IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 14, 2020, accepted February 29, 2020, date of publication March 4, 2020, date of current version March 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978208

Compositional Semantics Network With Multi-Task Learning for Pun Location

JUNYU MAO, RONGBO WANG, XIAOXI HUANG, AND ZHIQUN CHEN
School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Rongbo Wang (wangrongbo@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China Youth Fund under Grant 61202281, and in part by the Ministry of Human Resources and Social Security under Grant 12YJCZH201.

ABSTRACT A pun is always humorous and has strong interactive value in people's daily communication. It creates a humorous effect in a certain context, in which a word implies two or more meanings by using polysemy (homographic pun) or phonological similarity to another word (heterographic pun). Pun location is a task to identify the pun word in a given text, which is of great significance to understand humorous texts. Existing methods generally adopt single long sequence structure but cannot well capture the rich semantics of pun words in sentences. We present an approach that considers long-distance and short-distance semantic relations between words simultaneously. For the long-distance semantic relation, we introduce multi-level embeddings to represent the most relevant aspects of the data. For the short-distance semantic relation, we exploit the complex-valued model with a self-adaptive selection mechanism based on multi-scale of input information. Meanwhile, we propose a new classification task to distinguish the homographic pun and heterographic pun. We introduce it as an auxiliary to jointly train the original pun location task, which first learns the location of different types of puns together. Experiment results show that the latest state-of-the-art results can be achieved through our model.

INDEX TERMS Pun location, quantum theory, multi-task learning, attention mechanism, deep learning.

I. INTRODUCTION

There is a kind of language structure known as the pun in natural language texts, which is also a common rhetorical method that the author intends to make a certain word simultaneously having two or more different meanings. Puns are always humorous and have strong interactive value in people's daily communication. For example, a pun is often used as a means of humor in an advertisement to give listeners an enjoyable experience [1]. Therefore, the study of puns is a significant research subject with a wide range of practical applications.

Redfern [2] categorizes the pun into two groups, namely homographic puns and heterographic puns, respectively utilizing different senses of the same written word and different senses of the similar written or pronounced word. Our work focuses on these two types of puns. Puns that have two distinct meanings but share the same pronunciation and spelling are homographic puns. For instance: *"I'd like to tell you a chemistry joke but I'm afraid of your* ***reaction****."*

For the two meanings of the word *reaction*, one is a more conventional meaning denoting response, while the other is semantically related to *chemistry* denoting the chemical process. Homographic puns can also be created through the phrase, for instance: *"The fire chief was always asked* ***burning*** *questions."* The pun word *burning* means urgent need to be solved via compounding *burning questions* and also means on fire corresponding to *fire*. Puns that generate two distinct meanings by exploiting a similar pronunciation or nearly spelling with the latent target word are heterographic puns. For example: *"When my camera fell in the toffee I was making, I got a very* ***candied*** *picture."* In addition to the meaning sweet corresponding to *toffee*, *candied* also implies ''candid'' since the similar spelling. *Candied* is the surface sign, and ''candid'' is the latent target.

It can be seen that the pun word plays an important role in a pun. To understand puns better, it is necessary and meaningful to identify the pun word in a given text, which is regarded as the pun location, defined in SemEval 2017 Task 7 [3]. In the research of homographic and heterographic puns, an extremely clear pattern was found by Ted Pedersen [4] that a pun word will appear at the end of a sentence, with a

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

sense having semantic relation to an earlier word, and another that is in accordance with the neighbouring context. However, most previous methods only focus on long-distance semantic relations but neglect the fact that local features also play an important role, for instance, sometimes the pun creates lexical ambiguity through word combinations. To address the problem, we propose a network structure that can capture long and short distance semantic relations between words simultaneously. Pun location is still a challenging task because each sentence only has one pun word in a limited annotated data, which makes the task hard to be generalized. In order to improve the generalization of the model effectively, we introduce a multi-task learning approach. Existing systems for pun location are usually based on a single type of pun. In this work, we present a sentence level classification task to distinguish homographic pun and heterographic pun for the first time. We add the category task as an auxiliary to regularize the model training and learn the location of different types of puns together. The contributions of this paper are listed as follows:

(i) We propose a compositional semantics network to capture the long and short range relations between words simultaneously. We take multi-level embeddings as input to learn the long-range relation and apply the complex-valued model with a self-adaptive selection mechanism to learn the short-range relation.

(ii) Based on the compositional semantics network, we introduce a multi-task learning approach. We present a new pun classification task and exploit it as an auxiliary to jointly train the location of different types of puns together.

(iii) Our proposed model leads to state-of-the-art performance on both the homographic dataset and heterographic dataset.

II. RELATED WORK

Pun recognition is a common task to identify if a sentence contains a pun. In this domain, there has been a lot of relevant researches. For example, Pedersen [4] proposed a Duluth system relying on word sense disambiguation with different configurations and measures of semantic relatedness. Indurthi and Oota [5] used a bi-directional LSTM network to detect homographic puns. Diao *et al.* [6] proposed a WECA network model, which takes the WordNet-Encoded embedding as input and combines with the context weights for recognizing homographic puns.

Pun location is a more challenging task, which aims to find a pun word in a given sentence. Sevgili *et al.* [7] proposed an N-Hance system supporting the recognition of a distinctive word which has a high association with the pun in the given sentence. It calculates the PMI between every pair of words in the context to detect and locate puns. Vechtomova [8] described a method locating a pun word by using corpus-based characteristics of a word. Indurthi and Oota [5] used a Bi-directional RNN to learn a classification model. Cai *et al.* [9] proposed a sense-award neural model which is based on different WSD results. Zou and Lu [10] proposed a framework jointing detection and location, which adopts an LSTM-CRF with character embedding to make labeling decisions. These methods mostly are modeled in long sequence structures. Pun interpretation is considered as a subsequent step for pun location, and it aims to annotate the two meanings of the given pun by reference to WordNet sense keys. Miller and Gurevych [11] used a Lesk algorithm [12] to calculate the scores of candidate sense and identify the double meanings.

Pun classification is related to our work, but there is little attention to it. Some other researchers are focusing on humour classification. Ahuja *et al.* [13] presented a theoretical framework for the classification of jokes into categories and sub-categories. The Dalian University of Technology proposed a Chinese humour type recognition task to distinguish homophone, heterography and reversal in CCL2018.

Multi-task learning is also related to our work. Multi-task learning has been applied successfully to many domains, such as natural language processing [14], speech recognition [15] and computer vision [16]. In some cases, our focus is only on the performance of one or more of the multi-task, and then we can do this by setting up auxiliary tasks with various attributes. For instance, Zhang *et al.* [17] took facial attribute inference and head pose estimation as auxiliary tasks to detect facial landmarks. In this paper, we use categorization as an auxiliary task to learn the primary pun location task.

III. METHOD

In this section, we present the proposed compositional semantics network with multi-task learning (CSN-ML). This model simultaneously considers the long distance and short distance relations between words, and uses a pun classification task as an auxiliary to joint train the location of different types of puns. The overall architecture of our model is shown in Figure1.

A. COMPOSITIONAL SEMANTICS NETWORK

The compositional semantics network consists of long distance semantic relation module and short distance semantic relation module. We fuse the information learned from the two above modules and get the integrated representation for each word.

1) LONG DISTANCE SEMANTIC RELATION

In order to capture the semantic relation between the pun word and the earlier word, we introduce embeddings of different levels according to the characteristics of both homographic and heterographic puns to represent the most relevant aspects of the data. Then the concatenation of different representations for input is modeled by an LSTM network.

Character Embedding Layer: Character layer is constructed to learn the words' spelling features which also can be supplementary for unknown words. Following Ma and Hovy [18], we use Convolutional Neural Networks (CNN) to extract character-level embedding of each word. The encoding of character is specified in an alphabet, and we randomly initialize a lookup table with values from a

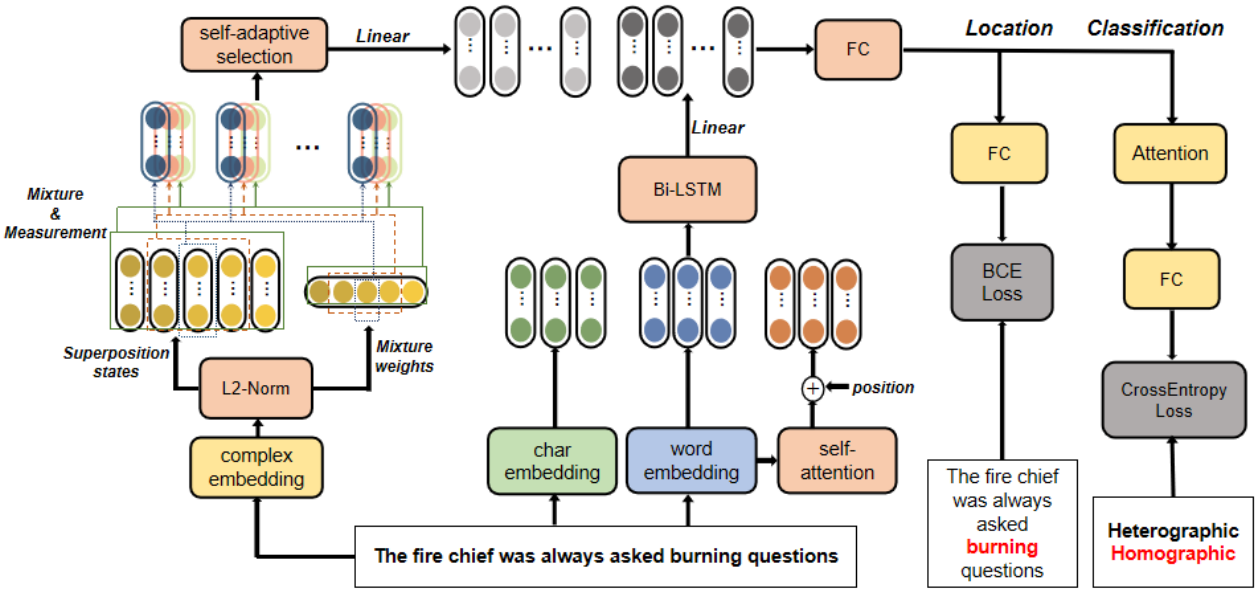

FIGURE 1. The architecture of Compositional Semantics Network with Multi Task Learning (CSN-ML). The input is first sent to the long distance and short distance semantic modules to learn different range features of words. And then the integrated representations are fed into two branches for jointly training.

uniform distribution, which can be fine-tuned during training. The character-level embeddings are computed by the CNN with character encodings as inputs. Then we get the character-level vector of each word $c \in \mathbb{R}^{d_c}$

Word Embedding Layer: Word layer is responsible for learning the static semantics of each word. We use pre-trained word vectors initialized by Glove [19] to obtain the fixed embeddings of words $w \in \mathbb{R}^{d_w}$.

Interacting Embedding Layer: Interacting layer is used to capture the correlation between words in a sentence, which is important in both homographic and heterographic puns. For instance:*"The orange squeezer was invented with some juicy information."* In this sentence, the pun word *juicy* has strong semantic relevance to *orange*, *squeezer* and *information*, respectively. The dot product can be viewed as a measure of the correlation between two vectors. Therefore, for extracting relevant information natively from word pairs, we introduce the self-attention mechanism combining with position encodings [20] on pre-trained word embeddings to get the interacting embeddings of words $s \in \mathbb{R}^{d_w}$.

Long Sequence Modeling Layer: After learning the different features of the input, we feed them into the long sequence modeling layer. The input $M \in \mathbb{R}^{(d_w+d_w+d_c)\times n}$ is the concatenation of character-layer, word-layer and interacting-layer embeddings. The Long Short Term Memory Network [21] is designed to solve long-term dependency problems, which is suitable for our task. We use a bi-directional LSTM network as the top layer to learn the temporal interactions and long range dependencies between words. The hidden state of forward LSTM and backward LSTM are concatenated at each time step, and then we get the concatenated hidden vectors $h_i, i = 1, \ldots, n$, for all words as the outputs of this module.

2) SHORT DISTANCE SEMANTIC RELATION

The uncertainty of Language is first reflected at the word level in an ambiguous scene, and second, there are also different word combinations at the semantic compounding level. Hence, we proposed a complex-valued model with scales selection mechanism to learn the short distance semantic relation.

N-Gram Feature Extraction Layer: We use a complex-valued network [22] to extract the n-gram features. The inputs are complex-valued embeddings [23] consisting of a real part and an imaginary part, which can be converted into amplitude and phase. The amplitude corresponds to the value of the traditional real-value vector (lexical meaning), while the phase may represent some higher-level semantics such as polarity, ambiguity or emotion. Following [22], each word w is normalized into a superposition state $|w\rangle$, where $||\overrightarrow{w}||$ denotes the 2-norm length of $\overrightarrow{w}$, namely $\pi(w)$, which is used to compute the relative weight of a word in a local context window:

$$|w\rangle = \frac{\overrightarrow{w}}{||\overrightarrow{w}||}, \pi\,(w) = ||\overrightarrow{w}|| \tag{1}$$

In order to capture the n-gram feature, we apply a weighted sliding window and construct a density matrix for a local window of length l. Hence, a sentence consists of a sequence of l-grams density matrices. The l-gram density matrix is calculated as follows:

$$\rho = \sum_{i}^{l} p(w_i)\,|w_i\rangle\,\langle w_i| \tag{2}$$

$p\,(w_i)$ is the softmax normalized word relative weight: $p\,(w_i) = \frac{e^{\pi(w_i)}}{\sum_j^l e^{\pi(w_j)}}$, where $\pi\,(w_i)$ is the word-dependent

weight described above. Then, the semantic measurements operators $\{|v_k\rangle\}_{k=1}^{K}$ are applied to the $\{\rho_i\}_{i=1}^{n}$, where n is the length of sentence. The K-by-n probability matrix P is: $P_{ik} = \langle v_k | \rho_i | v_k \rangle$, for $k \in (1, \ldots, K)$, $i \in (1, \ldots, n)$, and the $\{|v_k\rangle\}_{k=1}^{K}$ is trainable.

Self-Adaptive Selection Layer: We observe that that the pun word generates ambiguity also depends on different lengths of word combinations. To select the suitable size of the receptive field based on multiple scales of the local context, we introduce the self-adaptive selection mechanism. Our work focuses on the feature representations on word-level and the sliding windows are word-centered. For different sizes of local context window $j \in \{1, 3, \ldots, l\}$ yield different scales of probability matrix P_j. Inspired by Li *et al.* [24], we adopt a dynamic selection mechanism that allows each neuron to adaptively adjust its size of receptive field based on different scales for input information.

$$o = F_{fc}(\overline{P}) = \delta(\beta(W\overline{P})), o \in \mathbb{R}^{n \times d} \tag{3}$$

o, computed by Eq(3), is a compact feature which is created for leading to a more precise and adaptive selection. $\overline{P}$ is the mean of $P_1, \ldots, P_l$. δ is the ReLU function, β denotes the Batch Normalization, and $W \in \mathbb{R}^{K \times d}$ is the parameter for dimensionality reduction to improve the efficiency better.

$$a_{jc} = \frac{exp(ox_{jc})}{\sum_{j=1}^{l} exp(ox_{jc})} \tag{4}$$

Then, we apply softmax function on the channel-wise digits to make an adaptive selection on different spatial scales of information, where $x_j \in \mathbb{R}^{d \times K}$ and a_j denotes the soft attention vector for P_j. The final output P* is gained by the attention weights on different sizes of local context windows:

$$P_c^* = \sum_{j=1}^{l} a_{jc} P_{jc} \tag{5}$$

where $P^* = \{P_1^*, \ldots, P_K^*\}$, $P^* \in \mathbb{R}^{n \times K}$ and $c \in K$ denotes the channel dimension.

Fusion Layer: After learning the long distance and short distance semantic information, the next step is to fuse them to obtain the integrated representation. We first send the two above outputs h_i and P_i^* to the linear transformation layers respectively to get the vectors with the same dimension. The concatenation of them is fused through a fully-connected layer with *tanh* as activation function. The layer normalization is further applied, and then we get the integrated vector $r_i, i = 1, \ldots, n$.

B. MULTI-TASK LEARNING

After obtaining the composed semantic representation, we introduce a multi-task learning framework. Generally, the additional task plays the role of a regularizer to generalize the model [14], [25]. We present a pun classification task to distinguish homographic and heterographic puns for the first time. Used as an auxiliary branch, it not only can learn the main task location for both of them simultaneously but also can learn more general feature representations. We design a simple tagging scheme consisting of two tags {0, 1}: 0 tag means the current sentence is a homographic pun, 1 tag means the current sentence is a heterographic pun.

Multi-Task Loss Function: Our model contains two branches, one for location and one for classification. We design a multi-task loss L on a mini-batch of training samples to jointly train.

$$L = L_{loc} + \lambda L_{cla} \tag{6}$$

where L_{loc} is for pun location and L_{cla} is for pun classification. λ is a hyper-parameter, which is used to balance the losses. When the model converges, we calculate the ratio of the two losses described above to get the value of λ.

The Location Loss: For the pun location, the contexts in the corpus possess the property that each pun (and its latent target) contains exactly one content word (i.e., a noun, verb, adjective, or adverb). Therefore, similar to the work of (Cai *et al.*, 2018), we only make a prediction of a word when it belongs to the four types of parts of speech. The integrated vector r_i that has one of the four POS tags is sent to a linear transformation layer and we get a real number output g_i. Since there is only one pun word in each sentence in the experimental data set, we make a prediction using the sigmoid function on g_i. The k-th word will be taken as pun word if and only if g_i is the largest number out of all g_i, i $= 1, \ldots, n$, and $(g_k) > 0.5$. Viewing pun word as a word-level classification task, we use the Binary Cross Entropy loss to calculate L_{loc} as follows:

$$L_{loc} = -\sum_{s}\sum_{k}(1 - y_s^k)\log(1 - \hat{y}_s^k) \tag{7}$$

where s represents the index of the sentence, k is the word belongs to the four kinds of POS tags.

The Classification Loss: For the pun classification, we introduce an attention mechanism. Although homographic pun and heterographic pun have a similar sentence structure, their ways of generating ambiguity and the importance of each word in a sentence are different. Specifically,

$$u_i = tanh(W_w r_i + b_w) \tag{8}$$

$$a_i = \frac{exp(u_i^{\mathrm{T}} u_w)}{\sum_{i=1}^{n} exp(u_i^{\mathrm{T}} u_w)} \tag{9}$$

$$v_i = \sum_{i=1}^{n} a_i r_i \tag{10}$$

The u_i is the hidden representation of r_i through a full-connected layer with *tanh* as activation function. We calculate the similarity between u_i and the context vector u_w to get a normalized importance weight a_i through a softmax function. The u_i is randomly initialized and can be jointly learned during training. The high-level weighted sentence representation v_i is computed by Eq(10). Then we send the sentence vector v_i to a linear transformation layer and get the final predicted

distribution through a softmax function. The classification loss L_{cla} is calculated by cross-entropy as follows:

$$L_{cla} = -\sum_{s}\sum_{j} y_s^j \log \hat{y}_s^j, \tag{11}$$

where s denotes the index of sentence and j denotes the index of category.

IV. DATASETS AND SETTINGS

We evaluate our model on two benchmark datasets from the SemEval 2017 Task 7 [3]. The homographic dataset consists of 2,250 contexts with 1,607 containing a pun, and the heterographic dataset consists of 1,780 contexts with 1,271 containing a pun. Since our work focuses on pun location, we only use sentences with pun words. To make direct comparisons with prior studies, following Cai *et al.* [9] and Zou and Lu [10], we apply 10-fold cross validation. The outputs of all 10 folds are accumulated and then the precision, recall and F1 scores of homographic and heterographic datasets are calculated respectively. For each fold, we randomly select 10% instances from the training set as a validation set. In the long distance semantic relation module, word embeddings are initialized with the 100-dimensional Glove [19], and the size of hidden vectors for LSTM is 300. We randomly initialize the 30-dimensional character encodings with a uniform $[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$, where dim $= 30$. In the short distance semantic relation module, the amplitudes are initialized with 100-dimension Glove vectors, with comparable performance, and the phases are randomly initialized with a normal distribution of $[0, 2\pi]$. The semantic measurements $\{|v_k\rangle\}_{k=1}^{K}(K = 100)$ are initialized with a uniform distribution of $(0, 1)$, and during training, each measurement is restricted in unit length. The reduced dimension d is 16, and the value of λ for balancing loss is 0.17. We adopt stochastic gradient descent (SGD) [26] as the optimization algorithm with weight decay, and the learning rate is 0.015 with a learning rate decay. Meanwhile, we also use a dropout strategy to prevent the overfitting problem and the dropout is 0.5.

V. RESULTS

In this section, we discuss the experiment results on Compositional Semantics Network with Multi-task Learning. First, we do some detailed analysis to demonstrate the effectiveness of our model. Then, we compare the performance with existing methods.

A. DETAILED ANALYSIS

We conduct additional experiments for detailed analysis of the compositional semantics network and multi-task learning method as following.

1) ANALYSIS ON COMPOSITIONAL SEMANTICS NETWORK

In this part, we analyze the effects of long distance semantic relation module and short distance semantic relation module. The ablation experiments are based on the homographic dataset. Results are shown in Table1.

TABLE 1. Effects of compositional semantics network.

Models	Features	Settings	Precision	Recall	F1
Short	Word	Concatenate	67.62	61,73	64.51
	Word	Self-adaptive	68.21	63.41	65.72
Long	Word	-	84.36	73.49	78.55
	+Character	-	84.35	75.79	79.84
	+Interacting	-	83.74	76.60	80.01
	All	-	**84.66**	77.60	80.97
CSN(default)	Word	Concatenate	82.92	79.47	81.16
CSN(ensemble)	All	Self-adaptive	83.74	**81.39**	**82.55**

To verify the effectiveness of CSN, we design a series of models. First, we implement the single model which only introduces the short-distance semantic module or the long-distance semantic module. The short-distance semantic module is based on the complex-valued network. For ablating the self-adaptive selection layer, we replace it with the general concatenation operation. The results show that the dynamic selection mechanism is more flexible for our task. The long sequence structure is based on the BiLSTM network. Word, character and interacting embeddings represent different input features respectively. Both character-level and interacting-level embeddings contribute to the model's performance. The interacting embeddings perform better. We conjecture that the relevancy between words is a significant characteristic since the pun word always has a strong correlation with other words in a sentence. When all the input features are combined, we obtain the best results on the single long sequence model, which shows the effectiveness of multi-level input features. As we can see, the single long-distance semantic model outperforms the short-distance semantic model, we conjecture that the long sequence structure can better understand the global information, while the short-distance semantic module is better at handling the immediate context as a supplement.

Then, we implement the combined models CSN (default) and CSN (ensemble). When short-distance semantic information is assembled, all the results in Recall and F1 perform better than the single model. It shows the effectiveness of fusing long-distance and short-distance semantic information. And the CSN (ensemble) achieves the best performance (82.55% of F1).

2) ANALYSIS ON MULTI-TASK LEARNING

In this part, we merge heterographic with homographic datasets and shuffle them. CSN is the compositional semantic network described before. For ablating the multi-task learning mechanism, we design a CSN-N-ML model. The structure of CSN-N-ML is the same as CSN except for the input data set (both homographic and heterographic puns). The CSN-ML is the compositional semantic network with multi-task learning. The results are shown in Table2:

We can see that the model with multi-task learning (CSN-ML) performs best on heterographic puns and yields competitive results on homographic puns, compared to the models that do not jointly train the loss. For the CSN-N-ML model, we observed that the performance of

TABLE 2. Effects of multi-task learning.

System	Homographic			Heterographic		
	Precision	Recall	F1	Precision	Recall	F1
CSN	83.74	**81.39**	82.55	88.31	85.60	86.94
CSN-N-ML	84.84	80.09	82.39	**88.85**	84.66	86.70
CSN-ML	**85.04**	81.33	**83.14**	88.84	**85.76**	**87.27**

TABLE 3. Comparison of different methods on two benchmark datasets.

System	Homographic			Heterographic		
	Precision	Recall	F1	Precision	Recall	F1
Pedersen (2017)	44.00	44.00	44.00	-	-	-
Indurthi and Oota (2017)	52.15	52.15	52.15	-	-	-
Özge Sevgili et al.(2017)	42.69	42.50	42.59	65.92	65.15	65.53
Vechtomova (2017)	65.26	65.21	65.23	79.73	79.54	79.64
Cai et al. (2018)	81.50	74.70	78.00	-	-	-
Zou et al.(2019)	83.55	77.10	80.19	81.41	77.50	79.40
CSN-ML	**85.04**	**81.33**	**83.14**	**88.84**	**85.76**	**87.27**

location drops when the data increases. One possible reason is that different types of puns may slightly interfere with each other's location. And when the additional task is introduced, the performance of model is improved. It shows the effectiveness of multi-task learning.

3) ERROR ANALYSIS

We also studied the error outputs from our model and make some analysis. We found that it was challenging to predict labels in short texts in our model. For instance, *"A summer is a mathematician."* The pun word is *summer*, however, it could be challenging to identify the ambiguous word when the context information is limited. We also found some errors are due to the lack of background knowledge. For example, *"Humpty Dumpty had a great fall - and a pretty good spring and summer, too."* Here, *fall* is the pun word and *"Humpty Dumpty had a great fall"* is a nursery rhyme. To make correct predictions, background knowledge would be required in some cases.

B. COMPARISON WITH EXISTING METHODS

We compare our model with the prior methods and the results are shown in Table3. It is shown that the systems based on deep learning by (e.g., Cai *et al.* [9], Zou and Lu [10]) are generally superior to the rule-based systems by (i.e.,Perderen [4], Indurthi and Oota [5], Özge Sevgili *et al.* [7], Vechtomova [8]). Cai *et al.* [9] leveraged multiple WSD results and BiLSTMs to model sequences of word senses. Zou and Lu [10] proposed a framework based on Bi-directional LSTM-CNNs-CRF [18] for joint detection and location of puns. Compared with other methods, we adopt BiLSTM with multi-level embeddings which can learn the common features of homographic and heterographic puns to model the long sequence structure. And we also introduce short distance semantic information via the complex-valued network. Besides, we introduce a new classification task as an auxiliary to jointly train the model. Among all the methods, our model (CSN-ML) yields state-of-the-art Precision, Recall and F1 scores on both homographic and heterograhic datasets for pun location. It demonstrates the effectiveness and superiority of our model.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a Compositional Semantics Network with Multi-task Learning for Pun Location. We fuse the long-distance dependencies and local correlations to obtain the rich semantic information of the pun word in a sentence. In terms of the long-distance dependencies, we introduce multi-level input features. In terms of the local correlations, we propose a complex-valued model with a self-adaptive selection mechanism. Furthermore, we come up with a new classification task to distinguish homographic and heterographic pun. We exploit it as an auxiliary to jointly train the main task, which can learn the location of different types of puns simultaneously. The experimental results on two benchmark datasets show that our method achieves significant improvement over existing methods and produces a new state-of-the-art performance.

In future work, we would like to continue to study the interpretation and generation of puns. The research on puns for the Chinese language will also be an interesting direction for us in the future. All these are promising studies we can conduct in our future research.

REFERENCES

[1] M. van Mulken, R. V. Enschot, and H. Hoeken, "Levels of implicitness in magazine advertisements: An experimental study into the relationship between complexity and appreciation in magazine advertisements," *Inf. Des. J.*, vol. 13, no. 2, pp. 155–164, Jan. 2005.

[2] W. Redfern, "Puns," *Scriblerian Kit-Cats*, vol. 19, no. 2, p. 204, 1987.

[3] T. Miller, C. Hempelmann, and I. Gurevych, "SemEval-2017 task 7: Detection and interpretation of english puns," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 58–68.

[4] T. Pedersen, "Duluth at semeval-2017 task 7 : Puns upon a midnight dreary, lexical semantics for the weak and weary," 2017, *arXiv:1704.08388*. [Online]. Available: http://arxiv.org/abs/1704.08388

[5] V. Indurthi and S. R. Oota, "Fermi at SemEval-2017 task 7: Detection and interpretation of homographic puns in English language," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 457–460.

[6] Y. Diao, H. Lin, D. Wu, L. Yang, K. Xu, Z. Yang, J. Wang, S. Zhang, B. Xu, and D. Zhang, "WECA: A WordNet-encoded collocation-attention network for homographic pun recognition," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2507–2516.

[7] Ö. Sevgili, N. Ghotbi, and S. Tekir, "N-hance at SemEval-2017 task 7: A computational approach using word association for puns," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 436–439.

[8] O. Vechtomova, "UWaterloo at SemEval-2017 task 7: Locating the pun using syntactic characteristics and corpus-based metrics," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 421–425.

[9] Y. Cai, Y. Li, and X. Wan, "Sense-aware neural models for pun location in texts," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 546–551.

[10] Y. Zou and W. Lu, "Joint detection and location of english puns," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2117–2123.

[11] T. Miller and I. Gurevych, "Automatic disambiguation of english puns," in *Proc. 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 719–729.

[12] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine code from an ice cream cone," in *Proc. 5th Annu. Int. Conf. Syst. Document.*, 1986, pp. 24–26.

[13] V. Ahuja, T. Bali, and N. Singh, "What makes us laugh? Investigations into automatic humor classification," in *Proc. 2nd Workshop Comput. Model. People's Opinions, Personality, Emotions Social Media*, 2018, pp. 1–9.

[14] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 160–167.
[15] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8599–8603.
[16] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[17] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 94–108.
[18] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," 2016, *arXiv:1603.01354*. [Online]. Available: http://arxiv.org/abs/1603.01354
[19] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[22] Q. Li, B. Wang, and M. Melucci, "CNM: An interpretable complex-valued network for matching," 2019, *arXiv:1904.05298*. [Online]. Available: http://arxiv.org/abs/1904.05298
[23] Q. Li, S. Uprety, B. Wang, and D. Song, "Quantum-inspired complex word embedding," 2018, *arXiv:1805.11351*. [Online]. Available: http://arxiv.org/abs/1805.11351
[24] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.
[25] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*. [Online]. Available: http://arxiv.org/abs/1701.00160
[26] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks Trade*. Berlin, Germany: Springer, 2012, pp. 421–436.

JUNYU MAO was born in Jiangxi, China, in 1996. She received the bachelor's degree in computer science and technology from Nanchang University, in 2016. She is currently pursuing the master's degree with the Computer Technology Department, Hangzhou Dianzi University.

RONGBO WANG was born in Yiwu, Zhejiang, China, in 1978. He received the B.S. degree in computer and application from the Zhejiang University of Technology (ZJUT), Hangzhou, China, in 1999, the M.S. degree in computer science and technology from Zhejiang University (ZJU), Hangzhou, in 2002, and the Ph.D. degree in electronic and information engineering from Hong Kong Polytechnic University, Hong Kong, China, in 2005. Since 2005, he has been an Associate Professor with the College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou. His research interests include natural language processing, question and answering systems, and machine learning.

XIAOXI HUANG was born in Wenzhou, Zhejiang, China, in 1979. He received the B.S. degree in computer science and technology and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2001 and 2009, respectively. From 2010 to 2011, he was a Postdoctoral Researcher with the Center of Study of Language and Cognition, Zhejiang University. Since 2012, he has been an Associate Professor with the College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou. His research interests include natural language processing, metaphor computation, and machine learning.

ZHIQUN CHEN was born in Nanchang, Jiangxi, China, in 1973. He received the B.S. degree in computer software and theory from Jiangxi Normal University, Jiangxi, in 1996, and the M.S. degree in computer application from Zhejiang University, Zhejiang, China, in 1999. Since 2006, he has been a Vice Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. His research interests include natural language processing, text mining, and artificial intelligence.