Dynamic Modeling Cross-Modal Interactions in Two-Phase Prediction for Entity-Relation Extraction

Shan Zhao, Minghao Hu, Zhiping Cai, and Fang Liu, *Member, IEEE*

***Abstract*— Joint extraction of entities and their relations benefits from the close interaction between named entities and their relation information. Therefore, how to effectively model such cross-modal interactions is critical for the final performance. Previous works have used simple methods, such as label-feature concatenation, to perform coarse-grained semantic fusion among cross-modal instances but fail to capture fine-grained correlations over token and label spaces, resulting in insufficient interactions. In this article, we propose a dynamic cross-modal attention network (CMAN) for joint entity and relation extraction. The network is carefully constructed by stacking multiple attention units in depth to dynamic model dense interactions over token-label spaces, in which two basic attention units and a novel two-phase prediction are proposed to explicitly capture fine-grained correlations across different modalities (e.g., token-to-token and label-to-token). Experiment results on the CoNLL04 dataset show that our model obtains state-of-the-art results by achieving 91.72% F1 on entity recognition and 73.46% F1 on relation classification. In the ADE and DREC datasets, our model surpasses existing approaches by more than 2.1% and 2.54% F1 on relation classification. Extensive analyses further confirm the effectiveness of our approach.**

***Index Terms*— Entity, interactions, modalities, relations.**

I. Introduction

EXTRACTION of entities and their relations from unstructured raw texts has attracted increasing attention due to its important application on knowledge base population, information retrieval, and question answering [1]. Given a sentence, the task aims to find the location and type of mentioned entities and further detect semantic relations among those entities. For example, in Fig. 1, "*Tanya*" is a person entity (Peop), while "*Shabds Hospital*" and "*Gainesville*" are two location entities (Loc) connected by a "*Located In*" relation.

Manuscript received 4 August 2020; revised 13 March 2021 and 29 June 2021; accepted 11 August 2021. Date of publication 25 August 2021; date of current version 1 March 2023. This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1405702. *(Corresponding author: Fang Liu.)*

Shan Zhao is with the School of Design, Hunan University, Changsha 410073, China, and also with the College of Computer, National University of Defense Technology, Changsha 417003, China (e-mail: zs50910@mail.ustc.edu.cn).

Minghao Hu is with the PLA Academy of Military Science, Beijing 100000, China (e-mail: huminghao16@gmail.com).

Zhiping Cai is with the College of Computer, National University of Defense Technology, Changsha 417003, China (e-mail: zpcai@nudt.edu.cn).

Fang Liu is with the School of Design, Hunan University, Changsha 410073, China (e-mail: fangl@hnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3104971.

Digital Object Identifier 10.1109/TNNLS.2021.3104971

Traditionally, the task of extracting semantic relations between entities is decoupled into a pipeline of two separated subtasks, namely, named entity recognition (NER) [2]–[4] and relation extraction (RE) [5]. Since named entities interact closely with their relation information (two location entities are usually linked with a "*Located In*" relation), joint models that simultaneously learn NER and RE have been proposed and have achieved promising results [6]–[9]. However, joint models only capture such *cross-modal* interaction by learning shared representations via multitask training but fail to take label information into account, which turns out to be a significant limitation. For example, if the model knows that "*Shabds Hospital*" and "*Gainesville*" are location entities beforehand, it can easily infer there may exist a "*Located In*" relation between them.

To overcome the problem of insufficient cross-modal interactions, some works [6], [8] propose to enhance downstream RE performance by leveraging label information extracted from the upstream NER process. These approaches adopt simple feature concatenation to fuse label information into contextualized representations, which results in promising performance improvement. However, such naive methods can only learn coarse-grained interactions of cross-modal instances via token-level semantic fusion but cannot effectively infer the correlation between each token and each tagging label (e.g., it is beneficial that "*Shabds Hospital*" is aware of "*Gainesville*" being assigned with a "*B-LOC*" tag). Moreover, token-level self-correlation is also important for both NER and RE, which has been ignored by previous RNN- or CNN-based models [10], [11]. For example, the fact that "*Shabds Hospital*" is highly relevant to "*Gainesville*," but less related with "*Tanya*" is helpful for entity recognition and relation classification. Furthermore, most of the proposed models do not focus on weighting the losses of the two tasks, which ignores the relative weighting between each task loss. Correct weighting losses are of importance for joint models.

To address the above issues, we propose a dynamic cross-modal attention network (CMAN) for joint entity and RE. Inspired by multimodal learning in computer vision [12], we view token and label spaces as two different modalities and attempt to model dense cross-modal interactions over these two spaces. To achieve this, we first design two basic attention units: a BiLSTM-enhanced self-attention (BSA) unit that aims to model intramodal interactions across different tokens (token-to-token); and a BiLSTM-enhanced label-attention (BLA) unit that is capable of modeling intermodal

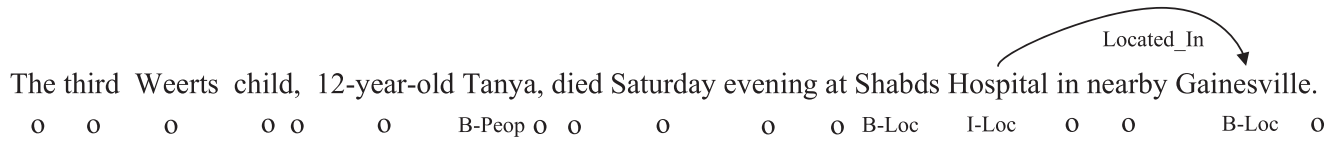

Fig. 1. Example from the CoNLL04 dataset, where the goal is to identify mentioned entities and corresponding relationships in the sentence.

interactions (label-to-token). BSA is able to build direct connections between two arbitrary tokens in a sentence despite their distances, while BLA explicitly leverages label-space information to enhance contextualized token representations. Then, we design a novel two-phase prediction, which dynamically controls label feature contributions and not only can take into account the interactions between token and label features in RE but also in NER. Given the token-label embeddings, CMAN first utilizes BSA units to generate self-aware token features and label information in the first-phase prediction, and then, in second-phase prediction, we construct the entire model by carefully stacking multiple attention units to form a deep network architecture for fully capturing cross-modal interactions, where gold label information is available only during training and is predicted by first-phase during inference. Next, we introduce homoscedastic uncertainty [13] to automatically weighting the losses of two-phase prediction. By adding homoscedastic uncertainty feed to losses, these homoscedastic uncertainties can learn a relative weighting automatically from the data and are robust to the weight initialization. Finally, we conducted extensive experiments on CoNLL04, ADE, and DREC datasets to evaluate the proposed model. In CoNLL04, our model obtains state-of-the-art results by achieving 91.72% and 73.46% F1 on entity recognition and relation classification, respectively. Moreover, our model surpasses existing approaches by more than 2.1% and 2.54% F1 score on relation classification in the ADE and DREC datasets, respectively.

We note that a shorter conference version of this article [14] is accepted for IJCAI 2020. Our initial conference paper only performs dense cross-modal interaction learning in relation classification. However, we argue that taking into account the interactions between token and label features is beneficial for entity recognition. To achieve this, we design a cross-modal interaction in two-phase prediction. Moreover, we propose a label gate to control label feature contributions. Finally, we introduce homoscedastic uncertainty to automatically weighting losses. This article also provides additional analysis of more datasets.

II. Related Work

A. Joint Entity-Relation Extraction

Due to the existence of close interactions between entity recognition and relation classification, joint models that simultaneously learn NER and RE have outperformed pipelined methods [16] by a large margin. Specifically, Miwa and Bansal [6] employ bidirectional tree-structured RNNs, which extracts relationships between entities based on word order information and dependent tree structure information. Wang *et al.* [10] extract relations using multilevel attention CNNs. Then, a novel tagging scheme is proposed to convert the joint extraction problem into a sequence labeling problem [17], which is usually solved by RNNs-based decoding strategies. Yet, this tagging scheme is difficult to handle multiple relationships, which are relatively rare in many datasets. Therefore, Bekoulis *et al.* [8] propose a multihead mechanism to support the prediction of multiple relationships. Compared to these approaches that adopt either RNNs or CNNs-based architecture, our model consists of cascaded attention units that combine bidirectional LSTM (BiLSTM) with multihead attention [18] to better capture correlations between any two modal instances despite their relative distance.

B. Label-Space Information

Recently, label information has been applied to NLP tasks and achieves ideal results. Specifically, label knowledge has been exploited in the text classification task [19]. Moreover, Cui and Zhang [20] introduce label embeddings to the NER task. However, label-space information has not been carefully studied in joint entity and RE. Prior approaches [6], [8] exploit a naive way, such as feature concatenation to utilize coarse-grained labels. In contrast, we aim to model dense cross-modal interactions over token-label spaces, which delivers significantly better performance.

C. Multimodal Learning

Multimodal learning is widely explored in computer vision and natural language processing [21], [22]. A typical task is a visual question answering (VQA) [23], which requires the model to perform fine-grained semantic understanding of both the image and the question. For example, Yu *et al.* [12] propose a modular attention mechanism to capture the interactions of multimodal instances (image and question). Inspired by recent advancements in this field, we regard token and label spaces as two different modalities and attempt to capture cross-modal interactions between them.

D. Weighting Losses of Joint Models

Prior approaches to simultaneously learning joint models use a native weighted sum of losses, where the loss weights are uniform or manually tuned. Bekoulis *et al.* [8] and Li *et al.* [24] add the two parts NER and RE of the loss directly as the final loss. Li *et al.* [9] introduce parameter λ to control the tradeoff between the two objectives. Kendall *et al.* [13] propose a principled approach to multitask

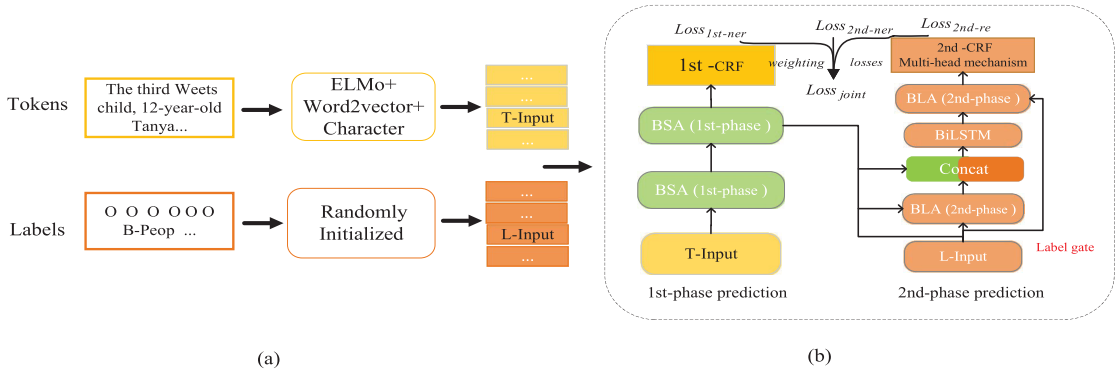

Fig. 2. Overall flowchart of CMAN. Tokens and labels are first represented as distributed representations from multiple perspectives. Then, self-aware token features and label information are obtained in the first-phase prediction. Finally, in second-phase prediction, a deep network architecture based on two attention units is then designed to utilize gold label information during training and predicted labels at inference time. A CRF [15] and a multihead mechanism [8] are also used to predict entities and their relations in second-phase prediction. T and L denote token and label, respectively. (a) Distributed representation in token-label spaces. (b) Dynamically modeling dense cross-modal interactions.

deep learning, which weighs multiple loss functions by considering the homoscedastic uncertainty of each task and make a good effect on image recognition. In this article, we try to introduce the homoscedastic uncertainty of NER and RE to weighting losses.

III. Proposed Model

In this section, we introduce the dynamic CMAN in detail, which is shown in Fig. 2. We first obtain fixed-dimensional representations of token and label from different perspectives (see Section III-A). Then, we design a BSA unit and a BLA unit (see Section III-B). These two units are built to explicitly leverage token-label spaces information for modeling cross-modal interactions (e.g., token-to-token and label-to-token). After that, we adopt several BSA units to extract self-aware token features and a conditional random filed (CRF) [15] to predict entities labels in first-phase prediction (see Section III-C). Moreover, in second-phase prediction, a dynamic network architecture based on these two units is carefully designed to utilize gold label information during training and predicted labels at inference time, and another CRF and a multihead mechanism [8] are used to predict entities and their relations (see Section III-D). Finally, we introduce homoscedastic uncertainty [13] to automatically weighting the losses of two-phase prediction (see Section III-F).

A. Representations in Token-Label Spaces

As mentioned above, sequence tokens and tagging labels are viewed as two different modalities and, therefore, can be represented with different distributed representations. In the following, we will present how to construct these representations.

1) Token Representations: Word embeddings are used to map discrete words into continuous input vectors. Given a sentence containing n words, we map each token in the sentence to a real-valued embedding to express its semantic and syntactic meaning. Besides, we also utilize character embeddings, which is obtained by encoding character sequences with a BiLSTM. Then, the input of each token is a concatenation of character embeddings, word embeddings, and ELMo embeddings [25]. In this way, a sequence of input vectors $X \in \mathbb{R}^{n*d_w}$ can be obtained, where d_w is the token embedding dimension.

2) Label Representations: We adopt the Beginning, Inside, Outside (BIO) encoding scheme for NER, as illustrated in Fig. 1. Motivated by Miwa and Bansal [6], tagging labels are represented with randomly initialized vectors that are fine-tuned during training, thus yielding a sequence of label vectors $L \in \mathbb{R}^{n*d_l}$, where d_l is the label embedding dimension. Notice that ground-truth labels are used only during training, while predicted labels are utilized at inference time (see more details in Section III-E).

B. Two Basic Attention Units

We first present a general architecture that contains BiLSTM and multihead attention for encoding and attending any arbitrary sequence. Then, we build two attention units based on this architecture to capture dense correlations among token-label spaces, namely, a BSA unit and a BLA unit.

General Architecture: BiLSTM is superior in build contextualized representations for various NLP tasks, as shown in [11]. Hence, we utilize BiLSTM as the basic encoding component. Given a sequence of input vectors $X = [x_1, \ldots, x_n]$, a BiLSTM can be used to output hidden representations $H \in \mathbb{R}^{n*2d}$ as

$$\overrightarrow{h_t} = \overrightarrow{\text{LSTM}}(x_t, \overrightarrow{h_{t-1}}) \quad (1)$$

$$\overleftarrow{h_t} = \overleftarrow{\text{LSTM}}(x_t, \overleftarrow{h_{t+1}}). \quad (2)$$

Then, outputs of the forward and backward LSTM are concatenated at each timestep to get the final LSTM output

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right]. \quad (3)$$

Multihead attention [18] has proven to be effective for capturing long-range dependencies by explicitly attending to all positions. Therefore, we apply the multihead attention as the basic attending component for capturing arbitrary correlations. Typically, scaled dot-product is chosen as the similarity scoring function in multihead attention mechanism. Given input queries $Q \in \mathbb{R}^{n*d}$, keys $K \in \mathbb{R}^{n*d}$, and values

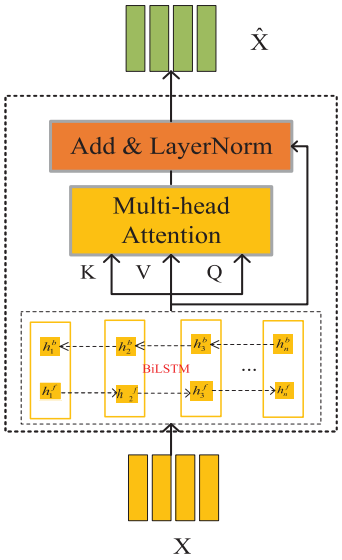

Fig. 3. BSA: it is composed of a BiLSTM layer and a self-attention layer, which aims to model intramodal interactions across different tokens.

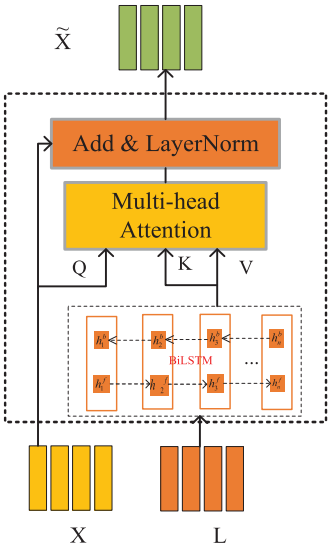

Fig. 4. BiLSTM-enhanced interattention (BIA): it contains a BiLSTM layer and an interattention layer, which is used to model intermodal interactions between label space and token space.

$V \in \mathbb{R}^{n*d}$, the matrix of outputs is computed using the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{4}$$

Multihead attention allows the model to jointly attend to information from different representation subspaces at different positions. It first maps the matrix of input vectors to query, key, and value matrices by using different linear projections. Then, z parallel heads are employed to perform attention operation in different parts of channels

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{5}$$

$$T = \text{Concat}(\text{head}_1, \ldots, \text{head}_z)W^o \tag{6}$$

where $W_i^Q \in \mathbb{R}^{d*d/z}$, $W_i^K \in \mathbb{R}^{d*d/z}$, $W_i^V \in \mathbb{R}^{d*d/z}$, and $W_i^o \in \mathbb{R}^{d*d}$ are trainable parameter matrices. Finally, a residual connection [26] along with layer normalization [27] is applied on H and T to produce the final output features

$$O = \text{LayerNorm}(T + Q). \tag{7}$$

1) BiLSTM-Enhanced Self-Attention: The BSA unit [see Fig. 3(a)] is designed to model token-to-token self-correlations. Taking one group of input token features $X = [x_1, \ldots, x_n]$, the BiLSTM is first used to capture rich contextual information over token space. Next, the multihead attention receives the encoded hidden representations $H = [h_1, \ldots, h_n]$, further learns the pairwise relationship between the paired sample $\langle h_i, h_j \rangle$ within H, and, finally, outputs attended output features by weighted summation across all instances. In summary, the computation of BSA unit can be defined as $\hat{X} = \text{BSA}(X)$.

2) BiLSTM-Enhanced Label-Attention: The BLA unit (see Fig. 4) is capable of modeling intermodal interactions from label space to token space. It takes two groups of features $X \in \mathbb{R}^{n*d_w}$ and $L \in \mathbb{R}^{n*d_l}$ as inputs. The BiLSTM component is first used to encode label features as $\tilde{L} = \text{BiLSTM}(L)$. Next, the BLA unit models the pairwise relationship between each paired sample $\langle x_i, \tilde{l}_i \rangle$ within X and $\tilde{L}$. Notice that we set token features X as query and set encoded label features $\tilde{L}$ as key and value so that each token can be fused with relevant label information. The calculation of the BLA unit can be summarized as $\tilde{X} = \text{BLA}(X, L)$.

C. First-Phase Prediction

Since there is no label information during entity recognition, we first pass token features X into several BSA units in a recursive manner

$$X^i = \text{BSA}^i(X^{i-1}) \quad \forall i \in [1, m] \tag{8}$$

where X^0 is the set as X and m is the number of BSA units.

Then, based on the extracted self-aware token features, we predict the entity by a conditional random field (CRF) [15]. Take self-aware token features $X^m = [x_1^m, \ldots, x_n^m]$ as inputs and a sequence of predicted taggings $A = [a_1, \ldots, a_n]$ as outputs. Let A' denote an arbitrary label distribution sequence (i.e., BIO tagging scheme); the probability of the label sequence A can be calculated using a softmax function

$$\Pr(A|X^m) = \frac{\prod_{i=1}^{n} \varphi_n(a_{n-1}, a_n, X^m)}{\sum_{a' \in A'} \prod_{i=1}^{n} \varphi_n\left(a'_{n-1}, a'_n, X^m\right)} \tag{9}$$

where $\varphi_n(a_n, a_{n-1}, L) = \exp(W_n X^m + b_n)$ is the potential function and W_n and b_n are the weight vector and bias, corresponding to label pair (a_{n-1}, a_n), respectively.

D. Second-Phase Prediction

Modeling dense cross-modal interaction learning (token-to-label) is only available in RE in one phase NER. Since extraction of entities in first-phase can provide the predicted label information, we, therefore, present a novel second-phase prediction to dynamically control label feature contributions and model dense cross-modal interaction learning in NER and RE by feeding input features into a deep network that contains carefully designed cascaded attention units. Finally, we employ

another CRF and a multihead mechanism [8] to predict entities and their relations.

1) Label Gate: Similar to previous work [6], [8], our model uses the entity tags as input to relation classification by learning label embeddings during inference. However, the entity recognition results are not always correct since they are predicted by the model during inference. Especially, for the dataset with the poor NER effect, the problem is particularly serious. Therefore, it is important to incorporate a gate to dynamically control the contribution of label features. Following the practice in previous work [28]–[30], we design a label gate by combining the information from the above self-aware token features X^m and encoded label representations $\tilde{L}$. We replace encoded label representations $\tilde{L}$ with upade label representations $\tilde{L}'$ in the BLA unit, which can be calculated as follows:

$$k = \sigma(W_x X^m + W_l \tilde{L} + b_l) \tag{10}$$

$$\tilde{L}' = k\tilde{L} \tag{11}$$

where W_l and W_x are parameters. σ is the sigmoid activation function.

2) Dense Cross-Modal Interaction Learning: Take the extracted self-aware token features X^m and label features L as inputs, and utilize a BLA unit to obtain initial label-aware token representations as

$$\tilde{X}^1 = \text{BLA}^1(X^m, L). \tag{12}$$

Next, we apply a concatenation-style residual connection [26] on previous input and output token features and further use another BiLSTM to fuse their semantic meanings

$$\tilde{X}^2 = \text{BiLSTM}([X^m; \tilde{X}^1]). \tag{13}$$

Finally, taking $\tilde{X}^2$ and L as inputs, we apply another BLA unit to capture deep cross-modal correlations to form the final label-aware token features as

$$\tilde{X}^3 = \text{BLA}^2(\tilde{X}^2, L). \tag{14}$$

Now, $\tilde{X}^3$ is capable of capturing rich cross-modal interactions and is suitable for the task of entity recognition and relation classification.

3) Entities-Relations Prediction: Due to considering the interaction between each token and each tagging label in entity recognition, we adopt another CRF to extract the final entity. Taking fusing representation $\tilde{X}^3$ as inputs and a sequence of predicted taggings $\tilde{A} = [\tilde{a}_1, \ldots, \tilde{a}_n]$ as outputs. Let $\tilde{A}'$ denote the set of tagging labels (i.e., BIO scheme); the probability of the tagging sequence can then be calculated as follows:

$$\Pr(\tilde{Y}|\tilde{X}^3) = \frac{\prod_{i=1}^{n} \varphi(\tilde{a}_{i-1}, \tilde{a}_i, \tilde{X}^3)}{\sum_{\tilde{a}' \in \tilde{A}'} \prod_{i=1}^{n} \varphi(\tilde{a}'_{i-1}, \tilde{a}'_i, \tilde{X}^3)} \tag{15}$$

where $\varphi(\tilde{a}_{i-1}, \tilde{a}_i)$ is the transition score from $\tilde{a}_{i-1}$ to $\tilde{a}_i$ calculated by $\exp(\tilde{W}_\varphi \tilde{X}^3 + \tilde{b}_\varphi)$, and $\tilde{W}_\varphi$ and $\tilde{b}_\varphi$ are trainable weight and bias.

For RE, we utilize the multihead mechanism for predicting relation, of which details can be found from [8]. Suppose that fusing features $\tilde{X}^3 = [\tilde{x}_1^3, \ldots, \tilde{x}_n^3]$ are given as inputs, and C is a set of relation labels. The idea of this mechanism is to predict a score for each tuple (w_i, w_j, c_k), where w_i is the head token, w_j is the tail token, and c_k denotes the kth relation between them. Note that each pair of tokens $\langle w_i, w_j \rangle$ can have multiple heads, where each head computes a score for one relation. We calculate the score between w_i and w_j given a relation c_k as follows:

$$s(\tilde{x}_i^3, \tilde{x}_j^3, c_k) = V_k \tanh(U_k \tilde{x}_i^3 + W_k \tilde{x}_j^3 + b_k) \tag{16}$$

where $V_k \in \mathbb{R}^{\tilde{d}}$, $W_k \in \mathbb{R}^{\tilde{d}*2d}$, $U_k \in \mathbb{R}^{\tilde{d}*2d}$, and $b_k \in \mathbb{R}^{\tilde{d}}$ are parameters for the kth relation, and $\tilde{d}$ is the intermediate hidden size. Next, the probability of token w_i selected as the head of token w_j with the relation c_k is calculated as

$$\begin{aligned} &\Pr(\text{head} = w_i, \text{relation} = c_k | w_j) \\ &\quad = \sigma\left(s(\tilde{x}_i^3, \tilde{x}_j^3, c_k)\right) \end{aligned} \tag{17}$$

where σ stands for the sigmoid function.

E. Training and Inference

During training, we optimize the parameters of the model by minimizing the following conditional likelihood for NER:

$$\mathcal{L}_{\text{1st-ner}} = -\log\Pr(Y|X^m) \tag{18}$$

$$\mathcal{L}_{\text{2nd-ner}} = -\log\Pr(\Pr(\tilde{Y}|F)). \tag{19}$$

As for RE, the cross-entropy loss is applied for training

$$\mathcal{L}_{\text{re}} = \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{k=1}^{o} -\log\Pr(\text{head} = w_i \text{relation} = c_k | w_j) \tag{20}$$

where o is the number of relations. For the joint entity and RE task, we calculate the objective as

$$\mathcal{L}_{joint}(w; \theta) = \mathcal{L}_{re} + \mathcal{L}_{\text{1st-ner}} + \mathcal{L}_{\text{2nd-ner}} \tag{21}$$

where w refers to tokens and θ denotes model parameters.

Since gold NER tagging information is only available during training, we, therefore, use pseudolabels predicted by the first phase at inference time.

To directly compare with previous works, we also apply adversarial training (AT) [8], which can be used to improve the robustness of neural models by adding small perturbations to training data

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{joint}}(w; \theta) + \mathcal{L}_{\text{joint}}(w + \eta_{\text{adv}}; \theta) \tag{22}$$

where η_{adv} is the worst case perturbation.

F. Weighting Loss With Homoscedastic Uncertainty

Our model has multiple objectives ($\mathcal{L}_{\text{1st-ner}}, \mathcal{L}_{\text{2nd-ner}}$, and $\mathcal{L}_{\text{re}}$). The naive approach to combining multiobjective losses would be to simply perform a weighted linear sum of the losses for each individual task [31], [32]

$$\mathcal{L}_{\text{joint}} = \sum_i w_i \mathcal{L}_i. \tag{23}$$

However, there are a number of issues with this method. Namely, model performance is extremely sensitive to weight selection, w_i. These weight hyperparameters are expensive to tune, often taking many days for each trial. Therefore,

it is desirable to find a more convenient approach that is able to learn the optimal weights. In Bayesian modeling, homoscedastic uncertainty is aleatoric uncertainty that is not dependent on the input data. It is not a model output, rather it is a quantity that stays constant for all input data and varies between different tasks. It can, therefore, be described as task-dependent uncertainty [13]. In our joint framework, the task uncertainty captures the relative confidence between tasks, reflecting the uncertainty inherent to the classification task.

In this section, we derive a joint model loss function with homoscedastic uncertainty. We exploit the idea of homoscedastic uncertainty [13] as a way to learn a relative weighting automatically from the data and is robust to the weight initialization. Specifically, we generate three model outputs that are token representation for the first-phase NER and token-label fusion representation for the second-phase NER and RE. Therefore, we assume the model's observation noise parameter σ_1, σ_2, and σ_3 so-called homoscedastic uncertainty terms, which can capture how much noise we have in the outputs. We proceed here directly to the loss that is in our case given as $\mathcal{L}_{\text{joint}}(w;\theta) = \mathcal{L}'_{\text{re}} + \mathcal{L}'_{\text{1st-ner}} + \mathcal{L}'_{\text{2nd-ner}}$ instead of (21). Here,

$$\mathcal{L}'_{\text{re}} = \frac{1}{\sigma_1^2}\mathcal{L}_{\text{re}}(w;\theta) + \log\sigma_1^2 \tag{24}$$

$$\mathcal{L}'_{\text{1st-ner}} = \frac{1}{\sigma_2^2}\mathcal{L}_{\text{1st-ner}}(w;\theta) + \log\sigma_2^2 \tag{25}$$

$$\mathcal{L}'_{\text{2nd-ner}} = \frac{1}{\sigma_3^2}\mathcal{L}_{\text{2nd-ner}}(w;\theta) + \log\sigma_3^2. \tag{26}$$

We can notice that the model parameters θ (mainly weight parameters) are inversely proportional to homoscedastic uncertainty terms σ_1, σ_2, and σ_3. As the noise decreases, we can see that the weight of the respective objective increases. $\log\sigma_1^2$, $\log\sigma_2^2$, and $\log\sigma_3^2$ act as a regularizer to avoid the trivial solution. This is similar to the concept introduced by Kalervo *et al.* [33] to improve the effect of image recognition classifiers. This is because it is more numerically stable than regressing the variance, σ^2, as the loss avoids any division by zero. The exponential mapping also allows us to regress unconstrained scalar values. The main advantages of this method are that the difficult, time-consuming, and very expensive steps of tuning the weights by hand can be replaced.

IV. Experiments

A. Dataset

To evaluate the performance of our model, we conduct experiments on three datasets: 1) the CoNLL'04 dataset with entity and relation recognition corpora [34]; 2) adverse drug events (ADEs) [35]; and 3) DREC dataset [36]

CoNLL04: This dataset contains sentences with annotated named entities and relations extracted from news articles. We use the splits defined by Gupta *et al.* [37] and Eberts and Ulges [38]. There are four entity types in the dataset ("*Location*," "*Organization*," "*Person*," and "*Other*") and five relation types ("*Kill*," "*Live in*," "*Located in*," "*OrgBased in*," and "*Work for*").

ADE: The ADE dataset aims to extract two kinds of entities ("*Drugs*" and "*Diseases*") and relations about which drug is associated with which disease. It consists of 4272 sentences and 6821 relations extracted from medical reports that describe the adverse effects arising from drug use. To directly compare with previous works, we evaluate our model using tenfold cross validation similar to prior approaches on the ADE dataset [8], [24].

DREC: The DREC dataset contains sentences with annotated named important entities of a property (e.g., floors and spaces) from classifieds. There are nine entity types in the dataset ("*Neighborhood*," "*Floor*," "*Extra building*," "*Extra Invalid*," "*Field*," "*Space and Property*," "*Other*," and "*Subspace*"). Also, there are two relation classes "*Part-of*" and "*Equivalent*." Following Bekoulis *et al.* [36], we also use 70% for training, 15% for validation, and 15% as test set.

We adopt standard Precision (Prec), Recall (Rec), and F1 score to evaluate the model. We use the strict evaluation: the boundary and type of extracted entities should be both correct for NER; named entities and the type of their relationships should be both correct for RE.

1) Implementation Details: We utilize the 50-D word embeddings used in [36], which are pretrained on Wikipedia, for the CoNLL04 corpus. For the ADE dataset, we used 200-D embeddings used by Bekoulis *et al.* [36] and trained on a combination of PubMed and PMC texts with texts extracted from English Wikipedia [36]. Finally, we use the 128-D word2vec embeddings used by Bekoulis *et al.* [36] trained on a large collection of 887k Dutch property advertisements 7 for the DREC dataset. We regularize our network using dropout with a rate tuned on the development set (the dropout rate is 0.2 for embeddings and 0.1, 0.3, and 0.3 for BiLSTM on three datasets, respectively). We utilize two BSA units in the first phase ($m = 2$) and set the dimensionality of the hidden size d as 128. We choose 25 as the dimensionality of label embeddings d_l. The size of character embeddings is 128, while the dimensionality of ELMo [25] is 1024. The Adam optimizer with a learning rate of 0.0005 is used to optimize parameters. The training takes 180 epochs for convergence. For a fair comparison, all experiments are implemented in Tensforflow and conducted using a GeForce GTX 1080Ti with 11-GB memory.

B. Quantitative Results

In this section, we present the performance of different models on three datasets.

CoNLL04: For the CoNLL04 dataset, we compare the proposed model with several competing approaches and show the results in Table I. It can be seen that our model achieves state-of-the-art performance on entity recognition and relation classification by obtaining 91.72 and 73.46 F1, respectively. Compared with prior competing SpERT method [38] that relies on pretrained language model (BERT) [42], our approach gets absolute F1 improvements of 2.78% and 1.99% on NER and RE, respectively. We find even stronger performance increases with respect to NER (+8.11%) and RE (+11.51%) compared to the multihead + AT baseline [8], which uses

TABLE I

COMPARISON OF OUR METHOD WITH OTHER COMPETING APPROACHES IN TERMS OF F1 SCORE ON THE CONLL04 DATASET. BEKOULIS *et al.* [8][2], LI *et al.* [9][4], EBERTS AND ULGES [38][5], MIWA AND SASAKI [39][1], AND TRAN AND KAVULURU [40][3]. RESULTS WITH ∗ INDICATE THAT THE STUDY APPLY BERT AS THEIR CORE MODEL. ALL BASELINE RESULTS ARE OBTAINED FROM THEIR ORIGINAL PAPERS

Models	Entity	Relation
Table Representation[1]	80.70	61.00
NN CRF [2]	82.10	62.50
Global Optimization [3]	85.60	67.80
Multi-head [4]	83.04	72.04
Multi-head + AT [5]	83.61	61.95
Relation-Metric with AT [6]	84.57	62.28
Multi-turn QA [7]∗	87.80	68.20
SpERT [8]∗	88.94	71.47
CMAN (BiLSTM)	**91.72**	**73.46**

TABLE II

PERFORMANCE OF OUR METHOD AND OTHER COMPETING APPROACHES IN TERMS OF F1 SCORE ON THE ADE DATASET. BEKOULIS *et al.* [8][3], LI *et al.* [24][2], EBERTS AND ULGES [38][5], TRAN AND KAVULURU [40][4], AND LI *et al.* [41][1]. RESULTS WITH ∗ INDICATE THAT THE STUDY APPLIES BERT AS THEIR CORE MODEL. ALL BASELINE RESULTS ARE OBTAINED FROM THEIR ORIGINAL PAPERS

Models	Entity	Relation
Joint Model[1]	79.50	63.40
Neural joint model[2]	84.60	71.40
Multi-head + AT [3]	86.73	75.52
Relation-Metric with AT [4]	87.02	77.19
SpERT [5]∗	89.25	79.24
CMAN (ours)	**90.12**	**81.34**

feature concatenation for capturing interactions in token-label spaces and applies a multihead mechanism for RE decoding. The above results indicate the effectiveness of our method and suggest that CMAN is able to model dense cross-modal interactions for joint entity and RE.

ADE: Table II presents the performance comparison between our approach and other competitive methods on the ADE dataset. Compared to the latest SpERT model, our approach only has a slight improvement (+0.87%) on NER. However, it can be found that our proposed model significantly outperforms SpERT by 2.1% F1 on RE. We think the reason may be that the ADE dataset contains fewer relations than CoNLL04, which is relatively easy for RE.

DREC: We also evaluate our model on the DREC dataset. Table III presents the performance comparison between our approach and other competitive methods. To fairly compare with previous works, we only use word embedding following Bekoulis *et al.* [8]. Compared to the latest multihead + AT model, our approach only has a slight improvement (+0.92%) on NER. However, it can be found that our proposed model significantly outperforms multihead + AT by 2.54% F1 on RE. The result strongly verifies that our method is compatible with other domains and can achieve state-of-the-art results.

TABLE III

PERFORMANCE OF OUR METHOD AND OTHER COMPETING APPROACHES IN TERMS OF F1 SCORE ON THE DREC DATASET. BASELINE RESULTS ARE REPORTED IN [8]

Models	Entity	Relation
Attentive LSTM	79.11	49.70
Multi-head	81.39	52.26
Multi-head + AT	82.04	53.12
CMAN (ours)	**82.96**	**55.66**

C. Performance Against Entity Distance

Fig. 5 shows F1 scores of the baseline model and CMAN under different entity distances on the CoNLL04 test set. Since multihead + AT [8] adopts CRF and multihead mechanism for NER&RE decoding, we, therefore, set it as the baseline model. The CoNLL04 test set is split into three parts according to the metric of entity distance. We measure distance by computing the absolute character offset between the last character of the first occurring entity and the last character of the second mentioned entity, which is, henceforth, simply referred to as entity distance. The results indicate that CMAN significantly outperforms the baseline across different entity distances. In particular, the F1 score of CMAN is nearly 18.18% greater than that of the baseline for RE when the entity distance is more than 20 characters. It demonstrates that CMAN has a much greater advantage than the baseline in dealing with entities that are far apart from each other. The reason is that CMAN can detect token-level self-correlations by modeling dense intramodal interactions among tokens via the proposed BSA unit. Besides, we can notice that the effect of entity distance on RE is significantly higher than the impact on NER, likely due to that RE relies more on finding relevant distant entities.

D. Performance Against Sentence Length

To investigate the influence of sentence length, we analyze the performance of the baseline model and CMAN under grouped sentence lengths on the CoNLL04 test set, which is shown in Fig. 6. Similarly, the multihead + AT model is used as the baseline. We partition the sentence length into four groups ([0–19], [20–34], [35–49], [≥50]). We can observe that CMAN performs way better than the baseline under different sentence lengths. Moreover, the improvement achieved by CMAN is further enhanced when the sentence length consistently increases. In particular, CMAN outperforms the baseline by 11.06% and 22.03% F1 scores for NER and RE, respectively, when the sentence length is large than 50. These results demonstrate that CMAN is more effective in terms of long sentences. It also verifies that our model can capture the global dependencies of the whole sentence.

E. Performance Against Training Data Size

To explore the effectiveness of different training data sizes, we analyze the performance of our CMAN model and baseline

[1]https://github.com/bekou/multihead_joint_entity_relation_extraction

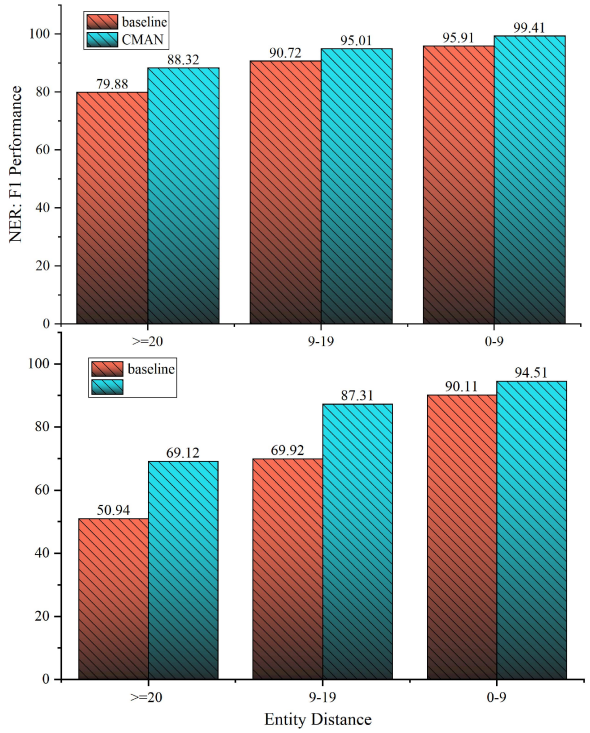

Fig. 5. Comparison of the baseline and CMAN under different entity distances on the CoNLL04 test set. We use multihead + AT as the baseline. We get the result of baseline by running the model source code.[1]

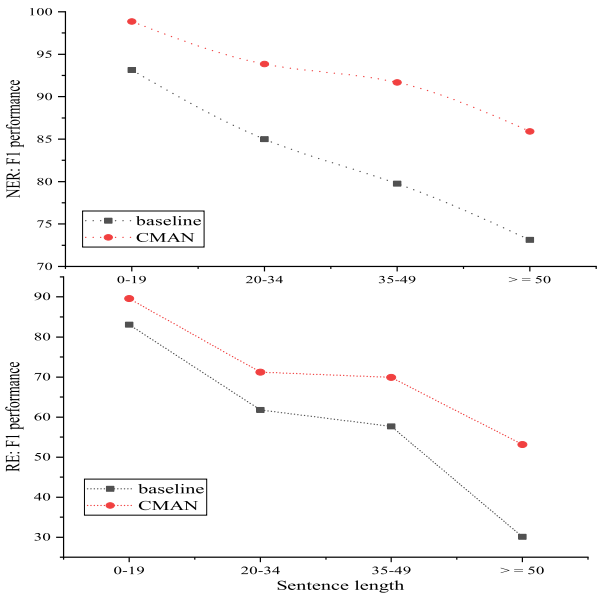

Fig. 6. Comparison of the baseline and CMAN under different sentence lengths on the CoNLL04 test set, where multihead + AT is used as baseline. We get the result of baseline by running the model source code.[1]

with respect to different training data sizes on the CoNLL04 dataset, which is shown in Fig. 7(a) and (b). We consider five training settings (20%, 40%, 60%, 80%, and 100% of the training data). Here, the multihead + AT model is also used as the baseline. CMAN consistently outperforms the baseline under the same amount of training data. When the size of training data increases, we can observe that the performance gap becomes more obvious. Particularly, using 60% of the training data, the CMAN model is able to achieve an F1 score of 86.82 and 62.91 on NER and RE, respectively, higher

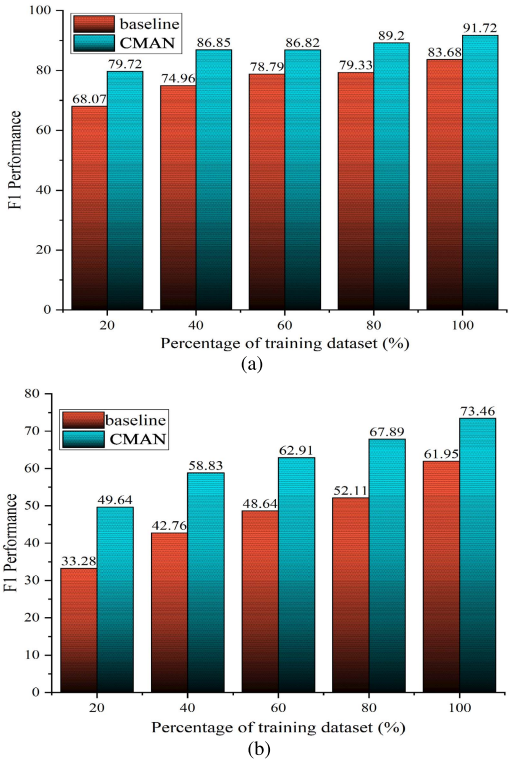

Fig. 7. Comparison of our CMAN and baseline against different training data sizes on CoNNL04 datasets, where multihead + AT is used as baseline. We get the result of baseline by running the model source code.[1] (a) Result on NER. (b) Result on RE.

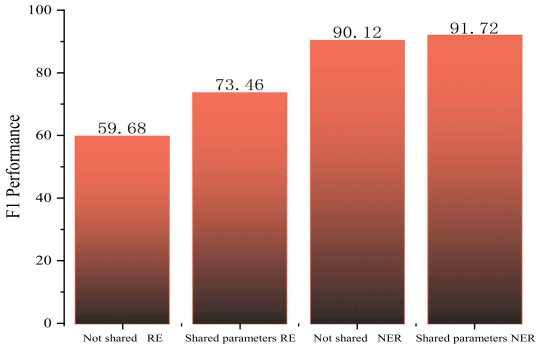

Fig. 8. Comparison of F1 scores with or without shared parameters on NER and RE.

TABLE IV
PERFORMANCE OF CMAN UNDER DIFFERENT LABEL REPRESENTATIONS ON THE DREC TEST SET

Model	Entity	Relation
Encoded label representations	82.02	54.23
Upade label representations	82.96	55.66

than the baseline trained on the whole dataset. These results demonstrate that our model is more effective in terms of using training resources.

F. Effectiveness of Joint Model

To analyze the influence of the joint model, we conduct some comparative experiments by BSA units with shared parameters or without shared parameters, which is shown in Fig. 6.

TABLE V
RELATIVE DECODING-TIME SPEED OF DIFFERENT MODELS ON JOINT TASKS

Method	CoNLL04		ADE		DREC	
	0verall-F1	Speedup	0verall-F1	Speedup	0verall-F1	Speedup
Multi-head + AT	72.78	1x	81.13	1x	67.58	1x
CMAN (ours)-Homoscedastic Uncertainty	81.89	12	84.69	8	68.64	10
CMAN (ours)	82.59	1.62	85.73	1.75	69.31	1.82

TABLE VI
PERFORMANCE OF CMAN UNDER DIFFERENT NEURAL NETWORKS ON THE CONLL04 TEST SET

Model	Entity	Relation
CMAN (LSTM)	90.50	71.86
CMAN (GRU)	88.58	69.57
CMAN (BiGRU)	90.88	72.16
CMAN (BiLSTM)	91.72	73.46

As can be seen from Fig. 8, without shared parameters hurts the performance: the F1 score on NER and RE decreases significantly by 1.6% and 13.78%, respectively. These results indicate that the joint model can simultaneously learn NER and RE and further promotes each other. Therefore, our joint model can represent both entities and relations with shared parameters in a single model.

G. Effectiveness of Label Gate

In order to analyze the influence of the label gate, we evaluate our model with a label gate or without a label gate on the DREC test set. To make the comparison fair, we set all hyperparameters unchanged that only feed the model with different encoded label representations. The results are shown in Table IV. We replace the encoded label representations $\tilde{L}'$ with upade label representations $\tilde{L}$ and find that the performance increase to 82.96 and 55.66 (+0.94% and 1.43%) F1 on NER and RE, respectively. These results indicate that our model can control label feature contributions, especially when much unincorrect label information hinders learning.

H. Performance Against Neural Networks

To analyze the influence of different neural networks, we conducted experiments (compared with LSTM, GUR, and BiGRU) on the CoNLL04 dataset, as shown in Table VI. It can be seen that CMAN (BiLSTM) achieves the best performance on entity recognition and relation classification by obtaining 91.72 and 73.46 F1, respectively. Compared with LSTM and GUR, BiLSTM and BiGRU have stronger performance increases with respect to NER and RE. The reason is that we can efficiently make use of past features (via forwarding states) and future features (via backward states) for a specific time frame [43]. Moreover, BiLSTM slightly increases by 0.84% and 1.3% on NER and RE, respectively, compared to the BiGRU. We think the reason may be that BiLSTM has more parameters and can provide better semantic representations of the token.

I. Performance Against Efficiency

To explore the efficiency of our model, we conducted experiments of inference time on all datasets. Table V lists the relative decoding time on three of the test sets compared to the multihead + AT. To fairly compare with the baseline model, we report the decoding time using the same batch size for each method. As we can see, multihead + AT runs 1.62, 1.75, and 1.82 times faster than our CMAN on three datasets, respectively. The reason is that the used multihead attention module is very time-consuming. However, when we use manual weighting losses, the time consumed is 12, eight, and ten times than multihead + AT on three datasets, respectively. It is worth noting that time consumed by manual weighting losses is random. Moreover, our CMAN achieves the best F1 score results than the baseline model and CMAN-Homoscedastic Uncertainty model.

TABLE VII
ABLATIONS ON THE CONLL04 DATASET

Model	Entity	Relation
CMAN	91.72	73.46
- Self-attention in BSA	90.01	69.98
- BLA unit	90.76	71.34
- Both units	89.86	69.21
- Homoscedastic Uncertainty	91.02	72.76
replace BiLSTM with MLP	90.79	72.16

J. Ablation Study

We conduct an ablation study to investigate the effectiveness of our attention units and network architecture in Table VII. First, since the BiLSTM layer in BSA is a necessary component to encode tokens, we only remove the self-attention module to perform the ablation. We can observe that the F1 score drops by 1.71% and 3.48% for NER and RE tasks, respectively, indicating that self-attention is critical for capturing self-correlations among tokens. Second, we ablate the BLA unit, use self-aware token features for both tasks, and find that the performance slightly decreases, showing the beneficial effect of incorporating label-space information. Deleting both attention units leads to further worse results on NER (−1.86%) and RE (−4.25%), which suggests that modeling dense cross-modal interactions plays a vital role in joint learning. After that, to further investigate the influence of weighting losses, we remove homoscedastic uncertainty in NER and RE and lead to further worse results on NER (−0.5%) and RE (−0.7%) on the CoNLL04 dataset, which suggests that weighting losses are beneficial for our model. Finally, to test the network architecture, we replace BiLSTM with multilayer perceptron (MLP) and find that the performance significantly drops to 90.79 and 72.16 F1 scores, implying the importance of building contextualized representations.

V. Conclusion

In this article, we propose a deep CMAN for the task of joint entity-RE. The network aims to capture dense cross-modal interactions by leveraging NER label information, where two basic attention units are proposed to model token-to-token and label-to-token correlations synergistically. Particularly, compared with our initial conference version, this article can dynamically control label feature contributions and take into account the interactions between token and label features in the extraction of entities and their relations. Moreover, we introduce homoscedastic uncertainty to automatically weighting the losses of two-phase Prediction. Finally, we evaluate the proposed method on CoNLL04, ADE, and DERC datasets. The results show that CMAN achieves new state-of-the-art performance compared to other competing approaches.

References

[1] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *Proc. 57th Annu. Meeting Assoc. for Comput. Linguistics*, 2019, pp. 1–13.

[2] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticæ Investigationes*, vol. 30, no. 1, pp. 3–26, Jan. 2007.

[3] S. Zhao, M. Hu, Z. Cai, H. Chen, and F. Liu, "Dynamic modeling cross- and self-lattice attention network for Chinese NER," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 16, pp. 14515–14523.

[4] S. Zhao, Z. Cai, H. Chen, Y. Wang, F. Liu, and A. Liu, "Adversarial training based lattice LSTM for Chinese clinical named entity recognition," *J. Biomed. Informat.*, vol. 99, Nov. 2019, Art. no. 103290.

[5] N. Bach and S. Badaskar, "A review of relation extraction," *Literature Rev. Lang. Statist. II*, vol. 2, pp. 1–15, Nov. 2007.

[6] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," 2016, *arXiv:1601.00770*. [Online]. Available: http://arxiv.org/abs/1601.00770

[7] H. Adel and H. Schütze, "Global normalization of convolutional neural networks for joint entity and relation classification," 2017, *arXiv:1707.07719*. [Online]. Available: http://arxiv.org/abs/1707.07719

[8] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Adversarial training for multi-context joint entity and relation extraction," 2018, *arXiv:1808.06876*. [Online]. Available: http://arxiv.org/abs/1808.06876

[9] X. Li *et al.*, "Entity-relation extraction as multi-turn question answering," 2019, *arXiv:1905.05529*. [Online]. Available: http://arxiv.org/abs/1905.05529

[10] L. Wang, Z. Cao, G. De Melo, and Z. Liu, "Relation classification via multi-level attention CNNs," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1298–1307.

[11] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *Proc. 55th Annu. Meeting Assoc. for Comput. Linguistics*, vol. 1, 2017, pp. 917–928.

[12] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 6281–6290.

[13] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[14] S. Zhao, M. Hu, Z. Cai, and F. Liu, "Modeling dense cross-modal interactions for joint entity-relation extraction," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, C. Bessiere, Ed., Jul. 2020, pp. 4032–4038.

[15] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th*, 2001.

[16] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, "A rich feature vector for protein-protein interaction extraction from multiple corpora," in *Proc. EMNLP*, 2009, pp. 121–130.

[17] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," 2017, *arXiv:1706.05075*. [Online]. Available: http://arxiv.org/abs/1706.05075

[18] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[19] G. Wang *et al.*, "Joint embedding of words and labels for text classification," 2018, *arXiv:1805.04174*. [Online]. Available: http://arxiv.org/abs/1805.04174

[20] L. Cui and Y. Zhang, "Hierarchically-refined label attention network for sequence labeling," 2019, *arXiv:1908.08676*. [Online]. Available: http://arxiv.org/abs/1908.08676

[21] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, "Shared predictive cross-modal deep quantization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5292–5303, Nov. 2018.

[22] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2189–2201, Jun. 2020.

[23] S. Antol *et al.*, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2425–2433.

[24] F. Li, M. Zhang, G. Fu, and D. Ji, "A neural joint model for entity and relation extraction from biomedical text," *BMC Bioinf.*, vol. 18, no. 1, p. 198, Mar. 2017.

[25] M. E. Peters *et al.*, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[27] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450

[28] Q. Zhang, J. Fu, X. Liu, and X. Huang, "Adaptive co-attention network for named entity recognition in tweets," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.

[29] D. Lu, L. Neves, V. Carvalho, N. Zhang, and H. Ji, "Visual attention model for name tagging in multimodal social media," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 1990–1999.

[30] J. Yu, J. Jiang, L. Yang, and R. Xia, "Improving multimodal named entity recognition via entity span detection with unified multimodal transformer," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3342–3352.

[31] P. Rolet, M. Sebag, and O. Teytaud, "Integrated recognition, localization and detection using convolutional networks," in *Proc. ECML Conf.*, 2012, pp. 1255–1263.

[32] M. Teichmann, M. Weber, M. Zollner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time joint semantic reasoning for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1013–1020.

[33] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis," in *Proc. Scand. Conf. Image Anal.* Cham, Switzerland: Springer, 2019, pp. 28–40.

[34] D. Roth and W.-T. Yih, "A linear programming formulation for global inference in natural language tasks," in *Proc. 8th Conf. Comput. Natural Lang. Learn. (CoNLL HLT-NAACL)*, 2004, pp. 1–8.

[35] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *J. Biomed. Inform.*, vol. 45, no. 5, pp. 885–892, May 2012.

[36] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "An attentive neural architecture for joint segmentation and parsing and its application to real estate ads," *Expert Syst. Appl.*, vol. 102, pp. 100–112, Jul. 2018.

[37] P. Gupta, H. Schütze, and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *Proc. COLING*, 2016, pp. 2537–2547.

[38] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," 2019, *arXiv:1909.07755*. [Online]. Available: http://arxiv.org/abs/1909.07755

[39] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1858–1869.

[40] T. Tran and R. Kavuluru, "Neural metric learning for fast end-to-end relation extraction," 2019, *arXiv:1905.07458*. [Online]. Available: http://arxiv.org/abs/1905.07458

[41] F. Li, Y. Zhang, M. Zhang, and D. Ji, "Joint models for extracting adverse drug events from biomedical text," in *Proc. IJCAI*, vol. 2016, 2016, pp. 2838–2844.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[43] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: http://arxiv.org/abs/1508.01991