

AKRaNLU @ CLEF JOKER 2023: Using Sentence Embeddings and Multilingual Models to Detect and Interpret Wordplay

Ryan Rony Dsilva

Purdue University, West Lafayette, IN, USA

Abstract

In this paper, we present our work for the Automatic Wordplay Analysis (JOKER) Lab at CLEF 2023. The objective of the JOKER Lab is to advance the field of automatic methods for interpreting, generating, and translating wordplay. Our participation involved two specific tasks: pun detection and pun location with interpretation. In pun detection, we employed sentence embeddings to classify puns, while for the second task, we treated the pun location sub-task as a token classification task using XLM-RoBERTa. To interpret puns, we utilized sentence embeddings in conjunction with WordNet to identify the intended senses of the pun word. We present experiments on the training data, and the results for pun detection and location on the test dataset are summarized in tables for English, French, and Spanish puns, along with an analysis of errors.

Keywords

sentence embeddings, puns, wordplay, token classification, wordnet

1. Introduction

Wordplay refers to a literary technique that relies on words that sound alike but have different meanings. Wordplay can be crafted using words that share the same pronunciation and spelling, words with different spellings but the same pronunciation, and words with different spellings and similar pronunciations [1]. The CLEF 2023 JOKER [2] track proposed three tasks - Task 1: Detection of puns in English, French, and Spanish; Task 2: Location of puns in English, French, and Spanish, and Interpretation of puns in English and French, and Task 3: Translation of puns from English to French and Spanish. We participated in tasks 1 and 2. The previous edition of this competition set the foundations for the direction of automatic wordplay analysis. We use sentence embeddings with binary classification to detect puns and treat pun location as a token classification task to predict labels: (1) - for a punning word or (0) - for not a punning word for every word in the sentence. For the interpretation of puns, we compare the similarity of the embeddings of the synonyms of the located pun word with the overall sentence embedding and pick the top two senses of the word. We summarise the results of our experiments and provide an analysis of errors to aid future research.

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18-21, 2023, Thessaloniki, Greece


✉ dsilvar@purdue.edu (R. R. Dsilva)

🌐 <https://www.ryandsilva.dev/> (R. R. Dsilva)

🆔 0009-0008-4158-4504 (R. R. Dsilva)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

2. Methodology

2.1. Data Preparation

The data from [2] was used, and text preprocessing was applied. All input sentences were read in the UTF-8 encoding, converted to lowercase, and punctuation was stripped from the sentences except for hyphens. The decision to keep the hyphens was made after examining the dataset, wherein many instances included hyphens in the pun word. After preprocessing, 375 instances of pun words from the training set changed their form: “Lee, Chamorro” became “lee chamorro,” “en? - ...fermo” became “en - fermo,” “Stan, Lee” became “stan lee” and so on. This was observed for proper nouns like names, pun words with special characters like punctuation, pun words that had two forms (bovine | divine), and words that appeared in another form that did not appear exactly in the sentence (dégainait). For the pun location task, these instances were excluded from the training set using a Python script. The exclusion criteria were only applied to the training data, while preprocessing was applied to both training and test data. The final size of the training dataset for the pun location task was 4817 instances for English, Spanish, and French combined.

2.2. Pun Detection

Two main strategies were used for pun detection: sentence embeddings [3] with a binary classifier and sequence classification using XLM-RoBERTa [4].

The intuition for using sentence embeddings stems from the basic idea of word vectors and how they can be used to represent a word. Similarly, we wanted to explore if sentence embeddings could capture the representation of the entire sentence such that certain qualities about the sentence can be learned, which in this case is whether the given sentence is a pun or not. We used the `paraphrase-multilingual-mpnet-base-v2` model introduced in [5] to get the sentence embeddings. These sentence embeddings are of size 768, which are used as input to a classifier along with the labels 0 or 1 for each sentence. We experimented with various classifier models like SVC [6] (Linear, Polynomial¹, and RBF²), Random Forests [7], and logistic regression but then built a neural network for the task. We built a 6-layer neural network with a classifier head which gave us the best results across all three languages: English, French, and Spanish. The idea for the neural network stems from ColBERT [8], where a supervised binary classifier was built to use BERT embeddings as input to detect humor. We used the Sentence Transformers library [3] to generate the sentence embeddings and scikit-learn [9] and Keras [10] to build the classification models. The results of our experiments on the training data are summarized in Table 1.

The second method we experimented with was sequence classification using the XLM-RoBERTa-Large model. We used the HuggingFace Transformers library [11] to implement this.

We also conducted experiments to evaluate the impact of the training data size on our models. These results show an improvement in performance with an increase in dataset size. However,

¹Polynomial SVC with degree=3

²RBF SVC with $\gamma=0.05$ and $C=1$

Table 1
Metrics for Pun Detection - Training Data

	Precision	Recall	Accuracy	F-Score
SVC (Linear)	0.5800	0.5700	0.5700	0.5700
SVC (Poly-3)	0.5500	0.5500	0.5500	0.5400
SVC (RBF-0.05-1)	0.5900	0.5800	0.5800	0.5600
Random Forests	0.5200	0.5200	0.5200	0.5100
Gradient Boosted Forests	0.5800	0.5700	0.5700	0.5600
Logistic Regression	0.5800	0.5800	0.5800	0.5700
Custom NeuralNet	0.6200	0.6200	0.6200	0.6100
XLM-RoBERTa-Large	0.5261	1.0000	0.5261	0.6895

Table 2
Metrics for Pun Detection - Impact of Training Data Size

	Precision	Recall	Accuracy	F-Score
SentEmb-NeuralNet with 25%	0.5600	0.5600	0.5600	0.5400
SentEmb-NeuralNet with 50%	0.5800	0.5800	0.5800	0.5600
SentEmb-NeuralNet with 75%	0.5700	0.5700	0.5700	0.5700
SentEmb-NeuralNet with 100%	0.6200	0.6200	0.6200	0.6100
XLM-RoBERTa-Large with 25%	0.4994	1.0000	0.4994	0.6662
XLM-RoBERTa-Large with 50%	0.7048	0.4238	0.6235	0.5293
XLM-RoBERTa-Large with 75%	0.6441	0.5365	0.6204	0.5854
XLM-RoBERTa-Large with 100%	0.5261	1.0000	0.5261	0.6895

the results are not convincing enough to say that increasing the dataset size will necessarily improve results. The findings are presented in Table 2.

2.3. Pun Location

We employed the token classification method for locating puns as demonstrated by [12]. We used the NP tagging scheme because it could capture instances where there could be more than one word that formed the pun. Based on the training split, we assigned a tag of 1 to every pun word and 0 to every word that is not a pun word. We used the XLM-RoBERTa-Large model to perform this token classification.

2.4. Pun Interpretation

Interpretation of puns used the results from the pun location subtask to disambiguate the appropriate senses of the pun word according to the sentence and find synonyms for those senses. We used WordNet³ [13] as our sense dictionary. More specifically, since we used WordNet from the *nltk* library, we used the Open Multilingual WordNet [14], which gave us access to WordNet in English, French, and Spanish. We first collected the synonyms of the pun

³<https://www.nltk.org/howto/wordnet.html>

word from WordNet and then computed the similarity between the word embedding of each synonym and the sentence embedding of the text. Both these embeddings are of size 768, and we use the cosine similarity function from the sentence transformers library. The intuition for this stems from the idea that embeddings of the synonyms of the senses that most accurately fit the sentence would be closer to the sentence in the vector space. The synonyms in WordNet each represent a distinct concept. Hence the top-2 synonyms were selected to be the most appropriate for the pun word in the sentence.

All the code for this work is made available on GitHub⁴.

3. Results

The metrics of precision, recall, accuracy, and f-score are reported for Task 1, and accuracy is reported for Task 2, Sub-Task 1. The metrics in the tables below are computed from the test dataset.

3.1. Pun Detection

Tables 3, 4, and 5 summarise the results for our runs. We submitted classifications for 3183 instances in English, 12873 instances in French, and 2230 instances in Spanish. The low scores suggest that sentence embeddings used directly may not capture the property of the sentence that makes the sentence a pun.

Furthermore, the dataset had sentences that were augmented in such a way that the pun word was deliberately replaced with a word that still makes sense in the context but makes the sentence such that it is not a pun anymore. The methods used here failed to capture that aspect accurately. The XLM-RoBERTa-Large model particularly suffered due to this and over-predicted the puns to such an extent that for English and French, no true negatives were predicted. Some examples of incorrect classifications include:

Surfing is a swell sport!
Actual: 1 Predicted: 0

I used to be a banker but I lost motivation.
Actual: 0 Predicted: 1

In the first example, the model fails to recognize the nuanced meaning of the word "swell," which could mean excellent, great, or fantastic (informally) or the rising and falling motion of the waves. Informal definitions and slang might not be accurately captured with this approach. The second example demonstrates what the authors of the dataset describe as a negative example where the pun word of "interest" was substituted to be "motivation," which has a similar sense but loses the quality that makes this sentence a pun.

⁴<https://github.com/RyanDsilva/clef-2023-joker>

Table 3
Metrics for Pun Detection - English

	Precision	Recall	Accuracy	F-Score
SentEmb-NeuralNet	0.2630	0.8640	0.3500	0.4032
XLM-RoBERTa-Large	0.2542	1.0000	0.2542	0.4053

Table 4
Metrics for Pun Detection - French

	Precision	Recall	Accuracy	F-Score
SentEmb-NeuralNet	0.4118	0.7389	0.4572	0.5289
XLM-RoBERTa-Large	0.4123	1.0000	0.4123	0.5839

Table 5
Metrics for Pun Detection - Spanish

	Precision	Recall	Accuracy	F-Score
SentEmb-NeuralNet	0.4140	0.7227	0.4476	0.5264
XLM-RoBERTa-Large	0.4256	0.9969	0.4270	0.5965

Table 6
Pun Detection - Examples

Sentence	Pun Word	Predicted
Weather forecasters have to have lots of degrees.	degrees	degrees
Quand des éléphants entrent dans un bar, le patron sait qu'il peut s'attendre à des gros pour boire.	des gros pour boire	gros
C'est entre mon nez et mon menton, dit Tom la bouche encœur.	la bouche encœur	la bouche encœur
Some people with a lot of vision started the blind institute.	vision	blind

3.2. Pun Location

Table 7 summarises the results for submission for the pun location task. We submitted results for 1205 instances in English, 4655 instances in French, and 960 instances in Spanish. Partial accuracy refers to the accuracy calculated from the submitted instances instead of the entire test dataset. One flaw noticed during experiments was that our model could not always capture instances where the pun word was a phrase of multiple words instead of a single word. Some examples of predictions are included in Table 6.

The predictions show a variety of findings. The model does have the capability to detect a pun phrase, but it is not consistent. The last example is interesting because it picked the correct concept from the sentence (vision-blind) but failed to mark the right word, which makes it a pun. The word "vision" should have been predicted due to its dual meaning of the ability to see

Table 7
Accuracy for Pun Location

	Accuracy	Partial Accuracy
English	0.7917	0.7917
French	0.4136	0.4713
Spanish	0.5615	0.5615

Table 8
Pun Interpretation - Examples

Sentence	Interpretation	Predicted
This is where I keep my arrows, said Tom, quivering.	palpitate; quake; quiver / quiver	beat; pulsate; quiver / palpitate; quake; quiver
News of a coming flood was leaked.	leak / leak	leak; leak out / leak
"It's a unit of electric current," said Tom amply.	richly / A; amp; ampere	richly / fully

and the ability to plan the future with wisdom. An overall accuracy of 79% was obtained on the training dataset.

3.3. Pun Interpretation

The results for the test dataset were not ready in time for this paper, and hence only analysis of the experiments on the training data is provided. Some examples of interpretation outputs⁵ are included in Table 8.

In the examples of common words like "quiver" and "leaked," our solution performed well, but for words that did not have the same form across senses, like in the example for "amply," our system could not predict the correct interpretation. Here, "amply" was the form for one sense which means "richly," and "amp" was the other form which means "ampere." WordNet in other languages is also less extensive than the one for English, which limits this approach. Matching synonyms that included special characters in French, like carets, returned no results, which resulted in no output from the system.

4. Conclusion and Future Scope

Our implementation of sentence embeddings to detect properties of sentences could prove to be the basis when used in conjunction with other aspects, such as phonetics, to identify puns. While building the neural network for classification, we were limited by time and believed that a better model could be built to detect the properties of sentence embeddings that make it a pun. Moreover, the dataset could be augmented to add more negative samples from other pun datasets. A better way of handling circumflex accent (caret) marks and the instances marked by

⁵In Table 8, the symbol "/" separates the two senses and ";" separates alternate words for the same sense.

the exclusion criteria could further improve the system’s capabilities. Furthermore, WordNet could not match multi-word instances to their appropriate synonyms for the interpretation task, particularly for French instances. Leveraging the large knowledge base of LLMs might prove useful for interpretation tasks, but we did not explore those areas in this paper.

References

- [1] J. M. Taylor, L. J. Mazlack, Computationally recognizing wordplay in jokes, in: Proceedings of the Annual Meeting of the Cognitive Science Society, volume 26, 2004.
- [2] L. Ermakova, T. Miller, A.-G. Bossler, V. M. Palma Preciado, G. Sidorov, A. Jatowt, Overview of JOKER - CLEF-2023 track on Automatic Wordplay Analysis, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), LNCS, Springer, 2023.
- [3] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: <https://arxiv.org/abs/1908.10084>.
- [4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.
- [5] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2020. URL: <https://arxiv.org/abs/2004.09813>.
- [6] C. Cortes, V. Vapnik, Support-vector networks, *Machine learning* 20 (1995) 273–297.
- [7] L. Breiman, Random forests, *Machine learning* 45 (2001) 5–32.
- [8] I. Annamoradnejad, G. Zoghi, Colbert: Using bert sentence embedding for humor detection, *arXiv preprint arXiv:2004.12765* 1 (2020).
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [10] F. Chollet, et al., Keras, <https://keras.io>, 2015.
- [11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [12] Y. Zou, W. Lu, 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), Association

- for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 2117–2123. URL: <https://aclanthology.org/N19-1217>. doi:10.18653/v1/N19-1217.
- [13] C. Fellbaum, WordNet: An Electronic Lexical Database, Bradford Books, 1998. URL: <https://mitpress.mit.edu/9780262561167/>.
- [14] F. S. Madonsela, M. J. Mafela, M. L. Mojapelo, M. R. Masubelele, Proceedings of the 8th global wordnet conference, gwc 2016, in: Global WordNet Association, 2016.