International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 1, pp: 1296-1300 www.ijaem.net ISSN: 2395-5252

Detection and Location of English Puns using Neural Model

Minimol M, Naseer C
Computer Science and Engineering GEC Palakkad Palakkad, India
Computer Science and Engineering GEC Palakkad Palakkad, India

----
Date of Submission: 01-07-2020 Date of Acceptance: 16-07-2020

ABSTRACT: Natural languages like english has a wide range of vocabulary. While considering such a vocabulary of englsih, pun is a joke which exploits the different possible meanings of a word by exploiting the polysemy or phonological similarity. So in the area of Natural Language Processing (NLP), pun is a wordplay in which a word carries multiple (two or more) meanings. There are different types of puns. This work deals with two kinds of puns:
1) Homographic pun and 2) Heterographic pun. In homographic puns, it exploits polysemy where the two meanings of the word share the same pronunciation. In heterographic puns, it utilise the phonological similarity of a word with another word. Here the proposed method deals with two tasks related to the pun: 1) Pun detection and 2) Pun location. Pun detection is the process of finding out whether the given text (sentence) contains a pun or not. And pun location is a task which will be done only after the pun detection task, whose aim is to find out the exact pun word that makes the entire sentence as a pun. This work approaches the pun detection and location tasks as a sequence labelling task with a tagging scheme. The pun detection and location tasks is expressed as a sequence labeling problem, which allows us to detect and locate a pun in a sentence by assigning each word a tag. To capture such a structural property of puns, it used word position knowledge in the structured prediction model. A BiRNN + CRF architecture is able to do the pun detection and location tasks in a sequence labelling mannner. Experimental results shown that the method of sequence labelling is really effective in pun detection and location tasks.
***Index Terms*:** Pun, BiRNN, CRF, sequence labelling

I. INTRODUCTION

Pun is not a new term in the field of natural language processing (NLP), but it got relatively less attention in this field. Most of the people are using the puns in their daily life in social media platforms, comedy shows etc. But the people are unaware about the wide range of puns. While considering the large vocabulary of english, there are words which 1) have same pronunciation but different spelling and meaning (homophone), 2) have same spelling, different pronunciation and meaning (homograph), 3) have same spelling and pronun- ciation but have different meanings (mixture of homophone and homograph). The pun is also called paronomasia, is used for intended humorous or rhetorical effects. Term that always

Identify applicable funding agency here. If none, delete this. mixed with puns in a wrong manner is malapropism. Malapropism is considered as an incorrect variation on a cor- rect expression. But a pun involves expressions with multiple correct interpretations. Ultimately puns may be regarded as jokes or idiomatic constructions. Puns is not emerged in the recent years but was used by the famous writers and comedians in the ancient times itself. Puns were used in ancient Egypt for the development of myths and interpretation of dreams. The famous Roman play writer Plautus was famous for puns as he used puns and word games in his works. In literature also, the famous writers have used the concept puns in their works. William Shakespeare has used a wide range of puns in his popular works. Use of such puns revealed the intelligence and cleverness of a character in the work and also the creative thinking of the writer. Consider the below given examples.

1) A happy life depends on a liver.
2) Atheism is a non-prophet institution.

In the first example, the word *liver* can refer to either a body organ, or simply a person who lives. So it is treated as a homographic pun. Similarly, while considering the second example, the word *prophet* is used instead of the word *profit* which in turn produces a humorous effect. So it is treated as a heterographic pun.

The proposed system detects the puns in a given text. And it has to detect whether the given text is a pun or not. If yes, then it has to find out the exact word that makes the entire sentence as a pun. Here the proposed model is based on a Bidirectional Recurrent Neural Network (BiRNN) for pun detection and location. And the model is trained by a dataset provided by the previous researchers in the same area of pun detection and location.

A. Applications

There are so many applications of pun detection and pun location. Some of the important fields or systems that utilize pun words are given below.

1) Comedy and jokes: Puns are one of the common sources of humour in jokes and comedy shows. They are often used in the punch line of a joke, where they typically give a humorous meaning to it.
2) Literature: Non-humorous puns were and are a standard poetic device in English literature. Many famous writers have used puns and other forms of wordplay.
3) Rhetoric: Puns can function as a rhetorical device, as it is mainly used for rhetorical effect also.
4) Design: Used for making interesting titles and the names of places, characters, and organizations, and in advertis- ing and slogans.
5) In the media: It can be argued that paronomasia is common in the media, especially headlines, to draw the reader's interest.

The section 2 includes various methods for both homo- graphic and heterographic pun detection and location. Section 3 includes the basic architecture and concepts used. Section 4 discuss about the neural model working.

II. METHODS FOR PUN DETECTION AND LOCATION

This section deals with various methods for pun detection and location.

A. Methodology-1 Samuel Doogan et.al 2017

It introduced a system consists of probabilistic models for each type of puns using Google n-grams and Word2Vec. The polysemous words have identical spelling but different meanings. So detecting homographic puns is solely dependent on context information. They introduced Idiom Savant, a computational system that capable of classifying and analyzing heterographic and homographic puns. For heterographic pun detection they have used a CMU dictionary. And they have shown that using n-grams in combination with the CMU dic- tionary can accurately model heterographic pun. However this model has a number of drawbacks. In the case of homographic pun detection, lack of adequate contextual information is a drawback [1].

B. Methodology-2 Vijayasaradhi Indurthi et.al 2017

This methodology described the first competitive evaluation for the automatic detection, location, and interpretation of english puns. It participated in two subtasks pun detection and interpretation related to homographic puns. It used a BiRNN model for the pun detection task and another algorithm is used to find out the exact pun word in the text. It achieved compar- atively better results than the previous works for these tasks. It used the pre-trained 50 dimensional GloVe embeddings which were trained on about 6B words from the twitter using the Continuous Bag of Words architecture[2].

C. Methodology-3 Yitao Cai et.al 2018

The methodology focused on the task of pun location, whose aim was to identify the pun word in a given short text. They proposed a sense aware neural model to solve the pun location task. The model first obtained several WSD results for the given text, and then it leveraged a bidirectional LSTM network to model each sequence of word senses. The outputs at each time step for different LSTM networks were then concatenated for prediction. And the final prediction showed the exact pun word in the given text. They have used both baseline model and sense aware neural model for the pun location prediction. Their proposed sense-aware neural model is different from the baseline neural model. Because in sense- aware neural model, it models multiple sequences of word senses corresponding to different WSD results. So the sense- aware model works on the sense level, but the baseline model works on the word level[3].

D. Methodology-4 Yanyan Zou a et.al 2019

This work introduced joint detection and location of en- glish puns using a single model as a joint sequnce labelling task[4]. They have proposed two new tagging schemes for the joint detection and location of puns. This proposed model is much more effective in handling both homographic and heterographic puns. They demonstrated that the detection and location of puns can be jointly addressed by a single model. They implemeted method for pun detection and location tasks as a single sequence labeling problem. It allowed them to jointly detect and locate a pun in a sentence by assigning each word a tag. They designed two tagging schemes (NP and BPA) for this purpose. The proposed architecture consists of Bidirectional LSTM network with conditional random fields (CRF) as shown in figure 1.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 1, pp: 1296-1300 www.ijaem.net ISSN: 2395-5252

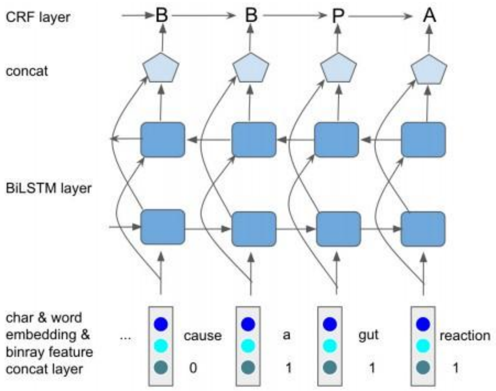

Fig. 1. Joint detection and location of puns architecture

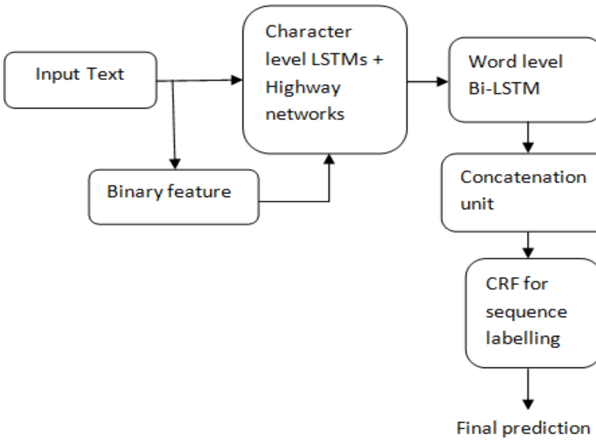

Fig. 2. Basic architecture of the pun detection and location system

III. ENGLISH PUN DETECTION AND LOCATION USING NEURAL MODEL

The important characteristics or conclusions about the puns drawn from the previous researches are listed below:

1) There will be a maximum of one pun in each con- text[4][5].

2) Each pun contains exactly one content word, which can be a noun, a verb, an adjective, or an adverb[4].

3) The pun word in a sentence usually appears towards the end of the sentence[4][6]. That is pun word tends to appear in the second half of the sentence.

4) Stopwords and non content words are not puns[2].

The neural model utilise these observations for the pun detection and location tasks. Figure 2 shows the basic archi- tecture of the proposed pun detection and location system. The proposed system detects the puns in a given text. And it has to detect whether the given text is a pun or not. If yes, then it has to find out the exact word that makes the entire sentence as a pun. Here the proposed model is based on a Bidirectional Recurrent Neural Network (BiRNN) for pun detection and location. And the model is trained by a dataset provided by the previous researchers in the same area of pun detection and location.

A. Pipeline Architecture

The overall system architecture is shown in Figure 3. The pipeline architecture takes the context as the input. And it will do the pun detection task on the context. It will determine whether the given context contains pun or not. If the context contains pun then it will do the pun location task and will produce the result. If the pun detection task results the context does not contain pun it will directly stop the further task. So the main advantage of this pipeline architecture is that it won't go for pun location task if the context does not contain pun. Hence this pipeline method yields better results.

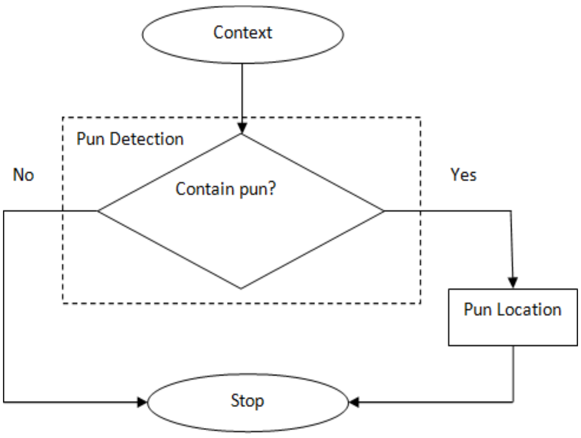

Fig. 3. Pipeline Architecture

B. Concepts used in the Neural model

The main concepts used in this method are listed below.

1) Sequence Labeling
2) Language Models
3) Multi-Task Learning
4) Character RNNs
5) Highway Networks
6) Conditional Random Fields

7) Viterbi Decoding

IV. PUN DETECTION AND LOCATION

The main goal of this neural model is that for a given context from the training set, it has to generate its correspond- ing gold tag sequence using a deterministic procedure. It has been proven that the neural model can work well on sequence labelling tasks[7][8][9]. And the Bidirectional RNN is built on top of the Conditional Random Fields (CRF)[10][11] architecture to make labeling decisions, which is one of the most popular classical models for sequence labeling.

The model works as follows. We are inputing a con- text/sentence $s = (s_1, s_2, ..., s_n)$ to the character level RNN and word level RNN and generate the char and word em- bedding. It is using sub-word information in this tagging task because it can be a powerful indicator of the tags, whether they're parts of speech or entities. Also the sub-word features, which are the outputs of the character RNNs, are also enriched with additional information. And this knowledge is needed to predict the next word in both forward and backward directions. So the sequence tagging model uses both word-level infor- mation in the form of word embeddings and character-level information up to and including each word in both directions, enriched with the knowledge required to predict the next word in both directions.

The Bidirectional RNN encodes these features into new features at each word containing information about the word and its neighborhood, at both the word-level and the character- level. This forms the input to the Conditional Random Field. The concatenation of this transformed character embeddings, the pre-trained word embeddings[12], and the position indi- cators are taken as input of the model. The binary feature is introduced and that indicates if a word is located at the first or the second half of an input sentence to capture such positional information. This binary indicator can be mapped to a vector representation using a randomly initialized embedding table, but here we are directly using the binary indicator with our input. Feed this input together with binary indicator into a BiRNN to capture contextual information.

Next we have to feed this input into a CRF layer to capture label dependencies and make final tagging decisions at each position. The methodology can be simply explained in three steps as follows.

1) The character RNN in the forward direction predicts the next word at the end of every word. And these character RNNs leverages sub-word information to predict the next word.

2) The character RNN in the backward direction predicts the next word at the end of every word. We use the outputs of these two character-RNNs as inputs to our word-RNN and Conditional Random Field (CRF) to perform our primary task of sequence labeling.

3) A Bidirectional word RNN and a conditional random field (CRF) together predict the tag of each word.

V. CONCLUSION AND FUTURE SCOPE

This work is proposed to perform pun detection and location tasks using the same architecture from a sequence labeling perspective. The method utilised the observation that each text in the corpora contains a maximum of one pun and also the interesting structural property such as the fact that most puns tend to appear at the second half of the sentences. Hence, a new tagging scheme is designed. This method is generally applicable to both heterographic and homographic puns.

Low word coverage, detection errors, lack of adequate contextual information are the main challenges in this method also. However in the future works we can include much better neural network architecture and WSD algorithms. We can explore the pun in other languages too. And this will be really interesting.

REFERENCES

[1]. Doogan, Samuel and Ghosh, Aniruddha and Chen, Hanyang and Veale, Tony. 2017. "Idiom savant at semeval-2017 task 7: Detection and interpretation of english puns", In Proceedings of the 11th international workshop on semantic evaluation.

[2]. Indurthi, Vijayasaradhi and Oota, Subba Reddy. Fermi at semeval- 2017 task 7:"Detection and interpretation of homographic puns in english language", In Proceedings of the 11th international workshop on semantic evaluation.

[3]. Cai, Yitao and Li, Yin and Wan, Xiaojun. 2018."Sense-aware neural models for pun location in texts", In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

[4]. Zou, Yanyan and Lu, Wei. 2019." Joint detection and location of english puns", arXiv preprint arXiv: 1909.00175.

[5]. Miller, Tristan and Hempelmann, Christian F and Gurevych, Iryna. SemEval-2017 Task 7:"Detection and interpretation of English puns", In Proceedings of the 11th International Workshop on Semantic

Evaluation.
[6]. Miller, Tristan and Turkovic´, Mladen. 2016.”Towards the automatic detection and identification of English puns”, The European Journal of Humour Research.
[7]. Chiu, Jason PC and Nichols, Eric. 2016.”Named entity recognition with bidirectional LSTM-CNNs”, Transactions of the Association for Computational Linguistics.
[8]. Ma, Xuezhe and Hovy, Eduard. 2016.”End-to-end sequence labeling via bi-directional lstm-cnns-crf”, arXiv preprint arXiv: 1603.01354.
[9]. Luo, Fuli and Liu, Tianyu and He, Zexue and Xia, Qiaolin and Sui, Zhi- fang and Chang, Baobao. 2018.”Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention”, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.
[10]. Lafferty, John and McCallum, Andrew and Pereira, Fernando CN. 2001.”Conditional random fields: Probabilistic models for segmenting and la- beling sequence data”, In Proceedings of Association for Computational Linguistics.
[11]. Graves, Alex and Schmidhuber, Ju¨rgen. 2005.”Framewise phoneme classification with bidirectional LSTM and other neural network archi- tectures”, Elsevier.
[12]. Pennington, Jeffrey and Socher, Richard and Manning, Christopher D. 2014. ”Glove: Global vectors for word representation”, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

International Journal of Advances in Engineering and Management
ISSN: 2395-5252

IJAEM

Volume: 02 Issue: 01 DOI: 10.35629/5252

www.ijaem.net
Email id: ijaem.paper@gmail.com