# University of Amsterdam at the CLEF 2024 Joker Track

Emma Schuurman, Mick Cazemier, Luc Buijs and Jaap Kamps

*University of Amsterdam, Amsterdam, The Netherlands*

**Abstract**

This paper reports on the University of Amsterdam's participation in the CLEF 2024 Joker track. Our overall goal is to investigate non-literal use of language, such as in humor and wordplay, that are still challenging current information retrieval and natural language processing technology. Our specific focus is to investigate how an effective wordplay detector can be used for the humorous search results or candidate translations, within the context of the track's humor retrieval, classification, and translation tasks.

Our main findings are the following. First, standard ranking approaches are effective for retrieving relevant sentences given a query, but a pun classification filter is effective to select humorous results. Second, a BERT encoder based classifier obtains reasonable performance in classifying different aspects of humor, with some distinctions being hard for both models and humans. Third, sequence to sequences machine translation models provide high quality descriptive translation, yet preserving the wordplay across languages remains challenging. More generally, we revisited the CLEF 2023 Joker Track's Pun Detection task, and were able to build effective neural pun classifiers. The value of these classifiers was demonstrated as a filter on the results of a standard ranker for the Humor-aware IR task of the CLEF 2024 Joker Track.

**Keywords**

Information Storage and Retrieval, Natural Language Processing, Wordplay translation, Humor retrieval, Humor classification

## 1. Introduction

The CLEF 2024 Joker track investigates possible solutions to the challenges of automated analysis and processing of humor. The Joker track series aims to advance the development of interpretation, generation and translation of wordplay, by bringing together computer scientists, linguists and translators. The CLEF 2024 Joker Track builds upon the findings from last year's edition. The CLEF 2023 Joker track results have shown that wordplay detection, localization and translation remain a challenge for state of the art systems. The CLEF 2024 Joker track reuses the corpus previously used for pun detection, and creates a new task on humor-aware information retrieval. The CLEF 2024 Joker track also introduces a entirely new task on humor classification, and continues the important pun translation task.

Our main approach also builds on the CLEF 2023 Joker Track, as we revisit last year's task on pun detection. We conduct an extensive analysis of the three tasks of the track: Task 1 on *Humor-aware Information Retrieval*; Task 2 on *Humor Classification*; and Task 3 on *Pun Translation*. For details on the exact track setup, we refer to the Track Overview paper CLEF 2024 LNCS proceedings [1], as well as the detailed task overviews in the CEUR proceedings [2, 3, 4].

Our main aim is to investigate how an effective wordplay detector can be used for the humorous search results or candidate translations, within the context of the track's humor retrieval, classification, and translation tasks. Specifically, our idea is to build an effective pun detector, and use this to filter out those results that are wordplay or puns. For example, as discussed in detail below, we can use a pun detector to filter wordplay from the results of a standard search engine focusing on topical relevance only. But in the same way it could be used to filter out the humorous text from the negatives in the Humor Classification corpus of Task 2. And we can have standard machine translation systems generate sets of different translations, and detect which of these preserve the wordplay.

The rest of this paper is structured as follows. Next, in Section 2 we discuss our experimental setup and the specific runs submitted. Section 3 discusses the results of our runs. Section 4 provides a detailed

*CLEF 2024: Conference and Labs of the Evaluation Forum, September 9–12, 2024, Grenoble, France*

kamps@uva.nl (J. Kamps)

0000-0002-6614-0087 (J. Kamps)

**Table 1**
CLEF 2024 Joker Track Submissions

| Task | Run | Description |
|---|---|---|
| 1 | UAms_Task1_Anserini_bm25 | BM25 baseline (Anserini, stemming) |
| 1 | UAms_Task1_Anserini_rm3 | RM3 baseline (Anserini, stemming) |
| 1 | UAms_Task1_bm25_CE100 | BM25 + Crossencoder top 100 |
| 1 | UAms_Task1_rm3_CE100 | BM25/RM3 + Crossencoder top 100 |
| 1 | UAms_Task1_bm25_BERT_Filter | BM25 + Filter on BERT WordPlay classifier (keeps 76%) |
| 1 | UAms_Task1_rm3_BERT_Filter | BM25/RM3 + Filter on BERT WordPlay classifier (keeps 46%) |
| 1 | UAms_Task1_rm3_T5_Filter1 | BM25/RM3 + Filter on WordPlay classifier (keeps 53%) |
| 1 | UAms_Task1_rm3_T5_Filter2 | BM25/RM3 + Filter on WordPlay classifier (keeps 43%) |
| 2 | UAms_Task2_BERT_ft | BERT classifier (fine-tuned) |
| 3 | UAms_Task3_Marian_ft | Marian Finetuned |
| 3 | UAms_Task3_T5-base_ft | T5-base Finetuned |

analysis of pun detection and topical relevance versus humor-aware IR for each task. We end in Section 5 by discussing our results and outlining the lessons learned.

## 2. Experimental Setup

In this section, we will detail our approach for the three CLEF 2024 Joker track tasks, as well as for the CLEF 2023 Joker track pun localization task.

For details of the exact task setup and results we refer the reader to the detailed overview of the track in [1] and [5]. The basic ingredients of the track are:

**Corpus** For Task 1, there is a large corpus of 61,268 documents (usually a single sentence each) for the retrieval task.

**Train Data** For Task 1, there are 12 train queries with relevance judgments (between 5 and 452 judgments per query, and between 4 and 281 relevant per query).

For Task 2, there are 1,742 sentences in the training set all labeled as either 'SC', 'EX', 'WS', 'SD', 'AID', 'IR', or 'WT'. These labels represent the type of humor that the sentence contains.

For Task 3, there are 1,405 English wordplays, with a total of 5,838 professional human French translations.

**Test Data** For Task 1, there are 57 test queries. These include the train queries, so there are a total of 45 unseen queries on which the test evaluation is based. For these unseen queries there is a total of 1,168 relevant documents, or an average of 26 per query.

For Task 2, there are 6,642 unlabeled sentences that contain one of the earlier described types of humor. In the final test evaluation set, there are 722 sentences with one of the labels on humor genre and technique.

For Task 3, there are 4,501 English wordplays. In the final test evaluation set there are 376 source sentences, and 834 human reference translations into French by professional translators.

We created runs for all the three tasks of the 2024 track and the localization task of the 2023 track, which we will discuss in order.

### 2.1. Task 1: Humor-aware Information Retrieval

This task asks to retrieve short humorous texts for a query. We submitted eight runs in total, shown in Table 1.

**Baseline Rankers** We first submitted four baseline runs focusing on regular information retrieval effectiveness. Two are vanilla baseline runs on an Anserini index, using either BM25 or BM25+RM3 with default settings [6].[1] The other two runs are neural cross-encoder rerankings of these runs, based on zero-shot application of an MSMARCO trained ranker, reranking the top 100 of either the BM25 or the BM25+RM3 baseline run.[2] We submitted four runs aiming to take the pun detection of the results into account.

**SimpleT5** SimpleT5, built on top of Pytorch-lightning and Transformer, streamlines the training of T5 models for different NLP tasks [7]. T5 stands for Text-To-Text Transfer Transformer, which means that this model can take text as input and produce new text as output. This text-to-text structure makes it possible to apply the same model, decoding process and training procedure for different tasks like summarization, classification and translation [8].

For the detection task, we loaded a SimpleT5 model using its built-in 'from_pretrained' method with the model name "t5-small", specifying its size. To train this model, the data was preprocessed by first merging the train and qrels file on 'id', to create a single file with the columns 'text' and 'wordplay'. These two columns were then renamed to 'source_text' and 'target_text'. Additionally, we added the prefix 'Detect pun:' to each line since T5 models expect a task related prefix.

After preprocessing, we used train_test_split to split the data into a training and a test dataset with 90% of the data allocated for training and 10% for testing. We trained two versions of the SimpleT5 model: version 1 with a batch size of 6 and version 2 with a batch size of 8.

To test the model for inference, we loaded the best-trained model by selecting the one with the lowest validation loss. To evaluate this model's performance, we applied 'model.predict' on each sentence of the test dataset, and then compared the output to its actual label.

**BERT** The Bidirectional Encoder Representations from Transformers (BERT) model [9] is another NLP model based on the transformer architecture. This model has been widely used and over 150 studies have been done on the model [10] (as of 2020). One major advantage of the BERT model is that a pretrained model can be finetuned with just one additional output layer to create models for a variety of tasks.

The detection task meant that the "AutoModelForSequenceClassification" was loaded to load a pretrained BERT model for sequence classification. We did this using the "base-bert-uncased" model. The data was preprocessed in similarly to the T5 model, however the prefix wasn't added since that was unnecessary for the BERT model. We split the data using the same 90 % of the data as training data. Additionally, the "AutoTokenizer" function was used to tokenize the data using the tokenizer associated with the BERT model. Additionally the data was batched using "Datacollatorwithpadding."

To train the model for this specific dataset, we trained the model using Low-Rank Adaptation (LoRa) [11]. This approach greatly reduced the number of trainable parameters and the time needed to retrain the model by freezing the pre-trained model weights and injecting trainable rank decomposition matrices into selected layers of the Transformer architecture. Furthermore, the training parameters were refined through iterative adjustments and empirical evaluation, wherein different values were tested to assess their impact on performance.

To evaluate the performance, the model was evaluated on the test split of the dataset. The evaluation mostly focused on improving the F1 score, using accuracy as a secondary performance metric.

## 2.2. Task 2: Humor Classification

This task asks to classify text according to genre and technique. We submitted a single run, also shown in Table 1.

[1] https://github.com/castorini/pyserini

[2] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

**BERT** The same BERT model was retrained for Task 2 using the same data preprocessing as was used for Task 1. The only difference was that the number of classes was increased from two ("yes" and "no" for whether the sentence contained a pun) to seven for the different types of humor that a sentence could contain.

### 2.3. Task 3: Pun Translation

This task asks to translate puns from english to french. We submitted two runs shown in Table 1.

**MarianMT** MarianMT is a "sequence-to-sequence" (Seq2Seq) model based on the Marian framework. Marian, first introduced in 2017, is written entirely in C++, which supports faster training and translation [12, 13]. MarianMT provides pre-trained models, which are smaller than most other translation models, about 298 MB on disk, compared to other transformer-based translation models that exceed 1 GB [14, 15]. The size of MarianMT makes the model useful for fine-tuning on custom datasets for specific tasks.

For the translation task, the MarianMTModel and MarianTokenizer were loaded from the transformers library, using the model checkpoint "Helsinki-NLP/opus-mt-en-fr". Before training, the data was preprocessed by merging the input en qrels files on "id_en", to create a single csv file. The columns were renamed to "English" and "French", no prefix was needed. The preprocessed data was divided with train_test_split into a 90/10 split of respectively training and test data. After which the training data was further divided into training and validation sets using a 80/20 split. Additionally, the data was batched by setting the 'per_device_train_batch_size' parameter in the Seq2SeqTrainingArguments equal to 8. The evaluation is performed based on the validation set at the end of each epoch by computing the BLEU score as the evaluation metric.

**T5-base** As stated in previous paragraphs, a T5 model can be used for different NLP tasks. It is suitable for machine translation due to its ability to understand natural language and generate contextually relevant information [16]. The 'T5ForConditionalGeneration' and the model name 't5-base' were used to load the T5-base model for English to French translation.

The preprocessing of the data was done similarly to the preprocessing for the MarianMT model. The split of the test, validation and train set was also done in the same manner. The 'T5Tokenizer' was used to tokenize the data before training. Training was done with the same number of epochs, batch size and evaluation metric as used for MarianMT.

## 3. Experimental Results

In this section, we will present the results of our experiments, in three self-contained subsections following the CLEF 2024 Joker Track tasks.

### 3.1. Task 1: Humor-aware Information Retrieval

We discuss our results for Task 1, asking to retrieve short humorous texts for a query.

Table 2 shows the performance of the Task 1 submissions on the train data. We submitted four runs focusing purely on standard retrieval effectiveness. First, the two Anserini baselines using BM25 with or without RM3 query expansion perform also reasonable for the pun retrieval task. The RM3 models outperforms vanilla BM25 on almost all measures, and higher initial precision. This highlights the fact that the pun retrieval task still requires puns to be "relevant" to the topic, and hence that focusing purely on topical relevance provides a reasonable baseline approach. Second, the zero-shot reranking with a cross-encoder does not lead to an improvement of retrieval effectiveness. However, some of these runs have high fractions of unjudged documents in the top of the ranking, up to 50% of the top 10 results. This is a call to caution for interpreting these scores as reliable performance estimations. However, we

**Table 2**
Evaluation of Joker Task 1 (train data).

| Run | MRR | Precision | | | NDCG | | | Bpref | MAP |
|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 10 | 20 | 5 | 10 | 20 | | |
| UAms_Task1_Anserini_bm25 | 0.1906 | 0.1167 | 0.1583 | 0.1361 | 0.1008 | 0.1598 | 0.2272 | 0.2376 | 0.1582 |
| UAms_Task1_Anserini_rm3 | 0.2407 | 0.1667 | 0.1750 | 0.1250 | 0.1506 | 0.1896 | 0.2339 | 0.2989 | 0.1725 |
| UAms_Task1_bm25_CE100 | 0.1233 | 0.0833 | 0.0750 | 0.0889 | 0.0685 | 0.0682 | 0.1300 | 0.0922 | 0.0702 |
| UAms_Task1_rm3_CE100 | 0.1231 | 0.0833 | 0.0917 | 0.1028 | 0.0685 | 0.0801 | 0.1422 | 0.0921 | 0.0712 |
| UAms_Task1_bm25_BERT_Filter | 0.2217 | 0.1167 | 0.1750 | 0.1639 | 0.1077 | 0.1848 | 0.2830 | 0.2721 | 0.1894 |
| UAms_Task1_rm3_BERT_Filter | 0.3679 | 0.2333 | 0.2333 | 0.1722 | 0.2254 | 0.2682 | 0.3312 | 0.3649 | 0.2295 |
| UAms_Task1_rm3_T5_Filter1 | 0.3813 | 0.2500 | 0.2750 | 0.1667 | 0.2480 | 0.3057 | 0.3119 | 0.2899 | 0.2138 |
| UAms_Task1_rm3_T5_Filter2 | 0.3373 | 0.2833 | 0.2833 | 0.1917 | 0.2551 | 0.3095 | 0.3367 | 0.3464 | 0.2326 |

**Table 3**
Evaluation of Joker Task 1 (test data).

| Run | MRR | Precision | | Recall | | | NDCG | Bpref | MAP |
|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 10 | 5 | 10 | 20 | | | |
| UAms_Task1_Anserini_bm25 | 0.1873 | 0.0489 | 0.0556 | 0.0564 | 0.0819 | 0.1624 | 0.2417 | 0.0928 | 0.0800 |
| UAms_Task1_Anserini_rm3 | 0.1977 | 0.0578 | 0.0622 | 0.0611 | 0.0830 | 0.1511 | 0.2677 | 0.0921 | 0.0845 |
| UAms_Task1_bm25_CE100 | 0.0762 | 0.0356 | 0.0267 | 0.0332 | 0.0388 | 0.0964 | 0.1749 | 0.0610 | 0.0416 |
| UAms_Task1_rm3_CE100 | 0.0749 | 0.0356 | 0.0267 | 0.0332 | 0.0388 | 0.0967 | 0.1769 | 0.0602 | 0.0410 |
| UAms_Task1_bm25_BERT_Filter | 0.1883 | 0.0489 | 0.0844 | 0.0590 | 0.1165 | 0.1822 | 0.2430 | 0.1173 | 0.0878 |
| UAms_Task1_rm3_BERT_Filter | 0.2668 | 0.1111 | 0.1156 | 0.0882 | 0.1436 | 0.2079 | 0.2739 | 0.1608 | 0.1156 |
| UAms_Task1_rm3_T5_Filter1 | 0.2283 | 0.0933 | 0.1111 | 0.0861 | 0.1478 | 0.1943 | 0.2651 | 0.1628 | 0.1077 |
| UAms_Task1_rm3_T5_Filter2 | 0.2604 | 0.1067 | 0.1289 | 0.0882 | 0.1508 | 0.2261 | 0.2820 | 0.1841 | 0.1207 |

expect that the puns relevant for this task are judged, and hence the neural reranker is particularly attracting topically relevant non-pun passages. We will analyze this in more detail in Section 4.3 below.

We also submitted four runs post-processing the relevance-only rankings with different pun classifiers. First, the BERT pun classifier applied to the BM25 baseline does lead to slightly better results when compared to the base BM25 model. The model labels 76% of the passages as puns and thus 76 % is kept. This is the suspected reason for the small increase in performance. When applied to the RM3 baseline, the BERT filter causes a much larger increase in performance. Likely because for this model only 46% of the passages are labeled as puns. Second, we applied the two different versions of the SimpleT5 pun classifier on the RM3 baseline. Version 1 kept 53% of the passages and version 2 kept 43%. Comparing the results of the RM3 baseline without the pun classifier to those with the classifier shows a significant improvement in performance. The difference in performance between the Bert model applied on the RM3 baseline and the SimpleT5 model is marginal.

Table 3 shows the performance of the Task 1 submissions on the test data. None of our approaches was trained or informed by the train data, nor was pooling used to locate relevant documents (the recall base of the corpus is known to be complete). As a consequence, the results of the train and test data are comparable. First, we see again that the standard lexical ranking approaches perform reasonably, with a gain in performance when using blind feedback. We also see again that the neural rankers attract topically relevant, not non-humorous content. As a result, zero-shot rankers lead to a decrease in performance due to the large fraction of non-relevant documents. Second, the filter based on a pun classifier is again effective, leading to a notable increase in both precision and recall (MRR, NDCG, and MAP). We see again that the larger model of the T5 based pun classifier outperformsthe BERT based pun classifier.

**Table 4**
Evaluation of Joker Task 2: Train data (top) and test data (bottom).

| Run | Accuracy | Macro | | | Weighted | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 Score | Precision | Recall | F1 Score |
| UAms_Task2_BERT_ft | 0.6561 | 0.6286 | 0.6090 | 0.5672 | 0.6752 | 0.6561 | 0.6254 |
| UAms_Task2_BERT_ft | 0.6330 | 0.5724 | 0.5845 | 0.5221 | 0.6605 | 0.6330 | 0.6021 |

**Table 5**
CLEF 2024 Joker Task 3: Results on train (top) and test (bottom)

| Run | n | BLEU | Precisions | | | | Length | | BERTScore | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | Rat. | Tok. | n | P | R | F1 |
| *Reference (train)* | 1,405 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 18,592 | 5,838 | 1.0000 | 1.0000 | 1.0000 |
| UAms_Task3_Marian_ft | 1,405 | 0.6856 | 0.7750 | 0.7009 | 0.6584 | 0.6179 | 1.2743 | 23,692 | 5,838 | 0.8182 | 0.8284 | 0.8228 |
| UAms_Task3_T5-base_ft | 1,405 | 0.6056 | 0.7766 | 0.6335 | 0.5550 | 0.4925 | 1.0814 | 20,106 | 5,838 | 0.8435 | 0.8333 | 0.8380 |
| *Reference (test)* | 376 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 5,774 | 834 | 1.0000 | 1.0000 | 1.0000 |
| UAms_Task3_Marian_ft | 376 | 0.2395 | 0.4380 | 0.2620 | 0.1934 | 0.1483 | 1.1751 | 6,785 | 834 | 0.7952 | 0.8080 | 0.8011 |
| UAms_Task3_T5-base_ft | 376 | 0.4415 | 0.6566 | 0.4924 | 0.4091 | 0.3462 | 0.9553 | 5,516 | 834 | 0.8713 | 0.8616 | 0.8662 |

## 3.2. Task 2: Humor Classification

We continue with Task 2, asking to classify text according to genre and technique. We submitted one run using the BERT model. This model performed reasonably well on the 10% hold out part of the train dataset. However, we observed that this model still struggles with recognizing certain minority classes which leads to it predicting majority classes much more often than would be desired.

We only submitted a single run based on a simple BERT classifier trained on 90% of the released train data. Table 4 shows the performance of the Task 2 submission on the train data (top half) and the test data (bottom half). First, it is reassuring to observe slightly lower but similar performance for the test data than on the train data. Second, the performance is not very high but still reasonable given that this is a multi-class text classification problem with 7 (or 6) possible labels. Third, manual inspection of the data also suggests the classification task is non-trivial, also for humans.

The aim is to automatically classify text according to the following classes: incongruity-absurdity (AID), exaggeration (EX), irony (IR), sarcasm (SC), self-deprecating (SD), and wit-surprise (WS). Inspection of the confusion matrix (not shown) reveals a reasonable diagonal, in particular for the classes with the largest support in the train data (in particular "WS"). The distribution of the test data differs, with "AID" being the largest class, explaining the small drop in performance.

Our model systematically miss-classifies sentences labeled as "irony" with "sarcasm" and "exaggeration." Several of these examples seem to contain elements of irony (typically about a situation and an opposite expectation) and of sarcasm (a form of expression, assuming the utterance appeared in some conversational context), or elements of exaggeration in some sense.

## 3.3. Task 3: Pun Translation

We continue with Task 3, asking to translate puns from english to french. Our experiments are based on MarianMT and T5-base models, both focusing on general machine translation quality for English to French.

Table 5 shows the results of the CLEF 2024 Joker track's Task 3, both on the train data (top) and the test data (bottom). We make a number of observations. First, the general translation quality is high, with BLEU scores ranging from 44% (test) to 69% (train) and BERTScore F1 ranging from 83% (train) to 87% (test). Second, the larger T5-base MT model performs better than the MarianMT model,

**Table 6**
CLEF 2024 Joker Task 3: Example (id_en: en_1007)

| Run | Text |
|---|---|
| *Source* | Save the whales, spouted Tom. |
| *Reference(s)* | "Il faut sauver les baleines," jeta Tom avant de se tasser.<br>"Il faut sauver les baleines," interjeta Tom.<br>Moi je sauve les baleines, Tom s'en venta.<br>Louis évent-a le projet de sauvetage des baleines.<br>"Sauvez les baleines," proclama Tom à tout évent.<br>"Sauvez les baleines, cracha Toto, Cétacé!" |
| UAms_Task3_Marian_ft | "Sauvez les baleines," proclama Tom à tout évent. |
| UAms_Task3_T5-base_ft | "Sauvez les baleines," dit Tom. |

in particular on the test set. Both models were fine-tuned on some of the train data, with precautions against overfitting such as hold-out test and validation subsets, but the performance of the T5-base model generalizes better. Third, the performance on the test data is lower than on the train data. This may signal some degree of overfitting in training, but this is not clearly evident in manual inspection of the output. One important factor affecting the absolute scores is the number of reference translations, which is significantly higher for the train data (4.2 per pun) than on the test data (2.2 per pun).

These automatic evaluation measures reflect the whole translated sentence and are a necessary but not sufficient condition for correctly translating the wordplay. The ground truth consists of professional translations preserving the wordplay across languages, making the results indicative and encouraging for pun translation, but also suggest the value of further qualitative analysis of the output.

Table 6 shows an example from the train data set. The top half of the table shows the English pun, and the six French translations made by professional translators. We make a number of observations. First, there is notable variation in the different translations, highlighting the complexity and creative element required. This also highlights the value of obtaining multiple translations from different professional translators. Second, many of the common non-pun words are shared between the different translations, which can lead to overemphasizing these in the MT evaluation measures that run on singular references like BERTscore. Measures that can naturally deal with multiple references may be preferable, and motivate the use of classic BLEU.

The bottom half of the table shows the generated translations. We make a number of observations again. First, the overall quality of the machine translations is very impressive, both in this example throughout the entire output. There are no fluency or other issues, and the output captures the literal content of the English pun adequately for understanding the topic and meaning. Second, some of the generated translations capture the literal content of the source, but not preserve the wordplay. For example, the T5-base output in this case is *"Save the whales," said Tom*, which is factually correct but not a wordplay. Third, some of the generated translations do both preserve the content and the wordplay. For example, the MarianMT output in this case is exactly matching one of the human professional translations, which creatively uses a similar wordplay with the meaning of *évent* referring to both a whale's blowhole, and to "in any case."

Our analysis revealed both the quality of current machine translation, and well as the complexity of preserving the wordplay in a literally correct translation. We observed also that the models are able to generate creative translations preserving the wordplay, but that the most likely translation or the first one generated by the model may not be a pun. This observation supports our general idea to generate multiple translations with the model, and use an effective pun detector to choose one of these translations in case it is likely preserving the wordplay. We are currently running experiments on using beam search to generate a diverse set of translations, and use a French pun detector to select the most promising candidate. Preliminary results demonstrate the viability of this approach.

**Table 7**
Evaluation of the CLEF 2023 Joker Pun Detection Task (English)

| Model | F1 Score | Precision | Recall | Accuracy |
|---|---|---|---|---|
| BERT | 0.70 | – | – | 0.72 |
| SimpleT5_V1 | 0.80 | 0.72 | 0.90 | 0.76 |
| SimpleT5_V2 | 0.80 | 0.74 | 0.87 | 0.77 |

# 4. Analysis

In this section, we will present further analysis, including a direct evaluation of the used humor classifier for pun detection.

## 4.1. CLEF Joker 2023 Task 1: Pun Detection Revisited

We revisit the CLEF 2023 Joker Track, and in particular the Task 1 on pun detection[17]. As detailed above, our overall approach to the Joker Track tasks is based on exploiting a pun detector to select wordplay among candidate results. For example, in the humor retrieval setting, this would allow us to avoid topically relevant non-humorous content. Similarly, in the translation setting, this would allow us to select one of the possible translation candidates preserving the wordplay. In this section, we will evaluate the quality of the pun detector directly, rather than in an end to end evaluation of the other Joker tasks.

### 4.1.1. Approach

This is essentially a classification task, with a large set of sentences of which some are wordplay and others are linguistically similar sentences without humorous content. The dataset consisted of 5,293 English sentences, with 58% being positive examples of sentences containing a pun and 42% being negative examples. The goal was to develop a model that could tell if an English sentence contained a pun or not. We used a SimpleT5 and a Bert model as described in the experimental setup in Section 2.

For more details on this task, we refer to the Track Overview paper CLEF 2023 Joker Track Overview paper [5], and the detailed overview of this particular task [17].

### 4.1.2. Results

The performance of our pun detection models is shown in Table 7. Both models perform well in detecting puns in English sentences. Based on these metrics, the SimpleT5 model seems to perform slightly better in detecting puns. The results achieved are a significant improvement on those achieved in 2023, we suspect that this has to do with the fact that we took steps to avoid overfitting. We found that our models tended to achieve similar results to those achieved in 2023 when we did not enact safeguards against it.

The performance of the pun classifier is note-worthy as it allowed us to address several of the Joker track tasks: we used the classifier "as is" as a filter on the relevance-based retrieval results for Joker 2024 Task 1, and we are planning to filter the most promising of multiple generated translation for Joker 2024 Task 3.

There is an important difference between our evaluation (on a hold-out validation set) and the official results on a (not released) test set in the CLEF 2023 Joker Track's overview paper [17]. The best performing system in the track in 2023 scored an F1 of 53.61 % which is far lower than a majority class prediction on the test data. While not tested on the exact sentences, the performance of our pun classifier is encouraging, and the quality has been validated by the use of these pun classifiers to address the humorous information retrieval task of Joker 2024.

**Table 8**
CLEF 2023 Joker Pun Localization Task 2 (test data)

| Task | Model | Accuracy |
|---|---|---|
| English | RoBERTa-large | 0.89 |
| French | CamemBERT-large_filtered | 0.25 |
| | CamemBERT-base_filtered | 0.50 |
| | CamemBERT-base_train_filtered | 0.46 |
| | CamemBERT-base_unfiltered | 0.41 |
| | CamemBERT-base_filtered_single_word | 0.51 |

## 4.2. CLEF 2023 Joker Task 2: Pun Localization Revisited

We continue our quest to revisit the CLEF 2023 Joker Track, and also focus on the Task 2 on pun localization [18]. This task asks to localize the pun word within a sentence. We are interested in this task, as detecting the ambiguous pun word is essential for developing more effective pun detectors (discussed in Section 4.1). Moreover, detecting the pun location allows for giving special attention to this word in the pun translation models and approaches (discussed in Section 3.3). It also allows to focus the pun translation evaluation specifically to the matching source and reference pun words.

### 4.2.1. Approach

For the Pun Localization task, there are 2,315 English sentences, each with the pun word labeled in the sentence. The corpus used to train the French model consists of 2,001 French sentences labeled in the same way. We randomly sampled these datasets, and 80% was used as the train data, and the remaining 20 % was used as the test set. This meant that 463 English sentences and 420 French sentences were used to evaluate the models.

We use two primary models: RoBERTa-large for the English data and CamemBERT for the French data.

**RoBERTa-large** RoBERTa-large is a state-of-the-art transformer model based on the earlier-described BERT architecture. It is an extension of the RoBERTa-base model, featuring 24 layers and 335 million parameters [19]. The extensive pre-training of this model makes it suitable for pun localization, because it allows the model to understand complicated linguistic patterns, similarly to BERT.

For the localization task, the data was preprocessed by using the RoBERTa tokenizer from the transformers library to tokenize the text data. This ensured consistent input representations. Padding and truncation was applied to standardize the input length across all sequences. Token-level labels were assigned to identify the positions of pun words within the sentences. In cases where standard tokenization did not capture pun variations (such as capitalizations or apostrophes), alternative methods were applied to ensure coverage.

**CamemBERT** CamemBERT is a specialized transformer model designed specifically for French natural language understanding tasks [20]. Built upon the RoBERTa architecture, CamemBERT features 110 million parameters in its base configuration and 335 million parameters in its large configuration. The model was pre-trained on a large French corpus, this makes it particularly effective for the pun localization task in French sentences.

The preprocessing for the localization task was identical to the preprocessing used for the RoBERTa. However, since a part of the dataset could not be tokenized successfully, multiple runs were done using the CamemBERT-model. For the filtered runs, the sentences that could not be tokenized were excluded (110 sentences). Another run was done using data that only contained a single instance of wordplay per sentence (165 sentences excluded), similarly to the English dataset.

#### 4.2.2. Results

Table 8 shows the results for the pun localization task. We evaluated the performance of multiple CamemBERT models on the French dataset, as well as the RoBERTa-large model on the English dataset. The results reveal significant variability in accuracy across model configurations. RoBERTa-large demonstrated exceptional accuracy, achieving a score of 0.89. In contrast, CamemBERT-large struggles with a notably lower accuracy of 0.25. CamemBERT-base_filtered showed significant improvement, achieving an accuracy of 0.50. CamemBERT_base_train_filtered performed worse at 0.46, and CamemBERT-base_unfiltered manages to achieve only 0.41 accuracy. CamemBERT-base_filtered_single_word performs better at 0.51 but still fell well short of RoBERTa-large.

These findings highlight the sensitivity of CamemBERT models to preprocessing variations and dataset characteristics, particularly concerning unlabeled sentences and multiple instances of wordplay within a sentence. RoBERTa-large's superior performance suggests robustness in handling the English pun localization task. This might indicate simpler localization requirements or better model alignment for English puns.

Based on the findings from CLEF Joker 2023 Track [18], which established benchmark scores of 83.15% accuracy for English and 41.35% accuracy for French using different approaches, these models are an improvement over the current state-of-the-art.

This concludes our efforts to revisit the Joker 2023 tasks, and directly related these tasks to the Joker 2024 tasks. The pun translation task of Joker 2023 **??** was continued into Joker 2024 [4], and discussed already above in Section 3.3.

### 4.3. Task 1: Topical Relevance versus Wordplay Retrieval

In this section, we will investigate the humor-aware retrieval models of Section 3.1 in terms of their ability to retrieval topically relevant information.

#### 4.3.1. On-Topic versus Humorous

In our results on Task 1 (Humor-aware Information Retrieval) above, we observed a low performance for neural rerankers based on effective zero-shot cross-encoders. These models have shown highly effective zero-shot performance for passage retrieval in numerous domains, and we speculated that the loss of performance is due to the models attracting many topically relevant but non-humorous results.

The task 1 corpus is constructed in a particular way, with known relevant puns treated as relevant only. In order to make the task challenging, a high fraction of topically relevant but non-humorous text is added to the corpus. Plus there is additional non-relevant content in the larger corpus. The provided train qrels allow us to reconstruct this for the 12 train topics. Specifically, there is a total of 562 relevant and humorous results (on average 46.8 per query) and a total of 1,827 other topically relevant results (on average 152.3 per query, and combined 199.1 on average). So within all combined topically relevant content, only 23.5% are humorous text and a majority of 76.5% is non-humorous.

#### 4.3.2. Results

Table 9 (top half) shows the retrieval effectiveness of zero-shot cross-encoder rerankers for both the lexical baseline models BM25 and BM25+RM3. We again observe that the performance drops considerably, and decreases when reranking a larger set of results. This can be explained by our analysis above on the distribution of humorous texts within the set of topically relevant documents.

We changed the relevance assessments into a set graded judgments, where we reward both aspects. Specifically, we treat topically relevant documents as relevance level 1, and relevant humorous content as relevance level 2. Graded measures like NDCG will still prioritize humorous texts, but boolean measures will treat all topically relevant results in the same way. Table 9 (bottom half) shows the matching results when looking at all topically relevant content. We observe that the performance drop disappears, and that scores can even increase when reranking a larger set of results. As the topic set

**Table 9**
Joker Task 1: Finding Puns (top) or topical relevance (bottom) on train data

| Run | MRR | Precision | | | NDCG | | | Bpref | MAP |
|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 10 | 20 | 5 | 10 | 20 | | |
| UAms_Task1_Anserini_bm25 | 0.1906 | 0.1167 | 0.1583 | 0.1361 | 0.1008 | 0.1598 | 0.2272 | 0.2376 | 0.1582 |
| UAms_Task1_bm25_CE50 | 0.1248 | 0.0833 | 0.0750 | 0.1028 | 0.0697 | 0.0683 | 0.1498 | 0.1155 | 0.0668 |
| UAms_Task1_bm25_CE100 | 0.1233 | 0.0833 | 0.0750 | 0.0889 | 0.0685 | 0.0682 | 0.1300 | 0.0922 | 0.0702 |
| UAms_Task1_bm25_CE1000 | 0.1039 | 0.0833 | 0.0750 | 0.0806 | 0.0660 | 0.0666 | 0.1188 | 0.0687 | 0.0898 |
| UAms_Task1_Anserini_rm3 | 0.2407 | 0.1667 | 0.1750 | 0.1250 | 0.1506 | 0.1896 | 0.2339 | 0.2989 | 0.1725 |
| UAms_Task1_rm3_CE50 | 0.1259 | 0.1000 | 0.0833 | 0.1056 | 0.0806 | 0.0754 | 0.1582 | 0.1233 | 0.0662 |
| UAms_Task1_rm3_CE100 | 0.1231 | 0.0833 | 0.0917 | 0.1028 | 0.0685 | 0.0801 | 0.1422 | 0.0921 | 0.0712 |
| UAms_Task1_rm3_CE1000 | 0.1038 | 0.0833 | 0.0667 | 0.0833 | 0.0660 | 0.0618 | 0.1238 | 0.0837 | 0.0957 |
| UAms_Task1_Anserini_bm25 | 0.6597 | 0.5500 | 0.5333 | 0.5111 | 0.3182 | 0.3477 | 0.4125 | 0.6510 | 0.3503 |
| UAms_Task1_bm25_CE50 | 0.8917 | 0.5833 | 0.5167 | 0.5056 | 0.3453 | 0.3267 | 0.3976 | 0.2897 | 0.1622 |
| UAms_Task1_bm25_CE100 | 0.8056 | 0.5167 | 0.5000 | 0.4917 | 0.3048 | 0.3076 | 0.3757 | 0.3655 | 0.1959 |
| UAms_Task1_bm25_CE1000 | 1.0000 | 0.5500 | 0.5083 | 0.5083 | 0.3435 | 0.3312 | 0.3935 | 0.6510 | 0.3639 |
| UAms_Task1_Anserini_rm3 | 0.7282 | 0.5833 | 0.5250 | 0.4944 | 0.3686 | 0.3659 | 0.4105 | 0.6682 | 0.3528 |
| UAms_Task1_rm3_CE50 | 0.8917 | 0.6000 | 0.5167 | 0.4861 | 0.3562 | 0.3312 | 0.3930 | 0.2847 | 0.1590 |
| UAms_Task1_rm3_CE100 | 0.8056 | 0.5167 | 0.5167 | 0.5111 | 0.3048 | 0.3198 | 0.3907 | 0.3652 | 0.1972 |
| UAms_Task1_rm3_CE1000 | 1.0000 | 0.5500 | 0.5000 | 0.5056 | 0.3435 | 0.3262 | 0.3951 | 0.6682 | 0.3682 |

**Table 10**
Evaluation of the CLEF 2023 Joker Pun Detection Task (French) on 10% hold-out (top) and train/test data (bottom)

| Model | n | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| Dummy-Model | 399 | 0.49 | 0.47 | 0.52 | 0.50 |
| DistilBERT-base | 399 | 0.71 | 0.71 | 0.69 | 0.68 |
| DistilBERT-FT1 | 399 | 0.72 | 0.71 | 0.69 | 0.70 |
| DistilBERT-FT2 | 399 | 0.70 | 0.57 | 0.75 | 0.65 |
| DistilBERT (FT1) *train* | 3,999 | 0.9395 | 0.9475 | 0.9304 | 0.9389 |
| DistilBERT (FT1) *test* | 17,791 | 0.7518 | 0.7189 | 0.7009 | 0.7098 |

is relatively small with 12 queries, there is no clear pattern, and the main gain seems to be in early precision. This analysis confirms that zero-shot rerankers are effective in terms of topical relevance, but that dedicated models for humor-aware IR are needed in order to effectively retrieve humorous text. This also highlights the value of the pun detector based approach we proposed in this paper, and which led to clear improvements of retrieval effectiveness for humor-aware IR.

## 4.4. Task 3: Multiple Translation Candidates

In this section, we will investigate the pun translation models of Section 3.3 in terms of their ability to generate multiple candidate translation.

## 4.5. Filtering for Wordplay

In this section, we discuss further experiments based on 1) MarianMT to generate multiple, and different, candidate translations, 2) building an effective pun detector for French, and 3) using this wordplay detection to retrieve the most likely pun translation.

We construct a pun detector for French, following the CLEF 2023 Joker Pun Detection Task [17] discussed above in Section 4.1. We split the train data into 90% training and 10% hold-out test data. Table 10 (top) shows the performance on the small hold out test set. First, as this is a single boolean

**Table 11**
CLEF 2024 Joker Task 3: Filtering with wordplay detection (test).

| Run | n | BLEU | Precisions | | | | Length | | BERTScore | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | Rat. | Tok. | n | P | R | F1 |
| *Reference (test)* | 376 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 5,774 | 834 | 1.0000 | 1.0000 | 1.0000 |
| MarianMT (optimized) | 376 | 0.5100 | 0.7169 | 0.5480 | 0.4520 | 0.3810 | 1.042 | 6,015 | 834 | 0.8985 | 0.8965 | 0.8973 |
| MarianMT/Pun Detector | 376 | 0.4663 | 0.6902 | 0.5085 | 0.4061 | 0.3318 | 1.050 | 6,061 | 834 | 0.8853 | 0.8849 | 0.8849 |

prediction with a 50/50 balanced classes, a dummy classifier from `scikit learn` gives a random prediction with uniform probabilities, and scores indeed around 0.50. Second, DistilBERT-base scores notably better, and fine-tuning by hyperparameter optimization leads to further improvement on F1. We decided to continue with the "FT1" version optimizing precision, rather than the "FT2" version optimizing recall, as our ultimate French pun detection model. Table 10 (bottom) shows the performance of this pun detector over the entire train data, and the Joker 2023 test data. While the train performance is an overestimation, as 90% of this data is seen in training, the model did not seem to suffer from significant over-fitting in manual inspection. This is confirmed with the performance on the official test set, where we observe impressive performance on this complex task. To put this score in perspective, the highest performing score at the CLEF 2023 Joker Pun Detection Task was 0.6645 [17].

Our French pun classifier is trained to provide "pun" and "non-pun" class probabilities, and in the above we treated the predicted class with the highest probability as the Boolean pun prediction asked in the CLEF 2023 Joker Pun Detection Task. We made runs in which we have our translation model generate five candidate translations using beam search, and select the candidate translation with the highest pun classification probability directly.

### 4.6. Results

Table 11 shows the evaluation over the entire output. We make the following observations. First, MarianMT optimized to generate multiple translations using beam search over the generate output performs better than the base MarianMT finetuned on the train data (shown in Table 5 before). We initially observed very similar candidates, with either all or none containing wordplay. We took special effort to generate a sufficient diverse candidates, and increasing the likelihood that one of them satisfies our pun detector. The beam search seems favorable for the setting of the task, as entertaining multiple candidates ultimate leads to a better highest ranked candidate by the model itself. Second, the filtered run selecting the candidate translation with the highest probability to be a wordplay also outperforms the earlier standard finetuned MarianMT model. Third, when evaluating over the entire generated sequence, we see that the model without the explicit filter (returning the most likely translation according to the model) scores higher than the filtered output (returning the most likely wordplay according to the pun detector). This may be due, in part, to the evaluation over the entire prediction containing many non-pun words. As can be expected, our model indeed increases the number of estimated wordplays according to the used pun detector for French.

## 5. Discussion and Conclusions

This paper detailed the University of Amsterdam's participation in the CLEF 2024 Joker track. We conducted a range of experiments, for each of the three tasks of the track. For Task 1 on *Humor-aware Information Retrieval*, we observed that standard ranking approaches are effective for retrieving relevant sentences given a query, but a pun classification filter is effective to select humorous results. For Task 2 on *Humor Classification*, we submitted preliminary approaches based on a BERT encoder based classifier to obtain reasonable performance in classifying different aspects of humor, with some distinctions being hard for both models and humans. For Task 3 on *Pun Translation*, we experimented with sequence to

sequences machine translation models to provide high quality descriptive translation, yet preserving the wordplay across languages remains challenging. For the task on *Pun Localization*, we observed that while RoBERTa-large and CamemBERT-base models improved on the current state-of-the-art models in locating instances of wordplay within sentences, however, achieving robust localization across different languages remains a persistent challenge.

Our specific focus is to investigate how an effective wordplay detector can be used for the humorous search results or candidate translations, within the context of the track's humor retrieval, classification, and translation tasks. We revisited the CLEF 2023 Joker Track's Pun Detection task, and were able to build effective neural pun classifiers. The value of these classifiers was demonstrated as a filter on the results of a standard ranker for the Humor-aware IR task of the CLEF 2024 Joker Track.

## Acknowledgments

This research was conducted as part of the final research projects of the Bachelor in Artificial Intelligence at the University of Amsterdam, We thank the coordinator Dr. Sander van Splunter for his support and flexibility to work around the CLEF deadlines. We also thank the track and task organizers for their amazing service and effort in making realistic benchmarks for analyzing and processing humorous text available. Jaap Kamps is partly funded by the Netherlands Organization for Scientific Research (NWO CI # CISC.CC.016, NWO NWA # 1518.22.105), the University of Amsterdam (AI4FinTech program), and ICAI (AI for Open Government Lab). Views expressed in this paper are not necessarily shared or endorsed by those funding the research.

## References

[1] L. Ermakova, T. Miller, A. Bosser, V. M. Palma-Preciado, G. Sidorov, A. Jatowt, Overview of the CLEF 2024 JOKER track: Automatic humour analysis, in: L. Goeuriot, G. Q. Philippe Mulhem, 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, 2024.

[2] L. Ermakova, et al., Overview of the CLEF 2024 JOKER task 1: Humour-aware information retrieval, in: G. Faggioli, et al. (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[3] V. M. Palma-Preciado, et al., Overview of the CLEF 2024 JOKER task 2: Humour classification according to genre and technique, in: G. Faggioli, et al. (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[4] L. Ermakova, et al., Overview of the CLEF 2024 JOKER task 3: Translate puns from english to french, in: G. Faggioli, et al. (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[5] L. Ermakova, T. Miller, A. 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 - 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18-21, 2023, Proceedings, volume 14163 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 397–415. URL: https://doi.org/10.1007/978-3-031-42448-9_26. doi:10.1007/978-3-031-42448-9_26.

[6] J. Lin, X. Ma, S. Lin, J. Yang, R. Pradeep, R. F. Nogueira, Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations, in: F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, T. Sakai (Eds.), SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, ACM, 2021, pp. 2356–2362. URL: https://doi.org/10.1145/3404835.3463238. doi:10.1145/3404835.3463238.

[7] S. Roy, Simplet5 — train t5 models in just 3 lines of code, 2021. URL: https://github.com/Shivanandroy/simpleT5.
[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.
[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.
[10] A. Rogers, O. Kovaleva, A. Rumshisky, A primer in BERTology: What we know about how BERT works, Transactions of the Association for Computational Linguistics 8 (2020) 842–866. URL: https://aclanthology.org/2020.tacl-1.54. doi:10.1162/tacl_a_00349.
[11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. arXiv:2106.09685.
[12] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, A. Birch, Marian: Fast neural machine translation in C++, in: F. Liu, T. Solorio (Eds.), Proceedings of ACL 2018, System Demonstrations, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 116–121. URL: https://aclanthology.org/P18-4020. doi:10.18653/v1/P18-4020.
[13] Neptune.ai, Hugging face pre-trained models: Find the best, https://neptune.ai/blog/hugging-face-pre-trained-models-find-the-best, 2024. Accessed: 2024-06-04.
[14] HuggingFace, MarianMT documentation, https://huggingface.co/transformers/v3.5.1/model_doc/marian.html, 2024. Accessed: 2024-06-04.
[15] K. S. Kalyan, Pretrained language models for neural machine translation, https://medium.com/@kalyanks/pretrained-language-models-for-neural-machine-translation-b2cdd2b22e78, 2023. Accessed: 2024-06-04.
[16] H. Bartsch, Unlocking the power of T5: The versatile language model for text-to-text tasks, https://aggregata.de/en/blog/pretrained-transformer/t5/, 2024. Accessed: 2024-06-04.
[17] L. Ermakova, T. Miller, A. Bosser, V. M. Palma-Preciado, G. Sidorov, A. Jatowt, Overview of JOKER 2023 automatic wordplay analysis task 1 - pun detection, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1785–1803. URL: https://ceur-ws.org/Vol-3497/paper-149.pdf.
[18] L. Ermakova, T. Miller, A. Bosser, V. M. Palma-Preciado, G. Sidorov, A. Jatowt, Overview of JOKER 2023 automatic wordplay analysis task 2 - pun location and interpretation, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1804–1817. URL: https://ceur-ws.org/Vol-3497/paper-150.pdf.
[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.
[20] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, E. de la Clergerie, D. Seddah, B. Sagot, Camembert: a tasty french language model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020. URL: http://dx.doi.org/10.18653/v1/2020.acl-main.645. doi:10.18653/v1/2020.acl-main.645.