Sarcasm Detection in a Less-Resourced Language

Lazar Đoković
lazardjokoviclaki02@gmail.com
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Marko Robnik-Šikonja
marko.robnik@fri.uni-lj.si
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Abstract

The sarcasm detection task in natural language processing tries to classify whether an utterance is sarcastic or not. It is related to sentiment analysis since it often inverts surface sentiment. Because sarcastic sentences are highly dependent on context, and they are often accompanied by various non-verbal cues, the task is challenging. Most of related work focuses on high-resourced languages like English. To build a sarcasm detection dataset for a less-resourced language, such as Slovenian, we leverage two modern techniques: a machine translation specific medium-size transformer model, and a very large generative language model. We explore the viability of translated datasets and how the size of a pretrained transformer affects its ability to detect sarcasm. We train ensembles of detection models and evaluate models' performance. The results show that larger models generally outperform smaller ones and that ensembling can slightly improve sarcasm detection performance. Our best ensemble approach achieves an F_1-score of 0.765 which is close to annotators' agreement in the source language.

Keywords

natural language processing, large language models, sarcasm detection, neural machine translation, BERT model, GPT model, LLaMa model, ensembles

1 Introduction

Sentiment analysis is a popular task in natural language processing (NLP), concerned with the extraction of underlying attitudes and opinions, usually categorized as "positive", "negative", and "neutral". Detection of sentiment is challenging if the utterances are ironic or sarcastic. *Sarcasm* is a form of verbal irony that transforms the surface polarity of an apparently positive or negative utterance/statement into its opposite [6]. Sarcasm is frequent in our day-to-day communication, especially on social media [5]. This poses a significant problem for sentiment analysis tools since sarcasm polarity switches create ambiguity in meaning. Sarcasm is highly dependent on its context. For example, the sentence "*I just love hot weather*" could be interpreted as sarcastic, depending on the situation, e.g., during summer heat waves.

Historical developments of sarcasm detection are surveyed by Joshi et al. [3], while recent developments are covered by Moores and Mago [5]. The problem of automatic sarcasm detection in text is most commonly formulated as a classification task. Unfortunately, sarcasm detection is affected by the lack of large-scale, noise-free datasets. Existing datasets are mostly harvested from microblogging platforms such as Twitter and Reddit, relying on user annotation via distant supervision through hashtags, such as *#sarcasm*, *#sarcastic*, *#not*, etc. This method is popular since 1) only the author of a post can determine whether it is sarcastic or not, and 2) it allows large-scale dataset creation. However, this method introduces large amounts of noise due to lack of context, user errors, and common misuse on social media platforms. The sarcasm detection datasets created through manual annotation tend to be of higher quality but are typically much smaller. These problems are further compounded for non-English datasets, both manually labeled and automatically collected. Further, as sarcasm strongly relies on its context, using classical machine translation (MT) from English often produces inadequate results. This makes sarcasm detection in less-resourced languages, like Slovenian, an even bigger challenge. Therefore, developing reliable sarcasm detection models is of crucial importance for robust sentiment analysis in these languages.

We develop a methodology for sarcasm detection in less-resourced languages and test it on the Slovenian language. We address the problem of missing datasets by comparing state-of-the-art machine translation with large generative models. We explore the viability of such datasets and how the number of parameters affects a model's ability to detect sarcasm. We construct various ensembles of large pretrained language models and evaluate their performance.

The rest of this work is organized as follows. In Section 2, we discuss the proposed approach for detecting sarcasm in a less-resourced language such as Slovenian. We present the creation of a dataset, details of the training methodology and deployed ensemble techniques. We lay out our experimental results and their interpretations in Sections 2.3 and 4. In Section 5, we provide conclusions and directions for future work.

2 Sarcasm Detection Dataset

Existing attempts at automatic sarcasm detection have resulted in the creation of datasets in a small number of languages with differing sizes and quality. It is unclear if models trained on these datasets would generalize well to unseen languages [1]. Since no dataset exists for Slovenian, we leverage recent advances in machine translation and large language models (LLMs) to create a dataset for supervised sarcasm detection. We thus apply a translate-train approach when fine-tuning our models.

The prevalence of research done on sarcasm in English means that English datasets are usually larger and of higher quality than their counterparts in other languages. Further, as the translation from (and to) English is usually of better quality, we consider only English datasets.

Preliminary tests showed poor quality and poor translation ability of `Sarcasm on Reddit`[1] dataset, and `News Headlines Dataset For Sarcasm Detection`[2]. Hence, we chose the recent `iSarcasmEval`[3] dataset from the `SemEval-2022` shared task. We

[1] www.kaggle.com/datasets/danofer/sarcasm
[2] www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection
[3] github.com/iabufarha/iSarcasmEval

Information Society 2024, 7–11 October 2024, Ljubljana, Slovenia

https://doi.org/10.70314/is.2024.scai.4212

believe that relatively low performance scores obtained in this shared task could be improved with the use of larger LLMs.

2.1 iSarcasmEval Dataset

`iSarcasmEval` is a dataset of both English and Arabic sarcastic and non-sarcastic short-form tweets obtained from Twitter. We use only the English part, which is pre-split into a train and test set. Both sets are unbalanced, the former having 867 sarcastic and 2601 non-sarcastic examples, while the latter has 200 sarcastic and 1200 non-sarcastic examples. The authors of the shared task claim that both distant supervision and manual annotation of datasets produce noisy labels in terms of both false positives and false negatives [1]. Thus, they ask Twitter users to directly provide one sarcastic and three non-sarcastic tweets they have posted in the past. These responses are then filtered to ensure their quality. The produced dataset is not entirely clean since it contains links, emojis, and capitalized text. We chose to leave all of these potential features in the text, as they commonly occur in online conversations and could be indicative of sarcasm.

Let us mention, that an ensemble approach with 15 transformer models and transfer from three external sarcasm datasets proved to be the most accurate modeling technique for English [9] achieving an F_1-score of 0.605.

2.2 Translating iSarcasmEval

Our preliminary testing using smaller BERT-like classifiers showed that the models learned the distribution of the data and defaulted to the majority classifier (1200/1400 = 0.857 test accuracy). To try to dissuade this, we merged the train and test sets, kept all the sarcastic instances, and randomly sampled an equal number of non-sarcastic examples. This left us with a balanced dataset of 2134 samples.

To enable task specific instructions that would preserve sarcasm, we skipped classical machine translation tools, and tried two alternative translation approaches:

- using a medium-sized T5 model trained specifically for neural machine translation,
- leveraging a significantly larger model via OpenAI's API.

The T5 model uses both the encoder and decoder stacks of the Transformer architecture and is trained within a text-to-text framework. We chose Google's 32-layer T5 model called `MADLAD400-10B-MT`[4], which has 10.7 billion parameters and is pretrained on the `MADLAD-400` [4] dataset with 250 billion tokens covering 450 languages. Fine-tuning for machine translation was done on a combination of parallel data sources in 157 languages, including Slovenian.

As a generative model, we chose OpenAI's decoder-based `GPT-4o-2024-05-13`[5]. Its true size is not known to the public, but it's speculated that it is significantly smaller than `GPT-4`, since it is much faster and more efficient. OpenAI claims that it has the best performance across non-English languages of any of their models, thus making it suitable for our task.

When *prompting* generative decoder-based models, it is necessary to craft clear and specific instructions to achieve the best results. We used few-shot learning [2], and randomly sampled three training instances, manually translated them, and included them in the following prompt where the double forward slash was used as a delimiter between the query and the expected response.

You will be provided with a sarcastic/non-sarcastic sentence in English, and your task is to translate it into the Slovenian language. It should keep the original meaning. Examples:

- *love getting assignments at 6:25pm on a Friday!! // obožujem, ko mi v petek ob 18:25 pošljejo naloge!!*
- *I still can't believe England won the World Cup // Še vedno ne morem verjeti, da je Anglija zmagala na svetovnem prvenstvu*
- *taking spanish at ut was not my best decision* 😐 *// Učenje španščine na UT ni bila moja najboljša odločitev* 😐,

We manually assessed the outputs of both transformers in order to determine the best translations for fine-tuning detection models.

2.3 Translation Results

During translation, the T5 model sometimes had trouble with examples that had multiple newline characters in a row. It occasionally dropped parts of texts it didn't understand (mostly slang and various types of informal text styles). This shows that a 10B parameter model is not large enough to robustly translate all features of a language such as English into a less-resourced language such as Slovenian.

On the other hand, the GPT model performed surprisingly well in most instances and it seemed to have a more nuanced understanding of phrases used in online speech. It consistently translated entire texts, keeping the original structure and meaning. Consequently, we used GPT's translations when training sarcasm detection models. The translations can be seen in our repository[6].

3 Model Training

We tested the performance of a wide range of LLMs of different sizes. Their overview is contained in Table 1.

Table 1: Summary of used sarcasm detection models.

Model	Parameters
`SloBERTa`	110M
`BERT-BASE-MULTILINGUAL-CASED`	179M
`XLM-RoBERTa-BASE`	279M
`XLM-RoBERTa-LARGE`	561M
`META-Llama-3.1-8B-INSTRUCT`	8.03B
`META-Llama-3.1-70B-INSTRUCT`	70.6B
`META-Llama-3.1-405B-INSTRUCT`	406B
`GPT-3.5-TURBO-0125`	?
`GPT-4o-2024-05-13`	?

3.1 Encoder Models Under 1B Parameters

The four smallest models are encoder-based models that embed input text and use a classification head to assign it a class. They required additional fine-tuning to perform sarcasm detection. For these models, we conducted hyperparameter optimization.

We chose the `SloBERTa` [7, 8] model in order to check whether using a monolingual Slovenian model impacts sarcasm detection performance. We also wanted to compare BERT and RoBERTa-like models, so we used their multilingual variants and fine-tuned them on Slovenian data.

The models were trained for a maximum of five epochs with the use of early stopping, where the training was halted if the validation loss didn't improve after two epochs.

[4] huggingface.co/google/madlad400-10b-mt
[5] platform.openai.com/docs/models/gpt-4o
[6] github.com/GalaxyGHz/Diploma

3.2 Llama 3.1 Models

Since the teams that competed in the 2022 shared task on sarcasm mostly used BERT and RoBERTa models, we extend the testing to include significantly larger models. We chose Meta's open-source Llama family of models, more specifically, their newest Llama 3.1 variants. These come in three different sizes, which was perfect for studying the effects of parameter counts on sarcasm detection. We decided to use the "`instruct`" version of all three models since these were fine-tuned to be better at following instructions.

When prompting LLama and GPT generative models, the following few-shot classification prompt was given, with two positive and two negative examples randomly sampled from our dataset.

You will be provided with text in the Slovenian language, and your task is to classify whether it is sarcastic or not. Use ONLY token 0 (not sarcastic) or 1 (sarcastic) as in the examples:

- *Spanje? Kaj je to... Še nikoli nisem slišal za to? 1*
- *Lepo je biti primerjan z zidom* 😂 *1*
- *To sploh nima smisla. Nehaj kopati. 0*
- *Dne 12. 10. 21 ob 10:30 je bil nivo reke 0,37 m. 0.*

We used full versions of the 8B and 70B parameter models, while the 405B parameter model was loaded in 16-bit precision mode. To minimize the use of resources and costs, we employed LoRA parameter-efficient fine-tuning. We provided the models with training and validation sets and trained them for a maximum of 10 epochs. No hyperparameter optimization was conducted in this case due to time constraints. We used the validation loss to choose the best model, and we used the same early stopping technique as with the smaller models.

3.3 GPT 3 and 4 Models

We also tested two models offered on the OpenAI platform, `GPT-4o-2024-05-13` and `GPT-3.5-TURBO-0125`. We first used them in few-shot mode and classified all our examples without any fine-tuning. When fine-tuning, the platform's tier system limited us to only the smaller `GPT-3.5-TURBO-0125` model. We fine-tuned the model for a maximum of three epochs. In the end, we used the model with the lowest validation loss to classify the examples in the test set.

3.4 Sarcasm Detection Ensembles

When constructing ensemble models, we tried two techniques: stacking and voting. In both cases, we used the predicted probability of the sarcastic class from each model as input features.

3.4.1 Stacking With Regularized Logistic Regression. Our first ensemble used stacking approach, and logistic regression with Ridge regularization as the meta-level classifier. This choice was motivated primarily by the need for feature selection, as we wanted to identify the most important model predictions and determine which models would be assigned a lower weight. The best models were then used for voting.

3.4.2 Standard and Mixed Voting. The second ensembling method was voting. We tried cut-off-based mixed voting inspired by [9]. Formally, we used hard voting when the absolute difference between the number of sarcastic and non-sarcastic predictions was greater than n, and we used soft voting otherwise. We optimized the value of n based on the ensembles performance on our validation set.

When n is set to zero, this approach is equivalent to hard voting, and in the case of n being equal to the predictor count, it is equivalent to soft voting. We report both results. Additionally, we compare the results of voting using all trained models with the results obtained by using only the models with large weights in our regularized logistic regression ensemble.

4 Sarcasm Detection Results

Table 2 summarizes all our results. It is roughly sorted by model size, smaller models being on top and larger ones being on bottom. The `(NFT)` tag indicates that a model was not fine-tuned, while the `(LoRA)` tag means that a model was trained with LoRA. Results are rounded to three decimal places.

Table 2: Summary of performance results for all tested models. The best scores are in bold.

Model	Accuracy	F_1-score
`SloBERTa`	0.621	0.632
`BERT-BASE-MULTILINGUAL-CASED`	0.499	0.666
`XLM-RoBERTa-BASE`	0.578	0.579
`XLM-RoBERTa-LARGE`	0.550	0.597
`Llama-3.1-8B-INSTRUCT (NFT)`	0.560	0.676
`Llama-3.1-8B-INSTRUCT (LoRA)`	0.569	0.682
`Llama-3.1-70B-INSTRUCT (NFT)`	0.660	0.724
`Llama-3.1-70B-INSTRUCT (4-bit-LoRA)`	0.637	0.717
`Llama-3.1-405B-INSTRUCT (16-bit-NFT)`	0.686	0.751
`GPT-3.5-TURBO-0125 (NFT)`	0.564	0.679
`GPT-3.5-TURBO-0125`	0.749	0.760
`GPT-4o-2024-05-13 (NFT)`	0.686	0.746
`L2-LOGISTIC-REGRESSION`	**0.759**	**0.765**
`L2-LOGISTIC-REGRESSION-NON-COMMERCIAL`	0.707	0.749
`HARD-VOTING-ALL`	0.670	0.738
`SOFT-VOTING-ALL`	0.658	0.732
`HARD-VOTING-BEST-5`	0.686	0.749
`SOFT-VOTING-BEST-5`	0.686	0.749

Individual Model Performance

Out of all of the used models, only `BERT-BASE-MULTILINGUAL--CASED` failed to learn any pattern in our data and defaulted to the dummy classifier.

`GPT-3.5-TURBO-0125` sometimes predicts the correct token but then continues to generate additional text, such as 11 and 1\n1. This happens with a small quantity of examples in our testing set. We decided to truncate these responses and only kept the first token as the answer.

The Llama models sometimes refused to generate tokens zero or one. We decided to drop these examples altogether. We report the test accuracy and trained the ensemble models without them.

Smaller encoder models performed poorly when compared to some of the larger models. Only the `SloBERTa` model manages to achieve an accuracy above 0.6. Despite being the smallest of the four small models we tested, `SloBERTa` performed the best. This suggests that the three larger multilingual encoder models may lack sufficient understanding of Slovenian. It also highlights that model size alone does not necessarily correlate with better performance when it comes to sarcasm detection.

The Llama models fared better, achieving accuracies of up to 0.686 with the 405B model being comparable to GPT-4o in performance. They still fell short of the fine-tuned `GPT-3.5-TURBO-0125` model, which managed to successfully classify about three-quarters of our examples with a F_1-score of 0.76.

Some models had significantly higher F_1-scores and lower accuracies. We show the confusion matrix of one of the models

Table 3: Confusion Matrix for non-fine-tuned `Llama-3.1-405B-INSTRUCT` model.

Predicted \ Actual	Positive	Negative
Positive	202	123
Negative	11	91

that exhibited the largest difference in Table 3. These models seem to have a tendency to incorrectly classify non-sarcastic text as sarcastic, leading to a high rate of false positives.

Our testing also showed that loading the `Llama-3.1-70B-INSTRUCT` model in 4-bit mode and fine-tuning it with LoRA does not produce satisfactory results, and it is thus better to conduct full fine-tuning with the smaller Llama model or to use one of OpenAI's models via their fine-tuning API.

`GPT-3.5-TURBO-0125` performed the best among individual models, so if costs associated with the use of OpenAI's API are acceptable, we recommend its use for sarcasm detection in Slovenian. This shows that very large models can effectively identify sarcasm. We believe that with better parameter tuning, Llama 8B could be one of the best (and most economical) options for sarcasm detection in Slovenian, provided that the user has sufficient hardware resources.

Ensemble Model Performance

We observed that the regularized logistic regression mostly relied on the best-performing models. Its focus on the best model (`GPT-3.5-TURBO-0125`), however, suggests that there is significant overlap between the various model predictions.

We decided to discard `BERT-BASE-MULTILINGUAL-CASED` when constructing our voting ensembles since its dummy classification didn't contribute to overall model performance. Both of these two voting classifiers had an odd number of predictors, so there was no need for a tie-breaker mechanism.

Voting proved to be ineffective in our setups, even scoring lower than some of its base models. hard voting generally outperformed soft voting. We also note that there was no benefit in using mixed voting, at least for the sets of predictors that we obtained as hard voting always had a higher accuracy. This was true for both the classifiers that used all and only five of the base models.

Regularized logistic regression managed to improve on the scores of individual models, raising accuracy by one percent, thus achieving the best performance out of all of the tested approaches. This shows that there is still performance to be gained from ensembles; however, it is still necessary to use commercial models for top performance.

5 Conclusion

In this work, we presented the task of sarcasm detection in the less-resourced Slovenian language. Our code and results are freely available[7].

We tackled the missing dataset problem by using two LLMs to perform neural translation of an English dataset into Slovenian. The translations generated by `GPT-4o-2024-05-13` outpaced those generated by a large T5 model specifically trained for neural machine translation in terms of quality.

We used this data to train a plethora of Transformer-based models in various settings. We found that fine-tuning `GPT-3.5-TURBO-0125` via OpenAI's API results in the highest individual Slovenian sarcasm detection power, but we also note that a possible alternative could be local fine-tuning of the `Llama-3.1-8B-INSTRUCT` model. Our testing shows that using aggressive quantization combined with LoRA results in significant performance degradation.

We also constructed ensemble models based on voting and stacking methods. Observations showed that voting didn't result in any performance improvements. On the other hand, stacking with the use of a regularized logistic regression managed to improve on the performance of its base models.

Additional work needs to be done in dataset construction. Sarcastic examples could be extended with context or labels of the types of sarcasm they represent. This might help guide models towards better understanding of sarcasm. Future work could also explore incorporating heterogeneous models into ensembles or the creation of Mixture of Experts (MoE) ensembles, whose baseline models would focus on different aspects of sarcasm.

[7] github.com/GalaxyGHz/Diploma

Acknowledgements

This research was supported by the Slovenian Research and Innovation Agency (ARIS) core research programme P6-0411 and projects J7-3159, CRP V5-2297, L2-50070, and PoVeJMo.

References

[1] Ibrahim Abu Farha, Silviu Vlad Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 task 6: iSarcasmEval, intended sarcasm detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 802–814. DOI: 10.18653/v1/2022.semeval-1.111.

[2] Tom Brown et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*. Vol. 33, 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[3] Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: a survey. *ACM Comput. Surv.*, 50, 5, Article 73, 22 pages. DOI: 10.1145/3124420.

[4] Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. [n. d.] Madlad-400: a multilingual and document-level large audited dataset. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* Article 2940, 13 pages.

[5] Bleau Moores and Vijay Mago. 2022. A survey on automated sarcasm detection on twitter. *arXiv preprint*. DOI: 10.48550/arXiv.2202.02516.

[6] Smaranda Muresan, Roberto Gonzalez-Ibanez, Debanjan Ghosh, and Nina Wacholder. 2016. Identification of nonliteral language in social media: a case study on sarcasm. *Journal of the Association for Information Science and Technology*, 67, 11, 2725–2737. DOI: 10.1002/asi.23624.

[7] Matej Ulčar and Marko Robnik Šikonja. 2021. Sloberta: slovene monolingual large pretrained masked language model. In *Proceedings of Data Mining and Data Warehousing, SiKDD*, 17–20. http://library.ijs.si/Stacks/Proceedings/InformationSociety/2021/IS2021_Volume_C.pdf.

[8] Matej Ulčar and Marko Robnik Šikonja. 2021. *Slovenian RoBERTa contextual embeddings model: SloBERTa 2.0. CLARIN.SI data & tools.* Nasl. z nasl. zaslona. Fakulteta za računalništvo in informatiko. http://hdl.handle.net/11356/1397.

[9] Mengfei Yuan, Zhou Mengyuan, Lianxin Jiang, Yang Mo, and Xiaofeng Shi. 2022. Stce at SemEval-2022 task 6: sarcasm detection in English tweets. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 820–826. DOI: 10.18653/v1/2022.semeval-1.113.