CLEF 2023 JOKER Tasks 2 and 3: Using NLP Models for Pun Location, Interpretation and Translation

Felix Ohnesorge[1], Mari Ángeles Gutiérrez[2]and Julia Plichta[3]

1 University of Kiel, 24118 Kiel, Germany
2 University of Cádiz, 11003, Cádiz, Spain
3 University of Gdańsk, 80-309 Gdańsk, Poland

Abstract

The translation of puns presents a significant challenge for translators and has garnered considerable attention in the field of translation studies. While translation technology aims to automate the translation process, little focus has been placed on the translation of wordplay. Addressing this gap, the CLEF2023 JOKER track aims to develop a multilingual corpus of wordplay and evaluation metrics to advance the automation of creative-language translation. This paper provides an overview of the track's Pilot Task 3, which specifically focuses on the translation of entire phrases containing wordplay, particularly puns.

Keywords

Wordplay, computation humour, pun, machine translation, deep learning

1. Introduction

As Ermakova *et al.* explain [1], humorous wordplay comprehension and translation often require recognizing implicit cultural references, understanding word formation processes, and discerning double meanings. These challenges are encountered by both humans and computers alike. Introducing the CLEF 2023 JOKER track, this paper adopts an interdisciplinary approach to facilitate the creation of reusable test collections, evaluation metrics, and methods for automatic wordplay processing. The track comprises interconnected shared tasks that encompass the detection, location, interpretation, and translation of puns. Furthermore, associated datasets and evaluation methodologies are described, and contributions leveraging this data are invited for further research and development.

Through the collaborative efforts of the CLEF JOKER track, advancements in automating the translation of wordplay are expected, paving the way for improved translation technology and enhancing cross-cultural communication in creative language contexts.

2. Related work

2.1. Task 1: Pun Detection

A pun is a form of wordplay that exploits multiple meanings or similar sounds of words to create a humorous or clever effect. Puns can be found in various forms, including plays on words, double entendres, homophones, and wordplay involving idioms or metaphors, which implies nuanced linguistic and cultural knowledge, making it difficult for algorithms to capture the full range of punning possibilities.

However, the methods based on natural language processing (NLP) and machine learning

[1] CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
EMAIL: felix-eutin@gmx.de (A. 1), mariangeles.gutierrez@uca.es (A. 2), j.plichta.162@studms.ug.edu.pl (A. 3)
ORCID: 0000-0001-7220-1846 (A. 2)

techniques examine factors such as word similarity, part-of-speech tags, syntactic patterns, and context, so, by comparing the different senses or meanings of words and their surrounding context, algorithms can identify potential puns.

In our case, we have been provided with the JOKER dataset, which contains around 5000 sample sentences, which are labeled with either a "yes" or a "no". This is not a big dataset when considering the high dimensionality we have in text classification. Normally we would use a cross-validation technique but that requires a lot of time and computational power, which google colab could not provide enough of. So, we used a 60/20/20 split on our dataset for training, testing and evaluation.

The pun detection is a binary classification task, for which we used several different methods. The methods (some of them quite simple and only as a reference) will be explained in detail later. In addition, all statistics can be found in Ermakova *et al.* [2]

Table 1
Results: Pun Detection

Method	Accuracy (F1)
Random	49%
Fasttext	70%
Ridge	76%
Bayes	62%
MLP	75%
SimpleTransformersT5	70%
SimpleTransformersRoberta	75%

2.1.1. Random

This Method yielded an accuracy of 49%, which is expected because our evaluation data contained 50% of "yes" samples and 50% of "no" samples.

2.1.2. Fasttext

This is an open-source library for text classification. It comes pretrained in multiple languages, which should help us because we only need to fine tune the model.

The trained model achieved a 70% accuracy on our evaluation data, with an equally distributed confusion matrix.

2.1.3. Ridge

Ridge is a classifier that works by regression and is most commonly used for multiclass classifications but can of cause also be used for binary classification.

This is not a pretrained model, so we had to train it with our available data exclusively. The model still achieved a slightly higher accuracy on our evaluation dataset, 76%.

2.1.4. Bayes

Bayes is a well-known classifier for multimodal classification. It is basically a simple probability calculation. Here an interesting thing happened. We achieved an accuracy of 62%, but the confusion matrix showed zero true positives and zero false positives. This indicates an underfitting.

2.1.5. MLP

We used the MLPClassifier from the sklearn.neural_network library. This is a multi-layer perceptron that optimizes the loss function by stochastic gradient descent. This is a very powerful method, but it has many parameters that can be adjusted. For example, the number of neurons per layer, the number of layers. As a solver we used Adam, as it is the most used one. As an activation function we used ReLU which is also widely used.

This multi-layer perceptron achieved an accuracy of 75%, scoring slightly behind Ridge.

2.1.6. SimpleTransformersT5

SimpleT5 models can be used for many tasks such as summarization, translation, text generation, and more. We used it for a binary classification which we did not expect to perform very well because it is not the intended task and is way too powerful. This showed i n training, which took an exceedingly long time and the model started overfitting after the second epoch. The only parameter we adjusted was the target_max_token_length, because we only wanted it to answer "yes" and "no" or 0 and 1, respectively.

As expected, the model only had a 70% accuracy on our evaluation data, which is slightly lower than most of the other methods.

2.1.7. SimpleTransformersRoberta

This is a transformer model with a binary text classification model on top of it. We expect this model to perform better than the others because it seems best fit to this problem.

The trained model achieved a 75% accuracy on our evaluation dataset.

2.2. Task 2: Pun Location

For the pun location, the main purpose is to identify which words carry the double meaning in a text known a priori to contain a pun. We have used the following methods, which will be explained in more detail later:

Table 2

Results: Pun Location

Method	Accuracy (F1)
SimpleT5	87%
SimpleTranformersT5	83%
Ridge	50%
Bayes	2%
Fasttext	5%

The metrics used for evaluation are:

- *Precision*: it measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives).
- *Recall*: it measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives).
- *F1-score*: it is the harmonic mean of precision and recall and provides a balanced measure of the model's performance. It takes both precision and recall into account and is particularly useful when dealing with imbalanced datasets.
- *Support*: The support column indicates the number of instances (samples) in the dataset that belong to each class.

A very big problem was the size of available data. For the sexism detection task, we had 7000 labeled tweets and for the pun detection around 5000 sample sentences. When training a neural network that has not been pretrained, this is not enough due to the high feature dimensionality of this task. Here it may be interesting to use different vectorization methods and analyze their impact on the result.

Normally when dealing with samples for training/evaluation a good technique would be the cross validation where all available data is split into multiple batches. We would then train a model on all batches but one and use the left-out batch for evaluation. This would be repeated for every batch. Unfortunately, this requires a lot of computing power and time, both of which we did not have. So, we used an 80/20 split on the training data for training/evaluation. For the ML models we split the

training data into a training and test set, again in an 80/20 relation.

The effectiveness of Bayes, as a method for multimodal classification, is limited in the pun location task due to an excessive number of modalities or an absence of a finite set of modalities. Bayes is a naïve classification method based on probabilities with multiple classes. In the pun detection setting we have a two-class decision problem with can be realized by the bayes classifier. Additionally, Fasttext and Ridge, both commonly employed in classification problems, are anticipated to exhibit suboptimal performance in this context.

Instead, pretrained models like SimpleT5 are expected to demonstrate superior performance in this task, which is why we have used them.

2.2.1. Results for SimpleT5

From the extensive list of classified terms, it can be said that the majority have reached 1.0 accuracy, which underlines the point that we did not have enough data. So, in summary, the SimpleT5 model achieved an accuracy of 87% on the classification task against our evaluation data set. The macro average and weighted average metrics indicate the overall performance across classes, with values of 0.77 and 0.87, respectively.

2.2.2. Results for SimpleTransformersT5

In the same line as the previous method, it achieved an accuracy of 83% on the classification task. The macro average and weighted average metrics indicate the overall performance across classes, with values of 0.71 and 0.70 for precision, recall, and F1- score. However, without the specific values of the confusion matrix, we cannot further analyze the model's performance for each class or discern the distribution of the predicted and true labels.

2.3. Task 3: Pun translation

In the context of pun translation, several models were employed, including those pretrained on translation tasks, as well as FastText and SimpleT5. Notably, FastText exhibited subpar performance due to its classification-oriented nature, which may not be ideally suited for the intricacies of pun translation. Similarly, SimpleT5 did not yield notably satisfactory results. However, it is worth mentioning that specific quantitative performance metrics are unavailable, precluding a more detailed assessment of their respective performances.

Regarding to the dataset with which we have worked, translations from English into French and Spanish have been obtained using two NPL models: OpusMT and M2M100.

From 1000 sentences provided, 30 have been selected to check the quality of the resulting translation. Taking said pre-selection as a reference, 96.7% of these sentences had obtained a good result in the Spanish translation. The sentences translated by OpusMT were more accurate, while M2M100 tends to make free translations, therefore meaningless occasionally.

In general, we have obtained very literal translations, so that is why 60% failed to recreate the pun in the target language. Following the “wordplay translation strategies” referenced by Ermakova *et al* [4], in those sentences where the pun also works in the target language, they have performed an isomorphic pun (that is, they have used the direct translation of the words because it coincides with the main meaning that gives sense to the pun). For example, in the sentence “When you're wearing a watch on an airplane, time flies” the translation is very direct, so in Spanish it retains the same meaning when the result shows: “Cuando llevas un reloj en un avión, el tiempo vuela”.

This implies that puns based on homophonic sounds are not detected or translated (ST: “You can't trust a tiger. You never know when he might be lion”; TT: “No puedes confiar en un tigre. Nunca se sabe cuándo podría ser león”), just like puns based on polysemous terms (ST: “If there's one person you don't want to interrupt in the middle of a sentence it's a judge”; TT: “Si hay una persona que no quieres interrumpir en medio de una sentencia es un juez” where the word ‘sentence’ loses one of its meanings, in this case ‘expression’ or ‘group of words’ besides ‘a punishment given by a judge’.

However, it is very interesting to see how OpusMT recognizes the wordplay’ semantic field in order to make a coherent translation: ST: “The designer wondered why his pirate room wasn't perfect, and the judge told him he went a little overboard”; TT: “El diseñador se preguntó por qué su habitación pirata no era perfecta, y el juez le dijo que se fue un poco por la borda” maintaining

references to vocabulary related to the sea, while

M2M100 made a neutral translation: “El diseñador se preguntó por qué su sala de piratas no era perfecta, y el juez le dijo que iba un poco por encima”.

It is also curious how OpusMT is able to distinguish a formal context: the second person 'you' has two translations in Spanish: 'tú' for informal situations (under certain pragmalinguistic conditions) and 'usted' for formal situations. In the following example, this model has been able to recognize a situation that requires polite language: ST: “Waiter, there are pennies in my soup!" Well, sir, you said you'd stop eating here if there wasn't some change in the food”; TT: “ ‘¡Camarero, hay centavos en mi sopa!’ Bueno, señor, usted dijo que dejaría de comer aquí si no había algún cambio en la comida”.

On the other hand, in a significantly lower proportion, we have found a few examples of pun’s omissions (“pun to zero”, according to Delabastita lists, cited in Ermakova *et al*. [3]): ST: “OLD POLICEMEN never die they just cop out”; TT: “Los viejos policías nunca mueren”.

3. Conclusion

In conclusion, this paper focused on the interdisciplinary approach taken in the CLEF 2023 JOKER track to address the challenges of wordplay comprehension, detection, location, interpretation, and translation. We have described the creation of reusable test collections, evaluation metrics, and methods for automatic wordplay processing.

For the task of pun detection, various methods were employed, including Random, Fasttext, Ridge, Bayes, MLP, SimpleTransformersT5, and SimpleTransformersRoberta. Each method was evaluated using accuracy (F1) as the performance metric, with Ridge and MLP achieving the highest accuracies of 76% and 75%, respectively. These results demonstrated the effectiveness of machine learning and neural network-based approaches in identifying puns.

In the pun location task, we used methods such as SimpleT5, SimpleTransformersT5, Ridge, Bayes, and Fasttext. The evaluation metrics used included precision, recall, F1-score, and support. SimpleT5 achieved the highest accuracy of 87%, indicating its efficacy in identifying the words carrying double meanings in puns.

Regarding pun translation from English into Spanish, different models were utilized, including OpusMT, M2M100, FastText, and SimpleT5. The translations obtained were mostly literal, resulting in a 60% failure rate in recreating the puns in the target language. The translation models struggled to capture homophonic sounds and polysemous terms, leading to the loss of puns based on those linguistic elements. OpusMT showcased the ability to recognize the semantic field of wordplay, while M2M100 provided more neutral translations.

Overall, the CLEF JOKER track and the methods employed in this paper showcased promising advancements in automating wordplay processing. The results highlighted the potential of machine learning, neural networks, and pretrained models in tackling the complexities of pun detection, location, and translation. Further research and development in this area can contribute to improved language technology and enhanced cross-cultural communication in creative language contexts.

4. Acknowledgements

At this point we want to thank the Erasmus program, the European University of the Seas (SEA-EU) and the Instituto de Investigación en Estudios del Mundo Hispánico (In-EMHis) for giving us the opportunity to take this course that promises to be innovative in our academic careers. We also want to acknowledge all the work done by the whole team of the CLEF 2023 project, especially Liana Ermakova, who hosted the Artificial Intelligence’s course and Séverine Allain and Ludmila Guillotin for the excellent coordination and the warm welcome from the Université de Bretagne Occidentale.

5. References

[1] Liana Ermakova, Fabio Regattin, Tristan Miller, Anne-Gwenn Bosser, Sílvia Araújo, Claudine Borg, Gaëlle Le Corre, Julien Boccou, Albin Digue, Aurianne Damoy, Paul Campen, and Orlane Puchalski. “Overview of the CLEF 2022 JOKER Task 1: Classify and explain instances of wordplay”. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, Proceedings of the Working Notes of CLEF 2022 -- Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th to 8th, 2022, volume 3180 of CEUR Workshop Proceedings (2022): 1641-1665.

[2] Liana Ermakova, Tristan Miller, Anne-Gwenn Bosser, Victor Manuel Palma Preciado,

Grigori Sidorov, and Adam Jatowt. 2023. Overview of JOKER - CLEF-2023 track on Automatic Wordplay Analysis. In Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Anastasia Giachanou, Dan Li, Mohammad Aliannejadi, Michalis Vlachos, Guglielmo Faggioli, Nicola Ferro (Eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023).

[3] L. Ermakova, T. Miller, F. Regattin, A. G. Bosser, C. Borg, E. Mathurin, G. Le Corre, S. Araújo, R. Hannachi, J. Boccou, A. Digue, A. Damoy and B. Jeanjean, Overview of JOKER@CLEF 2022: Automatic Wordplay and Humor Translation Workshop, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5-8, 2022, Proceedings, Sprinter-Verlag, Berlin, Heidelberg, 2022, pp. 447-469. doi: 10.1007/978-3-031-13643-6_27.

[4] Liana Ermakova, Fabio Regattin, Tristan Miller, Anne-Gwenn Bosser, Claudine Borg, Benoît Jeanjean, Élise Mathurin, Gaëlle Le Corre, Radia Hannachi, Sílvia Araújo, Julien Boccou, Albin Digue, and Aurianne Damoy. “Overview of the CLEF 2022 JOKER Task 3: Pun Translation from English into French”. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, Proceedings of the Working Notes of CLEF 2022 -- Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th to 8th, 2022, volume 3180 of CEUR Workshop Proceedings (2022): 1681-1700.