CLEF 2024 Joker Task 1: Exploring Pun Detection Using The T5 Transformer Model

Notebook for the JOKER Lab at CLEF 2024

Arina Gepalova[1,†], Adrian-Gabriel Chifu[1,*,†] and Sébastien Fournier[1,*,†]

[1] *Laboratoire d'Informatique et des Systèmes(LIS) UMR 7020, Aix-Marseille Université, Université de Toulon, CNRS, LIS, France*

Abstract

The paper presents an exploration of pun detection using the T5 transformer model. It focuses on the first task of the JOKER Lab at CLEF2024 which involves retrieving short humorous texts from a document collection and identifying texts containing puns. The implemented approach includes query extension, tokenization, similarity scoring, and the application of a pre-trained model. The study found that the T5 model achieved a 75.2% accuracy in pun detection, indicating that while the model is effective, there is still room for improvement, particularly in addressing data imbalance and refining query processes.

Keywords

pun detection, T5 transformer, humor analysis

1. Introduction

Automatic humor analysis, especially detecting puns/wordplay, — is a popular topic in NLP tasks since puns add extra complexity to natural language understanding because of their dual meaning. This year, we experimented with the T5 transformer model, that was trained for various NLP tasks and has an architecture designed for efficient fine-tuning on a specific dataset. This paper focuses on the 1st task of JOKER Lab [1] — retrieving short humorous texts from a document collection, and describes a method for detecting puns using a pre-trained LLM. The main steps of the work include query extension, tokenization, similarity scoring, and the use of pre-trained models to identify texts with puns.

The remainder of this paper is structured as follows: Section 2 describes the most recent methods used for pun detection. Section 3 provides a detailed description of the approach including statistics about the data set, explains the query preprocessing, the document retrieval process and the step of detecting puns that uses the flan-T5-base model. Section 4 presents the results. Finally, Section 5 concludes the paper.

2. Related work

Pun detection has been studied for many years. Recent approaches include deep learning methods, such as the BiLSTM-CRF model[2], which uses a tagging scheme to jointly detect and locate puns, identifying structural constraints and contextual information; a BiLSTM-CRF model[3] that uses word embeddings and contextual clues to improve the accuracy of pun detection.

Pun detection with large language models (LLMs) has been explored in previous years in CLEF2023[4] using ChatGPT[5] and simplet5 model[6], showing promising results in understanding and identifying puns in texts.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[*] Corresponding author.

[†] These authors contributed equally.

gepalovaarina@gmail.com (A. Gepalova); adrian.chifu@univ-amu.fr (A. Chifu); sebastien.fournier@univ-amu.fr (S. Fournier)

3. Approach

3.1. Data description

The data includes a set of documents with short texts (61,104 documents in total) and queries (a single word, a short expression or a proper noun). The training data consists of 12 queries accompanied by the corresponding relevance judgments, where relevant documents are marked 0 or 1 to indicate the absence or the presence of a pun in the text, respectively.

The number of labeled documents obtained from the training data to train a model equals to 2,389. The training dataset is imbalanced - 562 documents with pun/wordplay and 1,827 documents without pun/wordplay.

3.2. Method description

The task was divided into several stages: working with queries, expanding queries with synonyms, finding the best method for tokenization of queries and documents, choosing a threshold for the similarity score, and working with a pre-trained model to filter texts with puns.

3.3. Documents retrieval

3.3.1. Queries extension

To find all documents relevant to a query it was extended with its synonyms and compared similarity not only with an initial query but also with it's synonyms. Synonyms found with the `WordNet` database were taken into consideration if the similarity score with the initial query was more than 0.2.

The extension up to three expressions was chosen because with a larger number, synonyms may be found that are not related in meaning to the original query. Additionally, for some queries, the selected tool still finds only one or two synonyms. If the original request is a proper noun, it was left in the original single version.

Some examples of extended queries include:

- qid_test_15: horse - [horse, sawhorse, gymnastic horse]
- qid_test_52: Tom - [Tom]
- qid_test_50: vein - [vein, mineral vein, venous blood vessel]

3.3.2. Tokenization

Two approaches were used for tokenizing queries and texts before searching for relevant documents:

- bert-base-uncased[1] [7]: embeddings were calculated for extended queries and document texts. Cosine similarity was used as the similarity score.
- all-MiniLM-L6-v2[2]: this model returns embeddings in a suitable format: for each synonym, its vector was calculated separately. As a result, a query of dimension *(n, 384)* was obtained, where *n* - is the number of synonyms obtained in the previous step. Text of a document was processed in the initial format and a vector of dimension *(1, 384)* was obtained. The similarity was calculated for each synonym, and the maximum value found was taken as the final similarity score.

The model with the highest accuracy, all-MiniLM-L6-v2, was chosen as the final model.

[1] https://huggingface.co/google-bert/bert-base-uncased

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 1
all-MiniLM and BERT for tokenization

Model	Accuracy
`all-MiniLM-L6-v2`	**0.98**
`bert-base-uncased`	0.91

3.3.3. Similarity score

The next step was to find the optimal threshold for the similarity score to include all relevant documents. Accuracy was calculated for the range (0.6, 0.95) with the step of 0.05. The results are shown in Figure 1. The final threshold = **0.35** was chosen based on calculations where the F1-score reached its highest value compared to other thresholds higher than 0.35

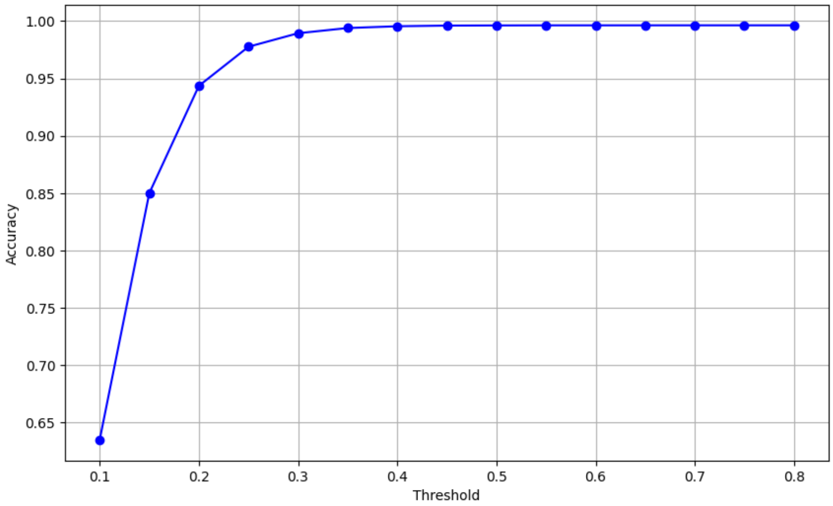

Figure 1: accuracy score for different thresholds of the similarity score

3.4. Wordplay detection

To identify texts containing puns/wordplay, the `flan-T5-base` model [8] was taken as the basis. Since 2,389 labeled documents could be obtained from the training data, the model was additionally trained on these data. 250 documents were excluded from training process to evaluate the performance on labeled data.

Prompt used to pass the model: “`Does the following text has pun/wordplay? {Text}` “

4. Results

The following metrics were calculated for 250 documents extracted from the labeled data:

- accuracy: 75.2%
- precision: 18.75%
- recall: 5.77%
- specificity: 93.43%

Additionally, detailed metrics for our approach are as follows:

- MAP (Mean Average Precision): 0.018
- NDCG (Normalized Discounted Cumulative Gain): 0.05

- Recall at 5: 0.032
- Recall at 10: 0.037
- Recall at 15: 0.043
- Recall at 20: 0.048
- Recall at 30: 0.052
- Recall at 100: 0.054
- Recall at 200: 0.054
- Recall at 500): 0.054
- Recall at 1000: 0.054
- Bpref (Binary Preference): 0.047
- Recip_rank (Reciprocal Rank): 0.13
- Precision at 1: 0.044
- Precision at 5: 0.058
- Precision at 10: 0.042

The imbalanced dataset leads to a higher number of false negatives, as the model is more likely to predict the majority class ('no wordplay'). It results in many missed actual positives, lowers the recall, and increases specificity.

The model was mostly confused in texts containing definitions. There are examples of texts that don't contain puns but were incorrectly identified by the model as containing puns:

1. "Warmhearted is having or showing a kind and generous attitude towards others."
2. "A gentleman (Old French: gentilz hom, gentle + man) is any man of good and courteous conduct."

The model showed better performance on identifying correct class (no pun/wordplay) on short texts of 1 sentence up to 15 words:

1. "Each taste bud contains 50 to 100 taste receptor cells."
2. "Saturninus takes her as his wife."

5. Conclusion

This paper explored the task of pun detection using the T5 transformer model. By implementing a series of steps including query extension, tokenization, similarity scoring, and leveraging pre-trained models, we achieved notable results in identifying puns within a large document collection.

The low precision and recall indicate that there are aspects to improve. Future work could focus on addressing the data imbalance, refining the query expansion and tokenization processes, and exploring additional training data.

Acknowledgments

We would like to thank the organizers of the JOKER Lab at CLEF 2024 for providing the opportunity to work on the task and for their support throughout the project.

References

[1] L. Ermakova, A.-G. Bosser, T. Miller, V. M. Palma Preciado, G. Sidorov, A. Jatowt, Overview of JOKER @ CLEF-2024: Automatic humour analysis, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Cham, 2024. To appear.

[2] Y. Zou, W. Lu, Joint detection and location of english puns, CoRR abs/1909.00175 (2019). URL: http://arxiv.org/abs/1909.00175. arXiv:1909.00175.

[3] L. Ren, B. Xu, H. Lin, L. Yang, Abml: attention-based multi-task learning for jointly humor recognition and pun detection, Soft Computing 25 (2021) 14109–14118.

[4] L. Ermakova, T. Miller, A.-G. Bosser, V. M. Palma Preciado, G. Sidorov, A. Jatowt, Overview of JOKER – CLEF-2023 track on automatic wordplay analysis, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer Nature Switzerland, Cham, 2023, pp. 397–415.

[5] Q. Dubreuil, UBO Team @ CLEF JOKER 2023 Track For Task 1, 2 and 3 -Applying AI Models In Regards To Pun Translation, 2023. URL: https://ceur-ws.org/Vol-3497/paper-155.pdf.

[6] Q. Dubreuil, UBO Team @ CLEF JOKER 2023 Track For Task 1, 2 and 3 -Applying AI Models In Regards To Pun Translation, 2023. URL: https://ceur-ws.org/Vol-3497/paper-155.pdf.

[7] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[8] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling Instruction-Finetuned Language Models, 2022. URL: https://arxiv.org/abs/2210.11416. doi:10.48550/ARXIV.2210.11416.