# Prompt Tuning with Contradictory Intentions for Sarcasm Recognition

**Yiyi Liu, Ruqing Zhang, Yixing Fan, Jiafeng Guo, Xueqi Cheng**
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{liuyiyi17s,zhangruqing,fanyixing,guojiafeng,cxq}@ict.ac.cn

## Abstract

Recently, prompt tuning has achieved promising results in a variety of natural language processing (NLP) tasks. The typical approach is to insert text pieces (i.e., templates) into the input and transform downstream tasks into the same form as pre-training. In essence, a high-quality template is the foundation of prompt tuning to support the performance of the converted cloze-style task. However, for sarcasm recognition, it is time-consuming and requires increasingly sophisticated domain knowledge to determine the appropriate templates and label words due to its highly figurative nature. In this work, we propose SarcPrompt, to incorporate the prior knowledge about contradictory intentions into prompt tuning for sarcasm recognition. SarcPrompt is inspired by that the speaker usually says the opposite of what they actually mean in the sarcastic text. Based on this idea, we explicitly mimic the actual intention by prompt construction and indicate whether the actual intention is contradictory to the literal content by verbalizer engineering. Experiments on three public datasets with standard and low-resource settings demonstrate the effectiveness of our SarcPrompt for sarcasm recognition.

## 1 Introduction

Sarcasm is a sophisticated language phenomenon in which one conveys implicit intention with the opposite meaning of what is said or written literally (Campbell and Katz, 2012; Joshi et al., 2015). Due to its high ambivalence and figurative nature, sarcasm recognition which targets to predict a text as sarcastic or non-sarcastic, becomes a particularly challenging classification task. With the usage of sarcasm becoming prevalent on social media platforms like microblogs and online forums, sarcasm recognition has received growing research attention to facilitate sentiment analysis applications. Recent advances have shown that Pre-trained Language Models (PLMs), such as RoBERTa (Liu et al., 2019b), BERT (Devlin et al., 2019), can achieve promising performance in many downstream Natural Language Processing (NLP) tasks (Xu et al., 2019; Liu et al., 2019a; Zhou et al., 2021). The success of PLMs has also attracted much attention for sarcasm recognition. Researchers mainly add extra classifiers on top of PLMs (Lou et al., 2021) to further train the models under classification objectives, or re-train popular PLMs by incorporating sentiment knowledge and external sarcastic corpus (Babanejad et al., 2020).

Despite fine-tuning has achieved satisfying results, some recent studies (Schick and Schütze, 2021a,b) have found that one of its critical challenges is the significant gap in objective forms between pre-training and fine-tuning. This largely limits the transfer and adaptation of knowledge in PLMs to downstream tasks. Recently, a series of studies propose to use prompt tuning (Han et al., 2021; Schick and Schütze, 2021a; Chen et al., 2022b) to bridge the gap between pre-training and fine-tuning. Specifically, the downstream task is formulated as a (masked) language modeling problem similar to the pre-training. By fusing the original input with the specially constructed prompt template to predict `[MASK]` and then mapping predicted words to corresponding labels, we can stimulate the task-related knowledge in PLMs to boost the model's performance.

Nevertheless, there are still several non-trivial challenges for sarcasm recognition with prompt tuning as follows. On the one hand, texts of a specific task can differ from that of PLMs used in prompt tuning (Chen et al., 2022b). For sarcasm recognition, the sarcastic data is usually composed of abundant deliberately ambiguous texts. In contrast, the corpora for PLMs in prompt tuning mainly include objective and deterministic texts (Liu et al., 2022). This greatly restricts PLMs from taking full advantage of their knowledge. On the other hand, though prompt tuning works well in text classifica-



*Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–339
May 2-6, 2023 

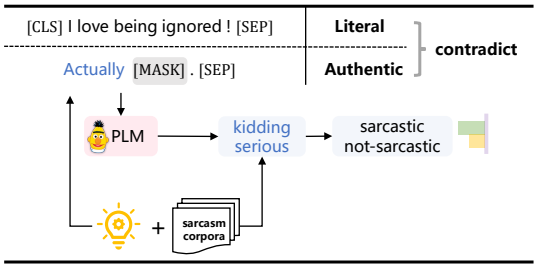


Figure 1: The example of prompt tuning to stimulate the contradictory for sarcasm recognition.

tion tasks, prompt templates and verbalizer are not easily transferable to sarcasm recognition. Both handcrafting an appropriate prompt template and choosing effective label words require domain expertise in sarcasm language. Furthermore, there exists deliberate ambiguity in sarcasm and the special written content cannot be ignored. In this sense, we argue that the power of prompt tuning has not been fully exploited for sarcasm recognition.

Shed light by the above challenges and insights, in this paper, we propose SarcPrompt, a novel prompt tuning method to leverage the contradiction knowledge-enhanced prompts to tune the PLMs. SarcPrompt is inspired by that sarcasm is known as "the activity of saying or writing the opposite of what you mean" (Tungthamthiti et al., 2014). According to this definition, we can recognize sarcasm by evaluating the inconsistency between the actual intention and the literal content in sarcastic texts. Based on intention contradiction, we carefully devise the prompt templates that can mimic the speakers' actual intention and then select label words to judge whether the prompt is contradictory to the literal content. Take the sentence in Figure 1 as an example, the original input is "I love being ignored". A good prompt template may denote "***Actually*** *[MASK]*". "Actually" is an indicator to trigger the authentic intention. If PLMs predict the masked position with a good label word "kidding", the new completed sentence denotes "I love being ignored. ***Actually*** *kidding*.". The intention of the prompt template completed with the predicted word is contradictory to the original input's content. Then we can intuitively recognize a sarcastic text.

To be specific, SarcPrompt mainly contains two steps: prompt construction and verbalizer engineering. Firstly, in the stage of prompt construction, we devise two kinds of sarcasm-specific prompts with different patterns to mimic the actual intention straightforwardly. Secondly, during the verbalizer engineering, we determine the label words that trigger contradictory or suggest sarcasm based on the statistical information of sarcastic corpora. Furthermore, we investigate a contrastive loss to comprehend various sarcastic contrast patterns, jointly with the cross-entropy loss for optimization. The contrastive loss strives to pull the representation of a sarcastic text towards that of other sarcastic texts in the same mini-batch, while pushing it away from representations of other non-sarcastic texts.

We conduct extensive experiments on three benchmark datasets for sarcasm recognition. Empirical experimental results demonstrate that our SarcPrompt[1] has achieved state-of-the-art performance under both standard supervised settings and low-resource settings.

## 2 Related Work

In this section, we briefly review two lines of related work, including sarcasm recognition and prompt tuning.

**Sarcasm Recognition.** Identifying sarcasm in texts has evolved from traditional methods to deep neural methods. Traditional approaches mostly utilize machine learning methods with manually engineered features (Riloff et al., 2013; González-Ibáñez et al., 2011; Patra et al., 2016; Hee et al., 2018b; González-Ibánez et al., 2011; Joshi et al., 2015). Deep neural models for sarcasm recognition aim to capture the contrast of sarcasm through the design of the model structure. They are mainly divided into two categories: contrast between words or phrases intra-sentence(Ghosh and Veale, 2016; Wu et al., 2018; Ghosh and Veale, 2018; Tay et al., 2018; Xiong et al., 2019; Lou et al., 2021) and contrast based on the essence of sarcasm phenomenon (Liu et al., 2022).

With the advent of pre-trained language models, a lot of sarcasm recognition methods based on PLMs have been proposed. There are two lines of utilizing PLMs: fine-tuning and pre-training. Fine-tuning mode uses PLMs as an encoder to obtain text representation (Lou et al., 2021; Liu et al., 2022; Li et al., 2021). Pre-training mode usually combines sentiment knowledge and external sarcastic corpora to advance the performance of sarcasm recognition (Babanejad et al., 2020). However, the existing methods using PLMs have many drawbacks which make PLMs ineffective in sarcasm recognition. Pre-training a new language model for a specific task is seriously affected by the external

[1]https://github.com/yiyi-ict/sarcprompt.

knowledge used, and also increases the complexity. And the acquisition and selection of external knowledge suitable for different sarcasm patterns are not easy.

**Prompt Tuning.** With the emergence of GPT-3 (Brown et al., 2020), prompt tuning, as a new paradigm for utilizing PLMs, has attracted more and more attention of researchers. Prompt tuning methods have achieved promising performance in many NLP tasks (Schick and Schütze, 2021a; Hu et al., 2022; Chen et al., 2022a; Zhong et al., 2021; Chen et al., 2022b). Especially, a lot of work has been proposed to solve text classification, which can be divided into two categories: manual prompts (Schick and Schütze, 2021a) and automatic prompts (Gao et al., 2021; Schick et al., 2020). Hu et al. (2022) propose to incorporate external knowledge into verbalizer for text classification. Chen et al. (2022b) propose to use input with prompt as a query to retrieve relevant task-specific data from large raw texts, which makes prompt tuning better fit classification tasks. Shin et al. (2020) propose gradient-guided search method to automatically generate prompt templates. However, these work cannot be adapted to sarcasm recognition directly. So far there is no prompt tuning method specially designed for sarcasm recognition.

## 3 Background

Before introducing SarcPrompt, we first briefly review the regular prompt tuning method for sentiment classification. Formally, let $M$ be a masked language model with vocabulary $V$ and mask token `[MASK]` $\in V$, and let $\mathcal{Y}$ be the set of labels for sentiment classification.

Traditional classification methods including fine-tuning PLMs train a model to take in an input $\mathbf{x} = (x_0, x_1, \ldots, x_n)$, generate a probability distribution over class $\mathcal{Y}$ and predict an output $y$ as $P(y|\mathbf{x})$. As for prompt tuning, the input $\mathbf{x}$ is wrapped with a *prompt template*. For example, assuming we need to classify the input sentence $\mathbf{x}$ = "Best pizza ever!" into label POSITIVE or NEGATIVE, the prompt template is defined as "It is `[MASK]`.". This is a general template widely accepted and used in text classification tasks (Schick and Schütze, 2021a; Hu et al., 2022), which is also applicable to sarcasm recognition task. Then the wrapped input is

$$\mathbf{x}_\text{p} = \mathbf{x} \text{ It is } \texttt{[MASK]}.$$

Then $M$ generates a probability over vocabulary $V$ on position `[MASK]`, which gives the probability of each token $v \in V$ being filled in `[MASK]` token $P_M(\texttt{[MASK]} = v|\mathbf{x}_\text{p})$. Furthermore, a *verbalizer* is to map from *label word* set $V_y \in V$ to the label space $\mathcal{Y}$. Corresponding to the above prompt, we may define $V_{pos}$ = {"positive"} as the label word of "positive" class, $V_{neg}$ = {"negative"} denotes the label word of "negative" class. Then the probability of label $y$ is calculated as

$$P(y|\mathbf{x}_\text{p}) = g(P_M(\texttt{[MASK]} = v|\mathbf{x}_\text{p})|v \in V_y),$$

where $g$ is a function that transforms the probability of label words into the probability of the class. If $P(y_{pos}) > P(y_{neg})$, we classify the instance into POSITIVE.

## 4 SarcPrompt

In this section, we introduce our prompt tuning model for sarcasm recognition (SarcPrompt) in detail. The overview of SarcPrompt is shown in Figure 2. We first introduce prompt construction and then verbalizer engineering. Finally we elucidate the training objective.

### 4.1 Prompt Construction

Recall that the representative characteristic of sarcastic texts is the contradiction between the literal content and the actual intention. The actual intention usually hides behind the literal content obscurely. The goal of prompt construction is to mimic the actual intention. Specifically, we define two kinds of prompts, including clash prompt and question prompt.

**Clash Prompt.** The goal of clash prompt for sarcasm recognition is to mimic the actual intention. For sarcastic texts, a good prompt template should reflect disapproval of the original text; for non-sarcastic texts, the prompt should agree with the meaning of the original text. Therefore, we utilize fact-related phrases which can either express the inconsistency with facts or consistency with facts.

We design five clash prompts $T_{c_1}(\mathbf{x})$~$T_{c_5}(\mathbf{x})$. Key phrases such as "in fact", "actually" are used to elicit whether the prompt clause has contradictory intentions with original input $\mathbf{x}$.

$T_{c_1}(\mathbf{x}) = \mathbf{x}$ Actually `[MASK]`.

$T_{c_2}(\mathbf{x}) = \mathbf{x}$ In reality, it was `[MASK]`.

$T_{c_3}(\mathbf{x}) = \mathbf{x}$ As a matter of fact, it was `[MASK]`.

$T_{c_4}(\mathbf{x}) = \mathbf{x}$ To tell you the truth, it was `[MASK]`.

$T_{c_5}(\mathbf{x}) = \mathbf{x}$ In fact, it was `[MASK]`.

**Question Prompt.** Question prompt is equiva-

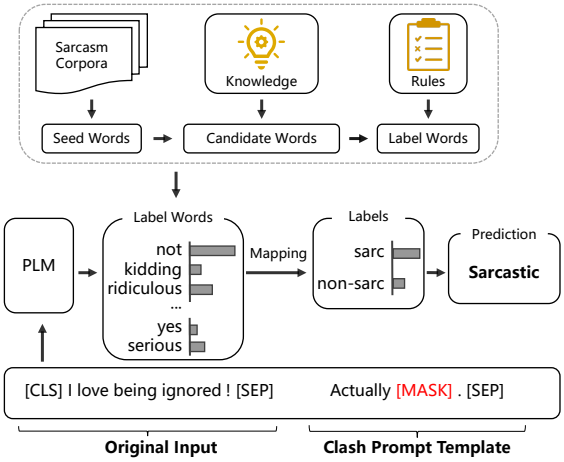


Figure 2: SarcPrompt specified with clash prompt.

lent to directly telling the model that the task is to identify sarcasm by asking a question. Inspired by the summary of prompt design for different tasks in (Liu et al., 2021), we propose three question prompt templates $T_{q_1}(\mathbf{x})$~$T_{q_3}(\mathbf{x})$ for sarcasm recognition. Compared with clash prompt, question prompt is a more direct way to judge whether the input text $\mathbf{x}$ is sarcastic.

$T_{q_1}(\mathbf{x}) = \mathbf{x}$ Are you kidding? `[MASK]`.

$T_{q_2}(\mathbf{x}) = \mathbf{x}$ Are you sarcastic? `[MASK]`.

$T_{q_3}(\mathbf{x}) = \mathbf{x}$ Are you ironic? `[MASK]`.

### 4.2 Verbalizer Engineering

By fusing the original input with the prompt template, we can obtain a new input $\mathbf{x}_\mathrm{p}$. Then PLMs can output a probability distribution over the vocabulary set $V$. The goal of verbalizer engineering is to determine label words that should be filled in the masked position, and then how to map label words to the corresponding labels. Verbalizer engineering is crucial because label words directly determine the corresponding relationship with sarcasm or not. Specifically, verbalizer engineering consists of two steps: label word searching and verbalizer mapping.

#### 4.2.1 Label Word Searching

The goal of label word searching is to find words that are appropriate in the masked position. The key idea is to find words that suggest the contradictory intention between the original input and the prompt template. For question prompt, "Yes" or "No" is enough to answer the questions. There is no need to search for label words.

For clash prompt, the label word searching mainly includes three steps: (1) to determine seed words based on the statistical information of sarcastic corpora; (2) to retrieve candidate words by knowledge based on the seed words, and (3) to denoise and obtain the final label words based on rules. In the next, we will introduce the process of label word searching for the clash prompt.

Table 1: Top 5 frequently appearing hashtags in a sarcastic tweet dataset.

| Hashtag | Frequency |
|---|---|
| #not | 42.12% |
| #sarcasm | 37.62% |
| #irony | 21.14% |
| #joke | 0.84% |
| #kidding | 0.37% |

**Seed Words.** The regular prompt tuning method usually uses the class name as the only label word for each class directly. For the sarcastic class, the label word can be "sarcasm", however, for the non-sarcastic class, the label word is not trivial to find. The intuitive idea of designing verbalizer is to reflect the contradiction of the literal content and the actual intention in one sarcastic text. We observe that people tend to add sarcasm-related hashtags to suggest the content they post expresses sarcastic meaning especially on the Twitter platform, which is shown below.

*I love waking up with migraines* ***#not***
*I just love when you test my patience!!* ***#sarcasm***

We count the frequency of the hashtags used in a sarcastic dataset consisting of tweets[2]. And the top five frequently appearing hashtags in sarcastic texts are shown in Table 1. Expressions in real scenes provide us with prior knowledge for determining label words of sarcasm recognition, which is exactly what we want in SarcPrompt. We use "not", "irony", "sarcasm", "kidding", "joke", which appear frequently in the sarcasm-related hashtags as seed words for sarcastic class. On the contrary, their antonyms serve as seeds for the non-sarcastic class. These words are well-suit to indicate that the prompt template has contradictory intentions with the original input's content.

**Candidate Words.** Based on the seed words, we can retrieve more candidate words from the knowledge base. The process of predicting masked words based on the context is not a single-choice procedure. There is no standard correct answer. Maybe abundant words fit this context. So it is necessary to expand the candidate word set.

[2]https://github.com/Cyvhee/SemEval2018-Task3

Table 2: Examples of the final label words for different prompts.

| Prompt Type | Label | Label Words |
|---|---|---|
| Clash Prompt | Sarcastic | not, sarcasm, irony, joke, kidding, no, ridiculous... |
| | Non-sarcastic | yes, do, so, serious, true, real, indeed... |
| Question Prompt | Sarcastic | yes |
| | Non-sarcastic | no |

Inspired by (Hu et al., 2022), we choose Related Words[3] as our external KB to widen the coverage of candidate words. Related Words is a knowledge graph $\mathcal{G}$ aggregated from multiple resources, including ConceptNet (Speer et al., 2017), WordNet (Pedersen et al., 2004), et al..

**Label Words.** After the expansion of seed words, we obtain comprehensive candidate words. However, the collected candidate words can be noisy since the vocabulary of the KB is not tailored for sarcasm. Thus we refine the candidate word set by frequency and part of speech. There are several criteria for denoising candidate words: (1) For words out of PLMs' vocabulary and sarcastic corpus, discard them; (2) Only keep words whose parts of speech are adjective, noun, verb, and adverb to ensure consistency with prompt templates syntactically and semantically. Finally, label words are defined in Table 2.

#### 4.2.2 Verbalizer Mapping.

The goal of verbalizer mapping is to map the predicted probability of label words to the final label $y$. For question prompt, since there is only one label word in each class, the probability of the label word is the probability of the corresponding class.

For clash prompt, there is more than one label word for each class. We use the average of label words based on label word searching process as the probability of each class, which is

$$P(\hat{y}) = \frac{\sum_{v \in V_y} P_M(\texttt{[MASK]} = v | \mathbf{x}_p)}{|V_y|}. \quad (1)$$

### 4.3 Training Objective

The process of SarcPrompt is a prompt-oriented fine-tuning approach (Gu et al., 2022). We need to compute the loss based on the supervised label of datasets and train the model. The training objective of SarcPrompt considers two aspects including cross-entropy loss and contrastive loss.

**Cross-entropy Loss.** The first training objective is to minimize the cross-entropy loss of the sarcasm label probability distribution. The cross-entropy loss is to ensure the basic ability of sarcasm recognition. The objective $L_{sarc}$ is formulated as:

$$L_{sarc}(\theta) = \sum \text{cross-entropy}(y, P(\hat{y})), \quad (2)$$

where $y$ is the groundtruth of the sarcasm label, and $P(\hat{y})$ is the predicted score.

**Contrastive Loss.** Inspired by (Khosla et al., 2020), we introduce contrastive learning objective to our SarcPrompt model. Sarcasm takes many patterns of expression such as sarcasm by clash, situational sarcasm, and other sarcasm (Hee et al., 2018a). Supervised contrastive learning is an automatic way of capturing the entailed similarity of various sarcasm patterns.

Specifically, for $(\mathbf{x}_i, y_i)$ within a batch, we first extract sentence representation $s_i$. Recall that in Equation 1, we obtain the probability distribution of the label by computing the score of predicted words in the masked position. This can not only be viewed as word representation in the masked position, but also be viewed as a kind of sentence representation. So we use this probability distribution as sentence representation $s_i$ to calculate the contrastive loss. Then the supervised contrastive loss in a batch $L_{con}$ is defined as:

$$P_{con}(i, c) = \frac{\exp\left(sim(s_i, s_c)/\tau\right)}{\sum_{b \in B, b \neq i} \exp\left(sim(s_i, s_b)/\tau\right)} \quad (3)$$

$$L_{con}(\theta) = \sum_{i \in B} -\log \frac{1}{C_i} \sum_{y_i = y_c, c \neq i} P_{con}(i, c). \quad (4)$$

Here $P_{con}(i, c)$ indicates the likelihood that $s_c$ is most similar to $s_i$. $\tau$ is the temperature of softmax. We use $sim(s_i, s_c) = s_i \cdot s_c$ for similarity calculation. Supervised contrastive loss $L_{con}$ is calculated for each input sentence representation $s_i$ in a batch. And $C_i = |\{c | y_c = y_i, c \neq i\}|$ is the number of samples in the same category $y_i$ in a batch.

Considering the two objectives, we obtain the final objective function $L$ by adding them together:

$$L(\theta) = \lambda_1 L_{sarc}(\theta) + \lambda_2 L_{con}(\theta),$$

[3]https://relatedwords.org

Table 3: Statistics of datasets. Avg $\ell$ denotes the average length of texts in the number of tokens.

| Dataset | Train | Valid | Test | Avg $\ell$ |
|---|---|---|---|---|
| IAC-V1[5] | 1,595 | 80 | 320 | 68 |
| IAC-V2[6] | 5,216 | 262 | 1,042 | 43 |
| Tweets[7] | 3,634 | 200 | 784 | 14 |

where $\theta$ is the parameter set of the model. $\lambda_1, \lambda_2$ are used to leverage the contributions.

## 5 Experimental Settings

### 5.1 Datasets

We carry out experiments on three benchmark datasets: IAC-V1, IAC-V2, and Tweets.

- **IAC-V1** (Lukin and Walker, 2017) and **IAC-V2** (Oraby et al., 2016) are collected from online political debates forum[4]. IAC-V2 contains more data than IAC-V1.
- **Tweets** dataset is proposed in SemEval 2018 Task 3 Subtask A (Hee et al., 2018a).

Table 3 reports the statistics. For all datasets, we follow the train/valid/test split in (Liu et al., 2022). All datasets are class-balanced.

### 5.2 Baselines

We adopt three types of baseline methods for comparison, including deep models without pre-training, pre-trained models with fine-tuning, and pre-trained models with prompt tuning.

**Deep Models without Pre-training.** We choose recent and widely compared deep models including LSTM-based Bi-LSTM (Hochreiter and Schmidhuber, 1997), CNN-LSTM-DNN (Ghosh and Veale, 2016), attention-based MIARN (Tay et al., 2018), graph neural network-based ADGCN (Lou et al., 2021) and state-of-the-art DC-Net (Liu et al., 2022). In this work, we study context-free sarcasm recognition. The input is a sentence without contexts like history posts and user profiles (Hazarika et al., 2018). So the baselines we choose are restricted with this area.

**Pre-trained Models with Fine-tuning.** For fine-tuning pre-trained models, we utilize RoBERTa$_{\text{base}}$ as the basic encoder to make a fair comparison. Both ADGCN and DC-Net report versions of fine-tuning pre-trained models in their paper. So we use the RoBERTa as backbone and implement ADGCN-RoBERTa and DC-Net-RoBERTa based on their released code.

**Pre-trained Models with Prompt Tuning.** Recall that in Section 3, we introduce a regular prompt tuning method for sentiment classification. We utilize this approach as the prompt tuning baseline. Specifically, the vebalizer of regular prompt tuning method is the same as clash prompt.

### 5.3 Implementation Details

Under the **standard supervised settings**, we utilize the whole datasets to fine-tune. For hyperparameters, we employ $\lambda_1 = 1$, and $\lambda_2$ among $\{0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$ on the validation set respectively. $0.005$, $0.5$ and $0.05$ are the best $\lambda_2$ for IAC-V1, IAC-V2 and Tweets respectively. As for the **low-resource settings**, we randomly sample $\mathbf{r}\%$ instances of each class from the initial training and validation sets to form the few-shot training and validation sets. $\mathbf{r}$ ranges from $\{1, 5, 10, 20\}$. Our model is implemented based on the open source PET[8]. PET (Schick and Schütze, 2021a) is a regular prompt tuning method that uses the class name as the only label word for each class. But we do not use any tricks in the original PET paper since we want to study the effect of sarcasm-specific templates and label words alone. For each method, we train them with three seeds and report the results of the best seed.

## 6 Experimental results

In this section, we report and analyze the experimental results to demonstrate the effectiveness of the proposed SarcPrompt method. Specifically, we target the following research questions:

- **RQ1:** How does SarcPrompt perform under the standard supervised settings?
- **RQ2:** How does SarcPrompt perform under the low-resource settings?
- **RQ3:** Which prompt performs best and why?
- **RQ4:** How does the contrastive loss affect the performance of SarcPrompt?
- **RQ5:** Can we better understand how SarcPrompt performs via some case studies?

### 6.1 Results under Standard Settings

To answer **RQ1**, we compare SarcPrompt with three kinds of strong baselines on three benchmark

[4]http://www.4forums.com/political/
[5]https://nlds.soe.ucsc.edu/sarcasm1
[6]https://nlds.soe.ucsc.edu/sarcasm2
[7]https://github.com/Cyvhee/SemEval2018-Task3
[8]https://github.com/timoschick/pet

Table 4: Precision, recall, macro $F1$, and accuracy under standard supervised settings. The "*" in the upper right corner of the model name represents that the results are retrieved from (Liu et al., 2022). Best results are bold.

| *Standard Supervised Settings* | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | IAC-V1 | | | | IAC-V2 | | | | Tweets | | | |
| | Pre. | Rec. | F1 | Acc. | Pre. | Rec. | F1 | Acc. | Pre. | Rec. | F1 | Acc. |
| *Deep models without pre-training* | | | | | | | | | | | | |
| Bi-LSTM* | 64.6 | 64.6 | 64.6 | 64.6 | 79.8 | 79.7 | 79.7 | 79.7 | 71.8 | 71.7 | 71.7 | 73.0 |
| CNN-LSTM-DNN* | 61.5 | 61.2 | 60.9 | 61.1 | 75.4 | 75.3 | 75.2 | 75.3 | 71.9 | 72.9 | 71.9 | 72.3 |
| MIARN* | 65.6 | 65.2 | 64.9 | 65.2 | 75.4 | 75.3 | 75.2 | 75.3 | 68.6 | 68.8 | 68.8 | 70.2 |
| ADGCN* | 64.3 | 64.3 | 64.3 | 64.3 | 81.0 | 80.9 | 80.9 | 80.9 | 72.6 | 73.2 | 72.8 | 73.6 |
| DC-Net* | 66.6 | 66.5 | 66.4 | 66.5 | 82.2 | 82.1 | 82.1 | 82.1 | 76.4 | 77.5 | 76.3 | 76.7 |
| *Pre-trained models with fine-tuning* | | | | | | | | | | | | |
| RoBERTa | 73.0 | 72.1 | 71.9 | 72.1 | 82.9 | 82.8 | 82.7 | 82.7 | 72.7 | 72.8 | 72.8 | 73.9 |
| ADGCN-RoBERTa | 72.5 | 72.4 | 72.4 | 72.4 | 82.2 | 82.1 | 82.1 | 82.1 | 71.3 | 71.9 | 71.4 | 72.2 |
| DC-Net-RoBERTa | 69.7 | 69.3 | 69.1 | 69.3 | 83.7 | 83.7 | 83.7 | 83.7 | 69.7 | 68.3 | 68.7 | 70.9 |
| *Pre-trained models with prompt tuning* | | | | | | | | | | | | |
| Prompt Tuning-RoBERTa | 72.5 | 72.1 | 72.0 | 72.1 | 83.4 | 83.3 | 83.3 | 83.3 | 71.8 | 72.7 | 71.8 | 72.3 |
| **SarcPrompt**-Question-RoBERTa | 73.7 | 73.0 | 72.9 | 73.0 | 84.3 | 84.2 | 84.2 | 84.2 | 74.1 | 75.2 | 73.7 | 74.0 |
| **SarcPrompt**-Clash-RoBERTa | **75.5** | **75.2** | **75.2** | **75.2** | **84.9** | **84.9** | **84.9** | **84.9** | **77.1** | **78.3** | **76.6** | **76.9** |

datasets under standard supervised settings. Table 4 shows the results. We observe that: (1) Fine-tuning models perform better than traditional deep models on IAC-V1 and IAC-V2, which demonstrates the effectiveness of fine-tuning models. However, on Tweets, the results for fine-tuning models are similar to or even worse than traditional deep models. The reason may be that tweets are non-standard and chatty compared with the corpora of pre-trained models. (2) The performance of applying prompt tuning method for text classification directly to sarcasm recognition is similar to or even worse than fine-tuning methods. This indicates that neither fine-tuning models nor regular prompt tuning method can make enough use of the knowledge in PLMs for sarcasm recognition.

When we look at the two types of SarcPrompt, we find that: (1) SarcPrompt with clash prompt outperform all the baseline methods. SarcPrompt with clash prompt improves a lot over regular prompt tuning, which demonstrates that it is improper to simply transplant the regular prompt tuning method to sarcasm recognition. By stimulating sarcastic characteristics, SarcPrompt is able to well exploit the capability of pre-trained language models in sarcasm recognition. (2) SarcPrompt with clash prompt performs better than with question prompt, showing clash prompt is more valid to reflect the contradictory intentions of sarcastic texts. (3) The improvements of SarcPrompt over baselines on Tweets are higher than that on both IAC-V1 and IAC-V2. The reason may be that the label words selection process relies on the seed words extracted from Tweets. It is necessary to explore other ways to obtain the prior information in the future.

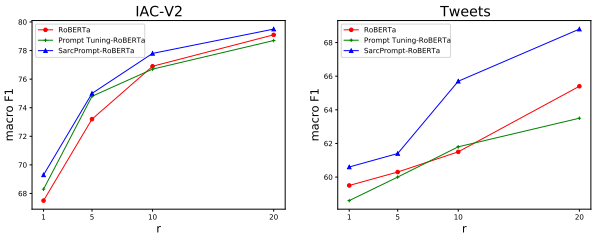


Figure 3: Macro $F1$ scores under low-resource settings, using 1%, 5%, 10%, and 20% instances for IAC-V2 and Tweets datasets.

## 6.2 Results under Low-resource Settings

In real-word scenes, it is often time-consuming and labor-intensive to collect annotated data to fine-tune classification models especially for difficult tasks like sarcasm recognition. To answer **RQ2**, we choose RoBERTa and Prompt Tuning-RoBERTa as baselines. For SarcPrompt, we use clash prompt type, which achieves the best performance under standard settings. IAC-V1 and IAC-V2 have the same source, and experimental results show the same trend under low-resource settings. So we report the results of IAC-V2 and Tweets datasets.

The experimental results are shown in Figure 3. We observe that: (1) Our SarcPrompt consistently outperforms the baseline methods with less training data on both datasets, which demonstrates the

Table 5: Cases of right and wrongly classified by SarcPrompt. The check mark indicates classification is correct, while the cross mark indicates wrong.

| ID | Input | Prompt with predicted word | Result |
|---|---|---|---|
| 1 | Pretty excited about how you gave up on me. | Actually *not*. | ✓ |
| 2 | I just love being ignored | Actually *kidding*. | ✓ |
| 3 | thanks I thought it was tomorrow | Actually *indeed*. | ✗ |
| 4 | make sure you don't say Christmas!! The decorate for the holiday. | Actually *yes*. | ✗ |

effectiveness of SarcPrompt model under low resource settings. (2) As $\mathbf{r}$ increases from 1 to 20, the improvement in our SacrPrompt over IAC-V2 decreases gradually. But for Tweets, with the increase of data, the improvement of SarcPrompt is also greater. The reason may be Tweets are short and non-standard, and may lack information. Also, SarcPrompt is based on the statistical information of Tweets. Therefore as the training data of Tweets increases, SarcPrompt can obtain more information and achieve better results, which is consistent with performance under standard settings.

### 6.3 Results of Different Prompts

To answer **RQ3**, we report the performance of different prompts of SarcPrompt model in Table 6. We observe that: (1) Among question prompts, $T_{q_1}$ performs best on IAC datasets and $T_{q_2}$ on Tweets. As for clash prompt, $T_{c_2}$ performs best on IAC datasets while $T_{c_1}$ on Tweets. It is worth noting that the best clash template on Tweets $T_{c_1}$ is short and colloquial, which is similar to the usage of Tweets. On the contrary, $T_{c_2}$ is a complete sentence. It matches IAC datasets, which are longer and more formal. (2) The effect of the template fluctuates more on IAC-V1 and Tweets datasets because they are relatively small and do not contain enough sarcastic patterns. An appropriate template can be well adapted to most of the data while an inappropriate template may perform poorly. This also shows that it is more difficult to find suitable templates for small datasets.

### 6.4 Ablation Study

To answer **RQ4**, we conduct an ablation study to analyze the impact of contrastive loss, which is shown in Table 7. Note that the removal of contrastive loss degrades the performance a lot, which indicates that the similarity information between the same class and the contrast information between different classes are significant in sarcastic expressions' learning.

Table 6: Performance (Macro $F1$) comparison of different prompt templates of SarcPrompt model. "AVG±VAR" means average results and the variances of question and clash prompts.

| Prompt Type | IAC-V1 | IAC-V2 | Tweets |
|---|---|---|---|
| Question-$T_{q_1}$ | **72.9** | **84.2** | 73.1 |
| Question-$T_{q_2}$ | 71.1 | 83.7 | **73.7** |
| Question-$T_{q_3}$ | 69.4 | 84.1 | 69.3 |
| AVG±VAR | 71.1±3.1 | 84.0±0.1 | 72.0±5.7 |
| Clash-$T_{c_1}$ | 73.4 | 83.3 | **76.6** |
| Clash-$T_{c_2}$ | **75.2** | **84.9** | 75.3 |
| Clash-$T_{c_3}$ | 70.8 | 83.5 | 73.8 |
| Clash-$T_{c_4}$ | 71.7 | 83.4 | 72.9 |
| Clash-$T_{c_5}$ | 73.0 | 84.6 | 73.7 |
| AVG±VAR | 72.8±2.8 | 83.9±0.6 | 74.5±2.2 |

Table 7: Performance (Macro $F1$) of SarcPrompt-Clash-RoBERTa with and without contrastive loss.

| Model | IAC-V1 | IAC-V2 | Tweets |
|---|---|---|---|
| SarcPrompt-Clash-RoBERTa | **75.2** | **84.9** | **76.6** |
| w/o contrative loss | 74.3 | 82.4 | 75.2 |

### 6.5 Case Study

To answer **RQ5**, we analyze sarcasm recognition results on Tweets by several cases in Table 5. We observe that: (1) Our SarcPrompt is good at dealing with input samples that contain subjective sentiment expressions, such as "excited" and "love" in the first and second samples. When combined with the prompt clause, the contradictory intention is obvious to recognize. (2) However, in the third and fourth samples, the contradictory intentions are not strong enough and hidden in deeper semantics. They require external contexts to assist sarcasm recognition. Specifically, the speaker remembers the wrong time in the third sample. In the fourth sample, the speaker complains the decoration is not good-looking for Christmas. This indicates that current templates and label words are not suitable for recognizing sarcasm in factual texts.

## 7 Conclusion

In this paper, we have proposed SarcPrompt, a simple and effective prompt tuning method to recognize sarcasm in texts. Specifically, we design several prompt templates to mimic the actual intention behind the sarcastic literal content. We define verbalizers based on the statistics of sarcastic corpus. Then it is able to determine whether the prompt is contradictory to the literal content by the predicted label words. Empirical experimental results show that SarcPrompt achieves state-of-the-art performance under both standard supervised settings and low-resource settings.

## Limitations

Although our SarcPrompt has achieved the SOTA performance on several benchmark datasets for sarcasm recognition, there are still limitations, mainly in the following aspects.

Firstly, we combine sarcastic characteristics into prompt tuning in a hard-coded form by manually designing prompt templates and label words based on the statistical information in sarcastic corpora. This hard-coded approach may not be able to adapt to all sarcasm patterns and may miss some good prompt templates or label words. Moreover, in the current verbalizer mapping process, we decay the weight of each label word according to the relationship in the knowledge base. The mapping approach is trivial and not learnable. Lastly, as we analyzed in the case study, current SarcPrompt is not good at dealing with situational sarcasm. In situational sarcasm pattern, there is no contradictory intention by looking at the literal content alone, which is hard to recognize even for humans.

In future work, we will explore continuous prompt templates and learnable mapping functions for prompt tuning in sarcasm recognition. Combining external knowledge is also a direction to make prompt tuning suitable for situational sarcasm in our future work.

## Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62006218 and 61902381, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2021100, the Young Elite Scientist Sponsorship Program by CAST under Grants No. YESS20200121, and the Lenovo-CAS Joint Lab Youth Scientist Project.

## References

Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. 2020. Affective and contextual embedding for sarcasm detection. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 225–243.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

John D Campbell and Albert N Katz. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*, 49(6):459–480.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022a. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2778–2788. ACM.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2022b. Adaprompt: Adaptive model training for prompt-based NLP. *CoRR*, abs/2202.04824.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

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, WASSA@NAACL-HLT 2016, June 16, 2016, San Diego, California, USA*, pages 161–169.

Aniruddha Ghosh and Tony Veale. 2018. Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 570–575.

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 581–586.

Roberto I. González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 581–586.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8410–8423. Association for Computational Linguistics.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1837–1848. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018a. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 39–50.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018b. We usually don't like going to the dentist: Using common sense to detect irony on twitter. *Comput. Linguistics*, 44(4).

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

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2225–2240. Association for Computational Linguistics.

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.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jiangnan Li, Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang. 2021. Sarcasm detection with commonsense knowledge. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3192–3201.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019a. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2431–2441.

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

Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, and Jiafeng Guo. 2022. A dual-channel framework for sarcasm recognition by detecting sentiment conflict. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1670–1680. Association for Computational Linguistics.

Chenwei Lou, Bin Liang, Lin Gui, Yulan He, Yixue Dang, and Ruifeng Xu. 2021. Affective dependency graph for sarcasm detection. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1844–1849.

Stephanie M. Lukin and Marilyn A. Walker. 2017. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. *CoRR*, abs/1708.08572.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn A. Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 31–41.

Braja Gopal Patra, Soumadeep Mazumdar, Dipankar Das, Paolo Rosso, and Sivaji Bandyopadhyay. 2016. A multilevel approach to sentiment analysis of figurative language in twitter. In *Computational Linguistics and Intelligent Text Processing - 17th International Conference, CICLing 2016, Konya, Turkey, April 3-9, 2016, Revised Selected Papers, Part II*, volume 9624 of *Lecture Notes in Computer Science*, pages 281–291. Springer.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet: : Similarity - measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*. The Association for Computational Linguistics.

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.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1010–1020.

Piyoros Tungthamthiti, Kiyoaki Shirai, and Masnizah Mohd. 2014. Recognition of sarcasms in tweets based on concept level sentiment analysis and supervised learning approaches. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 28, Cape Panwa Hotel, Phuket, Thailand, December 12-14, 2014*, pages 404–413. The PACLIC 28 Organizing Committee and PACLIC Steering Committee / ACL / Department of Linguistics, Faculty of Arts, Chulalongkorn University.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 51–56.

Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. 2019. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2115–2124.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2324–2335. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14612–14620. AAAI Press.