Homographic Puns Recognition Based on Latent Semantic Structures

Yufeng Diao[1,2], Liang Yang[1], Dongyu Zhang[1], Linhong Xu[3], Xiaochao Fan[1],

Di Wu[1], Hongfei Lin[1,*]

[1]Dalian University of Technology, Dalian, 116024, P.R.China
[2]Inner Mongolia University for Nationalities, Tongliao, 028043, P.R.China
[3]Dalian University of Foreign Languages, Dalian, 116044, P.R.China

Abstract. Homographic puns have a long history in human writing, being a common source of humor in jokes and other comedic works. It remains a difficult challenge to construct computational models to discover the latent semantic structures behind homographic puns so as to recognize puns. In this work, we design several latent semantic structures of homographic puns based on relevant theory and design sets of effective features of each structure, and then we apply an effective computational approach to identify homographic puns. Results on the SemEval2017 Task7 and Pun of the Day datasets indicate that our proposed latent semantic structures and features have sufficient effectiveness to distinguish between homographic pun and non-homographic pun texts. We believe that our novel findings will facilitate and stimulate the booming field of computational pun research in the future.

Keywords: homographic; puns; latent semantic structures;

1 Introduction

A pun is a form of wordplay in which one signifier suggests two or more meanings by exploiting polysemy, or phonological similarity to another signifier, for an intended humorous or rhetorical effect. Meantime, puns are a common source of humor in jokes and other comedic works. In literature, speeches and slogans, puns are also standard rhetorical ploys, where they can also be used non-humorously. For example, Sumerian cuneiform and Egyptian hieroglyphs were originally based on punning systems [1], and Shakespeare is famous for his puns [2] even in his non-comedic works. As we known, both humorous and non-humorous puns offer an interesting subject for extensive study, which leads to insights into the nature of wordplay and double-meaning.

The task of pun classification is significant in NLP and a number of relevant studies have focused on this. Such as Redfern divides puns into homophonic puns and homographic puns, which uses homonyms and the polysemy of the word respectively [3]. Delabastita [4] also classifies puns into four types: homonyny, homophony, honography, and paronyny. Most work is referenced with the classification system of Redfern, our work is also based on their analysis.

Homographic puns and homophonic puns have their own characteristics. One is to solve the problem about synonyms and the other is to solve the problem about homonyms. It cannot use the same model to distinguish the two types of puns. In our research, we mainly focus on homographic puns because they are most commonly used and easily accessible in existing text corpora. However, the homographic puns of the current works are not systematically deduced and interpreted from the features dimension.

To tackle the problem, we propose a computational semantic model to recognize homographic puns according to the related theory. This work is not the first to deal with puns recognition, but it is the first of its kind to recognize homographic puns with the four latent semantic structures on theory motivated feature design and analysis. Our contributions are listed in the following.

- The paper systematically derives the latent semantic structures behind homographic puns from puns theories, covering the four structures to affect factors.
- The paper identities sets of optimized and induced characteristics of each structure that distinguish homographic puns from non-homographic puns.
- Results on the datasets of SemEval Task7 and Pun of the Day show that our method is effective to recognize homographic puns.

2 Related Work

Puns have been discussed in rhetorical and literary criticism since ancient times, and in recent years have increasingly become a respectable research topic. Therefore, it is surprising that they have attracted little attention in the fields of computational linguistics and natural language processing [5]. In this section, we mainly review some previous work that is relevant to ours.

Some researchers studying puns tend to have a phonological or syntactic puns rather than semantic puns. Justine [6] proposed a computational model of linguistic humor in Puns, which enable powerful explanatory measures from the dimensions of ambiguity and distinctiveness. Aaron [7] considered that puns create humor through the relationship between a pun and its phonologically similar target. All of these are analyzed as phonological puns from ambiguity and so on.

Recently, Miller and Gurevych [8] proposed methods for homographic puns to identify the double meanings from word sense disambiguation. YuHsiang Huang [9] introduced a novel framework which considered positions as the important indicators for homographic pun location identification. However, the homographic puns in most of those works are not systematically deduced and interpreted from the features dimension.

Compared with puns recognition, puns generation has received quite a lot attention in the past decades [10,11]. Hempelmann [10] created a theory to model the factors to imperfect punning and outline the implementation of this measure for the evaluation of possible puns. Bryan [11] presented T-PEG, a system that utilized phonetic and semantic linguistic resources to automatically extract word relationships in puns automatically and store the knowledge in template form.

The application of puns in humor is also one of the focuses of this study. Taylor and Mazlack [12] proposed an N-gram approach based on the fixed syntactic context for identifying when puns are utilized for humorous effect in English jokes. Similar work can also be found in Taylor [13], which described humor recognition relying on Ontological Semantics by transforming content. Yang et al. [14] treated humor detection as a classification task, which identifies several semantic structures and applies a useful approach to recognizing humor.

It is an important research question with several real-world applications. For example, puns are particularly common in the advertising, where used not only to create humor but also to induce in the audience a valenced attitude toward the target [15,16]. It has often been argued that humor can enhance human-computer interaction [17] and appending the canned humor into a user interface can increase user satisfaction [18]. Puns are often used in a second language classroom. Mormot et al. [19] thought puns are useful teaching tools to improve the level of English for students. Although puns are often used in many discourse types, the applications cannot deal with them very well because of the ambiguity.

3 Features of Latent Structures behind Puns

In this section, we formulate homographic puns as a traditional classification problem. We propose the latent structures behind homographic puns in four aspects to compute and detect homographic puns: (a) Inconsistency; (b) Ambiguity; (c) Emotion and (d) Linguistic. For each latent structure, there is a list of features to capture the latent accessible indicators of homographic pun recognizing.

3.1 Inconsistency Structure

In Wales' point of view [20], the starting point of puns is that the speaker tries to use different meanings to produce something. According to philosopher Grice [21], he found that people in actual verbal communication do not always strictly abide by this principle and sometimes violates it either naturally or half unconsciously. So that this inconsistency is an important cause of the phenomenon of puns. A pun arises from the view of two or more incongruous and inapposite circumstances, considered as united in a complex object or assemblage.

For example, “Money doesn't grow on trees. But it blossoms at our branches.”

The following “Money doesn’t grow on trees” and “blossoms at our branches” example presents an inconsistency structure, which analyzed the effect of a pun.

We design two types of features, Separation and Repetition, to measure the semantic distance between word pairs in a sentence. The inconsistency of the pun can be seen as semantic incoherence, analyzed by semantic distance differences in puns, which can be calculated by Word Embedding and N-gram Language Model.

Word Embedding represents semantic information in lower dimensional dense space. The paper used Word2Vec[1] for Word Embedding. Meantime, training Language Model (LM) is a way to collect rules by utilizing the fact that words do not appear in an arbitrary order. We used the KenLM Toolkit [22] to train the N-gram LMs built from the external corpus, newswire sections of Brown corpus [23].

- Separation/Repetition: we compute the maximum/minimum semantic distance of word pairs in a sentence. This way we gain the Separation/Repetition feature by utilizing Word2Vec to compute the cosine similarity. The formula is as follows.

$$\mathrm{similarity(A,B)} = \frac{A \cdot B}{\|A\|_2 \cdot \|B\|_2} \tag{1}$$

- Semantic Coherence: we compute the score to measure the semantic coherence in a sentence by utilizing LM according to KenLM Toolkit.

3.2 Ambiguity Structure

Ambiguity means that a word may have multiple meanings [24], which represents the presence of incongruous sentence meanings is a critical component of many puns [8]. The pun is a clever intention to let one word relate to two aspects. For ambiguity of puns, the main reason is that the word has the meaning of the surface but is forced to produce another deeper and obscurer meaning structures because of the constraints of the pun context as shown in the example below. “Before he sold Christmas trees, he got himself spruced up.” In this sentence, we find that the word spruced not only has the meaning of the spruce tree but also has the meaning of making yourself or something look neater and tidier.

The multiple possible meanings of words supply people with different comprehensions. We apply the lexical resource WordNet[2] to obtain the ambiguity of the sentence. Firstly, we use an NLTK POS tagger to distinguish noun, verb, adjective, and adverb which mainly representing the ambiguity of the pun [5]. We calculate the semantic dispersion of a word combined with POS information as

$$\mathrm{PSD} = \frac{1}{P(|S_{pos}|,2)} \sum_{S_i,S_j \in S_{pos}} d(S_i, S_j) \tag{2}$$

[1] https://code.google.com/p/word2vec/
[2] http://www.nltk.org/howto/wordnet.html

where S_{pos} is the specific POS set of synsets (s_0, …, s_n) for a word which POS information is the same in a sentence; $P(|S_{pos}|,2)$ means the collection of two words from the synonym each time, and $d(s_i, s_j)$ is the length of the hypernym path between synsets (s_i, s_j) by WordNet. Then sum the semantic dispersion of the words in a sentence and divide by the sentence length. The features are as follows.

- Sense Farmost/Average/Closest: we also use an NLTK POS tagger to identify noun, verb, adjective, and adverb words. Then, we compute the largest/average/smallest Path Similarity of any of the above word senses according to the corresponding POS in a sentence [25].

3.3 Emotion Structure

Puns can produce euphemistic, subtle, and humorous effects. For example, Mulken [26] found that a pun in an advertisement is a way of humor so that the utterances can give listeners a pleasant experience. The friendly feeling may increase the audience's positive feelings and recognition of the advertised product. It is a fact that a pun is essentially associated with sentiment and subjectivity. For instance, a sentence is to be identified as a pun if it contains some words with a strong sentiment, such as “charged” as follows.

“The two guys caught drinking battery acid will soon be charged.”

Each word relating with positive or negative sentiments is the emotional reflection of the writer. To identify the word-level sentiment and affect, we utilize the open resource SenticNet [27], which provides annotations and rich effective information with measuring the subjectivity and sentiment of words. This resource enables us to design features of two types: polarity and sentics.

- Polarity: we compute sum of polarity scores, average of polarity scores, total absolute polarity scores, and average absolute polarity scores for all the words.
- Sentics: we respectively calculate the total score, average score, total absolute score, and average absolute score of all the words for the above four dimensions.

3.4 Linguistics Structure

Because our target texts are very short consisting of one or two sentences, we adopt the word-level syntax such as POS tagger, location, sentence length, and semantic information. For each aspect, we design useful features to capture the latent semantic information.

POS Feature. Each pun contains exactly one single content word (noun, verb, adjective, adverb) behind the sentence [5], which we named the candidate pun words. For example, “Boyle said he was under too much pressure.” Here, the pun word is “pressure” which is a noun. According to this, we use the POS tagger of NLTK to analyze the text. They also affect the semantic match information which we will introduce as below. POS features are as follows.

- Candidate pun word numbers: we compute the candidate pun word numbers of noun, verb, adjective, and adverb.
- Ratios of POS words: we compute the ratios of noun, verb, adjective, and adverb in a sentence.

Position Feature. In accordance with Miller [5], most puns are located towards the end of the context. Therefore, the position of the candidate pun words can affect the judgment of Puns. For example, "Here is how the track meet is going to run." The word "run" is the pun word whose location is at the end of the sentence. The features are as below.

- Position largest/smallest/average: we compute the largest/smallest/average position of the candidate pun words in a sentence.

Sentence Length. Barbieri and Saggion [28] proposed that the structural information is useful to measure the difference between instances. Sentences with different lengths will have a certain influence on whether or not they are puns.

- Sentence length: we calculate the length of any sentence.
- Sentence difference: we calculate the length difference from the previous sentence and average word length.

Semantic Information. We utilize the WordNet to analyze the ambiguity of the puns in section 4.2. In addition, we also capture the matching relationship and antonymy relationship between words in a sentence.

First, we consider the matching relation between noun and noun, verb and verb, adjective and adjective, adverb and adverb. Because the candidate pun words are from them, they have a latent semantic relation with the same type words. The semantic similarity can be computed by WordNet. For example, "I used to be a banker but I lost interest." The word "interest" is pun word. This word has two meanings: benefit and savor. Here "interest" is the meaning of benefit. We could compute the semantic similarity between (used, lost) and (banker, interest).

Then, we also measure the antonyms relation among the candidate pun words. The antonyms of the word "fall" are "ascent," "rise," "ascend," and "increase" according to WordNet. The detailed features are shown as follows.

- Largest similarity: we compute largest Path Similarity by matching relation between noun and noun, verb and verb, adjective and adjective, adverb and adverb.
- Antonym existence: we compute the existence of antonyms among the candidate pun words in a sentence.
- Antonyms largest/average: we compute the largest/average antonyms number of the candidate pun words in a sentence.

4 Experiments

We consider homographic puns recognition as a traditional text classification problem. In this section, we verify the performance of the disparate latent semantic structures we extracted on homographic puns recognition.

4.1 Experimental Setting

In this section, we first analyze the datasets used in our experiments, then introduce the evaluation metrics and baseline methods, and finally present the details of the training process of our proposed model.
Datasets To validate the effectiveness of the proposed model, we conduct experiments on two datasets: SemEval Task7[3] and Pun of the Day[4].
SemEval-2017 Task7 Data. This task is to detect and interpret English puns, containing homographic and heterographic puns. As our research interests are in lexical semantics rather than phonology, we focus on homographic puns, which are those described by Mill and Turković [5]. It contains punning and non-punning texts. Each text contains a maximum of one pun. Table 1 provides a detailed statistical description to our datasets.
Pun of the Day. The Pun of the Day dataset only includes pun text. To obtain negative samples for the pun classification task, this dataset collected the negative samples from four resources, namely AP News[5], New York Times, Yahoo! Answer[6], and Proverb. Table 1 provides a detailed statistical description to our datasets.

Table 1. Statistics on SemEval Task7 and Pun of the Day Datasets

Dataset	#Positive	#Negative	Average Length	Average Length of Positive Puns	Average Length of Negative Puns
Task7	1607	643	13.1	13.9	10.8
Pun of the Day	2423	2403	13.5	12.2	13.8

Metrics The standard precision, recall, accuracy, and F1 measures which is utilized in Semeval2017 task7 evaluation is adopted as the metrics.
Baselines We compare the following baseline methods.

- Bag of Words (**BOW**): The BOW is used to capture a series of words in a sentence which should distinguish pun and non-pun of homographic puns.
- Language Model(**LM**): The LM allocates a pun/non-pun probability based on a statistical method to the words of a sentence through probability distributions. It does not need a classifier to train the corpus.

3 SemEval2017 Task7:http://alt.qcri.org/semeval2017/task7/
4 Pun of the Day: http://www.punoftheday.com/
5 http://hosted.ap.org/dynamic/fronts/HOME?SITE=AP
6 https://answers.yahoo.com/

- AVGWord2Vec: It presents the average word embedding of a sentence according to the distributional latent semantic meaning representation [29].
- HPCF: Here, we denote the combination of the four latent structures as Homographic Puns Core Features (**HPCF**).
- AVGWord2Vec_ HPCF: Here, we combine HPCF with AVGWord2Vec which having a well performance of this task.

Training Details The paper conducts 5 fold cross-validation experiments, each using 60% of the samples for training a detecting model, 20% for estimating the parameters, and 20% for predicting new samples. The training corpus of word embedding is from Wiki. The dimension of word embeddings is 300.

We choose Gradient Boosted Decision Tree (GBDT), a powerful boosting method based on decision trees as our classification algorithm. This is consistent with Zhang and Liu's [30].

4.2 Homographic Puns Recognition

We investigate how the combination of the latent semantic structures performs compared with our suggested baselines and the results are presented in Table 2.

Table 2. Comparison of Different Methods of Homographic Puns Recognition

	SemEval2017 task7				Pun of the Day			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
HPCF	0.796	0.938	0.808	0.861	0.730	0.807	0.702	0.767
Bag of Words(BOW)	0.768	0.847	0.832	0.806	0.709	0.663	0.732	0.685
Language Model(LM)	0.588	0.774	0.688	0.668	0.510	0.764	0.508	0.612
AVGWord2Vec	0.716	0.800	0.803	0.756	0.901	0.899	0.903	0.900
BOW_HPCF	0.802	**0.954**	0.805	0.871	0.907	0.905	0.908	0.906
AVGWord2Vec_HPCF	**0.836**	0.944	**0.845**	**0.887**	**0.914**	**0.906**	**0.920**	**0.910**

First, HPCF contains Inconsistency, Ambiguity, Emotion, and Linguistic structures has a better performance for homographic puns recognition compared with BOW and LM. This proves that the latent semantic structure derived from the theory has enough rationality. The inappropriate LM also demonstrates, which we relieve the specific domain differences and capture the real puns. The inadequacy of BOW also indicates that we can understand the order in which words appear in the original sentence.

Second, BOW_HPCF, which is the combination of BOW and HPCF, is superior to BOW and HPCF in the two datasets. The reason is that it contains sufficient latent semantics information and the order of words in a sentence. But BOW_HPCF is inferior to AVGWord2Vec_HPCF. Because it involves enough latent structures such as Ambiguity structure but not enough distributional semantics.

Last, AVGWord2Vec_HPCF, which achieved 0.91 F-score, has the best classification performance in Pun of the Day. The reason is that this combination takes into consideration the latent semantic structures and semantic word meanings. In SemEval2017 task7, this conclusion is almost coincident besides that BOW_HPCF has the best recall. From the results, it finally indicates that our proposed latent semantic structures are efficient in interpreting homographic puns in depth.

The best performing system in the Semeval2017 Task7 is Fermi [31]. It also casts this problem as a supervised learning classification problem. This model uses a recurrent neural network to train the classifier. The result achieves 0.899 by F1-score. So in the future, we will try deep learning methods to settle this problem.

4.3 The Effect of Latent Semantic Structures

We examine the performance of above different structures by the same classifier GBDT on two datasets. To ensure fairness, we do not adjust any parameters here. We explore that how the different latent semantic structures affect homographic puns recognition performance and display the results in Figure 1. We have the following observations:

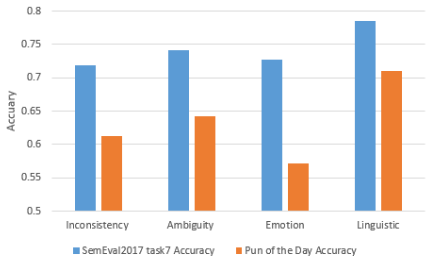

Fig. 1. Different Latent Structures' Contribution to Homographic Puns Recognition

First, according to the results, we can see that all the latent structures in the two datasets have a consistent performance. It is obvious that Linguistic structure performs the best among all the latent semantic structures in the two datasets. The reason is that puns are closely related to the location, part of speech, and collocation of ambiguous words.

Second, the performance of Ambiguity structure in the two datasets ranks second, which showed most puns are well structured and handled with multiple meanings or aspects. It is evident Emotion Structure performs the worst in Pun of the Day because the emotion of homographic puns is harder to mine and analyze.

Last, different from the Pun of the Day dataset, the worst performance of the structures in the SemEval task7 is the Inconsistency structure. The reason is the contrasting or incongruous meaning of the Inconsistency structure which played is puzzled to find abundant and useful information. This demonstrates that homographic puns latent structures are not expressed similarly in different datasets.

Then, it also examines the relevance among the four latent structures and conduct ablation experiments to examine the recognition. Each time, we remove one or two structures and observe how the performance changes. It summarizes the results in Table 3 and have the following observations.

Table 3. The Relevance of Latent Structures to Effect Homographic Puns Recognition.

	SemEval2017 task7				Pun of the Day			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
All	**0.796**	**0.938**	**0.808**	**0.861**	**0.730**	**0.807**	**0.702**	**0.767**
All–Inconsistency	0.783	0.928	0.800	0.849	0.718	0.791	0.692	0.753
All–Ambiguity	0.771	0.924	0.790	0.841	0.713	0.783	0.689	0.746
All–Emotion	0.789	0.934	0.803	0.855	0.730	0.803	0.702	0.765
All–Linguistic	0.759	0.931	0.776	0.836	0.659	0.733	0.641	0.694
Emotion+Linguistic	0.770	0.923	0.790	0.840	0.704	0.783	0.679	0.741
Ambiguity+Linguistic	0.780	0.929	0.797	0.848	0.725	0.809	0.697	0.765
Ambiguity+Emotion	0.751	0.927	0.770	0.830	0.656	0.729	0.638	0.691
Inconsistency+Linguistic	0.785	0.921	0.806	0.848	0.723	0.784	0.700	0.752
Inconsistency+Emotion	0.738	0.909	0.767	0.815	0.610	0.682	0.598	0.644
Inconsistency+Ambiguity	0.756	0.914	0.781	0.828	0.667	0.739	0.647	0.701

First, all the latent structures used together outperforms the other combinations in the two datasets. It demonstrates that the mutual interaction and influence of the whole latent structures can more effectively recognize the homographic puns. It also proves that our structures based on the homographic pun theory, which systematically derived and explained, are valid.

Second, for three structures working together to effect the recognition, All–Emotion (contains Inconsistency, Ambiguity, and Linguistic structure) performs the best in the two datasets, which means that Emotion offers little effective information to help the detection of puns, and meanwhile Emotion could be related loosely with other structures. In contrast with All–Emotion, the performance of All–Linguistic (contains Inconsistency, Ambiguity, and Emotion structure) in the two datasets performs worst. It is consistent with the above part that Linguistic knowledge is very important to identify puns and could be effectively matched with other latent structures.

Last, we validate how well two structures effect the detection of homographic puns between the latent structures. The results demonstrate that different collocation structures are represented differently in various contexts. In the SemEval2017 task7, Ambiguity+Linguistic and Inconsistency+Linguistic have the best performance meanwhile. That means Linguistic is related with Inconsistency or Ambiguity more than Emotion. In the Pun of the Day, Ambiguity+Linguisticalso has the strongest performance and Inconsistency+Linguistic comes second. As shown in Table 3, the rest of the distributions of puns recognition are generally consistent. In the two datasets, Inconsistency+Emotion has the worst performance, so we conjecture that Inconsistency and Emotion put together may hurt the performance.

5 Conclusion and Future work

In this work, we focus on understanding homographic pun language through homographic puns recognition. For this purpose, we presented a computational and effective approach to identify puns. We proposed four latent semantic structures behind the homographic puns based on relevant theory. In view of the designed sets of effective features related with each structure, we established different computational classifiers considering the association among the four structures to identify puns. The experimental results conducted on the two datasets show that our proposed latent semantic structures have sufficient effectiveness. The performances on homographic puns recognition are superior compared with several baselines.

As future work, we would like to find the characteristics of homographic and homophonic puns, employ the deep learning methods to recognize the puns, and then apply our discoveries to the procedure of automatic generation of puns. Those are all promising jobs we can pursue in the future.

Acknowledgments

This work is partially supported by grant from the Natural Science Foundation of China (No. 61632011, 61702080, 61602079), the Fundamental Research Funds for the Central Universities (No. DUT16ZD216, DUT17RC(3)016).

6 References

1. John Pollack. The Pun Also Rises[J]. Penguin Publishing Group. 2011.
2. Keiko Tanaka. The pun in advertising: A pragmatic approach[J]. *Lingua*, 1992,87(1):91–102.
3. Redfern W. Puns[J]. The Scriblerian and the Kit-Cats, 1987, 19(2): 204.
4. Delabastita D. Focus on the pun: wordlplay as a special problem in translation studies[J]. Target, 1994, 6(2): 223-243.
5. Tristan Miller and Mladen Turković. Towards the automatic detection and identification of English puns[J]. *European Journal of Humour Research,2016,*4(1):59–75.
6. Kao J T, Levy R, Goodman N D. A Computational Model of Linguistic Humor in Puns[J]. Cognitive science, 2015.
7. Jaech A, Koncel-Kedziorski R, Ostendorf M. Phonological Pun-derstanding[C]. Proceedings of NAACL-HLT. 2016: 654-663.
8. Miller T, Gurevych I. Automatic disambiguation of English puns[C]//ACL (1). 2015: 719-729.
9. Huang Y H, Huang H H, Chen H H. Identification of Homographic Pun Location for Pun Understanding[C]. Proceedings of the 26th International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee, 2017: 797-798.
10. Hempelmann C F. Paronomasic puns: Target recoverability towards automatic generation[J]. Dissertation Abstracts International, 2003,64(11):4029.
11. Hong B A, Ong E. Automatically extracting word relationships as templates for pun generation[C]. The Workshop on Computational Approaches To Linguistic Creativity. Association for

Computational Linguistics, 2010:24-31.
12. Julia M. Taylor and Lawrence J. Mazlack. Computationally recognizing wordplay in jokes. In Proceedings of the 26th Annual Conference of the Cognitive Science Society[C] (CogSci 2004), 2004:1315–1320.
13. Julia M Taylor. Computational detection of humor: A dream or a nightmare? the ontological semantics approach[C]. In Proceedings of the 2009 ACM International Joint Conference on-Web Intelligence and Intelligent Agent Technology, 2009(03): 429–432
14. Yang D, Lavie A, Dyer C, et al. Humor Recognition and Humor Anchor Extraction[C]. EMNLP. 2015: 2367-2376.
15. Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. Textual affect sensing for computational advertising[C]. In Proceedings of the AAAI Spring Symposium on Creative Intelligent Systems, 2008: 117–122, March.
16. Monnot, M. Puns in advertising: Ambiguity as verbal aggression[J]. Maledicta. 1982(6): 7–20.
17. Hempelmann, C. F. Computational humor: Beyond the pun [J]? In Raskin, V. (ed.), The Primer of Humor Research. Humor Research, Volume 8. Berlin: Mouton de Gruyter. 2008 :333–360
18. Morkes, J., Kernal, H. K. & Nass, C. Effects of humor in task-oriented human–computer interaction and computer-mediated communication: A direct test of SRCT theory[J]. Human–Computer Interaction. 1999, 14 (4): 395–435.
19. Mormot M, Adelstein A, Bulusu L. Immigrant Mortality in England and Wales 1970-1978[J]. Population Trends, 20.
20. Lems K. Laughing All the Way: Teaching English Using Puns[C]. English Teaching Forum. US Department of State. Bureau of Educational and Cultural Affairs, Office of English Language Programs, SA-5, 2200 C Street NW 4th Floor, Washington, DC 20037, 2013, 51(1): 26-33.
21. Wales K. A dictionary of stylistics[M]. Routledge, 2014.
22. Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modi- fied Kneser-Ney language model estimation[C]. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, Bulgaria, pages 690–696.
23. Henry Kucera and W. Nelson Francis. Computational Analysis of Present-day American English. Brown University Press, Providence, RI, USA. 1967.
24. Bekinschtein T A, Davis M H, Rodd J M, et al. Why clowns taste funny: the relationship between humor and semantic ambiguity[J]. Journal of Neuroscience, 2011, 31(26): 9665-9671.
25. Reyes A, Rosso P, Buscaldi D. From humor recognition to irony detection: The figurative language of social media[J]. Data & Knowledge Engineering, 2012, 74: 1-12.
26. Van Mulken M, Van Enschot-van Dijk R, Hoeken H. Puns, relevance and appreciation in advertisements[J]. Journal of pragmatics, 2005, 37(5): 707-721.
27. Cambria E, Hussain A. Sentic computing: Techniques, tools, and applications[M]. Springer Science & Business Media, 2012.
28. Barbieri F, Saggion H. Modelling Irony in Twitter: Feature Analysis and Evaluation[C]. LREC. 2014: 4258-4264.
29. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]. Advances in neural information processing systems. 2013: 3111-3119.
30. Zhang R, Liu N. Recognizing Humor on Twitter[C]. ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014:889-898.
31. Miller T, Hempelmann C F, Gurevych I. SemEval-2017 Task 7: Detection and interpretation of English puns[C]. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, BC. 2017.