

Exploring Humor in Natural Language Processing: A Comprehensive Review of JOKER Tasks at CLEF Symposium 2023

Aftab Anjum^{1,†}, Nikolaus Lieberum^{1,†}

¹Christian-Albrechts-Universität zu Kiel, Christian-Albrechts-Platz 4, 24118 Kiel, Germany

Abstract

As linguistic phenomena that showcase the richness and complexity of human language, puns pose significant challenges to natural language processing (NLP) systems. The significance of these tasks lies not only in the enhancement of humor recognition capabilities in AI but also in the improvement of machine translation systems, as puns often encapsulate cultural, idiomatic, and context-sensitive information [1]. We investigate a broad range of models, including traditional machine learning methods, such as Random Forests and Naive Bayes, and state-of-the-art deep learning architectures, like Transformer and BERT-based models. Experimental results show promising advancements in these areas, with specific models outperforming others depending on the task. The results contribute valuable insights towards the goal of improving the understanding, detection, and translation of puns in AI systems, ultimately promoting a more nuanced and culturally sensitive communication interface in AI technologies.

Keywords

Pun, Wordplay, Natural Language Processing, Computational Humour Detection, Humour Location, Machine Translation, Neural Networks

1. Introduction

Humor plays a significant role in human communication, and puns are a common form of linguistic humor. Wordplay, characterized by the creative manipulation of language rules, is a widely utilized source of humor across various creative fields such as literature, poetry, theater, advertising, and more. Its ability to capture attention, convey playfulness, and subvert expectations makes it a favored technique in titles, headlines, proper nouns, and slogans. Consequently, there is a high demand for the translation of wordplay.

However, despite the advancements in translation technology, there is a notable absence of specific support for humor and wordplay in current translation tools. The automation of humor and wordplay translation has received limited attention in research. Additionally, most AI-based translation systems heavily rely on training data, such as parallel corpora, which has historically lacked sufficient quantity and quality when it comes to humor and wordplay.

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

*Corresponding author.


†These authors contributed equally.

✉ afa@informatik.uni-kiel.de (A. Anjum); stu229910@mail.uni-kiel.de (N. Lieberum)

🌐 <https://www.zbw.eu/de/forschung/information-profiling-and-retrieval/profil-aftab-anjum> (A. Anjum)



© 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)

How wordplay adds humor, wit, and a layer of complexity to discourse, contributing to the pragmatic force of communication. Let's discuss a few of its aspects.

1. Double entendre: A form of wordplay that involves a phrase or expression with two different meanings, one usually more innocent or literal and the other often risqué or humorous. For example, "Time flies like an arrow; fruit flies like a banana." Here, the word "flies" is used in two different senses, creating a humorous effect.
2. A classic form of wordplay that relies on exploiting multiple meanings of a word or using words that sound similar but have different meanings. For example, "I used to be a baker, but I couldn't make enough dough." The pun on "dough" plays with its literal meaning as a baking ingredient and its slang meaning as money.
3. Irony: A form of wordplay that involves using words to convey a meaning that is the opposite of their literal sense. Irony often relies on context and the speaker's tone of voice. For example, when it's raining heavily outside, someone might say, "What lovely weather we're having!" The contradictory statement adds a humorous or sarcastic twist to the conversation.
4. Spoonerism: A type of wordplay that involves switching the initial sounds or letters of words to create a humorous effect. For example, "You have hissed all my mystery lectures and were caught fighting a liar in the quad." The words "missed" and "history lectures" are playfully interchanged in this sentence.

Initially, this paper will provide an overview of the fundamental concepts and significance of pun detection, pun location, interpretation, and pun translation in the realm of computational linguistic research.

Pun detection [2] involves the automatic identification of puns within text or speech. It is a fundamental task that serves as a foundation for subsequent analyses. Detecting puns is crucial for various applications, including natural language processing, sentiment analysis, and information retrieval. Accurate pun detection can enhance computational models' understanding of language and improve the performance of related tasks.

Pun location [3] aims to identify the specific words or phrases in a sentence that contribute to the creation of a pun. This task requires a deep understanding of language and context.

Pun translation [4] is the process of preserving the humor and wordplay in puns when translating them across different languages. This task is particularly challenging due to cultural and linguistic differences. Accurate pun translation not only facilitates cross-cultural communication but also contributes to machine translation systems' overall effectiveness and naturalness.

This paper aims to provide a comprehensive review of the importance of pun detection, pun location, interpretation, and pun translation in research. We will examine existing techniques, methodologies, and datasets utilized in these areas, highlighting their challenges and potential applications. The objective of the JOKER workshop is to foster collaboration among translators, linguists, and computer scientists to develop an evaluation framework for creative language. The pilot tasks encompass different objectives: Pilot Task 1 involves classifying single words containing wordplay based on a given typology and providing lexical-semantic interpretations. Pilot Task 2 focuses on translating single words that incorporate wordplay. Pilot Task 3 centers

around the translation of complete phrases that either encompass or include wordplay. Initially, the translation tasks will be targeted at English and French, but the inclusion of additional languages will be considered as more data becomes available.

For the Blended Intensive Program (BIP) Artificial Intelligence (AI) for Humanities in term of application: From Text Simplification to Automatic Humor Analysis, we apply a variety of machine learning and deep learning models to the tasks of pun detection, location, and translation. We commence our analysis by conducting essential Exploratory Data Analysis (EDA) on the provided dataset. Based on the insights gained from EDA, we make informed decisions throughout the stages of model selection and implementation. Our initial approach involves employing straightforward machine learning models, namely Naive Bayes (NB), Random Forest (RF), and TF-IDF Ridge. To transform the textual data into vectorized form, we utilize encoding techniques such as Bag of Words (BoW) and Frequency-Inverse Document Frequency (TF-IDF).

In addition, we delve into the realm of more advanced techniques by exploring complex deep learning models and leveraging pre-trained language models. The models under consideration include AI21, ST5, Bloom, FastText, Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Bert. Each of these models brings unique characteristics and capabilities to the table, allowing for a broad and in-depth exploration of the problem space. Based on the experimental results obtained by training our models on the [dataset], we observe promising performance, valuable insights into model interpretability, and indications of which model types excel in multilingual datasets.

This paper's overall objective is to evaluate and compare the performance of these diverse models in handling puns. We aim to identify the strengths and weaknesses of each model in the context of pun detection, location, and translation, thereby contributing valuable insights to the ongoing development of sophisticated and culturally aware AI systems.

The rest of the paper is organized as follows: Section 2 focuses on brief explanation of model designing and their implementations on three different task pun detection, location, and translation Section 3 related experiment; include dataset attributes, models hyper parameters setting during model training and results comaparison. Finally, Section 4 concludes the paper with a summary of key findings and the overall significance of advancements in pun analysis.

2. Related Work

Until now, Automatic Humour Analysis has garnered substantial attention in research, with various scholars focusing on different facets of the subject matter. Numerous researchers have dedicated their efforts to exploring distinct aspects within the field of Humour Analysis.

The study conducted by Antonio and Davide [5] delves into the automatic recognition of humor in Italian texts, focusing on the analysis of ambiguity, particularly morphosyntactic and syntactic ambiguity. Similarly Julia and Lawrence [6] explore the computational recognition of jokes, specifically wordplay jokes, using statistical language recognition techniques. Their research draws on Raskin's theory of humor as a foundational framework. Georgina and Sajjad Kianbakht [7] put forward a novel model for translating humor in narrative texts, such as novels. Their approach adopts a multidisciplinary perspective that recognizes the interdependence of language and culture, employing the analytical framework of cultural conceptualizations.

Antonio Reyes [8] focuses on analyzing humor and irony in social media, particularly on Twitter, and proposes a model for their automatic recognition based on textual features. The results of the experiments are positive for humor and encouraging for irony.

Hannu Toivonen and Matti Järvisalo concentrate on modeling incongruity, a crucial element of humor, in joke understanding. They propose a computational model that incorporates incongruity theory to analyze and generate humorous jokes.

Another work by Danushka Bollegala and Mitsuru Ishizuka [8] delves into the task of humor recognition and presents a method for extracting humor anchors, which are significant words or phrases contributing to the humorous effect.

Significant research is currently underway in the field of automatic humor detection, with new findings emerging alongside advancements in language models, particularly pre-trained transformer-based models.

3. Utilization of Existing Models and Techniques

In this section, we present the methodology employed for three different tasks in this study. We outline the steps taken to collect and preprocess the dataset, followed by the feature extraction process. Additionally, we describe the selection and training of the machine learning model, along with the evaluation metrics used to assess its performance.

3.1. Machine Learning

3.1.1. Text Vectorization

Text vectorization is the process of converting textual data into numerical representations that can be understood by machine learning algorithms. There exist numerous methods for text vectorization, each encompassing unique approaches and distinct characteristics. In the subsequent section, we elucidate the vectorization techniques implemented throughout our model training procedure.

1. Bag-of-Words (BoW): BoW represents a document as a collection of words, disregarding grammar and word order. It creates a vocabulary of unique words and assigns a binary or frequency-based value to each word indicating its presence or occurrence in the document.
2. Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF calculates the importance of a word in a document by considering its frequency within the document (term frequency) and inversely weighing it by the frequency of the word across all documents (inverse document frequency). This technique assigns higher weights to words that are more specific to a particular document.
3. Word Embeddings: Word embeddings are dense vector representations that capture semantic meaning and contextual relationships between words. Popular algorithms for generating word embeddings include Word2Vec, GloVe, and FastText. These embeddings can be pre-trained on large corpora or learned specifically for the task at hand.
4. Transformer-based Models: Transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), have revolutionized text vectorization. They utilize attention mechanisms to

capture contextual information from the entire input sequence, enabling more nuanced representations of words and sentences.

3.1.2. Machine learning Models

In this section, I will provide concise descriptions of the models utilized for this particular task. Below, you will find an overview of the various models employed and their respective functionalities.

Random Forest (RF) The Random Forest model is an ensemble learning method that combines multiple decision trees to make predictions. It operates by constructing a multitude of decision trees and aggregating their outputs to determine the final prediction. This approach enhances the model's accuracy and reduces overfitting by utilizing the diversity of multiple trees.

Naive Bayes (NB) The Naive Bayes model is a simple yet powerful probabilistic classifier that utilizes Bayes' theorem to make predictions. It assumes independence between the features and calculates the probability of a class given the input data. Despite its assumption of feature independence, Naive Bayes often performs well and is computationally efficient for text classification tasks.

Ridge The Ridge model is a linear regression technique that incorporates regularization to prevent overfitting in the presence of multicollinearity. It adds a penalty term to the loss function, shrinking the coefficients towards zero while still allowing them to have non-zero values. This helps in controlling the complexity of the model and improving its generalization performance.

3.1.3. Deep Learning and Generative Models

Multilayer Perceptron (MLP) The Multilayer Perceptron (MLP) model is a type of artificial neural network that consists of multiple layers of interconnected nodes, or neurons. It is a feed-forward neural network where information flows in one direction, from the input layer through the hidden layers to the output layer. The MLP is capable of learning complex patterns and non-linear relationships, making it suitable for a wide range of classification and regression tasks.

Long short-term memory (LSTM) The LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) that is specifically designed to capture and retain long-term dependencies in sequential data. It addresses the vanishing gradient problem of traditional RNNs by incorporating memory cells and gating mechanisms. These components allow the LSTM to selectively remember and forget information over extended time periods, making it effective for tasks involving sequential data such as natural language processing and time series analysis.

Bidirectional Encoder Representations from Transformers (BERT) The BERT (Bidirectional Encoder Representations from Transformers) model is a state-of-the-art language representation model based on Transformer architecture. It leverages a bidirectional training approach, allowing it to capture contextual information from both preceding and succeeding words. BERT exhibits exceptional performance in a wide range of natural language processing

tasks, including sentence classification, named entity recognition, and question-answering, by effectively encoding and understanding the intricate nuances of language.

FastText The FastText classification model is a text classification algorithm that utilizes word embeddings and character n-grams to represent and classify text. It breaks down words into subword units and generates vector representations for each subword, enabling it to handle out-of-vocabulary words effectively. FastText is known for its efficiency and accuracy in handling large text datasets, making it suitable for various classification tasks such as sentiment analysis and topic categorization.

SimpleT5 SimpleT5 is a model built on top of PyTorch Lightning and Transformers. It allows users to quickly train their T5 models, including T5, mT5, and byT5 models, with only a few lines of code. The T5 models, which can be trained using SimpleT5, are versatile and can be used for a variety of natural language processing (NLP) tasks. These tasks include summarization, question answering (QA), question generation (QG), translation, text generation, and more [9].

AI21 Labs - Jurassic-2 Grande Instruct The J2-Grande-Instruct model is a variation of the Jurassic-2 series developed by AI21. It is an auto-regressive language model based on the Transformer architecture and designed with modifications for improved efficiency. The models diverge from their GPT-3 counterparts in several aspects, including vocabulary size and the depth/width ratio of the neural net [10]. This model is specifically trained to handle instructions-only prompts, also known as "zero-shot" prompts, without the need for examples or "few-shot" prompts. It aims to provide a natural way to interact with large language models and is designed to give users an idea of the optimal output for their task without needing any examples [11].

BLOOM (BigScience Large Open-science Open-access Multilingual Language Model) The BLOOM model is an autoregressive Large Language Model (LLM) that leverages a decoder-only transformer architecture, derived from Megatron-LM GPT-2. It underwent training on approximately 366 billion tokens between March and July 2022, utilizing 1.6 Terabytes of pre-processed text. This extensive dataset included 350 billion unique tokens, encompassing 46 natural languages and 13 programming languages, enabling BLOOM to grasp a wide range of linguistic and programming contexts [12].

4. Experiments

4.1. Dataset

Wordplay encompasses a diverse range of linguistic phenomena that cleverly manipulate or defy the standard rules of pronunciation, spelling, word formation, and meaning in a language. Our extensive collection consists of more than two thousand translated instances of wordplay sourced from various mediums such as video games and literature, primarily in English and French. Each example has undergone meticulous manual classification, categorizing it based on a comprehensive inventory of wordplay types and structures, and further annotated to identify its specific lexical-semantic or morphosemantic components.

As the foundation for our study we use a given annotated database [13]. The database comprises three key columns: id (e.g., en_6889), text (e.g., "Soft drink inventors saw a big popportunity."), and a prediction target that varies depending on the specific task. For pun

detection, the target is a binary indicator (yes/no) specifying whether a pun is present in the sentence. For pun location, the target is the specific word in the sentence that forms the pun. Finally, for pun translation, the target is the French translation of the sentence, providing a means of examining how well the models can carry the pun across languages.

Table 1

Task 1 and Task 2 dataset statistics

| Language | Task 1 | | Task 2 | |
|----------|--------|--------|--------|-------|
| | Train | Test | Train | Test |
| English | 5,292 | 3,183 | 2,315 | 1,205 |
| French | 3,999 | 12,873 | 2,000 | 4,655 |
| Spanish | 1,994 | 2,241 | 876 | 960 |

Table 2

Task 1 dataset statistics

| Language | Train | | Test | |
|----------|----------|----------|----------|----------|
| | Positive | Negative | Positive | Negative |
| English | 3,085 | 2,207 | 809 | 2,374 |
| French | 1,998 | 2,001 | 5,308 | 7,565 |
| Spanish | 855 | 1,139 | 952 | 1,289 |

The statistical characteristics of the data-sets used for Task 1 and Task 2 can be observed in Table 1, with the data statistics sourced from the provided [14]. Upon examining the table, it becomes evident that the datasets exhibit a significant imbalance in class distribution for each language category. Specifically, in Task 1, there is a notable disparity between the number of samples belonging to the negative class in comparison to the positive class.

The observed data imbalance is particularly pronounced for Task 1, where the quantity of samples assigned to the negative class significantly exceeds the number of samples representing the positive class. This discrepancy raises concerns regarding the potential bias and limitations that may arise during the modeling and evaluation process. Such an imbalance in class distribution can lead to challenges in accurately representing and predicting the minority class, potentially affecting the overall performance and reliability of the models.

4.2. Results Analysis

In this section, we present the findings and results obtained from our study. The training measurements derived from training the basic machine learning and NLP language models are summarized in Tables 3 and 5. These tables provide a comprehensive overview of the performance metrics, including accuracy, F1 score, recall, and precision, achieved during the training phase.

The training evaluation results indicate that the models have been effectively trained on the available training data and demonstrate strong performance on the validation dataset. The metrics suggest high accuracy, balanced F1 scores, and satisfactory recall and precision rates. This implies that the models have successfully learned the patterns and characteristics of the training data, yielding promising results during the assessment on the validation set.

However, the evaluation results on the test dataset reveal a different picture. The performance on the test set is found to be unsatisfactory. This can be attributed to the presence of a highly imbalanced dataset. The imbalance in the dataset poses a significant challenge to the model's ability to generalize well to unseen data and accurately predict the minority classes.

Thus, it becomes essential to address the issue of data imbalance effectively. Failing to address this concern makes it exceedingly difficult for the model to achieve satisfactory performance on unseen data. By employing techniques specifically designed to handle imbalanced datasets, such as oversampling, class weighting, or the utilization of specialized algorithms, we can enhance the model's performance and improve its ability to generalize to new instances. Addressing the data imbalance is crucial in ensuring the reliability and robustness of the model's predictions.

Table 3

Accuracy, Precision, Recall and F1-Score on the Training Data-set of Task 1 (1.1).

| Model | Precision | Recall | F1-Score | Accuracy |
|------------|-------------|-------------|-------------|-------------|
| Jurassic-2 | 0.51 | 0.07 | 0.14 | 0.41 |
| BLOOM | 0.58 | 0.05 | 0.01 | 0.41 |
| FastText | 0.72 | 0.84 | 0.78 | 0.72 |
| RF-TFIDF | 0.99 | 0.99 | 0.99 | 0.99 |
| ST5 | 0.74 | 0.92 | 0.86 | 0.77 |
| TFidfRidge | 0.87 | 0.97 | 0.92 | 0.90 |

Table 4

Accuracy, Precision, Recall and F1-Score on the Test Data-set Task1 (1.1).

| Model | Precision | Recall | F1-Score | Accuracy |
|------------|-------------|-------------|-------------|-------------|
| Jurassic-2 | 0.27 | 0.09 | 0.019 | 0.74 |
| BLOOM | 0.30 | 0.03 | 0.07 | 0.74 |
| FastText | 0.25 | 0.80 | 0.39 | 0.35 |
| RF-TFIDF | 0.25 | 0.83 | 0.39 | 0.34 |
| ST5 | 0.26 | 0.93 | 0.41 | 0.34 |
| TFidfRidge | 0.26 | 0.93 | 0.41 | 0.34 |

Table 5

Accuracy score on Train Data set of Task 1 (2.1).

| Model | Accuracy |
|-------|-------------|
| Ai21 | 0.42 |
| BLOOM | 0.36 |
| ST5 | 0.85 |

Table 6

Accuracy score on Test Data set of Task 1 (2.1).

| Model | Accuracy |
|-------|-------------|
| Ai21 | 0.43 |
| BLOOM | 0.46 |
| ST5 | 0.80 |

5. Conclusion

In conclusion, the JOKER project has made significant advancements in enhancing our understanding and processing of creative language, particularly in the domain of humor and wordplay. The utilization of the JOKER dataset, which encompasses a vast collection of translated examples from diverse sources, has yielded valuable insights into the nuances of various wordplay types and structures. Throughout this workshop, our implemented model has demonstrated comparable performance, shedding light on the predictive capabilities of models for different languages. Wordplay and puns rely heavily on the unique linguistic characteristics of a language, such as homophones, double entendre, and phonetic similarities. Some languages naturally lend themselves to wordplay due to their specific phonetic or lexical properties. Furthermore, it is important to acknowledge that different languages exhibit distinct humor styles and preferences.

References

- [1] S. Knospe, *A Cognitive Model for Bilingual Puns*, 2015, pp. 161–193. doi:10.1515/9783110406719-008.
- [2] T. Miller, C. F. Hempelmann, I. Gurevych, Semeval-2017 task 7: Detection and interpretation of english puns, in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 58–68.
- [3] Y. Zou, W. Lu, Joint detection and location of english puns, arXiv preprint arXiv:1909.00175 (2019).
- [4] J. Vandaele, Wordplay in translation, *Handbook of translation studies 2* (2011) 180–183.
- [5] A. Reyes, D. Buscaldi, P. Rosso, An analysis of the impact of ambiguity on automatic humour recognition, in: *Text, Speech and Dialogue: 12th International Conference, TSD 2009, Pilsen, Czech Republic, September 13-17, 2009. Proceedings 12*, Springer, 2009, pp. 162–169.
- [6] 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.
- [7] G. Heydon, S. Kianbakht, Applying cultural linguistics to translation studies: a new model for humour translation, *International Journal of Comparative Literature and Translation Studies* 8 (2020) 1–11.
- [8] A. Reyes, P. Rosso, D. Buscaldi, From humor recognition to irony detection: The figurative language of social media, *Data & Knowledge Engineering* 74 (2012) 1–12.
- [9] S. Roy, simplet5, <https://pypi.org/project/simplet5/>, 2022. Accessed: 2023-06-05.

- [10] O. Lieber, O. Sharir, B. Lenz, Y. Shoham, Jurassic-1: Technical Details and Evaluation, Technical Report, AI21 Labs, 2023. URL: https://uploads-ssl.webflow.com/60fd4503684b466578c0d307/61138924626a6981ee09caf6_jurassic_tech_paper.pdf.
- [11] AI21, Instruct models, 2023. URL: <https://docs.ai21.com/docs/instruct-models>, accessed: 2023-06-05.
- [12] B. Workshop, Bloom: A 176b-parameter open-access multilingual language model, 2023. [arXiv:2211.05100](https://arxiv.org/abs/2211.05100).
- [13] V. M. P. Preciado, G. Sidorov, C. P. Preciado, Assessing wordplay-pun classification from joker dataset with pretrained bert humorous models, in: CEUR Workshop Proceedings, volume 3180, CEUR-WS, 2022, pp. 1828–1833.
- [14] A.-G. B. Liana Ermakova, Tristan Miller, Overview of joker - clef-2023 track on automatic wordplay analysis, in: CLEF 2023, 2023.