Computational Humor Detection on Novel Romanian Dataset

Mihalache Diana
mihalachediana15@gmail.com

Ursu Andrei
andreiursu234@gmail.com

Abstract

This paper proposes an in-depth study of a machine learning system and pretrained transformer models aimed at detecting the punchline and classifying the humorous sentences published in Romanian. A corpus of 2.400 manually annotated data was used, 50% of which contain humor and the rest of them are reflecting natural conversations. We then proceeded with conducting experiments: Dataset analysis, Syntax analysis, POS-tagging, followed by classical machine learning algorithms and transformer models (BERT). The best result is given by BERT models, especially XLM-RoBERTa, with an accuracy score of 98%.

1 Introduction

As stated in the Cambridge Dictionary, humor is defined by "the ability to be amused by something seen, heard, or thought about, sometimes causing you to smile or laugh". Despite this definition, we cannot explain in a universally accepted theory "what, why, how, when and to whom it is funny" (Taylor and Mazlack, 2004). Humor is representing a big part of the human experience, reflecting cultural nuances, social contexts, and cognitive processes. In addition, every known human civilization has also had at least some form of humor to make others laugh (Caron, 2002; Gervais and Wilson, 2005). Despite its ubiquitous presence in daily life, humor remains a complex phenomenon to define and understand, due to its subjective nature and diverse forms.

In recent years, the advancement of natural language processing (NLP) has sparked interest in the automated detection of humor. This task involves the identification and classification of humorous content within text, which is challenging due to the subtleties and variations of humor across different languages and cultures. Detecting humor requires an understanding of linguistic ambiguity, sarcasm, puns, and other rhetorical devices that convey humor. The ability to accurately detect humor has practical applications in several domains, including sentiment analysis, chatbots, content moderation, and entertainment. For example, enhancing chatbot interactions with humor can lead to more engaging and human-like conversations, while content-moderation systems can identify and filter potentially inappropriate or offensive humor in online platforms.

In this article, we explore several computational techniques that have been used for the detection and classification of humor, highlighting the challenges and limitations faced in the field. We introduce a new dataset and examine various models and linguistic features. Our aim is to provide a comprehensive overview of the current state of humor detection and suggest avenues for future research that could improve the effectiveness and accuracy of these systems.

2 Related Work

Recently, the field of humor and satire detection has become an area of interest, as researchers have explored various approaches to understand and automate this process. This NLP task is quite difficult because of its complexity and the wide variety of humor types that must be identified. However, most studies on this topic are conducted using English-language datasets. Given that there are very few Romanian language datasets for NLP tasks, we propose both the continuation of the ***SaRoCo: Detecting Satire in a Novel Romanian Corpus of News Articles*** article (Rogoz et al., 2021) and its extension with a newly created and manually annotated dataset, using it as a baseline. This article proposes a new dataset with 55,608 public news articles. Their results indicate that the machine-level accuracy for detecting satire in Romanian is relatively low, at under 73% on the test set, when compared to the human-level accuracy of 87%. The SaRoCo dataset was used also in ***Adversarial Capsule Net-***

works for Romanian Satire Detection and Sentiment Analysis (Echim et al., 2023) which proposes a novel approach for detecting satire and performing sentiment analysis in Romanian texts using Adversarial Capsule Networks (ACNs). The proposed framework surpasses existing methods for both tasks, achieving an accuracy of up to 99.08%, thereby demonstrating a significant improvement.

One of the first humor detection models used handcrafted humor features with simple models such as Naive Bayes and Support Vector Machines to separate jokes from other types of texts (Mihalcea and Strapparava, 2005). It achieved an impressive 97% accuracy in distinguishing jokes from news and 79% accuracy for jokes from a standard English corpus. However, the question arises as to whether the model is truly detecting humor or confusing the writing style and lexicon of jokes with their humorous qualities (Winters and Delobelle, 2020; West and Horvitz, 2019). Given that negative examples do not come from the same distribution or domain, these techniques likely learned that certain words appear more frequently in jokes than others (e.g., "bar" or "mother-in-law", "life", "kids"), but do not capture the nuances that make a text humorous (Winters, 2021). Also the article ***ColBERT: Using BERT Sentence Embedding for Humor Detection*** (Annamoradnejad, 2020) presents ColBERT, a method that leverages BERT sentence embeddings for detecting humor in text. The approach focuses on utilizing pretrained transformer models to better capture the nuances in humor, which can be difficult to detect using traditional machine learning techniques. The study provides a novel solution to humor detection, showing how BERT can be applied effectively for this task.

3 Methodology

We explore several computational techniques that have been used for the detection and classification of humor on a new dataset and examine various models and linguistic features that are presented in this chapter.

3.1 Dataset

The dataset contains 2.400 sentences, divided equally as humor sentences (marked with 1), which also have the punchline extracted, and non-humor sentences (marked with 0). The sentences that contain humor are extracted from stand-up specials, from both male and female comedians. The natural conversation sentences were extracted from Reddit. The data annotation was executed manually from both a male and a female perspective, so that the dataset reflects a balanced viewpoint and minimizes gender bias in the classification and analysis of humor. This dual-perspective approach aims to enhance the accuracy and fairness of the models trained on this data. We shuffled the data before splitting it into 3 datasets: 75% for the training phase of the model, 15% for the validation phase and 15% for the testing phase.

Dataset	Sentences
Training	1680
Validation	360
Testing	360

Table 1: Datasets

3.2 Dataset Analysis

To fully understand our dataset, we opted for an exploratory data analysis to ensure that the data in the dataset is independent and identically distributed. Thus, in the following figures, we explore the data structure using n-grams and POS tagging.

Label	Words	Characters
Humor	18.52	81.09
Punchline	5.53	12.83
Non Humor	15.71	75.13

Table 2: Average Lengths of the labels

Figures 1 and 2 present an analysis of the most frequent nouns found in two different classes of a humor dataset. The two classes represent non-humorous (Class 0) and humorous (Class 1) texts. The differences in vocabulary usage provide insights into the nature of humor in the dataset.

In the non-humorous class, the words suggest a focus on education, work, and general life experiences, indicating a more serious or neutral tone.

In the humorous class we notice a stronger presence of words related to people, relationships, and potentially humorous or taboo topics such as "sex" and "niciodata". This suggests that humor in this dataset is often derived from personal experiences, social interactions, and unexpected situations.

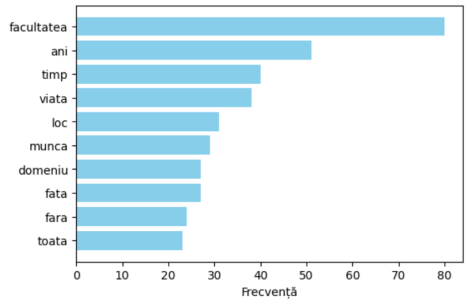

Figure 1: The first 10 most common NOUNS from the Non Humor Label

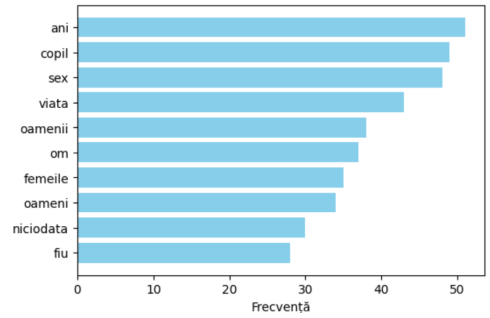

Figure 2: The first 10 most common NOUNS from the Humor Label

The tables 3 and 4 present the frequency of the most common bigrams and trigrams in the training data, highlighting the language patterns that frequently appear in humorous and non-humorous texts. As we can observe, non-humor texts contain structured, narrative-like phrases such as "am fost" (I was) and "am facut" (I did), indicating a more descriptive or expository style focused on past events. In contrast, humor texts feature more conversational and informal expressions like "iubita mea" (my girlfriend) and "fiu-miu" (my son), suggesting that humor often builds on direct speech, playful interactions, and exaggeration. Both categories use conditionals like "ar fi" (would be), but humor texts rely more on modal verbs ("ar putea" (could), "ar trebui" (should)) to introduce speculation or irony. Additionally, humor texts include rhetorical questions ("vrei auzi cand" (do you want to hear when?)), unexpected situations ("karaoke am zis"), and familial themes, making them more interactive and dynamic. Overall, non-humorous texts lean toward structured storytelling, while humorous ones favor dialogue, informality, and surprise elements.

Bigram	Frequency	Trigram	Frequency
am fost	34	fac viata mea	7
ar fi	29	am dat seama	5
am dat	26	cum as putea	5
am facut	26	ar trebui fac	4
dar am	26	am dat admiterea	4
au fost	25	cred ar fi	4
am avut	24	cate am inteles	3
as vrea	22	prima data cand	3
am vazut	19	primul semestru am	3
am intrat	19	900 lei luna	3

Table 3: Top 10 Bigrams and Trigrams for Non-Humor Texts

Bigram	Frequency	Trigram	Frequency
se numeste	32	cum ar fi	7
iubita mea	29	vrei auzi cand	6
ar fi	29	auzi cand esti	6
ar putea	25	ar trebui se	6
fiu miu	24	iubita mea zis	6
am luat	23	trebuiasca raspund lui	5
ar trebui	22	cum se numesc	5
am zis	20	lui fiu miu	5
au fost	18	karaoke am zis	5

Table 4: Top 10 Bigrams and Trigrams for Humor Texts

Figures 3 and 4 show the distribution of sentence lengths, measured by word count, for humorous and non-humorous texts. The first plot, for humorous texts, reveals a bimodal distribution. There is a peak around 10-12 words per sentence, as well as a larger peak around 100-120 words per sentence. This suggests that humorous writing often utilizes both shorter, punchier sentences as well as longer, more elaborate sentences. The second plot, for non-humorous texts, exhibits a more typical bell-shaped distribution centered around 20-25 words per sentence. This indicates that non-humorous writing tends to have a more consistent sentence structure, without the same variability seen in the humorous texts. The humorous writing leverages a wider range of sentence lengths compared to non-humorous writing, likely to create different rhythms and patterns that contribute to the comedic effect. The bimodal distribution for humorous texts is a notable difference from the more unimodal distribution for non-humorous texts.

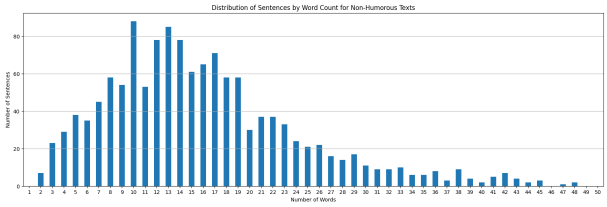

Figure 3: Non-Humor Word Count

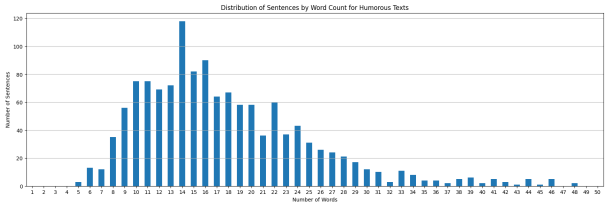

Figure 4: Humor Word Count

3.3 Preprocessing

For these, we preprocessed the texts using the TF-IDF vectorizer, and then used them to train the models. Their performance is evaluated on the validation and evaluation datasets, measuring accuracy using the `classification_report` function.

The preprocessing task for Romanian phrases involves two main steps: replacing special characters and removing prepositions and stop words. The first part is designed to handle the replacement of specific Romanian special characters with their non-accented equivalents. The replacements are defined as follows: ă is replaced with a, î and â are both replaced with i, ș is replaced with s, ț is replaced with t. The second part is used to remove both prepositions and stop words from the text, using a list with common stopwords in Romanian (Gene Diaz, 2016) and the Stanza package (Qi et al., 2020) that includes various functions such as tokenization and part-of-speech (POS) tagging. The stop words list includes common words such as "sau", "in", "pe", "la", "cu", etc., which are typically filtered out to reduce noise in the text data. By removing these elements, the preprocessing aims to retain only the most informative parts of the text, which are then used for further analysis or model training.

3.4 Supervised Methods

We used models as follows:

- **Logistic Regression** (Hosmer et al., 2013; James et al., 2013) is a statistical model used for binary classification, predicting the probability of an outcome that can only take one of two possible values.
- **Support Vector Machines (SVM)** (Cortes and Vapnik, 1995) is a supervised learning algorithm used for classification and regression tasks, which finds the optimal hyperplane that best separates data into different classes.
- **Naive Bayes** (Radford et al., 2021) is a probabilistic classification algorithm based on Bayes' theorem, assuming independence between features.
- **RoBERT** (Dumitrescu et al., 2020) is a Romanian language model based on the BERT architecture, designed for tasks like text classification, sentiment analysis, and named entity recognition in Romanian.
- **XLM-RoBERTa (XLM-R)** (Conneau et al., 2020) is a transformer-based multilingual language model that extends RoBERTa to support over 100 languages. It is designed for natural language understanding tasks across diverse languages, using a shared vocabulary and training on large-scale multilingual datasets.

Model	Test Accuracy
Logistic Regression	0.85
SVM	0.86
Naive Bayes	0.89
RoBERT	0.97
XLM-RoBERTa	0.98

Table 5: Test Accuracy Results

Their confusion matrices are listed in the appendix 12, 13, 14, 15, 16.

3.5 Unsupervised Methods

We used the K-means (Lloyd, 1982) for clustering and applied tokenization using RoBERT for the text. The K-means algorithm assigns each data point to the cluster with the nearest centroid, where the centroid is the mean of the data points in that cluster. The process iterates by updating the centroids until convergence, where the centroids no longer change significantly. Additionally, we compared the clustering results with the real labels, as it can be seen in the figures 5 and 6

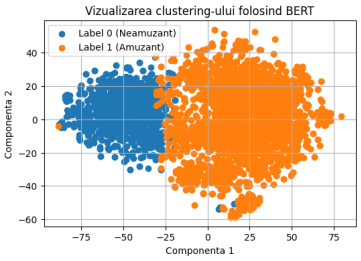

Figure 5: K-Means Clustering

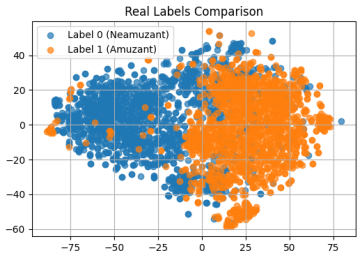

Figure 6: K-Means Clustering with real labels

4 Linguistics

We also chose to dig deeper from a linguistic perspective for the training dataset, using Dependency parsing and Constituency parsing. Parsing refers to the process of analyzing a sentence or text based on its grammatical structure. In NLP, parsing involves extracting the syntactic structure of a sentence, which helps in understanding relationships between words, their roles in a sentence, and how they interact to convey meaning. Parsing Romanian text comes with unique challenges due to its rich morphology, flexible word order, and complex syntactic rules.

4.1 Dependency Parsing

Dependency parsing is a syntactic analysis method where the structure is represented in terms of relationships between words. Each word in a sentence is connected to its head word through a directed edge, where each edge represents a grammatical relation. In dependency parsing, the main goal is to identify which word is the 'head' of each word in the sentence (e.g., subject, object, etc.). In figure 7 we can observ the frequencies of Dependency types on the humorous phrases.

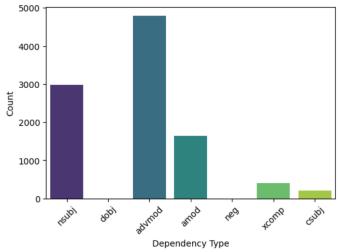

Figure 7: Distribution of Dependency types

The high frequency of adverbial modifiers (advmod) and nominal subjects (nsubj) suggests that humor often relies on modifiers (adverbs) and clear subject-verb structures. That indicates a presence of descriptive language, as it can be seen in the exemple from figure 8.

```
Sentence:
 Câinele meu răspunde la toate comenzile. Ăsta e jobul lui la Pizza Hut Delivery.
Dependencies:
Câinele (nsubj) --> răspunde
meu (det) --> Câinele
răspunde (ROOT) --> răspunde
la (case) --> comenzile
toate (det) --> comenzile
comenzile (obl) --> răspunde
. (punct) --> răspunde
Ăsta (nsubj) --> jobul
e (cop) --> jobul
jobul (ROOT) --> jobul
lui (det) --> jobul
la (case) --> Pizza
Pizza (nmod) --> jobul
Hut (flat) --> Pizza
Delivery (flat) --> Pizza
. (punct) --> jobul
```

Figure 8: Exemple for Dependency Parsing on Romanian

4.2 Phrase Structure Parsing (Constituency Parsing)

Phrase structure parsing is another approach that breaks down a sentence into subphrases or constituents (e.g., noun phrases, verb phrases). This method is based on the hierarchical structure of a sentence, where each part of the sentence corresponds to a larger grammatical unit, as it can be seen in figure 9.

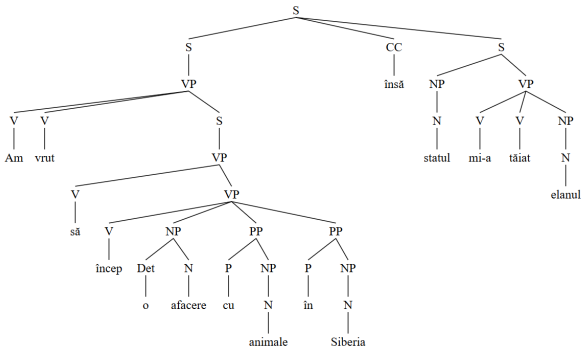

Figure 9: Exemple for Constituency Parsing on Romanian

As shown in the examples, both dependency parsing and constituency parsing can be used for humor detection. Humor often relies on unexpected sentence structures, word play, and relationships between words. By parsing Romanian sentences, we can identify constructions that deviate from the expected syntax and mark them as humorous. For example:

- **Dependency parsing** can reveal unusual relationships between subjects and objects, which suggest a surprise or twist—key elements in humor.
- **Phrase structure parsing** helps to identify non-standard structures that indicate humor, like unexpected noun phrases or verb phrases.

5 PunchLine Detection

As a final experiment, we attempted to see if a model can correctly detect where the punchline is located in the humorous text. For this, we will record the position of the first and last character of the punchline in relation to the entire text. The punchline is defined as the final and essential part of a joke that delivers the comedic effect or surprise. It is the moment when an unexpected twist, conclusion, or connection is delivered, provoking laughter. The punchline is often short, well-crafted, and strategically placed to maximize its humorous impact.

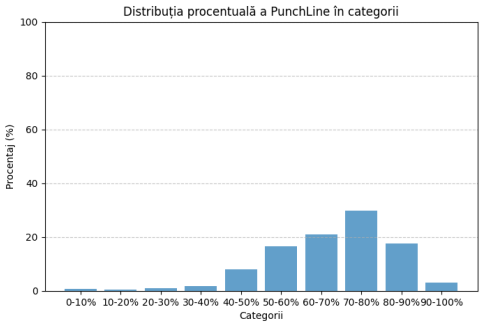

Figure 10: Punchline distribution in Humorous Texts

The data from figure 10 reveals a peak in punchline placement within the 70-80% range of jokes, with about 30% of punchlines occurring here. This aligns with comedy principles, where the setup typically occupies the first 70%, leading to the punchline near the end but not at the very end. Few punchlines appear in the first 40%, with a gradual increase peaking at 70-80%, followed by a sharp decline. This pattern supports the idea that effective joke structure involves sufficient setup without lingering too long after the punchline. We also concluded that the average punchline has **5.53** words and **12.83** characters.

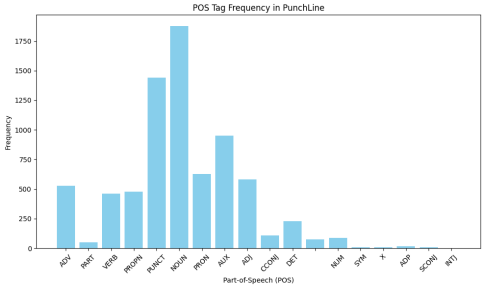

Figure 11: POS-tagging punchline

The graph 11 highlights the frequency of different POS tags in punchlines, revealing key patterns in joke structure. This distribution aligns with the comedic reliance on specific revelations and surprises, providing insights into joke construction and potential applications in computational humor generation.

- **NOUN** (aprox. 1900 occurrences) and **PROPN** (aprox. 1450 occurrences) dominate, indicating punchlines often center on specific objects, concepts, or names.
- **AUX** (aprox. 950 occurrences) and **PRON** show punchlines frequently involve actions or references.

Finally, trained the RoBERT model to see if it can correctly detect where the punchline is located in the humorous text. As metrics, we used Cosine Similarity and Levenshtein Distance. In the table 6 we can see the results we obtained for the detection of the punchline.

Model	RoBERT	RoBERT with transfer learning
Cosine Similarity	0.94	0.95
Levenshtein Distance	2.18	2.91

Table 6: Punchline Task Results

6 Future Work

We will continue expanding the dataset with more data and strive to diversify it as much as possible. Additionally, we will explore and test more robust methods for detecting the humorous part of each joke.

7 Conclusion

In conclusion, this research proposes a new dataset for detecting humor in medium and short-length sentences and serves as a valid starting point for analyzing and detecting humorous structures in Romanian texts.

Limitations

When it comes to the binary classification task, we noticed that the results were far beyond our expectations. In this context, we can still question how the models learn, specifically whether they truly understand humor or simply recognize certain specific linguistic structures. Although the obtained scores for punchline detection are quite good, we must mention that this model tends to learn more about the location of the punchline within the supporting texts. Thus, if we input a text with a punchline that is much longer than usual or placed in an unusual location, the model tends to identify as the punchline the groups of words located in the 70%-80% segment of the sentence's length.

8 Author contribution statement

We both shared the tasks for this paper as we considered equally. We both designed the research methodology, collected and preprocessed the dataset, ensuring data quality and consistency. Andrei conducted the majority of the experiments and performed the statistical analysis for both of the tasks. Diana trained the transformer models for the classification task, and wrote the majority of the manuscript, structured the results, and revised the paper for clarity. We both reviewed and approved the final version of the manuscript.

References

I. Annamoradnejad. 2020. Colbert: Using bert sentence embedding for humor detection. *arXiv*.

James E. Caron. 2002. From ethology to aesthetics: Evolution as a theoretical paradigm for research on laughter, humor, and other comic phenomena. *HUMOR*, 15(3):245–281.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of Romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4324–4328, Online. Association for Computational Linguistics.

Sebastian-Vasile Echim, Răzvan-Alexandru Smădu, Andrei-Marius Avram, Dumitru-Clementin Cercel, and Florin Pop. 2023. *Adversarial Capsule Networks for Romanian Satire Detection and Sentiment Analysis*, page 428–442. Springer Nature Switzerland.

Janimo Jani Monoses Gene Diaz. 2016. Stopwords for romanian (stopwords-ro).

Matthew Gervais and David Sloan Wilson. 2005. The evolution and functions of laughter and humor: A synthetic approach. *The Quarterly Review of Biology*, 80(4):395–430.

David W Hosmer, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied Logistic Regression*. John Wiley & Sons.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Springer.

Stuart P. Lloyd. 1982. A k-means clustering algorithm. *IEEE Transactions on Information Theory*, 28(2):129–136.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages.

A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, and I. Sutskever. 2021. Learning transferable visual models from natural language supervision. ArXiv preprint arXiv:2103.00020.

Ana-Cristina Rogoz, Mihaela Gaman, and Radu Tudor Ionescu. 2021. Saroco: Detecting satire in a novel romanian corpus of news articles. In *Proceedings of ACL*, page TBD.

J.M. Taylor and L.J. Mazlack. 2004. Computationally recognizing wordplay in jokes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.

Robert West and Eric Horvitz. 2019. Reverse-engineering satire, or "paper on computational humor accepted despite making serious advances".

Thomas Winters. 2021. Computers Learning Humor Is No Joke. *Harvard Data Science Review*, 3(2). Https://hdsr.mitpress.mit.edu/pub/wi9yky5c.

Thomas Winters and Pieter Delobelle. 2020. Dutch humor detection by generating negative examples.

A Appendix

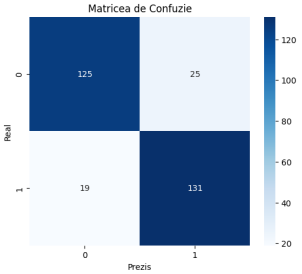

Figure 12: Logistic Regression Confusion Matrix

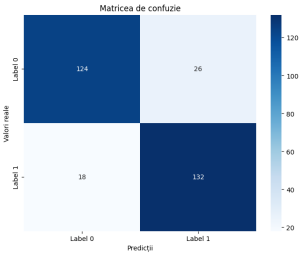

Figure 13: SVM Confusion Matrix

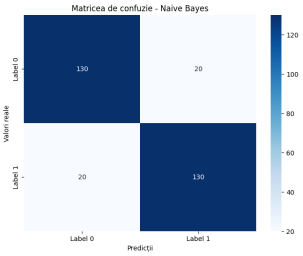

Figure 14: Naive Bayes Confusion Matrix

Figure 15: RoBERT Confusion Matrix

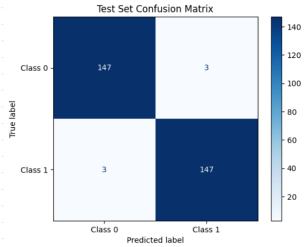

Figure 16: XLM-RoBERTa Confusion Matrix