MAF: A General Matching and Alignment Framework for Multimodal Named Entity Recognition

Bo Xu
xubo@dhu.edu.cn
School of Computer Science and Technology, Donghua University
Shanghai, China

Shizhou Huang
2202408@mail.dhu.edu.cn
School of Computer Science and Technology, Donghua University
Shanghai, China

Chaofeng Sha*
cfsha@fudan.edu.cn
School of Computer Science, Fudan University
Shanghai Key Laboratory of Intelligence Processing
Shanghai, China

Hongya Wang
hywang@dhu.edu.cn
School of Computer Science and Technology, Donghua University
Shanghai, China

ABSTRACT

In this paper, we study multimodal named entity recognition in social media posts. Existing works mainly focus on using a cross-modal attention mechanism to combine text representation with image representation. However, they still suffer from two weaknesses: (1) the current methods are based on a strong assumption that each text and its accompanying image are matched, and the image can be used to help identify named entities in the text. However, this assumption is not always true in real scenarios, and the strong assumption may reduce the recognition effect of the *MNER* model; (2) the current methods fail to construct a consistent representation to bridge the semantic gap between two modalities, which prevents the model from establishing a good connection between the text and image. To address these issues, we propose a general matching and alignment framework (*MAF*) for multimodal named entity recognition in social media posts. Specifically, to solve the first issue, we propose a novel cross-modal matching (*CM*) module to calculate the similarity score between text and image, and use the score to determine the proportion of visual information that should be retained. To solve the second issue, we propose a novel cross-modal alignment (*CA*) module to make the representations of the two modalities more consistent. We conduct extensive experiments, ablation studies, and case studies to demonstrate the effectiveness and efficiency of our method. The source code of this paper can be found in https://github.com/xubodhu/MAF.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; • **Computing methodologies** → **Information extraction**.

KEYWORDS

multimodal named entity recognition; contrastive learning

1 INTRODUCTION

Multimodal named entity recognition (*MNER*) has become an important research direction of named entity recognition (*NER*), which can improve text-based *NER* by using images as additional input [20]. It assumes that image information can help identify ambiguous named entities when text information is insufficient. For example, given the text 'Handsome Rob after a fish dinner', it is difficult for us to infer the type of named entity Rob. It may describe a person or an animal. With the help of its accompanying image (as shown in Figure 1), we can easily determine that its type is *MISC*.

Text: Handsome [**Rob MISC**] after a fish dinner.

Figure 1: An Example of Multimodal Named Entity Recognition. The Named Entity and Its Type are Highlighted in Brackets.

In this paper, we study *MNER* in social media posts. Compared with text-based *NER* methods, existing works have achieved good performance [2, 12, 14, 19, 22]. They mainly focus on using a cross-modal attention mechanism to combine text representation with image representation. [14] first proposes an *LSTM-CNN* architecture that combines text with image information via a general modality attention module. [12] proposes an attention-based model to extract image features from the regions in the image most related to the text and uses a gate to combine text features and image features. [22] proposes an adaptive co-attention network to control the combination of text representation and image representation dynamically. [19] proposes a multimodal transformer architecture, which captures the inter-modal interactions with a multimodal interaction module. In addition, [2] introduces image attributes and image knowledge to help capture the deep features of the image and improve the performance of the *MNER* model.

Despite their success, existing *MNER* methods still suffer from two weaknesses:

- Firstly, the current methods are based on a strong assumption that each text and its accompanying image are matched, and the image can be used to help identify named entities in the text. Therefore, when identifying named entities in text, both text information and image information must be considered. However, not all text is matched to their accompanying images, and considering the mismatched image information may mislead the model. For example, in Figure 2, there is no relationship between the object (a person) in the image and the named entity (Siri) in the text. If this mismatched image is considered, the *MNER* methods would regard Siri as the person in the image and make an incorrect prediction. However, thanks to the pre-trained model (i.e., *BERT* [4]), the text-based *NER* methods can easily infer that the type of Siri is *MISC*. This kind of mismatch is common, as reported in [18], there are about 33.8% of tweets that the textual content is not represented in the image and the image does not add additional content.

Text: Ask [**Siri MISC**] what 0 divided by 0 is and watch her put you in your place.

Figure 2: An Example of Mismatched Text-Image Pair.

- Secondly, current methods fail to construct a consistent representation to bridge the semantic gap between the two modalities. Since the representations of text and images come from different encoders, the representations between them are inconsistent. Therefore, it is difficult to directly use these inconsistent representations to capture the correspondence

between words in the text and regions in the image. For example, in Figure 1, the word Rob in the text corresponds to the region where the object Cat is in the image. Ideally, the word Rob should have a higher similarity with the region related to object Cat in the image, and should have a lower similarity with other regions in the image. However, due to the inconsistent representations between the text and the image, when calculating the similarity score, the similarity between the Rob in the text and the Cat in the image may be lower than the similarity of other regions. Thus, the inconsistent representations will prevent the model from establishing a good connection between the text and image.

To address these issues, we propose a general matching and alignment framework (*MAF*). Specifically, to solve the first issue, we propose a novel cross-modal matching (*CM*) module to calculate the similarity score between text and image, and use the score to determine the proportion of image information that should be retained. To solve the second issue, we propose a cross-modal alignment (*CA*) module to make the representations of the two modalities more consistent.

Our main contributions can be summarized as follows:

- Firstly, we propose a general matching and alignment framework for the *MNER* task, which can reduce the impact of mismatched text-image pairs and make the representations between the two modalities more consistent.
- Secondly, the two modules we proposed (*CA* and *CM*) are based on self-supervised learning, without requiring any additional data annotations, and can be easily extended to other multimodal tasks.
- Finally, experiments conducted on two widely used *MNER* datasets show that our method achieves the new state-of-the-art performance. We also conduct ablation studies and case studies to show that both the *CA* module and *CM* module play an essential role in our framework.

2 OVERVIEW

In this section, we first formulate our problem, and then introduce the framework of our system: *MAF*.

2.1 Problem Formulation

Given a text S and its associated image I as input, the task of *MNER* is to extract a set of named entities from S, and classify each extracted named entity into one of the pre-defined types. As most existing work on *MNER*, we formulate the task as a sequence labeling problem. Let $S = (s_1, s_2, ..., s_n)$ denote a sequence of input words, and $y = (y_1, y_2, ..., y_n)$ be the corresponding label sequence, where $y_i \in \mathcal{Y}$ and $\mathcal{Y}$ is the pre-defined label set with the *BIO2* tagging schema [16].

2.2 Framework

Our general matching and alignment framework (*MAF*) is shown in Figure 3, which contains four main components: (1) cross-modal alignment module; (2) cross-modal interaction module; (3) cross-modal matching module; (4) cross-modal fusion module.

The overall process is as follows: we first obtain the representation of each word and the entire text through *BERT* [4], and obtain
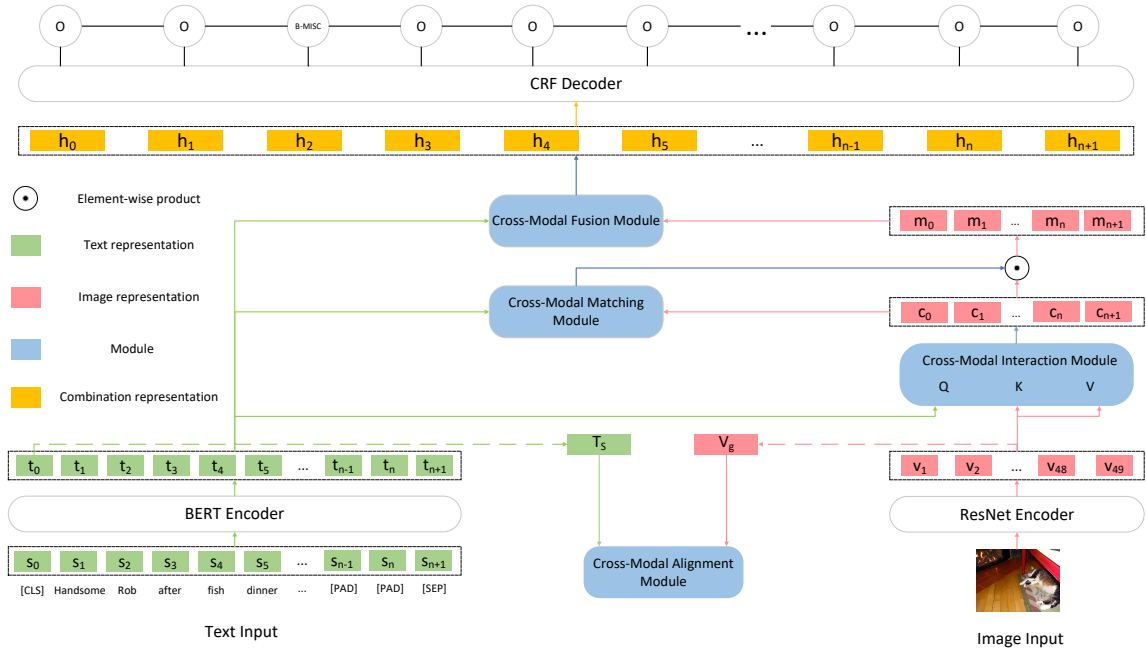
Figure 3: Overall Architecture of *MAF*.

the regional and global representation of the image through the *ResNet* [7]. Then, the representation of the entire text and global representation of the image will be fed to the cross-modal alignment module, and the representation of each word and the regional representation of the image will be fed to the cross-modal interaction module. The cross-modal alignment module is used to make the representations from the text encoder and the image encoder more consistent, and the cross-modal interaction module is used to get a text-aware image representation. Then we use the cross-modal matching module to determine the proportion of image information that should be retained. Finally, we use the cross-modal fusion module to fuse the representations of the two modalities, and feed them into a conditional random field layer to get the final prediction result. These modules are trained simultaneously.

3 METHOD

3.1 Input Representations

3.1.1 Text Encoder. We use *BERT* [4] as the text encoder, which obtains a deep bidirectional representation by pre-training on a large of corpus. Following [4], each input sentence needs to add a [CLS] token at the beginning and a [SEP] token at the end. We denote the text input as $S' = (s_0, s_1, ..., s_n, s_{n+1})$, where s_0 is the [CLS] token and s_{n+1} is the [SEP] token, s_1 to s_n represent the token sequence of the input sentence. The length of the input text is fixed to n, so the text longer than n will be truncated to n, and the text shorter than n will be filled with [PAD] token to n. We feed the input S' to *BERT* to obtain the representation of token sequence $T = (t_0, t_1, ..., t_n, t_{n+1})$, where $t_i \in \mathbb{R}^d$ corresponding to the representation of s_i. To get the representation of the entire text T_s, we feed t_0 to a fully connected layer with an activation function

of *Tanh*, which is used to obtain its final hidden state as a sentence representation [4].

3.1.2 Image Encoder. We use *ResNet* [7] as the image encoder, which is one of the state-of-the-art convolutional neural networks. According to [12], we first resize the image to 224×224 pixels, then the image is fed to *ResNet* to obtain the regional and global representation of the image. The regional representation of the image $J = (j_1, j_2, ..., j_{48}, j_{49})$ is from the convolutional layer of the last layer of *ResNet*, and the dimension is $2048 \times 7 \times 7$, where $7 \times 7 = 49$ is the number of regions in the image and 2048 is the dimension of the representation of each region in the image, and the size of each region is 32×32 pixels. We use an average pooling layer with a size of 7×7 to J to obtain the global representation of the image $V_g \in \mathbb{R}^{2048}$ to represent the entire image. At last, we project J to the same dimensions as the text representation: $V = W_j^T J$, where $W_j^T \in \mathbb{R}^{2048 \times d}$ is the weight matrix. Therefore, $V = (v_1, v_2, ..., v_{48}, v_{49})$, where $v_i \in \mathbb{R}^d$ represents i-th region of the image.

3.2 Cross-Modal Alignment Module

The previous model cannot align the representations between the two modalities, and the cross-modal alignment (*CA*) module is used to make the representations from the text encoder and the image encoder more consistent. Inspired by recent advances in contrastive learning [3, 5, 21], we propose a novel contrastive learning method for cross-modal consistent representation. The inputs of the *CA* module are the text representation T_s and the global representation of the image V_g. The entire process can be described concisely in three basic steps:

Firstly, we generate the positive and negative examples from a batch of (T_s, V_g) input pairs with size N. T_s^a is the text representation of the a-th pair in the batch, and V_g^b is the image representation of the b-th pair in the batch. We assume that the positive examples are the text and image representations from the same input pairs $\{(T_s^a, V_g^b)_{a=b}\}$, and negative examples are the representations from different input pairs $\{(T_s^a, V_g^b)_{a \neq b}\}$. Therefore, we can obtain 1 positive example and $N-1$ negative examples for each input pair in the batch. According to [3], the effect of contrastive learning is mainly affected by the number of negative examples, which is positively correlated with the number of negative examples. Thus the impact of a small number of mismatched pairs that may appear in the positive examples is negligible.

Secondly, for each example (T_s^a, V_g^b), we adopt two different *MLP*s with one hidden layer applying on the T_s^a, V_g^b to get the projected text representation $T_c^a \in \mathbb{R}^d$ and image representation $V_c^b \in \mathbb{R}^d$, respectively. We find that this *MLP* projection can help the encoders (in our setting, *BERT* and *ResNet*) to learn a better representation, as is also found in [3, 11].

Thirdly, we try to maximize the similarity of the positive examples and minimize the similarity of the negative examples by minimizing two contrastive loss functions, namely the image-to-text contrastive loss function and the text-to-image contrastive loss function [23]. The image-to-text contrastive loss function for the i-th positive projected pair in the batch is defined as follows:

$$\mathcal{L}_i^{(V_c \to T_c)} = -log \frac{exp(sim(V_c^i, T_c^i)/\tau)}{\sum_{j=1}^{N} exp(sim(V_c^i, T_c^j)/\tau)}, \tag{1}$$

where $sim(V_c^i, T_c^i) = (V_c^i)^T T_c^i / \|V_c^i\| \|T_c^i\|$ represents the cosine similarity between V_c^i and T_c^i, τ is the temperature parameter, which is a hyperparameter. The text-to-image contrastive loss for i-th positive projected pair is defined as follows:

$$\mathcal{L}_i^{(T_c \to V_c)} = -log \frac{exp(sim(T_c^i, V_c^i)/\tau)}{\sum_{j=1}^{N} exp(sim(T_c^i, V_c^j)/\tau)} \tag{2}$$

At last, we sum up the two losses for all positive projected pairs in the batch:

$$\mathcal{L}_{ca} = \frac{1}{N} \sum_{i=1}^{N} (\lambda_c \mathcal{L}_i^{(V_c \to T_c)} + (1 - \lambda_c)\mathcal{L}_i^{(T_c \to V_c)}), \tag{3}$$

where $\lambda_c \in [0, 1]$ is a hyperparameter. By minimizing the loss function, the representations from the text encoder and the image encoder will be more consistent.

3.3 Cross-Modal Interaction Module

To better infer the relationship between the image and the text, the cross-modal interaction (*CI*) module is used to obtain a text-aware image representation. Similar to [17, 19], it consists of two sub-layers. The first is a multi-head attention layer, and the second is a fully connected feed-forward network. We employ a residual connection around each of the two sub-layers, followed by layer normalization. The details are as follows.

An attention function can be described as mapping a query and a set of key-value pairs to an output [17]. As shown in Figure 3, we use the token sequence $T = (t_0, t_1, ..., t_n, t_{n+1}) \in \mathbb{R}^{d \times (n+2)}$ as queries,

and use the regional image representation $V = (v_1, v_2, ..., v_{48}, v_{49}) \in \mathbb{R}^{d \times 49}$ as key-value pairs to obtain the text-aware image representation as follows:

$$\alpha_i = softmax(\frac{[W_{qi}T]^T [W_{ki}V]}{\sqrt{d/m}}) \tag{4}$$

$$CV_i = \alpha_i [W_{vi}V]^T \tag{5}$$

$$CV = W'[CV_1; CV_2; ...; CV_m]^T, \tag{6}$$

where $\{W_{qi}, W_{ki}, W_{vi}\} \in \mathbb{R}^{d/m \times d}$ are the weight matrices for each query, key and value, $W' \in \mathbb{R}^{d \times d}$ is the weight matrix for m-head attention. $\alpha_i \in \mathbb{R}^{(n+2) \times 49}$ denotes the alignment score between each token (includes [CLS], [PAD] and [SEP]) and each image region. $CV_i \in \mathbb{R}^{(n+2) \times d/m}$ denotes the text-aware image representation from the i-th cross-modal attention head, $CV \in \mathbb{R}^{d \times (n+2)}$ is the text-aware image representations from m attention heads.

After that, we obtain the final text-aware image representation $C = (c_0, c_1, ..., c_n, c_{n+1})$ ($c_i \in \mathbb{R}^d$) as follows:

$$V' = LN(T + CV) \tag{7}$$

$$C = LN(V' + FFN(V')), \tag{8}$$

where LN is the layer normalization [1], FFN is the feed-forward network.

3.4 Cross-Modal Matching Module

The cross-modal matching (*CM*) module is used to determine the proportion of image information that should be retained. It receives the token sequence representation T and the text-aware image representation C as input, and outputs the relatedness between them. Due to the lack of explicit knowledge about the relatedness between text and images, we propose a self-supervised learning method to train the module. The entire process is described as follows:

Firstly, we generate the training examples from a batch of (T, C) input pairs with size N. Let T^a be the token sequence representation of the a-th pair in the batch, and C^b be the text-aware image representation of the b-th pair in the batch. We assume that the positive examples are the text image representations from the same input pairs $\{(T^a, C^b)_{a=b}\}$, and negative examples are the representations from different input pairs $\{(T^a, C^b)_{a \neq b}\}$. This assumption is similar to the *CA* module (See Section 3.2), but the input and output of the module and the process of generating positive and negative examples are different. Specifically, we randomly select $2k$ $(0 < k < N/2)$ input pairs from the batch and swap the image representations of the first half in the input pairs with the second half as the negative examples. Moreover, the remaining $N - 2k$ input pairs in the batch are positive examples. For example, there are three input pairs in the batch, which are (T^1, C^1), (T^2, C^2) and (T^3, C^3). Among them, (T^1, C^1) and (T^2, C^2) are selected to swap their image representations. Finally, we obtain two negative examples (T^1, C^2) and (T^2, C^1) and one positive example (T^3, C^3). There may also be mismatched examples in the positive examples, so we set the k value to be relatively small to reduce the negative impact caused by the mismatched examples while ensuring the

balance of the samples. We also try to crop the image to generate positive and negative samples, but the final effect is not improved.

Secondly, we train the *CM* module by using the generated training examples. Let $D_m = (D_{m1}, D_{m2}, ..., D_{mN})$ be the batch of training examples, D_{mi} is the i-th example. $T(D_{mi})$ and $C(D_{mi})$ are the text representation and image representation in D_{mi}, respectively. Specifically, for each example D_{mi}, we first concatenate its text representation $T(D_{mi})$ and image representation $C(D_{mi})$, and then flatten it into a fully connected layer. The prediction process is as follows:

$$F_i = Flatten([T(D_{mi}); C(D_{mi})]) \tag{9}$$

$$\hat{y_{mi}} = \sigma(W_f^T F_i), \tag{10}$$

where W_f is the weight parameter, σ is the sigmoid activation function. To train the module, we use binary cross-entropy [6] as our loss function $\mathcal{L}_{cm}$, which is defined as follows:

$$\mathcal{L}_{cm} = -\frac{1}{N} \sum_{j=1}^{N} y_{mj} \cdot log(\hat{y_{mj}}) + (1 - y_{mj}) \cdot log(1 - \hat{y_{mj}}) \tag{11}$$

Finally, we obtain the retained image representation as follows:

$$M = \hat{y_m} \odot C, \tag{12}$$

where C (See Eq 8) is the text-aware image representation of the input, y_m is the value of the real label of the positive and negative examples corresponding to 1 and 0 respectively, $\hat{y_m}$ is the corresponding predicted retained probability, and $\odot$ is the element-wise product function. $M \in \mathbb{R}^{d \times (n+2)}$ is the retained image representation.

3.5 Cross-Modal Fusion Module

Cross-modal fusion (*CF*) module is used to obtain the final cross-modal representation for each token. The entire process can be described as follows.

Firstly, we use a gate mechanism, similar to [12, 19], to dynamically control the combination of text and image representations at the token level. Given the text representation T (see Section 3.1.1) and the retained image representation M, the token-level gate g is calculated as follows:

$$g = \sigma(W_{g_t}^T T + W_{g_m}^T M), \tag{13}$$

where W_{g_t} and $W_{g_m} \in \mathbb{R}^{d \times d}$ are weighted matrices, σ is the element-wise sigmoid function. $g \in \mathbb{R}^{d \times (n+2)}$ is the token-level gate. Then we get the final token-level image representation R as follows:

$$R = g \odot M, \tag{14}$$

where $\odot$ is the element-wise product.

To integrate the word and the image representations at token-level, we concatenate T and R to obtain the final hidden representations $H = (h_0, h_1, ..., h_{n+1})$, where $h_i \in \mathbb{R}^{2d}$.

3.6 CRF Decoder

After image information has been incorporated into all tokens, we use the conditional random field (*CRF*) decoder to perform the *MNER* task, we feed final hidden representations H (see Section 3.5)

into a standard *CRF* layer, which predicts the probability of a sequence of predictions y through the original text S (see Section 2.1) and its associated image I (see Section 2.1) as follows:

$$P(y|S, I) = \frac{exp(\sum_{i=1}^{n} E_{h_i, y_i} + \sum_{i=0}^{n} T_{y_i, y_{i+1}})}{Z(H)} \tag{15}$$

$$Z(H) = \sum_{y} exp(\sum_{i=1}^{n} E_{h_i, y_i} + \sum_{i=0}^{n} T_{y_i, y_{i+1}}), \tag{16}$$

where E_{h_i, y_i} is the emission score of label y_i for the i-th token, $T_{y_i, y_{i+1}}$ is the transition score from label y_i to label y_{i+1}, and $Z(H)$ is a normalization [9] by summation of emission and transmission scores over all possible y sequences. To train the module, we use the log-likelihood loss as our loss function, which is defined as follows:

$$\mathcal{L}_{mner} = -\frac{1}{|D_{mner}|} \sum_{j=1}^{N} \left(logP(y^j|S^j, I^j) \right), \tag{17}$$

where $D_{mner} = \{S^j, I^j, y^j\}_{j=1}^{N}$ is the batch of training examples.

3.7 Model Training

In summary, our general matching and alignment framework consists of one supervised learning task (*MNER*) and two auxiliary self-supervised learning tasks (*CA* and *CM*). We train these three tasks jointly, and the final loss function is defined as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{ca} + \beta \mathcal{L}_{cm} + (1 - \alpha - \beta) \mathcal{L}_{mner}, \tag{18}$$

where $\mathcal{L}_{ca}$ is the loss function of *CA* (see Section 3.2), $\mathcal{L}_{cm}$ is the loss function used by the *CM* (see Section 3.4). $\mathcal{L}_{mner}$ is the loss function used by the *MNER* task (see Section 3.6). α and β are the hyperparameters.

4 EXPERIMENT

4.1 Dataset

We conduct experiments on two widely used *MNER* datasets, namely `Twitter2015` [22] and `Twitter2017` [12], which are collected from Twitter. Each tweet contains a text-image pair, where the textual content may not be in the image, and the text may contain zero or more named entities. There are four types of entities: Person (*PER*), Organization (*ORG*), Location (*LOC*) and others (*MISC*). We use the pre-processed datasets provided by [19] [1]. Table 1 shows the number of entities for each type and the counts of multimodal tweets in the training, development, and test sets of the two datasets.

4.2 Metrics

We use the F1 score (**F1**) of each type and overall precision (**P**), recall (**R**) and F1 score (**F1**) to evaluate the performance of the *MNER* models, which are widely used in many recent works [2, 12, 14, 19, 22]. In this experiment, for a fair comparison, we use the code provided by [19] for evaluation [1].

4.3 Parameter Settings

We conduct all the experiments on `NVIDIA GTX 2080 Ti` GPUs with PyTorch 1.7.1. The parameter settings of our framework are as follows:

---
[1] https://github.com/jefferyYu/UMT

Table 1: The Statistics Summary of Two *MNER* Datasets.

Type	TWITTER-2015			TWITTER-2017		
	Train	Dev	Test	Train	Dev	Test
PER	2,217	552	1,816	2,943	626	621
LOC	2,091	522	1,697	731	173	178
ORG	928	247	839	1,674	375	395
MISC	940	225	726	701	150	157
Total	6,176	1,546	5,078	6,049	1,324	1,351
# Tweets	4,000	1,000	3,257	3,373	723	723

- For the text encoder, we use $BERT_{base}$ [2] in our model, which contains an encoder with 12 layers (transformer blocks), 12 self-attention heads, and the hidden size of 768. The maximum length of the input text is 128, and other parameters in it is initialized with the pre-trained *BERT* model.
- For the image encoder, we use *ResNet152* [3] in our model, which is a pre-trained 152-layer *ResNet*.
- For the *CA* module, the size of the *MLP* hidden layer is 768, and the activation function is Relu. $\tau = 0.102$, $\lambda_c = 0.7$.
- For the *CI* module, the size of multi-head attentions m is 12.
- For the *CM* module, $k = 15$.
- For the training of the entire framework, the batch size N is 64, the number of training epochs is 24, and the learning rate in the entire network is $5e^{-5}$. The text encoder is fine-tuned, and the image encoder is not fine-tuned. The hyperparameters α and β are 0.2, 0.2, respectively. The above hyperparameters are obtained through a small grid search in the development set.

4.4 Baselines

To demonstrate the effect of our general alignment and matching framework (*MAF*), we first consider several representative text-based *NER* methods:

- *BiLSTM-CRF* [8], which is a classic *NER* model with a bidirectional *LSTM* layer and a *CRF* layer.
- *CNN-BiLSTM-CRF* [13], which is an improvement of *BiLSTM-CRF*. The embedding of each word is enhanced with its word embedding and *CNN*-based character-level word representations.
- *HBiLSTM-CRF* [10], which is a variant of *CNN-BiLSTM-CRF* by replacing the *CNN* layer with an *LSTM* layer to obtain the character-level word representations.
- *BERT* [4], which is a multi-layer bidirectional transformer encoder, followed by a softmax decoder.
- *BERT-CRF*, which is a multi-layer bidirectional transformer encoder, followed by a *CRF* decoder.
- *T-NER* [15, 22], which is a tweet-specific *NER* system. It uses a set of widely used effective features, such as the dictionary, contextual and orthographic features.

²https://github.com/google-research/bert
³https://download.pytorch.org/models/resnet152b121ed2d.pth.

Besides, we also consider a comparison with several *MNER* methods:

- *GVATT-HBiLSTM-CRF* [12], which uses the *HBiLSTM-CRF* as the text encoder, and uses the attention mechanism to combine image information with text information to obtain text-aware image representation.
- *GVATT-BERT-CRF* [19], which is a variant of the *GVATT-HBiLSTM-CRF* by replacing the text encoder with *BERT*.
- *AdaCAN-CNN-BiLSTM-CRF* [22], which is based on *CNN-BiLSTM-CRF* and uses an adaptive co-attention network to decided whether to attend to the image.
- *AdaCAN-BERT-CRF* [19], which is a variant of the *AdaCAN-CNN-BiLSTM-CRF* by replacing the text encoder with *BERT*.
- *UMT-BERT-CRF* [19], which is the state-of-the-art multimodal *NER* model that including a multimodal interaction module to obtain both image-aware word representations and word-aware visual representations and an auxiliary module to leverage purely text-based entity span detection.
- *MT-BERT-CRF* [19], which is a variant of *UMT-BERT-CRF* without the auxiliary module.
- *ATTR-MMKG-MNER* [2], which is a multimodal *NER* model that introduces both image attributes and image knowledge to help improve *NER* task.
- *MAF*, which is the model we proposed in this paper.

4.5 Effectiveness

We report the metrics of F1 score (**F1**) for every single type and overall precision (**P**), recall (**R**), and F1 score (**F1**) on two benchmark *MNER* datasets. Specifically, Table 2 shows the performance of 6 text-based models and 8 multimodal models on TWITTER-2015 and 5 text-based models and 7 multimodal models on TWITTER-2017. The detailed analysis is as follows.

Firstly, we compare all text-based *NER* methods. From the table, we find that the *BERT*-based methods perform best, indicating that transfer learning helps achieve state-of-the-art results for *NER* tasks by tuning pre-trained models instead of starting from scratch. In addition, we find that the model combining *BERT* and *CRF* has better performance than the model using *BERT* only, indicating that *CRF* can indeed effectively learn the constraints of the labels in the neighborhood and jointly predict the best chain of labels.

Secondly, we compare the *MNER* methods with their corresponding text-based *NER* competitors, such as *GVATT-HBiLSTM-CRF* and *HBiLSTM-CRF*. From the table, we find that almost all multimodal models are better than their corresponding text-based competitors, indicating that the image information on social media posts is indeed helpful for named entity recognition in text.

Finally, we compare our model with all other *MNER* methods. From the table, we find that our method achieves state-of-the-art performance on both datasets, demonstrating the effectiveness of our model. Especially on the TWITTER-2017 dataset, our model outperforms the state-of-the-art model (*UMT-BERT-CRF*) by 0.94 points in overall **F1**, which shows that the two modules we proposed can help the model better combine text representation and image representation.

Table 2: Performance Comparison on Two *MNER* Dataset. For a fair comparison, we refer to the results of baselines before *BERT-CRF* and *UMT-BERT-CRF* with the marker ♣ from [19], and the result of *ATTR-MMKG-MNER* and *T-NER* with the marker ♠ from [2].

| Methods | TWITTER-2015 | | | | | | | TWITTER-2017 | | | | | | |
| | Single Type (F1) | | | | Overall | | | Single Type (F1) | | | | Overall | | |
	PER.	LOC.	ORG.	MISC.	P	R	F1	PER.	LOC.	ORG.	MISC.	P	R	F1
BiLSTM-CRF	76.77	72.56	41.33	26.80	68.14	61.09	64.42	85.12	72.68	72.50	52.56	79.42	73.43	76.31
CNN-BiLSTM-CRF	80.86	75.39	47.77	32.61	66.24	68.09	67.15	87.99	77.44	74.02	60.82	80.00	78.76	79.37
HBiLSTM-CRF	82.34	76.83	51.59	32.52	70.32	68.05	69.17	87.91	78.57	76.67	59.32	82.69	78.16	80.37
BERT	84.72	79.91	58.26	38.81	68.30	74.61	71.32	90.88	84.00	79.25	61.63	82.19	83.72	82.95
BERT-CRF♣	84.74	80.51	60.27	37.29	69.22	74.59	71.81	90.25	83.05	81.13	62.21	83.32	83.57	83.44
T-NER♠	83.64	76.18	50.26	34.56	69.54	68.65	69.09	-	-	-	-	-	-	-
GVATT-HBiLSTM-CRF	82.66	77.21	55.06	35.25	73.96	67.90	70.80	89.34	78.53	79.12	62.21	83.41	80.38	81.87
AdaCAN-CNN-BiLSTM-CRF	81.98	78.95	53.07	34.02	72.75	68.74	70.69	89.63	77.46	79.24	62.77	84.16	80.24	82.15
GVATT-BERT-CRF	84.43	80.87	59.02	38.14	69.15	74.46	71.70	90.94	83.52	81.91	62.75	83.64	84.38	84.01
AdaCAN-BERT-CRF	85.28	80.64	59.39	38.88	69.87	74.59	72.15	90.20	82.97	82.67	64.83	85.13	83.20	84.10
MT-BERT-CRF	**85.30**	81.21	61.10	37.97	70.48	74.80	72.58	91.47	82.05	81.84	65.80	84.60	84.16	84.42
UMT-BERT-CRF♣	85.24	**81.58**	63.03	39.45	71.67	**75.23**	73.41	**91.56**	84.73	82.24	**70.10**	85.28	85.34	85.31
ATTR-MMKG-MNER♠	84.28	79.43	58.97	41.47	**74.78**	71.82	73.27	-	-	-	-	-	-	-
MAF (Ours)	84.67	81.18	**63.35**	**41.82**	71.86	75.10	**73.42**	91.51	**85.80**	**85.10**	68.79	**86.13**	**86.38**	**86.25**

Table 3: Comparison of Training and Testing Time (Seconds for each Epoch) and Number of Parameters (Millions) of Models on Two *MNER* Datasets.

| Methods | TWITTER-2015 | | TWITTER-2017 | | Size (M) |
	Training	Testing	Training	Testing	
UMT-BERT-CRF	102.035	30.002	85.971	6.281	208.29
MAF	**86.822**	**25.619**	**73.754**	**5.450**	**196.28**

Table 4: Ablation Study of our Matching and Alignment Framework. We turn off the *CA* module and *CM* module and both two modules on our full model respectively, which are represented as "w/o CA", "w/o CM" and "w/o CA + CM".

| Methods | TWITTER-2015 | | | TWITTER-2017 | | |
	P	R	F1	P	R	F1
MAF	**71.41**	75.32	**73.32**	**86.13**	**86.38**	**86.25**
w/o CA	70.89	**75.44**	73.09	83.75	84.68	84.21
w/o CM	70.96	74.73	72.80	85.40	84.46	84.93
w/o CA + CM	70.32	74.71	72.45	82.90	84.30	83.60

4.6 Efficiency

We also compare the runtime and model size between the state-of-the-art model (*UMT-BERT-CRF*) and *MAF*.

As shown in Table 3, the number of parameters of *UMT-BERT-CRF* and *MAF* is 196.28 million, 208.29 million, respectively. Although our model additionally proposes a cross-modal matching module and a cross-modal alignment module, because we simplified the cross-modal interaction module, the overall model size is still smaller than *UMT-BERT-CRF*. It can also be seen from the training and testing time of the model on the two data sets that our model training time is 14.91% and 14.21% faster than UMT, and the testing time is 14.61% and 13.23% faster than *UMT-BERT-CRF*. This proves the efficiency of our model.

4.7 Ablation Study

To investigate the effectiveness of the *CA* and *CM* modules proposed in our framework, we perform comparisons between the full model *MAF* and its ablation methods.

As shown in Table 4, *MAF* benefits from the *CA* module and *CM* module. Specifically, on the TWITTER-2015 dataset, without the *CA* module, **w/o CA** drops 0.23 F1 scores; without the *CM* module, **w/o CM** drops 0.52 F1 scores; without both the *CA* module and *CM* module, **w/o CA + CM** drops 0.87 F1 scores. On the TWITTER-2017 dataset, without the *CA* module, **w/o CA** drops 2.04 F1 scores;

without the *CM* module, **w/o CM** drops 1.32 F1 scores; without both the *CA* module and *CM* module, **w/o CA + CM** drops 2.65 F1 scores. These results indicate that both the *CA* module and *CM* module play an essential role in our framework. In addition, the effect of the ablation experiment on TWITTER-2017 is more obvious than that on TWITTER-2015. We think it is because TWITTER-2015 has more noises than TWITTER-2017, so after removing the *CA* and *CM* modules, the performance of the model does not decrease significantly.

4.8 Case Study

In order to verify the effectiveness of our *CA* module and *CM* module intuitively, we select four representative cases from the test set, and compare their prediction results with *UMT-BERT-CRF* and our model. Table 5 shows their prediction results. Next, we will analyze each case in detail.

In the first two cases, the textual content is represented in the image. For the first case, *UMT-BERT-CRF* corresponds the word HURRY to the object jersey in the image instead of corresponding the word ONE to the jersey. Therefore, *UMT-BERT-CRF* incorrectly

Table 5: Four Representative Cases from the Test Set of Two *MNER* Datasets and Their Prediction Results on the State-Of-The-Art Multimodal Method (*UMT-BERT-CRF*) and Our Method (*MAF, MAF w/o CA* and *MAF w/o CM*).

Methods	Importance of the *CA* Module		Importance of the *CM* Module	
	[HURRY O] GET ONE BEFORE THEYRE SENT TO AFRICA	The beautiful camel is called [Camille MISC]	[Aquamarine MISC] (2006)	#[Malevich PER] opens at Tate Modern on 16 July
UMT-BERT-CRF	[HURRY PER] ×	[Camille MISC] ✓	[Aquamarine ORG] ×	[Malevich PER] ✓
MAF	[HURRY O] ✓	[Camille MISC] ✓	[Aquamarine MISC] ✓	[Malevich PER] ✓
MAF w/o CA	[HURRY PER] ×	[Camille PER] ×	[Aquamarine MISC] ✓	[Malevich PER] ✓
MAF w/o CM	[HURRY O] ✓	[Camille MISC] ✓	[Aquamarine ORG] ×	[Malevich LOC] ×

predicts HURRY as *PER*, which indicates that *UMT-BERT-CRF* still has a gap between the text representation and the image representation. However, the two models that retain the *CA* module (*MAF* and *MAF w/o CM*) can predict HURRY correctly, and the model (*MAF w/o CA*) without the *CA* module incorrectly predicts HURRY. For the second case, *UMT-BERT-CRF*, *MAF* and *MAF w/o CM* can all correspond the word Camille with object camel in the image, so they can correctly predict Camille as *MISC*. However, after removing the *CA* module, *MAF w/o CA* corresponds Camille with the person in the image, so *MAF w/o CA* incorrectly predicts Camille as *PER*. This shows that the *CA* module can help the model to align the text and image representations.

In the last two cases, there is no textual content in the image, and the image information will have a negative impact on the prediction result. For the third case, *UMT-BERT-CRF* is affected by the object house in the image, and Aquamarine is predicted to be *ORG*, which shows that the *UMT-BERT-CRF* cannot filter the noise brought by the image well. Even if the image has a strong interference on the prediction result, our two models (*MAF* and *MAF w/o CA*) that retain the *CM* module can still determine that Aquamarine is a movie based on the year information (2006) in the text, and then predict it as *MISC*. The model (*MAF w/o CM*) without the *CM* module predicts Aquamarine as *ORG* like *UMT-BERT-CRF*. For the fourth example, *UMT-BERT-CRF*, *MAF*, and *MAF w/o CA* can filter out the noise of the image very well, so Malevich is correctly predicted as *PER*. However, after removing the *CM* module, the *MAF w/o CM* is affected by the geometric shapes in the image, which incorrectly predicts Malevich as *LOC*. This shows that the *CM* module can reduce the impact of mismatched text-image pairs.

5 RELATED WORK

In this section, we review and summarize the works that are most relevant to our research.

Starting with [14], multimodal named entity recognition (*MNER*) has become an important research direction in named entity recognition (NER) which significantly extends the conventional text-based *NER* by taking images as additional inputs [20].

The critical challenge is how to combine text representation with image representation. [14] first proposed an *LSTM-CNN* architecture that combines text with image information via a general modality attention module. [12] propose an attention-based model to extract image features from the regions in the image most related to the text and use a gate to combine text features and image features. [22] propose an adaptive co-attention network to dynamically control the combination of text representation and image representation. [19] propose a multimodal transformer architecture for the task of *MNER*, which captures the inter-modal interactions with a multimodal interaction module. [2] proposed a novel neural network model that introduces image attributes and image knowledge to help improve model performance for *MNER*.

However, these methods are all assumed that each text and its accompanying image are matched, and the image can be used to help identify named entities. Moreover, they fail to construct a consistent representation to bridge the semantic gap between the two modalities. Therefore, in this paper, we propose a general matching and alignment framework for *MNER* task, which can reduce the impact of mismatched between text and images and make the representations between the two modalities more consistent.

6 CONCLUSION

In this paper, we propose *MAF*, a general matching and alignment framework, which improves state-of-the-art performance on multimodal named entity recognition for social media posts. Specifically, we propose a cross-modal alignment module based on contrastive learning to make the text representations and image representations more consistent, and propose a cross-modal matching module to determine the proportion of image information that should be retained. We conduct extensive experiments, ablation studies and case studies to show that the *CA* module can help the model establishing a connection between the named entity in the text and the region where the corresponding object in the image is located, and reduce the interaction with other regions in the image. The *CM* module can help the model filter out most of the image information that is not related to the text, reducing the impact of mismatched images on the text.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *stat* 1050 (2016), 21.

[2] Dawei Chen, Zhixu Li, Binbin Gu, and Zhigang Chen. 2021. Multimodal Named Entity Recognition with Image Attributes and Image Knowledge. In *Database Systems for Advanced Applications*. 186–201.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[6] Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2020. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *International Conference on Learning Representations*.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[9] Fernando C. N. Pereira John D. Lafferty, Andrew McCallum. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of the 18th Intl. Conf. on Machine Learning (ICML-2001)*. 282–289.

[10] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*. 260–270.

[11] Tian Li, Xiang Chen, Shanghang Zhang, Zhen Dong, and Kurt Keutzer. 2021. Cross-domain sentiment classification with contrastive learning and mutual information maximization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8203–8207.

[12] Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1990–1999.

[13] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1064–1074.

[14] Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. 2018. Multimodal Named Entity Recognition for Short Social Media Posts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 852–860.

[15] Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*. 1524–1534.

[16] Erik F Sang and Jorn Veenstra. 1999. Representing text chunks. *arXiv preprint cs/9907006* (1999).

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[18] Alakananda Vempala and Daniel Preoţiuc-Pietro. 2019. Categorizing and inferring the relationship between the text and image of twitter posts. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*. 2830–2840.

[19] Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020. Improving Multimodal Named Entity Recognition via Entity Span Detection with Unified Multimodal Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3342–3352.

[20] Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. 2021. Multi-modal Graph Fusion for Named Entity Recognition with Targeted Visual Guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 14347–14355.

[21] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. 2021. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 833–842.

[22] Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[23] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. 2020. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747* (2020).