MERE: A Deep Learning Architecture Using Multi-Fragment Ensemble for Relation Extraction

Hoang-Quynh Le and Duy-Cat Can
VNU University of Engineering and Technology,
44 Xuan Thuy Street, Cau Giay, Hanoi, Vietnam
lhquynh@vnu.edu.vn, catcd@vnu.edu.vn

Abstract

Overfitting is a significant challenge for deep learning models. Ensemble methods have been shown to effectively mitigate overfitting in a wide range of problems across different domains, especially within deep learning architectures. In this paper, we introduce an innovative deep learning model that integrates a multi-fragment ensemble mechanism to tackle the relation extraction problem. Our ensemble architecture is distinct from other models in building the base estimators using different data sizes and training them in an integrity deep learning model. Experiments on the Chemical-induced Disease relation and drug-drug interaction corpora show that the proposed model achieves competitive results, outperforming other models that do not consider inter-sentence relationships.

1 Introduction

Overfitting happens when a model performs well on its training data but struggles to generalize to new and unseen data. This is a common issue in deep learning, where the model shows low training errors but struggles with unseen data, indicating low bias but high variance. High variance, reflected by the difference between validation and training errors, means the model has poor predictive ability on the validation set. To deeply verify the model's capabilities and stability, we built a baseline deep learning model (as described in Section 3.1) to make a detailed analysis. This model would be used as base estimator in multi-fragment ensemble architecture. Figure 1 presents the results of running the baseline model 100 times on the same training dataset to analyze the standard deviation across multiple runs. The size of the training dataset varied from 10% to 100% of original training data. The difference of $F1$ between several runs is not too much (0.47% on original training data). However, when surveying P and R we can see that the model's stability is quite bad, the standard deviations of P and R are very high: 2.53% for P and 2.55% for R on original training data. These standard deviations increase when we decrease the size of training data.

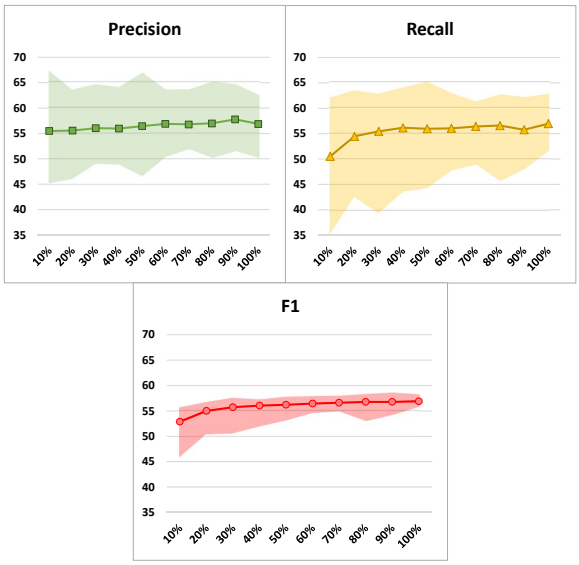

Figure 1: The range of baseline model's results on BC5 CDR test set. We trained the baseline models on various sizes of training dataset from 10% to 100%. The coloured fields represent the range of values in 100 runs, from the lowest to the highest result. The line shows the averaged results.

It is said that *'unity is strength'*, i.e., an individual can make a mistake in giving judgement, but the decision of the crowd can often produce a much more accurate (or at least less inaccurate) decision. Dietterich (2000) defined ensemble methods as the strategy of constructing multiple models (often referred to as 'weak learners' or 'base learners') and then classifying new data based on a weighted combination or vote of their predictions. This leads to the central hypothesis of ensemble methods: by correctly combining weak models, we can achieve more accurate and robust results. This approach

is highly effective in reducing variance, mitigating overfitting, and enhancing both stability and accuracy (Kowsari et al., 2019). Following the analysis above, our deep models are high variance and facing with overfitting problem. The parallel ensemble approach, with bagging being the most well-known method, is designed to reduce variance, thereby helping to prevent overfitting and enhance stability and accuracy..

In this work, we present MERE - a novel deep learning architecture using **M**ulti-fragment **E**nsemble for **R**elation **E**xtraction problem. MERE is the integrated model of deep learning estimators trained on different data sizes. The main contributions of this work are:

- We developed a deep learning model that utilizes advanced techniques and explored the variances and biases of the trained models across different data sizes.
- We enhanced the bagging ensemble method by integrating a multi-fragment ensemble into a deep learning model. The results showed that this enhanced model performs effectively on two benchmark datasets for relation extraction.

2 Related Work

Semantic relation extraction (RE) is a fundamental natural language processing task and has been studied extensively. Many approaches for RE have been developed, and recent advancements in deep learning have further fueled interest in applying neural architectures to this problem. The models based on convolutional neural networks (Zeng et al., 2014) and bidirectional long short-term memory networks (Zhang et al., 2015) are among the earliest research efforts to apply deep learning to RE. Recently, attention mechanisms have been widely adopted for RE tasks. Zheng et al. (2017) integrated an attention mechanism with Long Short-Term Memory networks to classify drug-drug interactions from the literature. BRAN (Verga et al., 2018) is a convolutional neural network with multi-layer attention designed to operate RE on abstract-level graph.

The ***bagging*** (standing for 'bootstrap aggregating') algorithm was introduced by Breiman (1996) (Breiman, 1996) as a voting ensemble method. In reality, we cannot build fully independent models for bagging, because it would require too much data. So, as its full name, bootstrap aggregation, bagging relies on the good 'approximate properties' of bootstrap samples (representativity and independence) to build almost independent models.

Bootstrapping is a sampling technique where subsets of observations are created from the original dataset by randomly drawing instances. Each bootstrap dataset effectively serves as a nearly independent sample from the true distribution, introducing expected diversity through the use of different datasets. In bagging, this bootstrap replica of the original training data is used to train a base model, and this process is repeated to generate multiple base models. Since the bootstrap datasets are approximately independent and identically distributed, the resulting base models exhibit similar properties. The ensemble's output does not alter the expected result but helps to reduce variance. In traditional bootstrapping, instances are drawn *with replacement*, so some data points may be repeated or omitted, with each instance having an equal probability of appearing in the new datasets.

Ensemble mechanisms frequently achieve top rankings in various natural language processing shared tasks, such as the Bionlp-2016 Bacteria Biotope event extraction (Mehryary et al., 2016) and SemEval-2017 ScienceIE (Ammar et al., 2017). Bagging has proven to be effective in a wide variety of problems in several domains including RE. Surdeanu et al. (2012) demonstrated that, in practice, a simple bagging model often achieves marginally better performance, by a few tenths of a percent, compared to training a single mention classifier on the latent mention labels produced in the last training iteration. In BRAN model (Verga et al., 2018), the simple ensemble technique also helped to boost the F1 for 2.2%. Yang et al. (2018) proposed an ensemble deep neural network model to extract relations via an Adaptive Boosting LSTMs with Attention model. Christopoulou et al. (2020) developed an ensemble deep learning model for extracting adverse drug events and medication relations from electronic health records. Weber et al. (2022) combined 10 pre-trained transformer-based models by averaging the predicted probabilities from each base model. Their findings revealed that ensembling models derived from a single base model outperformed those using different pre-trained language models on the DrugProt dataset. The Ensemble-of-Experts framework (Zhou et al., 2024) utilized a cascade voting mechanism to aggregate the capabilities of augmented models, facilitating rehearsal-free continual RE.

3 Proposed Model

In this work, we develop an baseline relation classification model and investigate the variants and biases of this model trained on different data size and different data distribution. Each baseline model f with its parameter θ is considered as a base estimator in a larger ensemble models. Instead of training base estimator independently, we construct a multi-fragment ensemble model on the top of these base estimators. The entire ensemble model is trained with a data masking block in a integrity deep learning model.

3.1 Base estimator

The overall architecture of our base estimator model is shown in Figure 2. Given a sentence and its dependency tree, we build our model on the sentence that contained two nominals and the shortest dependency path (SDP) between them. After an BERT-based embedding layer, each token on the sentence is represented by a vector. We gather the dependency features for each token from the dependency tree and apply a dual attention layer to obtain the context vector for each token. These sequence of vectors is then fed to a convolution layer with multi-kernel size to capture convolved features along the input sentence that can be used to determine which relation two nominals are of.

3.1.1 Input Representation

The main goal of this step is to transform each token into the vector space with D dimensions. For token representation, we utilized two types of word information, including:

- ***bioBERT*** (Lee et al., 2020): To model the sequential information on the original sentence, we use the pre-trained bioBERT along the sentence $\mathbf{S} = \{\mathbf{t}_i\}_{i=1}^{n}$ as follow:

$$\mathbf{H} = \text{bioBERT}(\mathbf{S}) = \{\mathbf{h}_i\}_{i=1}^{n} \quad (1)$$

- ***POS tag embeddings***: we embed the token's grammatical tag into a vector $\mathbf{t}_i$ using a randomly initialized look-up table and update this parameter on model learning phase. These parameters are shared between base estimators in the ensemble mechanism.

Finally, the concatenation between two presented vector is transformed into an D-dimensional vector to form the representation $\mathbf{x}_i \in \mathbb{R}^D$ of the token. I.e.,

$$\mathbf{x}_i = \tanh\left([\mathbf{h}_i \frown \mathbf{t}_i]\,\mathbf{W}^{\mathsf{x}} + \mathbf{b}^{\mathsf{x}}\right) \quad (2)$$

where $\mathbf{W}^{\mathsf{x}}$ and $\mathbf{b}^{\mathsf{x}}$ are trainable parameters of the network, $\frown$ denotes the concatenation of two vector.

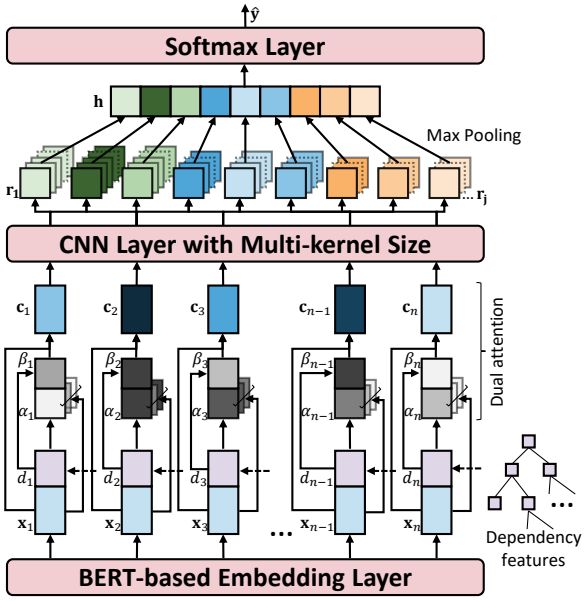

Figure 2: The architecture of base estimator in multi-fragment ensemble model. $\mathbf{x}_i$ is the presentation of the i^{th} token from the output of BERT encoder. d_i is the distance from the token i to the nearest node (token) on the SDP between two arguments. $\mathbf{c}$ is the context vector obtained by a scalar multiplication between attention weights α and β with token vector $\mathbf{x}$. CNN layer in this figure contains three different kernel sizes (three colors respectively).

3.1.2 Dual attention phase

We observe that, the original sentence contains many redundant information that does not help to classify the relation between to entity. Besides, information about the position of entities in the sentence also plays an important role in relational classification problem. However, the presentation of tokens using sequential modeling with bioBERT omits this information.

In this phase, we utilized the dependency tree and a dual attention architecture to capture the most important token. As illustrated in Figure 2, we employ two sequential attention layers on the sequence of input token, including:

- ***Multi-head self attention layer***: learns the importance weight for each token using its information in the relation with two nominals.

- ***Heuristic dependency attention layer***: calculates the attention score for each token using the distance information on the dependency tree.

Multi-head self-attention layer: We apply a multi-head self-attentive network on each token where the attention weights are calculated based on the concatenation of itself with two nominal BERT vectors $\mathbf{v}_1$ and $\mathbf{v}_1$, as follow:

$$\begin{aligned} \bar{\mathbf{X}}_1 &= \left\{\mathbf{x}_i^\frown \mathbf{v}_1\right\}_{i=1}^N = \left\{\bar{\mathbf{x}}_{1i}\right\}_{i=1}^N \\ \mathbf{e_1} &= \left\{\bar{\mathbf{x}}_{1i}\mathbf{W}^{\mathsf{e}} + b^{\mathsf{e}}\right\}_{i=1}^N = \left\{e_{1i}\right\}_{i=1}^N \\ \alpha_{1i}^{\mathsf{s}} &= \text{sigmoid}(e_{1i}) \end{aligned} \tag{3}$$

and

$$\begin{aligned} \bar{\mathbf{X}}_2 &= \left\{\mathbf{x}_i^\frown \mathbf{v}_2\right\}_{i=1}^N = \left\{\bar{\mathbf{x}}_{2i}\right\}_{i=1}^N \\ \mathbf{e_2} &= \left\{\bar{\mathbf{x}}_{2i}\mathbf{W}^{\mathsf{e}} + b^{\mathsf{e}}\right\}_{i=1}^N = \left\{e_{2i}\right\}_{i=1}^N \\ \alpha_{2i}^{\mathsf{s}} &= \text{sigmoid}(e_{2i}) \end{aligned} \tag{4}$$

where $\mathbf{W}^{\mathsf{e}} \in \mathbb{R}^{2D\times 1}$ and $b^{\mathsf{e}} \in \mathbb{R}$ are weight and bias term.

Heuristic dependency attention layer: The works of Can et al. (2019) and Le et al. (2018) demonstrated the effectiveness of the shortest dependency path on the task of RE. Therefore, we apply a heuristic attentive layer behind the multi-head self-attention layer based on the distances $d_1, d_2, ..., d_N$ to keep track of how close each token is to the nearest token on the SDP.

We heuristically choose a function to transform the distances $d_1, d_2, ..., d_N$ into the heuristic attention weight, as follow:

$$\alpha_i^{\mathsf{h}} = \text{sigmoid}(\beta d_i^2) \tag{5}$$

where $f(d) = \beta d^2$ is the activation function with $\beta = -0.03$.

The final dual-attentive context vector $\mathbf{c}_i$ of the target token is product of token's original vector with the calculated attention scores. I.e.,

$$\mathbf{c}_i = \alpha_{1i}^{\mathsf{s}} \times \alpha_{2i}^{\mathsf{s}} \times \alpha_i^{\mathsf{h}} \times \mathbf{x}_i \tag{6}$$

We further re-center and re-scale these context vector using a batch normalization (Ioffe and Szegedy, 2015) layer to keep model more stable and to accelerate the training procedure.

3.1.3 CNN layer with Multi-kernel size

The sequence of context vectors is gathered to form a matrix $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^n$. We build a common CNN text classification model on this $\mathbf{C}$. Generally, we define the vector $\mathbf{c}_{i:i+j}$ as the concatenation of j tokens, spanning from $\mathbf{c}_i$ to $\mathbf{c}_{i+j-1}$. I.e.,

$$\mathbf{c}_{i:i+j} = \mathbf{c}_i^\frown \mathbf{c}_{i+1}^\frown ... ^\frown \mathbf{c}_{i+j-1} \tag{7}$$

To extract local features from the context vector sequence, we perform k convolution operations with a region size of r on every possible window of r consecutive tokens to generate a convolved feature map. Subsequently, a max pooling layer collects the most significant features from each feature map. In other words, the convolutional layer calculates a feature f of the convolved feature vector using a filter size of r as described below:

$$f = \max_{0\le j\le N-r+1} \left[\mathbf{c}_{j:j+r}\mathbf{w}^{\mathsf{c}} + b^{\mathsf{c}}\right] \tag{8}$$

where $\mathbf{w}^{\mathsf{c}} \in \mathbb{R}^{D\times 1}$ and $b^{\mathsf{c}} \in \mathbb{R}$ are the trainable parameters of the convolutional layer. With k convolution operations, we could produced a convolved feature vector with k dimensions. In this work, to capture more n-grams features, we use various kernel size from 3 to 5 tokens.

A softmax classifier is then built on the output $\mathbf{f}$ of the convolutional layer to predict a K-class distribution over labels $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \text{softmax}\left(\mathbf{f}\mathbf{W}^{\mathsf{y}} + \mathbf{b}^{\mathsf{y}}\right) \tag{9}$$

where $\mathbf{W}^{\mathsf{y}} \in \mathbb{R}^{D\times K}$ and $\mathbf{b}^{\mathsf{y}} \in \mathbb{R}^K$ are parameter of the network to be learned.

3.2 The Multi-fragment Ensemble Deep Learning Architecture

3.2.1 The Overall Architecture

The overall of multi-fragment ensemble architecture is illustrated in Figure 3. To take advantage of the high variance of the baseline deep learning models, we build an ensemble model over the top of these base estimators.

Model Data Masking: Firstly, we created a mask $\mathbf{M}$ for each base estimator with a fixed probability α. This data mask has same size with the input dataset and is randomly initialized to the values 0 and 1, with the probability of value 1 being α. I.e.,

$$\mathbf{M} = \left[\begin{cases} 1 | \text{rand}() \le \alpha \\ 0, otherwise \end{cases}\right]^N \tag{10}$$

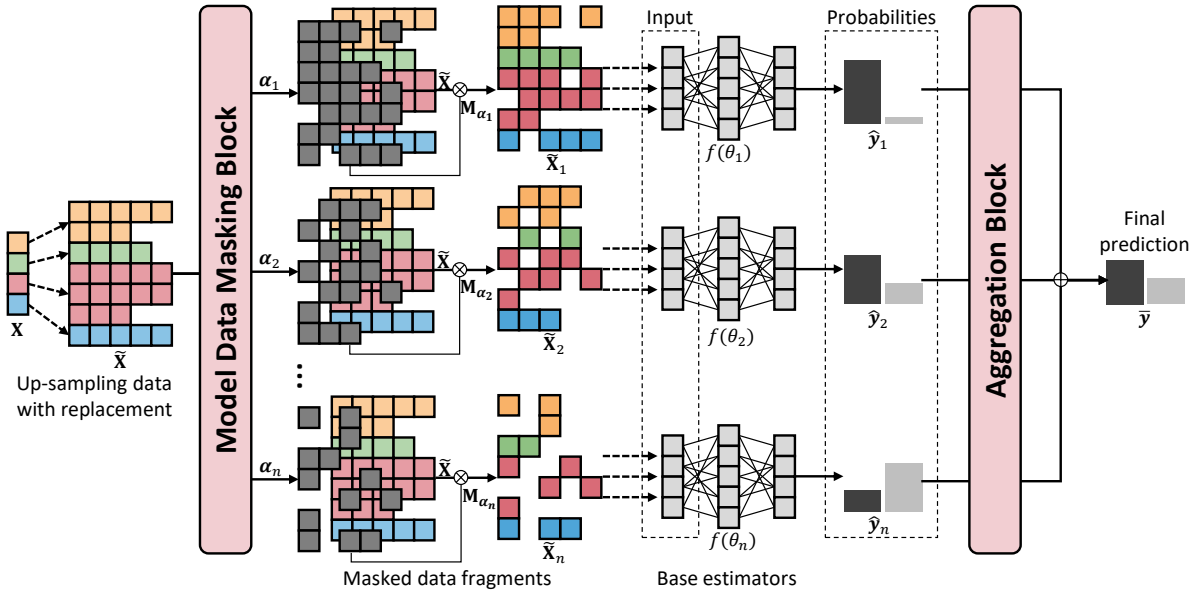

Figure 3: The multi-fragment ensemble deep learning architecture. Each colored square node is a example in dataset. The matrix-shaped figure is for demonstration purpose, the actual data is vector of examples. α_i is the probability of 1 value on data mask for the i^{th} estimator. $\bigotimes$ denotes the element-wise matrix multiplication. $\bigoplus$ denotes the element-wise average of two vector. $f(\theta)$ is the base estimator with parameters θ.

The mask will stay constant during the training phase to ensure that a base estimator is only trained on a part of input data.

Base Estimator on Masked Data Fragments: During the training phase, we used the 0-1 data mask to decide which model should contribute to the final prediction. When a sample is fed, the models with corresponding mask value of 1 will be included for prediction. These models will be updated simultaneously in the deep learning model through error backpropagation. Otherwise, the model with this value of 0, will be omitted in the set of estimator models. Therefore, in the error back-propagation step, the corresponding model could not be trained (the parameters of the corresponding model are not updated).

During the testing phase, the data mask is deactivated that all estimator will be used in the final prediction.

Aggregation block: For each instance, the prediction from each model is considered as a vote. There are various methods to combine the results from the base models using voting mechanisms. Two straightforward yet effective ensemble methods are the strict majority vote (Mehryary et al., 2016) and weighted sum of results (Verga et al., 2018). The soft-voting strategy uses the probabilities returned by base models then average them to get the final probabilities for prediction. In our experiments, the strict majority vote has yielded better results, so we use this approach along with a threshold-moving technique to enhance performance (Kambhatla, 2006; Collell et al., 2018).

In the experiment, we use different threshold α from 0.1 to 1.0 to construct data mask for different bootstrap data size. Each threshold α_i, we use k_i base estimators to form total of $K = k_1 + k_2 + ... + k_n$ predictions, denote as $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_i\}_{i=1}^{K} \tag{11}$$

We then aggregate the output of these models by soft- or hard-voting, as follow:

$$\bar{\mathbf{y}} = \frac{1}{K}\sum_{i=1}^{K} f(\hat{\mathbf{y}}_i) \tag{12}$$

with f is identical function for soft-voting whilst f is round function for the simple majority vote.

3.2.2 Number of base estimators

The number of base estimators is a hyper-parameter we need to decide for proposed ensemble model. Typically, this number is chosen heuristically by increasing the number of based estimators on development set until the $F1$ begins to stop showing improvement. Based on our hardware limitations, we heuristically select 100 as the number of base models to construct the ensemble for the BioCreative V dataset.

3.2.3 The size of bootstrap training data

In some cases, we maintain the original size of training data, but it's not a strict requirement. In our experiments, we allow the size of bootstrap data run from 10% to 100% of original training data size, with two approaches: with and without replacement. To conduct the with-replacement random data sampling experiment, we add an up-sampling data with replacement before the masking block of ensemble model.

An interesting observation in the experiment shows that the best results are achieved at the size of 70%, not 100%. This observation raises a question about choosing the suitable size of bootstrap training datasets: What size should we choose? And why don't we use different sizes to make new datasets that are more different, then gain diversity? In this work, we train the base models on different sizes of bootstrap data, from 50% to 100%.

3.2.4 Model training

In this ensemble architecture, we are less concerned if the individual model is overfitting of the training data. For this reason and efficiency, the individual models may grow deeper to have both high variance and low bias. Therefore, the early-stopping technique is no longer used. Base on the experiments, we fix 15 epochs of training on BioCreative V dataset. We also omit the dropout layer in the training phase of ensemble model. Other parameters are kept the same as when using a single baseline model.

4 Results and Discussion

Experiments were carried out using two benchmark RE corpora: the Chemical-induced Disease corpus (from BioCreative V shared task, 2015) and the Drug-Drug Interaction corpus (form Semveval DDI shared task, 2013). The Chemical-induced Disease (BC5 CDR) corpus (Li et al., 2016) comprises $1,500$ PubMed articles annotated with $3,116$ chemical-induced disease relationships. The Drug-Drug Interaction (DDI) corpus (Herrero-Zazo et al., 2013) includes 792 documents from the DrugBank database and 233 Medline abstracts, annotated with $5,028$ drug-drug interactions. For each dataset, we utilized official task evaluations based on $F1$ score, precision (P), and recall (R), focusing solely on actual relations at the abstract level. To assess the MERE model's effectiveness, we compared it against the average performance of 100 models trained on data sets equivalent in size to the original training data. The term *'Baseline without replacement'* refers to training all models on the same original dataset, with any performance differences attributed to model variations such as random seeds and initializations.

Model	Features	P	R	F1
Baseline	Baseline	58.72	57.50	58.1
BioCreative official results*	Co-occurrence	16.43	76.45	27.05
	Averaged result	47.09	42.61	43.37
	Best result	55.67	58.44	57.03
ASM	Dependency graph	49.00	67.40	56.80
hybridDNN	Syntactic features	62.15	47.28	53.70
	+ Context	62.39	47.47	53.92
	+ Position	62.86	47.47	54.09
ME+CNN	Sentence context	59.70	57.50	57.20
	+ Cross-sentence	60.90	59.50	60.20
	+ Post processing	55.70	68.10	61.30
BRAN	BRAN	55.60	70.80	62.10
	+ Data	64.00	69.20	66.20
	+ Ensemble	**65.40**	71.80	**68.40**
RbSP	Attentive augmented SDP	57.68	57.27	57.48
	+ Ensemble	58.78	57.20	57.98
	+ Post processing	52.38	72.65	60.78
MERE	**mf-60/REP**	*63.54*	*58.31*	*60.79*

**Provided by the BioCreative 2015 organizer.*
Results are reported in %.
Highest result in each column is highlighted in bold.

Table 1: The comparison of MERE with other comparative models on BC5 CDR corpus.

Model	Features	P	R	F1
2-phase classification-Hybrid kernel SVM	Heterogeneous set of feature	64.6	65.6	65.1
2-phase classification-SVM	Rich features	73.6	70.1	71.8
biLSTM + Attention	Position-aware attention + Pre-processing	**75.8**	**70.3**	**73.0**
RbSP	Attentive augmented SDP	54.0	57.1	55.5
MERE	**mf-10/REP**	*61.9*	*58.7*	*60.3*

Results are reported in % at abstract level.
Highest result in each column is highlighted in bold.

Table 2: The comparison of MERE with other comparative models on DDI corpus.

Performance comparison with comparative models:

We make the comparison between our proposed

models and comparative models on BC5 CDR corpus in Table 1. We evaluate the MERE model by comparing it with three types of competitors: (i) Baseline models (a base models, bootstrap data set were built with or with our replacement), (ii) The first-ranked result in the original challenges, (iii) State-of-the-art mode. For BC5 CDR corpus, we use three competitor results that only worked on intra-sentence RE: ASM (Approximate Subgraph Matching on the dependency graph (Panyam et al., 2018)), hybridDNN (LSTM and SVM (Zhou et al., 2016)) and RbSP (LSTM with attention mechanism (Can et al., 2019)). Two models capable of identifying inter-sentence relations are ME+CNN and BRAN. ME+CNN, which achieved top results in the BC5 CDR task, combines a CNN for intra-sentence relation extraction with a maximum entropy model for inter-sentence relations (Gu et al., 2017). BRAN employs a CNN with multi-layer attention to work on abstract-level graphs (Verga et al., 2018). MERE yields very competitive results when compared to other models that did not take into account the inter-sentence relationships (Zhou et al., 2016; Panyam et al., 2018; Gu et al., 2017; Can et al., 2019).

To provide a more comprehensive comparison and analyze the impact of the multi-fragment ensemble model on imbalanced data, we tested the model on the DDI corpus. In DDI corpus, the negatives take up 85.3%. The remaining 14.7% consisted of four different relation labels with 5%, 21%, 31% and 40% of positive data, equivalent to 0.74%, 3.09%, 4.56% and 5.88% of the total data. Comparative models include Chowdhury and Lavelli (2013), which employs a two-phase classification approach using a hybrid kernel SVM, where one classifier detects positive instances and another classifies them. Similarly, Raihani and Laachfoubi (2017) used a comparable SVM-based architecture. Zhou et al. (2018) combined binary and multi-class softmax functions with an RbSP model LSTM featuring an attention mechanism (Can et al., 2019). The experimental results and comparisons are presented in Table 2. We therefore conducted a grid search tuning and got the best results with $10\% - 50\%$ of negative data with MERRE models (called mf-10/REP configuration). The results are far below the comparative models. However, this result proves that the multi-fragment ensemble model has a better effect on unbalanced data. Compared to the baseline model, MERE helps to increase P by 7.9%, R by 1.6% and $F1$ by 4.8%. These improvements are significantly greater than the increases observed with the MERE model on the BC5 CDR corpus.

The MERE model with mf-60/REP mechanism takes $587,209$ seconds to train 100 RbSP base models (20 epochs per model) and 792 seconds to generate their outputs as well as vote for final output.

Multi-fragment analysis:

We also performed multiple experiments on the BC5 CRD corpus to thoroughly evaluate the multi-fragment mechanism, analyze the impact of bootstrap training data size, and compare the effects of using replacement versus non-replacement approaches for selecting training data. Table 3 and Figure 4 show the detailed experimental results on BC5 CDR corpus. The interesting observation is, using fewer data may bring better ensemble results. Using the traditional bagging ensemble mechanism, the best $F1$ archived at 70% data for replacement ensemble model (58.76%), and 50% data for the without-replacement ensemble (58.28%). Comparing to the size of 100% data, the result of the replacement ensemble model increases 0.66%, while the with out replacement ensemble model increases 0.55%. The MERE mechanism demonstrates its effectiveness, helps to boost the $F1$ of replacement ensemble model for 2.69% and without-replacement ensemble model for 0.97%.

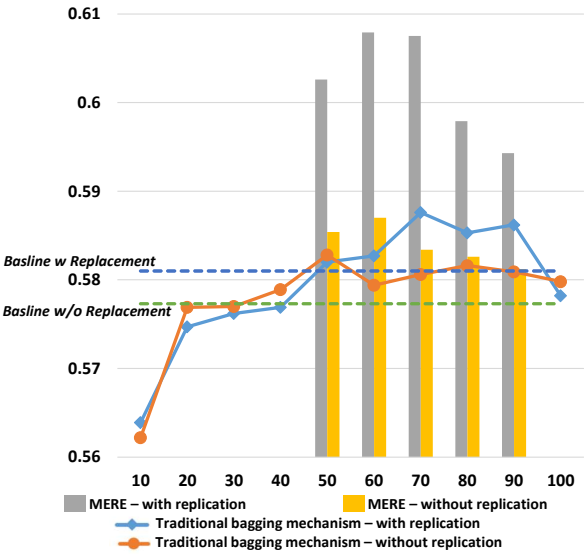

Figure 4: The changes of multi-fragment ensemble model's results with different sizes of training data.

Threshold-moving analysis:

Our model on the BC5 CDR corpus is a binary classifier with only one relation chemical-disease, and the other is negative. The threshold of the hard-

		With replacement*			W/o replacement*		
		P	R	F1	P	R	F1
Baseline		*58.72%*	*57.50%*	*58.10%*	*57.68%*	*57.77%*	*57.73%*
Traditional bagging mechanism. Using different size of bootstrap data†	10	57.38%	55.42%	56.39%	58.28%	54.30%	56.22%
	20	58.84%	56.17%	57.47%	59.30%	56.17%	57.69%
	30	58.33%	56.92%	57.62%	58.69%	56.73%	57.70%
	40	58.89%	56.55%	57.69%	58.29%	57.49%	57.89%
	50	58.63%	57.77%	58.20%	59.00%	57.58%	*58.28%*
	60	59.27%	57.30%	58.27%	58.80%	57.11%	57.94%
	70	59.68%	57.86%	*58.76%*	58.75%	57.39%	58.06%
	80	59.51%	57.58%	58.53%	58.85%	57.49%	58.16%
	90	59.81%	57.49%	58.62%	59.21%	57.02%	58.09%
	100	59.25%	56.45%	57.82%	58.78%	57.20%	57.98%
MERE - Multi fragment bootstrap‡	mf-50	62.87%	57.86%	60.26%	**60.56%**	56.64%	58.54%
	mf-60	**63.54%**	58.26%	**60.79%**	60.49%	**57.02%**	**58.70%**
	mf-70	63.40%	**58.31%**	60.75%	60.36%	56.45%	58.34%
	mf-80	61.73%	57.96%	59.79%	59.76%	56.83%	58.26%
	mf-90	61.74%	57.29%	59.43%	59.68%	56.64%	58.12%

The highest results in each column are highlighted in bold.
The highest F1 of traditional bagging mechanism are highlighted in underline.
*Bootstrap data sets were built with or without replacement.
†The size of bootstrap data compared to the original size of training data, run from 10% to 100%.
‡Multi-fragment bootstrap 'mf-n' means using several bootstrap sizes, run from n% to 100%

Table 3: MERE detailed results on BC5 CDR corpus.

voting mechanism for the ensemble model can be used in a flexible mode to improve the results (Kambhatla, 2006; Collell et al., 2018). Choosing a threshold at $k\%$ means that we assign an instance as positive if and only if at least k models agree to give this instance a positive label. Moving from 10% to 100%, a high threshold helps to increase P, but a small threshold increases the R. This threshold can be adjusted according to the characteristics of the data; for example, in cases of imbalanced data with a minority of positive classes, a lower threshold can be set to prioritize the positive class. In these experiments, we move the threshold and explore the changes of P, R and $F1$ as in Figure 5. The best $F1$ is archived at threshold 40%, a slight increase compared to the traditional majority vote (50%). Applied post-processing rules, we reach 53.57% for P, 74.84% for R and 62.44% for $F1$.

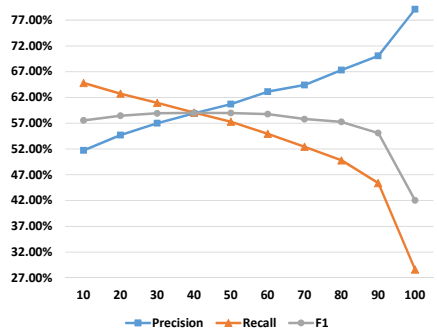

Figure 5: The changes $F1$ with different vote threshold on BC5 CDR corpus.

5 Conclusion

In this paper, we introduce MERE — the Multi-fragment Ensemble model — designed to address the overfitting challenges commonly associated with deep learning models while leveraging the benefits of ensemble mechanisms. MERE builds upon a novel base learning model that incorporates advanced deep learning techniques. Additionally, we enhance the model's variance and bias by experimenting with different data sizes, thereby validating the effectiveness of our multi-fragment ensemble approach. We assessed our model using two benchmark datasets: the chemical-induced Disease (BC5 CDR) corpus and the Drug-Drug Interactions (DDI) corpus, and compared its performance with leading state-of-the-art models. Additional experiments were conducted to evaluate the effectiveness of the model's main components. The results demonstrated both the advantages and robustness of our model. However, MERE only identifies relations within a single sentence, which explains the lower recall compared to systems that handle cross-sentence relations. We will address this limitation in future work.

Acknowledgments

We sincerely express our gratitude to Prof. Nigel Collier and Dr. Dang Thanh Hai for their support and encouragement during this work. We thank the reviewers for their comments and suggestions.

References

Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. 2017. The ai2 system at semeval-2017 task 10 (scienceie): semisupervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pages 592–596.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Duy-Cat Can, Hoang-Quynh Le, Quang-Thuy Ha, and Nigel Collier. 2019. A richer-but-smarter shortest dependency path with attentive augmentation for relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2902–2912, Minneapolis, Minnesota. Association for Computational Linguistics.

Md. Faisal Mahbub Chowdhury and Alberto Lavelli. 2013. Fbk-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013)*, pages 351–355. Association for Computational Linguistics.

Fenia Christopoulou, Thy Thy Tran, Sunil Kumar Sahu, Makoto Miwa, and Sophia Ananiadou. 2020. Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods. *Journal of the American Medical Informatics Association*, 27(1):39–46.

Guillem Collell, Drazen Prelec, and Kaustubh R Patil. 2018. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*, 275:330–340.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Jinghang Gu, Fuqing Sun, Longhua Qian, and Guodong Zhou. 2017. Chemical-induced disease relation extraction via convolutional neural network. *Database (Oxford)*, 2017:bax024.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 448–456. JMLR. org.

Nanda Kambhatla. 2006. Minority vote: at-least-n voting improves recall for extracting relations. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 460–466. Association for Computational Linguistics.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Hoang Quynh Le, Duy-Cat Can, Sinh T Vu, Thanh Hai Dang, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Large-scale exploration of neural relation classification architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2266–2277.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database Oxford*, 2016:baw068.

Farrokh Mehryary, Jari Björne, Sampo Pyysalo, Tapio Salakoski, and Filip Ginter. 2016. Deep learning with minimal training data: Turkunlp entry in the bionlp shared task 2016. In *Proceedings of the the 4th BioNLP Shared Task Workshop*, pages 73–81. Association for Computational Linguistics.

Nagesh C. Panyam, Karin Verspoor, Trevor Cohn, and Kotagiri Ramamohanarao. 2018. Exploiting graph kernels for high performance biomedical relation extraction. *Journal of biomedical semantics*, 9(1):7.

Anass Raihani and Nabil Laachfoubi. 2017. A rich feature-based kernel approach for drug-drug interaction extraction. *International Journal of Advanced Computer Science and Applications*, 8(4):324–330.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 872–884.

Leon Weber, Mario Sänger, Samuele Garda, Fabio Barth, Christoph Alt, and Ulf Leser. 2022. Chemical–protein relation extraction with ensembles of carefully tuned pretrained language models. *Database*, 2022:baac098.

Dongdong Yang, Senzhang Wang, and Zhoujun Li. 2018. Ensemble neural relation extraction with adaptive boosting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4532–4538.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 73–78.

Wei Zheng, Hongfei Lin, Ling Luo, Zhehuan Zhao, Zhengguang Li, Yijia Zhang, Zhihao Yang, and Jian Wang. 2017. An attention-based effective neural model for drug-drug interactions extraction. *BMC Bioinformatics*, 18(1):445.

Deyu Zhou, Lei Miao, and Yulan He. 2018. Position-aware deep multi-task learning for drug–drug interaction extraction. *Artificial intelligence in medicine*, 87:1–8.

Huiwei Zhou, Huijie Deng, Long Chen, Yunlong Yang, Chen Jia, and Degen Huang. 2016. Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database*, 2016:baw048.

Shen Zhou, Yongqi Li, Xin Miao, and Tieyun Qian. 2024. An ensemble-of-experts framework for rehearsal-free continual relation extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1410–1423.