Think Beyond the Word: Understanding the Implied Textual Meaning by Digesting Context, Local, and Noise

Conference Paper · July 2020

DOI: 10.1145/3397271.3401435

CITATIONS
2

READS
121

7 authors, including:

Guoxiu He
Wuhan University
10 PUBLICATIONS **25** CITATIONS

SEE PROFILE

Zhuoren Jiang
Zhejiang University
51 PUBLICATIONS **220** CITATIONS

SEE PROFILE

Xiaozhong Liu
Indiana University Bloomington
145 PUBLICATIONS **821** CITATIONS

SEE PROFILE

Think Beyond the Word: Understanding the Implied Textual Meaning by Digesting Context, Local, and Noise

Guoxiu He[1,2], Zhe Gao[2], Zhuoren Jiang[3], Yangyang Kang[2], Changlong Sun[2], Xiaozhong Liu[4*]
Wei Lu[1*]
[1]School of Information Management, Wuhan University, Wuhan, China
[2]Alibaba Group, Hangzhou, China
[3]School of Public Affairs, Zhejiang University, Hangzhou, China
[4]Indiana University Bloomington, Bloomington, United States
guoxiu.he@whu.edu.cn;gaozhe.gz@alibaba-inc.com;jiangzhuoren@zju.edu.cn
yangyang.kangyy@alibaba-inc.com;changlong.scl@taobao.com;liu237@indiana.edu;weilu@whu.edu.cn

ABSTRACT

Implied semantics is a complex language act that can appear everywhere on the Cyberspace. The prevalence of implied spam texts, such as implied pornography, sarcasm, and abuse hidden within the novel, tweet, microblog, or review, can be extremely harmful to the physical and mental health of teenagers. The non-literal interpretation of the implied text is hard to be understood by machine models due to its high context-sensitivity and heavy usage of figurative language. In this study, inspired by human reading comprehension, we propose a novel, simple, and effective deep neural framework, called **S**kim and **I**ntensive **R**eading **M**odel (SIRM), for figuring out implied textual meaning. The proposed SIRM consists of three main components, namely the skim reading component, intensive reading component, and adversarial training component. N-gram features are quickly extracted from the skim reading component, which is a combination of several convolutional neural networks, as skim (entire) information. An intensive reading component enables a hierarchical investigation for both sentence-level and paragraph-level representation, which encapsulates the current (local) embedding and the contextual information (context) with a dense connection. More specifically, the contextual information includes the near-neighbor information and the skim information mentioned above. Finally, besides the common training loss function, we employ an adversarial loss function as a penalty over the skim reading component to eliminate noisy information (noise) arisen from special figurative words in the training data. To verify the effectiveness, robustness, and efficiency of the proposed architecture, we conduct extensive comparative experiments on an industrial novel dataset involving implied pornography and three sarcasm benchmarks. Experimental results indicate that (1) the proposed model, which benefits from context and local modeling and consideration of figurative language (noise), outperforms existing state-of-the-art solutions, with comparable parameter scale and running speed; (2) the SIRM yields superior robustness in terms of parameter size sensitivity; (3) compared with ablation and addition variants of the SIRM, the final framework is efficient enough.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; **Neural networks**; • **Applied computing** → **Document analysis**; • **Information systems** → **Spam detection**;

KEYWORDS

implied textual meaning, semantic representation, text classification, deep neural networks

ACM Reference Format:
Guoxiu He, Zhe Gao, Zhuoren Jiang, Yangyang Kang, Changlong Sun, Xiaozhong Liu and Wei Lu. 2020. Think Beyond the Word: Understanding the Implied Textual Meaning by Digesting Context, Local, and Noise. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401435

1 INTRODUCTION

The Internet is not a safe place for children and teenagers to be roaming around. Harmful contents in online novels, microblogs, and reviews, like erotica, sarcasm, abuse, and violence, are polluting the web space. The extremes of bad environmental evolution can even lead to crime[1]. To keep a clean web environment for users, especially for children, the majority of websites spend a lot of efforts on preventing spam texts.

As shown in Figure 1, texts such as comments, blogs, and novels submitted to websites by authors are checked by spam-detection systems. The first barrier can prevent plenty of obvious spam texts. Then the suspect texts will be sent to auditors for a double-check. However, taking pornography as an example, 9/10 boys and 6/10 girls will be exposed to pornography before they turn 18, and the majority of online exposures are unwanted and unwarranted, which escape from spam-detection systems[2]. With long time struggling experience against spam-detection systems, it is obvious that some

*Corresponding authors.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401435

[1]https://www.bbc.com/news/world-asia-52030219
[2]https://everaccountable.com/blog/how-pornography-affects-teenagers-and-children/

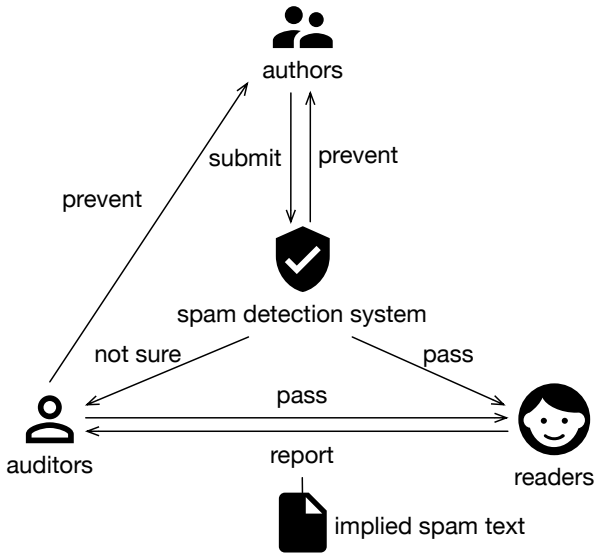

Figure 1: Procedure of Spam Detection.

authors of spam texts may purposely avoid using sensitive words to take advantage of the lack of human audits. In other words, detecting implicit harmful texts is not an easy task due to highly context-sensitive and figurative arousal contents. In particular, the battle against implicit spam/harmful texts is regularly considered as one of the key challenges in text classification and semantic understanding of long texts.

There is an exemplary paragraph from erotica shown in Figure 2. From learning viewpoint, this instance can be somehow difficult that we need to read it over and over to clarify its meaning. We can see that sentence (2), (4), and (5) are highly related to each other. And the all contexts lead to sentence (7). More specifically, it is sentence (5) that mainly makes the whole paragraph spam. Furthermore, words, even phrases, in this paragraph can highly likely be found in the normal samples. And even if *'clocks'* and *'Ben'* frequently appear in the spam samples of history, we can't use them as the main basis for speculating a new sample.

Another typical example about sarcasm is shown in Figure 3. From the sentence, *'rope'* (in sentence (2)) has two different meanings: literal meaning for *'bungee jumping'* (in sentence (1) and (3)) and implied meaning (*'umbilical cord'*) for *'came into this world'* (in sentence (2)). People can tell the difference after digesting the whole sentence. Furthermore, no matter how many times *'rope'* refers to *'umbilical cord'* in the training set, we can not assert that it conveys the same meaning in a new coming text.

From the two instances, we can see that language (e.g., sarcasm and metaphor) does not always express its literal meaning. People often use words that deviate from their conventionally accepted definitions in order to convey complicated and implied meanings [42]. Compared with standard (literal) text usage, the non-literal text can be associated with three typical linguistic phenomena as follows:

- From syntax and semantic viewpoints, the non-literal text is **highly context-sensitive**. People have to perceive the implied meaning of the text through unnatural language usage in context.
- From a lexicon viewpoint, the non-literal text is often created by presenting words which are equated, compared, or associated with **normally unrelated or figurative meanings**. These words express different or even opposite meanings, which could change the word distribution under a semantic topic or sentiment polarity and then hinder the training of machine models.
- In addition, some of these words frequently appearing in the training set will mislead the machine model in the inference process.

(1) I slowly moved up and down,
(2) timing my thrusts with the second hands on the clocks.
(3) As they began their final rotation of the year,
(4) I increased my speed.
(5) Ben wrapped his arm around my waist and dug himself further into me,
(6) which I didn't think was possible.
(7) My breath caught…

Figure 2: A typical case, which is a paragraph to express non-literal meaning about pornography.

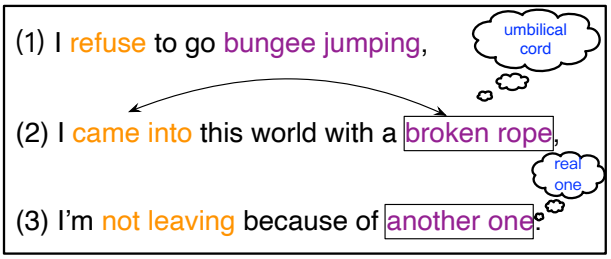

Figure 3: A typical case, which is a sentence to express implied sarcasm.

Existing text representation studies, which mainly rely on content embeddings [35] based deep neural networks [30] such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Attention Mechanisms, are not totally suitable for aforementioned problems. RNNs, such as Long Short Term Memory (LSTM) [19], Gated Recurrent Unit (GRU) [6], and Simple Recurrent Unit (SRU) [41] draw on the idea of the language model [3]. However, RNNs, including bidirectional ones, could neglect the long-term dependency, as demonstrated in [31, 40, 45], since the current term directly depends on the previous term as opposed to the entire information. Although attention mechanisms [47] over RNNs provide an important potential to aggregate all hidden states,

they focus more on the local part of a text. CNNs [24] can characterize local spatial features and then assemble these features by deeper layers which are expert in extracting phrase-level features. Self-attention mechanism [43] characterizes the dependency on one term with others in the input sequence and then encodes the mutual relationship to capture the contextual information. Unfortunately, all standard text representation models have not effectively utilized the contextual representation as input directly when encoding the current term, which is necessary to understand the implied meaning. There are also several tailored models for sarcasm detection [42], which concentrate more on word incongruity in the text.

1.1 Research Objectives

Hence, the study aims to cope with the following drawbacks which can be summarized as follows:

- Existing text representation models don't specifically design a mechanism to effectively and straightforwardly use context/global information when understanding the implied meaning of the input text.
- Meanwhile, all existing models neglect the potential bad effect of the figurative words which can be frequently appearing in the training set of implied texts.
- Existing methods do not take both model complexity and model performance into account at the time of design, which can be very important in practical applications.

To this end, we try to design a simple and effective model to interpret the implied meaning by overcoming the challenges mentioned above. From a human reading comprehension viewpoint, to understand a difficult text, a human may firstly skim it quickly to estimate the entire information of the target text. Then, in order to consume the content, he/she can read the text word by word and sentence by sentence with respect to the entire information. Furthermore, a human can skip some noisy or unimportant figurative phrases seen before to quickly consume the main idea of the target text. Inspired by the above procedure, in this study, we propose a novel deep neural network, namely Skim and Intensive Reading Model (SIRM), to address the implied textual meaning identifying problem. In particular, an adversarial loss is used in SIRM to eliminate the noisy information in training process. To the best of our knowledge, SIRM is the first model trying to simulate such human reading procedure for understanding and identifying the non-literal text.

Furthermore, taking efficiency into consideration, we design the details of the proposed SIRM under the Occam's Razor: *'More things should not be used than are necessary'*. In other words, under the premise of optimal task performance, we will remove unnecessary components and use the simplest architecture.

1.2 Contributions

Briefly, given the above research objectives, our main contributions of this work can be summarized as follows:

- The challenges of understanding the implied textual meaning are well investigated in this research, which can be summarized as follows: context-sensitivity and usage of figurative meaning. To the best of our knowledge, these challenges have not been thoroughly studied.
- We propose the SIRM to understand implied textual meaning where the intensive reading component, which enables a hierarchical investigation for sentence-level and paragraph-level representation, depends on the global information extracted by the skim reading component. The cooperation of the skim reading component and the intensive reading component in the SIRM achieves a positive impact on comprehending non-literal interpretation by modeling the contextual information directly.
- We introduce an adversarial loss as a penalty over the skim reading component to cut down noise due to special figurative words during the training procedure.
- We conduct extensive comparative experiments to show the effectiveness, robustness, and efficiency of the SIRM. Compared with the existing alternative models, the SIRM achieves superior performance on F1 score and accuracy with a comparable parameter size and training speed. In addition, the SIRM outperforms all other models according to model robustness. And the ablation and addition tests show that the final SIRM is efficient enough.

The remainder of this paper is structured as follows: Some related works are summarized in Section 2 and the details of the proposed SIRM are introduced in Section 3. Section 4 presents the experimental settings which is followed by results and analyses in Section 5. Our concluding remarks in Section 6.

2 RELATED WORK

This work is related to deep neural networks and semantic representation for text understanding.

Recently, a large number of CNNs and RNNs with potential benefits have attracted many researchers' attention. Existing efforts mainly focus on the application of LSTM [19, 38], GRU [6, 7], SRU [41], and CNNs [12, 23, 24] based on word embeddings [33, 35] drawing on the idea of either language model [3, 34] or spatial parameter sharing. And all these models have demonstrated impressive results in NLP applications. Many previous works have shown that the performance of deep neural networks can be improved by attention mechanism [2, 17]. In addition, the self-attention mechanism with position embedding characterizes the mutual relationship between one and others as a dependency to capture the semantic encoding information [43]. There are some other works that combine RNN and CNN for text classification [46, 50] or use a hierarchical structure for language modeling [32, 47]. Besides hybrid neural networks, graph based models [22, 37] and human behavior enhanced models [16, 18] are widely employed to capture textural semantics.

Recently, sarcasm detection, which is an important part of the implied semantic recognition, is widely studied by linguistic researchers [4, 5, 13, 21, 29]. [9] proves that it is important to consider several valuable expressive forms to capture the sentiment orientation of the messages. And external sentiment analysis resources are beneficial to sarcasm detection [49]. Furthermore, [42] realizes a neural network to represent a sentence by comparing word-to-word embeddings which achieves state-of-the-art performance. More specifically, an intra-attention mechanism allows their model to

search for conflict sentiments as well as maintain compositional information. For cyberbullying, [28] describes a close analysis of the language usage, identifies the most commonly used cyberbullying terms, and develops queries that can be used to detect cyberbullying content. [1] proposes a deep learning based model to detect cyberbullying across multiple social media platforms.

However, all these approaches mentioned above don't specifically make good use of the contextual representation as a straightforward input when understanding the implied meaning and they never worry about the possible noise such as special figurative phrases in the training data.

3 SKIM AND INTENSIVE READING MODEL

In this section, we propose a novel deep neural network inspired by reading comprehension procedure of people, namely Skim and Intensive Reading Model (SIRM), to address the essential issues for understanding texts with implied meanings. The architecture of the model is depicted in Figure 4.

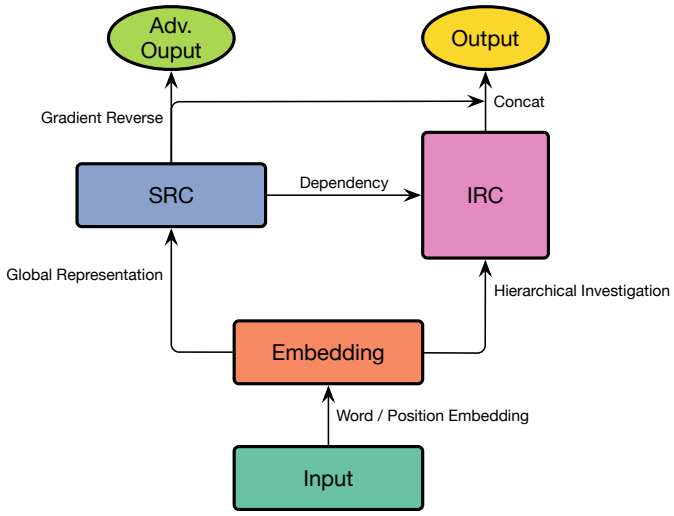

Figure 4: The architecture of the proposed Skim and Intensive Reading Model (SIRM), mainly including word embedding layer, position embedding, a skim reading component, an intensive reading component, a normal loss, and an adversarial loss. Each part of the SIRM is designed under the Occam's razor.

3.1 Overview

People always consume a difficult text word by word and sentence by sentence with respect to the global information extracted by reading quickly. Besides the input layer and the embedding layer, the SIRM consists of three main parts which are the skim reading component (SRC) associated with the adversarial loss part and the intensive reading component (IRC) to simulate the procedure of human reading comprehension. And for efficiency concern, the model is designed under the Occam's razor, which means we use the simplest and minimal component to realize each part of the SIRM. More specifically, the SRC is a set of shallow CNNs to enable global feature extraction, while the IRC is a hierarchical framework to enhance the contextual information, from sentence level to paragraph level. Finally, over the output layer, common cross entropy and an adversarial loss are utilized to represent the cost function of the end-to-end deep neural network.

3.2 Input

Each example of this task is represented as a set (p, y), where input $p = [s_1, \cdots, s_m]$ is a paragraph with m sentences, $s_i = [w_{i,1}, \cdots, w_{i,n}]$ is $i-$th sentence in paragraph p with n words, and $y \in Y$ where $Y = \{0, 1\}$ is the label representing the category of p. We can represent the task as estimating the conditional probability $Pr(y|p)$ based on the training set, and identifying whether a testing example belongs to the target class by $y' = argmax_{y \in Y} Pr(y|p)$.

3.3 Word Embedding

The goal of word embedding layer is to represent $j-$th word $w_{i,j}$ in sentence s_i with a d_e dimensional dense vector $\boldsymbol{x}_{ij} \in \mathbb{R}^{d_e}$. Given an input paragraph p, it will be represented as $\boldsymbol{P} = [\boldsymbol{S}_1, \cdots, \boldsymbol{S}_m] \in \mathbb{R}^{m \times n \times d_e}$, where each representation of sentence is a matrix $\boldsymbol{S}_i = [\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,n}] \in \mathbb{R}^{n \times d_e}$ consisting of word embedding vectors of $i-$th sentence.

3.4 Position Embedding

Position information can be potentially important for text understanding. In the SIRM, two types of position information are encoded, word position in a sentence, and sentence position in a paragraph. By leveraging the position encoding method from [43], word/sentence positions are captured via sine and cosine functions of different frequencies to the input embeddings. Furthermore, the positional encodings have the same dimension as the corresponding embedding matrix, so that the results can be easily aggregated. The mathematical formulas are shown as follows:

$$\begin{aligned} \boldsymbol{U}_{pos,2i} &= sin(pos/10000^{2i/d_e}) , \\ \boldsymbol{U}_{pos,2i+1} &= cos(pos/10000^{2i/d_e}) , \end{aligned} \tag{1}$$

where pos is the position and i denotes the dimension. Moreover, for any fixed offset k, $\boldsymbol{U}_{pos+k}$ can be represented as a sinusoidal function of $\boldsymbol{U}_{pos}$.

After that, we add corresponding position embedding matrix $\boldsymbol{U}_{1:n}$ to each sentence embedding matrix $\boldsymbol{S}_i$:

$$\boldsymbol{S}'_i = \boldsymbol{S}_i + \boldsymbol{U}_{1:n} = [\boldsymbol{x}'_{i,1}, \boldsymbol{x}'_{i,2}, \cdots, \boldsymbol{x}'_{i,n}] . \tag{2}$$

3.5 Skim Reading Component (SRC)

Since each word and sentence with implied textual meaning can be highly dependent on the contextual information, the proposed model needs to characterize the dynamic entire representation of given input in a quick manner like human reading shown in Figure 5.

A tailored CNN employs three key functions, e.g., sparse interaction, parameter sharing, and equivariant representation [46], which can encode the partial spatial information. Hence, in the SRC, we use CNN layers with different window sizes in order to extract features like n-gram. Given a paragraph embedding $\hat{\boldsymbol{P}} \in \mathbb{R}^{m \cdot n \times d_e}$

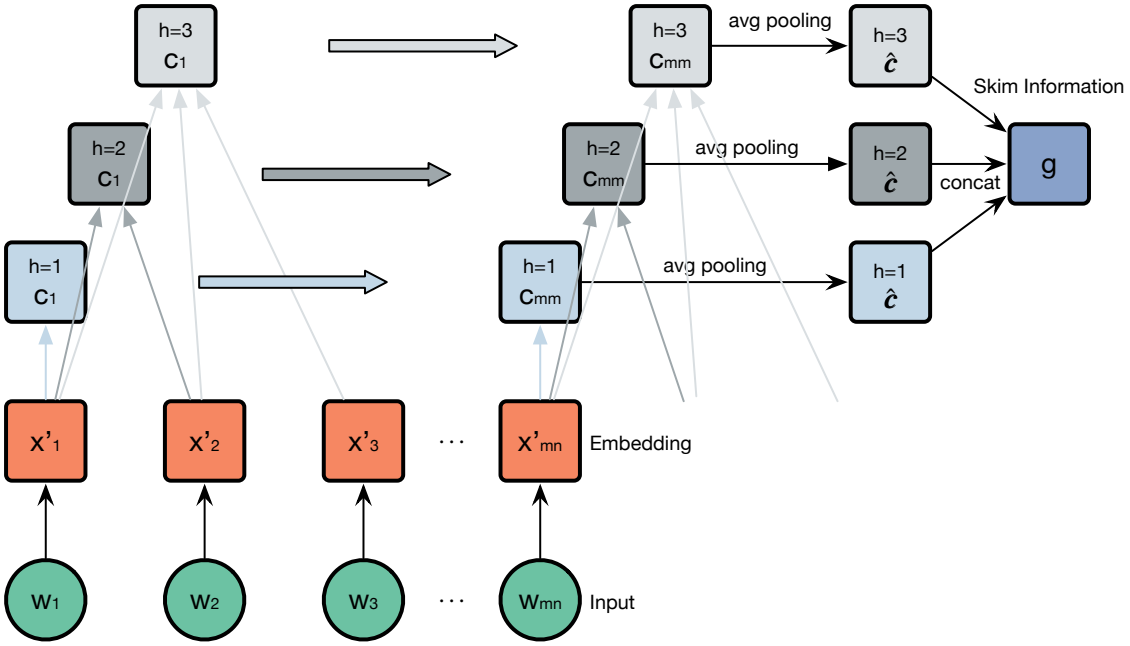

Figure 5: The SRC characterizes the entire information via convolutional neural networks with different kernel/window size.

reshaped from $\boldsymbol{P}$, the global feature is extracted as follows:

$$\boldsymbol{g} = SRC(\hat{\boldsymbol{P}}) \,. \tag{3}$$

More specifically, d_c convolution filters are applied to a window of h words to produce a corresponding local feature. For example, a feature $\boldsymbol{c}_i \in \mathbb{R}^{d_c}$ is generated from a window of words $\hat{\boldsymbol{P}}_{i:i+h-1}$:

$$\boldsymbol{c}_i = ReLU(\boldsymbol{W}_c * \hat{\boldsymbol{P}}_{i:i+h-1} + \boldsymbol{b}_c) \,, \tag{4}$$

where $*$ denotes the convolutional operation and the feature map from filter with the same shape is represented as $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_{m \cdot n - h + 1}]$, $\boldsymbol{W}_c \in \mathbb{R}^{h \times d_e \times d_c}$ is the weight matrix and $b_c \in \mathbb{R}^{d_c}$ is the bias.

We then apply an average-over-time pooling operation over the feature map and obtain the feature as follows:

$$\hat{\boldsymbol{c}} = \frac{1}{m \cdot n - h + 1} \sum_{i}^{m \cdot n - h + 1} \boldsymbol{c}_i \,. \tag{5}$$

In this part, we utilize filters with c kinds of window size to extract more accurately relevant information by taking the consecutive words (e.g., n-gram) into account, and then concatenate all $\hat{\boldsymbol{c}}$ from these filters to get the global semantic feature mentioned above which is represented as $\boldsymbol{g}$, where $\boldsymbol{g} \in \mathbb{R}^{c \cdot d_c}$.

3.6 Intensive Reading Component (IRC)

Inspired by the human reading comprehension procedure, the IRC employs a hierarchical framework to characterize and explore the implied semantic information from sentence level to paragraph level. In other words, the sentence encoding outcomes will be used as the input of the paragraph-level part. The structure of IRC is shown in Figure 6.

For sentence-level part (IRC_S), given i−th sentence embedding $\boldsymbol{S}'_i$ from embedding layer with position embedding and the global

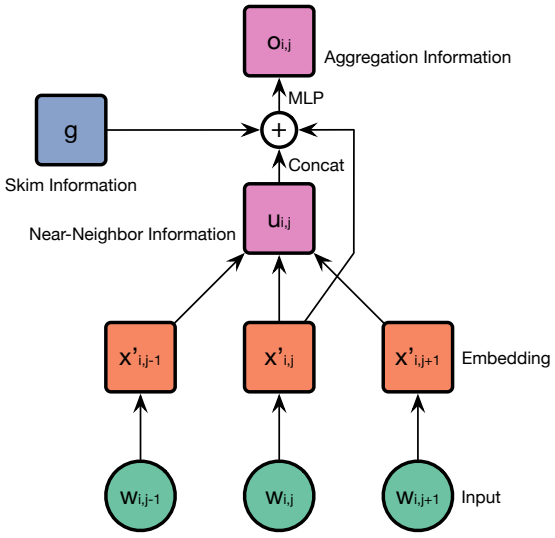

Figure 6: The IRC encodes the current embedding, the near-neighbor information, and the skim information with a dense connection.

information $\boldsymbol{g}$ from the SRC, the sentence encoding information is extracted as a vector shown below:

$$\boldsymbol{o}_i = IRC_S(\boldsymbol{S}'_i, \boldsymbol{g}) \,, \tag{6}$$

and the paragraph embedding information is represented as a matrix: $\boldsymbol{O} = [\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_m]$.

Before paragraph-level model (IRC_P), a corresponding position embedding matrix $\boldsymbol{U}$ is added to $\boldsymbol{O}$:

$$\boldsymbol{O}' = \boldsymbol{O} + \boldsymbol{U}_{1:m} \ . \tag{7}$$

Then, the paragraph is encoded as a vector shown below:

$$\boldsymbol{o}_P = IRC_P(\boldsymbol{O}', \boldsymbol{g}) \ . \tag{8}$$

Note that both IRC_S and IRC_P share the same structure, but the trainable parameter values are quite different. The detailed component descriptions can be found as follows.

3.6.1 Near-Neighbor Information Encoder. For people, in order to understand the implied meaning of the current word/sentence, besides the entire information of the whole paragraph, the near-neighbor information around the word/sentence, in a size $2 \cdot k + 1$ word/sentence window, also plays an important role in characterizing the contextual information of the target word/sentence.

Hence, we pad both k words/sentences at the head and tail for input sentence embedding $\boldsymbol{S}'_i$ or paragraph embedding $\boldsymbol{O}'$, respectively. Taking the sentence-level part as an example, d_{ns} filters, with window size $2 \cdot k + 1$, are applied to produce the near-neighbor information. So, the near-neighbor information of j−th word in i−th sentence is represented as a vector $\boldsymbol{u}_{ij} \in \mathbb{R}^{d_{ns}}$:

$$\boldsymbol{u}_{i,j} = f(\boldsymbol{W}_{ns} * \boldsymbol{S}'_{i,j-k:j+k} + \boldsymbol{b}_{ns}) \ , \tag{9}$$

where $\boldsymbol{W}_{ns} \in \mathbb{R}^{(2 \cdot k+1) \times d_e \times d_{ns}}$ denotes the weight matrix and $b_{ns} \in \mathbb{R}^{d_{ns}}$ is the bias.

Finally, the near-neighbor information of all words in i−th sentence is encoded as a matrix: $\boldsymbol{U}_i = [\boldsymbol{u}_{i,1}, \boldsymbol{u}_{i,2}, \cdots, \boldsymbol{u}_{i,n}]$. The near-neighbor information is an important part of contextual information for the current word.

3.6.2 Dense Connection. To comprehensively understand implied semantics of a given text, the main effort of this work is to take advantage of the contextual information as guidance and dependency on each word/sentence like people always do. Hence, inspired by [20], the most direct idea is to concatenate the entire information (the skim information), the near-neighbor information, and the pure word/sentence embedding, and then feed them into a Multilayer Perceptron (MLP), also named dense connection layer, to realize an aggregate encoding. Taking the sentence-level part as an example, the aggregate encoding is achieved as below:

$$\begin{aligned} \boldsymbol{t}_{i,j} &= [\boldsymbol{g} \oplus \boldsymbol{u}_{i,j} \oplus \boldsymbol{x}'_{i,j}] \ , \\ \boldsymbol{o}_{i,j} &= relu(\boldsymbol{W}_{ls} \cdot \boldsymbol{t}_{i,j} + \boldsymbol{b}_{ls}) \ , \end{aligned} \tag{10}$$

where $\oplus$ is the concatenation operation, $\boldsymbol{W}_{ls} \in \mathbb{R}^{d_{as} \times (c \cdot d_c + d_{ns} + d_e)}$ denotes the weight matrix, and $\boldsymbol{b}_{ls} \in \mathbb{R}^{d_{as}}$ represents the bias.

Eventually, i−th sentence from sentence-level model is encoded as $\boldsymbol{o}_i \in \mathbb{R}^{d_{as}}$:

$$\boldsymbol{o}_i = \frac{1}{n} \sum_{j}^{n} \boldsymbol{o}_{i,j} \ . \tag{11}$$

For paragraph-level IRC, the outputs from the near-neighbor information encoder and the aggregate encoder are represented as $\boldsymbol{U}_P \in \mathbb{R}^{m \times d_{np}}$ and $\boldsymbol{o}_P \in \mathbb{R}^{d_{ap}}$, respectively.

Note that, a gate mechanism [6] could replace the dense connection, and an attention mechanism [47] could replace the last average pooling. The results of comparison are shown in Figure 9.

3.7 Output

Undertaking the paragraph encoding $\boldsymbol{g}$ and $\boldsymbol{o}_P$ from the SRC and IRC respectively, a Multilayer Perceptron (MLP) is applied to generate the output y':

$$y' = \sigma(\boldsymbol{W}_o \cdot (\boldsymbol{o}_P \oplus \boldsymbol{g}) + \boldsymbol{b}_o) \ , \tag{12}$$

where $\boldsymbol{W}_o \in \mathbb{R}^{d_{ap} + c \cdot d_c}$ denotes the weight matrix and b_o represents the bias.

Here, the output y' is the probability of the target category.

3.8 Model Training with Adversarial Learning

In the SIRM, the skim information $\boldsymbol{g}$ is extracted from a set of shallow CNNs. Because this feature is similar to n-gram instead of deep semantic representation, it can be polluted by noisy information such as special phrases highly related to the training data, e.g., some special figurative phrases.

Hence, the proposed model should be able to penalize the features strongly associated with the training data, while the general features should be boosted for the IRC optimization.

In this study, we implement this idea by utilizing an adversarial learning mechanism when training the model. For more theoretical details, refer to [10, 11, 15, 36]. Specifically, we add a MLP over the SRC shown as follows:

$$y'' = softmax(\boldsymbol{W}_g \cdot \boldsymbol{g} + \boldsymbol{b}_g) \ , \tag{13}$$

where $\boldsymbol{W}_g \in \mathbb{R}^{2 \times c \cdot d_c}$ denotes the weight matrix and $b_g \in \mathbb{R}^2$ represents the bias.

Since the n-gram based global feature tends to be overfitting during the training procedure, we expect $\boldsymbol{g}$ to have a bit low performance when directly connecting to the output.

In a word, the final loss needs to minimize the normal loss and maximize the adversarial learning based loss, which is represented as:

$$\zeta(y, y', y'') = min\zeta(y, y') + max\lambda \cdot \zeta_{adv}(y, y'') \ , \tag{14}$$

where both ζ and ζ_{adv} are the negative log likelihood and λ is an adjustment factor which is far less than 1. In addition, ζ_{adv} is named as adversarial loss (Adv) in this paper.

The SIRM is an end-to-end deep neural network, which can be trained by using stochastic gradient descent (SGD) methods, such as Adam [27]. More implementation details will be given in the experiments section 4.

4 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the proposed SIRM against baseline models and several variants of SIRM in terms of performance, robustness, and efficiency. As a byproduct of this study, we release the codes and the hyper-parameter settings to benefit other researchers[3].

[3] https://github.com/GuoxiuHe/SIRM

Table 1: Statistics for all datasets: l is the length of text and +/- is the proportion of positive and negative samples.

Name	Train Size	Test Size	Total Size	Max l	Min l	Avg l	+/-
Industry/spam	20,609	6,871	27,480	3,447	149	393	1/3
Tweets/ghosh	50,736	3,680	54,416	56	6	17	1/1
Reddit/movies	13,535	1,504	15,039	129	6	13	1/1
IAC/v1	1,483	371	1,854	1,045	6	57	1/1

4.1 Datasets

In order to validate the performance of the proposed SIRM and make it comparable with alternative baseline models, we conduct our experiments on one real-world industrial (Alibaba Literature[4]) spam detection dataset about novel involving implied pornography and three publicly available benchmark datasets about sarcasm detection. Details for all datasets are summarized in Table 1 and described as below:

- Industrial Novel Dataset Involving Implied Pornography: We evaluate the performance of the proposed SIRM on a Chinese online novel collection about spam detection in novels involving implied pornography. The pornographic novels are firstly complained/reported by readers, e.g., parents of children/teenagers, and then confirmed by auditors. Note that the authors of these novels may purposely avoid using explicit and sensitive words instead of figurative words because of the censorship.
- Tweets/ghosh[5]: Ghosh and Veale [13, 14] collected a sarcasm dataset from tweets which is the the world's biggest microblogging platform. The labels are hash tags e.g. 'sarcasm', 'sarcastic', and 'ironie'.
- Reddit/movies[6]: This sarcasm dataset is collected by [25] from Reddit, which is one of the world's largest online communities. The labels are annotated with the '/s' tag left by authors themselves. In our experiments, we choose the subset from the subreddit '/r/movies'.
- IAC/v1[7]: We use a sarcasm dataset collected from Internet Argument Corpus (IAC) by [44]. There are two versions and we choose the IAC-V1.

4.2 Baselines

We employ the following baseline models (also see Table 2) for comparison, including word embedding [35] based shallow neural networks, deep learning based models, and recent state-of-the-art models:

NBOW [39]: is a simple model based on word embeddings with average pooling.

CNN [26]: is a simple CNN model with average pooling using different kernels. There are 7 kinds of filters whose widths are from 1 to 7 and each has 100 different ones.

LSTM [19]: is a vanilla Long Short-Term Memory Network. We set the LSTM dimension to 100.

Atten-LSTM [47]: is a LSTM applying an attention mechanism. The dimension is set to 100.

GRNN [48]: employs a gated pooling method and a standard pooling method to extract content features and contextual features respectively from a gated recurrent neural network. This model has been demonstrated improvement compared to feature engineering based traditional models for sarcasm detection.

SIARN and **MIARN** [42]: capture incongruities between words with an intra-attention mechanism. A single-dimension intra-attention and a multi-dimension one are employed by SIARN and MIARN respectively. Both of them are the state-of-the-art models for sarcasm detection. We use the default settings by the authors.

Self-Atten [43]: is the state-of-the-art model from Google to encode deep semantic information using self-attention mechanism[8]. For the feasibility of training because of the large scale parameters, we set all dimensions as 64 just like ours and other hyper-parameters are the same as given settings.

4.3 Evaluation Metrics

We choose to report parameter size (Param) and running time (Time) for evaluating the efficiency of the proposed SIRM against all baseline models. More specifically, the unit of the parameter size is thousand. And then, the whole running time of the NBOW is selected as the unit of Time. For effectiveness, we select Macro-Averaged F1 Score to show the performance for the three benchmark datasets (label-balanced) and employ F1 score for the industry dataset (label-unbalanced). In addition, we report accuracy for all of datasets.

4.4 Experiment Settings

For experiment fairness, we exploit the same data preprocessing as [42]. For the SIRM, the number of convolution filters d_c in the SRC is 16 and the window size h is from 1 to 4. The near-neighbor size k is 1. The dimension d of all other layers are all set to 64. The adjustment factor λ for adversarial loss is 1×10^{-6}. In addition, the learning rate is 1×10^{-3} and the batch size is 64. For Chinese novel dataset, we use *JIEBA*[9] for tokenization. Furthermore, the statistical significance is conducted via the t-test with p-value$< 10^{-3}$.

5 RESULTS AND ANALYSIS

In this section, we give detailed data analysis, experimental results, and analysis to show insights into the proposed SIRM comparing with other baselines.

5.1 Data Analysis

We visualize the word distribution in industry implied pornographic data as shown in Figure 7 in order to show the characteristic of the dataset and the challenge in the perspective of word level. X-axises

[4]https://www.aliwx.com.cn/
[5]https://github.com/AniSkywalker/SarcasmDetection
[6]http://nlp.cs.princeton.edu/SARC/0.0/
[7]https://nlds.soe.ucsc.edu/sarcasm1
[8]https://github.com/tensorflow/models/tree/master/official
[9]https://github.com/fxsjy/jieba

Table 2: Experimental results of performance comparison.

	Industry/spam		Tweets/gosh		Reddit/movies		IAC/v1	
Model	Macro F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy
NBOW	84.96%	92.25%	72.42%	69.37%	68.50%	68.18%	61.32%	59.61%
CNN	85.46%	92.42%	74.84%	74.54%	65.50%	65.03%	60.98%	58.40%
LSTM	82.16%	90.86%	75.08%	75.16%	67.71%	66.74%	44.73%	53.84%
Atten-LSTM	83.74%	91.40%	75.15%	73.73%	65.20%	63.84%	61.80%	60.46%
GRNN	86.30%	93.04%	79.43%	79.24%	64.59%	63.19%	52.45%	54.78%
SIARN	77.73%	92.91%	78.84%	79.59%	67.50%	68.17%	60.86%	61.33%
MIARN	86.14%	92.25%	72.71%	72.31%	63.44%	62.12%	55.74%	58.95%
Self-Atten	86.99%	93.48%	76.01%	75.19%	66.29%	65.47%	61.32%	60.12%
SIRM	**88.18%***	**93.94%***	**82.54%***	**82.38%***	**70.01%***	**69.94%***	**63.01%***	**62.13%***

Table 3: Experimental results of efficiency comparison.

Model	Param	Time
NBOW	**10.3**	**1**
CNN	30.3	2
LSTM	60.6	18
Atten-LSTM	71.0	22
GRNN	131.0	33
SIARN	100.9	150
MIARN	102.3	180
Self-Atten	254.9	17
SIRM	63.7	2

are top 100 words in the whole corpus. And Y-axis is proportion of each word. Here, we can see that words in spam corpus share similar distribution like words in normal corpus. This evidence may interpret why word-based models don't work well to understand implied meaning.

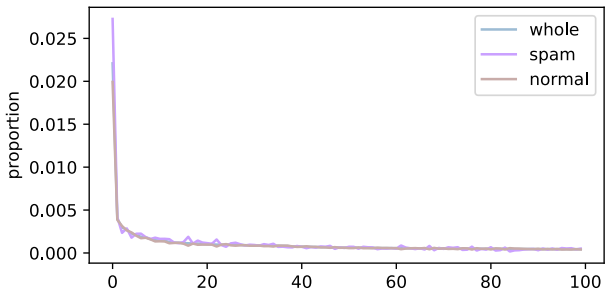

Figure 7: Visualization of word distribution.

5.2 Performance Comparison

The parameter size, the running time and the performance of the SIRM compared with baseline models are shown in Table 2 and Table 3.

NBOW realizes a decent performance for all datasets, especially for Reddit/movies. More importantly, NBOW has the lowest parameter size and achieves the least time cost. That means the NBOW can be a good choice in the vast majority of cases, also demonstrated by [8, 39].

Unfortunately, the standard text representation models, such as CNN, LSTM, Atten-LSTM, and GRNN, don't outperform NBOW significantly. And they can not even achieve a stable performance across all datasets because of the lack of the training data. For example, the GRNN performs well on Tweets/gosh and Industry/spam, but works worse on Reddit/movies and IAC/v1. The state-of-the-art models SIARN, MIARN, and Self-Atten don't perform well as expected in this work intuitively. With more parameters and running time cost, these models may be even worse than NBOW. Moreover, RNN based models take more time than other models.

The proposed SIRM significantly outperforms all the baseline models according to accuracy and F1 score. It is clear that the proposed SIRM, along with SRC, IRC, and Adv, can be more stable on all datasets which have diverse data sizes and text lengths, with the architecture specially designed to simulate human's reading comprehension procedure. Furthermore, other recent advanced models do not perform well due to the indifference of contextual information (for the word/sentence) and the bad impact of figurative expression.

For example, SIRM makes it to identify *[Tweets/ghosh: sarcasm] 'Do you know what I love? Apartment construction at 7 a.m. 3 mornings in a row!'*, but SIARN and Self-Atten fail to do so. The reason may be that words in the last two sentences look irrelevant or not explicitly contradictory to the first sentence, which will mislead the two models. But, the SIRM can capture the real meaning by reading each word/sentence with the global knowledge.

It's worth mentioning that the parameter size and running time cost of the SIRM is comparable with all baselines. That is because we haven't used any recurrent unit which means the SIRM can be totally parallel during training and inference by using a small scale of GPU memory.

5.3 Parameter Size Sensitivity

As shown in Figure 8, parameter size sensitivity of the proposed SIRM against other baseline models is investigated based on Tweets/gosh. It is obvious that the proposed SIRM outperforms all representative baseline models, especially the tailored and state-of-the-art model for sarcasm detection, SIARN, according to F1 score and accuracy with all alternative parameter sizes. In contrast, Self-Atten achieves a lower score even than NBOW at the lowest dimension setting while the NBOW reaches the most stable performance among

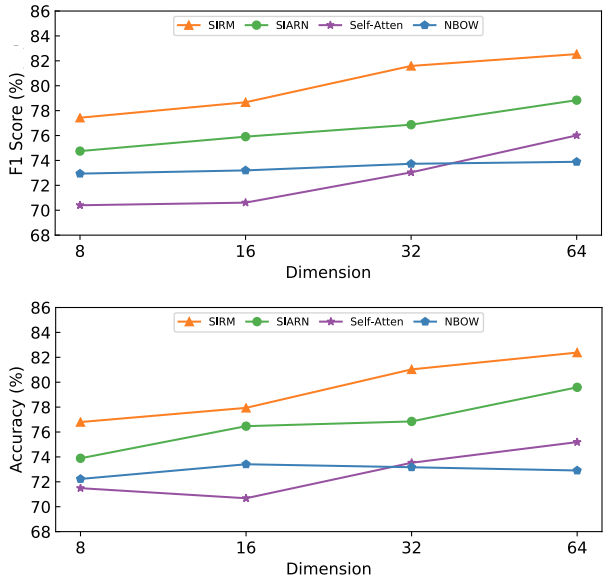

Figure 8: Results of comparison for parameter size sensitivity: x-axis is main dimension of the model.

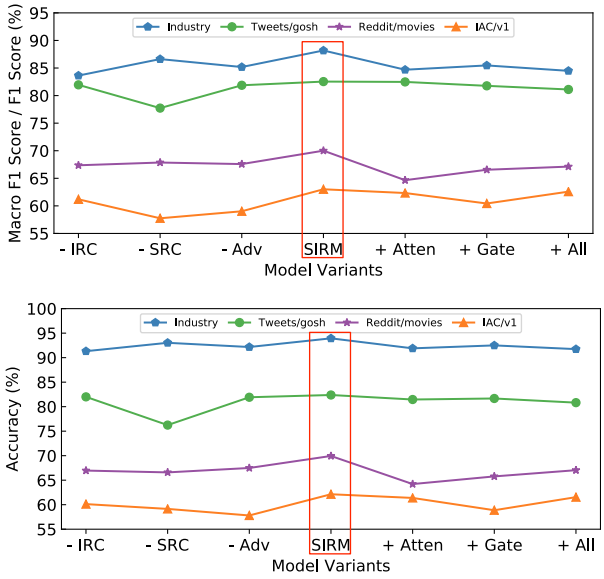

Figure 9: The performance of ablation and addition for the SIRM: - denotes the ablation and + denotes the addition.

all alternative dimension settings, which means there is no significant improvement with the increase of dimension. The tailored and state-of-the-art model for sarcasm detection, SIARN, realizes a good performance but can also be improved significantly.

In a word, all these evidences demonstrate the robustness and superiority of the proposed SIRM.

5.4 Ablation and Addition of SIRM

For efficiency purposes, we design each part of the SIRM with respect to Occam's razor. Hence, we investigate the impact of the complexity of the SIRM shown in Figure 9. By removing each part, such as IRC, SRC, and Adv, there is a decrease compared with the SIRM. That is because each part of the SIRM plays a different, necessary, and important role for implied semantic meaning understanding across different datasets.

Meanwhile, we find that using a more sophisticated component to replace the simple one is not advisable. Gate and attention mechanisms don't make performance increase. In particular, the more complex of the component, the more space and running time it will take.

Hence, taking the model complexity into consideration, the proposed SIRM realizes the best performance with the simplest implementation.

6 CONCLUSION

In this study, we propose a novel model, namely Skim and Intensive Reading Model (SIRM), for understanding and identifying the implied textual meaning in a quick manner. In SIRM, the SRC is designed to capture the dynamic global information, while the IRC is employed to characterize the fine semantics via a hierarchical framework by taking the contextual information and local features into consideration with the dense connection. In addition, the adversarial loss is applied over the SRC to eliminate the potential noise in training set. We conduct extensive experiments on an industrial spam dataset of novels involving implied pornography and several sarcasm benchmarks. The data analysis provides insights into the challenges of this task from the word viewpoint. And the results indicate that the proposed model practically outperforms all alternative baselines and ablation and addition variants of the SIRM, in the light of performance, robustness, and efficiency.

In the future, we will investigate and study a simpler and advantageous model to cover more scenarios involving implied textual meaning.

7 ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their insightful feedback and comments. This work is supported by the National Natural Science Foundation of China (71673211, 71704137, 61876003), Guangdong Basic and Applied Basic Research Foundation (2019A1515010837), and a scholarship from the China Scholarship Council (201906270034).

REFERENCES

[1] Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*. Springer, 141–153.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations* (2015), 1–15.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[4] Elisabeth Camp. 2012. Sarcasm, pretense, and the semantics/pragmatics distinction. *Noûs* 46, 4 (2012), 587–634.

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

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1724–1734.

[7] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*. 1–9.

[8] Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2126–2136.

[9] Elisabetta Fersini, Enza Messina, and Federico Alberto Pozzi. 2016. Expressive signals in social media languages to improve polarity detection. *Information Processing & Management* 52, 1 (2016), 20–35.

[10] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37. PMLR, 1180–1189.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[12] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th international conference on machine learning (ICML-17)*. 1243–1252.

[13] 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*. 161–169.

[14] Aniruddha Ghosh and Tony Veale. 2017. Magnets for sarcasm: making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 482–491.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[16] Guoxiu He, Yangyang Kang, Zhe Gao, Zhuoren Jiang, Changlong Sun, Xiaozhong Liu, Wei Lu, Qiong Zhang, and Luo Si. 2019. Finding Camouflaged Needle in a Haystack? Pornographic Products Detection via Berrypicking Tree Model. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 365–374.

[17] Guoxiu He and Wei Lu. 2018. Entire Information Attentive GRU for Text Representation. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, 163–166.

[18] Guoxiu He, Yunhan Yang, Zhuoren Jiang, Yangyang Kang, Xiaozhong Liu, and Wei Lu. 2020. Implicit Products in the Decentralized eCommerce Ecosystems. In *JCDL*. ACM.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[21] Stacey L Ivanko and Penny M Pexman. 2003. Context incongruity and irony processing. *Discourse Processes* 35, 3 (2003), 241–279.

[22] Zhuoren Jiang, Zhe Gao, Guoxiu He, Yangyang Kang, Changlong Sun, Qiong Zhang, Luo Si, and Xiaozhong Liu. 2019. Detect Camouflaged Spam Content via StoneSkipping: Graph and Text Joint Embedding for Chinese Character Variation Representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6188–6197.

[23] Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 562–570.

[24] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Volume 1: Long Papers* (2014), 655–665.

[25] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A Large Self-Annotated Corpus for Sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. 641–646.

[26] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. 1746–1751.

[27] Diederik P Kingma and Jimmy Ba. [n. d.]. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*. 1–15.

[28] April Kontostathis, Kelly Reynolds, Andy Garron, and Lynne Edwards. 2013. Detecting cyberbullying: query terms and techniques. In *Proceedings of the 5th annual acm web science conference*. 195–204.

[29] Florian Kunneman, Christine Liebrecht, Margot Van Mulken, and Antal Van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management* 51, 4 (2015), 500–509.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[31] Zhuohan Li, Di He, Fei Tian, Wei Chen, Tao Qin, Liwei Wang, and Tieyan Liu. 2018. Towards Binary-Valued Gates for Robust LSTM Training. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 2995–3004. http://proceedings.mlr.press/v80/li18c.html

[32] Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 899–907.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[34] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, Vol. 2. 3.

[35] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[36] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*. 1–15.

[37] Shirin Noekhah, Naomie binti Salim, and Nor Hawaniah Zakaria. 2020. Opinion spam detection: Using multi-iterative graph-based model. *Information Processing & Management* 57, 1 (2020), 102140.

[38] Karl Pichotta and Raymond J Mooney. 2016. Using sentence-level LSTM language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers*. 279–289.

[39] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers*. 440–450.

[40] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *International Conference on Learning Representations*. 1–14.

[41] Sida I. Wang Hui Dai Tao Lei, Yu Zhang and Yoav Artzi. 2018. Simple Recurrent Units for Highly Parallelizable Recurrence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4470–4481.

[42] Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with Sarcasm by Reading In-Between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1010–1020.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[44] Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A Corpus for Research on Deliberation and Debate.. In *LREC*. Istanbul, 812–817.

[45] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *The Annual Meeting of the Association for Computational Linguistics*. 1288–1297.

[46] Chenglong Wang, Feijun Jiang, and Hongxia Yang. 2017. A hybrid framework for text modeling with convolutional RNN. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2061–2069.

[47] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[48] Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers*. 2449–2460.

[49] Shiwei Zhang, Xiuzhen Zhang, Jeffrey Chan, and Paolo Rosso. 2019. Irony detection via sentiment-based transfer learning. *Information Processing & Management* 56, 5 (2019), 1633–1644.

[50] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630* (2015).