Building a Bridge: A Method for Image-Text Sarcasm Detection Without Pretraining on Image-Text Data

Xinyu Wang[1]**, Xiaowen Sun**[1]**, Tan Yang**[2]**, Hongbo Wang**[1]
[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
[2]School of Computer Science (National Pilot Software Engineering School),
Beijing University of Posts and Telecommunications
[1,2]{xinyu.wang, sxw, tyang, hbwang}@bupt.edu.cn

Abstract

Sarcasm detection in social media with text and image is becoming more challenging. Previous works of image-text sarcasm detection were mainly to fuse the summaries of text and image: different sub-models read the text and image respectively to get the summaries, and fuses the summaries. Recently, some multi-modal models based on the architecture of BERT are proposed such as ViLBERT. However, they can only be pretrained on the image-text data. In this paper, we propose an image-text model for sarcasm detection using the pretrained BERT and ResNet without any further pretraining. BERT and ResNet have been pretrained on much larger text or image data than image-text data. We connect the vector spaces of BERT and ResNet to utilize more data. We use the pretrained Multi-Head Attention of BERT to model the text and image. Besides, we propose a 2D-Intra-Attention to extract the relationships between words and images. In experiments, our model outperforms the state-of-the-art model.

1 Introduction

It is becoming popular today for people using text with images to express their emotions and feelings in social media. This makes sarcasm detection more challenging. Sometimes, only when the text and image are read together can one know whether it is sarcasm. For example in Figure 1, which are from the multimodal Twitter dataset (Cai et al., 2019), the images contain the necessary information to determine whether it is a sarcasm.

The previous works about the image-text sarcasm detection (Cai et al., 2019) and also the image-text sentiment analysis (Gaspar and Alexandre, 2019; Huang et al., 2019; Zhao et al., 2019; Kruk et al., 2019) have about two steps: (1) summarizing the image and text; (2) fusing the summaries of the image and text. Although some works try to explore the early fusion, it is still limited. Some details of text and image would be dropped when summarizing.

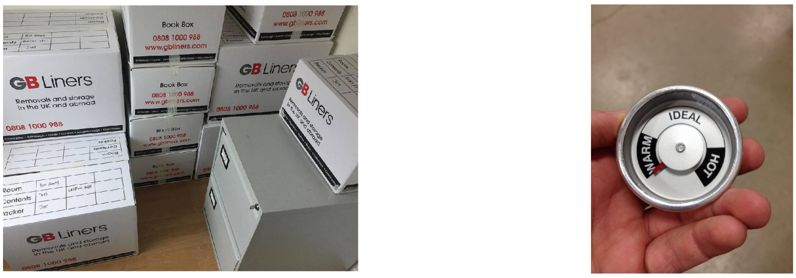
Recently, some multi-modal models based on the architecture of BERT are proposed such as ViLBERT (Lu et al., 2019a,b), LXMERT (Tan and Bansal, 2019), VisualBERT (Li et al., 2019), and B2T2 (Alberti et al., 2019). However, these models are pretrained only on image-text data. In contrast, BERT can be pretrained on much lager text data than image-text data. ResNet can also make use of more image data.

In linear algebra, matrix multiplication can be understood as a kind of vector space transformation. In this paper, we provide a new perspective, the vector space transformation perspective, on this task. We propose a model to connect the text and image, and design a Bridge Layer to build the connection. The low-level and high-level image features are passed into BERT (Devlin et al., 2019) as the embedding of BERT.

We use the pretrained BERT and pretrained ResNet directly without any further pretraining for this task. Any BERT-like models that are based on Transformer (Vaswani et al., 2017) can still be used in our model in the future. Any visual models can be used in our model in the future as well. Besides, our model does not require huge computing resources and time for pretraining.

Based on the idea that sarcasm relies on the semantic relationships and contrasts between words, Tay et al. (2018) uses a softmax and a max function to extract the relationships and contrasts. However, the max function drops some information. In this paper, we propose a method called 2D-Intra-Attention with a 2D-softmax to handle the 2D relationships. Assuming n is the number of inputs, with the 2D-softmax, n^2 relationships are considered every time. In contrast, with the max function (Tay et al., 2018), only n relationships are con-

Proceedings of the First International Workshop on Natural Language Processing Beyond Text, pages 19–29
Online, November 20, 2020. ©2020 Association for Computational Linguistics
http://www.aclweb.org/anthology/W23-20%2d



(a) “packing is so relaxing”

(b) “finally! a thermometer that meets my precision requirements for cooking.”

Figure 1: Examples of image-text sarcasm.

sidered. Besides, we also add the image features into the 2D-Intra-Attention, so the relationships between words and images are considered.

In experiments, our model outperforms the state-of-the-art model. Our main contributions are summarized as follows:

- We connect the text and image: we use image features extracted by pretrained ResNet as the input of the pretrained BERT and utilize the Multi-Head Attention of BERT to model the image features.
- We propose a 2D-softmax to model the 2D relationships considering n^2 relationships and add image features to the 2D-Intra-Attention to extract the relationships and contrasts between words and images.
- We use the pretrained BERT and ResNet directly. Our model can adopt new BERT-like models or visual models in the future and does not require extra computing resources and time for pretraining.

2 Related Work

2.1 Text-Only Sarcasm Detection

Earlier works about sarcasm detection mainly focused on the text. Traditional methods consider and extract various features (Carvalho et al., 2009; Davidov et al., 2010; Veale and Hao, 2010; González-Ibáñez et al., 2011; Reyes et al., 2013; Riloff et al., 2013; Liebrecht et al., 2013; Ptáček et al., 2014; Barbieri et al., 2014; Rajadesingan et al., 2015; Bouazizi and Ohtsuki, 2015; Joshi et al., 2015), including n-grams, punctuations, sentiment, emoticons, incongruity, word frequency, syntactic patterns, etc. Then the deep learning came to sarcasm detection. Many methods based on CNN, LSTM (Hochreiter and Schmidhuber, 1997), and GRU (Cho et al., 2014) were proposed (Bamman and Smith, 2015; Ghosh and Veale, 2016; Zhang et al., 2016; Amir et al., 2016; Poria et al., 2016; Ghosh and Veale, 2017; Peled and Reichart, 2017; Felbo et al., 2017). The deep learning based methods achieved good performance. After the BERT was proposed (Devlin et al., 2019), some works tried to use BERT and achieve better performance (Castro et al., 2019; Badlani et al., 2019; Mao and Liu, 2019). However, these methods were mainly using the semantic features on the top layer extracted by BERT.

2.2 Multimodal Sarcasm Detection

Previous works explored the character and behavior of the reader for multimodal sarcasm detection (Mishra et al., 2016a,b). Some works tried to introduce visual information in sarcasm detection (Schifanella et al., 2016; Cai et al., 2019), but the fusion is mainly used for summaries. Besides sarcasm detection, some works about multimodal sentiment analysis have been done (Wang et al., 2017; Zadeh et al., 2017; Poria et al., 2015; Gu et al., 2018; Gaspar and Alexandre, 2019; Huang et al., 2019; Zhao et al., 2019). Some ideas of multimodal sentiment analysis are similar to multimodal sarcasm detection, so it is also possible to adapt our method to sentiment analysis in the future.

3 Approach

Figure 2 shows the architecture of our model. Our model contains two parts: Image-Text Fusion and 2D-Intra-Attention.

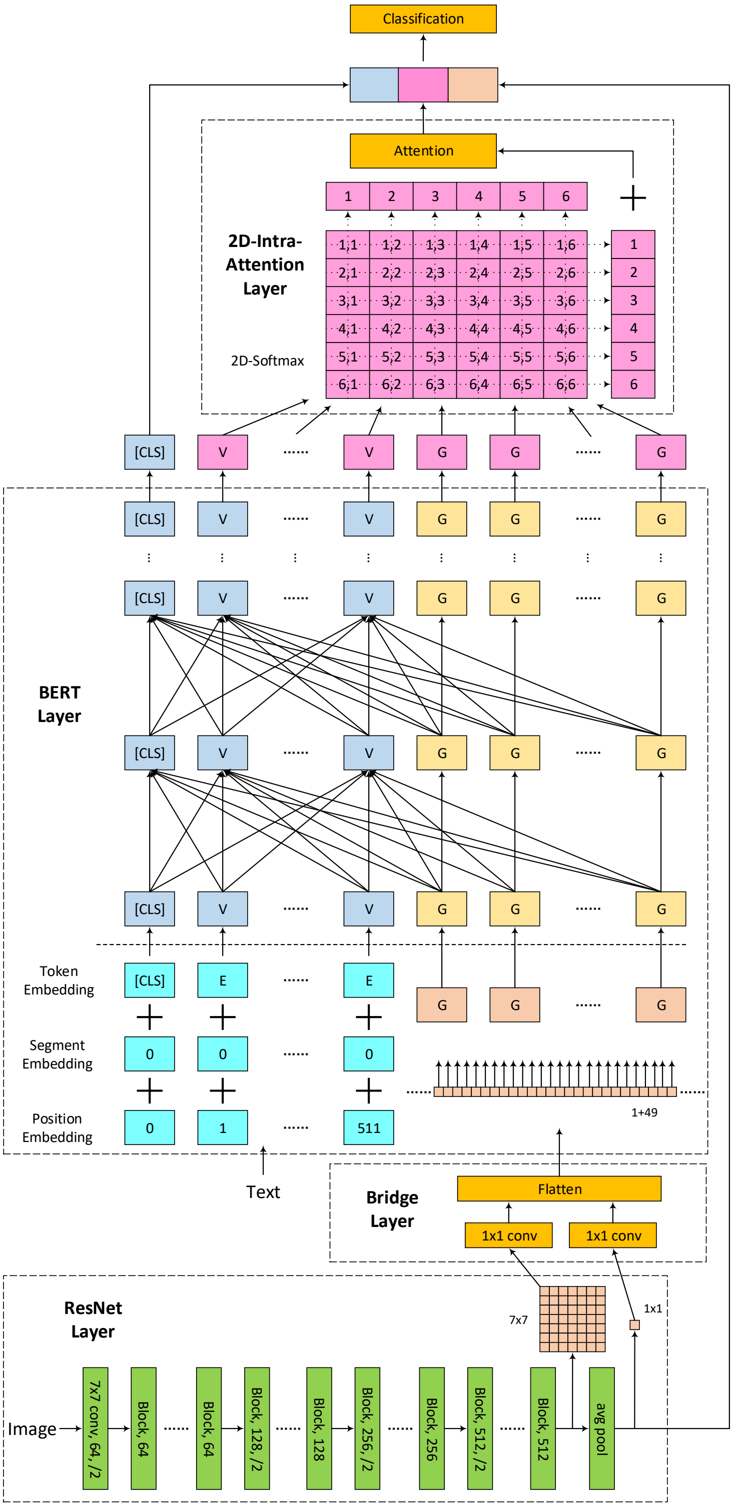


Figure 2: Architecture of our model, where "V" denotes the text and "G" denotes the image.

3.1 Image-Text Fusion

Image-Text Fusion includes BERT Layer, ResNet Layer, and Bridge Layer. In this paper, the term "BERT" refers to the BERT-like models (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019; Sanh et al., 2019), because any one of them and even the new BERT-like models in the future can be used in our model. Similarly, other visual models can also replace the ResNet (He et al., 2016) in our model.

3.1.1 ResNet Layer

ResNet Layer provides the detail and summary of an image. The “Block” of ResNet Layer in Figure 2 means the “building block” in (He et al., 2016), which contains two 3x3 convolution layers, or two 1x1 convolution layers and one 3x3 convolution layer. The image features are the tensor of size 7x7 after the “Block, 512” called feature 7x7 and the tensor of size 1x1 after “avg pool” called feature 1x1 in the following.

The feature 7x7 provides the details of the image. The feature 1x1 provides a summary of the image. In this way, every word in the text can pay attention to the different parts of the image and get more detail information.

3.1.2 Bridge Layer

Bridge Layer is to build the connection between ResNet and BERT. Bridge Layer is very important since ResNet and BERT are pretrained in different spaces. The image features of ResNet cannot be passed into BERT directly. The term “space” here means the vector space or semantic space and is to describe the representations of ResNet and BERT. Bridge Layer maps the image features from ResNet space into BERT space.

Formally, the image features of 7x7 and 1x1 are passed into two 1x1 convolutions respectively, one for feature 7x7 and another for feature 1x1. For the two 1x1 convolutions, the kernel size is 1x1, the stride is 1, the padding length is 0, the number of input channel is the number of the channel of image features such as 1024 or 2048, and the number of output channel is the hidden size of BERT such as 768 or 1024. The function of 1x1 convolutions here equals to fully connected layers. Using 1x1 convolutions and fully connected layers are both feasible when implementing the Bridge Layer. The outputs of the two 1x1 convolutions are flattened and passed into BERT as the embedding of BERT as shown in Figure 2.

The purpose of Bridge Layer is only to build the connection and do transformation instead of learning something. Other methods such as 3x3 conv are suboptimal because it is more likely to overfit than learning something we believe. The task of learning image information should be done by ResNet and the task of integrating image and text should be done by BERT.

3.1.3 BERT Layer

BERT Layer has two parts of inputs. One part is the normal text input. Another part is the image features that have been mapped into BERT space by Bridge Layer. The text is passed through the embedding layer and then the transformer, whereas the image features are passed into the transformer directly without going through the embedding layer.

The text and image features are passed through Multi-Head Attention in different ways. Formally, the attention for words of text is:

$$\begin{aligned} \boldsymbol{v}_i^{(l)} =& \text{MultiHeadAttention}(\\ & \boldsymbol{v}_1^{(l-1)}, \boldsymbol{v}_2^{(l-1)}, \ldots, \boldsymbol{v}_{|\boldsymbol{v}|}^{(l-1)}, \\ & \boldsymbol{g}_1^{(l-1)}, \boldsymbol{g}_2^{(l-1)}, \ldots, \boldsymbol{g}_{|\boldsymbol{g}|}^{(l-1)}) \end{aligned} \tag{1}$$

where $\boldsymbol{v}_i^{(l)}$ denotes the i-th word at the l-th layer and $\boldsymbol{g}_i^{(l)}$ denotes the i-th image feature at the l-th layer; the $|\boldsymbol{v}|$ denotes the number of words and the $|\boldsymbol{g}|$ denotes the number of image features. In this way, every word $\boldsymbol{v}_i$ can pay attention to other words and image features. A word can get detail information from the 7x7 features and summary of the image from the 1x1 features.

However, the attention for image features is:

$$\boldsymbol{g}_i^{(l)} = \text{MultiHeadAttention}(\boldsymbol{g}_i^{(l-1)}) \tag{2}$$

Even though Bridge Layer has mapped image features into BERT space, the mapped image features are still not text, and BERT is never pretrained on the image features. Besides, the CNN of ResNet has a stronger capacity to learn images and the spatial relationships. The relationships between image features have been learned in ResNet. Image feature $\boldsymbol{g}_i$ can only “see” itself. The way of $\boldsymbol{g}_i$ passing through Multi-Head Attention is similar to passing through a fully connected layer. One head of the normal Multi-Head Attention (Vaswani et al., 2017) is:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \tag{3}$$

where Q, K, V are query, key, value. For the attention of $\boldsymbol{g}_i$, the output of softmax will be 1 for $\boldsymbol{g}_i$ and be 0 for others. Therefore, the attention of one vector $\boldsymbol{g}_i$ of one head of the Multi-Head

Attention becomes:

$$
\begin{aligned}
&\text{Attention}_g(\boldsymbol{Q}_{\boldsymbol{g}_i}\boldsymbol{W}_Q, \boldsymbol{K}\boldsymbol{W}_K, \boldsymbol{V}\boldsymbol{W}_V) \\
&= \text{softmax}(\frac{(\boldsymbol{Q}_{\boldsymbol{g}_i}\boldsymbol{W}_Q)(\boldsymbol{K}\boldsymbol{W}_K)^T}{\sqrt{d_k}})(\boldsymbol{V}\boldsymbol{W}_V) \\
&= \boldsymbol{I}_{\boldsymbol{g}_i}(\boldsymbol{V}\boldsymbol{W}_V) \qquad (4) \\
&= (\boldsymbol{I}_{\boldsymbol{g}_i}\boldsymbol{V})\boldsymbol{W}_V \\
&= \boldsymbol{V}_{\boldsymbol{g}_i}\boldsymbol{W}_V \\
&= \boldsymbol{g}_i^T\boldsymbol{W}_V
\end{aligned}
$$

$\boldsymbol{W}_Q \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_K \in \mathbb{R}^{m \times d}$, and $\boldsymbol{W}_V \in \mathbb{R}^{m \times d}$ are parameters for attention calculation as shown in (Vaswani et al., 2017); m denotes the hidden size of BERT and d denotes the hidden size of one head of Multi-Head Attention. $\boldsymbol{I}_{\boldsymbol{g}_i} \in \mathbb{R}^{1 \times n}$ is a vector where only the $(i + |\boldsymbol{v}|)$-th element is 1 and others are 0. n is the total number of words and image features, where $n = |\boldsymbol{v}| + |\boldsymbol{g}|$. $\boldsymbol{Q}_{\boldsymbol{g}_i} \in \mathbb{R}^{1 \times m}$ is the $(i + |\boldsymbol{v}|)$-th vector of $\boldsymbol{Q} \in \mathbb{R}^{n \times m}$. $\boldsymbol{V}_{\boldsymbol{g}_i} \in \mathbb{R}^{1 \times m}$ is the $(i + |\boldsymbol{v}|)$-th vector of $\boldsymbol{V} \in \mathbb{R}^{n \times m}$. The output of one head of attention of $\boldsymbol{g}_i \in \mathbb{R}^m$ is $\boldsymbol{g}_i^T\boldsymbol{W}_V$. The attention for $\boldsymbol{g}_i$ is only to map the $\boldsymbol{g}_i$ from the previous-layer semantic space into the next-layer semantic space with $\boldsymbol{W}_V$.

$\boldsymbol{g}_i$ "seeing" other words and images does not perform well because it will cause noises and overfitting. On the other hand, $\boldsymbol{g}_i$ has to "see" itself because the Multi-Head Attention layer can map the $\boldsymbol{g}_i$ from the previous layer into the next layer. If we use Bridge Layer to map the image features from image space into the next-layer semantic space directly, the model cannot utilize the existed parameters of BERT and need to learn the same information from the beginning.

3.1.4 Space Transformation

To explain the idea behind Image-Text Fusion that matrix multiplication is a kind of vector space transformation, Figure 3 shows the process in space view. Bridge Layer is the connection between the image space and the BERT embedding space. The image features are projected from image space into BERT embedding space by Bridge Layer. The Multi-Head Attention of BERT projects the image features from BERT embedding space into BERT Layer space, then projects the image features from the previous-layer space into next-layer space at every layer.

3.1.5 Some Details

In this section, we will introduce some important details about the Image-Text Fusion.

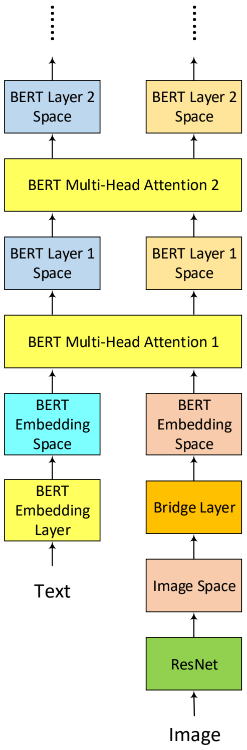


Figure 3: The space transformation view.

Learning Rate Since BERT and ResNet are pretrained models, they usually use a small learning rate. However, Bridge Layer is not pretrained and need to build the connection quickly. The learning rate of Bridge Layer should be greater than BERT and ResNet. In this way, Bridge Layer can learn fast and catch up with BERT and ResNet. This is very important because using the same learning rate will hinder the model from convergence. Empirically, we suggest that the learning rate of Bridge Layer should be at least 10x greater than BERT and ResNet.

Length Limitation The input length of BERT should be less than 512 because only 512 position embeddings are pretrained. However, the image features skip the embedding layer, so introducing image features will not influence the input length limitation of text.

Image Features Sequence The Multi-Head Attention does not consider position by itself, so BERT introduces the position embedding. However, since image features skip the position embedding, the input order of image features is trivial.

3.2 2D-Intra-Attention

We propose a 2D-softmax function to handle the 2D scores. We also add image features to the 2D-Intra-Attention to explore the contrasts and disparities between words and images.

Formally, we define the outputs of both words and images of BERT as $\{\boldsymbol{h}_i\}_{i=1}^{n}$, where n is the total number of words and image features. In other words: $n = |\boldsymbol{v}| + |\boldsymbol{g}|, \boldsymbol{h}_i = \boldsymbol{v}_i, \boldsymbol{h}_{i+|\boldsymbol{v}|} = \boldsymbol{g}_i$.

A pair is defined as:

$$\boldsymbol{h}_{ij} = [\boldsymbol{h}_i; \boldsymbol{h}_j] \tag{5}$$

where $[.;.]$ denotes the concatenation and $\boldsymbol{h}_{ij}$ is a vector. Every $\boldsymbol{h}_{ij}$ is passed through a fully connected layer to get score s_{ij} as:

$$s_{ij} = \boldsymbol{W}_s \boldsymbol{h}_{ij} + b_s \tag{6}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{1\times 2m}$ and $b_s \in \mathbb{R}^1$ are learnable parameters, and m denotes the hidden size of BERT. Then values of s_{ij} where $i = j$ are masked, which is similar to (Tay et al., 2018), and s_{ij} where $i > |\boldsymbol{v}|$ and $j > |\boldsymbol{v}|$ are masked as well.

The 2D-softmax is:

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{p=1}^{n} \sum_{q=1}^{n} e^{s_{pq}}} \tag{7}$$

This 2D-softmax considers s_{ij} from two dimension instead of only one. Then the attention weight $\hat{a}_i$ is calculated as:

$$\hat{a}_i = \frac{1}{2}\sum_{p=1}^{n} a_{ip} + \frac{1}{2}\sum_{q=1}^{n} a_{qi} \tag{8}$$

The a_{ij} in 2D is projected into the $\hat{a}_i$ in 1D. The a_{ip} and a_{qi} are divided by 2 because every a_{ij} is added twice. The final step of 2D-Intra-Attention is:

$$\hat{\boldsymbol{h}} = \sum_i \hat{a}_i (\boldsymbol{W}_a \boldsymbol{h}_i) \tag{9}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{m\times m}$ is a learnable parameter.

With the 2D-softmax, n^2 pairs can be considered. For example, if a word has obvious contrasts with many other words or other parts of images, the attention weight of the word will be high.

3.3 Final Fusion

The concatenation of the `[CLS]` of BERT, the $\hat{\boldsymbol{h}}$ from 2D-Intra-Attention, and the features 1x1 from ResNet are passed through a fully connected layer and a sigmoid function for classification.

4 Experiments

4.1 Training Details

In this section, we will introduce the details and hyper-parameters for training our model.

Pretrained model Pretrained BERT-base-uncased (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019) with 12 layers, and pretrained ResNet50 (He et al., 2016) with 50 layers are used. The ResNet50 we employ is provided by PyTorch (Paszke et al., 2019).

Optimizer The optimizer is Adam (Kingma and Ba, 2014) for BERT with linear schedule and a warm-up ratio of 0.05.

Learning rate The learning rate for RoBERTa and ResNet50 is 1e-5, and for other parameters including Bridge Layer is 1e-3.

Image preprocessing For predicting, we resize the original image making the smaller edge of the image is 224, then crop the image at the center. For training, we implement data augment for images including random crop and randomly change the brightness, contrast and saturation of the image.

Parameters number The number of parameters of our model for experiments is 151M. The learnable parameters are initialized by (He et al., 2015).

GPU & Environment The model is running on a GPU of NVIDIA GeForce RTX 2080 Ti. Due to the limited GPU RAM, we use gradient accumulation for training. The operating system is Ubuntu 18.04. We use PyTorch 1.4.0 (Paszke et al., 2019) and Transformers 2.4.1 (Wolf et al., 2019) to implement our model. We also use mixed precision training with NVIDIA Apex 0.1 (Micikevicius et al., 2017) to accelerate our model.

Running time It takes an average of 343 seconds per epoch. We run the model 10 times and record the best result.

Metrics The metrics for evaluation are F1-score, precision, recall, and accuracy, which are implemented by Scikit-learn (Pedregosa et al., 2011).

4.2 Comparison

The dataset for experiments is the multimodal image-text Twitter dataset (Cai et al., 2019). This data contains image and text as shown in Figure 1.

The description of other compared models are as follows:

	F1-score	Precision	Recall	Accuracy
IARN (Tay et al., 2018)	0.7894	0.7991	0.7799	0.8343
DMAF (Huang et al., 2019)	0.7891	0.7479	0.8352	0.8224
MMHFM (Cai et al., 2019)	0.8018	0.7657	0.8415	0.8344
VisualBERT (Li et al., 2019)	0.7968	0.7666	0.8294	0.8351
LXMERT (Tan and Bansal, 2019)	0.8014	0.7783	0.8259	0.8393
ViLBERT (Lu et al., 2019b)	0.8171	0.7752	0.8637	0.8468
Our model using BERT	0.8235	0.8001	0.8484	0.8564
Our model using RoBERTa	0.8605	0.8295	0.8939	0.8851

Table 1: Comparison of different models.

IARN Multi-dimensional Intra-Attention Recurrent Network (IARN) (Tay et al., 2018). This model proposed a looking in-between method for text-only sarcasm detection.

DMAF Deep Multimodal Attentive Fusion (DMAF) (Huang et al., 2019). We use this image-text sentiment analysis model in comparison since sarcasm detection and sentiment analysis share some similarities.

MMHFM Multi-Modal Hierarchical Fusion Model (MMHFM) (Cai et al., 2019), a fusion model for image-text sarcasm detection.

VisualBERT VisualBERT (Li et al., 2019) is a pretrained visual-text model for vision-and-language tasks, which consists of a stack of Transformer layers.

LXMERT LXMERT (Tan and Bansal, 2019) is a pretrained visual-text model learning the vision-and-language connections based on a large-scale Transformer model.

ViLBERT ViLBERT (Lu et al., 2019a,b) is a pretrained visual-text model which extends the BERT architecture to a multi-modal model. ViLBERT was proposed in (Lu et al., 2019a) at first, then was improved by multi-task training in (Lu et al., 2019b).

Table 1 shows the results. Since ViLBERT is based on BERT (Devlin et al., 2019), we also use BERT (Devlin et al., 2019) in our model to give a fair comparison. Our model with BERT outperforming other models verifies the effectiveness of our model. Moreover, due to the advantage that our model can adopt different pretrained models, if we use RoBERTa (Liu et al., 2019), which was proposed at the time close to ViLBERT, our model can outperform other models significantly. One improvement of RoBERTa comes from using larger data, and our model can make use of the data by adopting RoBERTa.

Our model outperforming ViLBERT and other pretrained visual-text models is mainly because ViLBERT is only pretrained on limited image-text data. In contrast, our model utilizes more unsupervised text data and image data, and only needs to learn a transformation.

4.3 Ablation Studies

In this section, pretrained BERT-base-uncased (Devlin et al., 2019) and pretrained ResNet50 (He et al., 2016) are used. The term "classification" here means the classification layer at the top of Figure 2, which contains a fully connected layer and a sigmoid function. The description of different sets are as follows:

BERT A text-only model that uses the [CLS] of BERT (Devlin et al., 2019) for classification.

BERT + 1D-Intra-Att A text-only model that uses the output of 1D-Intra-Attention (Tay et al., 2018), whose inputs are the outputs of BERT, for classification.

BERT + 2D-Intra-Att A text-only model that uses the output of 2D-Intra-Attention, whose inputs are the outputs of BERT, for classification. 1D-Intra-Attention (Tay et al., 2018) was designed for text-only model, so we also add 2D-Intra-Attention to this text-only model to compare these two attentions.

Concatenation of BERT and ResNet An image-text model that concatenates the [CLS] of BERT whose inputs are text and the output of ResNet for classification. In other words, the model

	F1-score	Precision	Recall	Accuracy
BERT	0.8051	0.7741	0.8388	0.8381
BERT + 1D-Intra-Att	0.8074	0.7766	0.8407	0.8430
BERT + 2D-Intra-Att	0.8088	0.7782	0.8420	0.8439
Concatenation of BERT and ResNet	0.8144	0.7830	0.8485	0.8436
Concatenation of BERT and ResNet + 2D-Intra-Att	0.8168	0.7856	0.8505	0.8476
Image-Text Fusion	0.8214	0.7992	0.8449	0.8541
Image-Text Fusion with Bridge Layer using 3x3 conv	0.8202	0.7978	0.8438	0.8509
Image-Text Fusion + 2D-Intra-Att (our model)	0.8235	0.8001	0.8484	0.8564

Table 2: Ablation Studies.

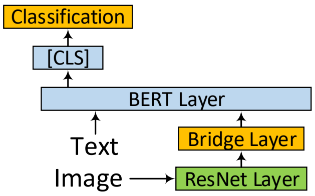


Figure 4: Overview of **Image-Text Fusion** used in ablation studies.

uses image features but does not use them as the inputs for BERT.

Concatenation of BERT and ResNet + 2D-Intra-Att An image-text model that concatenates the [CLS] of BERT whose inputs are text, the output of ResNet, and the output of 2D-Intra-Attention for classification.

Image-Text Fusion The Image-Text Fusion part in this paper. It is important to note that the output of ResNet is used in Final Fusion for classification instead of in Image-Text Fusion. We do not use the output of ResNet for classification here but only the [CLS] as shown in Figure 4, so the image information must go through Bridge Layer and BERT Layer to reach the classification. If Bridge Layer cannot transform image features or BERT Layer cannot integrate text and image, the result should be similar to **BERT** or even worse because image features may cause noises.

Image-Text Fusion with Bridge Layer using 3x3 conv The Image-Text Fusion in this paper that uses the [CLS] of BERT for classification with Bridge Layer using 3x3 conv with padding length 1 instead of 1x1 conv.

Table 2 shows the results. Both **Image-Text Fusion** and **BERT** only use the [CLS] of BERT for classification, and the difference is that BERT Layer of **Image-Text Fusion** has image input. This is proof that BERT Layer and Bridge Layer are effective because image information must go through them to reach the classification. BERT Layer and Bridge Layer must handle image inputs well to give a better result. With image input, the score of **Concatenation of BERT and ResNet** is improved by 0.93% compared with **BERT**, but is still worse than ViLBERT. **Image-Text Fusion** achieves 1.63% improvement compared with **BERT** and outperforms ViLBERT without 2D-Intra-Attention.

The bad result of **Image-Text Fusion with Bridge Layer using 3x3 conv** verifies the effectiveness of using 1x1 conv in Bridge Layer. Our idea for Bridge Layer is just transforming so that the model can utilize pretrained parameters as much as possible instead of learning them from the beginning.

2D-Intra-Attention gives 0.14% improvement for **BERT + 2D-Intra-Att** compared with **BERT + 1D-Intra-Att** and 0.37% improvement compared with **BERT**. Also, 2D-Intra-Attention gives 0.21% improvement for **Image-Text Fusion + 2D-Intra-Att** compared with **Image-Text Fusion**.

5 Conclusion

In this paper, we propose an image-text model for image-text sarcasm detection. We propose a novel way to integrate image and text information. Our model outperforms the state-of-the-art model. Comparing with multi-modal models, our model utilizes more text and image data instead of only the image-text data. Our model can adopt different pretrained language models and visual models directly without any further pretraining.

References

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. 2019. Fusion of detected objects in text for visual question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2131–2140, Hong Kong, China. Association for Computational Linguistics.

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany. Association for Computational Linguistics.

Rohan Badlani, Nishit Asnani, and Manan Rai. 2019. An ensemble of humour, sarcasm, and hate speechfor sentiment classification in online reviews. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 337–345, Hong Kong, China. Association for Computational Linguistics.

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58, Baltimore, Maryland. Association for Computational Linguistics.

Mondher Bouazizi and Tomoaki Ohtsuki. 2015. Sarcasm detection in twitter:" all your products are incredibly amazing!!!"-are they really? In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multimodal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, Florence, Italy. Association for Computational Linguistics.

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio De Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

António Gaspar and Luís A. Alexandre. 2019. A multimodal approach to image sentiment analysis. In *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, pages 302–309, Cham. Springer International Publishing.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

Aniruddha Ghosh and Tony Veale. 2017. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, Copenhagen, Denmark. Association for Computational Linguistics.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, Portland, Oregon, USA. Association for Computational Linguistics.

Yue Gu, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. 2018. Multimodal affective analysis using hierarchical attention strategy with word-level alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2235, Melbourne, Australia. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Feiran Huang, Xiaoming Zhang, Zhonghua Zhao, Jie Xu, and Zhoujun Li. 2019. Image–text sentiment analysis via deep multimodal attentive fusion. *Knowledge-Based Systems*, 167:26–37.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, and Ajay Divakaran. 2019. Integrating text and image: Determining multimodal document intent in instagram posts. *arXiv preprint arXiv:1904.09073*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

CC Liebrecht, FA Kunneman, and APJ van Den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019a. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2019b. 12-in-1: Multi-task vision and language representation learning. *arXiv preprint arXiv:1912.02315*.

Jihang Mao and Wanli Liu. 2019. A bert-based approach for automatic humor detection and scoring. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS, Bilbao, Spain (9 2019)*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016a. Predicting readers' sarcasm understandability by modeling gaze behavior. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016b. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Berlin, Germany. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Lotem Peled and Roi Reichart. 2017. Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1690–1700, Vancouver, Canada. Association for Computational Linguistics.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Meth-*

ods in Natural Language Processing, pages 2539–2544, Lisbon, Portugal. Association for Computational Linguistics.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1601–1612, Osaka, Japan. The COLING 2016 Organizing Committee.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on Czech and English twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Rossano Schifanella, Paloma de Juan, Joel Tetreault, and Liangliang Cao. 2016. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1136–1145.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020, Melbourne, Australia. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Tony Veale and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. In *ECAI*, volume 215, pages 765–770.

H. Wang, A. Meghawat, L. Morency, and E. P. Xing. 2017. Select-additive learning: Improving generalization in multimodal sentiment analysis. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 949–954.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan. The COLING 2016 Organizing Committee.

Ziyuan Zhao, Huiying Zhu, Zehao Xue, Zhao Liu, Jing Tian, Matthew Chin Heng Chua, and Maofu Liu. 2019. An image-text consistency driven multimodal sentiment analysis approach for social media. *Information Processing & Management*, 56(6):102097.