sensors

MDPI

Article

Multidimensional Latent Semantic Networks for Text Humor Recognition

Siqi Xiong [1,2], Rongbo Wang [1,2,*], Xiaoxi Huang [1,2] and Zhiqun Chen[1,2]

[1] College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China; xsqzjyy@163.com (S.X.); huangxx@hdu.edu.cn (X.H.); chenzq@hdu.edu.cn (Z.C.)
[2] Institute of Cognitive and Intelligent Computing, Hangzhou Dianzi University, Hangzhou 310018, China
* Correspondence: wangrongbo@hdu.edu.cn

Abstract: Humor is a special human expression style, an important "lubricant" for daily communication for people; people can convey emotional messages that are not easily expressed through humor. At present, artificial intelligence is one of the popular research domains; "discourse understanding" is also an important research direction, and how to make computers recognize and understand humorous expressions similar to humans has become one of the popular research domains for natural language processing researchers. In this paper, a humor recognition model (MLSN) based on current humor theory and popular deep learning techniques is proposed for the humor recognition task. The model automatically identifies whether a sentence contains humor expression by capturing the inconsistency, phonetic features, and ambiguity of a joke as semantic features. The model was experimented on three publicly available wisecrack datasets and compared with state-of-the-art language models, and the results demonstrate that the proposed model has better humor recognition accuracy and can contribute to the research on discourse understanding.

Keywords: humor recognition; humorous semantic features; discourse understanding; human-computer interaction; deep learning

check for updates

Citation: Xiong, S.; Wang, R.; Huang, X.; Chen, Z. Multidimensional Latent Semantic Networks for Text Humor Recognition. *Sensors* **2022**, *22*, 5509. https://doi.org/10.3390/s22155509

Academic Editor: Raffaele Gravina

Received: 30 May 2022
Accepted: 20 July 2022
Published: 23 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Humor is an important flavoring agent in human life and communication, often easing tensions and providing an expression way for socially unacceptable feelings, behaviors, and impulses [1]. However, text is an important medium for conveying humor, and as the field of artificial intelligence (AI) places more and more focus on the study of text, the exploration of deeper semantic directions of text is becoming increasingly popular, one of which is computational humor. The computational humor is based on humor theory in linguistics to reveal the mechanisms of human humor and to build a linguistic model with humor cognitive functions for providing a basis for deeper human–computer interaction [2]. For practical applications, if computers can recognize humor expression, then true intentions of authors can be understood more accurately for making human–computer interaction more interesting and engaging.

Jokes, generally short but simple and clever, are an important style for humor expression. Identifying whether a joke is humorous or not is an important task in computational humor. Humor recognition is often referred to as a binary classification task that detects whether a text is humorous or not. As there are different forms of humorous texts, such as dialogues, stories, and short jokes (witticisms), we cannot use a unified model to identify them. Therefore, in this paper, the humor recognition task focuses on short jokes because they can use fewer words to express humor. They have no complex semantic structures and longer contexts of long jokes, they have no complex interpersonal relationships of dialogues, and they usually generate humor through puns, rhymes, and inconsistencies in semantics. Normally, the punch line of a joke is brought in by a few wise words at the end

to create inconsistencies and thus make the text humorous [3]. Humor usually also has the ambiguity and phonetic features mentioned.

In recent years, there has been a growing emphasis on humor recognition, mostly in the context of textual humor recognition; many of the assessment tasks also have humor-related tasks [4,5] identifying humor by incorporating theoretical understanding of the linguistics of humor and extracting humorous features [6,7]. Many linguistic features are proposed for humor recognition, such as incongruity structure, ambiguity theory, interpersonal effect, phonetic style, and so on. However, manually constructing the features needed for humor recognition is difficult and laborious. With the development of deep-learning-based approaches, a number of methods have been proposed in recent work; for example, Kumar et al. [8] propose a combination of convolutional neural networks (CNN) and long short-term memory (LSTM), with the addition of a highway to enhance performance; Weller et al. [9] proposed the use of the transformer architecture to take advantage of its learning from the context of sentences; Lu Ren et al. [10] proposed to combine humor recognition and pun recognition, training the two tasks jointly, thus enhancing performance. There is also humor recognition through multimodal means [11,12]. Unlike these works, we propose a deep learning method approach based on humor linguistics to capture inconsistency features, phonetic features, and ambiguity features in humor, introducing the incongruity of humor caused by semantic inconsistency, lexical, and ambiguity.

In this paper, we propose a model to implement these methods that analyzes multiple characteristics of humor and extracts multiple features of a joke in order to determine whether a sentence contains humor or not. The contributions of the proposed are listed as follows:

- In order to identify the inconsistency, fuzzy features, and phonetic features of humor, a model is designed which is able to extract the fragment embeddings of jokes via RoBERTa word embeddings and CNNs, semantic inconsistency features using Glove word embeddings, CNNs, and transformer encoders, phonetic features in jokes via Carnegie Mellon University's pronunciation dictionary (CMU), and CNNs and fuzzy features using WordNet, Bi-LSTM, and attention mechanisms.
- Compared to previous work, this paper uses a new extraction approach for the three features proposed above. Different from exploring the relationship between words, in terms of extracting text inconsistencies, we no longer extract inconsistency features between words and words, but inconsistency features between snippets; obtain the phonetic features of sentences by converting words into phonetic sequences; and extraction of fuzzy features by exploring the relationship between words and synonyms.
- The model achieves superior results compared to the state-of-the-art models available on three datasets, pun-of-the-day [13], 200K-Oneliners [14], and SemEval-2021 Task-7 [15], respectively.

2. Related Work

2.1. Domestic and International Theoretical Research on Humor

The definition of humor in Cambridge Dictionary is the ability to be amused by something seen, heard, or thought about, sometimes causing you to smile or laugh, or the quality in something that causes such amusement. At present, there are comprehensive studies on humor, namely, three main theories: the superiority theory from a social behavioral perspective, the release theory from a psychoanalytical perspective, and the perverse theory from a psycho-cognitive perspective [16].

The study of humor theories provides important guidance on semantic and pragmatic rules for humor recognition, and builds the theoretical foundation for extracting humor features for humor recognition in this paper.

2.2. Relevant Studies in Text Humor Recognition

Naturally, there are many studies on humor detection in the form of one-liners. Feature engineering approaches are highly popular, exploiting a diverse range of features such

as stylistic [17], distribution of part of speech [18], and affective dimensions [19]. Purandare and Litman [20] analyzed acoustic–prosodic and linguistic features to automatically recognize humor during spoken conversations.

In recent years, humor recognition has been a very challenging task in natural language processing, making many researchers work for it. Mihalcea and Strapparava [3] extracted head rhymes, antonyms, and adult slang features for humor recognition. Mihalcea et al. [21]. artificially divided the witticisms into two parts: the "set up" and the "punchline", and used the semantic conflicts between them as the basic features of humor recognition, using semantic similarity calculation and some humorous linguistic features for humor recognition. Zhang et al. [19] designed up to 50 humorous features in 5 major categories based on the Twitter corpus. Yang et al. [13] constructed four types of features for humor recognition, including inconsistency, ambiguity, interpersonal effects, and speech style.

In deep learning, humor recognition has also achieved excellent results. Bertero and Fung [22] combined word-level and audio-frame-level features and used RNNs and CNNs to predict humorous discourse. De Oliveira and Rodrigo [23] also applied LSTM to detect humor from reviews in the Yelp dataset and achieved satisfactory results. With the development of pretrained language models in natural language processing, some excellent pretrained language models have been applied for humor recognition tasks. Mao and Liu [24] proposed a humor recognition method based on BERT. Ma et al. [25] proposed an enhanced inference Bert (EI-Bert) based on different feature sentence pairs for humor recognition. At the same time, humorous datasets are also very difficult to obtain. Wu J. et al. [26] constructed a novel multimodal dataset named MUMOR for humor recognition, and Blinov V. et al. [27] collected a dataset of jokes and funny dialogues in Russian from various online resources and complemented them carefully with unfunny texts with similar lexical properties. The dataset comprises more than 300,000 short texts, which is significantly larger than any previous humor-related corpus.

However, the above research results only extracted one or two features without combining them; at the same time, researchers focused only on word-to-word relationships while ignoring snippet-to-snippet relationships. Therefore, this paper proposes a humor recognition model that incorporates text fragment inconsistency, phonetic features, and fuzzy features.

3. Analysis of Humor Features

Based on previous research on humor theories, this paper proposes three potential semantic features about humor, namely, inconsistency feature, speech feature, and fuzzy feature, for which the identical three features are designed and implemented in this paper, and then these features are combined for humor recognition.

3.1. Inconsistent Feature

The semantic scripting theory of humor [19] stated that inconsistency was one of the important causes of humor. The production of humor often depends on some incongruous, contradictory, or obviously contradictory combination [28]. That is, there is no direct relationship between one idea and other ideas. For example, "A clean desk is a sign of a cluttered desk drawer". In this example, there is a certain inconsistency between "clean desk" and "cluttered desk drawer", thus generating a certain humorous effect. Since it is difficult to directly confirm the inconsistency, semantic analysis is required to simplify this work. In recent years, with the development of deep learning applications, it has achieved remarkable results in text semantic representation [29].

3.2. Fuzzy Feature

The relevance theory of humor [30] mainly explores and analyzes humor from a common phenomenon in natural language, namely, ambiguity. When a word without ambiguous meanings has multiple meanings [31], it usually becomes an important sign

of producing humor. The main reason for ambiguity is that some words have superficial meanings, but they are "forced" to produce a deeper and more obscure meaning due to the constraints of context. For example, "Did you hear about the guy whose whole left side was cut off? He's all right now". In this sentence, besides the meaning of "on the right", "right" has another meaning of "recovery" when combined with "all". In order to solve similar problems, WordNet is usually introduced, and the farthest and nearest paths of the meaning contained in it are compared, so as to realize the discovery of ambiguity features. Based on this point, the humor caused by the ambiguity of words or phrases is identified.

3.3. Phonetic Feature

Other theories of humor also suggest that phonological properties are also important in generating humor [32]. Phonological properties make texts that are not originally humorous or funny. Jokes usually depend on being read aloud to produce comic effects through head rhymes, superlatives, rhymes, etc. Similar methods are often found in newspaper headlines, hymns, and jingles. Head rhyming chains usually refer to two or more words that begin with the identical pronunciation, while rhyming refers to words that end a sentence with the identical syllable between them. In order to extract features of pronunciation types, the Carnegie Mellon University Pronunciation Dictionary (CMU Pronouncing Dictionary) can be applied to implement humorous recognition of pronunciation categories.

4. Methods

In this section, the model is introduced in detail. The model improves the humor recognition through the three dimensions of joke inconsistency, phonetic features, and fuzzy features. The framework of this model is shown in Figure 1.

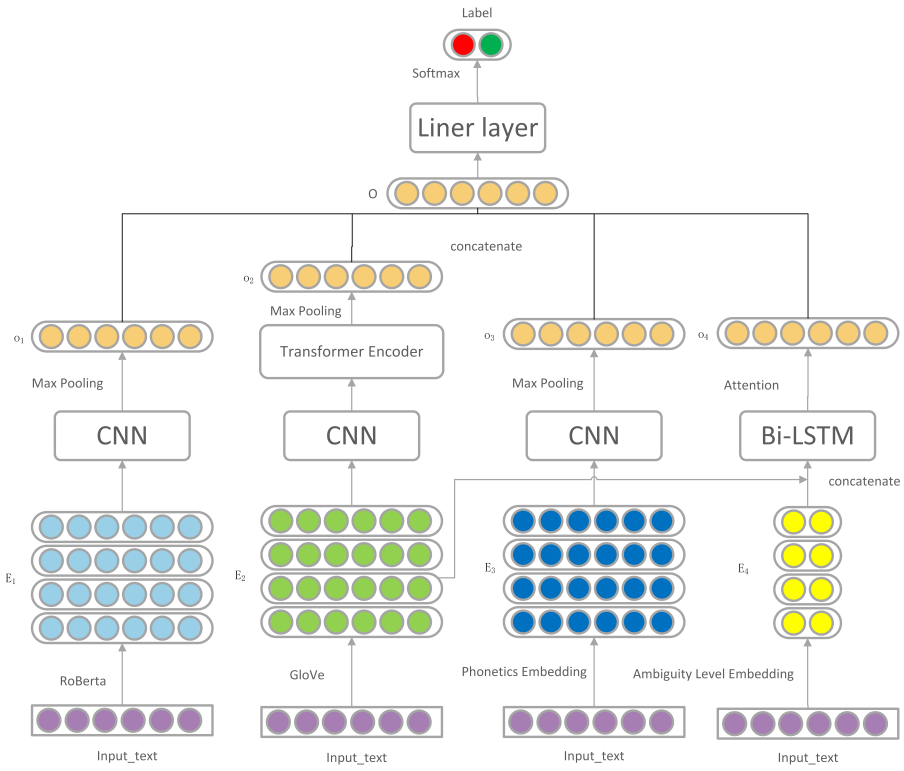

Figure 1. The framework diagram of the model consists of four parts: snippet embedding module, snippet semantic inconsistency module, voice feature module, and ambiguity module.

The framework of the model consists of four main components:

1. The snippet embedding module that uses RoBERTa and convolutional neural networks for joke snippet extraction embedding;

2. Semantic inconsistency extraction module applies convolutional neural network and transformer encoder to extract the semantic inconsistency of jokes;
3. Phonetic feature module to determine humor by convolutional neural network speech embedding;
4. Using Bi-LSTM and attention mechanism for textual-ambiguity-related ambiguity modules.

The details of the proposed model in this paper are presented in the following sections.

4.1. Snippet Embedding Module

Previous studies on humor recognition about inconsistency focused only on the inconsistency between words. Cao [33] proposed the relationship between fragments and improved the model on the above basis. RoBERTa was used to embed words, and convolutional neural network (CNN) was used to intercept fragments to give sentence representation.

4.1.1. Word Embedding

RoBERTa is an upgraded version of BERT [34,35] and has superior performance compared to BERT. RoBERTa improves upon BERT as follows:

1. Longer training time, larger batch size, more training data;
2. Removes the next predict loss;
3. Longer training sequences;
4. Dynamic adjustment of the masking mechanism.

In this paper, RoBERTa word embedding is chosen as the word embedding of the snippet embedding module, and after word embedding, it can be represented as $E_1 \in \mathbb{R}^{N\times e}$, where N is the sequence length and e is the word embedding dimension.

4.1.2. Convolutional Layer

Since the final result of this module is a sentence snippet embedding, the jokes are required to be divided, and the convolutional neural network is just adapted to this module, which can be applied to grasp contextual local features by implementing convolutional operations between a convolutional kernel and a series of word embeddings. From this, the convolution filter size is set to $f \in \mathbb{R}^{l\times e}$ and the step length equals 1. The input is operated by the convolution layer to obtain $C \in \mathbb{R}^{(l-N+1)\times e}$, and C is a set of vectors C=$\{c_1, c_2, c_3, \ldots, c_{l-N+1}\}$. The formula is as follows:

$$c_i = \sum E_{i,e} \otimes f_{N,e}, \tag{1}$$

Finally, the max pooling layer is passed until $o_1 \in \mathbb{R}^n$, and n is the convolutional layer output channel size.

4.2. Semantic Inconsistency Extraction Module

Inconsistency is widely considered as a humorous feature, and inconsistencies between semantics are not necessarily inconsistencies between words, but also between snippets. Sentence inconsistencies can be considered to some extent as oppositions or contradictions between semantic blocks of sentences.

4.2.1. Word Embedding

This module applies GloVe [36] for word embedding to obtain a sentence-level representation, which is fixed during training. The tokens of all unknown words are replaced with "<unk>". To obtain a uniform sentence length, each edited sentence is either truncated or filled with "<pad>". Embeddings for unknown word symbols and padded token symbols are initialized with zero vectors and random vectors, respectively. Finally, the word embedding is denoted as $E_2 \in \mathbb{R}^{N\times b}$; N is the sequence length and b is the embedding dimension.

4.2.2. Sentence Slicing and Capturing Semantics

Sentences are divided into many parts, as described in Section 4.1.2, the semantic block vector is obtained by dividing, and then the semantic inconsistency information of the text is obtained by the transformer encoder [37]. The attention vector a_i from a self-attentive structure is computed as follows:

$$a_i = \sum_{j=1}^{l-m} softmax\left(\frac{Q_i K_j^T}{\sqrt{e_i}}\right) V_j \tag{2}$$

where e_i is the dimension of Q_i and j is the number of sentence snippets. Q, K, and V represent three different types of encoded sentence fragment representations from GloVe. They can be calculated as follows:

$$S_{Q,K,V} = W^i_{Q,K,V} * c^i + b^i_{Q,K,V} \tag{3}$$

where c^i is the i-th segment representation, $W^i_{Q,K,V}$ is the corresponding projection matrix, and $b^i_{Q,K,V}$ is the bias term. Finally, the output $o_2 \in \mathbb{R}^n$ is obtained, where n is the output channel size of the convolutional layer.

4.3. Phonetic Feature Module

For a joke, in addition to inconsistency, phonetic feature is also an important feature for identifying humor. It makes a sentence humorous that is originally semantically not humorous, so we cannot ignore it.

4.3.1. Phonetic Embedding

In this paper, we use Carnegie Mellon University (CMU)'s pronunciation dictionary to obtain the speech representation of jokes $E_3 \in \mathbb{R}^{N \times k}$, for example, "however" can be decomposed into "HH AW2 EH1 V ER0".

4.3.2. Convolutional Layer

Then, we use the convolutional neural network to obtain the local features of the speech sequence and obtain the output, $o_3 \in \mathbb{R}^n$.

4.4. Ambiguity Module

Ambiguity, the disambiguation of words that have multiple meanings, is an important part of many humorous jokes. That is, humor is produced by exploiting semantic and pragmatic ambiguities, which are closely related to the different meanings that a word, phrase, or sentence may have. Therefore, this paper pays special attention to the ambiguity of words, and explores the possible impact of these ambiguous words on humor recognization.

4.4.1. Word Ambiguity Embedding

For the ambiguity of each word, this paper adopts a scoring mechanism according to the number of ambiguities in each word. For an input sentence, S={w_1,w_2,w_3,...,w_N}, using WordNet as an external resource, it calculates the number of synonyms for each w_i and classifies them according to the number of synonyms. In this paper, four levels are taken. Stop words are one level independently, and each level is equal to a different weights, which is initialized by a random vector, where N is the length of a sentence, and finally the ambiguous word level sequence T={t_1,t_2,t_3,...,t_N} of the sentence is obtained; $t_i \in \mathbb{R}^N$, k is the dimension.

4.4.2. Word Embedding

To concatenate the word embedding vector obtained in Section 4.2.1 with the word ambiguity embedding vector obtained in Section 4.4.1, the output vector is $E_4 \in \mathbb{R}^{(k+b) \times N}$.

4.4.3. Semantic Understanding Layer

Bi-directional long short-term memory (Bi-LSTM) [38] is added to the embedding layer to model the temporal interaction between humorous text words. The Bi-LSTM consists of a forward LSTM [39] and a backward LSTM to represent the contextual representation in two opposite directions, which avoids the vanishing gradient and scaling problems by learning the long-term dependencies of the text. The design of Bi-LSTM includes three gates and one unit for modeling semantic and contextual relations. This module applies the output of the embedding layer as input information, and then the update process of each forward LSTM network of Bi-LSTM is formulated as follows:

$$i_t = \sigma\left(\overrightarrow{W_i} \bullet \left[\overrightarrow{h_{t-1}}, \overrightarrow{E_{4,t}}\right] + \overrightarrow{b_i}\right) \tag{4}$$

$$f_t = \sigma\left(\overrightarrow{W_f} \bullet \left[\overrightarrow{h_{t-1}}, \overrightarrow{E_{4,t}}\right] + \overrightarrow{b_f}\right) \tag{5}$$

$$o_t = \sigma\left(\overrightarrow{W_o} \bullet \left[\overrightarrow{h_{t-1}}, \overrightarrow{E_{4,t}}\right] + \overrightarrow{b_o}\right) \tag{6}$$

$$g_t = \tanh\left(\overrightarrow{W_c} \bullet \left[\overrightarrow{h_{t-1}}, \overrightarrow{E_{4,t}}\right] + \overrightarrow{b_c}\right) \tag{7}$$

$$\overrightarrow{c}_t = f_t * \overrightarrow{c_{t-1}} + i_t * g_t \tag{8}$$

$$\overrightarrow{h_t} = o_t \odot \tanh\left(\overrightarrow{c_t}\right) \tag{9}$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{10}$$

where t is the step size, f_t is the forgetting gate, o_t is the output gate, c_t is the storage cell, and σ is the sigmoid activation function. W_i, W_f, W_o, and W_c are learned weights; b_i, b_f, b_o, and b_c are bias values; $\overrightarrow{h_t}$ is the output of the forward LSTM, which together with the output of the backward LSTM ($\overleftarrow{h_t}$) forms the vector h_t.

4.4.4. Attention Layer

In order to obtain the attention signal of humorous sentences according to the given ambiguity level of each word, the proposed model designs an attention mechanism [40]. The forward neural network is applied to calculate the semantic relevance of each word and its ambiguity level; the formula is as follows:

$$g_{am} = ReLU(W_{am}h_t + b_{am}) \tag{11}$$

$$\alpha_{am} = softmax(\omega^t g_{am}) \tag{12}$$

$$v_{am} = H\alpha_{am}^T \tag{13}$$

where $g_{am} \in \mathbb{R}^{(d+k)\times N}$, $\alpha_{am} \in \mathbb{R}^N$, $\omega \in \mathbb{R}^{d+k}$, and $v_{am} \in \mathbb{R}^{d\times m}$; α_{am} is the importance weight normalized by the softmax function; v_{am} is the context vector; and H is the output of the Bi-LSTM network.

This design is able to assign an appropriate word importance score by computing the semantic relatedness between a word and its ambiguity score. The final output $o_4 \in \mathbb{R}^n$.

4.5. Prediction Layer

The output vectors obtained by the previous four modules are connected, and finally the complete prediction layer input vector $O \in \mathbb{R}^{4n}$ is obtained, and this input vector is injected into the prediction layer. The formula is as follows:

$$v = ReLU(W_p[o_1, o_2, o_3, o_4] + b_p) \tag{14}$$

where $W_p \in \mathbb{R}^{4n\times 2}$ is the weight, b_p is the bias value, $v \in \mathbb{R}^2$ is result, and the obtained vector goes through the softmax classification layer to obtain the predicted value:

$$\hat{y} = softmax\left(W_f v + b_f\right) \tag{15}$$

where W_f is the weighting matrix, b_f is the bias value, and $\hat{y}$ is the predicted label of the proposed model.

We apply cross-entropy loss in MLSN model. The loss is given by:

$$loss = -\sum_{i=1}^{M}\sum_{j=1}^{N} y_i^j \log \hat{y}_i^j + \lambda\|\theta\|^2 \tag{16}$$

Here, M is the total number of all texts, and N is the number of classes. y is the true label of the text, and y denotes the predicted label of our model. i is the index of the text, j is the index of class, is the regularization parameter, and means all of the parameters in the model. The goal of the training is to minimize the loss function.

5. Experiment and Result Analysis

This section mainly introduces the experimental data and various baselines, compares the performance of each classifier, and concludes that the performance of the proposed model in this paper is better than the current state-of-the-art models.

5.1. Experimental Data and Evaluation Criteria

Pun-Of-The-Day (Puns): The dataset was collected by Yang et al., with humorous texts from websites of the same name, and non-humorous texts from AP News, New York Times, Yahoo News, and Proverbs. To avoid the classification problem caused by data imbalance, the numbers of negative and positive cases are basically the same distribution, and the total number of negative samples is 2403.

200K-Oneliners: The dataset contains 200k labeled short texts, evenly distributed between humorous and non-humorous. It is much larger than previous datasets and includes texts with similar textual features. The correlation between the number of characters and the target is not significant (+0.09), there is no significant connection between the target value and the sentiment feature, and the average sentence length is 12.

SemEval-2021 Task 7: The dataset contains 8000 training sets, 1000 validation sets, and 1000 test sets with an average length of 24.9. They collected 10,000 texts from Twitter and the Kaggle Short Jokes dataset, and had each annotated for humor and offense by 20 annotators aged 18–70.

The data distribution of the three datasets is shown in Table 1 below.

Table 1. Sample distribution of each dataset.

Data	Positive	Negative
200K-Oneliners	100k	100k
SemEval-2021 Task 7	6179	3821
Pun-Of-The-Day	2403	2403

Evaluation indicators: because the humor recognition in this paper is only to determine whether the text is humorous or not, which is essentially a binary classification task, this paper uses evaluation metrics that are widely applied in classification tasks: accuracy (Acc), precision (P), completeness (R), and F-measure (F1). The datasets mentioned above are divided into training set, validation set, and test set according to the ratio of 8:1:1.

$$P = TP/(TP + FP) \tag{17}$$

$$R = TP/(TP + FN) \tag{18}$$

$$F1 = 2 * P * R/(R + P) \tag{19}$$

$$Acc = (TP + TN)/(TP + FP + TN + FN) \tag{20}$$

Among them, TP (true positive) is a correct positive, indicating that the prediction is positive, and the prediction is correct, so it is actually a positive example. FP (False positive) is a false positive, which means that the prediction is positive, the prediction is wrong, and it is actually a negative case. FN (false negative) is a false negative, which means that the prediction is negative, the prediction is wrong, and it is actually a positive example.

5.2. Baselines

CNN + HN + F [41]: this method applies CNN with increasing filter size and highway layer [42] for humor recognition.

MAIS [33]: this method employs humorous snippet embeddings and contextual semantic inconsistency features to identify humor.

ABML [10]: The model unifies the two highly pertinent tasks, including the humor recognition and pun detection. In the ABML model, they design a co-encoder module to capture the common features between the two tasks by weight sharing. Apart from the co-encoder module, they also design two private encoder modules for the two tasks, respectively. The private encoder module is used to capture the private semantic feature of the two tasks.

BERT: Google released BERT in 2018 and successfully achieved the results of State Of The Art in 11 NLP tasks, winning praise from the natural language processing community. It is one of the current state-of-the-art natural language processing pretraining models.

XLNet [43]: Following BERT, Google and CMU jointly launched an improved version of BERT, XLNet, which optimized the shortcomings of BERT and achieved State Of The Art results in 18 tasks, especially in the field of text classification.

5.3. Experimental Settings

The model in this paper runs on a TITAN RTX GPU. The data of SemEval-2021 Task 7 has a fixed length of 64 after RoBERTa marking, and the length of Pun and 200K-Oneliners is 32. Similarly, the data for SemEval-2021 Task 7 has a fixed sequence length of 64 in the GloVe embedding, and a length of 32 for Pun and 200K-Oneliners. The length of phonetic embedding is fixed at 200. The hidden size of LSTM is 128, and the output channel size of the convolutional layer is 64. In GloVe word embedding, words that are not in the word list are always replaced by "<unk>", and sequences that do not reach a fixed length are filled with "<pad>". Stack transformer encoder self-attentive header is set to four and the number of layers is four. For the ambiguity embedding, the ambiguity level is initialized using vectors and the dimension is chosen to be four. ReLU is chosen as the activation function, dropout is set to 0.1 in the prediction layer, and the softmax function is applied. The batch size is 128, the learning rate is lr = 5×10^{-6}, and learning rate decay is applied. The cross-entropy loss function is used as the loss function, the Adam optimizer is used for optimization, and the training epoch is 10.

5.4. Results and Analysis

Three sets of experiments were performed, and the baseline for each set is constructed as the following:

1. CNN+HN+F: the model settings are set according to the original text, the convolutional neural network filter size is (5–7), and the number of highway layer layers is three.
2. MAIS: the model is designed according to the original text, the snippet embedding adopts BERT, the semantic embedding adopts GloVe word embedding, and the convolutional neural network is applied to obtain the snippet embedding.
3. ABML: The model is fine-tuned on the original basis according to the characteristics of the dataset, and the other two datasets are trained together with the pun dataset.

4. BERT and XLNet: it is very convenient to train the pretrained model by using the pretrained model provided by Hugging Face [44] and the interface provided by the transformers package, choosing the hidden vector as the pretrained model of 768.

5.4.1. Results

The results of the model in the three datasets are shown in the following three tables (Tables 2–4).

Table 2. Evaluation results of each model on Pun of the Day dataset.

Model	Acc	P	R	F1
CNN+HN+F(2018)	0.894	0.866	0.940	0.901
ABML(2021)	0.954	0.944	0.926	0.935
Bert(2018)	0.878	0.867	0.909	0.877
XLNet(2019)	0.988	0.986	0.984	0.992
MAIS(2021)	0.748	0.747	0.740	0.744
MLSN	**0.994**	**0.996**	**0.989**	**0.994**

Table 3. Evaluation results of each model on 200k-Oneliners dataset.

Model	Acc	P	R	F1
CNN+HN+F(2018)	0.943	0.955	0.930	0.943
ABML(2021)	0.955	0.957	0.954	0.955
Bert(2018)	0.985	0.987	0.983	0.085
XLNet(2019)	0.983	0.985	0.982	0.983
MAIS(2021)	0.959	0.961	0.958	0.960
MLSN	**0.986**	**0.988**	**0.984**	**0.986**

Table 4. Evaluation results of each model on SemEval-2021 Task 7 dataset.

Model	Acc	P	R	F1
CNN+HN+F(2018)	0.771	0.834	0.784	0.808
ABML(2021)	0.920	0.933	0.931	0.936
Bert(2018)	0.918	0.935	0.956	0.935
XLNet(2019)	0.921	0.944	0.927	0.935
MAIS(2021)	0.939	**0.958**	0.941	0.950
MLSN	**0.954**	0.945	**0.979**	**0.963**

The confusion matrix for the model on the three data sets is shown below (Figures 2–4):

Figure 2. Confusion matrix of each model on Pun of the Day dataset.

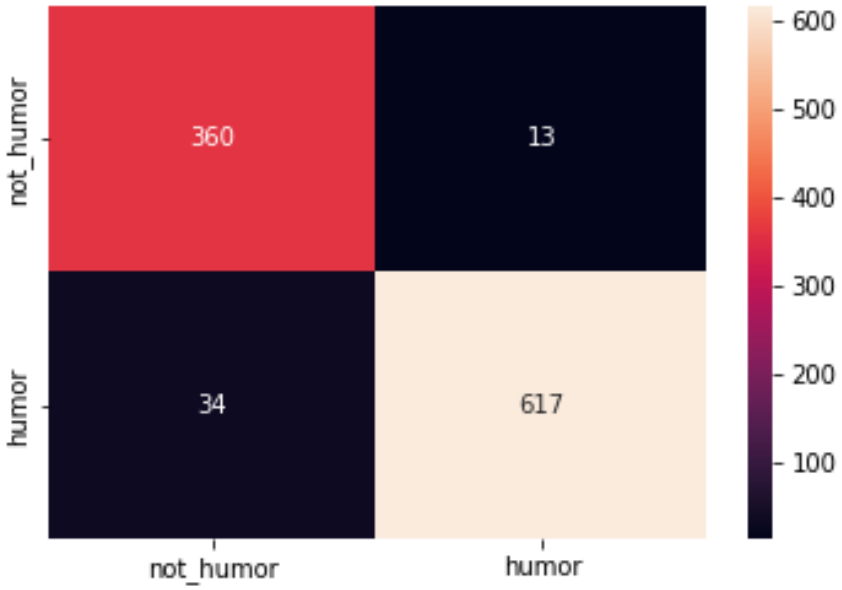

Figure 3. Confusion matrix of each model on model on SemEval-2021 Task 7 dataset.

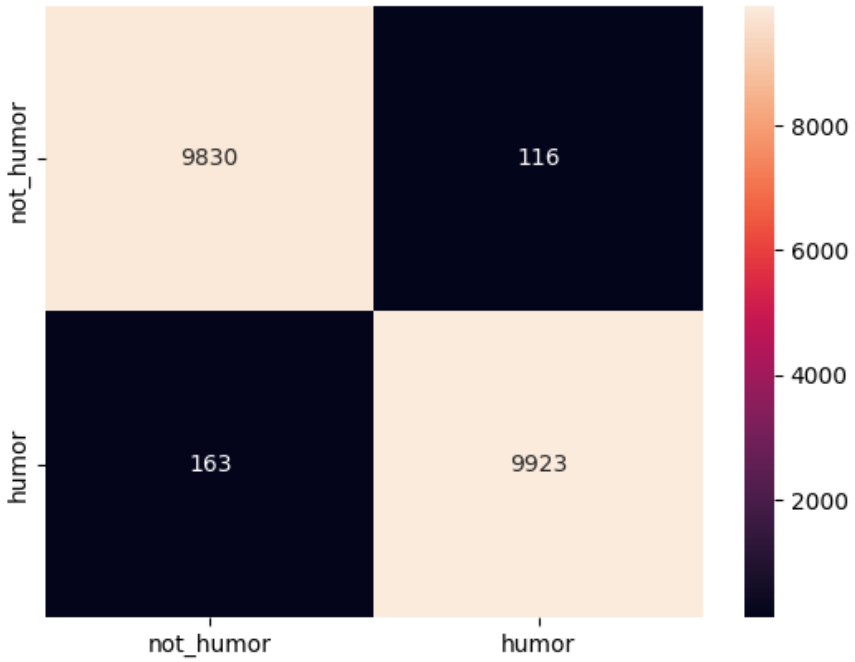

Figure 4. Confusion matrix of each model on model on 200k-Oneliners dataset.

5.4.2. Analysis

Table 2 below shows the results of the five baseline models and the model designed in this paper implemented on puns. From Table 2, it can be seen that the performance of the model designed in this paper is superior to all other models, proving the advanced nature of the proposed model in pun recognition.

For 200K-Oneliners, the large amount of data in this dataset led to good performance for each model, with BERT outperforming XLNet; although the difference in performance is small, the model in this paper is slightly better than XLNet.

The performance of each model in SemEval-2021 Task 7 is shown in Table 4. In this dataset, XLNet performs slightly better than BERT, but still not as well as the model in this paper, which again reflects that, in the field of humor recognition, extracting the deeper semantics of humor can better improve the performance of humor recognition.

By testing the model proposed in this paper and the most commonly applied language models on the above three datasets, it is concluded that in the field of humor recognition, the ability of humor recognization can be effectively improved by extracting semantic features based on humor theories, and the inconsistency of humor is better reflected not only between words but also between snippets.

6. Conclusions and Future Work

The main idea of this paper is to automatically recognize humor by extracting three types of features, namely, joke snippet inconsistency, phonetic features, and ambiguity, as well as sentence snippet embedding. These features are implemented based on humor theories. For each feature, this paper proposes relevant methods for extracting the corresponding features and combines these methods together to implement a compositive model. The model is validated on three publicly available saucy datasets and compared with the most popularly applied language models. Secondly, using the most commonly used pretrained language model word embedding can capture the semantics, proving that high-dimensional word vectors can express deeper lexical information, which is helpful for humor recognition.

However, this method still has some shortcomings; for example, the extraction method of speech features is still not comprehensive enough. Is there a better method? There is also no unified dataset standard for researchers to use. These shortcomings are to be corrected in future research.

In the future, we will continue to explore more and more effective features about humor based on humor theories and apply them to deep learning humor recognition, such as syntactic features, lexical features of jokes, etc., which are worthy research directions.

Author Contributions: Conceptualization, R.W., X.H. and Z.C.; Data curation, S.X.; Investigation, S.X., R.W., X.H. and Z.C.; Methodology, S.X. and R.W.; Project administration, S.X.; Supervision, R.W. and X.H.; Writing—original draft, S.X.; Writing—review and editing, X.H. and Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gironzetti, E.; Attardo, S.; Pickering, L. Smiling and the negotiation of humor in conversation. *Discourse Process.* **2019**, *56*, 496–512.
2. Fan, X.; Yang, L.; Lin, H.; Diao, Y.; Shen, C.; Chu, Y. Humor Recognition Based on Latent Semantic Features. *Chin. J. Inf.* **2021**, *35*, 38–46. [CrossRef]
3. Mihalcea, R.; Strapparava, C. Making computers laugh: Investigations in automatic humor recognition. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 531–538.
4. Meaney, J.; Wilson, S.; Chiruzzo, L.; Lopez, A.; Magdy, W. Semeval 2021 task 7: Hahackathon, detecting and rating humor and offense. In Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Virtual Event, Bangkok, Thailand, 5–6 August 2021; pp. 105–119.
5. Grover, K.; Goel, T. HAHA@ IberLEF2021: Humor Analysis using Ensembles of Simple Transformers. In Proceedings of the IberLEF@ SEPLN, Malaga, Spain, 21–24 September 2021; pp. 883–890.
6. Liu, L.; Zhang, D.; Song, W. Exploiting syntactic structures for humor recognition. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1875-1883.
7. Liu, L.; Zhang, D.; Song, W. Modeling sentiment association in discourse for humor recognition. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 586–591.
8. Kumar, V.; Walia, R.; Sharma, S.J.M.T.; Applications. DeepHumor: A novel deep learning framework for humor detection. *Multimed. Tools Appl.* **2022**, *81*, 16797–16812.
9. Weller, O.; Seppi, K. Humor detection: A transformer gets the last laugh. *arXiv* **2019**, arXiv:1909.00252.
10. Ren, L.; Xu, B.; Lin, H.; Yang, L.J.S.C. ABML: Attention-based multi-task learning for jointly humor recognition and pun detection. **2021**, *25*, 14109–14118.
11. Chauhan, D.S.; Singh, G.V.; Majumder, N.; Zadeh, A.; Ekbal, A.; Bhattacharyya, P.; Morency, L.-P.; Poria, S. M2H2: A Multimodal Multiparty Hindi Dataset For Humor Recognition in Conversations. In Proceedings of the 2021 International Conference on Multimodal Interaction, Montreal, QC, Canada, 18–22 October 2021; pp. 773–777.
12. Hasan, M.K.; Lee, S.; Rahman, W.; Zadeh, A.; Mihalcea, R.; Morency, L.-P.; Hoque, E. Humor knowledge enriched transformer for understanding multimodal humor. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 12972–12980.
13. Yang, D.; Lavie, A.; Dyer, C.; Hovy, E. Humor recognition and humor anchor extraction. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2367–2376.
14. Annamoradnejad, I.; Zoghi, G. Colbert: Using bert sentence embedding for humor detection. *arXiv* **2020**, arXiv:2004.12765.
15. Samson, M.; Gifu, D. FII FUNNY at SemEval-2021 Task 7: HaHackathon: Detecting and rating Humor and Offense. In Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Virtual Event, Bangkok, Thailand, 5–6 August 2021; pp. 1226–1231.
16. Hui, C.; Xiong Y. A review of western humor theory. *Foreign Lang. Study* **2005**, 5–8.[CrossRef]
17. Ortega-Bueno, R.; Muniz-Cuza, C.E.; Pagola, J.E.M.; Rosso, P. UO UPV: Deep linguistic humor detection in Spanish social media. In Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018)

Co-Located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, 18 September 2018; pp. 204–213.
18. Kiddon, C.; Brun, Y. That's what she said: Double entendre identification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 89–94.
19. Zhang, R.; Liu, N. Recognizing humor on twitter. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 889–898.
20. Purandare, A.; Litman, D. Humor: Prosody analysis and automatic recognition for f* r* i* e* n* d* s. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; pp. 208–215.
21. Mihalcea, R.; Strapparava, C.; Pulman, S. Computational models for incongruity detection in humor. In Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, Iasi, Romania, 21–27 March 2010; pp. 364–374.
22. Bertero, D.; Fung, P. Deep learning of audio and language features for humor prediction. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portoroz, Slovenia, 23–28 May 2016; pp. 496–501.
23. De Oliveira, L.; Rodrigo, A. Humor detection in yelp reviews. *Retrieved Dec.* **2015**, *15*, 2019.
24. Mao, J.; Liu, W. A BERT-based Approach for Automatic Humor Detection and Scoring. In Proceedings of the IberLEF@ SEPLN, Bilbao, Spain, 24 September 2019; pp. 197–202.
25. Ma, J.; Xie, S.; Jin, M.; Lianxin, J.; Yang, M.; Shen, J. XSYSIGMA at SemEval-2020 task 7: Method for predicting headlines' humor based on auxiliary sentences with EI-Bert. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Online, 12–13 December 2020; pp. 1077–1084.
26. Wu, J.; Lin, H.; Yang, L.; Xu, B. MUMOR: A Multimodal Dataset for Humor Detection in Conversations. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Qingdao, China, 13–17 October 2021; pp. 619–627.
27. Blinov, V.; Bolotova-Baranova, V.; Braslavski, P. Large dataset and language model fun-tuning for humor recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4027–4032.
28. Raskin, V. Semantic mechanisms of humor. In Proceedings of the Annual Meeting of the Berkeley Linguistics Society, Berkeley, CA, USA, 19–21 February 1979; pp. 325–335.
29. Braslavski, P.; Blinov, V.; Bolotova, V.; Pertsova, K. How to evaluate humorous response generation, seriously? In Proceedings of the 2018 Conference on Human Information Interaction & Retrieval, New Brunswick, NJ, USA, 11–15 March 2018; pp. 225–228.
30. Sperber, D.; Wilson, D. Relevance: Communication and cognition. In *Citeseer*; Wiley: Hoboken, NJ, USA, 1986; p. 142.
31. Hasan, M.K.; Rahman, W.; Zadeh, A.; Zhong, J.; Tanveer, M.I.; Morency, L. Ur-funny: A multimodal language dataset for understanding humor. *arXiv* **2019**, arXiv:1904.06618.
32. Attardo, S.; Raskin, V. Script theory revis (it) ed: Joke similarity and joke representation model. *Humor* **1991**, *4*, 293–348.
33. Cao, D. Self-Attention on Sentence Snippets Incongruity for Humor Assessment. *J. Phys.: Conf. Ser.* **2021**, *1827*, 012072.
34. Zhuang, L.; Wayne, L.; Ya, S.; Jun, Z. A Robustly Optimized BERT Pre-training Approach with Post-training. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, Hohhot, China, 13–15 August 2021; pp. 1218–1227.
35. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
36. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
38. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1789.
39. Chen, L.; Lee, C. Predicting audience's laughter using convolutional neural network. *arXiv* **2017**, arXiv:1702.02584.
40. Mnih, V.; Heess, N.; Graves, A. ecurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014.
41. Chen, P.-Y.; Soo, V.-W. Humor recognition using deep learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 113–117.
42. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015.
43. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.
44. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M. Huggingface's transformers: State-of-the-art natural language processing. *arXiv* **2019**, arXiv:1910.03771.