The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Adaptive Co-Attention Network for Named Entity Recognition in Tweets

Qi Zhang, Jinlan Fu, Xiaoyu Liu, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, P.R. China
{qz, fujl16, liuxiaoyu16, xjhuang}@fudan.edu.cn

Abstract

In this study, we investigate the problem of named entity recognition for tweets. Named entity recognition is an important task in natural language processing and has been carefully studied in recent decades. Previous named entity recognition methods usually only used the textual content when processing tweets. However, many tweets contain not only textual content, but also images. Such visual information is also valuable in the name entity recognition task. To make full use of textual and visual information, this paper proposes a novel method to process tweets that contain multimodal information. We extend a bi-directional long short term memory network with conditional random fields and an adaptive co-attention network to achieve this task. To evaluate the proposed methods, we constructed a large scale labeled dataset that contained multimodal tweets. Experimental results demonstrated that the proposed method could achieve a better performance than the previous methods in most cases.

Introduction

Named entity recognition (NER) tries to identify the named entities in text. Named entities fall into pre-defined categories such as the names of persons, organizations, and locations. Along with the rapid development of social media (e.g., Twitter, Facebook), tweets have become important resources for various applications such as breaking news aggregation (Ritter et al. 2012; Cui et al. 2012), the identification of cyber-attacks (Ritter et al. 2015) or natural disasters (Neubig et al. 2011; Bruns and Liang 2012), and mining disease outbreaks (Paul and Dredze 2011). Hence, locating and classifying named entities from tweets has become one of the indispensable tasks of these applications.

Previous methods have studied the NER problem from different aspects. Various supervised methods with manually constructed features have been proposed to perform this task. Zhou and Su (2002) proposed the use of an HMM-based chunk tagger. Chieu et al. (2002) introduced a maximum entropy approach with global information. Support vector machines have also been used on the NER task (Isozaki and Kazawa 2002). Because knowledge plays an important role in the NER task, different kinds of resources have also been taken into consideration (Borthwick et al. 1998; GuoDong and Jian 2004; Kazama and Torisawa 2007). Numerous methods have been introduced(Ritter et al. 2011; Liu et al. 2011; Li et al. 2012; de Oliveira et al. 2013; Li et al. 2015) to process the content posted by users on social media sites. Ritter et al. (2011) treated the classification and segmentation of named entities as separate tasks. Large scale open-domain ontology is included for classification. Li et al. (2015) introduced an iterative method to split tweets into meaningful segments and evaluated the method on the NER task. Because of the characteristics of social media, the performances of methods on the tweets are much lower than the performances on newswire domain.

Figure 1: An example of multimodal tweets. In this tweet, "*Rocky*" is the name of the dog.

We can see that most of the previous methods only considered the textual content. However, many tweets contain not only textual content, but also images. Figure 1 gives a multimodal tweet example. If we only have the tweet content "Rocky is ready for snow season," "Rocky" is usually recognized as the name of a person by humans and the previous methods. However, in this tweet, "Rocky" should be labeled as belonging to the category of animal names. Without taking visual information into consideration, various tweets cannot be easily recognized. Moreover, according to a statistic, more than 42% of tweets contain more than one image[1]. Hence, processing these multimodal tweets has become an

[1]https://thenextweb.com/socialmedia/2015/11/03/what-analyzing-1-million-tweets-taught-us/

important task.

To incorporate visual information for the recognition of named entities on multimodal tweets, this paper proposes a novel neural network based method to achieve the task. Following previous methods, we also convert the NER task into a sequence labeling problem. Hence, we extend the bi-LSTM-CRF method (Huang, Xu, and Yu 2015), which has been successfully used in various sequence labeling tasks, as the basic architecture. Because of the 140 character limitation of Twitter and the oral language environment, tweets contain numerous nonstandard spellings, abbreviations, and unreliable capitalization. We combine the character embeddings and word embeddings as the input. To incorporate the visual information, we propose an adaptive co-attention network to join the textual and visual information together. In visual attention, we can capture the image regions that the word at time step t is more related to based on the prediction. In textual attention, we can obtain the words in the text that the word at time step t is more related to based on prediction. Because not all of the words in the text have corresponding visual signals such as "a," "an," and "directly," we also introduce a gated multimodal fusion module to decide when to rely on visual information. To filter out the noise brought by the visual information, we use a filtration gate module as a filter. To evaluate the proposed method, we constructed a large-scale dataset, which contained manually labeled multimodal tweets. Experimental results on our dataset showed that our method could achieve better performance than previous methods.

The main contribution of this paper can be summarized as follows:

- The task of recognizing named entities on multimodal tweets is novel and has not been carefully studied in previous methods. In this paper, we defined the problem and evaluated several methods for this task.
- We introduced an adaptive co-attention network that combines visual and textual information to recognize named entities.
- We constructed a large-scale dataset to evaluate the performances of different NER methods on multimodal tweets.

The Proposed Method

In this work, we propose a novel neural network architecture called the Adaptive Co-attention Network (ACN) that will learn the shared semantics between text and images. The architecture of our proposed method is shown in Figure 2. For clarity, we describe our model in three parts:*Feature Extractor*, *Adaptive Co-attention Network*, and *CRF Tagging Models*. We will illustrate the details of the proposed framework in the following section.

Feature Extractor

Image Feature Extraction The image features were extracted from 16-layer VGGNet (Simonyan and Zisserman 2014). Previous studies (Gao et al. 2015) used features from the last layer that produced a global vector, but we want the spatial features of different regions. Therefore, we chose the features from the last pooling layer. We first resized the images to 224×224 pixels, and then retained the features from the last pooling layer, which has a dimension of $512 \times 7 \times 7$. The 512 number is the dimension of the feature vector for each region, and 7×7 is the number of regions. Therefore, an image could be represented as $\tilde{v}_I = \{\tilde{v}_i | \tilde{v}_i \in \mathbb{R}^{d_v}, i = 1, 2, \cdots, N\}$, where $N = 7 \times 7$ is the number of image regions, and $\tilde{v}_i$ is a 512 dimensional feature vector for image region i.

For calculation convenience, we transform each feature vector to a new vector with the same dimensions as the text vector using a single layer perceptron.

$$v_I = tanh(W_I \tilde{v}_I + b_I), \tag{1}$$

where $\tilde{v}_I$ is the output of 16-layer VGGNet, and v_I is the image feature map after transforming by single layer perceptron.

CNN for Character-level Representation Character-level embedding could alleviate rare word problems and capture helpful morphological information, like prefixes and suffixes. Let $\mathcal{C}$ be the vocabulary of characters. After a character lookup table, a word will be projected to a sequence of character vectors: $[c_1, c_2, \cdots, c_m]$, where $c_i \in \mathbb{R}^{d_c}$ is the vector for the i-th character in the word and m is word length. For the convolutional operation, k groups of filter matrices $[C_1, C_2, \cdots, C_k]$ with different sizes $[l_1, l_2, \cdots, l_k]$ are applied. The transformed sequences F_j will be obtained as follows:

$$F_j = [\cdots; tanh(C_j \cdot F_{[i:i+l_j-1]} + b_j); \cdots],$$

where i is the index of the convolutional window. Then, we apply a max-over-time pooling operation over the feature map, and take the maximum value as the feature corresponding to filter C_j.

$$w_j^{'} = max(F_j).$$

Finally, we obtain the representation $w^{'}$ for a word by concatenating all the mappings.

$$w^{'} = [w_1^{'} \oplus w_2^{'} \oplus \cdots w_k^{'}].$$

Bidirectional LSTM The Long Short-Term Memory network (LSTM) (Hochreiter and Schmidhuber 1997) is a variant recurrent neural network (RNN) designed to solve the issue of learning long-term dependencies. Formally, at time t, the memory c_t and the hidden state h_t are updated with the following equations:

$$\begin{bmatrix} \tilde{\mathbf{c}}_t \\ \mathbf{o}_t \\ \mathbf{i}_t \\ \mathbf{f}_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} T_{\mathbf{A},\mathbf{b}} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix},$$

$$\mathbf{c}_t = \tilde{\mathbf{c}}_t \odot \mathbf{i}_t + \mathbf{c}_{t-1} \odot \mathbf{f}_t,$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

where $\odot$ is the element-wise product and σ is the element-wise sigmoid function. i_t, f_t, and o_t denote the input, forget, and output gates at time step t respectively. x_t is the input

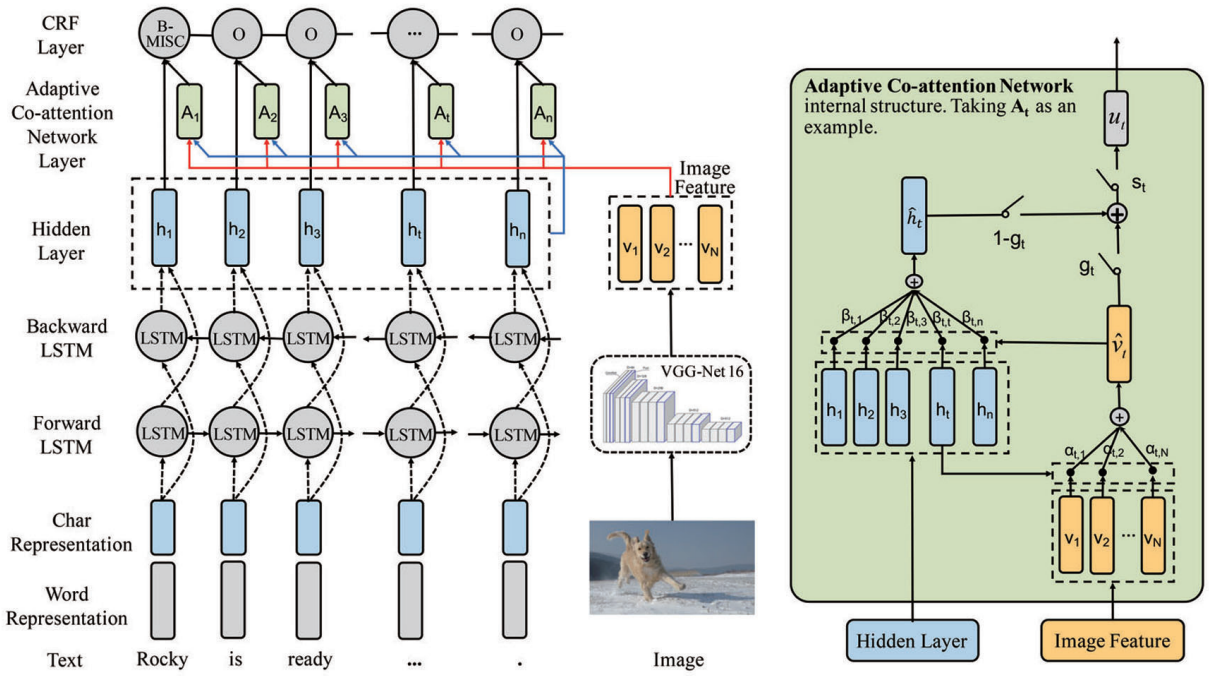

Figure 2: The general architecture of the proposed approach. The left part is the main framework of this work. The right part is the detailed structure of the adaptive co-attention network.

vector at time t, and $T_{A,b}$ is an affine transformation, which depends on parameters of the network A and b.

Notice that LSTM takes only past information. However, context information from the future could also be crucial. To capture the context from both the past and the future, our feature extractor is a bidirectional LSTM. Bidirectional LSTM contain two separate LSTMs to capture both past and future information, where one encodes the sentence from start to end, and the other encodes the sentence from end to start. Thus, at each time state t, we can obtain two representations, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, then the two representations are concatenated to form the final output:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]. \tag{2}$$

Given a sentence $x = (x_0, x_1, ..., x_n)$, we use CNN to extract the character-level word representation $w^{'}$. Then, we get the ordinary word representation $w^{''}$ using the word look-up table, and we obtain a feature-rich word representation by concatenating the two representations at position t, i.e. $w_t = [w_t^{''}, w_t^{'}]$. We encode the sentence with a bidirectional LSTM, and by replacing the word w_t with h_t, we can interpret h_t as a representation summarizing the word at position t along with its contextual information. So we get the textual feature matrix $x = \{h_j | h_j \in \mathbb{R}^d, j = 1, 2, \cdots, n\}$, where d is the dimension of the LSTM's hidden state and n is the sentence length.

Adaptive Co-attention Network

The Adaptive Co-attention Network (ACN) is a multimodal model using the co-attention process, which includes visual attention and textual attention to capture the semantic interaction between different modalities. We propose the use of a gated multimodal fusion module to fuse the features adaptively. Then, to reduce the possibility of noise introduced in the multimodal, we utilize the filtration gate to adaptively filter out some of the useless multimodal information. The details of this network will be illustrated as follows.

Word-Guided Visual Attention In most cases, a word is related to a small region of the input image. Therefore, applying the whole image feature with a word feature to predict the label could lead to suboptimal results because regions that are irrelevant to the word introduce noise. Instead, we apply a word-guided visual attention module to decide which image region to attend to. Further, our model has the ability to gradually filter out the noise and pinpoint the regions that are highly relevant to the current word.

Given a word feature, h_t is obtained by Equation (2) and the image feature matrix v_I is obtained by Equation (1). We feed these through a single layer neural network followed by a softmax function to generate the attention distribution over the N regions of the image:

$$z_t = tanh(W_{v_I} v_I \oplus (W_{h_t} h_t + b_{h_t})),$$
$$\alpha_t = softmax(W_{\alpha_t} z_t + b_{\alpha_t}),$$

where $h_t \in \mathbb{R}^d$, d is the dimension of word or image feature, $v_I \in \mathbb{R}^{d \times N}$, N is the number of image regions. W_{v_I}, W_{h_t} and W_{α_t} are parameters, and W_{v_I}, $W_{h_t} \in \mathbb{R}^{k \times d}$ and $W_{\alpha_t} \in \mathbb{R}^{1 \times 2k}$. $\alpha_t \in \mathbb{R}^N$, which corresponds to the attention probability of each image region given h_t. In addition, we use $\oplus$ to denote the concatenation of the image feature

matrix and word feature vector. The concatenation between a matrix and a vector is performed by concatenating each column of the matrix by the vector.

Based on the attention distribution $\alpha_{t,I}$, which is the weight corresponding to each image region, the new image vector related to word h_t can be obtained by:

$$\hat{v}_t = \sum_i \alpha_{t,i} v_i.$$

Image-Guided Textual Attention In the previous section, we used word-guided visual attention to obtain a new image representation that is relevant to word h_t at position t, in other words, word-guided visual attention decided that predicting the label for h_t should attend to which image regions. However, we have no idea which words in the text are more relevant to word h_t. Therefore, we propose an image-guided textual attention model using the new image vector $\hat{v}_t$ to conduct the textual attention. Then, we acquire a new representation, $\hat{h}_t$, for text based on the energy function and probability distributions.

$$\begin{aligned} z_t^{'} &= tanh(W_x x \oplus (W_{x,\hat{v}_t}\hat{v}_t + b_{x,\hat{v}_t})), \\ \beta_t &= softmax(W_{\beta_t} z_t^{'} + b_{\beta_t}), \\ \hat{h}_t &= \sum_j \beta_{t,j} h_j, \end{aligned}$$

where $x = (h_1, h_2, \cdots, h_n)$, $x \in \mathbb{R}^{d\times n}$ and $\hat{v}_t \in \mathbb{R}^d$. n is the maximum length of the tweet, and d is the dimension of feature representation. W_x, $W_{x,\hat{v}_t} \in \mathbb{R}^{k\times d}$ and $W_{\beta_t} \in \mathbb{R}^{1\times 2k}$, thus $\beta_t \in \mathbb{R}^n$, which is the attention distributions of word in the sentence. Similar to visual attention, we use $\oplus$ to denote the concatenation of the tweet feature matrix and image vector, and we concatenate each column of the matrix using the vector.

Gated Multimodal Fusion Based on the visual and textual attention, we propose a gated multimodal fusion (GMF) method to fuse the multimodal features. In named entity recognition tasks, when predicting the label of a word, GMF trades off how much new information the network is considering from the image with the text containing the word. A GMF is used as an internal unit to find an intermediate representation feature based on a combination of features from different modalities. For each word, we obtain a new visual feature based on visual attention and a new textual feature based on textual attention, then we compute a multimodal gate to fuse them because of the different dependencies on images for different words. The GMF is defined as:

$$\begin{aligned} h_{\hat{v}_t} &= tanh(W_{\hat{v}_t}\hat{v}_t + b_{\hat{v}_t}), \\ h_{\hat{h}_t} &= tanh(W_{\hat{h}_t}\hat{h}_t + b_{\hat{h}_t}), \\ g_t &= \sigma(W_{g_t}(h_{\hat{v}_t} \oplus h_{\hat{h}_t})), \\ m_t &= g_t h_{\hat{v}_t} + (1 - g_t) h_{\hat{h}_t}, \end{aligned}$$

where $W_{\hat{v}_t}$, $W_{\hat{h}_t}$, W_{g_t} are parameters, $h_{\hat{v}_t}$ and $h_{\hat{h}_t}$ are the new image vector and new text vector, respectively, after transformation by single layer perceptron. $\oplus$ is the concatenating operation, σ is the logistic sigmoid activation, g_t is the gate applied to the new image vector $h_{\hat{v}_t}$, and m_t is the multimodal fusion feature between the new visual feature and new textual feature.

Filtration Gate Because the named entity recognition task is built on the text, text information is the most important feature and cannot be ignored. In our model, the word at position t is applied to guide the visual attention in our adaptive co-attention network. In addition, we also use the word at position t as a part of the input to the decoder, while the other part is the multimodal fusion feature m_t. But, the visual feature is unnecessary, when predicting the label of a verb or adverb. Because the multimodal fusion feature contains visual feature more or less and it may introduce some noise, we use a filtration gate to combine features from different signals that better represent the information needed to solve a particular problem. The filtration gate is a scalar in the range of $[0, 1]$. When the multimodal fusion feature is helpful to improve the performance of a certain type of word, the filtration gate is 1, otherwise, the value of the filtration gate is 0. The filtration gate s_t and the input feature to the decoder $\hat{m}_t$ are defined as follows:

$$\begin{aligned} s_t &= \sigma(W_{s_t,h_t} h_t \oplus (W_{m_t,s_t} m_t + b_{m_t,s_t})), \\ u_t &= s_t(tanh(W_{m_t} m_t + b_{m_t})), \\ \hat{m}_t &= W_{\hat{m}_t}(h_t \oplus u_t), \end{aligned}$$

where W_{s_t,h_t}, W_{m_t,s_t}, W_{m_t}, $W_{\hat{m}_t}$ are parameters, h_t is the hidden state of bidirectional LSTM at time t, u_t is the reserved multimodal features after filtration gate filter out noise, and $\oplus$ is the concatenating operation.

CRF Tagging Models

It has been shown that Conditional Random Fields (CRF) can produce higher tagging accuracy in sequence labeling tasks because CRF considers the correlations between labels in neighborhoods. For example, an adjective has a greater probability of being followed by a noun than a verb in POS tagging task, and **I-PER** cannot follow **B-LOC** in NER with a standard BIO2 annotation (Sang and Veenstra 1999). Therefore, instead of decoding each label independently, we model them jointly using a CRF.

We use $X = \{w_0, w_1, \cdots, w_T\}$ to represent a generic input sequence, where w_i is the vector of the i-th word. $y = \{y_0, y_1, \cdots, y_T\}$ represents a generic sequence of labels for X. Y denotes all possible tag sequences for a sentence X. Given sequence X, all the possible label sequences y can be calculated by the following equation:

$$p(y|X) = \frac{\prod_{i=1}^{T} \Omega_i(y_{i-1}, y_i, X)}{\sum_{y^{'}\in Y} \prod_{i=1}^{T} \Omega_i(y_{i-1}^{'}, y_i^{'}, X)},$$

where $\Omega_i(y_{i-1}, y_i, X)$ and $\Omega_i(y_{i-1}^{'}, y_i^{'}, X)$ are potential functions.

We use the maximum conditional likelihood estimation for CRF training. The logarithm of likelihood is given by:

$$L(p(y|X)) = \sum_i logp(y|X).$$

Maximum conditional likelihood logarithm tries to learn parameters that maximize the log-likelihood $L(p(y|X))$.

In the decoding phrase, we predict the output sequence that obtains the maximum score given by:

$$y^* = argmax_{y' \in Y} p(y|X).$$

Experiment

In this section, we evaluate our method on a manually annotated datasets and show that our system outperforms the baselines. Precision (Prec.), Recall, and F1 are used as the evaluation metrics in this work.

Datasets Construction

Datasets Collection A collection of tweets will be used in the experiments. We use Twitter's API[2] to collect the tweets. The collection includes 26.5 million tweets. From these tweets, we drop the non-English tweets and extract those containing images, leaving 4.3 million tweets. The tweets we collected through Twitter's API are based on users, and the topics covered are diverse in nature. However, those tweets are extremely relevant to the users' hobbies and habits. In order to reduce the specificity introduced by users' preferences, we randomly sampled 50,000 tweets from those containing images, then labeled them with two independent annotators.

	Train	Dev	Test	Total
Person	2217	552	1816	4583
Location	2091	522	1697	4308
Organization	928	247	839	2012
Misc	940	225	726	1881
Total Entity	6176	1546	5078	12784

Table 1: Named entity type counts in the train, development and test sets.

Manual Annotation One important question was which entity types should be covered. Some Twitter NER datasets have applied 10 top-level Freebase categories (Ritter et al. 2011; Strauss et al. 2016) or have included product classes in addition to PLO (Person, Location, Organization) (Liu et al. 2011). However, most of the existing NER datasets only focus on entity classes: Person, Location, Organization, such as NER datasets construct by Derczynski et al. (2016) and Finin et al. (2010), or focus on entity classes: Person, Location, Organization and Misc, such as CoNLL-2003(Tjong Kim Sang and De Meulder 2003). Therefore, we follow the standard annotation naturally, and the entity types in our datasets are **Person**, **Location**, **Organization**, and **Misc**. Apart from being well understood by annotators, these four categories match the other existing NER datasets well, and thus it could be combined with other NER datasets for domain transfer learning, multitask learning, and so on.

We use the *BIO2* (Sang and Veenstra 1999) annotation standard in this work. We remove the tweets which do not mention any named entity, following the CoNLL-2003 and other NER data. In addition, tweets with length less than 3 or hardly to be understood were filtered out. Finally, we get 8,257 tweets posted by 2116 users. The total number of entities is 12,784. We split the dataset into three parts: training set, development set, and testing set, which contain 4,000, 1,000, and 3,257 tweets, respectively. The named entity type counts in the training, development, and test sets are shown in Table 1.

Baselines

In this part, we will describe the models in the comparisons of our main experiments. Our experiments mainly concern two groups of models: previous state-of-the-art models and the variant models of our model. The models are listed as follows:

Previous State-of-the-art Methods: To illustrate how well our model can handle the named entity recognition task, we compare our model to the following existing state-of-the-art models.

- **T-NER**: The T-NER [3] proposed by Ritter et al. (2011) is a tweet-specific NER system. T-NER used a set of widely-used effective features, including dictionary, contextual and orthographic features. We applied T-NER to train a model with our training set, then evaluated it using our testing set.
- **Stanford NER**: The Stanford NER[4] proposed by Finkel et al. (2005) is a widely used tool for the named entity recognition task. Similarly, we trained the Stanford NER with our training set, then evaluated it using our testing set.
- **BiLSTM+CRF**: BiLSTM+CRF was proposed by Huang et al. (2015). Unlike the original model, we did not use any hand-made features, such as spelling features and context features.
- **CNN+BiLSTM+CRF**: This model was proposed by Ma and Hovy (2016) and is a truly end-to-end system, requiring no feature engineering or data preprocessing. Thus, it is suitable for many sequence labeling tasks. It was reported to have achieved the best result (F1-Measure of 91.21%) on the CoNLL 2003 test set.

Variant Models To analyze the contribution of each component in our model, we ablate the full model and demonstrate the effectiveness of each component.

- **CBCFuFi**: This model is a part of our model without the CRF component. Instead, we use softmax as a multi-classifier after combining both the multimodal feature and word feature.
- **CBCFiC**: This model is a variant of our model, but this model has no fusion gate. It directly concatenates the features from different modalities. At each time step, we use a filtration gate to filter out the noise introduced by image, then we concatenate the multimodal feature with the word feature to make the CRF input feature.

[2]https://dev.twitter.com

[3]https://github.com/aritter/twitter_nlp

[4]https://nlp.stanford.edu/software/CRF-NER.shtml

	PER. F1	LOC. F1	ORG. F1	MISC F1	Overall Prec.	Overall Recall	Overall F1
T-NER* (Ritter et al. 2011)	64.00	59.85	22.43	15.67	44.07	56.16	49.38
CNN+BiLSTM+CRF* (Ma and Hovy 2016)	71.25	67.73	38.33	21.98	60.55	54.84	57.56
Stanford NER (Finkel, Grenager, and Manning 2005)	73.85	69.35	41.81	21,80	60.98	62.00	61.48
BiLSTM+CRF (Huang, Xu, and Yu 2015)	76.77	72.56	41.33	26.80	68.14	61.09	64.42
CNN+BiLSTM+CRF (Ma and Hovy 2016)	80.86	75.39	47.77	32.61	66.24	68.09	67.15
T-NER (Ritter et al. 2011)	**83.64**	76.18	50.26	**34.56**	69.54	68.65	69.09
CBCFuFi (CNN+BiLSTM+Coatt+Fusion+Filtration)	80.79	74.94	46.63	28.45	67.69	65.00	66.32
CBCFiC (CNN+BiLSTM+Coatt+Filtration+CRF)	81.40	77.27	51.04	32.61	67.77	68.48	68.12
CBCFuC (CNN+BiLSTM+Coatt+Fusion+CRF)	82.67	77.49	**53.15**	32.72	69.28	**69.05**	69.16
Our Model (CNN+BiLSTM+Coatt+Fusion+Filtration+CRF)	81.98	**78.95**	53.07	34.02	**72.75**	68.74	**70.69**

Table 2: The performances of different approaches on our datasets. The first part is the results of classic methods. The T-NER* and CNN+BiLSTM+CRF* were trained on the WNUT-16 datasets and then evaluated on our testing set. The second part is the main results of our method's variant models.

- **CBCFuC**: This model is also a variant of our model without the filtration gate. After obtaining the fused features, both visual attention and textual attention, we concatenate the fusion feature with the word feature at this time step, then the CRF is applied for learning and inference.

Parameter Setting

To initialize the word embeddings used in our model, we pre-trained the word embedding on 30 million tweets, and the dimension was set to 200. Words that are out of the embedding vocabulary are initialized by randomly sampling from a uniform distribution of $[-0.25, 0.25]$. The dimension for character embeddings is set to 30, and is initialized randomly from a uniform distribution of $[-0.25, 0.25]$. The sentence length is set to 35, the word length is set to 30, and we apply truncating or zero-padding as necessary. In the character-level module, we use three groups of 32 filters, with window sizes of (2,3,4), while the output dimension of the bidirectional LSTM is set to 200. The optimizer is Rmsprop, and its learning rate is set to 0.19.

Results and Discussion

The results on our manually annotated datasets are shown in Table 2, and we have gathered several experiment findings from the results.

Discussion on State-of-the-art Methods First, it is obvious that the open tool *T-NER* trained with our training set achieved a better performance compared to other classic methods, and the well-known *Stanford NER* trained with our training set achieved a lower performance. In addition to the differences of the manual features and knowledge bases used, *Stanford NER* was designed for newswire NER datasets, and thus it did not adapt to user-generated text. However, *T-NER* was devised specifically for NER of Twitter text, and naturally, the evaluation results of that method are better.

Second, our model is more suitable to other datasets or other sequence labeling tasks, compared to the *T-NER* system. Because of many specific hand-crafted features and the particular dictionaries introduced, the *T-NER* system achieved competitive performance by training on our training set, then evaluating on our testing set. However, when *T-NER* trained on the WNUT-16 datasets and was evaluated on our testing set, the performance dropped sharply to 49.38%. We can observe the results of this setting through the *T-NER** in Table 2. But, our single modal model *CNN+BiLSTM+CRF* trained on the WNUT-16 datasets achieved 57.56% performance, and we can obtain this result from *CNN+BiLSTM+CRF**. Therefore, the *T-NER* system is heavily dependent on a specific corpus, and our model is more applicable to other datasets due to the better performance achieved and the fact that it requires no feature engineering.

Third, the performance improved a lot when image information is introduced. Our model is based on *CNN+BiLSTM+CRF* by adding an ACN module that can deal well with the image feature, and the performance of our model is much better than *CNN+BiLSTM+CRF*. Moreover, the performance of our model is better than other methods that introduced many specific hand-crafted features and the particular dictionaries such as *T-NER* and *Stanford NER*. In addition, the performance of *BiLSTM+CRF* is lower than all of our variant models. Therefore, the performance improves a lot, when introducing the image for named entity recognition in tweets.

Discussion on Variant Models All the components of our model play an important role in improving performance. If any component is missing, then the performance will decrease.

CBCFuFi is a variant of our model without the CRF component. Compared to our model's performance, the F1 score is reduced by 4.37%. By adding the CRF component for joint decoding, we can achieve significant improvements.

CBCFiC is also a variant of our model without fusion gate. The role of the fusion gate is to fuse features from different modalities. The F1 score of this model is decreased by 2.57%. Our model improved a lot by adaptively considering how much the image feature is fused into the multimodal feature at each time step.

CBCFuC is a mutation of our model without the filtration

gate component. Filtration gate is to decide how much the multimodal feature was introduced into the decoding layer. The F1 score dropped by 1.53% compared to the performance of our model. Therefore, eliminating the multimodal noise introduced into our model is necessary, and the filtration gate can effectively solve this problem.

Parameter Sensitivity

In this section, we evaluate our model on different settings of the parameters. Specifically, we are concerned about the impact of dropout and the dimensions of the parameters.

	Overall	PER.	LOC.	ORG.	MISC
No	68.16	81.41	76.75	49.41	30.79
Yes	**70.69**	**81.98**	**78.95**	**53.07**	**34.02**

Table 3: Results with and without dropout on our datasets (F1 score).

First, we compared the results achieved by our model with and without dropout layers, and show those results in Table 3. All other hyper-parameters remain the same as our best model. After using dropout, the F1 score has improved in each category and overall. This demonstrates the effectiveness of dropout in reducing overfitting. Dropout is essential for state of the art performance, and the improvement is statistically significant. Our model achieved an essential and improved performance, because of introducing dropout.

Dim.	Overall	PER.	LOC.	ORG.	MISC
100	69.83	82.28	75.77	54.25	30.31
150	69.94	81.41	77.83	53.51	32.94
200	70.69	81.98	78.95	53.07	34.02
250	69.12	81.32	76.30	49.40	28.54
300	68.77	81.46	75.38	50.68	33.44
400	68.42	80.28	76.85	49.13	33.09

Table 4: Results of our proposed model influenced by different embedding dimension (F1 score).

Second, we evaluated our model on different parameter's dimensions. The bidirectional LSTM hidden state dimension is equal to the image vector dimension. Because of the need for calculations in our work, we set all of the parameters' dimensions to be hidden state dimensions. We listed the result our model achieved on different parameter dimensions, as shown in Table 4. We discovered that the closer the dimensions of the parameters are to the word embedding dimension, the better the performance is. In our work, the word embedding dimension was set to 200, and we can see that when the dimension equals 200, we get the best results in our model.

Related Work

Traditionally, high-performance approaches in named entity recognition task require large amounts of specific knowledge and hand-crafted features. Kazama and Torisawa (2007) explored the use of Wikipedia as external knowledge to improve named entity recognition. For each candidate word sequence, the method retrieved the corresponding Wikipedia entry and extracted a category label. These category labels were used as features in a CRF-based NE tagger. Recently, some methods combine traditional methods with neural networks methods have been proposed. Huang et al. (2015) extracted spelling features and context features to enhance the performance of NER. These features were connected with word features obtained from BiLSTM, then used as inputs features of CRF decoder. After that, systems that do not employ feature engineering, proprietary lexicons, hand-made features, and rich entity linking information were proposed. Ma et al. (2016) proposed a neutral network architecture that benefits from both word- and character-level representations automatically, by using the combination of bidirectional LSTM, CNN and CRF. This system is truly end-to-end, required no feature engineering or data preprocessing.

In recent years, NER in informal texts such as status messages on Twitter has raised concern. Ritter et al. (2011) proposed LabeledLDA to exploit Freebase dictionaries and developed a supervision T-NER system. Liu et al.(2011) combined a K-Nearest Neighbors (KNN) classifier with a linear Conditional Random Fields (CRF) model under a semi-supervised learning. Li et al. (2012) proposed an unsupervised NER system for the targeted Twitter stream, called TwiNER. This approach dealt with the stream, and only identified if a phrase is an entity or not. Li et al. (2015) proposed a novel framework for tweet segmentation, called HybridSeg, and applied the segment-based part-of-speech (POS) tagging in named entity recognition.

Conclusion

In this work, we proposed a novel multimodal model for named entity recognition in tweets, which consider the image posted by users. We introduced an adaptive co-attention network to decided whether to attend to the image. And if so, to which regions. We introduced a gated multimodal fusion module to decide how much visual features are fused into the network at each time step. We further introduced a filtration gate module to adaptively adjust how much multimodal information is to be considered at each time step. Because of the constraints between the adjacent labels, we used CRF as a decoder in our model. In addition, we built a multimodal named entity recognition datasets for the tweets, then evaluated our model using it. Compared to other state-of-the-art models that use only text information, the performance of the proposed method is much better.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61532011, 61473092, and 61472088) and STCSM (No.16JC1420401,17JC1420200).

References

Borthwick, A.; Sterling, J.; Agichtein, E.; and Grishman, R. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proc. of the Sixth Workshop on Very Large Corpora*, volume 182.

Bruns, A., and Liang, Y. E. 2012. Tools and methods for capturing twitter data during natural disasters. *First Monday* 17(4).

Chieu, H. L., and Ng, H. T. 2002. Named entity recognition: a maximum entropy approach using global information. In *ACL-2002*, 1–7. Association for Computational Linguistics.

Cui, A.; Zhang, M.; Liu, Y.; Ma, S.; and Zhang, K. 2012. Discover breaking events with popular hashtags in twitter. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 1794–1798. ACM.

de Oliveira, D. M.; Laender, A. H.; Veloso, A.; and da Silva, A. S. 2013. Fs-ner: a lightweight filter-stream approach to named entity recognition on twitter data. In *Proceedings of the 22nd International Conference on World Wide Web*, 597–604. ACM.

Derczynski, L.; Bontcheva, K.; and Roberts, I. 2016. Broad twitter corpus: A diverse named entity recognition resource. In *COLING*, 1169–1179.

Finin, T.; Murnane, W.; Karandikar, A.; Keller, N.; Martineau, J.; and Dredze, M. 2010. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 80–88. Association for Computational Linguistics.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, 363–370. Association for Computational Linguistics.

Gao, H.; Mao, J.; Zhou, J.; Huang, Z.; Wang, L.; and Xu, W. 2015. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in Neural Information Processing Systems*, 2296–2304.

GuoDong, Z., and Jian, S. 2004. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, 96–99. Association for Computational Linguistics.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Isozaki, H., and Kazawa, H. 2002. Efficient support vector classifiers for named entity recognition. In *ACL-2002-Volume 1*, 1–7. Association for Computational Linguistics.

Kazama, J., and Torisawa, K. 2007. Exploiting wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 698–707.

Li, C.; Weng, J.; He, Q.; Yao, Y.; Datta, A.; Sun, A.; and Lee, B.-S. 2012. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 721–730. ACM.

Li, C.; Sun, A.; Weng, J.; and He, Q. 2015. Tweet segmentation and its application to named entity recognition. *IEEE TKDE* 27(2):558–570.

Liu, X.; Zhang, S.; Wei, F.; and Zhou, M. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 359–367. Association for Computational Linguistics.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Neubig, G.; Matsubayashi, Y.; Hagiwara, M.; and Murakami, K. 2011. Safety information mining-what can nlp do in a disaster-. In *IJCNLP*, volume 11, 965–973.

Paul, M. J., and Dredze, M. 2011. You are what you tweet: Analyzing twitter for public health. *Icwsm* 20:265–272.

Ritter, A.; Clark, S.; Etzioni, O.; et al. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP-2011*, 1524–1534. Association for Computational Linguistics.

Ritter, A.; Etzioni, O.; Clark, S.; et al. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1104–1112. ACM.

Ritter, A.; Wright, E.; Casey, W.; and Mitchell, T. 2015. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, 896–905. International World Wide Web Conferences Steering Committee.

Sang, E. F., and Veenstra, J. 1999. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 173–179. Association for Computational Linguistics.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Strauss, B.; Toma, B. E.; Ritter, A.; de Marneffe, M.-C.; and Xu, W. 2016. Results of the wnut16 named entity recognition shared task. *WNUT 2016* 138.

Tjong Kim Sang, E. F., and De Meulder, F. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 142–147. Association for Computational Linguistics.

Zhou, G., and Su, J. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 473–480. Association for Computational Linguistics.