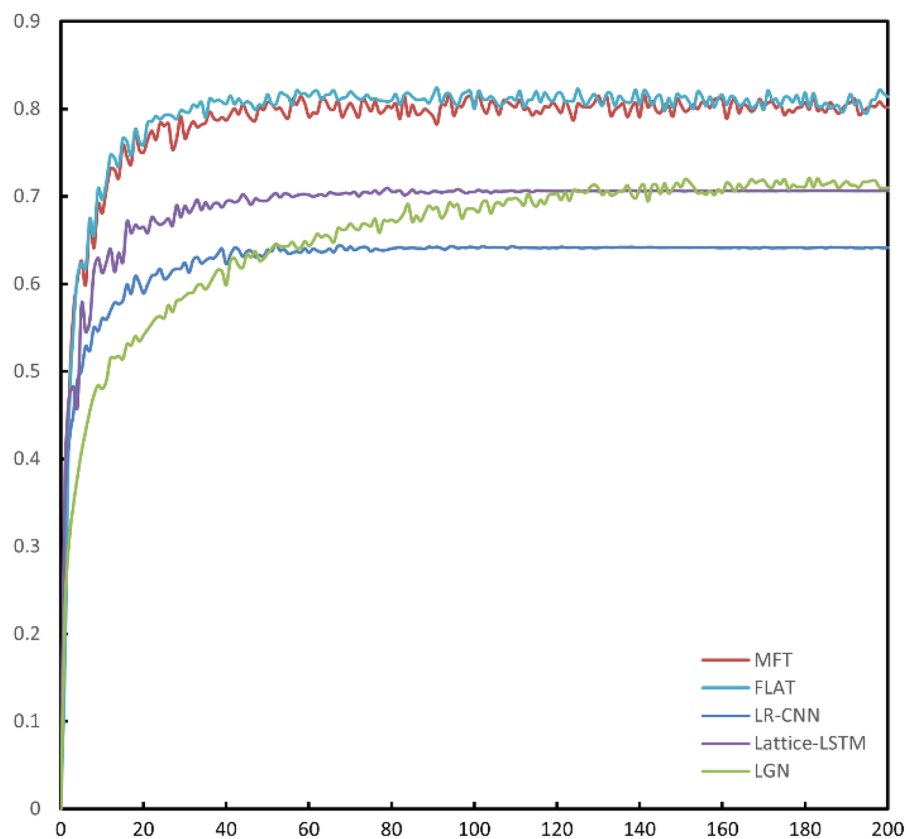


Figure 10. Recall curves during training of all models on the aerospace dataset. MFT’s recall is a bit lower than FLAT, but at the 160th epoch it’s basically equal to FLAT’s level.

Type	Train	Dev	Test	F ₁
ACAO	2023	254	206	75
AASLS	453	59	54	61.11
TOAV	3467	418	441	80.95
CAS	1413	165	153	64.05
SMAP	7380	897	930	78.49
SAA	550	55	84	61.90
ATAE	25,922	3232	3312	83.48

Table 5. MFT’s F₁ scores for different entity types.

Models	P	R	F ₁
LatticeLSTM ¹	52.71	53.92	53.13
LR-CNN ¹²	65.06	50.00	56.54
LGN ¹³	–	–	60.21
FLAT ³	–	–	60.32
MFT	63.72	65.03	64.38
FLAT + BERT ³	–	–	68.55
MFT + BERT	68.33	71.09	69.71

Table 6. Weibo NER results. Significant values are in bold.

Models	P	R	F ₁
LatticeLSTM ¹	94.81	94.11	94.46
LR-CNN ¹²	95.37	94.84	95.11
LGN ¹³	95.28	95.46	95.37
FLAT ³	–	–	95.45
MFT	96.05	95.52	95.78
FLAT + BERT ³	–	–	95.86
MFT + BERT	96.24	95.77	96.01

Table 7. Main results on resume NER. Significant values are in bold.

Dataset	Method	P	R	F ₁
Aerospace	Concatenation	81.63	80.78	81.21
	Addition	82.03	79.80	80.89
Weibo	Concatenation	63.72	65.03	64.38
	Addition	66.08	58.09	61.83
Resume	Concatenation	96.05	95.52	95.78
	Addition	94.03	93.35	94.72

Table 8. Result of different feature fusion method.

Dataset	Type	P	R	F ₁
Aerospace	Double half-step FFN	81.27	79.59	80.43
	Single full-step FFN	82.63	78.85	80.74
	Double full-step FFN	81.63	80.78	81.21
Weibo	Double half-step FFN	66.08	58.09	61.83
	Single full-step FFN	67.14	60.92	63.88
	Double full-step FFN	63.72	65.03	64.38
Resume	Double half-step FFN	94.97	95.09	95.03
	Single full-step FFN	95.64	95.42	95.59
	Double full-step FFN	96.05	95.52	95.78

Table 9. Result of different FFN.

Dataset	Model architecture	P	R	F ₁
Aerospace	MFT	81.63	80.78	81.21
	Double full-step FFN	82.63	78.85	80.74
	Radical feature module	80.47	80.01	80.24
Weibo	MFT	63.72	65.03	64.38
	Double full-step FFN	67.14	60.92	63.88
	Radical feature module	–	–	60.32
Resume	MFT	96.05	95.52	95.78
	Double full-step FFN	95.64	95.42	95.59
	Radical feature module	–	–	95.45

Table 10. Result of ablation study.

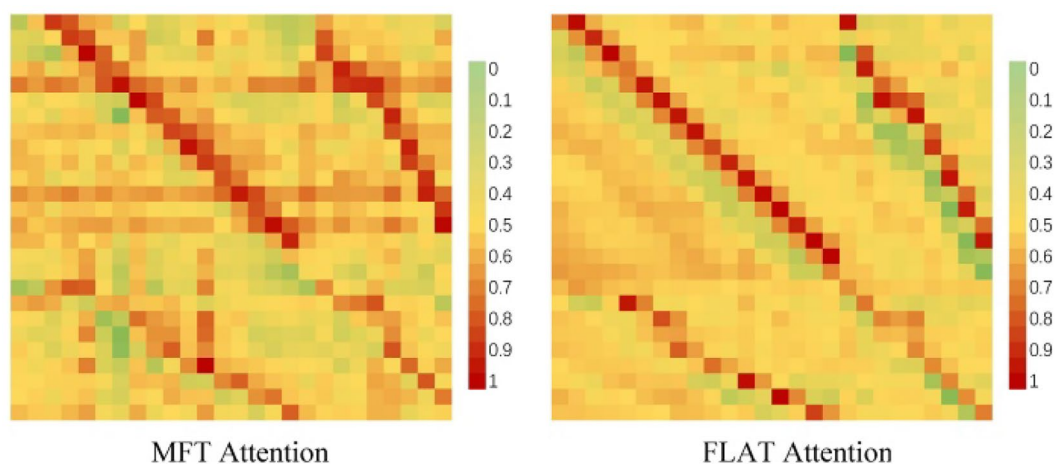


Figure 11. Visualization of attention for MFT and FLAT. MFT has a broader focus and more semantic features are extracted by the self-attentive mechanism.

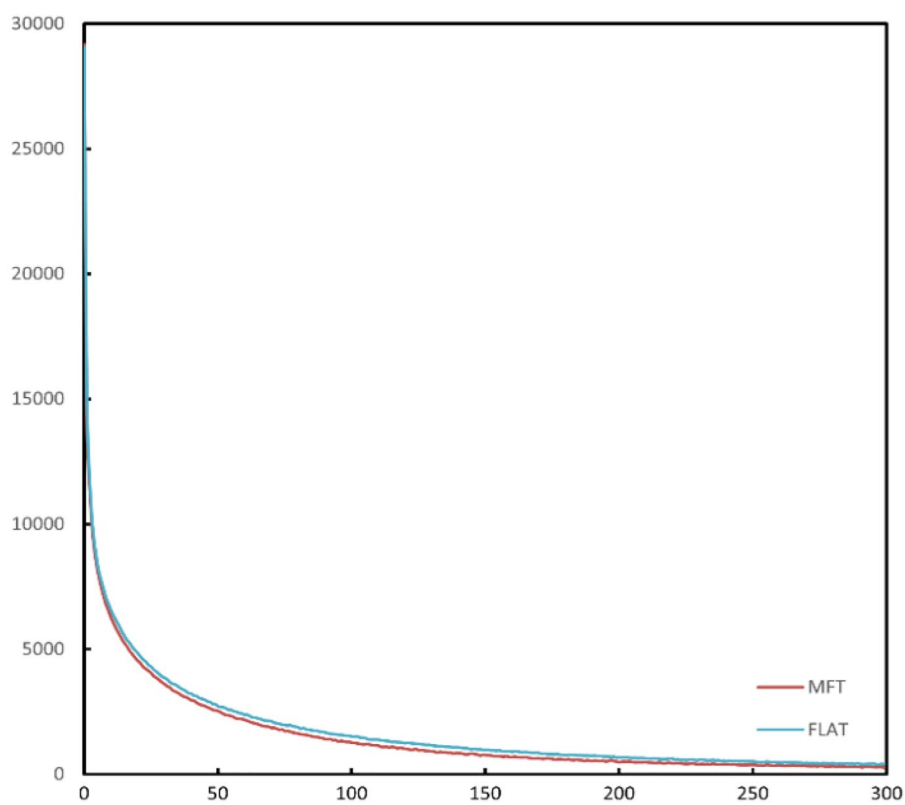


Figure 12. Loss curves for MFT and FLAT. MFT converges faster than the FLAT model and has lower losses.

Data availability

The datasets generated and analyzed during the current study are available in the GitHub repository, https://github.com/Coder-XIAOKAI/Aerospace_NERdatasets.

Received: 25 February 2023; Accepted: 23 December 2023

Published online: 08 January 2024

References

1. Zhang, Y. & Yang, J. Chinese NER using lattice LSTM. In *56th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Melbourne, Australia, 1554–1564* (2018).

2. Dai, Z. *et al.* Transformer-XL: Attentive language models beyond a fixed-length context. In *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2978–2988 (2019).
3. Li, X., Yan, H., Qiu, X. & Huang, X. *FLAT: Chinese NER Using Flat-Lattice Transformer* 6836–6842 (Association for Computational Linguistics, 2020).
4. Dong, C., Zhang, J., Zong, C., Hattori, M. & Di, H. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition. In *5th International Conference on Natural Language Processing and Chinese Computing (NLPCC)*, *24th International Conference on Computer Processing of Oriental Languages (ICCPOL)*, 239–250. (Kunming Univ Sci & Technol, 2016).
5. Hammerton, J. Named entity recognition with long short-term memory. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 172–175 (2003).
6. Huang, Z., Xu, W. & Yu, K. Bidirectional LSTM-CRF models for sequence tagging (2015) arXiv preprint [arXiv:1508.01991](https://arxiv.org/abs/1508.01991).
7. Collobert, R. *et al.* Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**, 2493–2537 (2011).
8. Dos Santos, C. & Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 69–78 (2014).
9. Vaswani, A. *et al.* Attention is all you need. In *31st Annual Conference on Neural Information Processing Systems (NIPS)*. Long Beach, CA (2017).
10. Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X. & Zhang, Z. Star-transformer. In *Proceedings of the 2019 Conference of the North* (2019).
11. Liu, Z., Zhu, C. & Zhao, T. Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? In *6th International Conference on Intelligent Computing, Changsha, Peoples R China*, 634–640. (2010).
12. Gui, T. *et al.* CNN-based Chinese NER with Lexicon rethinking. In *28th International Joint Conference on Artificial Intelligence, Macao, Peoples R China*, 4982–4988 (2019).
13. Gui, T. *et al.* A lexicon-based graph neural network for Chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1040–1050 (2019).
14. Meng, Y. *et al.* Glyce: Glyph-vectors for Chinese character representations. In *33rd Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada (2019).
15. Xu, J., Zhu, J., Zhao, R., Zhang, L. & Li, J. J. Astronautics named entity recognition based on CRF algorithm. *Electron. Des. Eng.* **25**, 42–46 (2017).
16. Tong, B., Pan, J., Zheng, L. & Wang, L. Research on named entity recognition based on bert-BiGRU-CRF model in spacecraft field. In *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, 747–753 (IEEE, 2021).
17. Tikayat Ray, A., Pinon-Fischer, O. J., Mavris, D. N., White, R. T. & Cole, B. F. *aeroBERT-NER: Named-entity recognition for aerospace requirements engineering using BERT*. In *AIAA SCITECH 2023 Forum*, 2583 (2023).
18. Gulati, A. *et al.* Conformer: convolution-augmented transformer for speech recognition. (2020). arXiv preprint [arXiv:2005.08100](https://arxiv.org/abs/2005.08100).
19. Cuong, N. V., Ye, N., Lee, W. S. & Chieu, H. L. Conditional random field with high-order dependencies for sequence labeling and segmentation. *J. Mach. Learn. Res.* **15**, 981–1009 (2014).
20. He, H. & Xu, S. *F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media* 713–718 (Association for Computational Linguistics, 2017).
21. Peng, N. & Dredze, M. Improving named entity recognition for Chinese social media with word segmentation representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin: ACL 2016*, 149–155 (2016).

Author contributions

Q.Y. proposed a new named entity recognition method for aerospace and wrote a new code for the recognition method. Y.L. constructed the new aerospace named entity recognition dataset and validated and compared the proposed method. J.C. validated and compared the new proposed method, and completed the manuscript writing and calibration. Z.Z. provided the named entity recognition corpus. X.H. wrote part of the code and participated in the writing and correction of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to X.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024