Sarcasm Detection with Commonsense Knowledge

Jiangnan Li, Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang

***Abstract*—Sarcasm is commonly used in today's social media platforms such as Twitter and Reddit. Sarcasm detection is necessary for analysing people's real sentiments as people usually use sarcasm to express a flipped emotion against the literal meaning. However, the current works neglect the fact that commonsense knowledge is crucial for sarcasm recognition. In this paper, we propose a novel architecture in deep learning for sarcasm detection by integrating commonsense knowledge. To be specific, we apply the pre-trained COMET model to generate relevant commonsense knowledge. Besides, we compare two kinds of knowledge selection strategies to investigate how commonsense knowledge influences performance. Finally, a knowledge-text integration module is designed to model both text and knowledge. The experimental results demonstrate our model's effectiveness on three datasets, including two Twitter datasets and a Reddit dataset.**

***Index Terms*—Commonsense knowledge, sarcasm detection, deep learning, knowledge selection, knowledge-text integration.**

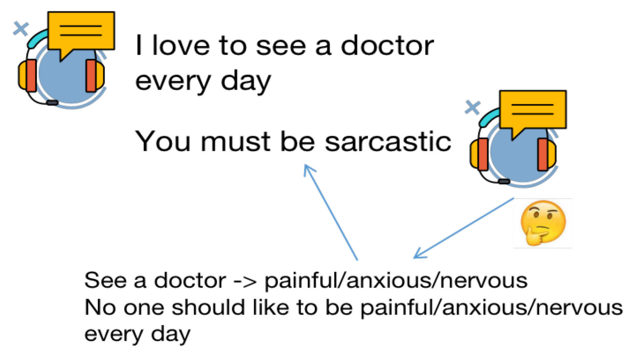

Fig. 1. Examples of commonsense knowledge aiding sarcasm recognition.

I. Introduction

SARCASM is a form of figurative language, defined as "the use of irony to mock or convey contempt"[1], which is ubiquitous in social media platforms such as Twitter and Reddit. People tend to use sarcasm to express the opposite of superficial meaning [1]. The utterance "I love to see a doctor every day" expresses sarcastic meaning. It shows a negative sentiment towards the situation of "see a doctor every day", even the utterance contains positive sentiment words such as "like". Thus, failure to detect sarcasm may degrade the performance of the applications requiring people's real minds, such as sentiment analysis and opinion mining [2]. Noticeably, many approaches have been proposed for sarcasm detection, including both text-based methods [3]–[6] and multimodal-based methods [7]–[10].

State-of-the-art sarcasm detection systems mainly rely on neural networks and attention mechanism. Poria *et al.*, [11] apply pre-trained CNN models to identify sarcasm. Both Tay *et al.*, [5] and Xiong *et al.*, [6] design attention-based models to capture the contradiction information between word pairs and use RNN based models to obtain compositional information. The two kinds of information are then used for classification. To boost the performance, several other works introduce multi-modality information for this task [8]–[10]. However, the existing works cannot deal with the sarcastic utterance that requires commonsense knowledge to understand.

Commonsense knowledge is information that humans typically have that helps them make sense of everyday situations [12]. Such information allows people to connect pieces of knowledge to reach a new conclusion [13]. Veale and Hao [14] point out that sarcasm understanding often relies on the commonsense knowledge of the world outside the text. Considering the example in Fig. 1, the listener can not recognize the speaker's sarcastic intention unless the listener realizes the phrase "see a doctor" is associated with commonsense knowledge like "painful", "anxious", "nervous" and so on. As a result, commonsense knowledge is beneficial to sarcasm understanding. However, the existing systems fail to involve commonsense knowledge for sarcasm detection.

In this paper, we propose a novel architecture for sarcasm detection by integrating commonsense knowledge to overcome the weaknesses mentioned above of previous works[5], [6], [15]. Our model consists of three main modules, including an encoding module, a commonsense reasoning module, and a knowledge-text integration module. In the encoding module, we use the pre-trained BERT [15] model to encode both input and knowledge. The commonsense reasoning module has two parts: they are knowledge generation and knowledge selection. To be specific, we apply the COMET [16] model to generate commonsense knowledge candidates in the knowledge generation part. COMET is an adaption framework for constructing commonsense knowledge. Given a subject and a relation, the COMET model generates knowledge candidates relevant to the

Manuscript received April 21, 2021; revised August 16, 2021; accepted September 23, 2021. Date of publication October 15, 2021; date of current version October 30, 2021. This work was supported by the National Natural Science Foundation of China under Grants 61976207 and 61906187. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zoraida Callejas. *(Jiangnan Li and Hongliang Pan contributed equally to this work.) (Corresponding author: Zheng Lin; Peng Fu.)*

Jiangnan Li, Hongliang Pan, and Zheng Lin are with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100089, China (e-mail: lijiangnan@iie.ac.cn; panhongliang@iie.ac.cn; linzheng@iie.ac.cn).

Peng Fu and Weiping Wang are with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100089, China (e-mail: fupeng@iie.ac.cn; wangweiping@iie.ac.cn).

Digital Object Identifier 10.1109/TASLP.2021.3120601

[1]Defined by https://www.vocabulary.com/dictionary/sarcasm

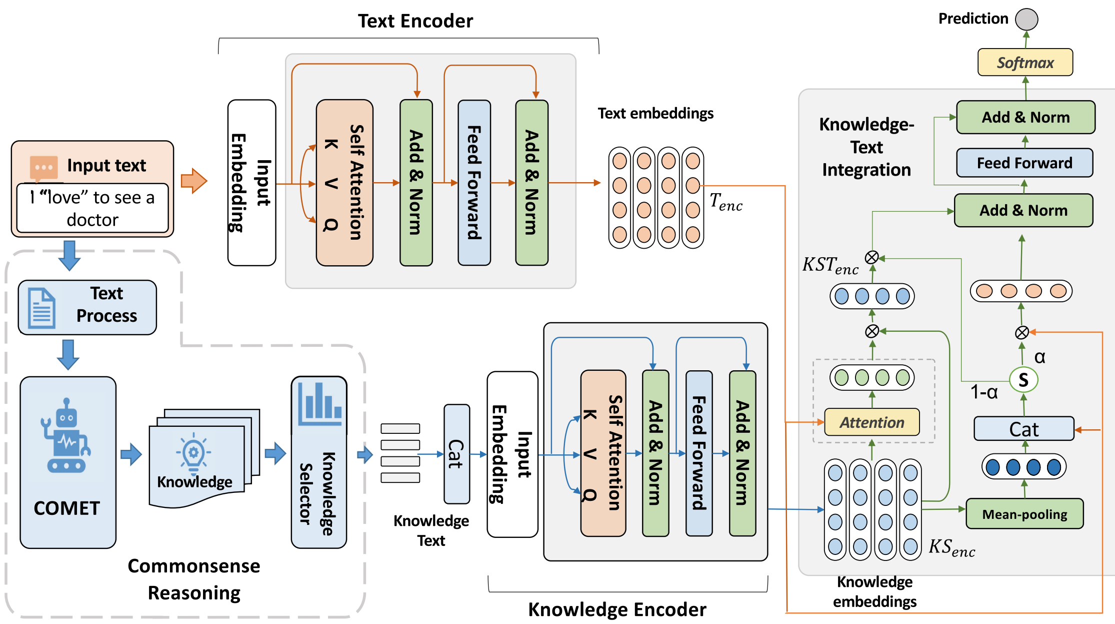

Fig. 2. Overview of our proposed model. Two separate pre-trained BERT models encode input text and knowledge text. The commonsense reasoning module is used to generate and select knowledge. The knowledge-text integration module learns the interaction of text embeddings and knowledge embeddings, and its output is used for prediction.

subject and relation with the highest probabilities. In the knowledge selection part, in order to investigate how commonsense knowledge influences the model's performance, we compare the explicit and implicit knowledge selection strategies, which are based on sentiment polarity and attention mechanism separately. The knowledge-text integration module is designed to combine the information from both the text and the knowledge. In the knowledge-text integration module, we design a gate mechanism to let the model dynamically learn how much information should be considered from text and from knowledge. After that, a residual structure is applied to fuse the information from both text and knowledge.

The main contributions of our work can be summaries as follows:

(a). We propose a novel architecture which involves commonsense knowledge for sarcasm detection, aiming to address the issue that existing sarcasm detection approaches fail to involve commonsense knowledge to identify sarcasm.

(b). We compare explicit and implicit knowledge selection strategies to investigate how commonsense knowledge influences the performance. Besides, we also design a knowledge-text integration module to combine the information from both text and knowledge.

(c). We conduct extensive experiments to prove our model's effectiveness and our model achieves the best performance on two Twitter datasets and a Reddit dataset.

II. Method

In this section, we first define the sarcasm detection task. Then, we describe the architecture of our proposed model in detail. Fig. 2 gives an overview of our model.

A. Task Definition

Sarcasm detection aims to identify if an utterance has sarcastic meanings. Formally, given a sequence X containing n words, $X = \{x_1, x_2, \ldots, x_n\}$, our model is supposed to correctly classify the given text into sarcasm or non-sarcasm categories.

B. Model Architecture

Our model consists of three modules: the encoding module, the commonsense reasoning module, and the knowledge-text integration module.

1) Encoding Module: The pre-trained language models [15], [17], [18] have shown the superiority in many natural language processing tasks, such as question answering and language inference, merely with minor task-specific architecture modifications. Unlike traditional fixed word embeddings, the word's representations vary according to their contexts in pre-trained language models, thus resulting in outstanding representations of text.

BERT is a pre-trained language model proposed by Devlin *et al.*, [15], which consists of multiple layers of bi-directional transformer encoders [19]. In our work, we apply a pre-trained BERT Base model (with 12 transformer blocks, feed-forward networks with 768 hidden units and 12 attention heads) to encode both the input text and the knowledge text. Specifically, given a sequence X containing s words, $X = \{x_1, x_2, \ldots, x_s\}$, where $x_i \in \mathbb{R}^d$ is the sum-up of word, segment, and position embeddings, s is the maximum length of the sequence, d is the emebdding size. The encoded sequence representations can be depicted as $X_{enc} \in \mathbb{R}^{d*s}$, which is the output of the last layer of BERT encoders and d is the hidden size of BERT. For the input text sequence $T = \{t_1, t_2, \ldots, t_m\}$, BERT can produce the encoded text representations $T_{enc} \in \mathbb{R}^{d*m}$.

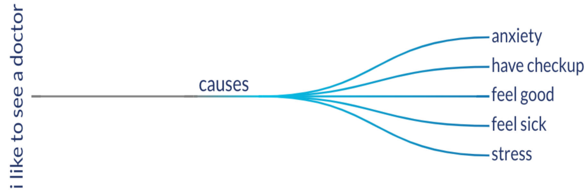

Fig. 3. The figure illustrates some commonsense knowledge generated by COMET.

2) Commonsense Reasoning Module: The Commonsense Reasoning Module contains two parts: Knowledge Generation and Knowledge Selection.

Knowledge Generation: We apply the COMET [16] model in our work to generate commonsense knowledge. COMET is an adaption framework for constructing commonsense knowledge based on pre-trained GPT [18] model. Bosselut el at. [16] use commonsense knowledge tuples (from ConeptNet [20] or ATOMIC [21] datasets) to fine-tune the GPT model and COMET learns to produce corresponding commonsense knowledge after training. Fig. 3 gives a visualization of some commonsense knowledge candidates generated by COMET. We borrow the idea from the reference [22] to generate commonsense knowledge, to be specific, before feeding our input text into COMET, we conduct a series of text processing steps, including punctuation and stop words elimination, token lowercase, and lemmatization. We further adopt the same configuration from the reference [22] using beam search, whose size is set as 5, to generate commonsense knowledge candidates. Our work uses the COMET model fine-tuned by ConceptNet tuples (subject-realtion-object), and we only leverage the causes relation to generate knowledge candidates like the reference [22].

Knowledge Selection: Two kinds of knowledge selection policies are considered in our work. They are the sentiment polarity-based explicit knowledge selection policy and attention-based implicit knowledge selection policy. The attention mechanism is commonly used to compute the similarity, the attention value indicates an item's importance. Then, the items can be chosen according to their importance. Obviously, sarcasm is always coupled with the sentiment. As a result, sentiment polarity is also considered in our work to select knowledge candidates. Given the knowledge set $K = \{k_1, k_2, \ldots, k_b\}$, where each k_i in K is a knowledge candidate. b is the number of knowledge candidates. The sentiment score of a token t in k_i is $sent(t)$, which is calculated using SentiWordNet[2]. Specifically, we sum the sentiment score of a word under different meanings.

The explicit knowledge selection policy depends on the sentiment polarity of knowledge candidates. We propose **majority, minority, and contrast sentiment-based methods** to select knowledge candidates.

- Algorithm 1 describes the majority sentiment-based knowledge selection method. We first compute the overall sentiment polarity of all the knowledge candidates, and we then reserve those knowledge candidates which have the same sentiment polarity with the overall sentiment polarity and discard others.
- Algorithm 2 describes the minority sentiment-based knowledge selection method. The minority sentiment-based method has a similar procedure. We only keep those knowledge candidates whose sentiment polarity contrasts the overall sentiment polarity.
- Algorithm 3 describes the contrast sentiment-based knowledge selection method. As for the contrast sentiment-based method, we reserve those knowledge candidates whose sentiment polarity contrasts the input text rather than the overall sentiment of knowledge candidates.

The knowledge candidates in selected knowledge set KS are then concatenated as the input to knowledge encoder to obtain the encoded knowledge representation $KS_{Eenc} \in \mathbb{R}^{d*n}$ under

[2]https://wordnet.princeton.edu/

Algorithm 1: Majority Sentiment-Based Knowledge Selection.

Input: Knowledge set K which has $\{k_1 ... k_b\}$
Output: Selected knowledge set KS

$sent_{score} = 0$
for $i = 0$ to b **do**
  for all $token$ such that $token \in k_i$ **do**
    $sent_{score} = sent_{score} + sent(token)$
  end for
end for
if $sent_{score} > 0$ **then**
  for all k_i such that $sent(k_i) > 0$ **do**
    put k_i in KS
  end for
else
  for all k_i such that $sent(k_i) < 0$ **do**
    put k_i in KS
  end for
end if

Algorithm 2: Minority Sentiment-Based Knowledge Selection.

Input: Knowledge set K which has $\{k_1 ... k_b\}$
Output: Selected knowledge set KS

$sent_{score} = 0$
for $i = 0$ to b **do**
  for all $token$ such that $token \in k_i$ **do**
    $sent_{score} = sent_{score} + sent(token)$
  end for
end for
if $sent_{score} > 0$ **then**
  for all k_i such that $sent(k_i) < 0$ **do**
    put k_i in KS
  end for
else
  for all k_i such that $sent(k_i) > 0$ **do**
    put k_i in KS
  end for
end if

Algorithm 3: Contrast Sentiment-Based Knowledge Selection.

Input: Knowledge set K which has $\{k_1...k_b\}$
Input text sequence X $= \{x_1...x_m\}$
Output: Selected knowledge set KS
$sent_{score} = 0$
for all $token$ such that $token \in X$ **do**
  $sent_{score} = sent_{score} + sent(token)$
end for
if $sent_{score} > 0$ **then**
  for all k_i such that $sent(k_i) < 0$ **do**
    put k_i in KS
  end for
else
  for all k_i such that $sent(k_i) > 0$ **do**
    put k_i in KS
  end for
end if

explicit knowledge selection, n is a hyper-parameter donating the max length of knowledge sequence.

Our implicit knowledge selection policy is based on the attention mechanism. Specifically, given the [CLS] token's representation $K_{CLS} \in \mathbb{R}^d$ and knowledge representations $K_{enc} \in \mathbb{R}^{d*n}$, The implicit selected knowledge representation is computed as follows:

$$KS_{Ienc} = softmax(K_{CLS}^T K_{enc}) K_{enc}^T \tag{1}$$

where $KS_{Ienc} \in \mathbb{R}^d$ is the knowledge representations after implicit knowledge selection.

3) Knowledge-text Integration Module: In this section, we design a knowledge-text integration module for learning the knowledge-text interaction. A vector containing the combined information is used for prediction.

To obtain the text-aware knowledge representations, we first conduct a text-guided attention to distribute attention weights to the selected knowledge tokens. The new representations of knowledge KST_{enc} can be computed as:

$$KST_{enc} = softmax(T_{enc}^T KS_{E/Ienc}) KS_{E/Ienc}^T \tag{2}$$

where $T_{enc} \in \mathbb{R}^{d*m}$ are the encoded representations of text and $KS_{Eenc/}KS_{Ienc} \in \mathbb{R}^{d*n}/\mathbb{R}^{d*1}$ are the representations of selected knowledge. $KST_{enc} \in \mathbb{R}^{m*d}$ is the text-aware knowledge representations.

In some circumstances, text and knowledge might contribute differently to the prediction. Thus we introduce a gate mechanism to let the model dynamically learn how much information should be considered from the text and how much information from the knowledge. The gate mechanism works as follows. We first perform a mean-pooling operation on $KS_{enc} \in \mathbb{R}^{d*n}$ to obtain a single vector $KS_{mean} \in \mathbb{R}^d$ representing the knowledge. The output α of the gate is performed as:

$$\alpha = sigmoid([T_{enc}^T \oplus KS_{mean}] W_g) \tag{3}$$

where $\alpha \in \mathbb{R}^{m*1}$ is a weight vector and $W_g \in \mathbb{R}^{2d*1}$ is a trainable parameter. $\oplus$ is the concatenation operation that concatenates every representation in T_{enc}^T with KS_{mean}. We reconstruct the knowledge representations and text representations with the weigth vector α, showing as $T_{enc}\alpha$ and $KST_{enc}(1-\alpha)$.

A residual structure is involved here to combine the information from both text and knowledge as:

$$Z = LN(T_{enc}\alpha + W_b KST_{enc}(1-\alpha)) \tag{4}$$

where LN is the layer normalization operation proposed by Ba *et al.*, [23] and $W_b \in \mathbb{R}^{d*d}$ is a trainable patameter.

$$\hat{Z} = LN(Z + MLP(Z)) \tag{5}$$

After that, a feed-forward network (a.k.a MLP) and another residual connection are employed on Z to obtain the vector $\hat{Z} \in \mathbb{R}^d$, which incorporats the combined information.

C. Prediction

The prediction part consists of a linear layer and a softmax function. The linear layer is used to reduce the dimension and the softmax function distributes probabilities to each category. Our model will classify the given text into the category with the highest probability. This procedure can be described as:

$$\hat{y} = Softmax(\hat{Z} W_c + b) \tag{6}$$

where $W_c \in \mathbb{R}^{d*2}$ is learnable parameter training along with the model. $\hat{y}$ is the classification result of our model.

D. Training objectives

Cross-entropy loss function is used in our work for optimizing the model.

$$J = -\sum_{i=1}^{N} [y_i log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)] + \lambda R \tag{7}$$

where J is the cost function. $\hat{y}_i$ is the prediction result of our model for sample i, and y_i is the true label for sample i. N is the size of training data. R is the standard L2 regularization and λ is the weight of R.

III. Experiment

In this section, we first describe the datasets, experimental settings, and baseline models. Then we compare the performance of our model and other baseline models and conduct a series of ablative experiments to verify the effectiveness of each module in our model. Finally, we perform a more in-depth analysis, which includes a parameter study, a case study and an error analysis.

A. Dataset

We evaluate our model on three datasets, including two Twitter datasets proposed by Ghosh *et al.*, [24] and Ptácek *et al.*, [25] and a Reddit dataset [26] with subreddits **politics**. We denote the three datasets as Twitter (Ghosh), Twitter (Ptácek), and SARC-Pol. It is worth noticing that the Twitter dataset proposed by Riloff *et al.*, [3] and the IAC datasets [27] are

TABLE I
DATASETS DESCRIPTION.

Dataset	Train	Dev	Test	Total
Tweets(Ghosh)	46070	5118	3742	54930
Tweets(Ptácek)	8497	1062	1062	10621
Reddit-pol	20842	2605	2605	26052

also commonly used in sarcasm detection. However, we find that only around one-third of the Riloff Twitter dataset (less than 1000 samples) are available and the COMET model is not suitable to generate commonsense knowledge for long text in the IAC dataset. Consequently, we discard these two datasets. In our work, each sample consists of a sequence of text with associated commonsense knowledge generated by COMET. Detailed statistics are summarized in Table I.

B. Baseline Models

We compare our model with the following baseline models.

- NBOW: The Neural bag-of-words model takes an average of word embedding vectors as sentence representation, which is passed to a logistic regression classifier.
- TextCNN: It is proposed by Kim [28], which is a deep learning model based on Convolutional neural network for addressing text classification tasks.
- Bi-LSTM: LSTM is a Long Short-Term Memory Network. We adopt a Bi-direction LSTM in our work as a baseline.
- SIARN: SIARN is proposed by Tay *et al.*, [5]. They first employ inner-attention for sarcasm detection as previous sequential models such as RNNs, which cannot capture the word pairs interaction.
- SMSD-BiLSTM: Following the work of Tay *et al.*, [5], Xiong *et al.*, [6] propose a self-matching network to capture sentence incongruity information by exploring word-to-word interaction. They further use an additional bi-directional LSTM encoder to cultivate sentence's compositional information.
- BERT: BERT is a pre-trained language model proposed by Devlin *et al.*, [15], which achieves outstanding results in many NLP tasks. We take it as a baseline to investigate whether the performance gain comes from BERT or our proposed method.
- miroblog: miroblog [29] is the best model in Figlang 2020 workshop of sarcasm detection [30]. Since miroblog focuses on modeling the conversational context of a response to be predicted, which does not accord with our case[3], we take off modules related to the conversational context. Therefore, miroblog that we consider is a BERT encoder with two-layer LSTMs and a NeXtVALD layer [31] stacked above.
- Know-BERT: It is the abbreviation of Knowledge-BERT model, which simply concatenates the text representation T_{enc} and knowledge representation KS_{enc}, followed by a linear layer for classification.

[3]Our method follows the previous works [5], [6] to deal with the conventional sarcasm detection, which contains no conversational context.

C. Experimental Settings

Our model is implemented in PyTorch [32], running on a NVIDIA TITAN RTX GPU. We implement the pre-trained BERT model using the **transformers** toolkit, which is released by Hugging Face[4]. We adopt Adam [33] as our optimizer and set the learning rate as 5e-5 with a warmup rate of 0.1. The batch size is fixed to 32 for training. We set the maximum length of the text to 40 and that of the knowledge to 20 so that most samples can be set to a proper length[5]. Our model is fine-tuned for 8 epochs, and the model with the best performance on the validation set is saved. Our codes and resources can be found at https://github.com/LeqsNaN/SarDeCK.

D. Experimental Results

In this section, we first compare our model with the baseline models on some standard metrics, including precision, recall, and F1 score. Then we analyse the performance of our model under different knowledge selection strategies. Precision describes how effective the model is in applying a label for a given category (high precision means few false positives samples). Recall describes how effective the model is in finding the relevant examples of a category (high recall means few false negatives samples). F1 score indicates a trade-off between the precision and the recall.

Table II reports the performance comparison of all models on three datasets. We observe that our method achieves the best performance across all datasets. Specifically, on the Twitter datasets from Ghosh[24] and Ptácek [25], our model achieves 3.56% and 1.22% improvement in terms of F1 compared with the fine-tuned BERT model. On the Reddit (SARC-pol) dataset, the performance gain of our model is around 3.18%. Further compared with miroblog that also introduces no commonsense knowledge, our method outperforms it by 3.22%, 1.33%, and 1.83% on Twitter (Ghosh), Twitter (Ptácek), and SARC-pol datsets respectively. Consequently, integrating commonsense knowledge information contributes to sarcasm detection, which complies with our motivation.

It is worth noticing that the pre-trained BERT based model outperforms the traditional deep learning models in most circumstances. In addition, miroblog only show relatively obvious improvement against BERT on SARC-pol. We owe these results to the outstanding text representations of BERT. Know-BERT model performs better than BERT on the Twitter (Ghosh) and SARC-pol dataset, but not on the Twitter dataset (Ptácek). It shows that merely concatenating text information and knowledge information might impede rather than enhance the performance.

Table III gives the comparison of models' performance under different knowledge selection strategies. $Model_{Maj}$, $Model_{Min}$, and $Model_{Con}$ represent the model under **majority, minority, and contrast sentiment-based** knowledge selection strategies respectively. $Model_{Attn}$ is the model with

[4]https://huggingface.co/transformers/

[5]If exceeding the max length, the text or knowledge will be truncated to the max length.

TABLE II
EXPERIMENTAL RESULTS ON THE DATASETS. THE BEST RESULTS ARE IN BOLD.

	Twitter(Ghosh)			Twitter(Ptácek)			SARC-pol		
Method	P	R	F1	P	R	F1	P	R	F1
NBOW	0.7303	0.7444	0.7281	0.8039	0.8027	0.8033	0.6983	0.6933	0.6950
TextCNN	0.7488	0.7635	0.7490	0.7993	0.7965	0.7978	0.7180	0.7183	0.7181
Bi-LSTM	0.7752	0.7885	0.7784	0.7994	0.8029	0.8009	0.7062	0.7100	0.7073
SIARN	0.7464	0.7601	0.7480	0.7897	0.7924	0.7909	0.7018	0.7020	0.7019
SMSD	0.7508	0.7662	0.7447	0.7966	0.7979	0.7972	0.7326	0.7331	0.7328
BERT	0.8178	0.8337	0.8217	0.8399	0.8465	0.8424	0.7339	0.7266	0.7292
miroblog	0.8211	0.8344	0.8251	0.8395	0.8441	0.8413	0.7525	0.7402	0.7427
Know-BERT	0.8437	0.8538	0.8477	0.8433	0.8361	0.8392	0.7521	0.7566	0.7536
Method(ours)	**0.8552**	**0.8598**	**0.8573**	**0.8516**	**0.8605**	**0.8546**	**0.7610**	**0.7610**	**0.7610**

TABLE III
EXPERIMENTAL RESULTS OF KNOWLEDGE SELECTION STRATEGY. THE BEST RESULTS ARE IN BOLD.

	Twitter(Ghosh)			Twitter(Ptácek)			SARC-pol		
Method	P	R	F1	P	R	F1	P	R	F1
BERT	0.8178	0.8337	0.8217	0.8399	0.8465	0.8424	0.7339	0.7266	0.7292
Model$_{Maj}$	0.8249	0.8376	0.8292	**0.8516**	**0.8605**	**0.8546**	0.7532	0.7519	0.7525
Model$_{Min}$	**0.8552**	**0.8598**	**0.8573**	0.8361	0.8426	0.8386	0.7555	0.7554	0.7555
Model$_{Con}$	0.8349	0.8471	0.8392	0.8393	0.8413	0.8402	**0.7610**	**0.7609**	**0.7610**
Model$_{Attn}$	0.8341	0.8467	0.8385	0.8404	0.8417	0.8410	0.7570	0.7571	0.7571
Model$_{All}$	0.8344	0.8437	0.8381	0.8352	0.8385	0.8367	0.7520	0.7526	0.7523

attention-based knowledge selection and $Model_{All}$ uses all the knowledge candidates without any selection. It shows that knowledge selection improves model performance across all three datasets. We also notice that explicit knowledge selection strategies work better than attention-based knowledge selection. As sarcasm is often related to sentiment, the explicit knowledge selection strategies involve sentiment information, thus achieving better results. However, the best results of different datasets are achieved by different selection strategies, which means that the best knowledge selection strategy might rely on data distributions. For instance, the topic of all samples in the SARC-pol dataset is about politics, whereas there are no fixed topics in the other two Twitter datasets.

E. Ablation Study

We give out an ablation study in this part. Firstly, we only feed the knowledge tokens to a pre-trained BERT to study how merely use knowledge information performs. To investigate the effectiveness of the gate mechanism and the residual structure in the knowledge-text integration module, we first remove the gate and directly pass T_{enc} and KST_{enc} to the residual part. After that, we exclude the residual structure from our model and merely concatenate αT_{enc} and $(1-\alpha)KST_{enc}$, following a classification layer. Besides, we further implement two other integration methods, which are concatenation and element-wise addition, to investigate how our proposed module performs.

Table IV gives the results of ablative experiments. It shows that only using knowledge information does not perform satisfactorily, demonstrating that commonsense knowledge only plays a supporting role. The gate's absence leads to decreased results, proving that it is meaningful to dynamically distribute weights to text and knowledge. The model without residual structure also impedes the performance. Finally, our model's integration way performs better than concatenation and element-wise addition, which indicates our module's effectiveness.

TABLE IV
EXPERIMENTAL RESULTS ON ABLATION STUDY. THE BEST RESULTS ARE IN BOLD. WE ABBREVIATE TWITTER (GHOSH) AS TWITTER (G) AND TWITTER (PTÁCEK) AS TWITTER (P).

	Twitter(G)	Twitter(P)	SARC-pol
Method	F1 score		
Model(know)	0.6268	0.5986	0.3833
Model(ours)	**0.8573**	**0.8546**	**0.7610**
Model(-gate)	0.8127	0.8292	0.7574
Model(-res)	0.8445	0.8365	0.7523
Model(+concat)	0.8373	0.8387	0.7596
Model(+add)	0.8354	0.8296	0.7399

F. Model Analysis

The impact of the number of knowledge candidates: The number of knowledge candidates l_m is an important hyper-parameter to control how much knowledge information is involved to enhance sarcasm detection. In this part, we study the effect of the number of knowledge candidates on our knowledge selection. The performance of our model is measured with l_m ranging from 1 to 5. We can see from fig 4 that the F1 score roughly has an increasing trend as the number of knowledge candidate increases, though it fluctuates on the two Twitter datasets. Across all three

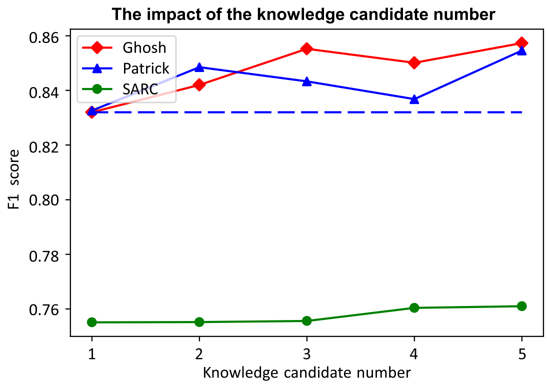

Fig. 4. The performance curves with a variety of l_m from 1 to 5. The performance on all datasets shows the increasing trend as l_m increases. The dotted line in blue is the F1 score when $l_m = 1$ on Twitter (Ptácek). Although the performance on Twitter (Ptácek) fluctuates when $l_m = 3, 4$, the F1 score is still above the dotted line.

datasets, the model performs worst when l_m is 1, and it performs best when involving all the available knowledge candidates. As a result, under the best knowledge selection strategy, an increasing number of knowledge candidates can enhance the performance. This also demonstrates that our knowledge selection strategy can effectively extract useful information from proper number of knowledge candidates.

Case study: In this section, we demonstrate some examples correctly classified by our model but wrongly classified by BERT. We also extract the attention maps to see how commonsense knowledge help improve the performance. Fig. 5 gives an attention visualization. We notice that our model attends to the tokens which form a contradiction with the knowledge. In the first example, our model attends to the token "love", which contrasts to the token "headache" in knowledge. The second and the third instances also present similar patterns. Consequently, by introducing commonsense knowledge, our model finds the contradiction between the sentence and the knowledge, which helps to obtain the correct result.

Error analysis: We also perform an analysis of the wrongly predicted samples. The bellowing shows some examples that our model fails to label it right.

- "Good job, Rosenthal."
- "Dad sounded excited i got a new job."
- "Answering the phone to be screamed at come 7:00 is just what I was looking forward to."

We divide the wrongly classified examples into three categories. The first one is those sarcastic samples that require background knowledge rather than commonsense knowledge to be recognized (refers to the first example). We treat the distinction between intended and perceived sarcasm as the second category. To be specific, the second example is labelled as sarcasm by its author from an intended perspective. However, the audience might take it as not sarcasm from a perceived perspective. Finally, in order to understand the sarcasm in the third instance, the classifier needs to know that the sentence segment "Answering the phone to be screamed at come 7:00" implies a negative sentiment. However, for this instance, the COMET model only generates some superficial commonsense knowledge like "pick up receiver". As a result, our model fails to predict it right.

For the first category, if the conversation context of the sentence can be provided, the detection can be more precise. For example, if the former context "I treat my students well but they do not reciprocate me" is provided to "Good job, Rosenthal", the prediction of "Good job, Rosenthal" is obviously sarcastic. This type of detection that involves conversational context for a sentence is called conversational sarcasm detection [30]. As for our method ignoring such observation, we will extend our method to satisfying both conversational and conventional sarcasm detection in the future work. In this way, our method can better settle the error cases mentioned above.

IV. Related Work

Sarcasm is an extensively studied linguistic phenomenon by linguistic scholars [34]–[36]. Automatic sarcasm detection has attracted the NLP researchers' interest as the rising social media and sentiment analysis [11]. The existing text-based approaches can be classified into three categories: rule-based approaches, feature-based machine learning approaches, and deep learning-based approaches [1].

Rule-based approaches attempt to detect sarcasm through specific evidence. Veale *et al.*, [14] present a nine-step method for separating ironic from non-ironic similes with the help of Google searches. Riloff *et al.*, [3] notice that a common form of sarcasm on Twitter consists of a positive sentiment and a negative situation. Thus, they develop an iterative algorithm to expand positive and negative phrase sets. The collected phrases are used to detect sarcasm. Maynard *et al.*, [37] analyse the sentiment and sarcasm within hashtags. They design a hashtag tokenizer and compile a set of rules to determine the sentiment polarity when knowing sarcasm.

However, rule-based methods heavily rely on fixed patterns, and it is challenging to handle the sarcastic text out of the designed patterns. Joshi *et al.*, [4] propose a system harnessing explicit and implicit incongruity features for sarcasm detection. They used SVM as the classifier, and their model outperforms two past works with 10%-20% F-score improvement. Ghosh *et al.*, [38] treat the sarcasm detection task as a word sense disambiguation problem. They applied the SVM classifier with modified kernel and word embeddings.

Recently, scholars apply deep learning methods to improve performance. Zhang *et al.*, [39] use a gated recurrent network to encode the text and a pooling network to extract contextual features. Poria *et al.*, [11] used pre-trained CNNs to extract sentiment, emotion and personality features for sarcasm detection. Tay *et al.*, [5] propose an intra-attention network to model the incongruity between word pairs in the sentence. As the isolated sentence sometimes shows no sarcasm, some researchers [40]–[44] involve the conversational context of the sentence for prediction. This type of conversational sarcasm detection is becoming a hot spot in recent years, and some workshops [30] raise attention to it.

Some valuable works concentrate on sarcasm detection beyond the text. Schifanella *et al.*, [45] firstly concatenate image

	Sentence	Knowledge	Correctly classified
BERT	I love these study hours in junker !	N/A	No
Model(ours)	I love these study hours in junker !	headache	Yes
BERT	Running on 3 hours of sleep . didn't even touch my humanities , good start to a good day	N/A	No
Model(ours)	Running on 3 hours of sleep . didn't even touch my humanities , good start to a good day	fatigue tiredness	Yes
BERT	So happy to just find out Uoit decided to reschedule all my lectures and tutorials for me to night classes at exact same time	N/A	No
Model(ours)	So happy to just find out Uoit decided to reschedule all my lectures and tutorials for me to night classes at exact same time	headache migraine	Yes

Fig. 5. The figure illustrates the attention visualization of some sarcastic samples. Words with higher attention weights are marked by brighter colour.

features and text features to identify sarcasm. Followingly, Cai *et al.*, [9] treat text feature, image feature, and image attributes as three modalities and propose a hierarchical fusion model for this work. Pan *et al.*, [10] design a system which models both intra-modality incongruity and inter-modality incongruity. Castro *et al.*, [8] propose a new dataset which consists of audiovisual utterances. Mishra *et al.*, [7] propose a cognitive NLP system for sentiment and sarcasm classification.

V. Conclusion

In this paper, to settle the absence of commonsense knowledge in sarcasm detection, we propose a novel BERT-based model that can effectively process commonsense knowledge. Specifically, we utilize the COMET model fine-tuned on ConceptNet to generate commonsense knowledge candidates. To extract useful knowledge information from these candidates, we compare two kinds of knowledge selection strategies: explicit knowledge selection and implicit attention-based knowledge selection. The explicit selection includes majority, minority, and contrast sentiment-based methods. Furthermore, we design a knowledge-text integration module to combine the information from both text and knowledge. To testify the effectiveness of our method, we conduct experiments on three datasets. Our method achieves better performance compared with baselines. More in-depth analysis is provided to show the behavior of our proposed modules, which further elucidates the validity of our method. In the future work, we will expand our method to satisfy more categories of sarcasm detection so that limitations of our method can be alleviated.

References

[1] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 73:1–73:22, 2017.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2007.

[3] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 704–714. [Online]. Available: http://aclweb.org/anthology/D/D13/D13-1066.pdf

[4] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, 2015, pp. 757–762. [Online]. Available: https://www.aclweb.org/anthology/P15-2124/

[5] Y. Tay, A. T. Luu, S. C. Hui, and J. Su, "Reasoning with sarcasm by reading in-between," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1010–1020. [Online]. Available: https://www.aclweb.org/anthology/P18-1093/

[6] T. Xiong, P. Zhang, H. Zhu, and Y. Yang, "Sarcasm detection with self-matching networks and low-rank bilinear pooling," in *Proc. World Wide Web Conf.*, 2019, pp. 2115–2124. [Online]. Available: https://doi.org/10.1145/3308558.3313735

[7] A. Mishra, K. Dey, and P. Bhattacharyya, "Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 377–387. [Online]. Available: https://doi.org/10.18653/v1/P17-1035

[8] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, "Towards multimodal sarcasm detection (an_obviously_perfect paper)," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, pp. 4619–4629. [Online]. Available: https://www.aclweb.org/anthology/P19-1455/

[9] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in twitter with hierarchical fusion model," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, Florence, Italy, Jul. 28–Aug. 2, 2019, pp. 2506–2515. [Online]. Available: https://doi.org/10.18653/v1/p19-1239

[10] H. Pan, Z. Lin, P. Fu, Y. Qi, and W. Wang, "Modeling intra and inter-modality incongruity for multi-modal sarcasm detection," in *Proc. Conf. Empirical Methods Natural Lang. Process, Findings*, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, 2020, pp. 1383–1392. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.124/

[11] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," in *Proc. COLING*, 2016, pp. 1601–1612. [Online]. Available: https://www.aclweb.org/anthology/C16-1151/

[12] F. Ilievski, A. Oltramari, K. Ma, B. Zhang, D. L. McGuinness, and P. A. Szekely, "Dimensions of commonsense knowledge," *Knowl. Based Syst.*, vol. 229, 2021, Art. no. 107347.

[13] S. Storks, Q. Gao, and J. Y. Chai, "Recent advances in natural language inference: A. survey of benchmarks, resources, and approaches," 2019, *arXiv:1904.01172*. [Online]. Available: https://arxiv.org/abs/1904.01172

[14] T. Veale and Y. Hao, "Detecting ironic intent in creative comparisons," in *Proc. ECAI - 19th Eur. Conf. Artif. Intell.,* Lisbon, Portugal, *Proc., Ser. Front. Artif. Intell. Appl.*, vol. 215, H. Coelho, R. Studer, and M. J. Wooldridge, Eds. IOS Press, 2010, pp. 765–770. [Online]. Available: https://doi.org/10.3233/978-1-60750-606-5-765

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Minneapolis, MN, USA, vol. 1, pp. 4171–4186, 2019. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[16] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Çelikyilmaz, and Y. Choi, "COMET: commonsense transformers for automatic knowledge graph construction," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, Florence, Italy, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 4762–4779. [Online]. Available: https://doi.org/10.18653/v1/p19-1470

[17] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, New Orleans, Louisiana, USA, 2018, pp. 2227–2237. [Online]. Available: https://doi.org/10.18653/v1/n18-1202

[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[19] A. Vaswani*et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. 30, Annu. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need

[20] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Commonsense knowledge base completion," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 1445–1455. [Online]. Available: https://doi.org/10.18653/v1/p16-1137

[21] M. Sap *et al.*, "ATOMIC: An atlas of machine commonsense for if-then reasoning," in *Proc. 33rd AAAI Conf. Artif. Intell., 31st Innov. Appl. Artif. Intell. Conf., Proc. 9th AAAI Symp. Educ. Adv. Artif. Intell.*, Honolulu, Hawaii, USA. AAAI Press, 2019, pp. 3027–3035. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33013027

[22] T. Chakrabarty, D. Ghosh, S. Muresan, and N. Peng, "R^3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 7976–7986. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.711

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

[24] A. Ghosh and T. Veale, "Fracking sarcasm using neural network," in *Proc. 7th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal., WASSA, NAACL-HLT*, 2016, pp. 161–169. [Online]. Available: http://aclweb.org/anthology/W/W16/W16-0425.pdf

[25] T. Ptácek, I. Habernal, and J. Hong, "Sarcasm detection on czech and english twitter," in *Proc. COLING, 25th Int. Conf. Comput. Linguistics, Proc. Conf.*, Dublin, Ireland, J. Hajic and J. Tsujii, Eds. ACL, 2014, pp. 213–223. [Online]. Available: https://www.aclweb.org/anthology/C14-1022/

[26] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," in *Proc. 111th Int. Conf. Lang. Resour. Eval., LREC*, Miyazaki, Japan, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. European Language Resources Association (ELRA), Miyazaki, Japan, 2018, pp. 641–646. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2018/summaries/160.html

[27] S. M. Lukin and M. A. Walker, "Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue," in *Proc. Workshop Lang. Anal. Soc. Media*, Atlanta, GA, USA, Jun. 2013, pp. 30–40. [Online]. Available: https://aclanthology.org/W13-1104

[28] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process., Doha, Qatar, ACL*, 2014, pp. 1746–1751. [Online]. Available: https://doi.org/10.3115/v1/d14-1181

[29] H. Lee, Y. Yu, and G. Kim, "Augmenting data for sarcasm detection with unlabeled conversation context," in *Proc. 2nd Workshop Figurative Lang. Process., Fig-Lang, ACL 2020*, 2020, pp. 12–17. [Online]. Available: https://doi.org/10.18653/v1/2020.figlang-1.2

[30] D. Ghosh, A. Vajpayee, and S. Muresan, "A report on the 2020 sarcasm detection shared task," in *Proc. 2nd Workshop Figurative Lang. Process., Fig-Lang, ACL*, 2020, pp. 1–11. [Online]. Available: https://doi.org/10.18653/v1/2020.figlang-1.1

[31] R. Lin, J. Xiao, and J. Fan, "Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification," in *Comput. Vis., ECCV 2018 Workshops, Munich, Germany, Proc., Part IV*, 2018, pp. 206–218. [Online]. Available: https://doi.org/10.1007/978-3-030-11018-5_19

[32] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. 32, Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2019, pp. 8024–8035. [Online]. Available: http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations,*, San Diego, CA, USA, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[34] J. D. Campbell and A. N. Katz, "Are there necessary conditions for inducing a sense of sarcastic irony?," *Discourse Processes*, vol. 49, no. 6, pp. 459–480, 2012.

[35] J. Eisterhold, S. Attardo, and D. Boxer, "Reactions to irony in discourse: evidence for the least disruption principle," *J. Pragmatics*, vol. 38, no. 8, pp. 1239–1256, 2006.

[36] R. Giora, "On irony and negation," *Discourse Processes*, vol. 19, no. 2, pp. 239–264, 1995.

[37] D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in *Proc. 9th Int. Conf. Lang. Resour. Eval.*, Reykjavik, Iceland,, 2014, pp. 4238–4243. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/summaries/67.html

[38] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1003–1012. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1116.pdf

[39] M. Zhang, Y. Zhang, and G. Fu, "Tweet sarcasm detection using deep neural network," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Proc. Conf.*, Osaka, Japan, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. ACL, 2016, pp. 2449–2460. [Online]. Available: https://www.aclweb.org/anthology/C16-1231/

[40] D. Bamman and N. A. Smith, "Contextualized sarcasm detection on twitter," in *Proc. 9th Int. Conf. Web Social Media, Univ. Oxford, Oxford, UK*, 2015, pp. 574–577. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10538

[41] A. Joshi, V. Tripathi, P. Bhattacharyya, and M. J. Carman, "Harnessing sequence labeling for sarcasm detection in dialogue from TV series 'friends'," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, Berlin, Germany, 2016, pp. 146–155. [Online]. Available: https://doi.org/10.18653/v1/k16-1015

[42] D. Ghosh, A. R. Fabbri, and S. Muresan, "The role of conversation context for sarcasm detection in online interactions," in *Proc. 18th Annu. SIGdial Meeting Discourse Dialogue*, Saarbrücken, Germany, 2017, pp. 186–196. [Online]. Available: https://doi.org/10.18653/v1/w17-5523

[43] A. Ghosh and T. Veale, "Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 482–491. [Online]. Available: https://doi.org/10.18653/v1/d17-1050

[44] Z. Wang, Z. Wu, R. Wang, and Y. Ren, "Twitter sarcasm detection exploiting a context-based model," in *Proc. Web Inf. Syst. Eng. – 16th Int. Conf.*, Miami, FL, USA, 2015, pp. 77–91. [Online]. Available: https://doi.org/10.1007/978-3-319-26190-4_6

[45] R. Schifanella, P. de Juan, J. R. Tetreault, and L. Cao, "Detecting sarcasm in multimodal social platforms," in *Proc. ACM Conf. Multimedia Conf.*, Amsterdam, The Netherlands, 2016, pp. 1136–1145. [Online]. Available: https://doi.org/10.1145/2964284.2964321

Jiangnan Li is currently a Postgraduate Student with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and the School of Cyber Security, University of Chinese Academy of Sciences. His current research interests include sentiment analysis and natural language processing.

Hongliang Pan is currently a Postgraduate Student with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and the School of Cyber Security, University of Chinese Academy of Sciences. His current research interests include sentiment analysis and natural language processing.

Zheng Lin is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and the School of Cyber Security, University of Chinese Academy of Sciences. Her current research interests include sentiment analysis and natural language processing.

Peng Fu is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His current research interests include sentiment analysis and natural language processing.

Weiping Wang is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His main research interests include Big Data and artificial intelligence