

PAPER • OPEN ACCESS

The system analysis and Research based on pun recognition

To cite this article: Siyu Jiang *et al* 2021 *J. Phys.: Conf. Ser.* **2044** 012190

View the [article online](#) for updates and enhancements.

You may also like

- [A novel simulation paradigm utilising MRI-derived phosphene maps for cortical prosthetic vision](#)
Haozhe Zac Wang and Yan Tat Wong
- [Data augmentation for enhancing EEG-based emotion recognition with deep generative models](#)
Yun Luo, Li-Zhen Zhu, Zi-Yu Wan et al.
- [Neural Network Pun Material Identification Framework based on Artificial Intelligence Learning](#)
Shashi Shekhar



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

The system analysis and Research based on pun recognition

Siyu Jiang¹, Zhiheng Zhang¹, Qiong Zhong¹, Jin Xie¹, Weilin Wu^{2,*}

¹Guangzhou Key Laboratory of Multilingual Intelligent Processing, School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, 510006, PR China

²College of Electronic Information, GuangXi University for Nationalities, Nanning, China

*Corresponding author: email: wuweilin@scut.edu.cn

Abstract. The pun recognition task is divided into pun detection task and pun localization task. Puns are divided into homophonic puns and heterographic puns. Until now, researchers have proposed a variety of recognition systems that try to construct features of puns from different perspectives. In this paper, we survey and analyse most of the classical pun recognition systems experimented in SemEval 2017 shared task 7 datasets, and summarize the main challenges in pun recognition tasks so far.

1. Introduction

The pun is a kind of wordplay that uses the polysemy or homonym of a word to suggest that the sentence has two or more meanings, so as to achieve humorous effects [1]. Typically, puns are classified into homophonic puns and heterographic puns. Homographic pun is the use of polysemy of a word to construct a pun. Heterographic pun describes the case that two different words have similar sounds and conform to the same context.

Pun recognition [2] is composed of two subtasks. The first is pun detection, which decides whether a text contains any puns. The second is pun positioning, where a pun sentence is given to identify which word in the sentence is the pun.

Although the pun appears frequently in our daily life, it has not received much attention in the field of computational linguistics and natural language processing. In real life, the process of human language communication is full of ambiguity, because different persons may understand the language from different angles. The existence of ambiguity has its own meaning. This kind of ambiguity is often intentionally created when people express their thoughts euphemistically. These euphemistic expressions such as innuendo, allusion and pun not only challenge the task of Word Sense Disambiguation (WSD), but also reveal a major difficulty in machine understanding of natural language, namely, how to recognize the underlying meaning while understanding the surface meaning of text. Puns are also used in interpersonal communication and can be used to break the ice, enhance the atmosphere and relax the mood. If there are puns in the process of human-computer interaction, it makes the process more real and interesting [3]. We believe that the development of pun recognition technology can effectively promote the development of machine translation, text emotion recognition and human-computer interaction.

Pun recognition models are divided into unsupervised models and supervised models. In supervised models, some researchers use the non-neural network method, while others use neural network



method. Unsupervised models and the non-neural network method usually construct features and models based on the definition of puns, statistical laws of pun samples, or assumptions summarized from the previous two methods. The neural network method relies more on word embedding, network structure and attention mechanism.

Although interest in puns has increased since the SemEval 2017 Shared Task 7 dataset was published in 2017, the study on pun recognition methods is rare in the literature. This paper focuses on classical pun recognition models based on SemEval 2017 Shared Task 7 dataset.

The rest of the paper is organized as follows. Section 2 is an overview of pun dataset and evaluation methods. Section 3 introduces the current pun recognition methods. Section 4 investigates and analyzes pun recognition methods. Section 5 summarizes the current challenges in the field of pun recognition.

2. Overview

2.1. Data Sets

The SemEval 2017 Shared Task 7 dataset (2) is the first and the largest public open dataset in the field of pun. It subdivides multiple data sets according to different subtasks. The first data set is the homophonic pun detection data set, which consists of 1607 texts containing puns and 643 texts without puns. The second data set is the homophonic pun location data set, which consists of 1607 texts containing puns in the homophonic pun dataset. Each word in the texts in the second data set is labeled O or P, where O means the word is not a pun and P means it is a pun. The third data set is the homophonic pun detection data set, which consists of 1271 texts containing puns and 509 texts without puns. The fourth data set is the homophonic pun positioning data set, which consists of all texts containing puns in the third data set, and the annotation scheme is the same as that in the second data set. Table 1 shows some SemEval 2017 Shared Task 7 statistics. Each sample of SemEval 2017 Shared Task 7 was manually annotated by Tristan Miller et al. The publication of this data set clarified the mission and objectives of the field of pun research, and enabled the performance of pun models to be evaluated more objectively, which greatly promoted the development of the field of pun research.

Table 1: SemEval 2017 shared task 7 dataset statistics

pun type	subtask	contexts	words	words / context		
				min	mean	max
homographic	detection	2250	24499	2	10.9	44
homographic	location	1607	18998	3	11.8	44
heterographic	detection	1780	19461	2	10.9	69
heterographic	location	1271	15145	3	11.9	69

2.2. Metrics Level

In the SemEval 2017 Shared Task 7 dataset, precision (P), recall (R) and F-Score (F_1) were used to calculate the score evaluation model. For the pun detection task, the metrics of all models are calculated according to the method proposed by Manning et al. [4]. For the pun positioning task, the unsupervised model and partially supervised model calculate the score according to P, R and F_1 [5] defined in the WSD domain:

$$P = \frac{\text{number of correct guesses}}{\text{number of guesses}} \quad (1)$$

$$R = \frac{\text{number of correct guesses}}{\text{number of contexts}} \quad (2)$$

$$F_1 = \frac{2PR}{P + R} \quad (3)$$

This takes into account that the model does not necessarily select puns for all the samples tested.

3. Pun Recognition

3.1. Unsupervised Pun Model

Idiom Savant. Idiom Savant (Samuel Doogan et al.) [6] is a model for detecting puns that relies on calculating word similarity. Idiom Savant argues that the different meanings of homophonic pun can only depend on contextual information because the spelling is the same. According to the theory of Feyaerts and Brone [7], if w_i is a pun, the model assumes that the two meanings of the i^{th} word w_i in a text containing N words will have a strong relationship with $c_b = \{w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_N\}$. Following the above theory, Samuel Doogan designed a fraction calculation function to calculate a score for each word in the text. If the average score of the two highest-graded words in the text is higher than 0.6, the text is considered to contain pun. The fraction calculation function is as follows:

$$f_{ws}(x) = \begin{cases} 0, & x < 0.01 \\ 1 - x, & x \geq 0.01 \end{cases} \quad (4)$$

$$\text{score}(w_i, i) = \sum_{j=1}^n p_{ij} f_{ij} \sum_{k=1}^l \sum_{m=1}^q f_{ws} \left(\frac{g_k g_m}{|g_k| |g_m|} \right) \quad (5)$$

where n is the total number of words in c_i ; l and g are the total meanings of w_i and w_j , respectively. g_k and g_m are gloss of the k^{th} meaning of w_i and gloss of w_j respectively. Glosses are taken from WordNet [8]. $f_{ws}()$ is the damping equation, and Idiom Savant argues that puns should not be too similar or too similar to other words. p_{ij} is the damping factor, which is 0.2 if the POS (part-of-speech) tag of w_i and w_j is the same, because in most cases, puns and their base words in context do not have the same POS tag. f_{ij} is also a damping factor, and if the frequency of w_i in the word corpus exceeds 100, it is 0.1, because each high-frequency word has a certain similarity score with each other phrase. Based on the score of each word calculated by the model in detecting homophonic pun, Samuel Doogan chose the word with the highest score and the closest to the end of the sentence as the puns. Samuel Doogan designed another method of fractional calculation for homophonic puns. The matching scores of word pairs are calculated as follows:

$$\text{score} = (\text{freq}_{n\text{gram}'} - \text{freq}_{n\text{gram}}) \quad (6)$$

$$\text{ratio} = \max(\text{ratio}_{ph}, \text{ratio}_{phs}, \text{ratio}_{ch}) \quad (7)$$

$$\text{ratio}_f(w_1, w_2) = \frac{\min_{w \in w_1, w_2} \|w\|_f - d_f(w_1, w_2)}{\min_{w \in w_1, w_2} \|w\|_f} \quad (8)$$

f belongs to $\{ph, phs, ch\}$, and $\text{freq}_{n\text{gram}}$ represents the frequency of n -gram in corpus. And a score is determined by the number of identical phonemes that two words have shared:

$$d_{ph}(AO, F, AH, S, AO, R, AH, F, AH, S) = 2 \quad (9)$$

All phonemes of each word are concatenated into the entire string and the Levenshtein distance of the two phonetic strings is calculated as a score:

$$d_{phs}(\text{"AOF AHS"}, \text{"AOR AHFAHS"}) = 3 \quad (10)$$

And the Levenshtein distance of the original string of two words is calculated as the score:

$$d_{ch}(\text{"office"}, \text{"orifice"}) = 2 \quad (11)$$

The phonemes of the words are provided by The Carnegie Mellon University Pronouncing Dictionary (CMU) [9]. The combination with the highest sentence score is considered to contain puns if the score exceeds the threshold. The model selects the word with the highest score as the candidate word, and then selects the word closest to the end of the sentence as the pun.

N-Hance. N-Hance is a pun recognition model designed by Sevgili [10] based on Pointwise Mutual Information (PMI) [11]. The hypothesis of Sevgili is similar to that of Idiom Savant, but N-Hance believes that a word in the text containing the pun has a strong connection to the pun, and that the humor of the pun is based on this word. So, N-Hance measures how much each word relates to all the other

words in the text, namely, PMI. The co-occurrence frequency of words can reflect the degree of connection between words. The PMI formula is as follows:

$$\text{pmi}(w_1, w_2) = \log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)} \quad (12)$$

$p(w_1)$ is the number of occurrences of w_1 in the corpus divided by the total number of words in the corpus. If the difference between the first and second ranked PMI in the text is above the threshold, the model determines that the text contains puns. N-Hance chooses the word near the end of the sentence in the pair with the highest PMI for the pun.

UWaterloo. UWaterloo (Victimae et.al) [12] is a rules-based pun positioning model. Victimae follows the Inverse Document Frequency (IDF) of words, Normalized Pointwise Mutual Information (NPMI), the position of the word in the text, the part of speech of the word, and Some rules based on grammatical characteristics to calculate the score for each word in the text. The IDF of a word, w , is calculated as $\text{IDF}_w = \log(N/n_w)$, where n_w is the number of occurrences of w in the expected database.

3.2. Supervised Pun Model.

JU_CSE_NLP. JU_CSE_NLP (Pramanick and Dipankar et. al) [13] is a pun recognition model designed based on some statistical patterns in the dataset. The rules summarized by Pramanick and Dipankar are related to the following aspects: whether the word is a stop word, the lexical nature of the word itself, the type of sentence the word belongs to, the lexical nature of the words around the word and whether the word ends in '-ed' or '-ing'. JU_CSE_NLP calculates the probability of a word being a pun according to a probability formula based on the different rules it meets. A word contains a pun if the probability value of the most likely word in the text to be a pun is greater than 0.25. JU_CSE_NLP calculates the probability of each word to be a pun based on the process of detecting puns. If the probability value of the word with the highest probability is higher than 0.25, the word is a pun.

FELR. Feng et al. [14] understood the task of pun recognition from a feature engineering perspective by constructing different features as inputs to a logistic regression model and measuring their impact according to the weights assigned to the features. They designed the FELR to construct features based on four aspects: lexicality, representation of the sentence, post-segmentation properties of the sentence, and word embedding. The rules it finally adopts prove that the contextually informative representation of sentences, the post-partitioning features of sentences, and word embeddings indeed play a role in detecting puns. FELR uses doc2vec [15] and BERT [16] to represent sentences and words. FELR constructs different features for the localization task, and the features it uses for homophonic pun indicate that word position, similarity between words, word frequency, word properties, word embedding and sentence representation all play a positive impact on the task of localizing homophonic pun. FELR locates harmonic puns using phonologically relevant features and indicates that text segmentation plays a more active role than that of locating homophonic pun.

Fermi. Fermi (Indurthi and Oota et. al) [17] proposed the Fermi model for recognizing puns based on pre-trained word embeddings and BiDirectional Recurrent Neural Network (Bi-RNN). A major contribution of Fermi was the first application of a neural network approach to the task of pun recognition. The structure of the neural network designed by Indurthi and Oota for the task of pun detection is very simple and classical, which provides an important reference for subsequent studies. The text words are processed by the embedding layer into the Bi-RNN, and the features learned from the Bi-RNN flow into a fully connected neural network activated using sigmoid. Fermi locates puns based on the maximum cosine similarity between the synonyms of two words.

WECA. Diao et al. [18] proposed the WordNet-Encoded Collocation-Attention Network for Homographic Pun Recognition (WECA) neural network model based on the Fermi model. WECA uses a Bidirectional Long Short-Term Memory (Bi-LSTM) network instead of Bi-RNN to capture contextual information and also introduces an attention mechanism [19] to capture features. The attention mechanism is formulated as follows:

$$u_{ij} = V \cdot \tanh(W_u h_{ij} + b_w) \quad (13)$$

$$a_{ij} = \frac{\exp(u_{ij})}{\sum_{t=1}^n \exp(u_{tj})} \quad (14)$$

$$c_{ij} = \sum_{i=1}^n a_{ij} h_{ij} \quad (15)$$

n is the text length, i means the word is the i^{th} word of the text, $j \in \{\text{nouns}, \text{verbs}, \text{adjectives}, \text{adverbs}\}$, words are grouped according to lexicality, u_{ij} is the word score, a_{ij} is the word weight, c_{ij} is the overall context vector of the word with lexicality j , and h_{ij} is the Bi-LSTM representation of the i^{th} word. WECA for a word w considers not only its word embedding, but also the sum of weights of all word sense vectors in WordNet for that word. The formula is as follows:

$$W = \sum_{s_i \in S^w} \frac{|L_i(s_i)|}{m} \sum_{l_j \in L_i(s_i)} w^{l_j s_i} \quad (16)$$

s_i is word sense i , l_j represents word j , S^w is all the senses of word w , $L_i(s_i)$ is the set of all words within word sense s_i , and $w^{l_j s_i}$ represents the embedding of word l_j of word sense s_i .

GRCA and PSUGA. Diao et al. [20] proposed a contextualized-representation Gated attention network (GRCA) model for detecting homophonic pun by improving WECA. The input features vary as word embeddings and contextual embeddings, which enter two systems with similar structures that are activated by different nonlinear functions, and then the gated-attention mechanism designed by GRCA combines the features of the two systems, and finally the fully connected neural network using sigmoid activation is used for classification. GRCA uses a Gated Recurrent Unit (GRU) to capture contextual information compared to WECA. Word embedding is input to a network consisting of a Bi-GRU, a CNN and an attention mechanism, and the output is defined as N_s , with each layer activated by the tanh function. Context embedding is input to a similar network that is activated by the ReLU function, and the output is defined as M_s . The strategy of the gated attention mechanism combining the two parts of information is:

$$P = \sigma(w_g(M_s \cdot N_s) + b_g) \quad (17)$$

$$R = P \cdot M_s + (1 - P) \cdot N_s \quad (18)$$

σ is the sigmoid function. M_s , N_s and P are concatenate as inputs to the softmax function:

$$\hat{y} = \text{softmax}(\tanh(w_y[M_s; N_s; R] + b_y)) \quad (19)$$

Another model Pronunciation and Spelling Understanding Gated Attention Network (PSUGA) proposed by Diao et al. [21] was used to detect harmonic puns. Its structure is consistent with CRGA, and the biggest difference lies in the input features. To represent the characteristics of the speech of harmonic puns, PSUGA uses phoneme embedding and text multilevel embedding as features. PSUGA uses CMU to convert words into phoneme sequences. Phoneme embedding is to transform the corresponding phonemes into vectors and map the corresponding words to a high-dimensional vector space.

PCPR. Zhou et al. proposed Pronunciation-attentive Contextualized Pun Recognition (PCPR) [22], which improves on PSUGA and GRCA. The input features of PCPR are phoneme embedding and BERT-generated contextual word embedding. The phoneme embedding is first captured by the attention mechanism, then spliced with the BERT-generated contextual word embedding to form a joint phonological contextual embedding, then by the attention mechanism to form a phonologically focused contextual embedding, and finally spliced with the BERT-generated overall sentence contextual embedding to form a holistic embedding, which is input to the fully connected layer classification. It is a word-level classification task, and the speech-focused contextual embedding of each word is input to the fully-connected layer for prediction.

Joint. Joint proposed by Zou and Lu [23] considers the pun recognition task as a sequence labeling problem and thus introduces a classical sequence labeling system, the bidirectional Long Short Term Memory networks (BiLSTM) on top of the Conditional Random Fields (CRF) architecture (BiLSTM-

CRF) [24]. Joint is based on the characteristics of the dataset - the sentences contain only one pun - the {B,P,A} tagging scheme is designed.

- B represents the word that appears before the pun word in the text.
- P represents pun word.
- A represents the word that appears after the pun word in the text.

Thus, whenever a word in a text is tagged with P, it means that the text contains a pun. Joint uses multilayer embedding, i.e., spliced character embedding, position embedding, and pre-trained word embedding as features, which are input to BiLSTM and CRF. CRF is an annotation system that finds the most appropriate annotation sequence for a new input sequence according to the learned conditional probability distribution. Conditional probability is defined as:

$$P(y|w_i) = \frac{\prod_i^n \exp(w_{y_{i-1}, y_i} z_i + b_{y_{i-1}, y_i})}{\sum_{\hat{y} \in Y} \exp(w_{\hat{y}_{i-1}, \hat{y}_i} z_i + b_{\hat{y}_{i-1}, \hat{y}_i})} \quad (20)$$

Y is the set of labels i.e. {B,P,A}, z_i is the representation of the word w_i learned in the LSTM, w_{y_{i-1}, y_i} and b_{y_{i-1}, y_i} are the parameters.

CSN-ML. The Compositional Semantics Network with Multi-Task Learning for Pun Location (CSN-ML) [25] designed by Mao et al. solves the pun locality task with a multi-task learning perspective. They proposed the auxiliary task of classifying harmonic puns and homophonic puns to help the model learn more general features and use short-range semantics as input features. The method of CSN-ML to obtain short-range semantics is to divide the text into n-grams, and then use a complex-valued network [26] to extract n-gram features.

4. Analyzing

Table 2: Pun recognition results in SemEval-2017. The top three scores of each metric are shown in bold. Models that use the WSD method to measure the performance of locative puns are underlined.

Model	Homographic Puns						Heterographic Puns					
	Pun Detection			Pun Location			Pun Detection			Pun Location		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
<u>JU_CSE_NLP</u>	72.51	90.79	80.63	33.48	33.48	33.48	73.67	94.02	82.61	37.92	37.92	37.92
<u>Idiom Savant</u>	73.00	98.00	84.00	66.36	66.27	66.31	87.04	81.90	84.39	68.45	68.45	68.45
<u>UWaterloo</u>	-	-	-	65.26	65.21	65.23	-	-	-	85.02	84.82	84.92
<u>N-Hance</u>	75.53	93.34	83.56	42.69	42.50	42.59	77.25	93.00	84.40	65.92	65.15	65.53
<u>FELR</u>	92.40	93.70	93.00	76.20	76.20	76.20	92.10	93.90	93.00	84.90	84.90	84.90
Fermi	90.24	89.70	89.97	52.15	52.15	52.15	-	-	-	-	-	-
SAM	-	-	-	81.50	74.70	78.00	-	-	-	-	-	-
WECA	88.19	90.64	89.21	-	-	-	-	-	-	-	-	-
GRCA	90.96	89.88	90.42	-	-	-	-	-	-	-	-	-
PSUGA	-	-	-	-	-	-	87.92	85.04	86.46	-	-	-
CSN-ML	-	-	-	85.00	81.30	83.10	-	-	-	88.80	85.80	87.30
Joint	91.25	93.28	92.19	83.55	77.10	80.19	86.67	93.08	89.76	81.41	77.50	79.40
PCPR	94.18	95.70	94.94	90.34	87.50	88.94	94.84	95.59	95.22	94.23	90.41	92.28

To explore better pun recognition schemes, the models are analyzed in this paper. Table II shows the performance summary of all models mentioned in this paper.

The best-performing unsupervised systems in the field of pun detection are Idiom Savant and N-Hance. Idiom Savant detects puns by measuring the similarity of word meanings to contextual segments

and introduces phonetic features to detect similarity in the task of harmonic pun detection. N-Hance achieves good results for both types of puns by relying only on PMI to measure similarity. UWaterloo performs well in the area of harmonic pun localization with its clever use of phonetic features and IDF. The best performing non-neural network system, FELR, demonstrates features with positive effects that corroborate with the unsupervised system described above. And text segmentation was found to be more useful for harmonic puns. The best performing neural network model, PCPR, was found to improve the performance of the model by applying phoneme embedding to homophonic pun.

5. Current Challenges

New advances in the field of pun recognition will be achieved if the following key challenges can be addressed.

1. A better solution to the pun sample scarcity problem. The above comparison shows the effectiveness of supervised methods such as neural networks in the field of pun recognition. Such models rely on a large amount of labeled data. However, in reality, pun collection is difficult because it generally requires people with extensive knowledge to accurately determine and classify puns. The SemEval 2017 shared task 7 dataset published by Miller et al. contains a total of 4030 pun samples, which reflects the difficulty of collecting and labeling puns. Thus, the pun recognition task can be considered as a small-sample recognition task, and the effectiveness of this idea is confirmed by the excellent performance of CSN-ML and PCPR applying different transfer learning methods to the pun recognition task.

2. Comprehension and recognition of pun phrases. Joint, CSN-ML, and PCPR all consider the accurate determination of pun phrases as their major challenge. They mainly believe that the key to this challenge is that pun phrases contain too few words and lack contextual information. For the example "Follow your knows.", a harmonic pun in a sentence consisting of only three words, it is difficult for the neural network model to make accurate judgments with such a lack of contextual information. Forty percent of Joint's errors are caused by short sentences, and PCPR concludes that the performance of the model degrades dramatically when processing sentences shorter than six words. Based on the number of pun phrases as a function of length in Figure 1, we argue that the lack of sample size of pun phrases is a significant limitation for approaches such as neural networks.

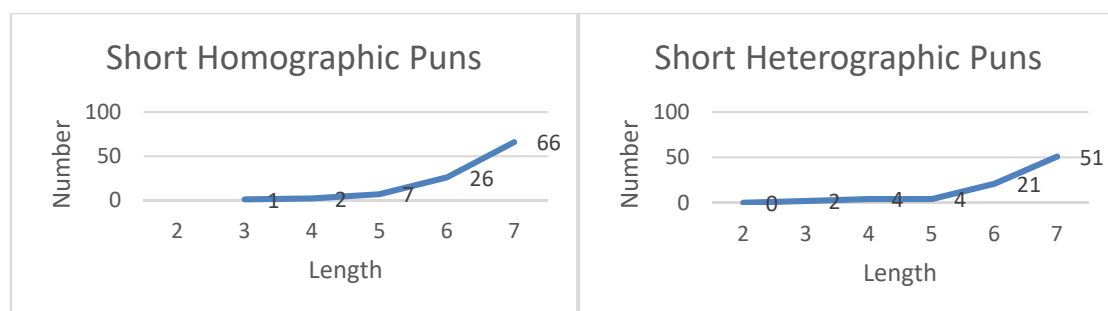


Figure 1: Statistical short puns

3. The identification of specific types of puns. To construct puns, people sometimes create words or splice words together, and Joint points out that 40% of the errors are due to unknown words in the pun sentence. For the example "The best angle from which to solve a problem is the try angle.", the similarity between the pronunciation of the words "try angle" and "triangle" is cleverly exploited to achieve the pun effect. These problems pose a major challenge for neural networks that rely on word embeddings.

6. Conclusion

This paper analyzes many classical pun recognition models and summarizes the major challenges in the field of pun recognition. Neural network models are the mainstream in the field of pun recognition. They are particularly good at processing pun sentences that contain rich context because they rely heavily on the information contained within contextual embeddings and word embeddings. All of the above models

have been studied for English puns. It is clear that puns are prevalent in different human languages, and even some puns are created cross-linguistically. The problem of identifying puns in downstream languages is a future major challenge.

References

- [1] Debra, A. (2017) Puns and tacit linguistic knowledge. *The Routledge Handbook of Language and Humor*. Routledge, pp. 80-94.
- [2] Tristan, M., Christian, F.H., Iryna, G. (2017) SemEval-2017 Task 7: Detection and Interpretation of English Puns. In: *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*.
- [3] Christian, F.H. (2008) Computational humor: Beyond the pun? *The Primer of Humor Research*. Mouton de Gruyter, pp. 333–360.
- [4] Christopher, D.M., Prabhakar R., Hinrich, S. (2008) *Introduction to Information Retrieval*. Cambridge University Press.
- [5] Martha, S.P., Hwee, T.N., Hoa T.D. (2007) Evaluation of wsd systems. *Word Sense Disambiguation: Algorithms and Applications*. Springer.
- [6] Samuel, D., Aniruddha, G., Hanyang, C., Tony, V. (2017) Idiom Savant at SemEval-2017 Task 7: Detection and interpretation of English puns. In: *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*.
- [7] Kurt, F., Geert B. (2002) Humor through double grounding: Structural interaction of optimality principles. *Odense Working Papers in Language and Communication*. pp. 312-336.
- [8] Christiane, F. (1998) *WordNet: An Electronic Lexical Database*. Massachusetts Institute of Technology.
- [9] Carnegie Mellon University. *The Carnegie Mellon University Pronouncing Dictionary*. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>
- [10] Özge, S., Nima, Ghotbi., Selma, T. (2017) N-Hance at SemEval-2017 Task 7: A computational approach using word association for puns. In: *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*.
- [11] Kenneth, W.C., Patrick, H. (1990) Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29
- [12] Olga V. (2017) UWaterloo at SemEval-2017 Task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In: *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pp. 420-424.
- [13] Dipankar, D., Aniket, Pramanick. (2017) JU_CSE_NLP at SemEval-2017 Task 7: Employing rules to detect and interpret English puns. In: *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*.
- [14] Jingyuan, F., Özge, S., Steffen, R., Eugen, R., Chris, B. (2020) Supervised Pun Detection and Location with Feature Engineering and Logistic Regression. *SwissText/KONVENS*.
- [15] Tomas, M., Quoc, Le. (2014) Distributed representations of sentences and documents. In: *International conference on machine learning*. Beijing. pp. 1188-1196.
- [16] Jacob, D., Ming-Wei, C., Kenton, Lee., Kristina, T. (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis. pp. 4171-4186.
- [17] Vijayasaradhi, I., Oota, S.R. (2017) Fermi at SemEval-2017 Task 7: Detection and interpretation of homographic puns in English language. In: *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pp. 456-459.
- [18] Yufeng, D., Hongfei, L., Di, W., Liang, Y., Kan, X., Zhihao, Y., Jian, W., Shaowu, Z., Bo, X., Dongyu, Z. (2018) WECA: A WordNet-Encoded Collocation-Attention Network for Homographic Pun Recognition. In: *the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels. pp. 2507-2516.

- [19] Dzmitry, B., Kyunghyun, C., Yoshua, B. (2015) Neural machine translation by jointly learning to align and translate. In: Proceedings of ICLR.
- [20] Yufeng, D., Hongfei, L., Liang, Y., Xiaochao, F., Di, W., Kan, X. (2019) CRGA: Homographic pun detection with a contextualized representation. Journal Pre-proof.
- [21] Yufeng, D., Hongfei, L., Liang, Y., Xiaochao, F., Di, W., Dongyu, Z., Kan, X. (2019) Heterographic pun recognition via pronunciation and spelling understanding gated attention network. In: The World Wide Web Conference. San Francisco. pp. 363-371.
- [22] Yichao, Z., Jyun-yu, J., Jieyu, Z., Kai-Wei, C., Wei, W. (2020) "The Boating Store Had Its Best Sail Ever": Pronunciation-attentive contextualized pun recognition. In: ACL2020.
- [23] Yanyan, Z., Wei, Lu. (2019) Joint detection and location of english puns. In: NAACL 2019. pp. 2117-2123.
- [24] Liyuan, L., Jingbo, S., Xiang, R., Frank, F.X., Huan, G., Jian, P., Jiawei, H. (2018) Empower sequence labeling with task-aware neural language model. In: AAAI.
- [25] Junyu, M., Rongbo, W., Xiaoxi, H., Zhiqun, C. (2020) Compositional semantics network with multi-task learning for pun location. IEEE.
- [26] Qiuchi, L., Benyou, W., Massimo, M. (2019) CNM: An interpretable complex-valued network for matching. arXiv.