Supervised Pun Detection and Location with Feature Engineering and Logistic Regression

Jingyuan Feng[⋆], **Özge Sevgili**[†], **Steffen Remus**[†], **Eugen Ruppert**[†], and **Chris Biemann**[†]

[⋆]Technische Universität Hamburg, Hamburg, Germany
[†]Universität Hamburg, Hamburg, Germany
jingyuan.feng@tuhh.de
{sevgili,remus,ruppert,biemann}@informatik.uni-hamburg.de

Abstract

Puns, by exploiting ambiguities, are commonly used in literature to achieve a humorous or rhetorical effect. Previous approaches mainly focus on machine learning models or rule-based methods, however, they have not addressed how and why a pun is detected or located. Focusing on this, we propose a system for recognizing and locating English puns. Regarding the fact of limited training data and the aim of measuring how relevant a predictor and its direction of the association is, we compile a dataset and explore different feature sets as input for logistic regression, and measure their influence in terms of the assigned weights. To our best knowledge, our system achieves better results than state-of-the-art systems on three subtasks for different types of puns respectively.

1 Introduction

Puns are a type of wordplay that deliberately exploits two or more different meanings of the same or similar words in a sentence. Puns utilizing the same word with ambiguous senses is known as homographic. *I used to be a banker but I lost **interest***; in this sentence, the trigger word "interest" could mean "curiosity" and "a fixed charge for borrowing money"[1]. Whereas, puns using different words with similar soundings are called heterographic. *Are evil wildebeests bad **gnus**?*; here, "gnus" and "news" (/nu:z/)[2] have the same pronunciation.

With such ambiguities, they can usually achieve a humorous or rhetorical effect. Puns can be seen not only as jokes, but also are widely used in literature and can be traced back as early as the Roman playwright Plautus (Pollack, 2011).

Puns can be a challenge to appreciate, even for humans. It requires not only sufficient associations but also rich English and background knowledge. A well-functioning system in machine translation may help non-English users better understanding literary criticism and analysis. Besides, it may also enhance the experience of human-computer interaction (Hempelmann, 2008).

This paper focuses on the detection and location of homographic and heterographic puns. The state-of-the-art systems mostly deployed rule-based strategies and a few purposed complex machine learning models. Their experimental results reached 83 % to 90 % F_1 in pun detection and 80 % in the pun location identification task on the SemEval-2017 Task 7 dataset (Miller et al., 2017). Our contributions are: (1) accumulating a dataset for pun detection; (2) utilizing a logistic regression model to show the relations with straightforward features on both sentence level and word level; (3) unveiling how puns may work according to their type.

2 Related Work

Most previous work focus on pun generation and modelling. Starting in 2004, Taylor and Mazlack (2004) used N-grams to recognize and locate wordplay. In 2015, Miller and Gurevych (2015) adapted knowledge-based Word Sense Disambiguation (WSD) to "disambiguate" different meanings of puns. By processing sentences, Kao

[1] http://wordnetweb.princeton.edu/perl/webwn
[2] https://dictionary.cambridge.org/dictionary/english/

et al. (2016) built an information-theory-based computational model for interpreted puns with "ambiguity" and "distinctiveness".

In the pun detection part, Sevgili et al. (2017) computed PMI scores for every pair of words and looked for the strong associations; Pedersen (2017) applied different settings of WSD approaches to voting for puns; Doogan et al. (2017) calculated phonetic distances for heterographic puns. Several papers also proposed supervised methods: Indurthi and Oota (2017) differentiated puns from non-puns using bi-directional Recurrent Neural Network (RNN) with word embeddings as features. Xiu et al. (2017) also trained a classifier, but on a self-collected training set, with features based on WordNet (Miller, 1995) and word2vec (Mikolov et al., 2013) embeddings. Diao et al. (2019) created the PSUGA model for heterographic puns, which applies a hierarchical attention mechanism to learn phoneme and spelling relations. For pun location identification, Doogan et al. (2017) selected words whose two senses having higher similarity scores with two different content words. Vechtomova (2017) developed eleven features as rules, including position information, PMI, TF-IDF, etc., to score candidate words. Zou and Lu (2019) jointly detected and located puns with tags from an LSTM (Long Short-Term Memory) and CRFs (Conditional Random Fields). Cai et al. (2018) also applied a BiLSTM, but based on sense-aware models.

Two works by Mao et al. (2020) and Zhou et al. (2020) were reviewed and published concurrently. Mao et al. (2020) captured long-distance and short-distance semantic relations between words; Zhou et al. (2020) combined contextualized word embedding and pronunciation embedding with a self-attentive encoder, reaching 2 %, 2 %, 13 % and 7 % increase in F-score on the four tasks respectively.

Previous studies focused on machine learning models or rule-based methods (Diao et al., 2019), however, they are not able to measure how associated a predictor with the purpose or its direction with. Instead of using rules that are mainly based on belief, or deploying neural networks which, due to their intrinsic complexity, indicate no clear clue on the relationship between input and output, we use logistic regression and combinations of widely-used terms. This gives us the best result so far and also provides us a valuable by-product, i.e. helping us to uncover hidden relationships in the puns.

3 Methods

The influence of each feature can be traced based on the results. These terms explore the statistic characteristics of a pun as well as its semantic properties. In general, they can be categorized into the following 4 types.

Part-of-speech (POS) tag: By analyzing the statistics of the dataset, nouns, verbs, adjectives and proper nouns take up about 98 % of all pun words. Besides, a verb-type pun word is almost certain to appear at the end.

Representation of the entire sentence: Pre-trained doc2vec (Le and Mikolov, 2014) and BERT (Devlin et al., 2019) language models are used to get a representation of the sentence as the contextual background for disambiguation.

Sentence separation: Many researchers believe that the pun word often locates in the latter half of a sentence. However, Sevgili et al. (2017), Oele and Evang (2017) lost structure when using PMI and WSD respectively; Vechtomova (2017) failed on most complex sentences by splitting with certain keywords. Instead, we use dependency parsing to extract the largest strict sub-tree in the sentence structure as the second part, leaving the rest as the first part (see Figure 1). It separates sentences and preserves the structure regardless of sentence types.

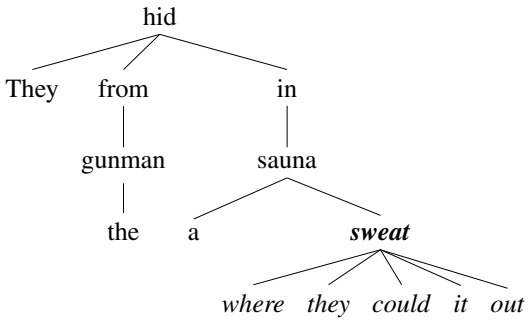

Figure 1: Example sentence for dependency parsing. *e.g., They hid from the gunman in a sauna where they could sweat it out.* After sentence separation: *where they could sweat it out* (important part) and *They hid from the gunman in a sauna* (the rest part).

Word embedding or meaning: We use GloVe (Pennington et al., 2014) to derive word embedding and other approaches like path distances of word senses in WordNet to get meanings for word pairs.

System	P	R	A	F_1
Zhou et al. (2020)[4]	**.942**	**.957**		**.949**
Zou and Lu (2019)[4]	.912	.933		.922
Indurthi and Oota (2017)[5]	.902	.897	.853	.900
Sevgili et al. (2017)	.755	.933	.736	.835
Pedersen (2017)	.783	.872	.736	.825
First setting	**.924**	**.937**	**.900**	**.930**
Second setting	.828	.928	.811	.875

Table 1: Homographic pun detection results with the top three teams from the competition and two best recent studies (the upper part). `f4`, `f5`, `f6`, `f8`, `f9`, `f10` and `f12` are used in both settings: 5-fold CV and the one using own training data.

System	P	R	A	F_1
Zhou et al. (2020)	**.948**	**.956**		**.952**
Zou and Lu (2019)	.867	.931		.898
Diao et al. (2019)[6]	.879	.851	.829	.865
Sevgili et al. (2017)	.773	.930	.755	.844
Doogan et al. (2017)	.871	.819	.784	.844
First setting	**.921**	**.939**	**.899**	**.930**
Second setting	.831	.938	.820	.881

Table 2: Heterographic pun detection with the top two teams from the competition and three best recent studies; the same features as homographic.

	Description
`f1`	The number of words regarding its POS tags.
`f2`	The distance of last appeared POS tags respectively normalized to sentence length.
`f3`	Individual sums of all founded PMI values according to POS tags.
`f4`	doc2vec sentence representation.
`f5`	doc2vec dot product for separated parts.
`f6`	doc2vec for both parts of the sentence.
`f7`	doc2vec cosine similarity of word pair from separated parts of the sentence in descending order. The first 10 values are taken.
`f8`	It has three elements: if the sentence contains a similar idiomatic representation; if it differs exactly one word; how much they are in common.
`f9`	Word similarity based on the shortest path in WordNet. We evaluated with path_similarity[7]. For each sentence, all word pairs from two subsentences are evaluated, and the 4 largest results are chosen.
`f10`	The number of associated words in the first part of the sentence. For each word ω from the second part that exists in Free Association corpus[8], we count how many content words from the first part are listed as associative words of word ω according to Free Association corpus.
`f11`	The number of words that are predicted differently if one word ω is masked, using BERT.
`f12`	Sentence representation using BERT.

Table 3: Feature lists for pun detection.

4 Experiments

4.1 Subtask 1: Pun Detection

Pun detection is a binary classification problem with a sentence as the input, and the decision of whether it is punning as the output.

Data: The published dataset (Miller et al., 2017) contains 2250 contexts for the homographic and 1780 for the heterographic type. We also generated a corpus from "Pun of the Day"[3], which contains mixed types of puns. After removing duplicates, we found 843 puns (disregarding pun types) with a significantly larger standard deviation of sentence length compared to the given corpus. We fitted the dataset by limiting the range of word counts and ended up with 707 puns and a variety of negative samples made of non-punning jokes, famous sayings and other short collections. The compiled dataset can be made available upon request.

Setting: In this subtask, we did experiments on two different settings: 5-fold cross-validation and purely with collected training data. For the first setting, we cross-validated with the official dataset, since it is not split by the provider. To make it comparable to the previous research, we tested all the folds independently and calculated the macro score in the end, thus the final result covers all benchmark data with 5 sub-experiments; for the second one, we trained on self-collected data and evaluated on the official dataset. Both experiment metrics use standard precision, recall, accuracy and F-score.

Features: Table 3 lists the features; among them, `f4`, `f5`, `f6`, `f8`, `f9`, `f10` and `f12` give a positive influence on the final result, and vice versa. To unify, we choose the same feature sets for both homographic and heterographic sets.

Results: Table 1 and 2 provide the experimental results for homographic and heterographic pun detection, respectively. The 5-fold CV utilizes all data provided in the task; the latter one does not use any data from the task for training.

From both results, our system with the first setting (5-fold cross-validation) leads to the best scores, compared with all the teams which used

[3]Pun of the Day: http://www.punoftheday.com/

[4]Both teams used 10-fold cross-validation.

[5]Trained on part of the dataset, evaluated on 675 of the 2250 homographic contexts according to task organizer.

[6]5-fold cross-validation on the original dataset and our compiled corpus from Pun of the Day.

[7]path_similarity from nltk returns a score denoting how similar two senses are, based on the shortest path that connects the senses in the is-a (hypernym/hyponym) taxonomy. http://www.nltk.org/howto/wordnet.html

[8]Free Association is a collection of word pairs that people tend to think first when given the other word: http://w3.usf.edu/FreeAssociation/AppendixC/

System	C	P	R	F_1
Zhou et al. (2020)		**.904**	**.875**	**.889**
Mao et al. (2020)		.850	.813	.831
Zou and Lu (2019)		.835	.771	.802
Cai et al. (2018)		.815	.747	.780
Doogan et al. (2017)	.999	.664	.662	.663
Vechtomova (2017)	.999	.653	.652	.652
Indurthi and Oota (2017)	**1.00**	.522	.522	.522
Our system	**1.00**	.762	.762	.762

Table 4: Homographic pun location results with the top three teams from the competition and three best recent studies; all features except `f2` and `f4` are used.

System	C	P	R	F_1
Zhou et al. (2020)		**.942**	**.904**	**.923**
Mao et al. (2020)		.888	.858	.873
Zou and Lu (2019)		.814	.775	.794
Vechtomova (2017)	.998	.797	.795	.796
Doogan et al. (2017)	**1.00**	.685	.685	.685
Sevgili et al. (2017)	.988	.659	.652	.655
Our system	**1.00**	**.849**	**.849**	**.849**

Table 5: Heterographic pun location results with the best three teams from the competition and three best teams from recent studies; all features except `f7` are used.

part of official data for training (all teams listed above except Sevgili et al. (2017) and Doogan et al. (2017)). Compared with the other teams participating in this subtask, our second setting (training separated) also outperforms them by about 4 %.

Besides, there is a 5 % performance drop from training in the 5-fold CV to the self-collected corpus respectively. This may result from multiple reasons. For instance, the self-collected corpus does not categorize the punning type; the non-punning samples may consist of some hidden puns; the two corpora vary in terms of their properties, etc.

Ablation test: In the ablation test, we found that the major factors are sentence representation (`f4`, `f12`), the relation between both parts (`f5`, `f6`, `f7` and `f10`) and word meaning (`f8`, `f9`). While sentence representation offers the basis, the relation between both parts also helps in general or individual word pairs.

Furthermore, since all features have different dimensions, `f12` occupies more than 2/3 of the feature space, and most of the top 15 % important components are from it. The third parameter in `f8` calculates the maximum ratio of overlapping words to the word length of the found idiomatic representation and always has the largest influence, then comes `f4`, and sometimes `f6`.

4.2 Subtask 2: Pun Location

Pun location is to find out which word in the content is punning, given a pun-containing sentence.

Data: The dataset provided by the organizer includes 1271 homographic puns and 1780 heterographic ones.

Setting: In this subtask, we used 5-fold cross-validation to test. Like in Subtask 1, the official dataset was randomly split into 5 folds. The predictions from each fold were then accumulated and calculated. Scores were computed using standard coverage, precision, recall and F-score measures.

Features: Table 6 lists all the features used in this subtask. All of them are word-based and each assigns a vector or value to words. `f2` and `f4` are only for heterographic since they contain homonymic information. We then concatenate all vectors from chosen features. After logistic regression, words with the highest score presume to be pun location.

Results: Our system achieves competitive results compared to teams from the competition. For location on homographic puns, our model's performance is lower than Zou and Lu (2019) and Cai et al. (2018), which use LSTM (see Table 4). Added pronunciation features (`f2` and `f4`), our system (using all features except `f7`) exceeds the state-of-the-art results by around 5 % for heterographic puns (see Table 5).

Ablation test: Both `f6` and `f7` lead to a significant increment in the result. They give words, especially content words, with latter order more weight. `f3` uses doc2vec to extract relations between separated parts, while `f9` is to find out the "surprise" within a pun using MaskLM. Together, they contribute to around 2 % improvement. These two features answer to our hypothesis, and also tend to focus on the differences between double meanings of the trigger words and the two parts of sentences. `f10` concatenates the GloVe vector to represent the word itself. It results in an approximately 7 % boost. Homonymic information (`f2`) helps, but still left much to explore. First, the data is heterographic instead of homophonic (*e.g.*, "orifice" and "office"). Besides, variances of the word are not considered (*e.g.*, "knowingly" and "no"). Third, puns may exploit names or compounds (*e.g.*, "Clarence" and "clearance").

This problem is remedied by adding word fre-

	Description
f1	Assign value 1 to the last content word in the sentence. Namely, if the feature is used, the last content word in sentences will be concatenated with vector [1], while a vector [0] for the other words.
f2	If there is the same pronunciation of the word in CMU Pronouncing Dictionary[9].
f3	Maximum doc2vec cosine similarity of word pair from separated parts of the sentence.
f4	The number of context words that have lower doc2vec cosine similarity with word ω than with any of ω's homonyms.
f5	Assign value 0, 1 or 2 for word ω according to its frequency in Brown corpus[10]. The rarer the word is, the higher the value it is assigned.
f6	The position of word ω in the sentence, assign 1 if it is in the second half, plus additional 1 if it also lays in the last quarter.
f7	Mark last N, V, Adj, Propn in form of a vector at their place (from subtask 1). For example, if word ω is the last verb in the sentence, its vector for this feature should be [0,1,0,0].
f8	doc2vec of the whole sentence (from subtask 1).
f9	The number of context words that are predicted differently using BERT if word ω is masked.
f10	GloVe vector of the word.

Table 6: Pun location features and description.

quency feature (f5) instead of given special attention to particular names or structure patterns. Although this feature works significantly in heterographic, it barely influences the homographic ones. Unlike heterographic puns, one needs a word with two senses that are widely known to people in the homographic case. So a rare word can hardly be used in that situation.

5 Conclusion and Future Work

We provide a dataset for pun detection and built a model that achieves state-of-the-art on three subtasks for different types of puns. We found that three things affect a pun: general interpretation of the content, relation for both parts and word meaning. In case we know it is punning as a prior, we can utilize *e.g.*, word position or "surprise" according to their type, to locate the punning word.

We deployed homophones to heterographic tasks; this could be an interesting topic for future work as well as a test of higher-order associations between word pairs. Nevertheless, with the improved results of pun detection and interpretation, our system provides a step for further understanding and interpretation, and may be assembled into machine translation in the future.

[9]Carnegie Mellon University (CMU) Pronouncing Dictionary is an open-source pronunciation dictionary: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[10]It is a general English-language corpus with a total of roughly one million words: https://archive.org/details/BrownCorpus

Acknowledgments

We thank the anonymous reviewers for suggestions on the submission; the paper was partially supported by the German Academic Exchange Service and partially supported by base.camp at Universität Hamburg.

References

Yitao Cai, Yin Li, and Xiaojun Wan. 2018. Sense-aware neural models for pun location in texts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 546–551, Melbourne, Australia.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, USA.

Yufeng Diao, Hongfei Lin, Liang Yang, Xiaochao Fan, Di Wu, Dongyu Zhang, and Kan Xu. 2019. Heterographic pun recognition via pronunciation and spelling understanding gated attention network. In *The World Wide Web Conference*, pages 363–371, San Francisco, CA, USA.

Samuel Doogan, Aniruddha Ghosh, Hanyang Chen, and Tony Veale. 2017. Idiom savant at SemEval-2017 Task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 103–108, Vancouver, Canada.

Christian F Hempelmann. 2008. Computational humor: Beyond the pun. *The Primer of Humor Research. Humor Research*, 8:333–360.

Vijayasaradhi Indurthi and Subba Reddy Oota. 2017. 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 (SemEval-2017)*, pages 457–460, Vancouver, Canada.

Justine T Kao, Roger Levy, and Noah D Goodman. 2016. A computational model of linguistic humor in puns. *Cognitive science*, 40(5):1270–1285.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, Beijing, China.

Junyu Mao, Rongbo Wang, Xiaoxi Huang, and Zhiqun Chen. 2020. Compositional semantics network with multi-task learning for pun location. *IEEE Access*, 8:44976–44982.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe, NV, USA.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of English puns. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 719–729, Beijing, China.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. Semeval-2017 task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada.

Dieke Oele and Kilian Evang. 2017. Buzzsaw at SemEval-2017 Task 7: Global vs. local context for interpreting and locating homographic English puns with sense embeddings. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 444–448, Vancouver, Canada.

Ted Pedersen. 2017. Duluth at SemEval-2017 Task 7: Puns Upon a Midnight Dreary, Lexical Semantics for the Weak and Weary. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 416–420, Vancouver, Canada.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

John Pollack. 2011. *The Pun Also Rises: How the Humble Pun Revolutionized Language, Changed History, and Made Wordplay More Than Some Antics*. Penguin, New York, NY, USA.

Özge Sevgili, Nima Ghotbi, and Selma Tekir. 2017. N-Hance at SemEval-2017 Task 7: A Computational Approach using Word Association for Puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 436–439, Vancouver, Canada.

Julia M Taylor and Lawrence J Mazlack. 2004. Computationally recognizing wordplay in jokes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 26, Chicago, IL, USA.

Olga Vechtomova. 2017. Uwaterloo at SemEval-2017 Task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 421–425, Vancouver, Canada.

Yuhuan Xiu, Man Lan, and Yuanbin Wu. 2017. Ecnu at semeval-2017 task 7: Using supervised and unsupervised methods to detect and locate English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 453–456, Vancouver, Canada.

Yichao Zhou, Jyun-Yu Jiang, Jieyu Zhao, Kai-Wei Chang, and Wei Wang. 2020. "The Boating Store Had Its Best Sail Ever": Pronunciation-attentive contextualized pun recognition. *arXiv preprint arXiv:2004.14457*.

Yanyan Zou and Wei Lu. 2019. Joint detection and location of English puns. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2117–2123, Stroudsburg, PA, USA.