Hindawi
Complexity
Volume 2020, Article ID 2509018, 9 pages
https://doi.org/10.1155/2020/2509018

Research Article

Phonetics and Ambiguity Comprehension Gated Attention Network for Humor Recognition

Xiaochao Fan,[1,2] Hongfei Lin ,[1] Liang Yang,[1] Yufeng Diao ,[1,3] Chen Shen,[1] Yonghe Chu ,[1] and Tongxuan Zhang[1]

[1]*School of Computer Science and Technology, Dalian University of Technology, Dalian, China*
[2]*School of Computer Science and Technology, Xinjiang Normal University, Urumqi, China*
[3]*School of Physics and Electronic Engineering, Inner Mongolia University for Natinalities, Tongliao, China*

Correspondence should be addressed to Hongfei Lin; hflin@dlut.edu.cn

Received 2 November 2019; Revised 30 January 2020; Accepted 26 February 2020; Published 29 April 2020

Academic Editor: Danilo Comminiello

Humor refers to the quality of being amusing. With the development of artificial intelligence, humor recognition is attracting a lot of research attention. Although phonetics and ambiguity have been introduced by previous studies, existing recognition methods still lack suitable feature design for neural networks. In this paper, we illustrate that phonetics structure and ambiguity associated with confusing words need to be learned for their own representations via the neural network. Then, we propose the Phonetics and Ambiguity Comprehension Gated Attention network (PACGA) to learn phonetic structures and semantic representation for humor recognition. The PACGA model can well represent phonetic information and semantic information with ambiguous words, which is of great benefit to humor recognition. Experimental results on two public datasets demonstrate the effectiveness of our model.

1. Introduction

Humor is frequently used in daily communication [1]. When interacting with people, if artificial intelligence (AI) systems, such as chatbots, can detect humor within the conversation, it will help them better understand the emotions of the human and help the AI make more appropriate decisions. Therefore, humor computation deserves particular attention, as it has the potential to turn computers into creative and motivational tools for human activity [2].

Humor recognition refers to determining whether a sentence in a given context expresses a certain degree of humor. Yang et al. [3] identified three semantic structures and a phonetic structure behind humor. Experimental results show that ambiguity and phonetic structures are important for humor recognition.

Phonetic structures, used as devices in humorous texts, usually take the form of alliteration or rhyme. Alliteration, rhyme, or word repetition are often used to evoke or enhance the effect of humor even if the content is not humorous.

Exp 1. “You can tune a piano, but you can’t tuna fish.”

In Exp 1, the humor does not come from the content of the sentence, but the words “tune” and “tuna” have the same pronunciation, which produces a comic effect. Hence, it shows that phonetic structures, such as alliteration, rhyme, and word repetition, play an important role in humorous texts.

Ambiguity [4] refers to some words with multiple meanings in a sentence causing different sentence comprehensions. Ambiguity and humor often go together [5], and it is a crucial component of many humorous texts [6].

Exp 2. “Did you hear about the guy whose whole left side was cut off? He’s all right now.”

Exp 2 shows humor caused by ambiguity. The word “right” is the ambiguous word, meaning “right side” or “okay”.

For the detection of phonetic structures and ambiguity in a humorous text, the most popular methods are based on complex feature engineering, such as semantic similarity and the number of rhyme chains. The idea of feature engineering is simple, but it is time consuming and cannot easily capture the latent semantic information behind humor. Recently, due to strong feature extraction capabilities, neural network-based approaches have become mainstream for this task. However, most researchers simply use the deeper neural network without modeling phonetic structure and ambiguity. Moreover, it is difficult to analyze the results of humor recognition.

To solve this problem, we propose an end-to-end neural network named Phonetics and Ambiguity Comprehension Gated Attention network to detect humor in text. The proposed model captures the phonetic information by Convolutional Neural Networks (CNN), combines with Bidirectional Gated Recurrent Units (Bi-GRU) and attention mechanism to build the information of context and ambiguous words, and applies gated mechanism to adjust the effects of the two kinds of information in the task of humor recognition. Our work makes three contributions:

(1) For solving phonetic structure and ambiguity features in humor recognition, we propose a novel framework named Phonetics and Ambiguity Comprehension Gated Attention network (PACGA), which can understand the phonetic representation by the CNN model, and learn latent semantic representation associated with ambiguous words by Bi-GRU and attention mechanism.

(2) We propose the gated attention strategy to exploit the combination of the phonetic structure and ambiguity in the humor recognition. Experimental results show that it is useful for humor recognition.

(3) Experimental results on the pun-of-the-day [3] and One liners 16000 [7] datasets demonstrate that our method achieves state-of-the-art performance compared with strong baselines. Furthermore, the detailed analysis reveals the interpreting ability of our proposed model in humor recognition.

1.1. Related Work. In this section, we will review related works on machine learning-based methods and deep learning-based methods for humor recognition.

Machine learning-based methods have been widely used to detect humor in text, which usually depends on feature extraction from text to train classifiers. Mihalcea and Strapparava [8] brought empirical evidence that computational methods can be successfully applied to the task of humor recognition in text. Zhang and Liu [9] designed about fifty features of five categories derived from influential humor theories, linguistic norms, and affective dimensions. Barbieri and Saggion [10] proposed a rich set of features, including ambiguity and phonetic structure. In recent work, Liu and Zhang [11] modeled sentiment association between discourse units to detect humor. They found that some syntactic structure features consistently correlated with humor in a separate paper [12]. Most of the abovementioned experimental results show that phonetic structure and ambiguity are primary features in humor recognition. However, the cost of constructing a large number of features is high and it also limits the generalization capability of the model.

Recently, deep learning-based methods have garnered considerable success in humor recognition. Bertero and Fung [13] combined word-level and audio frame-level features and used RNN and CNN to predict humorous utterances. In their other paper [14], CNN was used to encode utterances, and then Bi-LSTM was used to predict humor in dialogues [15]. Systematically, the performance of humor recognition based on CNN was compared with some well-established conventional methods using manual features. Chen and Soo [16] used CNN and Highway Networks to increase the depth of networks for humor detection. Zhao et al. [17] proposed a tensor embedding method to capture lexical similarity to detect humor. Blinov et al. [18] collected a dataset of jokes and funny dialogues in Russian and used language model fine-tuning for text classification. There is no doubt that deep learning-based methods can extract high-dimensional features automatically and achieve high performance in humor recognition. However, previous studies did not take into account the linguistic features of humor when using deep learning. They ignored the guidance of humor theory, and most of the experimental results are difficult to illustrate and explain.

2. Methods

In this section, we introduce our model, PACGA. Our model is able to improve humor recognition by considering both phonetic representation and latent semantic information associated with ambiguous words.

The overall architecture of PACGA is shown in Figure 1. The framework consists mainly of three parts: (1) a convolutional neural network for phonetic structure comprehension, (2) a Bi-GRU combined with attention mechanism for semantic comprehension associated with ambiguous words, and (3) a gated attention strategy is used to leverage phonetic representations and semantic representations to recognize humor. We describe the details of our model in the following sections.

2.1. Phonetics Comprehension Network (PCN). Many humorous texts play with sounds, creating incongruous sounds or words [3]. Mihalcea and Strapparava [7] claim that the phonetic features of humorous texts are at least as important as their content. For example, "More sun and air for son and heir;" "sun" and "son" and "air" and "heir" are homophones. Both of them make the sentence not only harmonious and pleasant but also interesting and humorous.

The pronunciation of words is not exactly the same as their spelling. In order to get the phonetic representation of words, we use the Carnegie Mellon University (CMU) pronouncing dictionary. The current phoneme set of CMU has 39 phonemes, which is more accurate than the version without lexical stress. We convert each word into its corresponding phoneme. For example, the pronunciation of "word" is ["W," "ER," "D"]. It should be noted that a word

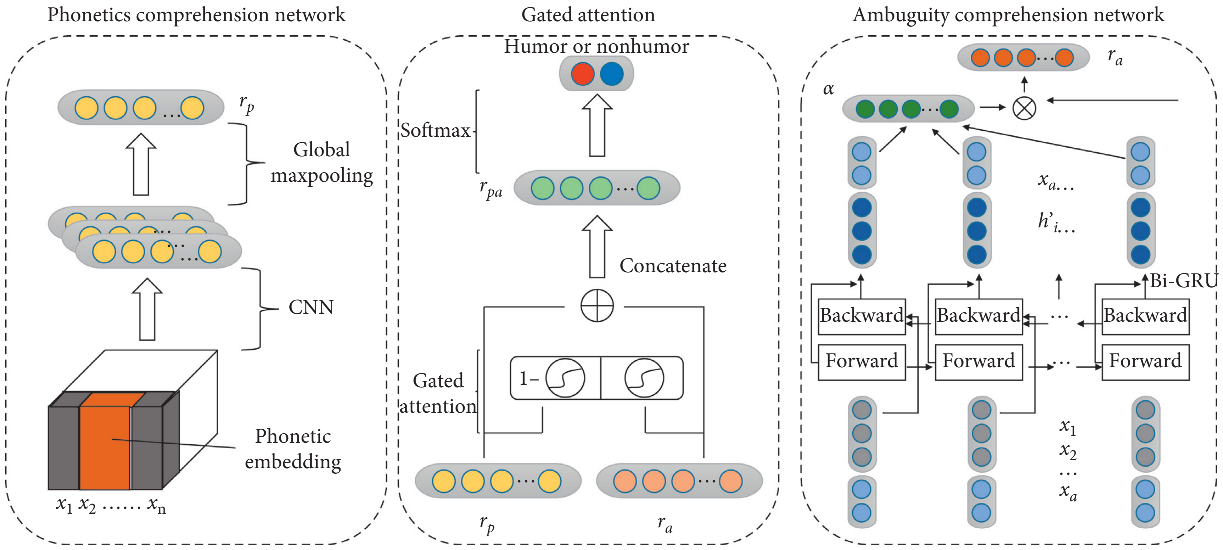


Figure 1: The Framework of Phonetics and Ambiguity Comprehension Gated Attention network.

may have more than one phonetic symbol in CMU. We use all the pronunciations of a dictionary entry for the speech extension and match any pronunciation as the speech extension of a word. Following Jaech's [19] work, we apply a substitution matrix between vowels and vowels and consonants and consonants. It can be used as a phonetic extension of the original word when the pronunciation is found in CMU after phoneme replacement.

2.1.1. Phonetics Embedding Layer. In the phonetics embedding layer, the pronunciation of each word can be mapped to a high-dimensional feature space for capturing the meaningful semantic information. For each word w_i, in a sentence $S\,S = \{w_1, w_2, \ldots, w_N\}$, $w_i \in \mathbb{R}^d$ and we convert the w_i into $P = \{p_1, p_2, \ldots, p_l\}$, $p_i \in \mathbb{R}^{d'}$ is the pronunciation of a word, where d and d' are the dimensional vector, N is the length of sentence, and l is the length of w_i. For the phonetics embedding, we randomly initiate.

2.1.2. Permute Layer. The permute layer can permute the dimensions of the input according to a given pattern. In our work, we aim to find out the pattern of alliteration or rhyme by the permute layer. The transformed matrix represents the pronunciation of different words among corresponding phonetics to feed the convolutional layer.

2.1.3. Convolutional Layer. We adopt the convolution operation in order to learn the local features of phonetic representation. In general, the convolutional layer uses a filter to extract local n-gram features. A filter can use a window of h words to generate the new feature map. c_t is a feature map which is produced by a window of words $x_{i:\,i+L-1}$. The formula is as follows:

$$c_t = f\left(wx_{i:\,i+L-1} + b\right), \tag{1}$$

where f is the nonlinear function ReLU, w is the filter to produce the feature map c_t, L is the length of the window, and b is the bias.

2.1.4. MaxPooling Layer. GlobalMaxPool2D is used to generate the phonetic representation after capturing the local speech features using two-dimensional CNN.

At this point, we get the phonetic representation r_p of a target sentence by the Phonetics Comprehension Network.

2.2. Ambiguity Comprehension Network (ACN). Ambiguity is the disambiguation of words with multiple meanings [20]. Humor and ambiguity often go together when a listener expects one meaning but is forced to use another meaning [3]. For a humorous example, "it is so hot that all the fans left after the baseball game." The surface meaning of "fans" is a ball game fan, but the implication may be that the electric fans are off. An ambiguous word with multiple possible meanings may lead the readers to misunderstand the sentence. It is the keyword that triggers humor. Furthermore, we also note that the multiple meanings of the ambiguous word are often quite different. To sum up, we pay attention to capturing ambiguous words in a sentence that can help us to improve humor recognition.

2.2.1. Word Embedding. Every word feature of a humorous text can be mapped to a high-dimensional feature space in this layer for capturing the meaningful semantic regularities. Here, GloVe [21] is applied as the pretrained word vector in order to produce the word embedding for detecting humor.

2.2.2. Ambiguous Word Embedding. The definition of an ambiguous word here is a word in a humorous sentence with multiple meanings that has the highest semantic similarity. Our work is strongly based on the intuition that humor arises from ambiguous words. In other words, the more meanings a word has and the higher the semantic distance between them, the more contribution it makes to humorous sentences. Here, we use WordNet to identify ambiguous words for detecting humor. Firstly, we ignore the stop words of a sentence. Then, we compute the number of synsets for each word though WordNet and select top T words as candidate ambiguous words. The semantic similarity can be computed among the meanings of each candidate word. Then, we choose the cosine similarity function to measure the semantic distance. Let $X = \{x_1, x_2, \ldots, x_N\}$, $x_i \in \mathbb{R}^d$ be word embedding, $X_i' = \{x_{i1}, x_{i2}, \ldots, x_{iK}\}$ be the synset of x_i, and K be the number of synonyms for the word x_i. The similarity is calculated as follows:

$$\mathrm{Sim}\left(\mathrm{X}_i'\right) = \max\left(\frac{\langle x_{im}, x_{in}\rangle}{\|x_{im}\| \cdot \|x_{in}\|}\right). \tag{2}$$

As a result, the word with the highest similarity is the selected ambiguous word to express humor in a sentence. The ambiguous word is represented as $x_a \in \mathbb{R}^d$.

To combine the information of ambiguity and context, we learn ambiguous word embedding for humor recognition. Since the common word embedding representations exhibit a linear structure, it makes it possible to meaningfully combine words by an elementwise addition of their vector representations [22]. In order to better take advantage of information within ambiguous, we append the ambiguous word representation to each word embedding in text. The ambiguous word embedding of a word x_i' for a specific target x_a is $x_i' = x_i \oplus x_a$, where $\oplus$ is the vector concatenation operation.

2.2.3. Bidirectional Gated Recurrent Units (Bi-GRU).

We leverage a Bi-GRU on top of the ambiguous word embedding to capture the features for humor recognition. The Bi-GRU is used over X to generate a hidden vector sequence $(h_1, h_2, \ldots, h_N)$. At each step s, the hidden vector h_s is computed based on the current vector x_s and the previous vector h_{s-1}. The formula is as follows:

$$\begin{aligned}
z_s &= \sigma\left(W_z x_s + U_z h_{s-1} + b_s\right),\\
r_s &= \sigma\left(W_r x_s + U_r h_{s-1} + b_r\right),\\
\widetilde{h}_s &= \tanh\left(W_h x_s + r_s \diamond U_h h_{s-1} + b_h\right),\\
h_s &= z_s h_{s-1} + \left(1 - z_s\right) \diamond \widetilde{h}_s,
\end{aligned} \tag{3}$$

where σ is the sigmoid function, z_s is the reset gate and r_s is the update gate, x_s represents the input, $\widetilde{h}_s$ is the candidate hidden state and h_s is the hidden state at time s, and $\diamond$ represents r elementwise multiplication operation.

Bi-GRU consists of two hidden states at each time step s: one is forward GRU $\overrightarrow{h}_s$ and the other is backward GRU $\overleftarrow{h}_s$. Finally, the two parts above are concatenated: $h_s = [\overrightarrow{h}_s; \overleftarrow{h}_s]$.

2.2.4. Ambiguity Attention Bi-GRU.

The standard Bi-GRU cannot pay attention to the ambiguity for humor recognition, even if we add ambiguous information in the embedding layer. To address this issue, we utilize the attention mechanism to capture the key part of the sentence in response to a given ambiguous word.

For each time step, Bi-GRU produces a hidden vector h_i. Furthermore, the ambiguous word representation x_a and hidden vector h_i are concatenated, $H\prime = \{h_1', h_2', \ldots, h_N'\}$, $H\prime \in \mathbb{R}^{2d\times N}$. H' is a matrix of hidden vectors, where d is the numbers of neurons and N is the length of the sentence. Then, we use the attention mechanism to produce an attention weight vector α and the weighted hidden vector r_a. The formulas are as follows:

$$\begin{aligned}
M &= \tanh\left(W_a H' + b\right),\\
\alpha &= \frac{\exp\left(W_\alpha M\right)}{\sum_i^n \exp\left(W_\alpha M\right)},\\
r_a &= H\prime \alpha^T,
\end{aligned} \tag{4}$$

where $M \in \mathbb{R}^{2d\times N}$, $\alpha \in \mathbb{R}^N$, and $r \in \mathbb{R}^N$. W_a and $W_\alpha \in \mathbb{R}^{2d}$ are parameters. α is a vector of ambiguity attention weights and r_a is a weighted representation of a given sentence with the special ambiguous word.

At this point, we get the ambiguity representation r_a by the Ambiguity Comprehension Network.

2.3. Gated Attention Mechanism.

After learning by the phonetics and ambiguity comprehension network, we combine the two parts to get the integrated representation. Intuitively, phonetic structure and ambiguity contribute differently to humor. Therefore, gated attention is leveraged to model the confidence of clues provided by the two parts. We calculate the value of the attention gate as follows:

$$g = \sigma\left(w\left[r_p; r_a\right] + b\right), \tag{5}$$

where σ is the sigmoid function, w is the weight matrix, and b is the bias.

In order to control the information between phonetic and ambiguous information, we use the value of attention gate g and $1 - g$ as the combination weights. The final representation of a sentence is as follows:

$$r_{pa} = g \odot r_p + (1 - g) \odot r_a, \tag{6}$$

where r_{pa} is the integrated representation, r_p is the phonetic representation, r_a is the ambiguous semantic representation, g is the combination weight, and $\odot$ is elementwise multiplication.

Humor recognition can be formalized into text classification. r_{pa} is the vector representation of the text and it can be used as the input to obtain the final classification result:

$$p = \left(W_p r_{pa} + b_p\right), \tag{7}$$

where p is the predicted probability of humorous text and W_p and b_p are the biases.

2.4. Model Training.

The model can be trained in an end-to-end way by backpropagation, and we use crossentropy loss as the loss function. Let y be the true distribution and $\widehat{y}$ be the predicted distribution for the text dataset. The goal of training is to minimize the loss function between y and $\hat{y}$ for all samples. We can formalize this process as follows:

$$\text{loss} = -\sum_{i}\sum_{j} y_i^j \log \hat{y}_i^j + \lambda\theta^2, \tag{8}$$

where i is the index of sentences, j is the index of class, λ is the L_2-regularization term, and θ is the parameter set.

3. Experiments

In this section, we first introduce the dataset and evaluation metrics. Then, we compare the performance of our model with several strong baselines in humor recognition. Finally, we give a detailed analysis of our method, including ablation experiments, visualization results, and error analysis.

3.1. Datasets and Evaluation Metrics. We conduct experiments on the widely used Pun-of-the-day dataset and oneliners 16000 dataset. Table 1 shows their detailed statistical distribution.

3.1.1. Pun-of-the-Day (Puns). This dataset was constructed by Yang et al. [3]. The humorous texts of this dataset are from the Pun of the Day website, and the negative samples are from AP News, New York Times, Yahoo! Answer, and Proverb. The dataset contains an equal number of positive and negative samples. The average length of sentences is 13.5 words.

3.1.2. Oneliners-16000 (Oliners). This dataset was constructed by [7]. Oneliners in this dataset are from some famous humorous websites, and the negative samples are from the titles of Reuter news. It is also a balanced dataset. The average length of sentences is 12.6 words.

3.1.3. Evaluation Metrics. We use Accuracy (Acc), Precision (P), Recall (R), and F-measure (F1) in our experiments to measure performance in humor recognition.

3.1.4. Training Details. We apply the proposed model to humor recognition tasks. In our experiments, for the ambiguity comprehension network, all word vectors are initialized by GloVe which trains on 6B tokens and 400k vocabulary words of Wikipedia 2014, and the dimension is 300. The size of units in Bi-GRU is 150 and dropout *dp* is in the range {0.25, 0.35, 0.5}. The learning optimizer *op* is in the range {RMSprop, Adadelta, Adam}. The learning rate is 0.0001. We use learning rate decay and early stop in the training process. For the Phonetics Comprehension Network, we firstly convert tokenized input sentences with phonetic vectors by random initialization. The range of filter sizes is {[2, 3, 4], [3, 4, 5]}. For each filter size, 128 filters are applied to the model. The top T in the range {1, 3, 5} are candidate ambiguous words.

Table 1: Statistics: Puns and Oliners.

Dataset	Positive	Negative
Puns	2423	2403
Oliners	16000	16000

We use 5-fold crossvalidation with a grid search method to select the optimal parameters. In detail, for each parameter, the following crossvalidation operations are performed. (1) The original dataset is randomly divided into five equally sized subsets. (2) For the five subsets, four subsets are used to train the model and the remaining subset is used as validation data for testing the model. (3) We repeat step (2) five times such that each of the five subsets is used as the validation data once. (4) The five results from the folds are averaged to produce results. Finally, the parameter pair with the highest results obtained by the crossvalidation process is set as the optimal parameters. In our experiments, dp is 0.35, op is Adam, filter sizes is [2, 3, 4], and T is 3.

3.2. Comparison with Existing Methods. We compare our proposed model with several baselines:

3.2.1. Support Vector Machine (SVM). This method uses all the features mentioned in the paper [3].

3.2.2. HCFWord2ve. The method is proposed by Yang et al. [3].

3.2.3. CNN. This method is proposed by Chen and Lee [15].

3.2.4. CNN + HN + F. This method was proposed by Chen and Soo [16].

3.2.5. TM. This method was proposed by Zhao et al. [17].

3.2.6. Syntactic. Liu [12] proposed to exploit syntactic structure features to enhance humor rrecognition.

3.2.7. Bi-LSTM + CNN. The method is a complete reimplementation of the proposed method in Bertero and Fung [14].

3.2.8. Bi-GRU. We employ word embedding and learn the latent semantic representations through Bi-GRU.

3.2.9. Bi-GRU + F. In addition to employing semantic representations learned automatically by Bi-GRU, the artificial features mentioned above are also incorporated into the network.

3.2.10. Bi-GRU + Att. We implement a deep learning Bi-GRU architecture with a focus on recognizing humorous text.

3.2.11. PACGA. We combine the phonetic structure and ambiguity information and use gated mechanism to adjust the effects of the two parts.

The results of the comparisons are listed in Tables 2 and 3. From the results, we observe that

(1) The traditional machine learning methods perform unsatisfactorily. The results on the two datasets show that their performance is lower than the neural network in many evaluation metrics. Furthermore, for the same artificial feature sets, the traditional machine learning methods exhibit different performances on the two datasets. For Puns, HCFWord2vec is better, but for Oliners, SVM is better. This shows machine learning-based methods depend on the construction of features, and their generalization ability is insufficient.

(2) TM employs a semisupervised label propagation procedure. It used a tensor embedding method for small sample humor recognition, but achieved only about 70% of F1.

(3) CNN performed worse than the Bi-GRU on both datasets (85.7% compared with 88.15% and 86.09% compared with 86.94%). CNN with extensive filter size, number and Highway Networks achieved high performance. The reason may be that the depth networks are of benefit for humor detection.

(4) Bi-LSTM + CNN, the combination of Bi-LSTM and CNN, performed worse than Bi-GRU on both datasets. By stacking a layer of a neural network onto another, a deep learning model can learn high-level features automatically. However, the hybrid LSTM and CNN cannot better extract latent semantic information for recognizing humor.

(5) Bi-GRU + F adds artificial features of humor to the model of Bi-GRU. We expected a higher performance than the Bi-GRU, but the results obtained are instead much lower on most of the evaluation metrics. The input of manually constructed features may conflict with semantic features that are automatically learned by the Bi-GRU. Therefore, adding too many artificial features into the deep learning methods cannot effectively improve humor recognition to some extent.

(6) Bi-GRU + Att uses the attention mechanism without the information of ambiguous word. Obviously, its experimental performance has not been greatly improved, which is largely due to its inability to pay close attention to features strongly related to humor.

(7) PACGA, our proposed method, achieved the comparable performance on both datasets for F1. For Puns, PACGA improved upon ordinary Bi-GRU by 2.12% for F1, and for Oliners by 2.27%. Even compared with the strong baseline CNN + HN + F, the performance of our model was superior. Our proposed model performed better than CNN with Highway Networks on Puns and achieved comparable results on Oliners (90.81 compared with 90.1% and 90.28% compared with 90.3%). This shows that our proposed phonetics information, ambiguity information, and gated attention mechanism have superior performance in humor recognition.

Table 2: Experimental results on Puns dataset. Best results are in bold. The results with superscript * are imported from the literature [3, 15–17].

Models	Acc (%)	P (%)	R (%)	F1 (%)
SVM*	83.85	85.91	82.52	84.18
HCFW2V*	85.4	83.4	88.8	85.9
Syntactic*	—	—	—	—
TM*	74.5	75.2	72.3	73.7
CNN*	86.1	86.4	86.4	85.7
CNN + HN + F*	**89.4**	86.6	**94.0**	90.1
Bi-lstm + CNN	85.38	81.42	91.97	86.37
Bi-GRU	87.72	84.23	92.46	88.15
Bi-GRU + F	87.14	**89.87**	83.34	86.48
PACGA	88.69	88.94	92.76	**90.81**

Table 3: Experimental results on Oliners dataset. Best results are in bold. The results with superscript * are imported from the literature [3, 12, 16, 17].

Models	Acc (%)	P (%)	R (%)	F1 (%)
SVM*	83.12	88.04	80.26	82.24
HCFW2V*	79.7	77.6	83.6	80.5
Syntactic*	85.0	82.7	89.1	85.8
TM*	70.5	72.1	66.7	69.3
CNN*	84.24	85.73	86.46	86.09
CNN + HN + F*	**89.7**	87.2	**93.6**	**90.3**
Bi-lstm + CNN	85.97	86.30	85.21	85.75
Bi-GRU	85.92	87.81	86.08	86.94
Bi-GRU+F	84.78	84.11	84.69	84.40
PACGA	89.47	**88.78**	91.84	90.28

(8) Compared with the baseline methods, our model achieves a higher accuracy score and F1 score for Puns, but lower precision and recall. We argue it is the different types of additional information which cause this phenomenon. Our model can learn latent semantic and phonetic information behind humor, such as phonetic structure and ambiguous information, and gated attention mechanism is applied to adjust the weight between them for proving more relevant features driven by humor theory, while the other methods usually only employ semantic information for obtaining high precision and recall compared with PACGA. Our model achieves the comparable performance on two datasets, which shows that our model has a better generalization capability.

3.3. Detailed Analysis. We conduct extra experiments to analyze our model in detail.

3.4. Analysis of Different Parts of PACGA. In order to show the effectiveness of different parts of our model, we split our model into two parts for verification. Firstly, we only use Bi-GRU without phonetics comprehension and ambiguity comprehension. Then, we implement PCN that considers phonetic embedding as input, and the CNN model is employed to recognize humor. In addition to phonetic information, we also try to distinguish humor only by using semantic information. Next, we design an ACN model that employs word embedding and ambiguous word information to learn potential humorous features based on Bi-GRU and attention mechanism. Finally, we introduce our proposed model PACGA. Tables 4 and 5 show the performance of all the models on both datasets:

(1) Tables 4 and 5 show that Bi-GRU achieves the worse performance which is consistent with our intuition. Without the phonetic structure and ambiguous word information, the performance of Bi-GRU in humor recognition is unsatisfactory.

(2) PCN only uses phonetic information, and its performance is significantly lower than the other models on both datasets. Obviously, only using a single model to capture phonetic features for detecting humor could not give a competitive performance. Semantic information plays an important role in the identification of humor.

(3) Compared with Bi-GRU, the performance of ACN is slightly improved. This shows that ambiguous word information and attention mechanism is helpful for Bi-GRU to focus on the latent sematic features of humor.

(4) Among all the methods, PACGA achieves the best performance for this task. The reason is our model considers the phonetic information, word information with ambiguous words, and gated attention mechanism.

Table 4: Analysis of the PACGA model on Puns.

Models	Acc (%)	P (%)	R (%)	F1 (%)
Bi-GRU	87.72	84.23	**92.46**	88.15
PCN	84.43	83.92	88.14	85.98
ACN	87.38	86.69	91.02	88.80
PACGA	**88.69**	**88.94**	92.76	**90.81**

Table 5: Analysis of the PACGA model on Oliners.

Models	Acc (%)	P (%)	R (%)	F1 (%)
Bi-GRU	85.92	87.81	86.08	86.94
PCN	83.97	85.12	83.87	84.49
ACN	86.64	87.39	87.47	87.42
PACGA	**89.47**	**88.78**	**91.84**	**90.28**

Table 6: Performance of combinational strategies.

Strategy	Models	Puns F1 (%)	Oliners F1 (%)
Single	PCN	85.98	84.49
	ACN	88.80	87.42
Combination	PAC-ST1	89.21	88.63
	PAC-ST2	89.33	88.72
	PAC-ST3	90.81	90.28

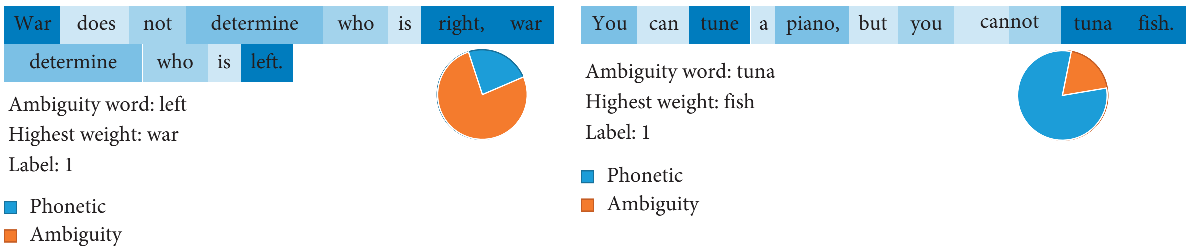
3.5. Impact of Different Combination Strategies. The combination strategy may affect the performance in humor recognition and measure the importance of our two main parts. Therefore, we design a series of experiments to explore the impact of different combination strategies. We adopt three strategies. (1) PAC-ST1: it directly combines the phonetic representation and ambiguity representation. (2) PAC-ST2: it assumes that two parts of information are of the same importance, and the parameter g is a constant, the value is 0.5. (3) PAC-ST3: the two parts of information have different importance. The gated attention is used to model the confidence of clues provided by the two parts.

We compare the single model and combination model with different strategies, and the results are given in Table 6. From the results, we find that all the combined models outperform the single model, which shows that both the phonetic structure and semantic information contribute to humor recognition. Among the combination models, the performance of PAC-ST1 and PAC-ST2 were roughly the same, and PAC-ST2 had a slight improvement. Furthermore, PAC-ST3 beat both of them by a large margin (1.48% or 1.56% on F1) for both datasets. This shows that our presented gated attention strategy to assemble information can better capture the inherent features behind humor.

3.6. Visualization of Attention. In order to validate the effectiveness of our model, PACGA, we visualize the attention layers for the sentences whose labels are correctly predicted.

From Figure 2, we can see that the common words, such as "is" and "does," are afforded little attention by our model, which justifies the intuition that common words make little contribution to identifying humor. Meanwhile, some specific words are crucial for humor. In Figure 2(a), the words "war," "right," "determines," and "left" have higher attention weights, which implies our model pays attention to those words, as we expect. It shows that ambiguous words can provide useful information for its context to adjust its attention, and it plays a great role in a humor recognition task. In Figure 2(b), obviously, the ambiguity is not the main reason for humor, and we pay much attention to the phonetic structure, which implies our model can learn the importance of phonetic structure and ambiguity for humor recognition. Thus, through the PACGA, we can well model phonetic structure and ambiguity, respectively, and then concatenate their representations by gated attention mechanism, which is helpful for humor recognition.

3.7. Error Analysis. We also conduct a preliminary error analysis in this section. Our aim is to find some problematic issues by studying some misclassified test cases and to improve the humor recognition of our model in the future.



(a)

(b)

Figure 2: Visualization of attention. A darker color means more importance. The pie chart shows the weights of the two parts based on gated attention mechanism.

Exp 3. The one who invented the door knocker got a no bell prize.

Exp 4. A tidy desk is a sign of a cluttered desk drawer.

For Exp 3, the true label is "humor," but our model predicted its label as "nonhumor." In this example, the punch line is "no bell prize," it sounds like "Nobel Prize." Obviously, this type of humor is caused by similarity in pronunciation, but "Nobel Prize" does not appear in the sentence, and our model cannot capture any phonetic information. Hence, some background knowledge would be required in order to predict the label correctly. For Exp 4, "tidy" and "cluttered" are opposites, and this kind of conflict makes a sentence humorous. Humor sometimes relies on two or more inconsistent, unsuitable, or incongruous parts or circumstances. Therefore, our model needs to be able to identify inconsistencies simultaneously.

4. Conclusions and Future Work

In this paper, we design an automatic computational neural network named Phonetics and Ambiguity Comprehension Gated Attention network (PACGA) to detect humor. The main idea of PACGA is to use phonetic structure and ambiguity for humor recognition. In our model, a phonetics comprehension network is used to understand the phonetic representation of CMU pronunciation dictionary by CNN. Ambiguity comprehension network leverages latent semantic representation associated with ambiguous words by Bi-GRU. Based on phonetics comprehension network and ambiguity comprehension network, gated attention mechanism is used for modeling the confidence of clues. Experiments on Puns and Oliners datasets verify that our proposed PACGA can learn effective information for phonetic structure and semantics which provide significant information for detecting humor. In addition, the detailed analysis and visualization of attention also show the validity and interpretation ability from different perspectives.

In the future, we would like to step further into how to integrate humor characteristics into a deep learning model. Certainly, how to use common sense for humor recognition is also an issue deserving of study.

Data Availability

All data analyzed during this study are public corpus, which can be obtained by sending an email to the dataset builder. The data "pun of the day" that support the findings of this study are openly available in [3]. The data "onelienrs-16000" that support the findings of this study are openly available in [7].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (nos. 61632011, 61572102, 61702080, 61602079, and 61806038), Ministry of Education Humanities and Social Science Project (no. 16YJCZH12), Fundamental Research Funds for the Central Universities (no. DUT18ZD102DUT19RC(4)016), National Key Research Development Program of China (no. 2018YFC0832101), and China Postdoctoral Science Foundation (no. 2018M631788).

References

[1] X. Yan and P. Ted, "Duluth at semeval-2017 task 6: language models in humor detection," 2017, https://arxiv.org/abs/1704.08390.

[2] O. Stock, C. Strapparava, and A. Nijholt, "The april fools' day workshop on computational humour," in *Proceedings of the Twentieth Twente Workshop on Language Technology*, Trento, Italy, April 2002.

[3] D. Yang, A. Lavie, C. Dyer, and E. Hovy, "Humor recognition and humor anchor extraction," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2367–2376, Lisbon, Portugal, September 2015.

[4] C. Bucaria, "Lexical and syntactic ambiguity as a source of humor: the case of newspaper headlines," *Humor*, vol. 17, no. 3, pp. 279–310, 2004.

[5] A. Reyes, P. Rosso, and D. Buscaldi, "From humor recognition to irony detection: the figurative language of social media," *Data & Knowledge Engineering*, vol. 74, pp. 1–12, 2012.

[6] S. Castro, M. Cubero, D. Garat, and G. Moncecchi, "Is this a joke? detecting humor in Spanish tweets," in *Proceedings of the Ibero-American Conference on Artificial Intelligence*, pp. 139–150, San José, Costa Rica, November 2016.

[7] R. Mihalcea and C. Strapparava, "Making computers laugh: investigations in automatic humor recognition," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 531–538, Vancouver, Canada, 2005.

[8] R. Mihalcea and C. Strapparava, “Computational laughing: automatic recognition of humorous one-liners,” in *Proceedings of Cognitive Science Conference*, pp. 1513–1518, Denton, TX, USA, 2005.

[9] R. Zhang and N. Liu, “Recognizing humor on twitter,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 889–898, Shanghai, China, November 2014.

[10] F. Barbieri and H. Saggion, “Automatic detection of irony and humour in twitter,” in *Proceedings of 5th International Conference on Computational Creativity (ICCC)*, pp. 155–162, Ljubljana, Slovenia, June 2014.

[11] L. Liu, D. Zhang, and W. Song, “Modeling sentiment association in discourse for humor recognition,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 586–591, Melbourne, Australia, July 2018.

[12] L. Liu, D. Zhang, and W. Song, “Exploiting syntactic structures for humor recognition,” in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1875–1883, Santa Fe, NM, USA, August 2018.

[13] D. Bertero and P. Fung, “Deep learning of audio and language features for humor prediction,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pp. 496–501, Portorož, Slovenia, May 2016.

[14] D. Bertero and P. Fung, “A long short-term memory framework for predicting humor in dialogues,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 130–135, San Diego, CA, USA, June 2016.

[15] L. Chen and C. M. Lee, “Convolutional neural network for humor recognition,” 2017, https://arxiv.org/pdf/1702.02584.pdf.

[16] P. Y. Chen and V. W. Soo, “Humor recognition using deep learning,” *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 113–117, 2018.

[17] Z. Zhao, A. Cattle, E. Papalexakis et al., “Embedding lexical features via tensor decomposition for small sample humor recognition,” 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)*, pp. 6377–6382, Hong Kong, China, November 2019.

[18] V. Blinov, V. Bolotova-Baranova, and P. Braslavski, “Large dataset and language model fun-tuning for humor recognition,” in *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4027–4032, 2019.

[19] A. Jaech, R. Koncel-Kedziorski, and M. Ostendorf, “Phonological pun-derstanding,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 654–663, San Diego, CA, USA, 2016.

[20] A. Fan, M. Lewis, and Y. Dauphin, “Hierarchical neural story generation,” 2018, https://arxiv.org/abs/1805.04833.

[21] J. Pennington, R. Socher, and C. Manning, “GloVe : global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014.

[22] J. Du, R. Xu, Y. He, and L. Gui, “Stance classification with target-specific neural attention networks,” in *International Joint Conferences on Artificial Intelligence*, Melbourne, Australia, August 2017.