electronics

Article

ENEX-FP: A BERT-Based Address Recognition Model

Min Li, Zeyu Liu, Gang Li, Mingle Zhou * and Delong Han

Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
* Correspondence: zhouml@qlu.edu.cn

Abstract: In e-commerce logistics, government registration, financial transportation and other fields, communication addresses are required. Analyzing the communication address is crucial. There are various challenges in address recognition due to the address text's features of free writing, numerous aliases and significant text similarity. This study shows an ENEX-FP address recognition model, which consists of an entity extractor (ENEX) and a feature processor (FP) for address recognition, as a solution to the issues mentioned. This study uses adversarial training to enhance the model's robustness and a hierarchical learning rate setup and learning rate attenuation technique to enhance recognition accuracy. Compared with traditional named entity recognition models, our model achieves an F1-score of 93.47% and 94.59% in the dataset, demonstrating the ENEX-FP model's effectiveness in recognizing addresses.

Keywords: named entity recognition; BERT; address recognition; adversarial training; NLP

Citation: Li, M.; Liu, Z.; Li, G.; Zhou, M.; Han, D. ENEX-FP: A BERT-Based Address Recognition Model. *Electronics* **2023**, *12*, 209. https://doi.org/10.3390/electronics12010209

Academic Editor: Valentina E. Balas

Received: 14 December 2022
Revised: 27 December 2022
Accepted: 28 December 2022
Published: 1 January 2023

1. Introduction

Addresses are an essential type of textual information in everyday life. Many scenarios require the registration of addresses, such as e-commerce shopping, takeaway delivery and census. Address element parsing is breaking down address text into semantically independent elements and identifying the type of these elements. Information extraction (IE) automatically helps classify, extract and reconstruct large amounts of content. One of the approaches is named entity recognition (NER), which may extract predetermined semantic kinds from text [1]. NER is a crucial component of IE and has an impact on various downstream activities, such as relationship extraction and knowledge disambiguation [2].

Deep learning (DL) has been used in NER [3]. Recurrent Neural Networks (RNNs) and their variants have successfully modeled sequence data [4]. In particular, bidirectional RNNs effectively exploit past and future information in a specific time frame [5]. Long-distance dependencies are more effectively captured by Long Short Term Memory (LSTM). However, LSTM modeling cannot encode backward–forward information. Therefore, Bi-directional Long Short-Term Memory (BILSTM) is proposed based on LSTM [6], which combines LSTM to better understand contextual information. Before transformer, people commonly used CNN, RNN and Encoder–Decoder [7], three primary feature extraction techniques. However, the transformer model uses a self-attention mechanism instead of RNN's sequential structure, allowing the model to be trained in parallel and to have global information. Bidirectional Encoder Representation from Transformers (BERT) is a pre-training model, it places a strong emphasis on pre-training the bidirectional transformer to provide deep bidirectional language representations by employing the new masked language model (MLM).

BILSTM captures the semantics of each word in context. However, Jacob Devlin et al. proposed that BERT has strong feature extraction ability and BERT is superior to LSTM in feature acquisition [8]. Therefore, adding aBERT language pre-processing model to BILSTM model can better obtain word embedding.

However, after the input of the semantic information migrated through the BERT model into the BILSTM model, the output results may be affected by the insufficient feature richness, the low efficiency of BILSTM and the poor performance of the context long-distance dependence problem. Additionally, the BILSTM outputs the scores for each category that corresponds to a word after pre-processing the data, with the highest score being reported as the output. However, there may be instances where I is the first word or there are multiple consecutive Bs, which will decrease the identification's accuracy.

In summary, common address recognition models cannot fully use the results output by BERT, and the output feature vectors need to be further rounded up. Aiming at the characteristics of address text, such as free writing and often omitting terms, this paper makes full use of the feature vectors extracted by BERT and the advantages of being embedded as word vectors, which can carry out accurate recognition of addresses. This paper puts forward three contributions, which can be summarized as follows:

- Firstly, this paper constructs the ENEX, which fully uses the output of BERT as word vector embedding and feature vector extraction and aggregates the output of BERT and BILSTM. After the extractor processes the address text, it can reduce the impact of long-distance dependence between the left and right contexts, obtain richer features and better extract entity features.
- Secondly, this paper proposes the FP model to process the feature vector and condition constraints, increase the dependence between the learning labels, further improve the generalization of the model and significantly improve its effectiveness.
- Finally, this paper uses a learning rate decay strategy and a hierarchical setting learning rate operation to improve the model's accuracy effectively. Adversarial training is added to enhance the robustness of the model.

The rest of the paper is structured as follows: There is a review of relevant research in this field in Section 2. Section 3 presents the ENEX-FP model and the optimization method. Section 4 presents the experiments and provides a discussion. Finally, Section 5 presents the conclusion.

2. Related Work

2.1. NER

Named entity recognition (NER) is the process of identifying relevant entities in a text. John D. Lafferty proposed Conditional Random Fields (CRFs) [9] which combines the characteristics of the hidden Markov model and the maximum entropy model is an undirected graph model that has shown promise in recent years for applications requiring sequence annotation. In 2015, Baidu [10] published a paper concerning the BILSTM-CRF model, which combines a CRF and a bidirectional long short-term memory network (LSTM); this addresses the issue of text annotation. The BERT-CRF model has been utilized for Beijing air pollution complaints to aid in the [11] addition of text data from responses to public complaints about air pollution in Beijing from 2019 to 2020. Lin Junting et al. [12] used a CNN-BILSTM-CRF model for NER of underground onboard equipment, the accuracy of this model on the marked metro vehicle-mounted fault data is up to 0.95, which is higher than other entity recognition models. Ref. [13] propose a Chinese address recognition method based on multi-feature fusion; the accuracy of the proposed method is 4 to 10 percentage points higher than other methods on the self-constructed dataset. In order to automatically extract medical knowledge such as disease and treatment terms from Chinese electronic medical records, Dong, X.S. et al. [14] suggested a bidirectional recurrent neural network for NER, in discharge summary and progress records; the MacroF-scores of the proposed method are 0.03 higher than those of the baseline method. However, after the transformer model was proposed by A. Vaswani et al. [15], the self-attentive mechanism was widely applied to NER.

Traditional lexicon and rule-based approaches for Chinese address recognition rely excessively on lexicons and rule bases and have low recognition rates for ambiguous and unregistered words. The approach put forward in [16] by Paolo Nesi et al. addresses

recognition by using techniques such as pattern matching, clustering and NLP to geolocate Web domains and businesses, with Precision and Recall both above 0.90; the system exhibits excellent skills for extracting pertinent information about the geographic location of the studied web domains. Grumiau Christopher et al. [17] proposed using the predictive power of geotagged datasets to identify users' relevant points of interest (POIs). These works have some impact on address recognition.

BERT, as the encoder part of transformer, can be executed concurrently compared to RNN and LSTM and can more thoroughly depict sentence semantics by extracting the relationship characteristics of words in sentences at various levels. Xu, Lei et al. [18] proposed to use BERT-BiLSTM-CRF combined with attention for NER. The experimental results show that the F1-score of this method in the Chinese NER task reaches 0.9512.

Paper [19] introduces the BERT-BiGRU-CRF model, which is specially designed for these linguistic irregularities. The accuracy of this model on the MSRA dataset is 0.981, higher than other recognition models such as BERT-BILSTM-CRF. The ENEX-FP model makes full use of the feature vectors extracted from the BERT model and the advantages of embedding as word vectors to obtain the aggregated feature vectors and the features are dropped out and constrained conditionally to obtain a better recognition model than BERT-BiGRU-CRF.

2.2. Model Optimization

The learning rate tuning strategy is essential for DL. Ref. [20] introduces a new method for setting learning rates, called cyclic learning rates, which eliminates the need to find optimal values and schedules for global learning rates experimentally. Pavel Izmailov et al. show [21] that simply averaging multiple points along a Stochastic Gradient Descent (SGD) trajectory with a cyclic or constant learning rate allows for better generalization than conventional training.

The robustness can be enhanced through adversarial training and the accuracy can be further enhanced. Ref. [22] discovered the phenomenon of adversarial samples; they found that models with different structures will be affected by adversarial samples, which indicates that adversarial samples reveal the basic blind spot of the algorithm. Ref. [23] proposed that the fundamental reason for the vulnerability of neural networks to adversarial examples is its linear characteristics, which also explains the generalization of adversarial examples on the structure and training set and, on this basis, proposed the Fast Gradient Sign Method (FGSM) algorithm to generate adversarial examples. Goodfellow made some improvements based on the previous FGSM attack method; FSGM takes the same step in each direction and the Fast Gradient Method (FGM) subsequently proposed [24] carries out the scale according to the specific gradient to obtain better confrontation samples.

Therefore, this paper improves the model's accuracy on address recognition by adjusting the learning rate parameters. The training efficiency and model robustness are improved by adversarial training.

3. Our Method

3.1. Overview

The ENEX-FP model has the architecture diagram shown in Figure 1. After tuning and validation, we set lstm_units = 128; there are 128 × 2 LSTMS in Figure 1. We use two BERT models, each using a 12-layer encoder. The encoder's architecture is the encoder of the transformer, we illustrate this architecture in Figure 2 of Section 3.3.

The model inputs the address text into the ENEX, which, after vector representation, is fed into the BERT model of the extractor to obtain the feature vector extracted from the context. This vector is fed into the BILSTM processing to obtain the text vector. Finally, the text vector and the feature vector extracted from the context are features aggregated and sent to the feature processor for feature rounding and constraint processing and the result is output. Steps are as follows: extract text features in the ENEX to obtain a vector of batch_size * seq_length*(2*lstm_units+hidden_size) and finally, enter the FP for some

feature rounding and perform feature space transformation to obtain a vector of batch_size * seq_length* num_labels dimensions, then carry out the process further. The constraints are then further optimized to compute the optimal annotation sequence.

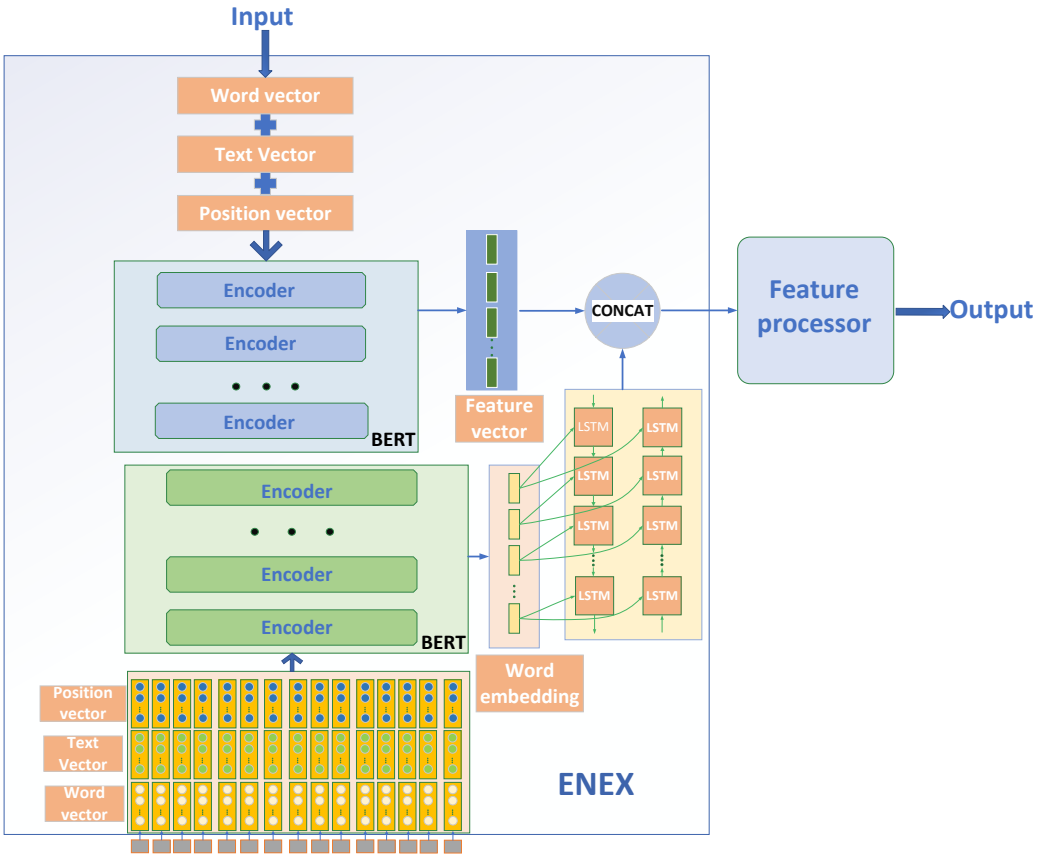


Figure 1. ENEX-FP model architecture.

3.2. Data Processing

During the training of this model, the sequence annotation method used for the address data is the BIOES annotation method. A train set, a dev set and a test set will be built from the address element resolution dataset. In the train set, the model is fitted to the data samples; in the dev set, the model's hyperparameters are changed; and in the test set, the final model's generalizability is assessed.

For deep learning approaches, a sizable annotated corpus is typically required. Otherwise, it is highly susceptible to overfitting and cannot achieve the expected generalization ability. We found in our experiments that the test results of some data labels, such as poi and community, could be better and the recognition results of these labels can be improved by data augmentation. Specifically, the corpus of the original address element resolution dataset is divided into sentences. Then the individual sentences are randomly spliced and used as the training corpus together with the original sentences. In addition, this paper uses the collected address elements, after manual BIOES annotation, to add them to the address element resolution dataset to obtain an expansion of the dataset, or uses random replacement to replace the labeled entities in the corpus with them to obtain an enhanced corpus.

3.3. ENEX Entity Extractor

The data after processing are fed into the ENEX for further processing. The primary baseline model of the ENEX is BERT, which is mainly used to generate feature vectors incorporating contextual information and consists of multiple encoders of the transformer; the transformer–encoder part is shown in Figure 2. This is done by mapping the Query,

Key and Value through h different linear transformations, stitching together the different Attentions and finally performing another linear transformation. The whole computational process can be represented as below.

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(head_1,\ldots,head_h)W^O \\ head_i &= Attention(QW_i{}^Q, KW_i{}^K, VW_i{}^V) \end{aligned} \tag{1}$$

where Q, K and V are the value of Query, Key and Value, and W denotes the matrix of weights.

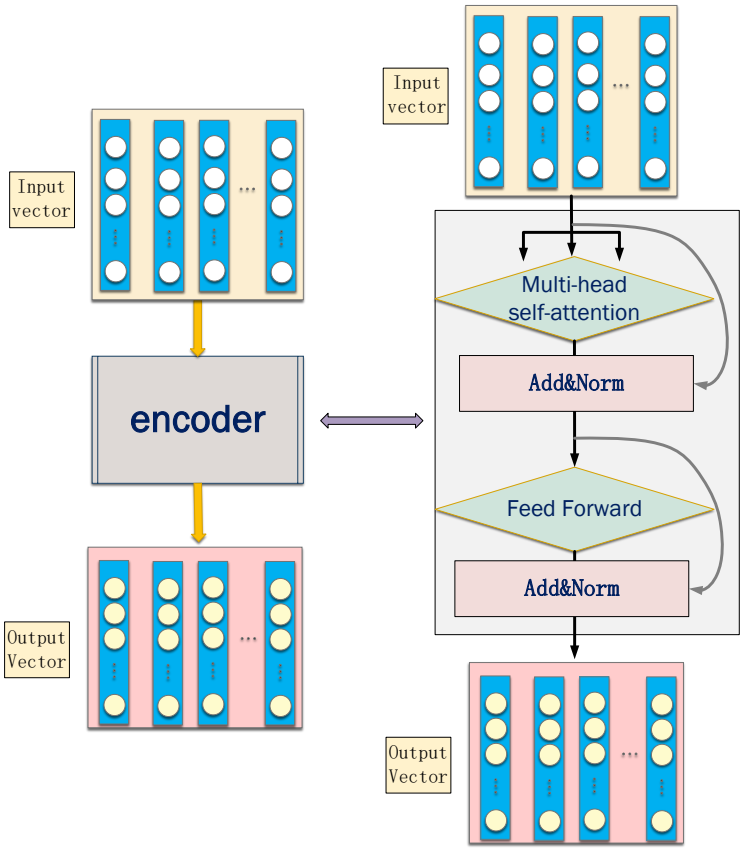


Figure 2. Transformer–encoder architecture.

In the base model, BERT outputs the vector in order to create the word embedding vector along with context data. This vector is then entered into the BILSTM model for feature extraction. To recognize the significance of long-distance information, LSTM introduces a memory unit and threshold mechanism. In ref. [25], to make optimal use of the feature data prior to and following the input, the author employs an improved threshold technique. The LSTM recurrent unit is shown in Figure 3. The LSTM consists of three gates: the forget gate is responsible for what history information is forgotten, the update gate is responsible for what history information is added and the output gate is the output [26]. Formulae for the three gates are shown below.

$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f) \tag{2}$$

$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u) \tag{3}$$

$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o) \tag{4}$$

where σ is the activation function, W is the weight matrix, b is the bias vector and $a^{\langle t-1 \rangle}$ is the update state at time t, and $\Gamma_f^{\langle t \rangle}$, $\Gamma_u^{\langle t \rangle}$, $\Gamma_o^{\langle t \rangle}$ are the outputs of the forget gate, the updating gate and the output gate.

After the feature vector enters BILSTM, it is trained to obtain the label prediction of each text. ENEX not only outputs the results of the model BERT-BILSTM but also outputs the extracted feature vectors. This happens because BERT, which has a higher ability to extract features, has learned the syntactic aspects and extensive semantic features of the text. To gain richer features and enhance the model's entity feature extraction, it can more effectively address the issues of long-distance dependence between the left and right contexts, insufficient features and text understanding errors.

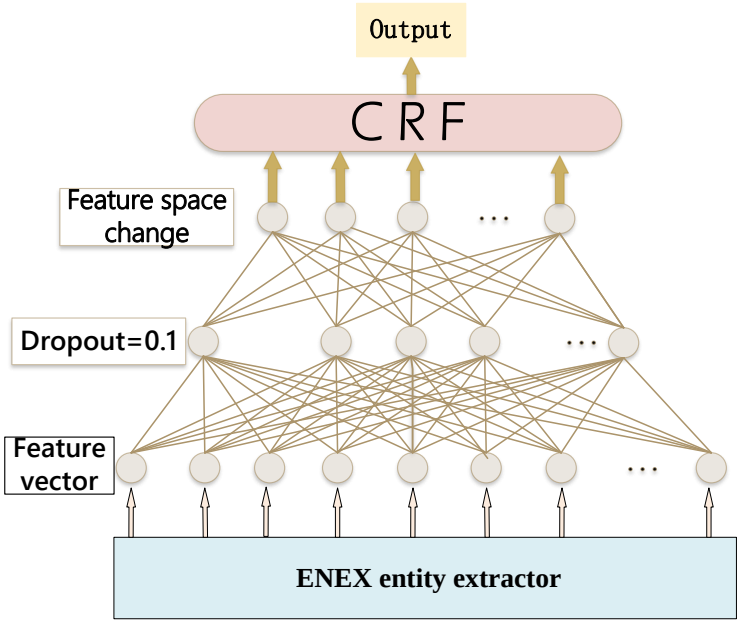


Figure 3. FP model.

3.4. Feature Processor

Processing the output of ENEX's feature fusion vectors, the research suggests an FP model. The FP model uses dropout to perform feature rounding on the input vectors, removing some dimensions of the input data to improve feature accuracy, mitigating overfit and mapping the dimensions of the feature-rounded vectors to the number of labels through a dense layer. The feature-transformed vectors are input to the FP module, as shown in Figure 3.

Figure 3 consists of four layers, which are the Feature vector extracted by ENEX, the Dropout layer, the Feature space change layer and the CRF model. After ENEX, the three-dimensional feature vector input to FP was [batch_size, seq_length, lstm_units * 2 + 768]. The value 768 is the output dimension of BERT. After parameter adjustment and verification, we finally set lstm_units = 128. In the process of setting the dropout, it is easy to produce the underfitting phenomenon when the dropout is too large. We use a dropout of 0.1 for experiments which can effectively reduce the occurrence of overfitting. We use dropouts of 0.1, 0.3 and 0.5 and find that the effect of a dropout of 0.1 is a little better. Therefore, 10% of the hidden neurons will be temporarily ignored after passing the dropout and the vector dimension will still be [batch_size, seq_length, lstm_units * 2 + 768]. After the feature space change layer, the dimension changes to [batch_size, seq_length, num_labels], where the dataset's total number of labels is num_labes. The vector is then input to the CRF layer for conditional constraints after feature conversion. The CRF layer can automatically learn these restrictions while processing training data.

The learning rate decay and stratified setting of the learning rate were also included in this model in this study, along with adversarial training, to further improve it.

3.5. Optimise the Model

3.5.1. Learning Rate

The learning rate is an important hyperparameter when optimizing neural networks. The value of the learning rate α is very critical, the larger the learning rate, the more quickly the weights will be updated. In a gradient descent method, if it is too large, it will not converge; if it is too small, it will converge too slowly.

The ENEX-FP model is optimized using a learning rate decay strategy. We use the Adam Learning Rate Optimizer, an extension of SGD [27]. Adam incorporates the advantages of Adagrad and the momentum gradient descent algorithm to accommodate both sparse gradients to mitigate gradient oscillations [28].

The BERT parameters have already reached a good level due to pre-training and the learning rate cannot be too large if it is to be kept from degrading. At the same time, the underlying structure is trained from scratch and training with a small learning rate is slow and difficult to synchronize with the BERT ontology training. This paper, therefore, sets up a learning rate stratification, setting a lower learning rate for the pre-training layer and a larger learning rate for the underjoin layer during training.

3.5.2. Antagonistic Training

Goodfellow first introduced adversarial training, which, in essence, entails perturbing the initial input sample to create an adversarial sample, which is subsequently trained. The purpose of adversarial training in NLP tasks is now more to regularize and enhance the generalization of the model rather than to protect against gradient-based malicious attacks. Madry's 2018 paper [29] proposes that adversarial training can be written uniformly in the following format.

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\Delta x \in \Omega} L(x + \Delta x, y; \theta) \right] \tag{5}$$

where $\mathcal{D}$ represents the distribution of the input samples, x represents the input, y represents the label, θ is the model parameter, Δx is the perturbation and Ω is the perturbation space.

This adversarial training process can be performed smoothly for tasks in computer vision, adding a continuous perturbation to the input. However, NLP is different. The input to NLP is text, which is essentially a one-hot vector and two different one-hot vectors, for which there is theoretically no "small perturbation"; hence, ref. [24] proposes to add the perturbation to the embedding layer. The model is further optimized using the FGM adversarial training, which increases the model's robustness.

4. Experiment

4.1. Datasets

The first dataset (DATA1) is the CCKS2021 Chinese Address Resolution Academic Assessment Task, a dataset provided by the Alibaba Dharma Institute for the resolution of Chinese address elements, which includes province, city, district, town, community, road, etc. In this paper, we added some collected Chinese addresses to this dataset, annotated them with BIOES sequences and added them to the dataset. This updated dataset is mainly used for entity recognition of Chinese addresses. The detailed statistics of the DATA1 corpus are summarized in Table 1. There are 9468 address texts in the train set, including 47,580 address elements. The dev set has 2367 address texts, including 11,854 address elements. The test set has 2708 address texts, including 13,590 address elements. What we need to do is to train and identify these address elements.

Table 1. Detailed statistics of the DATA1 corpus.

Corpus	Train Set	Test Set	Dev Set
Address Element Resolution	9468	2708	2367

We used open source address resolution data from Neural Chinese Address Resolution as the second dataset (DATA2), cleaned the data with rules and installed BIO parsing to parse the text, a total of 14,926 data were obtained, the type of address element is 23, including 84,662 address elements. According to 6:2:2 parsing, the data was arbitrarily split into a train set, a dev set and a test set. Detailed statistics of the DATA2 corpus are shown in Table 2.

Table 2. Detailed statistics of the DATA2 corpus.

Corpus	Train Set	Test Set	Dev Set
Neural Chinese Address Parsing	8957	2985	2985

4.2. Experimental Setup

Our model is mainly trained with the Tensorflow [30] and Keras libraries for code writing. The optimizer used is Tensorflow's Adam optimizer. The initial learning rate set in this paper is 0.00001 and the learning rate is decayed using Adam. BERT is the beginning learning rate according to the learning rate hierarchy and its successor module's learning rate is 100 times the original learning rate. For adversarial training, FGN is used for adversarial training in this paper. The detailed hyperparameter list is shown in Table 3.

Table 3. List of hyperparameters.

Parameters	Values
optimizer	Adam
batch_size	64
epoch	50
max_len	70
lstm_units	128
drop_rate	0.1
learning_rate	1×10^{-5}, 1×10^{-3}, 1×10^{-3}
activation	relu

In the process of super parameter selection, we used batch_size of 32, 64, 128 for experiments. The experimental results show that when batch_size is 32, the training efficiency is low and the accuracy rate varies considerably. When batch_size is set to 128, the required memory capacity increases. Finally, when using batch_size of 64, the model's overall effectiveness and its content capacity are well-balanced, which facilitates the completion of the experiment. In the process of setting dropout, it is easy to produce the underfitting phenomenon when the dropout is too large. When we use dropout of 0.1 for experiments, we can effectively reduce the occurrence of overfitting. With dropouts of 0.1, 0.3 and 0.5, we found that a dropout of 0.1 is a little better. In the address element dataset, the maximum sample length of the training set is 69 and the average sample length is 17, and the maximum sample length of the verification set is 76 and the average sample length is 16. We limit the maximum sample length to 70, which can effectively train and verify the address text. The design of other hyperparameters is also selected by us after a large number of repeated tests, which are favorable to the experimental results.

4.3. Assessment Indicators

The paper uses Precision, Recall and F1-score to evaluate our experimental results. Precision is the percentage of system results correctly identified and Recall is the percentage

of total entities correctly identified by the system. The following equations give Precision, Recall and F1-score:

$$Precision = \frac{\#\mathrm{TP}}{\#(\mathrm{TP} + \mathrm{FP})} \tag{6}$$

$$Recall = \frac{\#\mathrm{TP}}{\#(\mathrm{TP} + \mathrm{FN})} \tag{7}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

TP is a positive sample that the model projected to be in the positive class; FP is a negative sample that the model predicted to be in the positive class; FN is a positive sample that the model predicted to be in the negative class.

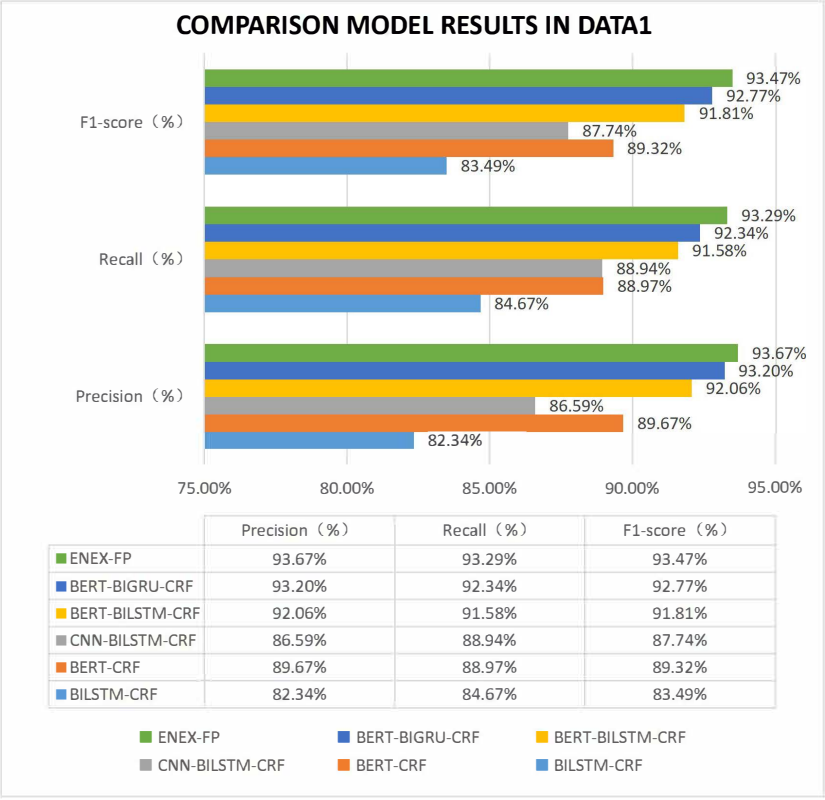
4.4. Experimental Results

Three experiments are conducted in this paper, namely the comparison model experiment, the ablation experiment and the data enhancement experiment, to demonstrate the accuracy of the model for address element recognition.

4.4.1. Comparative Model Experiment

In the comparative model experiment, we contrast our model with other widely used NER models. Under the same equipment and conditions and using the same dataset, we conducted at least eight repeated tests on every model, discarded its maximum and minimum values and obtained the average of the remaining results as the result of the experiment.

Figure 4 displays the outcomes of the comparative trials in DATA1.



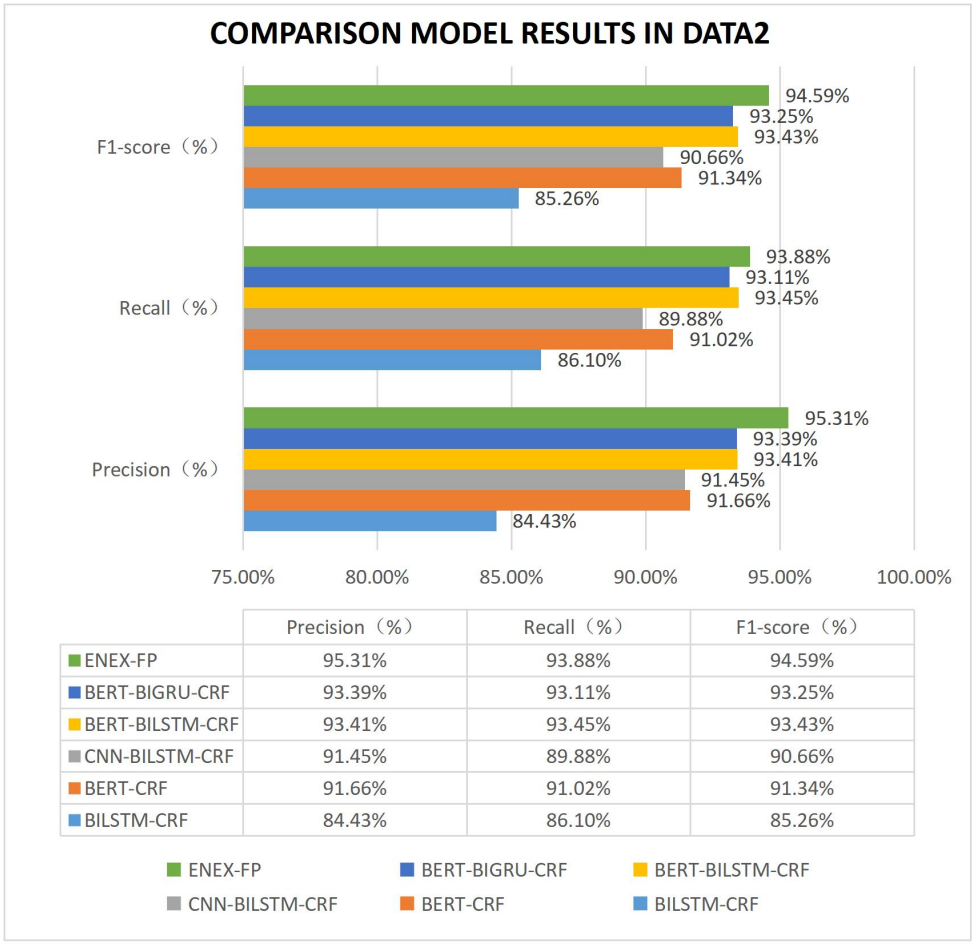
	Precision（%）	Recall（%）	F1-score（%）
ENEX-FP	93.67%	93.29%	93.47%
BERT-BIGRU-CRF	93.20%	92.34%	92.77%
BERT-BILSTM-CRF	92.06%	91.58%	91.81%
CNN-BILSTM-CRF	86.59%	88.94%	87.74%
BERT-CRF	89.67%	88.97%	89.32%
BILSTM-CRF	82.34%	84.67%	83.49%

Figure 4. Comparison model results in DATA1.

As seen from Figure 4, the model improved the Precision, Recall and F1-score in the recognition of address elements compared to other common NER models. The ENEX-FP model can improve the F1-score by 0.7% compared to the recent used BERT-BIGRU-CRF model and by 1.66% compared to the commonly used BERT-BILSTM-CRF model. Additionally, the model with the BERT outperforms the model without the BERT in terms of F1-score. This is due to the fact that the BERT model, in contrast to the other models, uses a word2vec splitter to create static word vectors before introducing the BERT model to pre-process the word vectors. As a result, the trial outcomes were all better than those of the model without word vector processing and met more criteria for satisfactory outcomes.

With simple cross-verification, according to 6:2:2 parsing, DATA2 was randomly split into a train set, a dev set and a test set. After that, the samples are mixed up and a new selection of the train set, dev set and test set is made using the 6:2:2 ratio to continue testing the model and training the data. After each sample shuffling, the number of address elements included in the train set, dev set and test set would change and the experimental result would also change relatively, but the difference was about 0.5%. We scrambled the text five times in total and conducted several experiments in each dataset. The final experimental result was determined as the average value of each model outcome. Figure 5 displays the outcomes of the comparative experiments in DATA2.

As seen from Figure 5, compared to other widely used NER models, the ENEX-FP's F1-score improved. Particularly, the F1-score can be improved by the ENEX-FP model by 1.34% compared to the recently BERT-BIGRU-CRF model and by 1.16% percent compared to the widely used BERT-BILSTM-CRF model.



	Precision（%）	Recall（%）	F1-score（%）
ENEX-FP	95.31%	93.88%	94.59%
BERT-BIGRU-CRF	93.39%	93.11%	93.25%
BERT-BILSTM-CRF	93.41%	93.45%	93.43%
CNN-BILSTM-CRF	91.45%	89.88%	90.66%
BERT-CRF	91.66%	91.02%	91.34%
BILSTM-CRF	84.43%	86.10%	85.26%

Figure 5. Comparison model results in DATA2.

Because BIGRU uses fewer parameters than BILSTM and converges more quickly, it has recently been widely used in NER problems. We compare the two models in two different datasets and discover that they are similar. In DATA1, the BERT-BIGRU-CRF F1-score is 0.96% higher than the BERT-BILSTM-CRF. In DATA2, the BERT-BILSTM-CRF F1-score is 0.18% higher than the BERT-BIGRU-CRF. After BERT word embedding, we eventually selected BILSTM as the model because BIGRU is an LSTM simplification. Three gates in the LSTM unit allow it to more effectively regulate gradient information propagation more than the BIGRU, reducing gradient disappearance and producing superior text characteristics. Based on the aforementioned models, our model makes full use of BERT's ability to extract features, not only using BERT output as word embedding but also aggregating BERT output feature vectors with BILSTM output. In order to increase feature accuracy and reduce overfitting, we also include a layer of dropout before the feature space changes, which eliminates some input data dimensions. To increase accuracy, we also use adversarial training and learning rate optimization. The experiment shows that ENEX-FP produces better experimental outcomes than other widely used address recognition models.

4.4.2. Ablation Experiments

In ablation experiments, the first group mainly trained the BERT-BILSTM-FP model and the ENEX-FP model to evaluate the effectiveness of the ENEX entity extractor and the FP module of the feature processor. We evaluated the models after each training. Table 4 displays the results of the ablation tests.

Table 4. Comparative results of the first ablation experiments.

Model	Precision	Recall	F1-Score
BERT-BILSTM-FP	92.25%	91.68%	91.96%
ENEX-CRF	92.38%	91.77%	92.04%
OUR MODELS	92.46%	92.06%	92.24%

The above experiments show that the ENEX is a little more effective than the baseline model BERT-BILSTM, where the F1-score can be improved by 0.28% in the comparison between ENEX-FP and BERT-BILSTM-FP. We also replace the FP with an existing CRF module and the F1-score can be improved by 0.2% in the comparison between ENEX-FP and ENEX-CRF. Therefore, the ENEN and FP model proposed in this paper work a little better in the recognition of addresses.

The second group focuses on training our proposed ENEX-FP model, as well as a model with adversarial training only, a model with learning rate decay and learning rate stratification, and a model with learning rate decay, learning rate stratification and adversarial training on top of this baseline model. Table 5 displays the results of the ablation tests.

Table 5. Comparative results of the second ablation experiments.

Model	Precision	Recall	F1-Score
ENEX-FP	92.46%	92.06%	92.24%
+ Counter training	92.88%	92.77%	92.82%
+ Learning rate optimization	93.02%	92.98%	93.00%
+ Learning rate optimization and adversarial training	93.67%	93.29%	93.48%

The results of the above experiments show that adding learning rate stratification to the model and learning rate decay adversarial training both improve the F1-score of the model. In the F1-score, the model with adversarial training only outperformed the ENEX-FP model by 0.58%, the model with learning rate optimization only outperformed the

model by 0.76% and the model both with learning rate optimization and using adversarial training outperformed the ENEX-FP model by 1.23%. Therefore, optimizing the model with learning rate and adversarial training gives the optimal F1-score, which is 1.23% higher than the baseline ENEX-FP model.

This paper also shows that the model after learning rate decay and learning rate stratification has better results than the baseline model. The learning rate parameter metric, which is a crucial parameter, decides whether or not the objective function will reach a local minimum. The target function can quickly arrive at a local minimum with the right learning rate. Finally, this study increases the learning rate by include adversarial training, which may enable the model to be further improved and, ultimately, become better optimized.

4.4.3. Data Enhancement Comparison Experiments

The experimental results of using the ENEX-FP model optimized by learning rate decay and hierarchical settings and adversarial training on the address element parsing dataset are shown in Table 6. From Table 6, we can recognize that the optimized model has different results in Precision, Recall and F1-score for different address labels, such as distance, prov, etc., which can reach almost 100%, but in poi, subpoi, community, etc., the recognition effect does not reach the expected value. The address element dataset is divided into sentences and then the individual sentences are randomly spliced and finally used as the training corpus together with the original sentences. In addition, this paper uses the collected address elements, after manual BIOES annotation, to add them to the DATA1 to obtain an expansion of the dataset. It also uses random replacement to replace the labeled entities in the corpus with them to obtain an augmented corpus. The experimental results of this paper are shown using the same model in Table 7.

Table 6. Experimental results of the address element parsing dataset.

	Precision (%)	**Recall (%)**	**F1-Score (%)**
poi	80.47%	84.29%	82.34%
town	96.71%	94.78%	95.73%
road	93.84%	95.22%	94.52%
floorno	94.31%	95.56%	94.93%
district	96.21%	93.84%	95.01%
prov	99.81%	99.90%	99.86%
subpoi	88.04%	83.41%	85.66%
roadno	98.88%	98.27%	98.58%
community	78.02%	82.77%	80.33%
houseno	98.68%	95.63%	97.13%
city	98.84%	97.71%	98.28%
devzone	97.59%	94.93%	96.24%
cellno	99.18%	98.37%	98.78%
assist	84.77%	88.26%	87.75%
intersection	86.74%	86.84%	86.78%
village_group	96.79%	97.87%	97.32%
distance	100.00%	100.00%	100.00%

Table 7. Experimental results for the data enhancement dataset.

	Precision (%)	Recall (%)	F1-Score (%)
poi	87.00%	84.03%	85.49%
town	93.31%	94.56%	93.93%
subpoi	87.67%	87.97%	87.81%
devzone	94.05%	99.75%	96.81%
roadno	97.85%	98.20%	98.03%
road	95.95%	96.95%	96.44%
houseno	98.24%	96.82%	97.52%
prov	99.68%	99.76%	99.72%
district	96.79%	96.38%	96.58%
community	82.82%	82.06%	82.43%
floorno	96.19%	95.73%	95.96%
city	98.35%	98.23%	98.29%
assist	85.45%	95.81%	90.33%
village_group	99.13%	97.62%	98.36%
cellno	92.17%	97.73%	94.86%
intersection	90.32%	87.50%	88.89%
distance	100.00%	100.00%	100.00%

From the results of the above experiments, this paper shows that there is a significant improvement in the effectiveness of the experiments after supplementation and random replacement of the corpus, especially for some low-precision tags. However, this data augmentation method is the most basic manual processing of the corpus, which is demanding in terms of time and effort. Therefore, it can be combined with the data augmentation in NER [31]. This is one of the focuses of future research in this paper.

5. Conclusions

The ENEX-FP model has the following improvements over other NER models. First of all, we proposed an ENEX entity extractor, which makes full use of the feature vectors extracted by BERT; BERT's output is not only embedded into the BILSTM model as a word vector but also the feature vector of BERT output is aggregated with the results of BILSTM. After the two vectors were combined, it was possible to address the impact of the left and right context's long-distance dependence, the lack of rich features, the text understanding and other issues and better entity feature extraction to fully obtain text features. Secondly, we propose the FP feature processor, which adds the dropout layer before the CRF layer and the full-connection layer, so it can round off the features of the input vector, remove some dimensions of the input data to improve feature accuracy and alleviate overfitting. FP can also perform feature space transformation on the vector to make conditional constraints on the text vector, solve some problems of BILSTM model output error and obtain the optimal address entity type. Finally, learning rate attenuation and layering are added to the model. The Adam learning rate optimizer is used to distinguish the learning rates of BERT and other layers, to improve the accuracy of the model. To increase the model's robustness and accuracy, the FGM adversarial training is added based on the learning rate.

Compared with other NER models, the ENEX-FP model has a noticeable improvement in address element recognition; especially after numerous trainings, the effect is more pronounced. This expected result shows the importance of the learning rate parameter and adversarial training. However, there are still some problems in this model. For example, there is no comparison experiment between the learning rate optimizer and adversarial training algorithm and the accuracy of some entity label recognition could be better. In short, the ENEX-FP model can identify addresses with high accuracy and the learning rate adjustment and adversarial training can also effectively improve the F1-score of the model. In addition, we believe that the characteristics obtained by the BERT model can be applied more widely. BERT has 12 layers of encoders, each of which takes the characteristics of the text, so these 12 layers of encoders may be processed further. This idea can probably be

added to any model that uses BERT. We will test this idea in future experiments. Finally, it is hoped that the problems mentioned above can be further solved in NER, obtain a better model, better deal with the identification of address elements and other application fields and achieve more excellent values.

Author Contributions: Propose research topics, M.L.; final review paper, M.L.; drafting the paper, Z.L.; funding acquisition, G.L. and M.Z.; Instructional support D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Plan of Youth Innovation Team Development of Colleges and Universities in Shandong Province (SD2019-161).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors appreciate all of the anonymous reviewers' insightful criticism and helpful recommendations.

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

References

1. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Int. J. Linguist. Lang. Resour.* **2007**, *30*, 3–26. [CrossRef]
2. Tang, X.; Huang, Y.; Xia, M.; Long, C. A Multi-Task BERT-BiLSTM-AM-CRF Strategy for Chinese Named Entity Recognition. *Neural Process. Lett.* **2022**, 1–21. [CrossRef]
3. Zhou, S.; Tan, B. Electrocardiogram soft computing using hybrid deep learning CNN-ELM. *Appl. Soft Comput.* **2020**, *86*, 105778. [CrossRef]
4. Li, J.; Sun, A.; Han, J.; Li, C. A Survey on Deep Learning for Named Entity Recognition. *arXiv* **2018**, arXiv:1812.09449.
5. Zou, H.; Liu, H.; Zhou, T.; Jiashun, L.; Zhan, Y. Short-Term Traffic Flow Prediction using DTW-BiGRU Model. In Proceedings of the 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Zhanjiang, China, 16–18 October 2020. [CrossRef]
6. Wang, Z.; Yang, B. Attention-based Bidirectional Long Short-Term Memory Networks for Relation Classification Using Knowledge Distillation from BERT. In Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 17–22 August 2020. [CrossRef]
7. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
8. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
9. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001; pp. 282–289.
10. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
11. Wang, X.; Zhu, Y.; Zeng, H.; Cheng, Q.; Zhao, X.; Xu, H.; Zhou, T. Spatialized Analysis of Air Pollution Complaints in Beijing Using the BERT+CRF Model. *Atmosphere* **2022**, *13*, 1023. [CrossRef]
12. Lin, J.; Liu, E. Research on Named Entity Recognition Method of Metro On-Board Equipment Based on Multiheaded Self-Attention Mechanism and CNN-BiLSTM-CRF. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–13. [CrossRef]
13. Wang, Y.; Wang, M.; Ding, C.; Yang, X.; Chen, J. Chinese Address Recognition Method Based on Multi-Feature Fusion. *IEEE Access* **2022**, *10*, 108905–108913. [CrossRef]
14. Dong, X.; Chowdhury, S.; Qian, L.; Guan, Y.; Yang, J.; Yu, Q. Transfer bi-directional LSTM RNN for named entity recognition in Chinese electronic medical records. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017. [CrossRef]
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 6000–6010.
16. Nesi, P.; Pantaleo, G.; Tenti, M. Geographical localization of web domains and organization addresses recognition by employing natural language processing, Pattern Matching and clustering. *Eng. Appl. Artif. Intel.* **2016**, *51*, 202–211. [CrossRef]
17. Grumiau, C.; Mostoufi, M.; Pavlioglou, S.; Verdonck, T. Address Identification Using Telematics: An Algorithm to Identify Dwell Locations. *Risks* **2020**, *8*, 92. [CrossRef]

18. Xu, L.; Li, S.; Wang, Y.; Xu, L. Named Entity Recognition of BERTBiLSTMCRF Combined with Self-attention. In *Web Information Systems and Applications. WISA 2021*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12999. [CrossRef]
19. Lv, X.; Xie, Z.; Xu, D.; Jin, X.; Ma, K.; Tao, L. Chinese Named Entity Recognition in the Geoscience Domain Based on BERT. *Earth Space Sci.* **2022**, *9*, E2021EA002166. [CrossRef]
20. Alyafi, B.; Tushar, F.I.; Toshpulatov, Z. Cyclical Learning Rates for Training Neural Networks with Unbalanced Datasets. In *Jmd in Medical Image Analysis and Applicationspattern Recognition Module*; University of Cassino and Southern Latium: Cassino, Italy, January 2018. [CrossRef]
21. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A.G. Averaging Weights Leads to Wider Optima and Better Generalization. *arXiv* **2018**, arXiv:1803.05407.
22. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
23. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
24. Miyato, T.; Dai, A.M.; Goodfellow, I. Adversarial Training Methods for Semi-Supervised Text Classification. *arXiv* **2016**, arXiv:1605.07725.
25. Graves, A.; rahman Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013. [CrossRef]
26. Sak, H.; Senior, A.; Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. *arXiv* **2014**, arXiv:1402.1128.
27. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
28. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2017**, arXiv:1706.06083.
30. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for Large-Scale machine learning. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
31. Ding, B.; Liu, L.; Bing, L.; Kruengkrai, C.; Nguyen, T.H.; Joty, S.; Si, L.; Miao, C. DAGA: Data Augmentation with a Generation Approach for Low-Resource Tagging Tasks. *arXiv* **2020**, arXiv:2011.01549.