

Optimization and Evaluation in Machine Learning Challenges

Doctoral thesis

Jakub Pokrywka

Supervisor: **prof. UAM dr hab. Filip Graliński**
Discipline: **Computer and Information Sciences**



Faculty of Mathematics and Computer Science
Adam Mickiewicz University, Poznań
Poznań, Poland
2023

Optymalizacja i ewaluacja w wyzwaniach uczenia maszynowego

Rozprawa doktorska

Jakub Pokrywka

Promotor: **prof. UAM dr hab. Filip Galiński**
Dyscyplina: **Informatyka**



Wydział Matematyki i Informatyki
Uniwersytet im. Adama Mickiewicza w Poznaniu
Poznań, Polska
2023

Abstract

To develop new machine learning methods, it is necessary to evaluate them reliably. This doctoral thesis discusses some aspects of preparing machine learning challenges and techniques for developing their solutions. The work consists of seven papers published in international conference proceedings concerning natural language processing, computer vision, and time series forecasting. The thesis author is the sole author of three of them, the first author of three others, and a second author of the remaining one. Three papers introduce new challenges, describing the methodology of dataset acquisition, preparation of dataset splits, choice of evaluation metric, and preparation of baselines. One paper reports the improvement of an existing challenge and evaluates various methods for it. The remaining three papers provide solutions to existing challenges, including model optimization techniques.

Streszczenie

W celu rozwoju nowych metod uczenia maszynowego konieczna jest ich rzetelna ewaluacja. Niniejsza praca doktorska opisuje pewne aspekty metodyki tworzenia wyzwań uczenia maszynowego oraz technik opracowywania ich rozwiązań. Praca składa się z cyklu siedmiu artykułów opublikowanych w materiałach pokonferencyjnych międzynarodowych konferencji. Publikacje dotyczą przetwarzania języka naturalnego, widzenia komputerowego i prognozowania szeregów czasowych. W trzech z nich autor dysertacji jest jedynym autorem, w innych trzech jest pierwszym autorem, w ostatniej jest drugim autorem. Trzy prace wprowadzają nowe wyzwania, opisując metodologię pozyskania datasetu, podziału między danymi trenującymi i testowymi, doboru metryk ewaluacyjnych, przygotowywania baseline. Jedna praca opisuje usprawnienie istniejącego wyzwania oraz ewaluuje szereg modeli w ramach tego wyzwania. Pozostałe trzy prace prezentują rozwiązania do istniejących wyzwań i zawierają między innymi techniki optymalizacji modeli.

Acknowledgements

I would like to thank my supervisor, prof. UAM dr hab. Filip Galiński, for his supervision during my research and teaching work. Collaborative research and industrial work with you has taught me a lot. Thank you for your countless clever research ideas and enthusiasm.

I would also like to express my gratitude to prof. Krzysztof Jassem, the director of my department and the Centre for Artificial Intelligence, where I work. Thank you for your advice on my Ph.D. work, for providing me with great opportunities for applied ML research, and for obtaining funds and computer resources.

Many thanks to my family for their constant love, support, and belief in me.

Contents

1	Introduction	1
1.1	Foreword	1
1.2	Scope of the Thesis	2
1.3	Structure of the Thesis	4
2	Research Papers Overview	7
2.1	Challenging America: Modeling language in longer time scales	8
2.2	Temporal Language Modeling for Short Text Document Classification with Transformers	9
2.3	Modeling Spaced Repetition with LSTMs	10
2.4	Using Transformer models for gender attribution in Polish	11
2.5	YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection	12
2.6	Efficient GPU Training of a Diversified Model Ensemble for the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022)	13
2.7	Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images	14
3	Research Papers	15
3.1	Challenging America: Modeling language in longer time scales	15
3.2	Temporal Language Modeling for Short Text Document Classification with Transformers	29
3.3	Modeling Spaced Repetition with LSTMs	38
3.4	Using Transformer models for gender attribution in Polish	47
3.5	YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection	53
3.6	Efficient GPU Training of a Diversified Model Ensemble for the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022)	61
3.7	Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images	69
4	Declarations of Contribution	75

Chapter 1

Introduction

1.1 Foreword

The supervised Machine Learning (ML) paradigm assumes that a model is trained on some data. ML models are often compared with each other using some evaluation metric. Metrics may be computed automatically (Accuracy, F_k -score, BLEU [1]) or may involve human intervention like manual annotations, A/B tests, etc. Some good practices have already been developed for structuring a machine learning problem; for example, creating training, development, and test dataset splits. This concerns not only evaluating a model on different data samples than the training sample, but also, for example, splitting samples between time periods or between different users, or balancing the classes in test data in some cases. Choosing an evaluation metric, which may depend on the class distribution, is a crucial step. A trivial example of metric choice is between accuracy and F_k -score for binary classification. Accuracy may be suitable for balanced class distribution, but the F_k score may be better for imbalanced datasets. The choice of the metric may also depend on its usability in real-case scenarios. A False Negative may be a more serious mistake than a False Positive for some medical diagnostic tests, so the k value should then be calibrated to pay more attention to recall than precision. Another example is the very advanced machine translation evaluation metric COMET [2], a neural model that aims to obtain a high correlation with human judgments, which has recently become very popular and slowly displaces metrics based on static formulas, such as BLEU or METEOR [3]. In this work, *Machine Learning Challenge (ML Challenge)* is defined as a dataset with an evaluation metric and task setup. The dataset is divided into training, validation, and test sets, with the training and validation datasets being optional. The setup includes rules, such as allowing public ML models but prohibiting the usage of annotated data other than provided training dataset.

A clearly defined task formulation with dataset splits, metrics, and setup is crucial for developing new ML models, because it enables researchers to compare different solutions. Sometimes, a dataset may not be published alongside a paper if it contains sensitive data or data valuable for business. The release of a vast good-quality dataset may lead to rapid progress in a field, as in the case of ImageNet [4] in computer vision or Google Ngram Viewer [5] in natural language processing. Sometimes the dataset is deficient, as in the Twitter Sentiment Analysis [6] work, where the authors published a dataset with only 1000 test-set samples. They probably did not foresee how impactful their work would be in the future. It may also happen that the problem on a certain dataset is solved, and the dataset needs to be upgraded, as in the case where MNIST [7] was upgraded to

Fashion-MNIST [8] or GLUE [9] was upgraded to SuperGLUE [10]. If there are no established datasets or baselines for an ML problem, authors may compare their solutions with methods that are too weak, as described in [11]. It is a good habit of machine learning challenge creators to deliver evaluation scripts for ease of use and trustworthy results, especially when the evaluation procedure is not obvious. An example may be the BLEU metric evaluation script, which besides the complicated formula, requires a specific text tokenization procedure. Different tokenization methods may produce different results. Another example of a complicated metric is Interpolated Average Precision (introduced in [12]), commonly used for object detection tasks [13]. It is even better when, in addition to evaluation scripts, there is a benchmark hosted on an evaluation server with hidden expected values for the test dataset, as in the GLUE benchmark or KLEJ benchmark [14]. This prevents cheating or accidental dataset leaks. Some platforms hosting multiple ML challenges are Kaggle, Gonito [15], and KnowledgePit [16]. Often the challenges are introduced as *shared tasks* (which is just a different name for ML challenges) for a workshop of a conference on a one-off basis (Semantic Shift Detection Challenge [17]) or cyclically, but with different data (Workshop on Machine Translation [18]). Some challenges, such as GLUE and SuperGLUE, are hosted continuously and independently of the workshops.

Apart from the perspective of ML challenge creators, there is also that of ML challenge participants. For creators, it is beneficial to see the perspective of participants. This allows them to create more interesting challenges that will bring progress in the field. Problems are valuable if they require the development of new innovative methods or at least the comparison of multiple existing methods. Less important are problems where participants compete only on available GPU resources using an existing framework resulting from an obvious choice. Often, contributed ML solutions are considered only in terms of a given evaluation metric. Yet, other aspects of the ML method are also assigned importance. This may concern computer resources or time required for both training and inference, the ability to be trained on a smaller dataset, error analysis, etc. However, for the best model performance, certain techniques, such as ensemble learning, prove beneficial in ML competitions. The most useful solutions are those that are meticulously described in a report of some form and are associated with a reproducible source code.

To conclude, the motivation for this work is to enable progress in the ML field by advancing the methodology of preparing ML challenges and developing solutions for them.

1.2 Scope of the Thesis

This doctoral thesis consists of a series of seven research papers concerning the process of preparing ML challenges and developing solutions for them. The list of papers is given in Table 1.1.

Title	Authors	Venue	Points
Challenging America: Modeling language in longer time scales	<u>Jakub Pokrywka,</u> <u>Filip Graliński,</u> Krzysztof Jassem, Karol Kaczmarek, Krzysztof Jurkiewicz, Piotr Wierzchoń	Findings of the Association for Computational Linguistics: NAACL 2022	140
Temporal Language Modeling for Short Text Document Classification with Transformers	<u>Jakub Pokrywka,</u> <u>Filip Graliński</u>	2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)	70
Modeling Spaced Repetition with LSTMs	<u>Jakub Pokrywka,</u> Marcin Biedalak, <u>Filip Graliński,</u> Krzysztof Biedalak	In Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023), In Print	70
Using Transformer models for gender attribution in Polish	Karol Kaczmarek, <u>Jakub Pokrywka,</u> <u>Filip Graliński</u>	2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)	70
YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection	<u>Jakub Pokrywka</u>	2022 IEEE International Conference on Big Data (Big Data)	70
Efficient GPU Training of a Diversified Model Ensemble for the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022)	<u>Jakub Pokrywka</u>	2022 IEEE International Conference on Big Data (Big Data)	70
Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images	<u>Jakub Pokrywka</u>	2022 IEEE International Conference on Big Data (Big Data)	70

Table 1.1: List of research papers included in the doctoral thesis. Points stand for Ministerstwo Edukacji i Nauki (Ministry of Science and Higher Education) points.

This work covers selected aspects of evaluation and optimization techniques for machine learning challenges, as the whole topic is very broad. It concerns natural language processing, computer vision, and time series forecasting tasks. The first three papers describe the creation of machine learning challenges with dataset generation procedures, dataset collection, and evaluation metric choice. These works also contain baseline solutions or more advanced models. The fourth paper, *Using Transformer models for gender attribution in Polish*, concerns a challenge introduced several years earlier [19], but introduces some dataset changes, a new evaluation metric, and a human baseline. It also contributes a variety of different ML solutions. The remaining three papers describe methods used for three different machine learning competitions hosted at the 2022 IEEE International Conference on Big Data. The fourth and subsequent works listed in the table concern some aspects of model optimization for shared tasks, for instance, ensemble learning, testing of different ML approaches (linear regression, support vector machines, gradient-boosted trees, neural networks), knowledge transfer from synthetic to real samples, and efficient GPU training. The solution described in *Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images* achieved second place in the shared task competition, and the solution described in *YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection* achieved third place. The author of this doctoral thesis is the sole or first author of all except one of the seven papers. He has presented all of the papers.

1.3 Structure of the Thesis

The remainder of the thesis is structured as follows. Chapter 2 presents an overview of the papers included in the doctoral thesis. Chapter 3 contains research papers in the same form as were published in the conference proceedings. Chapter 4 contains declarations of contributions to papers with more than one author.

Bibliography

- [1] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.
- [2] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, “Comet: A neural framework for MT evaluation,” *arXiv preprint arXiv:2009.09025*, 2020.
- [3] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, IEEE, 2009.
- [5] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, G. B. Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, *et al.*, “Quantitative analysis of culture using millions of digitized books,” *Science*, vol. 331, no. 6014, pp. 176–182, 2011.
- [6] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.
- [7] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [8] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017.
- [9] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Brussels, Belgium), pp. 353–355, Association for Computational Linguistics, Nov. 2018.
- [10] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “SuperGLUE: A stickier benchmark for general-purpose language understanding systems,” *arXiv preprint 1905.00537*, 2019.
- [11] J. Lin, “The neural hype and comparisons against weak baselines,” *SIGIR Forum*, vol. 52, p. 40–51, jan 2019.

- [12] G. Salton, “Introduction to modern information retrieval,” *McGraw-Hill*, 1983.
- [13] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
- [14] P. Rybak, R. Mroczkowski, J. Tracz, and I. Gawlik, “KLEJ: Comprehensive benchmark for Polish language understanding,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 1191–1201, Association for Computational Linguistics, July 2020.
- [15] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierzchoń, “Gonito.net – open platform for research competition, cooperation and reproducibility,” in *Proceedings of the 4REAL Workshop* (A. Branco, N. Calzolari, and K. Choukri, eds.), pp. 13–20, 2016.
- [16] A. Janusz, D. Ślęzak, S. Stawicki, and M. Rosiak, “Knowledge pit - a data challenge platform,” in *International Workshop on Concurrency, Specification and Programming*, 2015.
- [17] D. Schlechtweg, B. McGillivray, S. Hengchen, H. Dubossarsky, and N. Tahmasebi, “SemEval-2020 task 1: Unsupervised lexical semantic change detection,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 1–23, International Committee for Computational Linguistics, Dec. 2020.
- [18] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri, “Findings of the 2016 conference on machine translation,” in *Proceedings of the First Conference on Machine Translation*, (Berlin, Germany), pp. 131–198, Association for Computational Linguistics, August 2016.
- [19] F. Graliński, Ł. Borchmann, and P. Wierzchoń, ““He Said She Said” — a male/female corpus of Polish,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, eds.), (Paris, France), European Language Resources Association (ELRA), 2016.

Chapter 2

Research Papers Overview

2.1 Challenging America: Modeling language in longer time scales

Authors: Jakub Pokrywka, Filip Graliński, Krzysztof Jassem, Karol Kaczmarek, Krzysztof Jurkiewicz, Piotr Wierchoń

Venue: Findings of the Association for Computational Linguistics: NAACL 2022

Presentation type: Poster

Presenter: Jakub Pokrywka

Paper URL: <https://aclanthology.org/2022.findings-naacl.56>

Challenges URLs:

<https://gonito.csi.wmi.amu.edu.pl/challenge/challenging-america-word-gap-prediction>

<https://gonito.csi.wmi.amu.edu.pl/challenge/challenging-america-year-prediction>

<https://gonito.csi.wmi.amu.edu.pl/challenge/challenging-america-geo-prediction>

Challenges: This is a benchmark for temporal language models for longer time scales (several hundred years). The tasks are to predict a masked word given a date, predict a date given a text, and predict the geo coordinates of a newspaper given a text and date. In addition to the OCR-ed text, a newspaper scan is also provided. The long-term aim of the research is to gain an in-depth understanding of the world before the Internet age.

Author contribution:

Implementation of the algorithm for generating machine learning challenges according to the methodology proposed by prof. Graliński. Idea and implementation of the algorithm for selecting images and text fragments from newspapers. Preparation of scripts for pretraining the temporal language model. Proposal of Haversine metric for geo coordinate task. Creation of the baseline models for challenges. Results analysis. Writing of the article.

Paper overview:

This paper presents Challenging America: a set of three temporal NLP tasks with a large pretraining corpus. The tasks are masked language modeling, temporal classification, and geo coordinate prediction. The data is collected from the Chronicling America project. During the tasks, a special future-proof methodology for generating challenges was developed. This methodology splits newspaper editions between datasets and allows the creation of additional ML challenges without data contamination between them. An example task, released after the paper’s publication, cnlps-ticrc (<https://gonito.csi.wmi.amu.edu.pl/challenge/cnlps-ticrc/readme>) associated with the FeDCSIS conference (<https://fedcsis.org/sessions/aaia/cnlps>) was created with this methodology, which proves its future-proof usefulness. The usefulness of metrics for the tasks was discussed: perplexity hashed for masked language modelling, root mean squared error for fractional year prediction, and Harvesine for geo coordinate prediction. Strong neural baselines are provided: regular RoBERTa and temporal RoBERTa. All of the challenges are hosted on the Gonito platform with a test set with expected values hidden from challenge participants. The models and code were released alongside the paper.

2.2 Temporal Language Modeling for Short Text Document Classification with Transformers

Authors: Jakub Pokrywka, Filip Graliński

Venue: 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)

Presentation type: Oral presentation

Presenter: Jakub Pokrywka

Paper URL: <https://ieeexplore.ieee.org/document/9908605>

Challenges URLs:

<https://gonito.csi.wmi.amu.edu.pl/challenge/ireland-news-headlines>

<https://gonito.csi.wmi.amu.edu.pl/challenge/sentiment140>

Challenge: The task is to predict a document class based on text and temporal information of daily resolution. In contrast to the previous paper, the task introduced is downstream and on a short temporal scale (several decades).

Author contribution: Conceptualization and methodology. Selecting and preparing the text corpora. Creating diachronic challenges. Implementation of the machine learning models (especially based on temporal embeddings). Running experiments. Analyzing data and results. Writing most of the article.

Paper overview:

The impact of temporal information in the text classification task is measured, and different methods of incorporating date components into the language model are examined. These methods are prepending the date and two methods of creating embeddings from date components. Two challenges of text classification incorporating temporal metadata were created and hosted on the Gonito platform. Two dataset splits were presented: regular, and splitting between time periods. The source code of the experiments was released. The experiments showed that temporal language models achieve better results than regular language models, but the method of date component incorporation does not show significant differences in text classification results.

2.3 Modeling Spaced Repetition with LSTMs

Authors: Jakub Pokrywka, Marcin Biedalak, Filip Graliński, Krzysztof Biedalak

Venue: In Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023), In Print

Presentation type: Oral presentation

Presenter: Jakub Pokrywka

Paper URL: <https://www.insticc.org/node/TechnicalProgram/CSEDU/2023/presentationDetails/117240>

Challenge: The challenge is to predict the probability of a user recalling an item from a flash-card. The data consist of real user logs.

Author contribution: Implementation of part of the ML methods, including the idea and implementation of the XGBoost method with exponential decay. Analyzing the results on the general test data with regard to the various metrics. Writing of the article.

Paper overview:

Spaced repetition is a method humans use to learn information items, such as words in a foreign language. This technique aims to optimize time intervals between repetitions to maximize learning efficiency. The research describes several machine learning models for predicting the probability of a student recalling an item, which is crucial for creating a spaced repetition algorithm. The work involved collecting and creating a machine-learning challenge based on real user data from the SuperMemo learning platform. The dataset split takes into account student courses and students. Due to the label imbalance, the choice of evaluation metrics was crucial. The best-performing method turned out to be the novel approach of LSTM with an exponential decay model.

2.4 Using Transformer models for gender attribution in Polish

Authors: Karol Kaczmarek [Jakub Pokrywka](#), Filip Graliński

Venue: 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)

Presentation type: Oral presentation

Presenter: [Jakub Pokrywka](#)

Paper URL: <https://ieeexplore.ieee.org/document/9908765/>

Challenge URL:

<https://gonito.csi.wmi.amu.edu.pl/challenge/petite-difference-challenge2>

Challenge: This task, consisting of over 3 billion items, aims to predict the gender of the author of a text in Polish.

Author contribution: Implementation of TFIDF, fastText, and LSTM methods. Implementation of some of the models using Transformer architecture, including Monte-Carlo model averaging. Acquisition and supervision of annotators. Data annotation. Partial data preparation for contamination analysis. Writing of the article.

Paper overview:

The work in this paper approaches the problem of predicting the gender of an author of a given text in the Polish language. The research is based on a challenge previously released and hosted on the Gonito platform, but extends it with some dataset improvements and a Likelihood metric. The dataset is based on large Internet corpora. Many methods, including TF-IDF, fastText, LSTM, Polish RoBERTa with and without Monte-Carlo model averaging, and a human baseline, were tested, distinguishing self-contained and non-self-contained cases. All of the solutions, with source code, were submitted to the Gonito challenge and released. The Polish RoBERTa transformer model vastly outperforms other methods. The work also contains data contamination analysis and discussion.

2.5 YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection

Authors: [Jakub Pokrywka](#)

Venue: 2022 IEEE International Conference on Big Data (Big Data)

Presentation type: Oral presentation

Presenter: [Jakub Pokrywka](#)

Paper URL: <https://ieeexplore.ieee.org/document/10020576/>

Challenge URL: <https://vod2022.sekilab.global/>

Challenge: The task is to detect a vehicle in an image and predict the class and its orientation. For training, only synthetic data is allowed. The challenge's goal is to develop research on using low-cost synthetic data from simulators.

Paper overview:

This paper describes a third place-winning solution to the IEEE BigData 2022 Vehicle Class and Orientation Detection Challenge 2022. The shared task was to create an object detection model for vehicle class and orientation. The models were evaluated on real-world traffic images, but the shared task rules allowed model training solely on the synthetic datasets generated by a simulator. The proposed solution utilized an ensemble of object detection models with specially adjusted dataset augmentation settings to perform well on real-world images.

2.6 Efficient GPU Training of a Diversified Model Ensemble for the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022)

Authors: [Jakub Pokrywka](#)

Venue: 2022 IEEE International Conference on Big Data (Big Data)

Presentation type: Oral presentation

Presenter: [Jakub Pokrywka](#)

Paper URL: <https://ieeexplore.ieee.org/document/10020877/>

Challenge URL: <https://crddc2022.sekilab.global/>

Challenge: This is an object detection challenge for road damage. The machine learning models developed during the shared task may replace road damage detection vehicles with expensive specialized sensors.

Paper overview:

Road maintenance inspection is usually carried out with expensive specialized vehicles. The Crowdsensing-based Road Damage Detection Challenge (CRDDC2022) aims to create a road maintenance inspection solution utilizing images taken from low-budget smartphones mounted inside a car. This paper describes a method that consists of an ensemble of models on different levels of dataset augmentation settings. Successive models are initialized from the preceding runs (with lower dataset augmentation settings), which allows efficient GPU time utilization. The source code for the solution was released.

2.7 Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images

Authors: [Jakub Pokrywka](#)

Venue: 2022 IEEE International Conference on Big Data (Big Data)

Presentation type: Oral presentation

Presenter: [Jakub Pokrywka](#)

Paper URL: <https://ieeexplore.ieee.org/document/10020495>

Challenge URL: <https://knowledgepit.ai/privacy-preserving-matching-of-images/>

Challenge: This task aims to match a source and encoded images. The challenge tests the reliability of several encryption algorithms.

Paper overview:

The work describes a method for matching original and encrypted images, and is a part of the Privacy-preserving Matching of Encrypted Images shared task in the IEEE BigData 2022 Cup. Despite the fact that the inputs are images, gradient boosted trees were used instead of a neural network, in contrast to other top winning solutions. The method was chosen in view of the encryption method, which shuffles pixels. The input to the models consisted of RGB and grayscale histogram vectors. The solution achieved second place, trailing the winning solution by a mere 0.0031 accuracy.

Chapter 3

Research Papers

3.1 Challenging America: Modeling language in longer time scales

Challenging America: Modeling language in longer time scales

Jakub Pokrywka Adam Mickiewicz University
Filip Graliński Adam Mickiewicz University
Krzysztof Jassem Adam Mickiewicz University

Karol Kaczmarek Adam Mickiewicz University
Krzysztof Jurkiewicz Adam Mickiewicz University
Piotr Wierzchoń Adam Mickiewicz University
Applica.ai

Abstract

The aim of the paper is to apply, for historical texts, the methodology used commonly to solve various NLP tasks defined for contemporary data, i.e. pre-train and fine-tune large Transformer models. This paper introduces an ML challenge, named Challenging America (ChallAm), based on OCR-ed excerpts from historical newspapers collected from the *Chronicling America* portal. ChallAm provides a dataset of clippings, labeled with metadata on their origin, and paired with their textual contents retrieved by an OCR tool. Three, publicly available, ML tasks are defined in the challenge: to determine the article date, to detect the location of the issue, and to deduce a word in a text gap (cloze test). Strong baselines are provided for all three ChallAm tasks. In particular, we pre-trained a RoBERTa model from scratch from the historical texts. We also discuss the issues of discrimination and hate-speech present in the historical American texts.

1 Introduction

The dominant approach in the design of current NLP solutions is (pre-)training a large neural language model, usually applying a Transformer architecture, such as GPT-2, RoBERTa or T5, and fine-tuning the model for specific tasks (Devlin et al., 2019; Raffel et al., 2019). The solutions are evaluated on benchmarks such as GLUE (Wang et al., 2019b) or SuperGLUE (Wang et al., 2019a), which allow comparing the performance of various methods designed for the same purpose. An important feature of a good NLP benchmark is the clear separation between train and test sets. This requirement prevents data contamination, when the model (pre-)trained on huge data might have “seen” the test set in some form.

The expansion of digital information is proceeding in two directions on the temporal axis. In the forward direction, new data are made publicly available on the Internet every second. What is less

obvious is that, in the backward direction, older and older historical documents are digitized and disseminated publicly.

To the best of our knowledge, our paper introduces the first benchmark which serves to use and evaluate the “pre-train and fine-tune scenario” applied to a massive collection of historical texts.

The very idea of building language models on historical data is not new. The Google Ngram Viewer (Michel et al., 2011) is based on large amounts of texts from digitized books. The corpus as a whole is not open for the NLP community – only raw n-gram statistics are available. The temporal information is crude (at best, the year of publication is given) and the corpus is heterogeneous (in fact, it is a dump of digitized books of any origin).

In our research, we use one of the richest sources of homogeneous historical documents, **Chronicling America**, a collection of digitized newspapers that cover the publication period of over 300 years (with significant coverage of 150 years), and design an NLP benchmark that may open new opportunities for the modeling of the historical language.

Recently, time-aware language models such as Temporal T5 (Dhingra et al., 2021) and TempoBERT (Rosin et al., 2021) have been proposed. They focus on modern texts dated yearly, whereas we extend language modeling towards both longer time scales and more fine-grained (daily) resolution, using massive amounts of historical texts.

The contribution of this paper is as follows:

- We extracted a large corpus of English historical texts that may serve to pre-train historical language models (Section 5).

These are the main features of the corpus:

- the corpus size is 74 GB (201 GB of total raw text), which is comparable with

- contemporary text data for training massive language models, such as GPT-2, RoBERTa or T5;
 - the corpus is free of spam and noisy data (although the quality of OCR processing varies);
 - texts are dated with a daily resolution, hence a new dimension of time (on a fine-grained level) can be introduced into language modeling;
 - the whole corpus is made publicly available;
- Based on selected excerpts from *Chronicling America*, we define a suite of challenges (named *Challenging America*, or *ChallAm* in short) with three ML tasks combining layout recognition, information extraction and semantic inference (Section 7). We hope that *ChallAm* will give rise to a historical equivalent of the GLUE (Wang et al., 2019b) or SuperGLUE (Wang et al., 2019a) benchmarks.
 - In particular, we provide a tool for the intrinsic evaluation of language models based on a word-gap task, which calculates the model perplexity in a comparative scenario (the tool may be used in competitive shared tasks) (Section 7.3).
- We propose a “future-proof” methodology for the creation of NLP challenges: a challenge is automatically updated whenever the underlying corpus is enriched (Section 4).
- We introduce a method for data preparation that prevents data contamination (Section 4).
- We train base Transformer (RoBERTa) models for historical texts (Section 5). The models are trained on texts spanning 100 years, dated with a daily resolution.
- We provide strong baselines for three *ChronAm* challenges (Section 8).
- We take under consideration the issue of discrimination and hate speech in the historical American texts. To this end we have applied up-to date methods to tag the abusive content from the data (Section 9).

2 Related Machine Learning datasets and challenges

This section concerns ML challenges which deliver labeled OCR documents as training data, a definition of the processing task, and an evaluation environment to estimate the performance of uploaded solutions. More often than not, such challenges concern either layout recognition (localization of layout elements) or Key Information Extraction (finding, in a document, precisely specified business-actionable pieces of information). Layout recognition in Japanese historical texts is described in (Shen et al., 2020). The authors use deep learning-based approaches to detect seven types of layout element categories: Page Frame, Text Region, Text Row, Title Region, etc. Some Key Information Extraction tasks are presented in (Stanisławek et al., 2021). The two datasets described there contain, respectively, NDA documents and financial reports from charity organizations. The tasks for the datasets consist in detecting data points, such as effective dates, interested parties, charity address, income, spending. The authors provide several baseline solutions for the two tasks, which apply up-to-date methods, pointing out that there is still room for improvement in the KIE research area. A challenge that comprises both layout recognition and KIE is presented in (Huang et al., 2019) – the challenge is opened for the recognition of OCR-scanned receipts. In this competition (named ICDAR2019) three tasks are set up: Scanned Receipt Text Localization, Scanned Receipt OCR, and Key Information Extraction from Scanned Receipts.

A common feature of the above-mentioned challenges is the goal of retrieving information that is explicit in the data (a text fragment or layout coordinates). Our tasks in *ChallAm* go a step further: the goal is to infer the information from the OCR image rather than just retrieve it.

Similar challenges for two out of the three tasks introduced in this paper have been proposed before for the Polish language:

- a challenge for temporal identification (Graliński and Wierchoń, 2018); the challenge was based on a set of texts coming from Polish digital libraries, dated between the years 1814 and 2013;
- a challenge for “filling the gap” (Retro-Gap) (Graliński, 2017) with the same training

set as above.

The training sets for those challenges were purely textual. Here, we introduce the challenges with the addition of original images (clippings), though we do not use graphical features in baselines yet.

3 Chronicling America

In 2005 a partnership between the National Endowment for the Humanities and the Library of Congress launched the National Digital Newspaper Program, to develop a database of digitized documents with easy access. The result of this 15-year effort is Chronicling America – a website¹ which provides access to selected digitized newspapers, published from 1690 to the present. The collection includes approximately 140 000 bibliographic title entries and 600 000 library holdings records, converted to the MARCXML format. The portal supports an API which allows accessing of the data in various ways, such as the JSON format, BulkData (bulk access to data) or Linked Data,² or searching of the database with the OpenSearch protocol.³ The accessibility of data in various forms makes Chronicling America a valuable source for the creation of datasets and benchmarks.

The portal serves as a resource for various research activities. Cultural historians may track performances and events of their interest in a resource which is easily and openly accessible, as opposed to commercial databases or “relatively small collections of cultural heritage organizations whose online resources are isolated and difficult to search” (Clark, 2014). The database enables searching for the first historical usages of word terms. For instance, thanks to the Chronicling America portal, it was discovered in (Cibaroğlu, 2019) that the term “fake news” was first used in 1889 in the Polish newspaper *Ameryka*.

The resource is helpful in research aiming to improve the output of the OCR process. The authors of (Nguyen et al., 2019) study OCR errors occurring in several digital databases – including Chronicling America – and compare them with human-generated misspellings. The research results in several suggestions for the design of OCR post-processing methods. The implementation of an unsupervised approach in the correction of OCR

documents is described in (Dong and Smith, 2018). Two million issues from the Chronicling America collection of historic U.S. newspapers are used in a sequence-to-sequence model with attention.

Chronicling America is a type of digitized resource that may be of wide use for both humanities and computational research. We prepared datasets and challenges based on the data from the Chronicling America resource. We hope that our initiative will bring about research that will facilitate the development of ML-based processing tools, and consequently increase access to digitized resources for the humanities.

An example of an ML tool based on Chronicling America is described in (Lee et al., 2020). The task was to predict bounding boxes around various types of visual content: photographs, illustrations, comics, editorial cartoons, maps, headlines and advertisements. The training set was crowd-sourced and included over 48K bounding boxes for seven classes. Using a pre-trained Faster-RCNN detection object, the researchers achieved an average accuracy of 63.4%. Both the training set and the model weights file are publicly available. Still, it is difficult to estimate the value of the results achieved without any comparison with other models trained on the same data.

In our proposal we go a step further. We provide and make freely available training data from Chronicling America for three ML tasks. For each task we develop and share baseline solutions. Alternative solutions can be submitted to the Gonito⁴ evaluation platform (Graliński et al., 2016, 2019) to be evaluated automatically and compared against our baselines.

4 Data processing

The PDF files were downloaded from Chronicling America and processed using a pipeline primarily developed for extracting texts from Polish digital libraries (Graliński, 2013, 2019). Firstly, the metadata (including URL addresses for PDF files) were extracted by a custom web crawler and then normalized; for instance, titles were normalized using regular expressions (e.g. *The Bismarck tribune. [volume], May 31, 1921* was normalized to *THE BISMARCK TRIBUNE*). Secondly, the PDF files were downloaded and the English texts were processed into DjVu files (as this is the target format

¹<https://chroniclingamerica.loc.gov>

²<https://www.w3.org/standards/semanticweb/data>

³<https://opensearch.org/>

⁴<https://gonito.net>

Table 1: Statistics for the raw data obtained from the Chronicling America website

Documents with metadata obtained	1 877 363
... in English	1 705 008
... downloaded	1 683 836
... processed into DjVu files	1 665 093

for the pipeline) using the pdf2djvu tool⁵. The original OCR text layer was retained (the files were not re-OCR'd, even though, in some cases, the quality of OCR was low).

Table 1 shows a summary of the data obtained at each processing step. Two factors were responsible for the fact that not 100% of files were retained at each phase: (1) issues in the processing procedures (e.g. download failures due to random network problems or errors in the PDF-to-DjVu procedure that might be handled later); (2) some files are simply yet to be finally processed in the ongoing procedure.

The procedure is executed in a continuous manner to allow the future processing of new files that are yet to be digitized and made public by the Chronicling America initiative. This solution requires a *future-proof* procedure for splitting and preparing data for machine-learning challenges. For instance, the assignment of documents to the training, development and test sets should not change when the raw data set is expanded. Such a procedure is described in Section 6.

5 Data for unsupervised training

The state of the art in most NLP tasks is obtained by training a neural-network language model on a large collection of texts in an unsupervised manner and fine-tuning the model on a given downstream task. At present, the most popular architectures for language models are Transformer (Devlin et al., 2019) models (earlier, e.g. Word2vec (Mikolov et al., 2013) or LSTM models (Peters et al., 2017)). The data on which such models are trained are almost always modern Internet texts. The high volume of texts available at Chronicling America, on the other hand, makes it possible to train large Transformer models for historical texts.

Using a pre-trained language model on a downstream task bears the risk of *data contamination* – the model might have been trained on the task

⁵<http://jwilk.net/software/pdf2djvu>

test set and this might give it an unfair edge (see (Brown et al., 2020) for a study of data contamination in the case of the GPT-3 model when used for popular English NLP test sets). This issue should be taken into account from the very beginning. In our case, we release⁶ a dump of all Chronicling America texts (for pre-training language models), but limited only to the 50% of texts that would be assigned to the training set (according to the MD5 hash). This dump contains *all* the texts, not just the excerpts described in Section 6.2. As the size of the dump is 74.0G characters, it is on par with the text material used to train, for instance, the GPT-2 model.

We also release a RoBERTa Base ChallAm model trained on the text corpus. The model was trained from scratch, i.e. it was *not* based on the weights of the original RoBERTa model (Liu et al., 2019). The BPE dictionary was also induced anew.

Two versions of the RoBERTa ChallAm model were prepared: one⁷ was trained with temporal metadata encoded as a prefix of the form `year: YYYY, month: MM, day: DD, weekday: WD`, another⁸ for comparison, without such a prefix. The ChallAm models have the same number of parameters as the original RoBERTa Base (125M). Each model was trained on two Tesla V100 32GB GPUs for 9 days.

6 Procedure for preparing challenges

We created a pipeline that can generate various machine learning challenges. The pipeline input should consist of DjVu image files, text (OCR image), and metadata. Our main goals are to keep a clear distinction between dataset splits and to assure the reproducibility of the pipeline. This allows potential improvement to current challenges and the generation of new challenges without dataset leaks in the future. We achieved this by employing *stable* pseudo-randomness by calculating an MD5 hash on a given ID and taking the modulo remainder from integers from certain preset intervals. These pseudo-random assignments are not dependent on any library, platform, or programming language (using a fixed seed for the pseudo-random

⁶<https://gonito.net/get/data/challenging-america-full-train-dump-2021-10-26.tsv.xz>

⁷<http://gonito.net/get/data/roberta-challam-base-with-date-1325000.zip>

⁸<http://gonito.net/get/data/roberta-challam-base-without-date-1325000.zip>

generator might not give the same guarantees as using MD5 hashes), so they are easy to reproduce.

This procedure is crucial to make sure that challenges are *future-proof*, i.e.:

- when the challenges are re-generated on the same Chronicling America files, exactly the same results are obtained (including text and image excerpts; see Section 6.2);
- when the challenges are re-generated on a larger set of files (e.g. when new files are digitized for the Chronicling America project), the assignments of existing items to the train/dev/test sets will not change.

6.1 Dataset structure

All three of our machine learning challenges consist of training (train), development (dev), and test sets. Each document in each set consists of excerpts from a newspaper edition. One newspaper edition provides a maximum of one excerpt. Excerpts in the datasets are available as both a cropped PNG file from the newspaper scan (a “clipping”) and its OCR text. This makes it possible to employ image features in machine learning models (e.g. font features, paper quality). A solution might even disregard the existing OCR text layer and re-OCR the clipping or just employ an end-to-end model. (The OCR layer is given as it is, with no manual correction done – this is to simulate realistic conditions in which a downstream task is to be performed without a perfect text layer.)

Sometimes additional metadata are given. For the train and dev datasets, we provide the expected data. For the test dataset, the expected data are not released. These data are used by the Gonito evaluation platform during submission evaluation. All newspaper and edition IDs are encoded to prevent participants from checking the newspaper edition in the Chronicling America database. The train and dev data may consist of all documents which meet our criteria for text excerpts, so the data may be unbalanced with respect to publishing years and locations. We tried to balance the test sets as regards the years of publication (the year-prediction and word-gap challenges) or locations (the geo-prediction challenge), though it is not always possible due to large imbalances in the original material.

6.2 Selecting text excerpts

The details of the procedure for selection of text excerpts is given in Appendix A. A sample excerpt is

shown in Figure 1a. Note that excerpts are selected using a stable pseudo-random procedure based on the newspaper edition ID (similarly to the way the train/dev/test split is done, see Section 6.3).

6.3 Train/dev/test split

Each newspaper has its newspaper ID (i.e. normalized title, as described in Section 4), and each newspaper edition has its newspaper edition ID. We separate newspapers within datasets, so for instance, if one newspaper edition is assigned to the dev set, all editions of that newspaper are assigned to the dev set. All challenges share common train and dev datasets and no challenges share the same test set. This prevents one from checking expected data from other challenges. The set splits are as follows: 50% for train, 10% for dev, 5% for each challenge test set. This makes it possible to generate eight challenges with different test sets. In other words, there is room for another five challenges in the future (again this is consistent with the “future-proof” principle of the whole endeavor).

7 Challenging America tasks

In this section, we describe the three tasks defined in the challenge. They are released on the Gonito evaluation platform, which enables the calculation of metrics both offline and online, as well as the submission of solutions. An example of text from an excerpt given in those tasks is shown in Figure 1b.

7.1 RetroTemp

This⁹ is a temporal classification task. Given a normalized newspaper title and a text excerpt, the task is to predict the publishing date. The date should be given in fractional year format (e.g. 1 June 1918 is represented as the number 1918.4137, and 31 December 1870 as 1870.9973).

Hence, solutions to the challenge should predict the publication date with the greatest precision possible (i.e. day if possible). The fractional format will make it easy to accommodate even more precise timestamps, for example, if modern Internet texts (e.g. tweets) are to be added to the dataset.

Due to the regression nature of the problem, the evaluation metric is RMSE (root mean square error).

⁹<https://gonito.net/challenge/challenging-america-year-prediction>

Perhaps one of the most interesting political developments in the political history of California is that which has been disclosed as a result of the quarrel of Leland Stanford and Collis P. Huntington, of the Southern and Central Pacific Railways, and which has been suppressed as to details, after the scandal has embraced a whole continent. It is probable that much matter for good will ultimately result from this and other indecent developments. Prior to the arrival of Mr. Huntington on this Coast the people of California were in danger of being deluged in a stream of adulation directed towards Senator Stanford. Although Stanford notoriously purchased his seat in the United States Senate, and although his purchase of that seat, considering his obligations to Senator Sargent, was a matter of never to be forgotten treachery, the toad-eaters of the mighty Senator are intent upon having censures swung in his honor. Whatever good there may ever have been in Leland Stanford has been overwhelmed in a sea of toadyism for years. For a long and wearisome decade his ear has never been reached by the voice of the people. Enjoying a seat in the United States Senate purchased by coin, by coin he directs towns and cities to be illuminated in his honor. Nero, the corrupt Emperor of the Romans, never directed towards himself a more feculent stream of corrupt adulation than Stanford has caused to be discharged into fountains of bought public opinion, playing in his honor. During the coming campaign the people will at last have an opportunity of dismantling this edifice, raised to flagitious greatness, and which will be buried under the reprobation of the people.

(a) An excerpt.

Perhaps one of the most interesting political developments in the political history of California is that which has been disclosed as a result of the quarrel of Leland Stanford and Collis P. Huntington, of the Southern and Central Pacific Railways, and which has been suppressed as to details, after the scandal has embraced a whole continent. It is probable that much matter for good will ultimately result from this and other indecent developments. Prior to the arrival of Mr. Huntington on this Coast the people of California were in danger of being deluged in a stream of adulation directed towards Senator Stanford. Although Stanford notoriously purchased his seat in the United States Senate, and although his purchase of that seat, considering his obligations to Senator Sargent, was a matter of never to be forgotten treachery, the toad-eaters of the mighty} Senator are intent upon having censures swung in his ...

(b) Fragment of a text from an excerpt.

Figure 1: An example of an excerpt

The motivation behind the RetroTemp challenge is to design tools that may help supplement the missing metadata for historical texts (the older the document, the more often it is not labeled with a time stamp). Even if all documents in a collection are time-stamped, such tools may be useful for finding errors and anomalies in metadata.

7.2 RetroGeo

The task¹⁰ is to predict the place where the newspaper was published, given a normalized newspaper title, text excerpt, and publishing date in fractional year format. The expected format is the latitude and longitude. In the evaluation the distance on the sphere between output and expected data is calculated using the haversine formula, and the mean value of errors is reported.

The motivation for the task (besides the supplementation of missing or wrong data) is to allow research on news propagation. Even if a news article is labeled with the localization of its issue, an automatic tool may infer that it was originally published somewhere else.

¹⁰<https://gonito.net/challenge/challenging-america-geo-prediction>

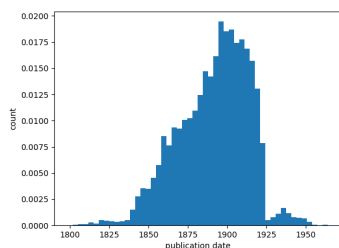
7.3 RetroGap

This¹¹ is a task for language modeling. The middle word of an excerpt is removed in the input document (in both text and image), and the task is to predict the removed word, given the normalized newspaper title, the text excerpt, and the publishing date in fractional year format (in other words, it is a cloze task). The output should contain a probability distribution for the removed word (not just a word or a single probability). The metric is perplexity; PerplexityHashed, to be precise, as implemented in the GEval evaluation tool (Graliński et al., 2019), the modification is analogous to LogLossHashed in (Graliński, 2017), its goal is to ensure proper evaluation in the competitive (shared-task) setup (i.e. avoid self-reported probabilities and ensure objective comparison of all reported solutions, including out-of-vocabulary words).

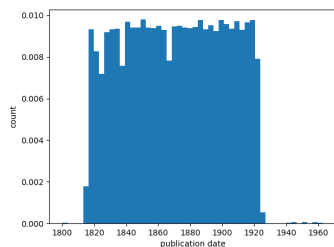
7.4 Statistics

The data consists of the text excerpts written between the years 1798 and 1963. The mean publication year of the text excerpts is 1891. Excerpts between the years 1833 and 1925 make up about 96% of the data in the train set (cf. Figure 2a), but only 85% in the dev and test sets, which are more uniform (due to balancing described in Section 4,

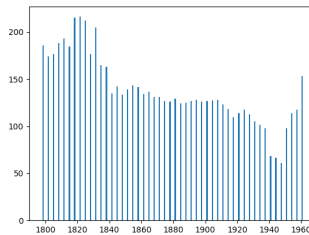
¹¹<https://gonito.net/challenge/challenging-america-word-gap-prediction>



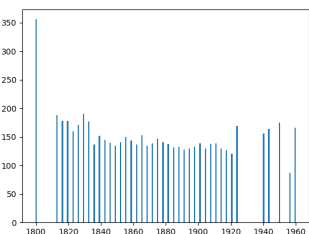
(a) Excerpt counts vs. publication dates in train set.



(c) Excerpt counts vs. publication dates in dev/test set.



(b) Average excerpt length vs. publication dates in train set.



(d) Average excerpt length vs. publication dates in dev/test set.

Figure 2: Statistics for the RetroTemp challenge

cf. Figure 2c). There are 432 000 excerpts in the train set, 10 500 in the dev set and 8 500 in the test set. These numbers are consistent across the challenges. The average excerpt length is 1 745 characters with 323.8 words, each one containing from 150 words up to 583 words.

The length of each text in the excerpts seems to have a negative correlation with publication date – the later the text was published, the shorter snippet text (on average) it contains (see Figure 2b and 2d).

8 Results

Strong baselines for all three tasks are available at the Gonito evaluation platform. The baselines (see Tables 2 and 3) include, for each model, its score in the appropriate metric as well as the Git SHA1 reference code in the Gonito benchmark (in curly brackets). Reference codes can be used to access any of the baseline solutions at <http://gonito.net/q>.

We distinguish between self-contained submissions, which use only data provided in the task, and non-self-contained submissions, which use external data, e.g. publicly available pre-trained transformers. Our baselines take into account only textual features.

More detailed analysis of the baseline performance is given in Appendix C. The current top performing models have the most difficulty with

texts which (1) are older, (2) contain OCR noise, (3) come from less popular locations (especially, in the west).

8.1 RetroTemp and RetroGeo

The baseline solutions for RetroTemp and RetroGeo were prepared similarly. RetroGeo requires two values (latitude and longitude) – we treat them separately and train two separate regression models for them.

For the self-contained models we provide the mean value from the train test, the linear regression based on TF-IDF and the BiLSTM (bidirectional long short-term memory) method.

For non-self-contained submissions, we incorporate RoBERTa (Liu et al., 2019) models released in two versions: base (125M params) and large (355M params). The output features are averaged, and the linear layer is added on top of this. Both RoBERTa and the linear layer were fine-tuned during training.

The best self-contained models are BiLSTM submissions in both tasks. Non-self-contained submissions result in much higher scores than self-contained models. In both tasks, RoBERTa-large with linear layer provides better results than RoBERTa-base.

For the RetroTemp challenge we also provide results obtained with the RoBERTa model pre-

trained from scratch (see Section 5). Even though the model without time-related prefix was used, the results are significantly better than the original RoBERTa Base: the confidence intervals obtained with bootstrap sampling are, respectively, 10.81 ± 0.21 and 12.10 ± 0.22 (single runs are reported).

Hyperparameter setup is described in Appendix B.

8.2 RetroGap

For non-self-contained submissions, we applied RoBERTa in base and large version without any fine-tuning. Since standard RoBERTa training does not incorporate any data, but text, we did not include temporal metadata during inference.

For self-contained submissions, we applied RoBERTa Challam base both in version with a date and without a date.

RoBERTa Challam base with date is better than RoBERTa Challam base without date. This means the incorporation of temporal metadata has a positive impact on the MLM task. Both self-contained submissions are better than the standard RoBERTa base, so our models trained on historical data performs better than models trained on regular data if the same base model size is considered. Since we did not train RoBERTa Challam large, we cannot confirm this holds true, when it comes to large RoBERTa models. The standard RoBERTa large is the best performing model, so in this case, a larger model is better even if not trained on the data from different domain.

9 Ethical issues

We share the data from Chronicling America, following the statement of the Library of Congress: “The Library of Congress believes that the newspapers in Chronicling America are in the public domain or have no known copyright restrictions.”¹²

Historical texts from American newspapers may be discriminatory, either explicitly or implicitly, particularly regarding race and gender. Recent years have seen research on the detection of discriminatory texts. In (Xia et al., 2020) adversarial training is used to mitigate racial bias. In (Field and Tsvetkov, 2020) the authors “take an unsupervised approach to identifying gender bias against women at a comment level and present a model that can

surface text likely to contain bias.” The most recent experiments on the topic ((Caselli et al., 2021), (Aluru et al., 2020)) result in re-trained BERT models for abusive language detection in English. We use one of them, DeHateBERT (Aluru et al., 2020), to detect the abusive texts in the ChallAm dataset. We tagged items that either (1) are marked as abusive speech by DeHateBERT with the probability greater than 0.75 or (2) contain words from a list of blocked words. The fraction of detected texts was 2.04-2.40 % (depending on the challenge and set). The tags along with the probabilities are available in the `hate-speech-info.tsv` files for each test directory.

Note that temporal and geospatial metadata might constitute useful features in future work on better detection of hate speech in historical texts.

10 Conclusions

This paper has introduced a challenge based on OCR excerpts from the Chronicling America portal. The challenge consists of three tasks: guessing the publication date, guessing the publication location, and filling a gap with a word. We propose baseline solutions for all three tasks.

Chronicling America is an ongoing project, as we define our challenge in such a way that it can easily evolve in parallel with the development of Chronicling America. Firstly, any new materials appearing on the portal can be automatically incorporated into our challenge. Secondly, the challenge is open for five yet undefined ML tasks.

Acknowledgements

This work was partially supported by the *Cyfrowa Infrastruktura Badawcza dla Humanistyki i Nauk o Sztuce DARIAH-PL* project (POIR.04.02.00-00-D006/20).

References

- Sai Saket Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. *Deep learning models for multilingual hate speech detection*. *ArXiv preprint*, abs/2004.06465.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric

¹²<https://chroniclingamerica.loc.gov/about>

Table 2: Baseline results for the RetroTemp/Geo challenges. * indicates non-self-contained models.

Model	RetroTemp		RetroGeo	
	git ref	RMSE	git ref	Haversine
mean from train	{fbf19b}	31.50	{766824}	1321.47
tf-idf with linear regression	{63c8d4}	17.11	{8acd61}	2199.36
BiLSTM	{f7d7ed}	13.95	{d3d376}	972.71
RoBERTa Base + linear layer*	{1159e6}	12.07	{08412c}	827.13
RoBERTa Large + linear layer*	{2e79c8}	8.15	{7a21dc}	651.20
RoBERTa ChallAm Base + linear layer*	{d0ddf4}	10.80	—	—

Table 3: Baseline results for the RetroGap challenge. * indicates non-self-contained models.

Model	git ref	Perplexity
RoBERTa base (no fine-tune)	{166e03}	72.10
RoBERTa large (no fine-tune)	{bf5171}	52.58
RoBERTa ChallAm Base (without date)*	{f96da0}	56.64
RoBERTa ChallAm Base (with date)*	{3ebfc0}	53.76

Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. [HateBERT: Retraining BERT for abusive language detection in English](#). In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Mehmet Cibaroğlu. 2019. Post-truth in social media. 6:87–99.

Maribeth Clark. 2014. [A survey of online digital newspaper and genealogy archives: Resources, cost, and access](#). *Journal of the Society for American Music*, 8:277–283.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and

William W. Cohen. 2021. [Time-aware language models as temporal knowledge bases](#).

Rui Dong and David Smith. 2018. [Multi-input attention for unsupervised OCR correction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2363–2372, Melbourne, Australia. Association for Computational Linguistics.

Anjalie Field and Yulia Tsvetkov. 2020. [Unsupervised discovery of implicit gender bias](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 596–608, Online. Association for Computational Linguistics.

Filip Graliński, Rafał Jaworski, Łukasz Borchmann, and Piotr Wierzchoń. 2016. Gonito.net – open platform for research competition, cooperation and reproducibility. In António Branco, Nicoletta Calzolari, and Khalid Choukri, editors, *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pages 13–20.

Filip Graliński and Piotr Wierzchoń. 2018. RetroC—A Corpus for Evaluating Temporal Classifiers. In *Human Language Technology. Challenges for Computer Science and Linguistics. 7th Language and Technology Conference, LTC 2015*, pages 101–111. Springer.

Filip Graliński, Anna Wróblewska, Tomasz Stanisławek, Kamil Grabowski, and Tomasz Górecki. 2019. [GEval: Tool for debugging NLP datasets and models](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural*

- Networks for NLP*, pages 254–262, Florence, Italy. Association for Computational Linguistics.
- Filip Graliński. 2013. Polish digital libraries as a text corpus. In *Proceedings of 6th Language & Technology Conference*, pages 509–513, Poznań. Fundacja Uniwersytetu im. Adama Mickiewicza.
- Filip Graliński. 2017. (Temporal) language models as a competitive challenge. In *Proceedings of the 8th Language & Technology Conference*, pages 141–146. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu.
- Filip Graliński. 2019. *Against the Arrow of Time. Theory and Practice of Mining Massive Corpora of Polish Historical Texts for Linguistic and Historical Research*. Wydawnictwo Naukowe UAM, Poznań.
- Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. [ICDAR2019 competition on scanned receipt OCR and information extraction](#). *2019 International Conference on Document Analysis and Recognition (ICDAR)*.
- Benjamin Lee, Jaime Mears, Eileen Jakeway, Meghan Ferriter, Chris Adams, Nathan Yarasavage, Deborah Thomas, Kate Zwaard, and Daniel Weld. 2020. The newspaper navigator dataset: Extracting and analyzing visual content from 16 million historic newspaper pages in Chronicling America.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *ArXiv preprint*, abs/1907.11692.
- Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Thi-Tuyet-Hai Nguyen, Adam Jatowt, Mickael Coustaty, Nhu-Van Nguyen, and Antoine Doucet. 2019. [Deep statistical analysis of OCR errors for effective post-OCR processing](#). In *Proceedings of the 18th Joint Conference on Digital Libraries, JCDL '19*, page 29–38. IEEE Press.
- Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. [Semi-supervised sequence tagging with bidirectional language models](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *ArXiv preprint*, abs/1910.10683.
- Guy D. Rosin, Ido Guy, and Kira Radinsky. 2021. [Time masking for temporal language models](#).
- Zejiang Shen, Kaixuan Zhang, and Melissa Dell. 2020. [A large dataset of historical Japanese documents with complex layouts](#). *ArXiv preprint*, abs/2004.08686.
- Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. [Kleister: Key information extraction datasets involving long documents with complex layouts](#). In *Document Analysis and Recognition – ICDAR 2021*, pages 564–579, Cham. Springer International Publishing.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. [Demoting racial bias in hate speech detection](#). In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 7–14, Online. Association for Computational Linguistics.

A Procedure for selecting text excerpts

The OCR text follows the newspaper layout, which is defined by the following entities: page, column, line. Each entity has x_0, y_0, x_1, y_1 coordinates of text in the DjVu document. Still, various errors may occur in the OCR newspaper layout (e.g. two columns may be split into one). We intend to select only excerpts which preserve the correct output. To this end, we select only excerpts that fulfill the following conditions:

1. There are between 150 and 600 text tokens in the excerpt. The tokens are words separated by whitespaces.

2. The y coordinates of each line are below the y coordinates of the previous line.
3. The x_0 coordinate of each line does not differ by more than 15% from the x_0 coordinate of the previous line.
4. The x_1 coordinate is not shifted to the right more than 15% from the x_1 coordinate of the previous line.

If the newspaper edition contains no such excerpts, we reject it. If there is more than one such excerpt, we select one excerpt using a stable pseudo-random procedure based on the newspaper edition ID.

This procedure produces text excerpts with images consisting of OCR texts only. The excerpts are downsized to reduce the size to an appropriate degree to maintain good quality. We do not pre-process images in any other way, so excerpts may have different sizes, height-to-width ratios, and colors.

B Hyperparameter setup

Hyperparameters were determined on the development set, training on a limited number of examples. In particular, for fine-tuning RoBERTa models the following hyperparameters were used:

- optimizer: AdamW
- learning rate: 0.000001
- batch size: 4
- early-stopping patience: 3
- warm-up steps: 10000

C Analysis of the best baselines

See Table 4 and 5 for the list of top 30 features correlating most with, respectively, the worst and bad results in ChallAm challenges (as returned by the GEval tool with the option `-worst-features-numerical-features` (Graliński et al., 2019)). The features are tokens within the input (`in:`), expected output (`exp:`) and the actual output (`out:`), or numerical features such as high/low value (`:=+/:=-`) or length/shortness of a text (`:+#/:-#`).

As can be seen the bottleneck for the current best model is due to:

- old texts (`:=` in RetroTemp),
- OCR noise (cf. short words such *ni*, *ol*, *j* or punctuation marks likely to be introduced by OCR misrecognitions),
- less popular publication locations (especially far west).

Obviously, year references (*1902*, *1904*) make it easy to guess the publication texts (in RetroTemp), whereas in RetroGap some non-content words such as *the*, *and*, *of* are easy to guess for the language model (even if their garbaged form, e.g. *ot*, *ol*, needs to be accounted for in the probability distribution).

Table 4: Features highly correlating with bad results

RetroTemp	RetroGeo	RetroGap
exp:=-	exp:=#+	exp:=#+
in<Text>;	in<Text>:=+	exp:,
in<Text>:nold	exp:-100.445882	exp:.
in<Text>:ni	exp:39.78373	out:.
in<Text>:she	exp:-115.763123	out:-
out:=-	exp:40.832421	in<LeftContext>:n
in<Text>:"	exp:-93.101503	out:,
in<Text>:aim	exp:44.950404	out;;
in<Text>:sav-	exp:-112.730038	out:'
in<Text>:ii	exp:46.395761	out:*
in<Text>:rifle	exp:-97.337545	in<RightContext>:*
in<Text>:hut	exp:37.692236	in<LeftContext>:>
in<Text>:!	exp:-76.062727	out:=#-
in<Text>:guilt	exp:39.697887	in<RightContext>:>
in<Text>:nLeave	exp:-106.487287	in<LeftContext>:i
in<Text>:ol	exp:31.760037	out:!
in<Text>:cold	exp:-81.772437	exp;;
in<Text>:contemplate	exp:24.562557	in<LeftContext>:*
in<Text>:nI	exp:-71.880373	in<RightContext>:l
in<Text>:thee	exp:44.814771	out:"
in<Text>:Ben-	out:=#+	out:
in<Text>:1945	exp:-135.313889	in<LeftContext>:l
in<Text>:God	exp:59.458333	out:1
in<Text>:it	exp:-112.077346	exp:"
in<Text>:noi	exp:33.448587	in<LeftContext>:<
in<Text>:man's	exp:-122.330062	in<LeftContext>:-
in<Text>:Roman	exp:47.603832	in<RightContext>:
in<Text>:I	exp:-112.942369	out:i
in<Text>:Henry	exp:46.128794	out:j
in<Text>:nford	exp:-90.184225	in<LeftContext>:e

Table 5: Features highly correlating with good results

RetroTemp	RetroGeo	RetroGap
in<Text>:Democratic	exp:44.007274	out:Of
in<Text>:defeat	exp:-80.85675	out:The
in<Text>:Secretary	exp:40.900892	out:ana
in<Text>:notice	exp:-77.804161	out:aud
in<Text>:July	exp:39.4301	out:by
in<Text>:General	exp:-79.96021	out:cf
in<Text>:1904	exp:37.274532	out:end
in<Text>:cent	exp:-82.137089	out:for
in<Text>:of	exp:38.844525	out:he
in<Text>:are	exp:-77.859581	out:in
in<Text>:will	exp:39.289184	out:io
in<Text>:1902	exp:-80.344534	out:lo
in<Text>:against	exp:39.280645	out:mat
in<Text>:nbeen	exp:-81.929558	out:of
in<Text>:Minnesota	exp:33.789577	out:ol
in<Text>:1903	exp:-77.321601	out:or
in<Text>:Judicial	exp:37.506699	out:ot
in<Text>:President	exp:-73.986614	out:tc
in<Text>:June	exp:-77.036646	out:te
in<Text>:to	exp:-77.047023	out:th
in<Text>:for	exp:-77.090248	out:tha
in<Text>:hereby	exp:-77.43428	out:that
in<Text>:States	exp:-80.720915	out:the
in<Text>:United	exp:37.538509	out:this
in<Text>:nLouisiana	exp:38.80511	out:tho
in<Text>:county	exp:38.81476	out:tie
in<Text>:State	exp:38.894955	out:tile
in<Text>:Is	exp:40.063962	out:to
in<Text>:cash	exp:40.730646	out:tu
in<Text>:In	out:-158.09514	out:und

3.2 Temporal Language Modeling for Short Text Document Classification with Transformers

Temporal Language Modeling for Short Text Document Classification with Transformers

Jakub Pokrywka, Filip Graliński

Adam Mickiewicz University,
 Faculty of Mathematics and Computer Science,
 Uniwersytetu Poznańskiego 4
 61-614 Poznań, Poland
 Email: {firstname.lastname}@amu.edu.pl

Abstract—Language models are typically trained on solely text data, not utilizing document timestamps, which are available in most internet corpora. In this paper, we examine the impact of incorporating timestamp into transformer language model in terms of downstream classification task and masked language modeling on 2 short texts corpora. We examine different timestamp components: day of the month, month, year, weekday. We test different methods of incorporating date into the model: prefixing date components into text input and adding trained date embeddings. Our study shows, that such a temporal language model performs better than a regular language model for both documents from training data time span and unseen time span. That holds true for classification and language modeling. Prefixing date components into text performs no worse than training special date components embeddings.

I. INTRODUCTION

MOST language models are trained solely on text data. Leveraging text domain, such as language [12] or style [10] into a language model may have a positive effect on it. Time of text authorship may be also considered as an input feature, but this poses specific challenges (and opportunities) as:

- time is continuous, whereas language is discrete, at any time moment, an event might change a language irreversibly and not trivial to combine time and language units both from the mathematical and practical standpoint;
- texts might reflect natural and social cycles (days, weeks, years, cyclical sport and political events);
- text content might be correlated with extralinguistic features, themselves correlating with time (e.g. air temperature).

Recently, the NLP community has started to use time as a feature in training and/or fine-tuning large neural models ([1], [16], [19]). Here, we analyze temporal language modeling in the context of two classification tasks in different timescales: Ireland News Headlines and Twitter Sentiment Analysis. We also incorporate date components other than year. We focus on examining different approaches to date incorporation (learnable embeddings, prefixing text) using periodic and non-periodic time features under a downstream classification task.

The contributions of this paper are as follows:

- two classification datasets were redefined in a common setup in which three time-related tasks are introduced: classification (possibly) using temporal metadata, predicting temporal metadata (as a regression task) and temporal language-modeling task (as a cloze task).
- we compared three methods for introducing temporal information into neural language models;
- we considered not only linear time, but also cycles such as years, weeks, and months;
- we measured the performance of RoBERTa [14] models in several setups on the two datasets (using different parts of the temporal information, and both fine-tuning and training from scratch);
- the relations between the temporal metadata, the texts and the results obtained were analyzed.

The datasets and source of our code are publicly available.

Generally, utilizing a date does not cost much effort, because many internet documents are available with a timestamp and it is possible to adapt existing models to new domain. Such temporal language models may contribute to:

- e-commerce search engines, e.g. users intention with short phrase "umbrella" may refer to umbrella protecting from a rain in the autumn or sun umbrella in the summer;
- other types of search engines, e.g. historical newspapers;
- OCR for historical documents.

II. DATASETS

Usually, text classification tasks do not incorporate time and other metadata. We suppose its impact is stronger for short texts due to shorter texts carrying less information. The time impact may be stronger for text, which may depend on people's mood or different interests. We carried out experiments with two large short-text classification datasets, where every sample is assigned a time stamp. One is spread over more than 20 years, the other ones — only 80 days. Both datasets are in English.

A. Ireland News

The dataset is available at Kaggle¹, its creator is Rohit Kulkarni. It consists of article headlines posted by the Irish

¹<https://www.kaggle.com/therohk/ireland-historical-news>

TABLE I: Categories count in datasets.

category	item			
	train	dev	test	test 20/21
news	603996	75963	75783	30278
business	162550	20330	20034	14477
sport	195384	24543	24346	13447
opinion	91697	11572	11528	8086
culture	67260	8525	8424	5643
life&style	65120	8093	8084	7188

Times newspaper. Each headline is accompanied by a timestamp and article category (text of an article is not included). There are six main categories: news, sport, opinion, business, culture, life&style. The datasets statistics are described in Table I. There are more fine-grained subcategories provided in the original dataset, but they vary over time, so we didn't make use of them in our experiments.

Timestamps range from 1996-01-01 to 2021-06-30. There are 1,611,495 such headlines in total.

We employed the date range from 1996-01-01 to 2019-12-31 for most of our experiments and created an additional test set, which consists of 2020-2021 years, which dates are non-overlapping with the rest of the dataset. We refer to this test set as **Ireland News 2020-2021**. The test set **Ireland News**, without year annotation, refers to time span from training data (1996-2019). Since train/dev/test split is not determined at the original dataset site, we assign each sample randomly to train/dev/test using the 80%/10%/10% split. This resulted in the 1,186,898 / 149,134 / 148,308 train/dev/test split. The average number of words in the dataset is 7.1 per headline.

B. Sentiment140

This sentiment analysis dataset is obtained and described in [2]. Since in the original dataset the train set contains 1,600,000 items (positive and negative tweets) and test set only 498 (positive, negative, and neutral tweets), we made significant modifications: neutral tweets were deleted from the test set, 100,000 random items were added to the test set, also a dev set was created by randomly selecting 100,000 samples from the train set. This resulted in the 1,400,000 / 100,000 / 100,359 train/dev/test split. Timestamps range from 2009-04-06 to 2009-06-25. The datasets set are balanced (~50% positive and ~50% negative tweets). The average number of words is 13.8 per item. Tweets are from users in different time zones. We take time local to the author of a tweet.

III. DATASETS ANALYSIS

The number of items per category differs in time. The distribution over days of month, months, years, weekdays in train datasets are presented in Figures 1 and 2 for, respectively, Sentiment140 and Ireland News. For the Sentiment140 dataset distribution over a year is not presented, since all items are from 2009. Mutual Information between presented factors and the class is given in Table V. In Ireland News, mutual information related to days of month and months is much lower than those of years and weekdays. In Sentiment140

mutual information is similar for days of month, months, and weekdays.

In both datasets, there are dependencies, which may be helpful for model performance. E.g. in Ireland News there are more sports texts on Friday and in Sentiment140 there are more negative texts on Wednesdays and Thursdays.

IV. TASKS

We created three tasks for each dataset: classification, 'fractional' year prediction, word gap prediction. Our main objective was to examine the impact of incorporating timestamps on text classification tasks. Fractional year prediction and word gap prediction tasks are mainly for analysis of the results in classification tasks.

We added timestamps in fractional-year form, which can be described by the following code:

```
days_in_year =
366 if year_is_leap_year else 365

fractional_year =
(year + (day_in_year-1+day) /
days_in_year )
```

Each item in our tasks is associated with a text, timestamp (day precision), fractional year, and category. Sample data is described in Table IV.

Each challenge for a given dataset uses the same train/dev/test split. The challenges are publicly available, courtesy of the site's owners, via the Gonito evaluation platform [3]. Source code of the challenge is available via the platform as well.

A. Classification

The task objective is to predict the headline category given text, date, and fractional year. The evaluation metric is simple accuracy. The challenges are available at: <https://gonito.net/challenge/ireland-news-headlines> (Ireland News) and <https://gonito.net/challenge/sentiment140> (Sentiment140). Dataset download and submission instructions are under the "How To" tab, source code is under the "All Entries" → catalog icon in each submission row.

B. Year prediction

The objective is to predict the year given the text. The metric is Root Mean Square Error (RMSE). The challenges are available at: <https://gonito.net/challenge/ireland-news-headlines-year-prediction> (Ireland News) and <https://gonito.net/challenge/sentiment140-year-prediction> (Sentiment140).

C. Word gap filling

The task objective is to predict a masked word, like in Masked Language Modeling, given text, date, fractional year. Word is defined by characters split by spaces. There is always exactly one masked word in each sample to

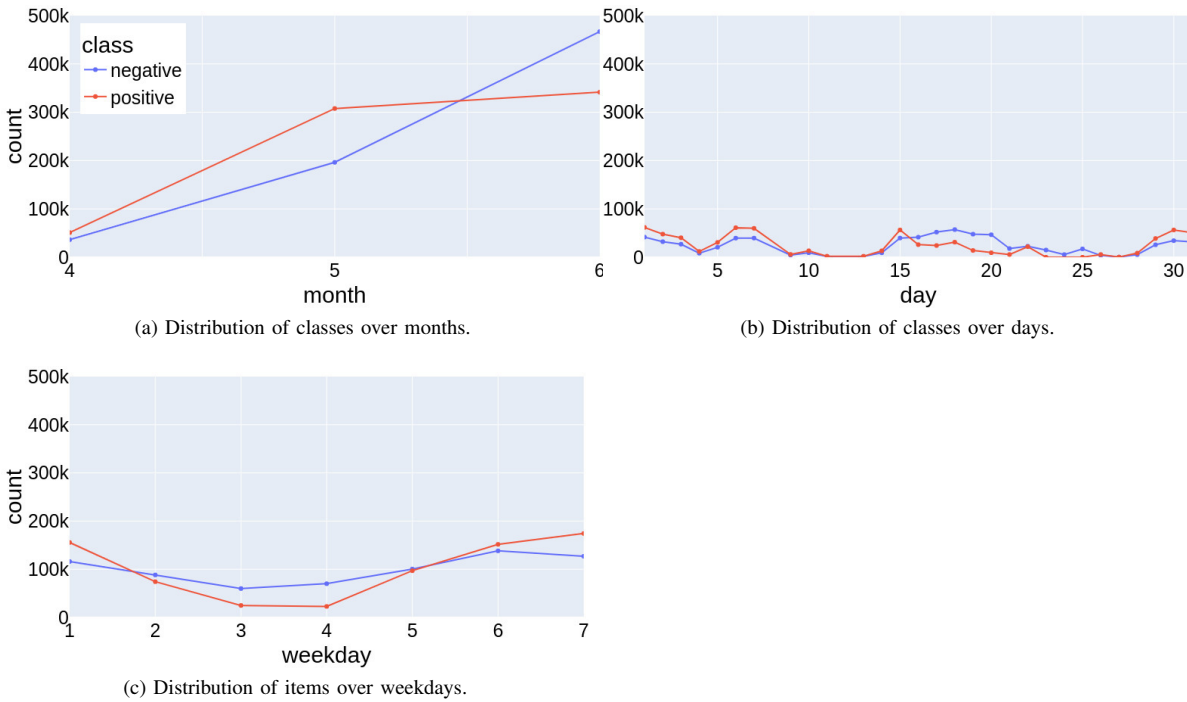


Fig. 1: Distribution of classes over date factors in Sentiment140 dataset. Distribution over year is not presented, since all items come from one year.

TABLE II: Samples from the Ireland News dataset. To check article-id visit www.irishtimes.com/article-id The article ID is not provided in the challenge.

fractional year	timestamp	text	category	article ID
2004.5082	20040705	Sudan claims it is disarming militias	news	1.1147721
2008.4426	20080611	Bluffer's guide to Euro 2008	sport	1.1218069
2017.1068	20170209	Gannon offers homes in Longview near Swords	life&style	1.2966726

predict. The metric is PerplexityHashed implemented in the GEval evaluation tool [4], which is a modified version of LogLossHashed as described by [5]. This metric ensures fair assessment disregarding model vocabulary. The challenges are available at: <https://gonito.net/challenge/ireland-news-headlines-word-gap> (Ireland News) and <https://gonito.net/challenge/sentiment140-word-gap> (Sentiment140).

V. METHODS

We used the RoBERTa model in the base version [14]. All models are described in this section. All code is publicly available via git commit hashes given in result tables.²

A. Regular Transformer as a baseline

The baseline is a regular RoBERTa with no temporal information. We refer to this method as noDate in result tables.

²Reference codes to repositories stored at Gonito.net [3] are given in curly brackets. Such a repository may be also accessed by going to <http://gonito.net/q> and entering the code there.

B. Temporal Transformer

We selected the following temporal information: year, month, day of the month (day), weekday. All of them are incorporated in our temporal models. We experimented with 3 ways of including temporal information into RoBERTa models. The first two involve slight RoBERTa model architecture changes and training new embeddings during RoBERTa training. The third one is only input data modification. They are described below.

1) *Date as embeddings added to every input token:* Temporal embeddings are added to every input token as: $embedding = token_emb + pos_emb + year_emb + month_emb + monthday_emb + weekday_emb$ for each $token_pos$. We refer to this method as addedEmbDate in result tables.

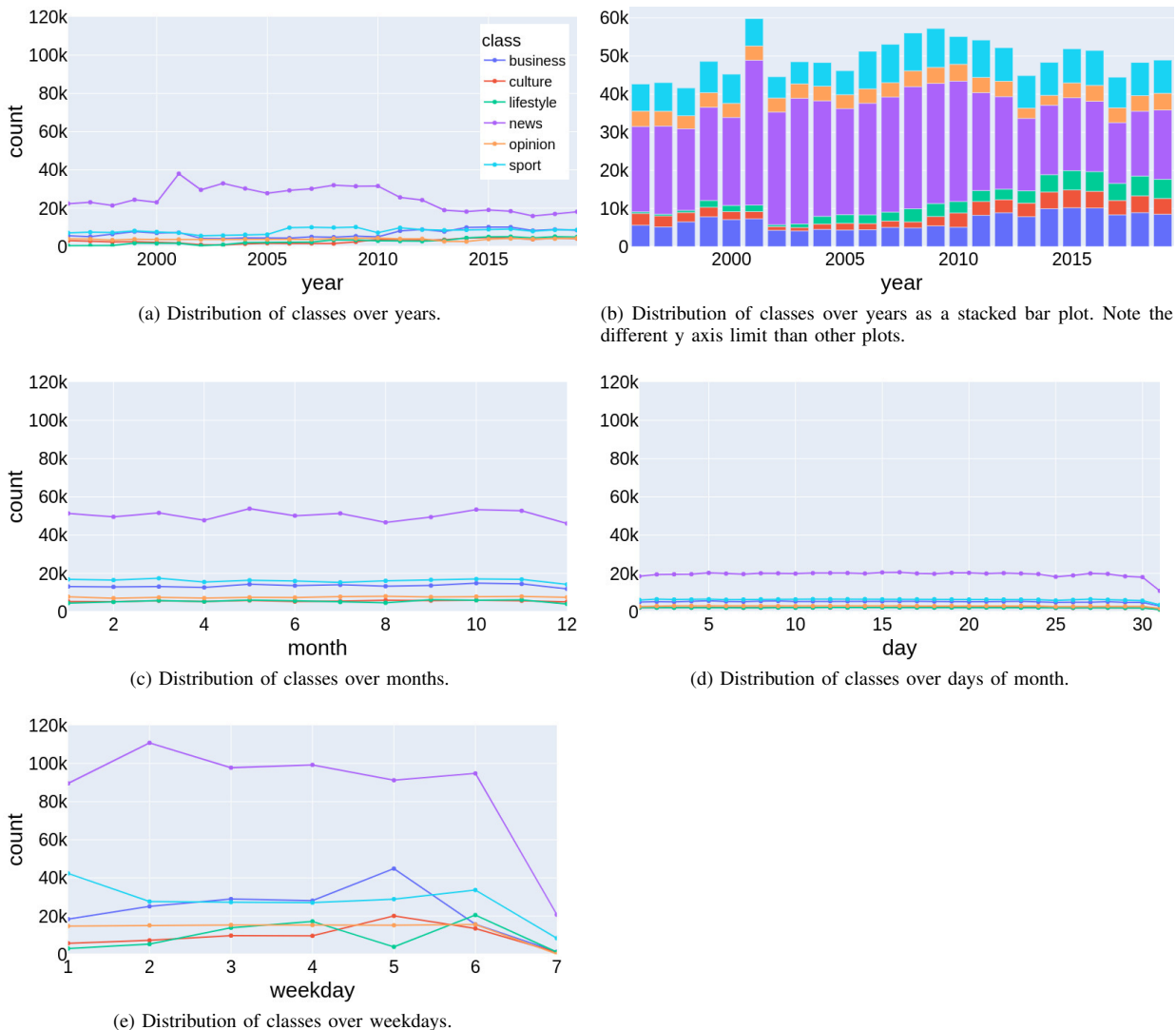


Fig. 2: Distribution of items over date factors in Ireland News dataset.

2) *Date as stacked embeddings*: Temporal embeddings are stacked at the beginning of the input sequence, as:

$$emb = \begin{cases} year_emb & \text{if } token_pos = 1 \\ month_emb & \text{if } token_pos = 2 \\ month_emb & \text{if } token_pos = 3 \\ weekday_emb & \text{if } token_pos = 4 \\ token_emb+ & \\ pos_emb & \text{otherwise} \end{cases}$$

Where all tokens are shifted 4 positions to the right, so first text token is on $token_pos = 5$. We refer to this method as `stackedEmbDate` in result tables.

3) *Date as regular text*: We only modify text input of model by adding temporal information with prefixes, so item with date `20040705` and text `Sudan claims it is disarming militias` is combined to text `year: 2004`

`month: 7 day: 5 weekday: 1 Sudan claims it is disarming militias.`

VI. EXPERIMENTS

A. Classification

We carried out experiments with text classification using all presented models. RoBERTa was finetuned and trained from pretrained checkpoints (which we refer to as `pretrained`) and with randomly initialized weights (which we refer to as `from scratch`). The only training objective is the classification task. We report the results in Table IV.

We examined the impact on classification by each date factor. Since all temporal data incorporation methods yield similar results, we chose the regular text date incorporation method due to ease of its use (only text modification with no architecture changes). The results are presented in Table V. To examine this model conditioned by different prefixes we

TABLE III: Model roberta-pretrained-textDate predictions depending on a given date in a development dataset. If a date is represented by a dash, it is not prefixed to the model, bolded dates are as they occur actually in the dataset, not bolded are random. The examples are cherry-picked. To check article-id visit www.irishtimes.com/article-id The article ID is not provided in the challenge.

text	article ID	timestamp	actual	prediction
New bridge for Calzaghe to cross	1.914946	20080419 Sat.	sport	sport
New bridge for Calzaghe to cross	1.914946	20130307 Thu.	-	life&style
New bridge for Calzaghe to cross	1.914946	-	-	news
Sydney stereotypes	1.1102371	20000913 Wed.	sport	sport
Sydney stereotypes	1.1102371	20110422 Fri.	-	opinion
Sydney stereotypes	1.1102371	-	-	sport
Róisín Meets... comedian Mario Rosenstock	1.2463531	20151212 Sat.	life&style	life&style
Róisín Meets... comedian Mario Rosenstock	1.2463531	20040725 Sun.	-	news
Róisín Meets... comedian Mario Rosenstock	1.2463531	-	-	news

TABLE IV: Classification results. Different date incorporation into model. Acc stands for accuracy. The bold results are best in its category (without and with external data).

method	Ireland News		Sentiment140	
	acc	gonito	acc	gonito
most frequent from train	51.10	{161712}	49.88	{b4b180}
roberta-pretrained-noDate	82.35	{daaaf9}	89.27	{a8d1b7}
roberta-pretrained-stackedEmbDate	87.65	{9e041f}	91.16	{252c0c}
roberta-pretrained-addedEmbdate	86.82	{cede76}	91.04	{aa28dc}
roberta-pretrained-textDate	87.84	{7c52ed}	91.13	{688320}
roberta-scratch-noDate	77.88	{0798d5}	83.38	{e984db}
roberta-scratch-stackedEmbDate	83.24	{74efba}	86.18	{e3ff3e}
roberta-scratch-addedEmbdate	81.96	{587033}	85.47	{1c122b}
roberta-scratch-textDate	83.16	{413f72}	86.02	{d969ca}

TABLE V: Classification accuracy results. Different date elements included. Acc stands for accuracy. MI stands for Mutual Information between a class and a date factor. MI for Sentiment140 between year and class equals 0, because there is only 2009 year in the dataset.

method	Ireland News			Sentiment140		
	Acc	Gonito	MI(1e-5)	Acc	Gonito	MI(1e-3)
roberta-pretrained-noDate	82.35	{daaaf9}	-	89.27	{a8d1b7}	-
roberta-pretrained-textDate	87.84	{7c52ed}	-	91.13	{688320}	-
roberta-pretrained-textDay	82.66	{ca5340}	9	90.16	{2c2d07}	58
roberta-pretrained-textMonth	82.72	{3d5bb6}	61	89.59	{64cc1b}	16
roberta-pretrained-textYear	85.90	{893bbe}	3354	89.32	{be6d55}	0
roberta-pretrained-textWeekday	84.46	{daf69a}	3127	89.60	{8abd71}	19

TABLE VI: Roberta-pretrained-textDate classification on development set result. All results comes from the same model, the only difference is the prefix construction. Prefix is a standard model mode, no-prefix is a mode where no date is prefixed, and random-prefixed stands for a mode, where the date prefix comes from random date 1996-01-01 to 2021-06-30.

model	dev acc
prefix	87.97
no-prefix	78.38
random-prefix	73.97

checked its performance with no prefix and random prefix settings. Results are in Table VI and Table VII. The samples

from different prefix settings are provided in Table IV.

To check model degradation, we made an inference on Ireland News test set from years 2020-2021. This is a time span later than training data, which comes from 1996-2019. The results are in Table VIII.

The impact of train dataset size is presented in Figure 3.

B. Year prediction

We choose two methods for year prediction. The first is a baseline using term frequency-inverse document frequency (TF-IDF) with logistic regression. The second is averaging all output embeddings of RoBERTa and feeding to linear regression (LR) layer. Both RoBERTa and linear regression weights are tuned during training. In both methods, the minimum (maximum) output is limited to the minimum (maximum)

TABLE VII: Classification improvement due to prefixing on roberta-pretrained-textDate model. All results comes from the same model, naming convention comes from Table VI.

dev set percentage	
accurate on both prefix and no-prefix	75.14
accurate on prefix, but not on no-prefix	12.83
accurate on no-prefix, but not on prefix	3.19
not accurate on prefix, nor on no-prefix	9.84

TABLE VIII: Classification accuracy results. Test set (years 2020-2021) comes from other time span than training set (years 1996-2019).

method	Ireland News (2020/21)	
	acc	gonito
most frequent	38.27	{953311}
roberta-pretr.-noDate	85.99	{e684b3}
roberta-pretr.-textDate	87.79	{5fba22}
roberta-pretr.-textYear	87.49	{8d5ad4}

fractional year found in the datasets. The results are presented in Table IX, along with a null-model baseline using the mean fractional year from the training set as the prediction for each data point.

C. Word gap filling

RoBERTa was finetuned and trained from a pretrained checkpoint and with randomly initialized weights. The training objective is Masked Language Modeling. Only prepending data to the input was considered as a method for introducing the data. See Table X.

VII. DISCUSSION

For both datasets including dates into RoBERTa models raises the accuracy score. This stands true for pretrained and randomly initialized models. Stacked embedding and date incorporation as a text give a similar result and both are slightly better than the method of adding embeddings to every input token. It's easier to modify input text than modify model architecture, hence we recommend embedding date by prefixing input texts. The greater mutual information is between each factor and class factor, the more the model gains in accuracy score. The model trained with a date prefix performs well, only when the prefix is provided. There is no gain from date prefixing for a 1k documents train dataset and the gain is constant over 100k documents train dataset. Predicting fractional year is difficult in both datasets because all models perform not much better than baseline. We hypothesize this is a reason why classification benefits from date metadata, since adding strongly correlated factors (like a date to text in this case) would not bring information gain.

The temporal models perform better also for test sets from unseen years. To our surprise, day of the month, month, weekday, year incorporation into model performs only marginally better than incorporation only year for Ireland News 2020-2021 dataset.

In pretrained models, date incorporation slightly lowers perplexity. Models with randomly initialized weights benefit hugely from date incorporation.

VIII. RELATED WORK

There are several studies concerning language model degradation over time and adaptation to newer data [13], [17], [6]. [7] focused especially on text classification. They considered years as well as cyclical intervals (e.g., January-March). Their method was to train separate models for different time spans. [8] proposed method based on using discrete multiple temporal word embeddings based on time domains for document classification using recurrent neural networks. [9] developed model-agnostic timed dependent embedding representation for time and evaluated on recurrent neural networks across various tasks. [1] introduced temporal T5 language model, where a year was prefixed into text input and finetuned on temporal data. The experiments focused on knowledge extraction from language models and showed their method performs better in terms of language modeling and question answering than T5 language model with no prefixed year. [19] incorporated both geographical and time data into a transformer model for a QA task employing year as well as month and day. [16] prefixed year for semantic change detection. Additionally, the authors proposed the training objective of masking year information during model training. However, both [1], [16] use only year metadata, in contrast to our study, where we also days of month, months, weekdays are taken into consideration. [18] trained an SVM model to predict the date of text as a classification problem and [11] use approach of neologism based approach. Very recently [15] released temporal NLP challenges based on a large corpus of historic texts but didn't include downstream tasks, such as classification. The corpus consists of texts covering over 100 years. They trained from scratch and fine-tuned temporal RoBERTa models based on day of month, months, weekdays, and year as a prefixed text. They proved that temporal language models perform better than standard language models.

IX. CONCLUSION

Transformer models benefit from temporal information data in classification tasks for short texts. We have proved that it's not only true for a year, but also other date factors, such as weekday, day of the month, and month. The greater the mutual information between a factor and a class, the greater the benefit. The result is important, because day of the month, month, weekday factors don't outdate after model training

TABLE IX: Fractional year prediction results, RMSE is for root-mean-square error, MAE – mean absolute error, LR – linear regression.

method	Ireland News			Sentiment140		
	RMSE	MAE	Gonito	RMSE	MAE	gonito
mean from train	6.76426	5.80722	{0b0e9c}	0.04674	0.03396	{4856c5}
TF-IDF + LR	5.32491	4.27185	{2226fb}	0.04917	0.03635	{579c8f}
RoBERTa + LR head	4.53676	3.38758	{632b5d}	0.04469	0.03289	{349e5b}
RoBERTa from scratch + LR head	4.51179	3.35951	{be0106}	0.04526	0.03222	{b672ee}

TABLE X: Word gap prediction results. Ppl hashed stands for perplexity hashed.

method	Ireland News		Sentiment140	
	ppl hashed	gonito	ppl hashed	gonito
equal probability	1024.0	{6bd5a8}	1024.0	{3de230}
RoBERTa from scratch	90.8	{9ac479}	51.0	{f0f343}
RoBERTa from scratch with time	46.0	{dc75a7}	46.1	{ddf16f}
RoBERTa no fine-tuning	51.0	{f0f343}	66.2	{e625c6}
RoBERTa fine-tuned	23.3	{42793a}	34.6	{a365da}
RoBERTa fine-tuned with time	21.6	{cfaf6c}	33.6	{37bd6e}

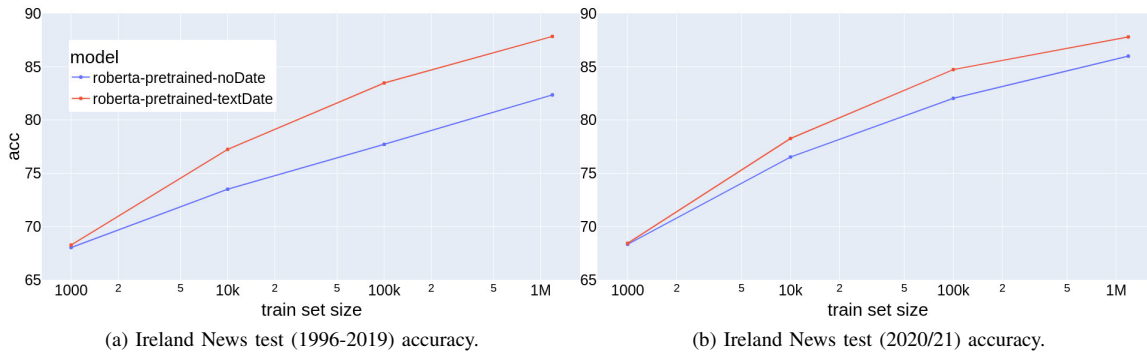


Fig. 3: Test set accuracy varying on train dataset size for model with and without date incorporation.

due to its cyclical nature, differently to year, which is linear. The best and simplest method for temporal data incorporation seems to be input text modification.

REFERENCES

[1] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, and W. W. Cohen. Time-aware language models as temporal knowledge bases, 2021.

[2] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Processing*, 150, 2009.

[3] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierchoń. Gonito.net – open platform for research competition, cooperation and reproducibility. In A. Branco, N. Calzolari, and K. Choukri, editors, *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pages 13–20. 2016.

[4] F. Graliński, A. Wróblewska, T. Stanisławek, K. Grabowski, and T. Górecki. GEval: Tool for debugging NLP datasets and models. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 254–262, Florence, Italy, 2019. Association for Computational Linguistics.

[5] F. Graliński. (Temporal) language models as a competitive challenge. In Z. Vetulani and P. Paroubek, editors, *Proceedings of the 8th Language & Technology Conference*, pages 141–146. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu, 2017.

[6] S. A. Hombaiyah, T. Chen, M. Zhang, M. Bendersky, and M. Najork. Dynamic language models for continuously evolving content. *ArXiv preprint, abs/2106.06297*, 2021.

[7] X. Huang and M. J. Paul. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699, Melbourne, Australia, 2018. Association for Computational Linguistics.

[8] X. Huang and M. J. Paul. Neural temporality adaptation for document classification: Diachronic word embeddings and domain adaptation models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4113–4123, Florence, Italy, 2019. Association for Computational Linguistics.

[9] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. A. Brubaker. Time2vec: Learning a vector representation of time. *ArXiv, abs/1907.05321*, 2019.

[10] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher. Ctrl: A conditional transformer language model for controllable generation. *ArXiv, abs/1909.05858*, 2019.

[11] V. Kulkarni, Y. Tian, P. Dandiwal, and S. Skiena. Simple neologism based domain independent models to predict year of authorship. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 202–212, Santa Fe, New Mexico, USA, 2018. Association for Computational Linguistics.

[12] G. Lample and A. Conneau. Cross-lingual language model pretraining. In *NeurIPS*, 2019.

[13] A. Lazaridou, A. Kuncoro, E. Gribovskaya, D. Agrawal, A. Liska, T. Terzi, M. Gimenez, C. de Masson d’Autume, S. Ruder, D. Yogatama,

- K. Cao, T. Kociský, S. Young, and P. Blunsom. Pitfalls of static language modelling. *ArXiv*, abs/2102.01951, 2021.
- [14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv preprint*, abs/1907.11692, 2019.
- [15] J. Pokrywka, F. Graliński, K. Jassem, K. Kaczmarek, K. Jurkiewicz, and P. Wierzchoń. Challenging America: Modeling language in longer time scales. *Findings of North American Chapter of the Association for Computational Linguistics*, 2022. forthcoming.
- [16] G. D. Rosin, I. Guy, and K. Radinsky. Time masking for temporal language models, 2021.
- [17] P. Röttger and J. Pierrehumbert. Temporal adaptation of BERT and performance on downstream document classification: Insights from social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2400–2412, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.
- [18] T. Szymanski and G. Lynch. UCD : Diachronic text classification with character, word, and syntactic n-grams. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 879–883, Denver, Colorado, 2015. Association for Computational Linguistics.
- [19] M. Zhang and E. Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

3.3 Modeling Spaced Repetition with LSTMs

Modeling Spaced Repetition with LSTMs

Jakub Pokrywka¹^a, Marcin Biedalak², Filip Graliński¹^b and Krzysztof Biedalak²

¹ Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Poland

² SuperMemo World

{jakub.pokrywka, filip.gralinski}@amu.edu.pl, {marcin.biedalak, krzysztof.biedalak}@supermemo.com

Keywords: spaced repetition, LSTM, metalearning

Abstract: Spaced repetition is a human learning technique focused on optimizing time intervals between a student’s repetitions of the same information items. It is designed for the most effective long-term high-retention knowledge acquisition in terms of a student’s time spent on learning. Repetition of an information item is performed when its estimated recall probability falls to the required level. Spaced repetition works particularly well for itemized knowledge in areas requiring high-volume learning like languages, computer science, medicine, etc. In this work, we present a novel machine-learning approach for the prediction of recall probability developed using the massive repetition data collected in the SuperMemo.com learning ecosystem. The method predicts the probability of remembering an item by a student using an LSTM neural network. In our experiments, we observed that applying the spaced repetition research expert algorithms (Woźniak et al., 2005), like imposing the negative exponential function as the output forgetting curve, increases the LSTM model performance. We analyze how this model compares to other machine-learning or expert methods such as the Leitner method, XGBoost, half-life regression, and the spaced repetition expert algorithms. We found out that the choice of evaluation metric is crucial. Furthermore, we elaborate on this topic, finally selecting macro-average MAE and macro-average Likelihood for the primary and secondary evaluation metrics.

1 INTRODUCTION

Spaced repetition is the idea to improve learning process for humans by optimizing the time intervals between which the same material, for instance a word in a foreign language (in general: a repetition item), is presented to the user. E-learning systems based on spaced repetition are used for courses based on a large number of atomic items, for instance in learning foreign (or programming) languages (especially their vocabulary) or acquiring fact-based knowledge.


The idea of spaced-repetition software was pioneered by Piotr Woźniak and SuperMemo with a number of expert algorithms. In this paper, we train a number of machine-learning-based systems, using massive repetition data obtained through the courtesy of SuperMemo.com platform, and compare them against simple baselines and the original spaced-repetition expert algorithms (Woźniak, 1990; Woźniak et al., 1995).


The contributions of this paper are as follows:

1. we propose a methodology for creating future-proof challenges from real-world data for training and testing spaced-repetition systems;
2. we discuss evaluation methodology and give motivation for using a new evaluation metric;
3. we propose a novel approach of a LSTM neural network with exponential decay and compare it to several baselines.

2 SUPERMEMO RESEARCH

SuperMemo is the world pioneer in applying optimized spaced repetition to computer-aided learning (Woźniak, 2018a). The name SuperMemo encompasses the method, software and company. In 1982, Piotr Woźniak, then a student of molecular biology, started experiments which led to the formulation of his first spaced repetition algorithm in 1985. In 1987, he created the first SuperMemo computer program. It applied the so-called SM-2 algorithm (Woźniak, 1990) which was later made public and has been used by other apps, including Anki, ever since. In the following years, Woźniak kept improving his expert

^a <https://orcid.org/0000-0003-3437-6076>

^b <https://orcid.org/0000-0001-8066-4533>

algorithm. Successive versions adapted to the actual memory retention measured individually for each user, thus allowing for truly individualized learning. Independently of this, Woźniak developed his theory of two components of memory (Woźniak et al., 1995), which was fully applied in the SM-17 algorithm in 2016.

While optimizing the machine learning algorithms described in this paper, we successfully used key elements of Woźniak’s research. In order to smooth the recall probability predictions yielded for increasing intervals, we forced the LSTM networks (see Section 8.7) to apply the negatively exponential function which, as proposed by Woźniak, represents the shape of the forgetting curve (Woźniak et al., 2005):

$$R = e^{-kt/S},$$

where:

- t — time,
- R — probability of recall at time t ,
- S — stability expressed by the inter-repetition interval that results in retrievability of 90% (i.e. $R = 0.9$),
- k — constant independent of stability.

To some surprise, it not only matched the original LSTM results but also slightly improved the algorithm metrics.

3 SUPERMEMO.COM ECOSYSTEM AND DATA

For training and testing the machine learning algorithms described in this paper, we obtained repetition data collected by the SuperMemo.com online and apps learning ecosystem. SuperMemo.com features over 250 ready language courses for 23 different languages in the premium version and allows users to learn from user-generated courses for free. SuperMemo.com is often applied by users to learn sciences requiring high-volume learning, including computer science, programming and medicine.

SuperMemo.com courses differentiate between presentation and repetition content. Presentation pages are used for explaining the material learned. When users progress through a course, presentation items are shown once by default. They may include comments, complex texts or even parts of a full feature interactive movie. Repetition items (exercises) contain atomic questions or tests which are then scheduled in repetitions according to the SuperMemo algorithm. While learning languages, these

are typically used to memorize vocabulary and grammar. When learning programming, exercises can be used to master coding rules and patterns (see Figure 1). In general, for optimum review scheduling and learning, repetition items are recommended to meet the *minimum information principle* (Woźniak, 1999) (i.e. should be atomic and as simple as possible).

SuperMemo repetition items typically test active production. Passive knowledge, like developed in multiple choice tests, is considered to be a different, limited competence. Therefore, unlike in other popular e-learning applications which often shuffle the same content along different types of multiple choice tests during a session, SuperMemo exercises are mostly question and answer pairs which are stable in their form. Each exercise, irrespective of whether it is active or passive (see Figure 2), is treated as a separate item with its own learning characteristics and history. Each repetition is rated once on the first contact during a session.

Exercises are rated on a 3-grade scale: *I know*, *Almost*, *Don’t know*. The first two are both positive grades meaning that the information is still remembered, with the difference that answers rated *I know* are not asked again in the same session, while those rated *Almost* can be drilled until they are recalled successfully. Based on the history of repetitions and grades, the SuperMemo algorithm proposes the next repetition for the day when the probability of recall by the user is expected to fall to 90% (see Figure 3).

The SuperMemo algorithm develops and maintains separate memory models for every user and course. Each exercise is scheduled for repetitions so as to statistically reach the expected level of retention.

4 RELATED WORK

Half-life regression is a model of space repetition, a modification of linear/logistic regression taking into account the forgetting curve (Settles and Meeder, 2016). Note that Duolingo, the system for which half-life regression was initially proposed, is based on an approach different from SuperMemo — a gold-standard value does not have to be 0 or 1, it is usually a fraction representing the percentage of successful attempts during a single session. In SuperMemo, a simpler model is assumed (following the minimum information principle), a model that can handle a larger variety of courses.

Deep reinforcement learning have been also applied to the problem of planning spaced repetition (Upadhyay et al., 2018; Yang et al., 2020; Sinha, 2019). The main drawback of reinforcement learning

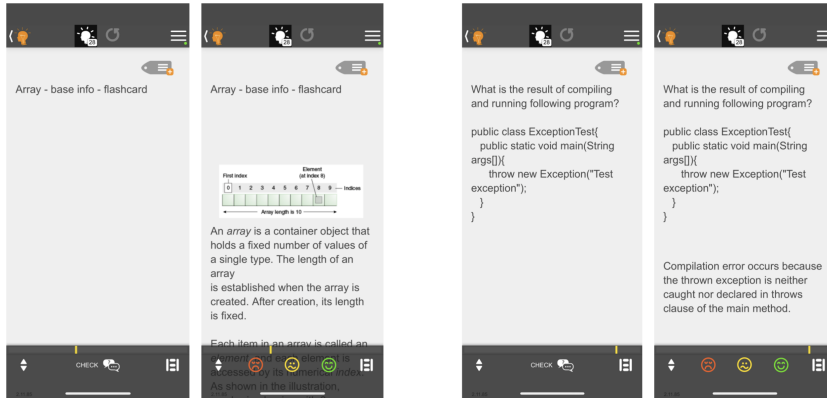


Figure 1: Pages from general computer science and Java 8 programming courses, source: SuperMemo application.

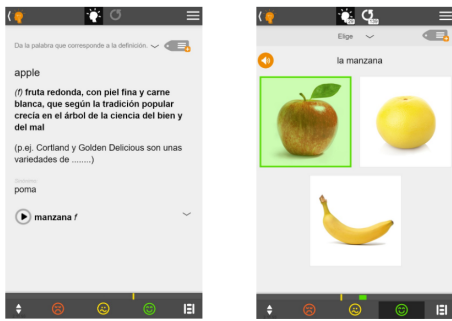


Figure 2: Active and passive exercises in SuperMemo are separate items for repetition scheduling, source: SuperMemo application.

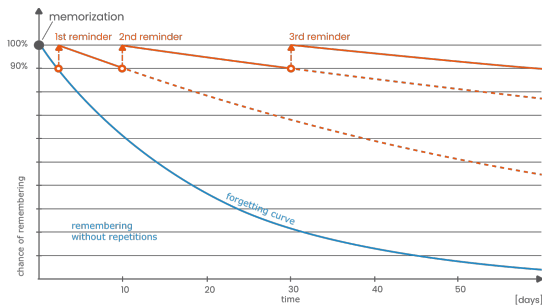


Figure 3: Forgetting curves applied in learning, source: www.supermemo.com

in this context is that it requires simulated environments for training and evaluation. In this paper, we opt for a more practical option of limiting ourselves to the paradigm of supervised learning.

5 DATA SET

The data set is based on real retention data from the SuperMemo application.

5.1 Assumptions

In order to make the challenge harder and to simulate cases when a new user joins the system or a new course is created, the train/dev/test split was prepared in such a way that no user and no course is shared between the data subsets. To be more precise, the MD5 sum is calculated for each user and for each course, and users and courses are (separately) assigned to the training, development and test sets based on its checksums. This assignment has the following consequences:

- the train/dev/test split is pseudo-random, but *stable*, when the splitting script is re-run on a (possible larger data set), no course/user will change assignments,
- some data is “lost”, e.g. a user-course pair for which the user is assigned to the train set and the course to the dev set will not be used,
- ...but on the other hand, we are avoiding unwanted data leakage between users and courses.

In other words, we pose the challenge as *meta-learning* problem. We do not want to learn features for specific users or courses, but rather learn *to learn* to predict retention probabilities even for *new* users (and courses).

For the development and testing sets, a single repetition is selected, again using a stable pseudo-random procedure based on MD5 checksums. All the repetitions up to this one are available, including repetitions related to other words (learning units), but the history *after* the target repetition is removed.

For the training set, simply all the repetitions are given.

The data set was prepared as a challenge on an internal instance of the Gonito platform for tracking evaluation results for machine-learning systems (Graliński et al., 2016).

Table 1: Basic statistics as regards the data set.

dataset	size	recall ratio
train	5757868	95.4%
dev	1152	89.1%
test	1611	88.6%

5.2 Statistics

Data used in this work was collected from users of SuperMemo.com platform where MongoDB is used to store information about every interaction with an item, see Table 1 for basic statistics. We obtained repetition date, previous and next interval set by algorithm, real interval from last repetition, grade. The task is to predict probability for the last grade.

6 EVALUATION METRICS

Selecting the most appropriate evaluation metric for spaced repetitions models is a challenging task. In (Settles and Meeder, 2016), three metrics were considered: mean absolute error (MAE), area under the ROC curve (AUC) and Spearman’s half-life rank correlation. In the case of the SuperMemo learning system, the quality of probabilities is crucial, not just the accuracy of predicted classes (forgotten vs retained), as repetition intervals are directly based on probability thresholds (which can be customized by the user). Hence, we discarded Spearman’s rank and AUC.

Apart from MAE, we measured the quality of probabilities using the *likelihood* metric, which is the geometric mean of probabilities assigned to the correct class by the model. This is a variant of log loss (if log loss is L , then likelihood is $1/e^L$), just made slightly easier to interpret for humans.

Contrary to MAE, likelihood (and log loss) are obviously highly sensitive to overconfident results. It takes one example in which 0 was returned as the probability for the positive class, or 1 for the negative one, for the likelihood metric to collapse to 0. Therefore, it is rational to assume some ϵ and return at least ϵ for an example with the presumably negative class and $1 - \epsilon$ for an example with the positive one.

Another problem is that there is a significant imbalance in the training and testing sets — most samples (around 90 %) belong to the positive class (a user retained a given unit in the memory) and a simple null-model baseline (return 1.0 for MAE and $1 - \epsilon$ for Likelihood) can lead to nominally high and hard-to-beat evaluation results. The approach we chose was to use the macro-average version of the metrics, i.e. the

evaluation metric is calculated separately for the negative and the positive class and then averaged. This way, we attach the same significance to both classes, no matter their numbers.

Finally, we selected macro-avg MAE as the main evaluation metric and macro-avg likelihood as the secondary metric (as implemented as `MacroAvg/MAE` and `MacroAvg/Likelihood` in the GEval evaluation tool (Graliński et al., 2019)). Our decision was confirmed by the fact that all reasonable methods we tested beat the simple null-model baseline if macro-avg MAE was used.

Note that for MAE the lower results, the better, for Likelihood — the other way round.

For a different approach for the evaluation of spaced-repetition systems, based on the idea of *algorithm contest*, see (Woźniak, 2018b).

7 EXISTING NON-ML METHODS

7.1 Null-Model Baseline

This is a simple baseline. During the inference, we just always return probability 0.89, i.e. the mean from training set for all samples.

7.2 Leitner

Leitner System is one of the simplest spaced repetitions methods, mainly used in flashcards (Leitner, 1999). The main idea is to repeat item on the next day after learning it, if the user recalls information correctly the interval is doubled. If not, the interval should be divided by 2 or set to 1.

7.3 SuperMemo Open-Source Algorithm SM-2

The first computer-based SuperMemo algorithm (Woźniak, 1990) which, for every item, tracks the number of times it has been successfully recalled and the interval (i.e. the number of days since the item has been repeated). For each review (attempt by the user to recall the item), the algorithm recalculates easiness factor (EF) based on the self-evaluated grade and sets the date for the next repetition. In this work, we recalculate probability by taking interval from algorithm and comparing it against the forgetting curve.

7.4 SuperMemo Expert Algorithm SM-17

The SM-17 algorithm, developed in the years 2014-16, applies the two component model of memory (Woźniak et al., 1995). Starting from a common memory model, SM-17 then stores and updates the DSR (difficulty, stability, retrievability) matrices with parameters individual for each user. Hill-climbing algorithms are used to find the best estimation of an expected forgetting curve.¹

8 MACHINE LEARNING METHODS

We used the following machine learning methods: logistic regression, feed-forward neural network, half-life regression, gradient boosting trees, and recurrent neural network (RNN). Some of them incorporate exponential decay from the forgetting curve assumption. This may help in training, but more importantly, it does not lead to a counter-intuitive result when the probability of recalling an item increases with time when a student does not study it. For all methods, we take the maximum item history sequence of 40 most recent repetitions. This is necessary for all methods, but RNN. Due to easier batching during RNN training, we also fixed the maximum sequence length to 40. If the item history sequence is shorter than 40, we fill values with -1 , unless stated otherwise in the method description. We always set the likelihood to $\max(\min(1 - \varepsilon, \hat{y}), \varepsilon)$, where $\varepsilon = 0.05$. If we did not, in the case when, e.g., model output is 0.0, and golden truth is 1 for even one item, the likelihood metric is always reduced to 0 for the whole data set. All methods were implemented in PyTorch (Paszke et al., 2019), except gradient boosting trees, where we used XGBoost library (Chen and Guestrin, 2016).

8.1 Logistic Regression

Logistic regression is a simple but effective machine learning method. In our case, it serves as a baseline machine learning method due to ease of training.

8.2 Feed Forward Neural Network

For a simple feed-forward neural network, we implemented a two-layer network with 4 hidden neurons, a

¹See https://supermemo.guru/wiki/Algorithm_SM-17 for the detailed description.

ReLU activation function in between, and a sigmoid activation function on top.

8.3 Half-Life Regression

Half-life regression (HLR) is the Duolingo method described in (Settles and Meeder, 2016). It is similar to logistic regression but imposes exponential decay of the forgetting curve.

It is based on an assumption of probability of recalling an item from memory is

$$p = 2^{-\frac{\Delta}{h}} \quad (1)$$

Where Δ is lag time (time in days elapsed from the last time the course item was reviewed), h is the half-life, which is the measure of the strength of students' memory of the course item.

Half-life is estimated:

$$\hat{h}_{\Theta} = 2^{\Theta \cdot x}, \quad (2)$$

where x are variables related to student course and item history and Θ are model parameters.

Thus, the estimated probability of recalling a word is:

$$\hat{p}_{\Theta} = 2^{-\frac{\Delta}{2^{\Theta \cdot x}}} \quad (3)$$

During HLR training, the following loss function ℓ is optimized:

$$\ell(\langle p, \Delta, x \rangle, \Theta) = (p - \hat{p}_{\Theta})^2 + \alpha \left(\frac{-\Delta}{\log_2(p)} - \hat{h}_{\Theta} \right) + \lambda \|\Theta\|_2^2, \quad (4)$$

where α and λ are hyperparameters.

This loss function optimizes not only \hat{p}_{Θ} , but \hat{h}_{Θ} as well. The $\lambda \|\Theta\|_2^2$ is model weights L_2 regularization. Optimizing \hat{h}_{Θ} was found to improve loss in the original Duolingo paper, but not on the SuperMemo data set. Finally, we employed the following loss:

$$\ell(\langle p, \Delta, x \rangle, \Theta) = (p - \hat{p}_{\Theta})^2 + \lambda \|\Theta\|_2^2 \quad (5)$$

We implemented this method with some slight adjustments for the SuperMemo data set. The main difference is that the SuperMemo data set allows only binary expected values (0 for not remembering in the first attempt in the session, 1 for remembering in the first attempt in the session). This is contrary to Duolingo data set, which allows continuous value based on the attempt number of remembering during the session. Besides, we slightly changed the minimum and maximum boundaries of duration elapsed

from the last word seen. It means that the word was last seen below 1 day; we set it to 1 day. If the word was last seen above 7 years, we set it to 7 years. This is due to numerical stability because exponential decay assumptions cause floating point overflow in some cases.

8.4 Standard Gradient Boosting Trees

Gradient boosting tree methods usually perform well when it comes down to tabular data. In our experiments, we employed XGboost (Chen and Guestrin, 2016) and used logistic regression as a loss function. The main advantage of this method is its ease of use and good performance out of the box. However, manual search for better than default hyperparameters did not yield significantly better results, so we keep them default.

8.5 Gradient Boosting Trees with Exponential Decay

XGBoost with logistic regression function does not ensure exponential decay assumption. We model the recall probability as

$$\hat{p} = e^{\frac{-\Delta}{o(x)}}, \quad (6)$$

where $o(x)$ is output of XGBoost model. Although, it would be technically difficult to employ this assumption into XGBoost and optimize p directly.

Instead, our approach was to optimize o using the formula:

$$o = \frac{-\Delta}{\ln(p)}. \quad (7)$$

Due to equation 7 indeterminacy, when $p = 0$ or $p = 1$, in practice we employed:

$$o = \frac{-\Delta}{\ln(\min(1 - \epsilon, \max(\epsilon, p)))}, \quad (8)$$

where $\epsilon = 0.05$.

During inference, recall probability is obtained with equation 6.

This approach does not require XGBoost model architecture modification but only target and prediction transformation. In this method, we used default XGBoost hyperparameters as well.

8.6 Standard RNN

RNN often yields good results in dealing with time series. RNN can take a sequence of any length as input, so its perfect for students learning history.

We implemented 1-layer Long short-term memory (LSTM) ((Hochreiter and Schmidhuber, 1997)) with 256 cell units and trained with MSELoss. During training, we set students learning history to a fixed sequence length of 40 for optimal batching.

8.7 RNN with Exponential Decay

In order to impose exponential decay, we model the probability of item recalling as

$$\hat{p}(x) = e^{\frac{-\Delta}{o(x)}}, \quad (9)$$

where $o(x)$ is output from the LSTM and Δ is lag time. Due to the floating point overflow of $e^{o(x)}$, we set the maximum lag time to 3 years instead of 7 years.

For missing values, we set the probability of remembering to 1 and the maximum lag-time, which is 7 years.

9 RESULTS

Due to instability of half-life regression, RNN, and RNN with exponential decay training, we trained the models 10 times and averaged the results.

The results for all described methods are presented in Table 2. The best performing method in the primary metric MacroAvgMAE is RNN with exponential decay, and the best performing method in the secondary metric MacroAvgLikelihood is the feed-forward neural network. Standard XGBoost model surpasses all other methods in not macro averaged metrics, both MAE and Likelihood. In terms of MacroAvgMAE imposing exponential decay helped achieve RNN model better results but worsened XGBoost score. The second best-performing method is SM17, which is an expert algorithm.

10 VERIFICATION ON SYNTHETIC TEST CASES

We prepared a small synthetic data set to verify the results in 8 different cases of user learning history. Each case consists of first student contact with an item and 3 consecutive recalls after some intervals and with relevant grades (*I know, Almost, Don't know*). After this, we check the probability of recalling an item after 10, 20, 30, 100, 1000 days intervals as returned by a given model; see results in 4.

After a manual inspection on this data set, we concluded:

Table 2: Results of ml and non-ml methods on the SuperMemo.com dataset. Bolded text indicates the best result in the given metric. MacroAvgMAE and MacroAvgLikelihood are primary and secondary metrics.

Method	Likelihood \uparrow	MAE \downarrow	MacroAvgLikelihood \uparrow	MacroAvgMAE \downarrow
mean from train	0.703 \pm 0.027	0.195 \pm 0.014	0.500 \pm 0.000	0.500 \pm 0.000
Leitner system	0.076 \pm 0.022	0.526 \pm 0.014	0.232 \pm 0.032	0.487 \pm 0.028
SM-2	0.143 \pm 0.040	0.411 \pm 0.014	0.269 \pm 0.038	0.427 \pm 0.025
SM-17	0.614 \pm 0.029	0.220 \pm 0.012	0.452 \pm 0.023	0.390 \pm 0.024
logistic regression	0.739 \pm 0.024	0.159 \pm 0.013	0.526 \pm 0.009	0.444 \pm 0.010
ff neural network	0.734 \pm 0.018	0.206 \pm 0.009	0.539\pm0.014	0.435 \pm 0.010
half-life regression	0.680 \pm 0.036	0.164 \pm 0.016	0.492 \pm 0.003	0.500 \pm 0.005
standard XGBoost	0.758\pm0.021	0.158\pm0.010	0.543 \pm 0.011	0.421 \pm 0.011
XGBoost with exp decay	0.715 \pm 0.025	0.196 \pm 0.013	0.509 \pm 0.002	0.488 \pm 0.003
standard RNN	0.744 \pm 0.022	0.163 \pm 0.012	0.531 \pm 0.009	0.440 \pm 0.010
RNN with exp decay	0.527 \pm 0.015	0.362 \pm 0.012	0.504 \pm 0.031	0.376\pm0.023

- XGBoost indicates high recall probability in all cases, even after 3 consecutive recall fails (case 2) and a long interval of 1000 days, which is not in accord with common sense. This model cannot be useful for real-world application, even though it achieved a good MacroAvgMAE result of 0.421, which also leads to the conclusion that automated metrics do not always reflect expectations,
- standard XGBoost learned decay of recalling probability; even though we did not impose it directly into the model,
- standard LSTM didn't learn decay of recalling probability (e.g. case 2) on its own,
- SM17, half-life regression and LSTM models with exponential decay behave as expected.

11 CONCLUSION

Herein, we compared spaced repetition algorithms based on neural networks (LSTMs) with simpler machine learning approaches and existing expert algorithms. In general, methods based on machine learning yielded promising results (with the best result according to our main evaluation metric obtained by an LSTM). Still, some caveats need to be expressed:

- machine-learning methods, including LSTMs, might give good results as measured with an evaluation metric, but still behaving in an impractical manner and breaking natural assumptions (e.g. probability of retention not decreasing even for very long intervals or even higher probabilities of retention for longer intervals),
- this can be alleviated with modifications transplanted from expert methods (e.g. forgetting curve as proposed by Woźniak),

- LSTM is susceptible to large variance and, in practical terms, is more complicated to use than expert methods,
- the ranking of methods depends heavily on the evaluation metric chosen, we claim the evaluation method we called MacroAvgMAE is the most reasonable, but still it is far from obvious how this relates to quality of learning, when a given method is embedded within a real learning application.

One area of improvement for the method based on LSTMs is to equip it with a mechanism to adapt for a specific user/course, just the way expert methods such as SM-17 do.

REFERENCES

- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.
- Graliński, F., Jaworski, R., Borchmann, Ł., and Wierchoń, P. (2016). Gonito.net – open platform for research competition, cooperation and reproducibility. In Branco, A., Calzolari, N., and Choukri, K., editors, *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pages 13–20.
- Graliński, F., Wróblewska, A., Stanisławek, T., Grabowski, K., and Górecki, T. (2019). GEval: Tool for debugging NLP datasets and models. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 254–262, Florence, Italy. Association for Computational Linguistics.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

	interval 4 [days]	standard XGBoost	predicted recall probability			
			SM17	half-life regression	standard RNN	RNN with exponential decay
case 1	10	0.96646112	0.98395239	0.98968891	0.94910389	0.99243098
interval after repetition 0-3 [days]	-	0.96660864	0.96816230	0.97948413	0.95318353	0.98491937
grade	5 5 5 5	0.96660864	0.95262560	0.96938458	0.95326728	0.97746462
	100	0.96660864	0.82322126	0.90154421	0.93841207	0.92683727
	1000	0.96635157	0.14294395	0.35470742	0.93588412	0.46777284
case 2	3	0.77914268	0.72900000	0.99688917	0.61612576	0.15706201
interval after repetition 0-3 [days]	-	0.77914268	0.34867844	0.98966815	0.51074898	0.00209044
grade	5 0 0 0	0.74191707	0.04239116	0.96932358	0.29470825	0.00000001
	100	0.72425246	0.00500000	0.90135512	0.13169448	0.00000000
	1000	0.72425246	0.00500000	0.35396419	0.46603218	0.00000000
case 3	10	0.96371633	0.98395239	0.98969721	0.93071842	0.99264401
interval after repetition 0-3 [days]	-	0.96387553	0.96816230	0.97950056	0.92755097	0.98534209
grade	0 5 5 5	0.96387553	0.95262560	0.96940897	0.92252403	0.97809392
	100	0.96387553	0.82322126	0.90161980	0.90989619	0.92882788
	1000	0.96359819	0.14294395	0.35500495	0.92156214	0.47791722
case 4	10	0.96515268	0.98395239	0.98968486	0.94910389	0.99243098
interval after repetition 0-3 [days]	-	0.96530581	0.96816230	0.97947613	0.95318353	0.98491937
grade	0 3 3 5	0.96530581	0.95262560	0.96937270	0.95326728	0.97746462
	100	0.96530581	0.82322126	0.90150737	0.93841207	0.92683727
	1000	0.96503896	0.14294395	0.35456251	0.93588412	0.46777284
case 5	3	0.82741082	0.72900000	0.99643895	0.77547282	0.19081855
interval after repetition 0-3 [days]	-	0.82741082	0.12157665	0.97649787	0.50620383	0.00001600
grade	5 5 0 0	0.79619277	0.04239116	0.96495476	0.39768663	0.00000006
	100	0.78114730	0.00500000	0.88788457	0.20852648	0.00000000
	1000	0.78114730	0.00500000	0.30448444	0.53229856	0.00000000
case 6	10	0.85279322	0.34867844	0.99029090	0.63909853	0.00507342
interval after repetition 0-3 [days]	-	0.85336500	0.12157665	0.98067606	0.45910034	0.00002574
grade	5 0 5 0	0.82585347	0.04239116	0.97115458	0.32612234	0.00000013
	100	0.81248188	0.00500000	0.90704298	0.16186571	0.00000000
	1000	0.81248188	0.00500000	0.37694561	0.65122515	0.00000000
case 7	10	0.96342510	0.98830385	0.99029309	0.91053045	0.98890287
interval after repetition 0-3 [days]	-	0.96358556	0.97674450	0.98068040	0.88104337	0.97792882
grade	5 0 5 5	0.96358556	0.96532035	0.97116103	0.84547615	0.96707666
	100	0.96358556	0.88900636	0.90706307	0.79662400	0.89440936
	1000	0.96330607	0.30835335	0.37702910	0.89024919	0.32761469
case 8	10	0.96646112	0.99323312	0.99237540	0.93407595	0.99398333
interval after repetition 0-3 [days]	-	0.96660864	0.98651204	0.98480893	0.93955332	0.98800296
grade	5 5 5 5	0.96660864	0.97983643	0.97730016	0.93952096	0.98205853
	100	0.96660864	0.93435507	0.92631755	0.96088064	0.94143713
	1000	0.96635157	0.50713017	0.46515633	0.95910031	0.54690665

Figure 4: Small synthetic data set for manual verification of the models. In the cases description, grade 0 stands for not recalling, 3 stands for almost recalling, 5 stands for recalling.

Leitner, S. (1999). *So lernt man lernen angewandte Lernpsychologie - ein Weg zum Erfolg*. Weltbild Verlag.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Settles, B. and Meeder, B. (2016). A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*, pages 1848–1858.

Sinha, S. (2019). Using deep reinforcement learning for personalizing review sessions on e-learning platforms with spaced repetition.

Upadhyay, U., De, A., and Gomez Rodriguez, M. (2018). Deep reinforcement learning of marked temporal point processes. *Advances in Neural Information Processing Systems*, 31.

Woźniak, P. A., Gorzelańczyk, E. J., and Murakowski, J. A. (1995). Two components of long-term memory. *Acta neurobiologiae experimentalis*, 55(4):301–305.

Woźniak, P. (1990). Optimization of learning. Master's thesis, University of Technology in Poznan. See also <https://www.supermemo.com/en/archives1990-2015/english/ol/sm2>.

Woźniak, P. (1999). Effective learning: Twenty rules of formulating knowledge. <https://www.supermemo.com/en/archives1990-2015/articles/20rules>. Accessed: 2022-05-09.

Woźniak, P. (2018a). The true history of spaced repetition. <https://www.supermemo.com/en/articles/history>. Accessed: 2022-05-09.

Woźniak, P. (2018b). Universal metric for cross-comparison of spaced repetition algorithms. https://supermemo.guru/wiki/Universal_metric_for_cross-comparison_of_spaced_repetition_algorithms. Accessed: 2022-05-09.

Woźniak, P. A., Gorzelańczyk, E. J., and A., M. J. (2005). Building memory stability through rehearsal. <https://www.supermemo.com/en/archives1990-2015/articles/stability>. Accessed: 2022-05-09.

Yang, Z., Shen, J., Liu, Y., Yang, Y., Zhang, W., and Yu, Y. (2020). Tads: learning time-aware scheduling policy with dyna-style planning for spaced repetition. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1917–1920.

3.4 Using Transformer models for gender attribution in Polish

Using Transformer models for gender attribution in Polish

Karol Kaczmarek
 Adam Mickiewicz University,
 Faculty of Mathematics and Computer Science,

Applica.ai Sp. z o.o.
 Email: karol.kaczmarek@amu.edu.pl

Jakub Pokrywka, Filip Graliński
 Adam Mickiewicz University,
 Faculty of Mathematics and Computer Science,
 Uniwersytetu Poznańskiego 4,
 61-614 Poznań, Poland
 Email: {firstname.lastname}@amu.edu.pl

Abstract—Gender identification is the task of predicting the gender of an author of a given text. Some languages, including Polish, exhibit gender-revealing syntactic expression. In this paper, we investigate machine learning methods for gender identification in Polish. For the evaluation, we use large (780M words) corpus "He Said She Said", created by grepping (for author's gender identification) gender-revealing syntactic expressions and normalizing all these expressions to masculine form (for preventing classifiers from using syntactic features). In this work, we evaluate TF-IDF based, fastText, LSTM and RoBERTa models, differentiating self-contained and non-self-contained approaches. We also provide a human baseline. We report large improvements using pre-trained RoBERTa models and discuss the possible contamination of test data for the best pre-trained model.

I. INTRODUCTION

The task of *gender identification or attribution* consists in predicting the gender of an author of a given text. As such, it is an example of text classification, is usually tackled using supervised machine learning, and is relatively popular in the NLP community. Some recent example of experiments in automatic gender identification for various languages are: [17], [27], [2], [14]. For a critical analysis of gender detection systems and their limitations, see [18].

Collections of gender-labeled texts are required if a system based on supervised machine learning is to be trained. The usual approach is to use metadata such as information on authors (of books, papers, social media posts, etc.). Interestingly, some languages exhibit gender-revealing first-person expressions (cf. *soy polaco* vs *soy polaca* in Spanish), and such expressions can be used to automatically label texts as written by a male or female in order to create a data set. This approach (*distant supervised learning*, [21]) is similar to using emoticons for sentiment analysis tasks [23], [9].

Some languages (e.g. Slavic languages) are more amenable to this distant supervised approach than others (e.g. English or Chinese). The approach was applied to Polish to create a large collection of texts, the "He Said She Said" (HSSS) corpus [10]. In this paper, we (1) re-state the original challenge as a classification task with a probability-based evaluation metric, (2) report on large improvements on the gender detection task using pre-trained RoBERTa models, and (3) discuss the

TABLE I
 THE "HE SAID SHE SAID" CHALLENGE IN NUMBERS.

		characters	words	items
train	total	1,240,131,217	177,428,897	3,601,424
	male	628,793,876	89,795,752	1,800,712
	female	611,337,341	87,633,145	1,800,712
dev-0	total	51,080,450	7,158,683	137,314
	male	26,066,897	3,641,716	68,657
	female	25,013,553	3,516,967	68,657
dev-1	total	51,009,045	7,275,691	156,606
	male	25,579,703	3,641,568	78,303
	female	25,429,342	3,634,123	78,303
test-A	total	43,597,629	6,234,069	134,618
	male	22,253,841	3,175,881	67,309
	female	21,343,788	3,058,188	67,309

possible contamination of test data with the data on which RoBERTa models were trained.

In Section II, we discuss the HSSS challenge along with the modifications in the data set done for the purposes of this paper. In the main Section III, we discuss the methods we applied to tackle the challenge of gender identification. Section IV summarizes the results. Finally, we discuss the issues of training/testing data contamination in Section V.

II. HE SAID SHE SAID TASK

Polish is one of the languages with a high frequency of gender-specific first-person expressions. (Only the few languages with gender distinction in the first person, e.g. Ngala [24], might have a higher frequency of such expressions.) This fact was leveraged to create a large gender-labeled corpus for Polish: the "He Said She Said" corpus [10]. Simply CommonCrawl dataset was grepped, using morphological dictionaries and handcrafted rules, for gender-specific first-person expressions. Obviously, there were some issues that needed to be addressed, e.g. quotes, titles, SEO spam.

Later, the corpus was turned into a classification challenge hosted at the Gonito.net platform [11]. All feminine gender-

specific first-person expressions were changed to masculine forms in order to prevent classifiers from using the simple gender-revealing syntactic features. Obviously, without this normalization step, the challenge would be trivial. The corpus was randomly split into 4 sets: train set, two development (validation) sets (dev-0 and dev-1) and test set (test-A). The split was based on the websites from which the texts originated, i.e. texts from the same website would belong to the same set. Also, the sets were balanced so that 50%/50% distribution would be obtained, not just for the whole data set, but also for *each* website. For instance, let's consider a message board about pregnancy, in general, there are many more texts written by women there (at least judging by gender-marked first-person expressions), but for the challenge, the same number of male and female texts would be sampled from such a website. This, along with the fact that texts are short, makes the challenge rather difficult.

The challenge was presented [11] to showcase the Gono.net platform and was discussed there only briefly. For more detailed information about the challenge, see Table I.

For this paper, two changes have been made to the original challenge:

- 1) *Likelihood* metric was chosen as the main metric (instead of simple accuracy), Likelihood is defined as the geometric mean of probabilities assigned to the gold-standard classes – the motivation was that accuracy is not enough to distinguish solutions of varying quality and confidence;
- 2) some unwanted blank characters were removed.

Some initial experiments with learning classifiers based on the HSSS data set were presented in [12].

III. METHODS

We introduce the structure of our experiments as follows. Subsection III-A describes human baselines. Subsection III-B describes TF-IDF (term frequency-inverse document frequency) based methods. Subsection III-C describes some neural methods. Both III-B and III-C are self-contained. This means not including any data apart from training data available in HSSS task. Subsection III-D describes pre-trained transformer models. Table III presents all classifiers results.

- self-contained – we use only data available from the HSSS task: train on the training set, validate on the dev-0 (validation) set and report results on the test-A (test) set. We will use 256 sequence length which covers most (over 90%) of the HSSS data to speed up the training process.
- non-self-contained – we use publicly available models, which were pre-trained on large amounts of data (may be contaminated by examples from the test or validation set). We will use the sequence length that was saved for these models, which is usually 512.

TABLE II
RESULTS ON THE TEST SET SAMPLE OF SIZE 800 CREATED FOR HUMAN EVALUATION.

method	test accuracy
TF-IDF + logistic regression	0.68500
Polish RoBERTa base	0.77125
LSTM (constrained)	0.73375
human 1	0.65250
human 2	0.67375
human 3	0.66250
human 4	0.65625
human ensemble	0.68125

A. Human Baseline

Four people (two females and two males) made predictions for random sample sets of size 200 for development set and 800 for the test set. They were explained how the dataset was created and asked not to look for the answer on the internet. We rejected human 1 result based on the development dataset result and created a human ensemble with the remaining 3 people predictions using majority voting. The results are presented with the best TF-IDF based, self-contained and overall methods in the Table II.

B. TF-IDF based methods

Term frequency-inverse document frequency (TF-IDF) is a common vector representation of a document in natural language processing. We use the TfidfVectorizer library from Scikit-learn with standard parameters. This includes word-level, lowercasing, *l2* normalization. We did not restrict the vocabulary size and we used word-level splitting. The following classifiers were trained using TF-IDF vectors: Logistic Regression, XGBoost Classifier, SVM.

1) *Logistic Regression*: We used LogisticRegression from Scikit-learn library with standard parameters, except for the maximum number of iteration. We trained until classifier convergence.

2) *Support Vector Machine Classifier*: Support-Vector Network [5] is a common algorithm, that circumvents non-linear separability of data as well as separate samples from different categories. Although, in this case, we chose LinearSVC from Scikit-Learn, which uses a linear kernel. The reason is memory and computation issues related to the high dimension of TF-IDF representation and the number of samples in the HSSS task. Again, we used standard parameters, except for no maximum number of iteration, which led to convergence. We do not report likelihood due to the fact that SVM does not yield probabilities.

3) *XGBoost Classifier*: Tree boosting is an effective and popular method for regression and classification. We used XGboost library [3] with the choice of the parameters suited for better classifier quality.¹ This includes gbtrees booster,

¹Some of the parameters were taken from <https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard>

learning rate set to 0.05 and max depth set to 3.

C. Neural Methods (*self-contained*)

1) *FastText*: FastText [15] is a shallow neural network library created for fast text classification model training and evaluation. We used a supervised setting with hyperparameter tuning, the word embeddings were initialized randomly. The best result was obtained with wordNgrams set to 2, word dimension set to 156, and context size window set to 5.

2) *LSTM*: Long Short Term Memory Networks [13] were used to obtain a state-of-the-art results on most NLP tasks before the era of Transformer language models [7]. In our tasks, for bidirectional LSTM, SentencePiece [19] tokenization performs better than word-level lowercase tokenization. Vocab size 50k was used with randomly initialized embeddings of size 100. We tried embedding size 300, but resulted in slightly worse classifier quality. We used one layer of 256 units, trained with Adam [16] optimizer with learning rate 0.001. The batch size used for training was 400 and sequences were trimmed and padded to 256 tokens.

3) *Transformer*: In the last time Transformer [26] and its modification like BERT [7], RoBERTa [20] or XLM-R [4] achieve state-of-the-art in the benchmarks such as GLUE [29] or SuperGLUE [28] benchmark. Most often used bidirectional Transformers are pre-trained on huge amounts of monolingual data in the Masked Language Model (MLM) process, where the model learns a bidirectional representation of tokens. Next, pre-trained models are finetuned to the specific task. This process reduces the time to train a new model from scratch and can be easily adapted to other tasks. In our case, the downstream task is classification, where the model uses a special token ([CLS], classification token), which represents the whole sentence and helps achieve better results.

We train self-contained classifier based on the RoBERTa model in two ways: with pre-training and without pre-training (train classifier from the scratch) stage. We only used the data that was available in the HSSS challenge to avoid any data leaks in the other data sets. To compare our methods we created Transformer with 8 layers, 8 heads, 256 sequence length and embedding size 512 and 2048 respectively for internal model representation and feed forward layer (after attention layer). We use 50k size vocabulary with Sentencepiece tokenization and randomly initialized embeddings of size 512. First, the model was pre-trained for 10 epochs with Masked Language Model (MLM) criterion and finetuned 10 epochs for the classification tasks. Second, the model was trained on the classification task for 20 epochs (comparing to the previous one, where it was 10 + 10 epochs for pre-training and classification) only. We pre-train and finetune with Adam optimizer with learning rate 0.0001 and 50 sentences per batch. Scores presented in the Table III show that the pre-training stage is the important element to achieve a better model for classification tasks.

D. Pre-trained Transformers

In this section we describe fine-tuning of models publicly available for Polish language: Polish RoBERTa [6] and multi-

lingual XLM-R [4] (which supports 100 languages including Polish). Both models are available in the two versions: base (with 12 layers) and large (with 24 layers). Monolingual models like RoBERTa are focused on achieving the best results in a given language. On the other hand, multilingual models support as many languages as possible with results similar to monolingual models. The disadvantage of multilingual models is the size of the vocabulary, which is several times larger than monolingual models like Polish RoBERTa. Bigger vocabulary needs more resources to fine-tune models, but may improve results by cross-language relationships.

1) *Polish RoBERTa finetuning*: We finetuned Polish RoBERTa [6] (base and large model) using fairseq library [22] for 5 and 3 epochs respectively for the base and large model. Further training resulted in lower development dataset accuracy. We used Adam optimizer with a learning rate 0.00001 and around 200k warmup steps. The maximum sequence we use is 512 as in original Polish RoBERTa.

2) *Polish RoBERTa finetuning with Monte-Carlo model averaging*: Common practice when using dropout is to scale weights during inference time. However, as described in [25] (section 7.5), further investigated in [8], this procedure is only an approximation of Monte-Carlo model averaging. We checked, whether the Monte-Carlo model averaging yields better results than standard weight scaling in our case. By setting Polish RoBERTa (both base and large) in the training mode (with active dropout), making predictions 12 times, and averaging likelihood, we obtained slightly better results in both cases.

3) *XLM-R finetuning*: We finetuned multilingual XLM-R [4] base and large for 1 epoch, further training does not improve results. Each of the models was trained with 512 tokens using Adam optimizer with a learning rate 0.00004. Batch size has been set to 10 and 25 for the base and large model. Results are available in the Table III.

4) *Polish RoBERTa last layer averaged*: For the evaluation of how much information about language Polish RoBERTa possesses, we conducted the following experiment. We extracted the last layer tokens and averaged them. Then, we trained logistic regression classifier with no Polish RoBERTa finetuning. This was done until classifier convergence.

5) *XLM-R last layer averaged*: We conducted the same experiment with XLM-R as in subsection III-D4.

6) *Polish RoBERTa fill mask*: In order to check the predicting power of only pre-trained Polish RoBERTa models, we conducted the following experiment. We masked all gender-revealing first-person expression and used the models in Masked Language Model setting. We choose one random expression and looked for the most probable word indicating gender in the first 10 model predictions. Only 6333 samples out of 137314 in the test set did not reveal first-person expression in the first 10 predictions. No training or development sets were used in this experiment. However, this method does not yield good results (though the trivial baseline was beaten).

IV. RESULTS

The self-contained models (BiLSTM and RoBERTa MLM + classifier) achieved better results than TF-IDF and fastText. The BiLSTM model achieves a bit better results than the Transformer base model, which suggests that the Transformer model needs more resources. The classifier trained from scratch (without pre-training) produces inferior results, and this shows again that the pre-training step is an important element in classification tasks. Neural methods achieve better results than the human baseline, but human results are comparable to TF-IDF.

Pre-trained models trained on the much larger data set than the HSSS data set achieve the best results. Monolingual and multilingual models achieve similar results, but XLM-R large achieve lower results than other pre-trained models, indicating that the bigger models may not improve results on the classification tasks. Polish RoBERTa large achieved similar results to the base version, which might mean that RoBERTa large needs more pre-training steps to get better results.

V. CONTAMINATION STUDY

Using a pre-trained language model (or any other solution not constrained to the train set provided with the challenge) raises the question of data contamination or train-test overlap, i.e. (1) was the test set represented in the training set of the language model?, (2) did it make the results better (e.g. due to memorization of test texts by the language model)? See [1] for the discussion of data contamination in the case of the GPT-3 model when used for popular English NLP test sets.

We carried out a contamination study on the solution based on the Polish RoBERTa model (the best solution so far). As the Polish RoBERTa was trained (among other sources) on CommonCrawl 2019/2020 [6], and the HSSS was prepared using CommonCrawl 2012-2015 (mostly 2012), the risk of contamination was real (a significant percentage of Web content from 2012-2015 could survive up to 2019).

We searched the contents of CommonCrawl 2019 (as provided to us by the authors of [6]²) for the six-gram fragments of the HSSS test set, obviously taking into account the fact that feminine gender-specific forms were modified during the preparation of the HSSS test set.

The summary of the contamination study is given in Table IV, where the results obtained with Polish RoBERTa are compared against the best constrained solution (an LSTM trained on the HSSS training set). The following conclusions can be made:

- results on the contaminated subset *are* better (and the difference of the Accuracy/Likelihood metrics on the contamination and not contaminated metric is significant), and this might indicate that the problem is real;
- still, the percentage of data contaminated is low (3%), hence the impact on the total is limited; if we were to lower the results on the contaminated subset to be

²Unfortunately, we were unable to check the other sources, though the probability of them contaminating the test set seems much lower

the same as on the uncontaminated subset, the accuracy would be lower only by a small margin;

- note that this is not a proof of contamination; the cause of better results on the contaminated subset might be different, for example it might have been caused by the fact that CommonCrawl 2019 for Polish RoBERTa was filtered by a language model, whereas for the HSSS data set — only using handcrafted heuristics, i.e. sentences might be longer and “proper” (e.g. say with fewer spam texts), hence easier for a classification task.

VI. CONCLUSIONS

We showed that a pre-trained Transformer model can obtain strong results for a challenging classification tasks on short texts. It turned out that predictions done by humans (even aggregated) were much worse. What is important is that influence of contamination of the training set was practically excluded.

REFERENCES

- [1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [2] B. Bsir and M. Zrigui. Bidirectional LSTM for author gender identification. In *International Conference on Computational Collective Intelligence*, pages 393–402. Springer, 2018.
- [3] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. New York, NY, USA, 2016. ACM.
- [4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.
- [5] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [6] S. Dadas, M. Perełkiewicz, and R. Poświęta. Pre-training Polish Transformer-Based language models at scale. In L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, pages 301–314. Cham, 2020. Springer International Publishing.
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018.
- [8] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of The 33rd International Conference on Machine Learning*, 06 2015.
- [9] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision, 2009.
- [10] F. Galiński, Ł. Borchmann, and P. Wierchoń. “He Said She Said” — a male/female corpus of Polish. In N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).
- [11] F. Galiński, R. Jaworski, Ł. Borchmann, and P. Wierchoń. Gonito.net — open platform for research competition, cooperation and reproducibility. In A. Branco, N. Calzolari, and K. Choukri, editors, *Proceedings of the 4REAL Workshop*, pages 13–20. 2016.
- [12] F. Galiński, R. Jaworski, Ł. Borchmann, and P. Wierchoń. Vive la petite différence! Exploiting small differences for gender attribution of short texts. *Lecture Notes in Artificial Intelligence*, 9924:54–61, 2016.
- [13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

TABLE III

RESULTS. *HUMAN BASELINE WAS EVALUATED ONLY ON THE RANDOM SAMPLE OF SIZE 800. †REFERENCES TO REPOSITORIES AT GONITO.NET [11] ARE GIVEN IN CURLY BRACKETS. SUCH A REPOSITORY MAY BE ALSO ACCESSED BY GOING TO HTTP://GONITO.NET/Q AND ENTERING THE CODE THERE.

method	test likelihood	test accuracy	gonito submission†
human baseline*	0.00000	0.68125	{87a138}
TF-IDF + logistic regression	0.55278	0.67175	{ecc1ee}
TF-IDF + linear SVM	0.00000	0.66477	{da348f}
TF-IDF + XGBClassifier	0.54269	0.65112	{5a17c9}
fastText	0.54541	0.67448	{4d18c0}
Bi-LSTM	0.57177	0.69786	{a0d38c}
RoBERTa MLM + classifier	0.57068	0.69153	{203325}
RoBERTa classifier (only)	0.55784	0.67951	{6756e6}
Polish RoBERTa (base) finetuned	0.60913	0.74185	{049966}
Polish RoBERTa (large) finetuned	0.60503	0.74388	{2b8541}
XLM-R (base) finetuned	0.60015	0.72356	{bdac6e}
XLM-R (large) finetuned	0.57141	0.69047	{bdac6e}
Polish RoBERTa (base) active dropout	0.62110	0.74332	{ea4b15}
Polish RoBERTa (large) active dropout	0.61949	0.74406	{2e89da}
Polish RoBERTa (large) last layer + logic regression	0.54113	0.65956	{582542}
XLM-R (large) last layer + logic regression	0.54067	0.65545	{115246}
Polish RoBERTa (large) fill mask	0.00000	0.55828	{11633b}

TABLE IV

CONTAMINATED VS NOT CONTAMINATED SUBSET OF THE TEST SET. P-VALUES ARE CALCULATED WITH THE MANN–WHITNEY U TEST.

		contaminated	not-contaminated	all	p-value
items	#	4,076	130,542	134,618	
	%	3.0%	97.0%	100.0%	
Polish RoBERTa base	Likelihood	0.64656	0.62032	0.62110	0.0000
	Accuracy	0.77159	0.74244	0.74332	0.0007
LSTM (constrained)	Likelihood	0.58305	0.57142	0.57177	0.0000
	Accuracy	0.70118	0.69776	0.69786	0.3549

- [14] S. Hussein, M. Farouk, and E. Hemayed. Gender identification of Egyptian dialect in Twitter. *Egyptian Informatics Journal*, 20(2):109–116, 2019.
- [15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
- [16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.
- [17] D. Kodyan, F. Hardegger, S. Neuhaus, and M. Cieliebak. Author Profiling with bidirectional RNNs using Attention with GRUs: Notebook for PAN at CLEF 2017. In *CLEF 2017 Evaluation Labs and Workshop—Working Notes Papers, Dublin, Ireland, 11-14 September 2017*, volume 1866. RWTH Aachen, 2017.
- [18] S. Krüger and B. Hermann. Can an online service predict gender? On the state-of-the-art in gender identification from texts. In *2019 IEEE/ACM 2nd International Workshop on Gender Equality in Software Engineering (GE)*, pages 13–16. IEEE, 2019.
- [19] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In E. Blanco and W. Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics, 2018.
- [20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019.
- [21] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.
- [22] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- [23] J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop*, pages 43–48, 2005.
- [24] A. Siewierska. Gender distinctions in independent personal pronouns. In M. S. Dryer and M. Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013.
- [25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.
- [27] R. Veenhoven, S. Snijders, D. van der Hall, and R. van Noord. Using translated data to improve deep learning author profiling models. In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)*, volume 2125, 2018.
- [28] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*, 2019.
- [29] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

3.5 YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection

YOLO with High Dataset Augmentation for Vehicle Class and Orientation Detection

Jakub Pokrywka
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Poznań, Poland
jakub.pokrywka@amu.edu.pl

Abstract—The popularity of computer vision systems supporting vehicle drivers and autonomous devices is increasing. Such systems require a huge quantity of annotated images. Creating such datasets is very expensive. However, it is possible to utilize synthetic images as a training dataset. In this paper, an object detection model for Vehicle Class and Orientation Detection is presented. The task set is to use solely synthetic images to train a model, which will then be evaluated on real-world images. The method takes advantage of high dataset augmentation and manual correction of training parameters regarding training statistics. The model performs surprisingly well in the detection of objects, but the assignment of classes to objects is not satisfactory. An error analysis is carried out, and propositions for future work are discussed. The described solution attained a 0.397 weighted mAP score on real-world images, achieving third place in the IEEE BigData 2022 Vehicle Class and Orientation Detection Challenge 2022.

Index Terms—Object Detection, Synthetic Data Generation, Ensemble Learning, Urban Street Analysis

I. INTRODUCTION

Training computer vision deep neural networks requires huge datasets of pictures with annotations such as [1], [2], consisting of hundreds of thousands and more samples. Collecting these images requires a lot of effort, as does their manual annotation. However, it is possible to train an object detection model using synthetic images generated from computer simulators. The cost of dataset preparation is then minimal. This work focuses on model training on synthetic images and evaluation on real-world image settings. The topic of this work is object detection of Vehicle Class and Orientation. Increasing urban traffic and the number of autonomous vehicles are inducing the rapid development of artificial intelligence solutions for both urban monitoring systems and vehicles themselves. The method developed in this paper is an approach to a task set in the IEEE BigData Cup 2022 competition, involving Vehicle Class and Orientation Detection in the real world using synthetic images from driving simulators. The aim of the shared task is, quoting its authors [3]:

- “To modify/develop object detection neural networks to improve real-world vehicle detections using models trained on synthetic datasets prepared in a simulator.

- To examine the effect of pixel-based image augmentation techniques to generate photo-realistic images on detection results in the real world.
- Study the effect of synthetic images on improving object class detections with fewer annotations.“

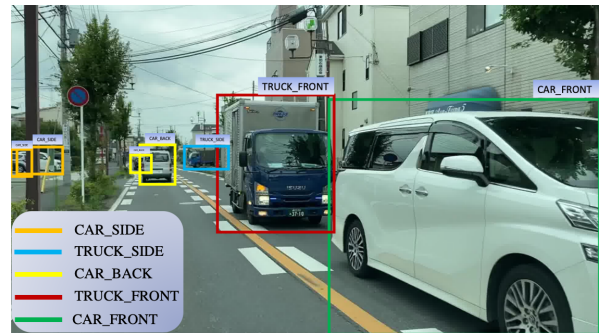


Fig. 1. Sample annotated image from the Vehicle Orientation Dataset. The picture comes from the project site.

II. RELATED WORK

The task of object detection is to localize an object in an image and assign a predefined class to it. Currently, deep-learning approaches outperform other methods. We distinguish two-stage and one-stage object detectors. Two-stage detectors make an object region proposal in the first place, and then classify and refine these region proposals using bounding box regression. Some two-stage detectors are R-CNN [4], Mask R-CNN [5], FPN [6], and Libra R-CNN [7]. One-stage object detectors do not use pre-generated region candidates, so that object classification and bounding box regression are performed in one stage. These include YOLO [8], YOLOv4 [9], RetinaNet [10], and SSD [11].

Vision deep neural networks benefit from using Image Data Augmentation. In this work, basic image manipulation was applied. A comprehensive overview of such techniques is found in [12].

Studies on utilizing synthetic images for training object detection neural networks include [13], [14]. Synthetic datasets have been generated and made publicly available, especially for urban traffic, such as SYNTHIA [15].

Computer vision systems for vehicles may provide significant assistance to human-driven and autonomous vehicles. For a survey of autonomous driving, see [16]. The topic of computer vision for autonomous vehicles is discussed in [17]. Datasets have been published for the development of autonomous vehicles, also containing labeled images [18].

III. CHALLENGE DESCRIPTION

This work proposes a solution for the Vehicle Class and Orientation Detection Challenge 2022, which is a part of the BigData Cup Challenges. The competition is related to the Vehicle Orientation (VO) Dataset project, described in the next subsection. Section III-B describes the Vehicle Class and Orientation Detection Challenge 2022 itself.

A. Original Vehicle Orientation Dataset

The competition is based on the project available at <https://github.com/sekilab/VehicleOrientationDataset> and described in [19], [20]. The original project data consists of a set of more than one million images of vehicles. More than 200,000 images are provided with vehicle orientation annotations. Annotations contain classes – car, bus, truck, motorcycle, bicycle – and the following orientation types: front, back, side. There are several pre-trained model weights for vehicle orientation available at the project site. A sample image with orientation is shown in Figure 1.

B. Vehicle Class and Orientation Detection Challenge 2022

The task of the challenge is to train a model on synthetic images, but solutions are evaluated on real-world images. The organizers of the competition released the Synthetic Vehicle Orientation (Synthetic VO) dataset for the purpose of model training. The dataset contains 63,066 images with annotations generated by the CARLA Simulator [21]. Sample images are given in Figure 3. The annotations are the same as in the VO dataset, but there is no bus class. Images represent different weather conditions, time of day, camera perspective, and traffic congestion. The classes are imbalanced and follow the long-tail distribution. This is partially similar to the VO dataset. Figure 2 depicts vehicles and class distributions for the VO and Synthetic VO datasets. Car and motorcycle vehicles are slightly overrepresented in Synthetic VO relative to the VO dataset; truck vehicles are about 2.5 times less frequent in Synthetic VO. A similar imbalance is noticeable with orientation annotations; for example, cars are more often pictured from a front view than a back view in the Synthetic VO dataset, while the opposite holds for the VO dataset. However, the imbalance is not a major one. The test dataset contains 3,000 real-world images. Sample images from the test dataset are shown in Figure 4.

Participants are allowed to use other synthetic images from driving simulators, such as CARLA and AirSim [22], or video games, such as Grand Theft Auto. It is forbidden to use real images with vehicle orientation annotations as a training dataset.

C. Metric

The metric is Weighted Mean Average Precision (weighted mAP), defined as:

$$\text{Weighted Mean Average Precision} = \sum_{k=1}^{12} w^k \times AP^k,$$

where w^k is a weight for each of 12 classes and $\sum_{k=1}^{12} w^k = 1$. The weighted metric was chosen by the competition organizers due to the long-tail class distribution. The weights of each class are based on its dominance in the test dataset.

IV. PROPOSED METHOD

The final solution is an ensemble of many object detection models trained with different parameters and weights initialized differently.

A. Data

The fact that training takes place on synthetic data and evaluation on real data raises the question of whether the validation set should be synthetic or real. On the one hand, a synthetic validation dataset checks whether the model is overfitted to specific training images (not to synthetic images in general). On the other, a real dataset checks the possibility of generalization to the actual test data. To utilize all of the available data for model training, I used the whole Synthetic VO dataset for training, so that I was able to choose only real data for the validation dataset. I used randomly chosen pictures from the VO dataset, because the competition organizers allowed their use for validation purposes. I decided not to generate synthetic data myself. However, if I had decided to do this, I would have used two validation datasets, one synthetic and one real, because I would not be limited by the synthetic dataset size.

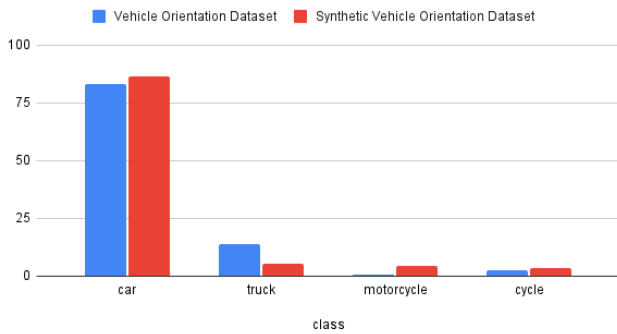
All images containing buses were excluded from the VO dataset, as they may be mistaken for trucks or other vehicles. From the remaining part, I randomly selected 9,335 pictures for the validation dataset.

B. Models

The proposed solution is an ensemble of many models trained in the YOLOv5 object detection library [23]. There are two types of model architectures used: YOLOv5l6 (76.8M params) and YOLOv5x6 (140.7M params). The models were trained with different data augmentation settings and different class box losses. In every case, the input image resolution was 1280. The first of the models is YOLOv5, with training parameters described as follows:

```
lr0: 0.01
lrf: 0.1
momentum: 0.937
weight_decay: 0.0005
warmup_epochs: 3.0
warmup_momentum: 0.8
warmup_bias_lr: 0.1
box: 0.05
```

Percentage vehicle distribution in real and synthetic datasets



Percentage class distribution in real and synthetic datasets

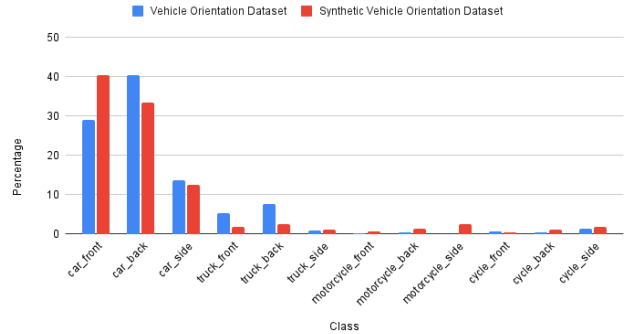


Fig. 2. VO and Synthetic VO percentage vehicles and class distribution. This figure does not include the bus class in the VO dataset, as that class is not included in the Synthetic VO dataset. Similarly, the total number of annotations for computing percentages excludes the bus class.



Fig. 3. Sample images from the training dataset of the Vehicle Class and Orientation Detection Challenge 2022



Fig. 4. Sample images from the test dataset of the Vehicle Class and Orientation Detection Challenge 2022

```

cls: 0.3
cls_pw: 1.0
obj: 0.7
obj_pw: 1.0
iou_t: 0.20
anchor_t: 4.0
fl_gamma: 0.0
hsv_h: 0.03
hsv_s: 0.85
hsv_v: 0.6
degrees: 10.0
translate: 0.3
scale: 0.9
shear: 0.2
perspective: 0.0
flipud: 0.0
fliplr: 0.5
mosaic: 0.3

```

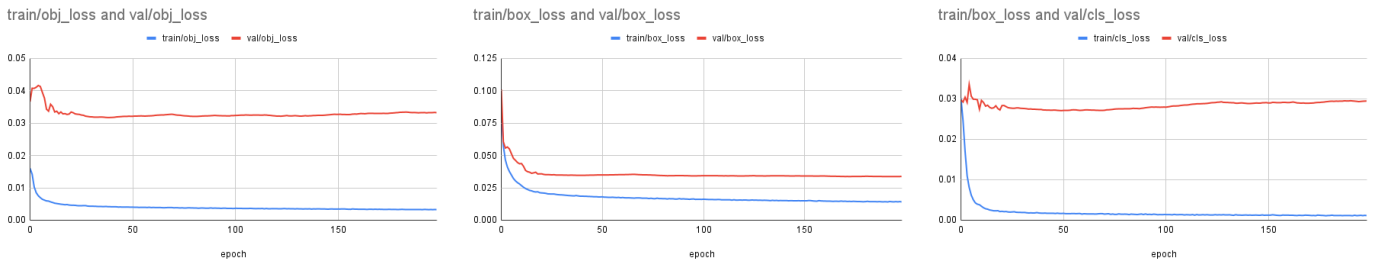
```

mixup: 0.1
copy_paste: 0.1

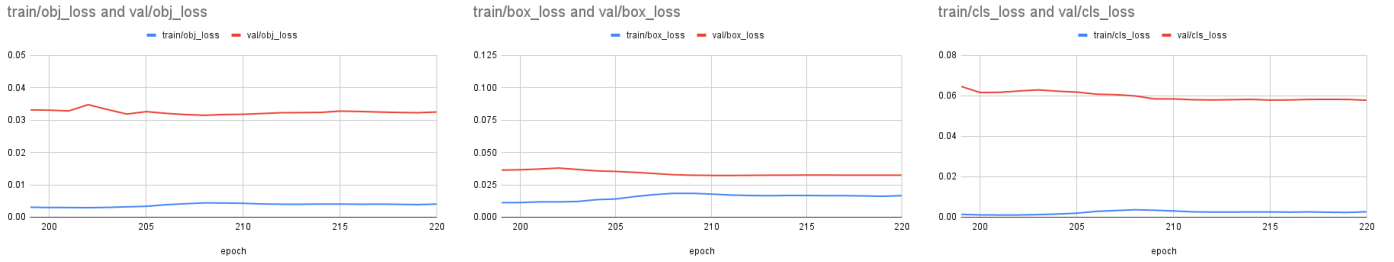
```

There are similar to the default high augmentation parameters in <https://github.com/ultralytics/yolov5/blob/master/data/hyps/hyp.scratch-high.yaml>, with changes, in particular the increased image HSV-Hue augmentation (hsv_h), image HSV-Saturation augmentation (hsv_s), HSV-Value augmentation (hsv_v), shear probability and translation probability, and decreased mosaic probability.

After inspection of the training plots, presented in Figure 5(a), I noticed increasing validation classification loss. Checking the validation images confirmed that boxes were often in the correct place, but the class was incorrect. To resolve this issue, I doubled the class loss gain parameter (cls) from 0.3 to 0.6. The plots of resumed training with the changed parameter are presented in Figure 5(b). Examples of data augmentations are given in Figure 6.



(a) Initial model training using the parameters given in section IV-B



(b) Model resumed from the previous checkpoint after changing class box gain from 0.3 to 0.6

Fig. 5. YOLOv5 with high data augmentation settings: model training losses. Note that the model is trained on synthetic images but validated on real images.

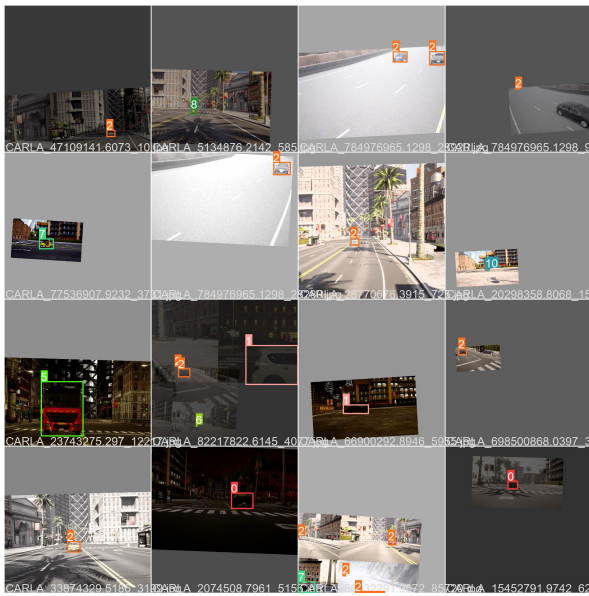


Fig. 6. Train augmentation examples

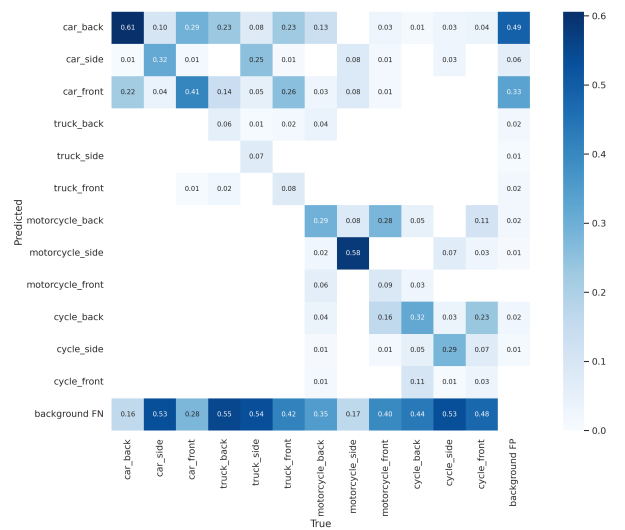


Fig. 7. Confusion matrix on validation data

For the next models, I used different data augmentation settings, based additionally on <https://github.com/ultralytics/yolov5/blob/master/data/hyps/hyp.scratch-med.yaml>, with modifications.

For the final solution inference, I used an Intersection Over Union threshold parameter of 0.35 and a very low confidence threshold (due to the use of the weighted mAP metric). I did not use Test Time Augmentations.

C. Result

The best single-model training achieved a score of 0.3473 weighted mAP. This is the result of an ensemble of the best and the last model checkpoint. Ensembling many models with different sizes (YOLOv5l6, YOLOv5x6) and different training parameters resulted in a final score of 0.3964. This outcome achieved the third position in the final competition ranking. Some model inference outputs for training and validation

datasets are shown in Figure 8.

D. Error analysis

The error analysis in this subsection is based on the manual inspection of images with predictions for validation and test datasets (some examples appear in Figure 8) and the confusion matrix on validation data (see Figure 7).

The object boxes are generally placed in the appropriate positions. Some background objects are sometimes misidentified, including traffic lights, traffic signs, minor objects on the pavement, and house elements.

In assigning classes to boxes, the model makes many mistakes. Car and truck classes are often confused, as are motorcycles and cycles. The orientation of the object is often given incorrectly.

V. POSSIBLE FUTURE WORK

Although the training dataset contains more than 60k images, for better model performance it could be even larger, due to the low cost of creating the synthetic dataset. The use of simulators other than CARLA may be especially beneficial. In particular, the problem of the model’s confusion of classes could be resolved by adding more models of vehicles to the simulators. This issue might also be addressed by generating a more similar percentage class distribution (Figure 2).

The training and test data are also different in that all test data are pictures from a camera inside a car, facing the car’s direction of travel, while most training data are pictures taken outside a vehicle, from many different points of view (see Figure 3 and Figure 4). The large difference in object position and dimension distributions is clearly visible when training and validation labels correlograms are compared (see Figure 9). For e.g. in the validation dataset object center y coordinate more closely resembles a normal distribution and has a smaller standard deviation than in the training dataset. In future work, it will be useful to explore whether generating a training dataset from the same camera perspective as the test dataset leads to a better model.

The fact of camera placement in the test data can also be utilized in a different way. For example, some additional models for post-processing object detection may be developed. Such a model would take into consideration, for example, that if there is a vehicle in the same road lane in front of a camera, it is probable that the vehicle is traveling in the same direction as the car with a camera, and therefore the vehicle orientation in the image is more likely to be back than front.

VI. CONCLUSION

In this paper, I have presented a solution for detecting Vehicle Class and Orientation in real-world images, which was trained solely on synthetic images. The method achieved third place in the Vehicle Class and Orientation Detection Challenge 2022 competition. It is based on high dataset augmentations, manual correction of training hyperparameters, and many ensembles of different model sizes and training hyperparameters. The trained model generally performs well, especially in

detecting vehicles. However, it sometimes confuses vehicle classes and orientations. With the aim of improving the model, an error analysis was carried out, and possible future work was discussed.

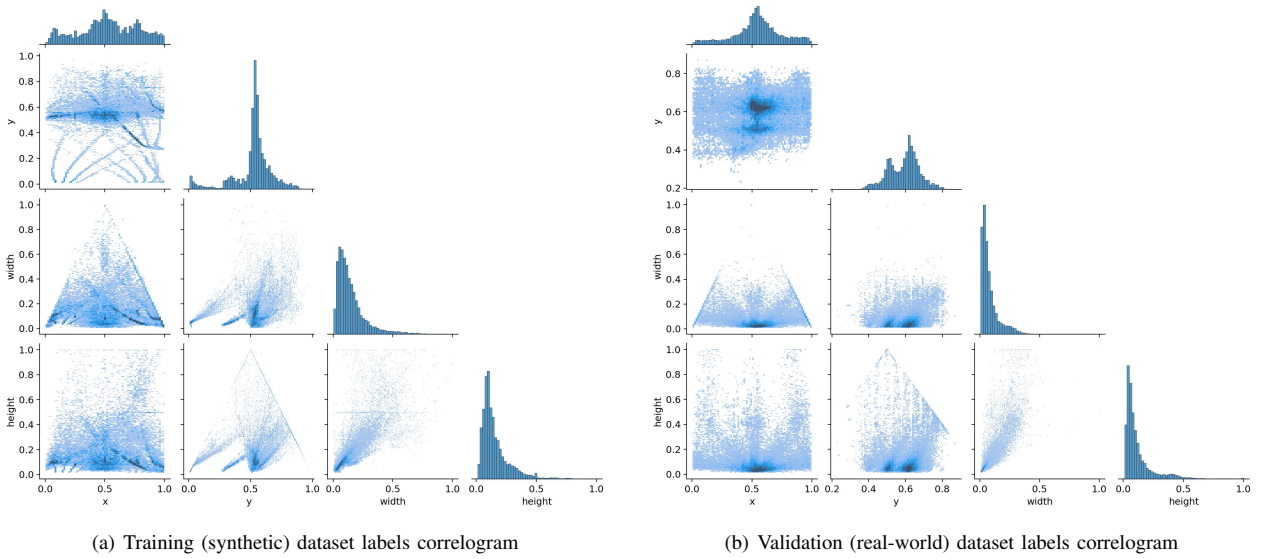


(a) Validation set labels



(b) Model inference on validation set

Fig. 8. Validation set labels and final model ensemble inference



(a) Training (synthetic) dataset labels correlogram

(b) Validation (real-world) dataset labels correlogram

Fig. 9. Training and validation dataset labels correlograms. Note that the x and y axes do not always start and end at 0 and 1.

REFERENCES

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
- [2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.
- [3] "Vehicle class and orientation detection challenge 2022." <https://vod2022.sekilab.global/>. Accessed: 2022-10-19.
- [4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.
- [7] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 821–830, 2019.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- [9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- [11] A. Del Signore, A. J. Hendriks, H. R. Lenders, R. S. Leuven, and A. Breure, "Development and application of the ssd approach in scientific case studies for ecological risk assessment," *Environmental Toxicology and Chemistry*, vol. 35, no. 9, pp. 2149–2161, 2016.
- [12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
- [13] A. Rozantsev, V. Lepetit, and P. Fua, "On rendering synthetic images for training an object detector," *Computer Vision and Image Understanding*, vol. 137, pp. 24–37, 2015.
- [14] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.
- [15] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [16] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020.
- [17] J. Janai, F. Güney, A. Behl, A. Geiger, *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.
- [18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscnescenes: A multi-modal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.
- [19] A. Kumar, T. Kashiyama, H. Maeda, and Y. Sekimoto, "Citywide reconstruction of cross-sectional traffic flow from moving camera videos," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 1670–1678, IEEE, 2021.
- [20] A. Kumar, T. Kashiyama, H. Maeda, H. Omata, and Y. Sekimoto, "Real-time citywide reconstruction of traffic flow from moving cameras on lightweight edge devices," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 192, pp. 115–129, 2022.
- [21] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.
- [22] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017.
- [23] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammama, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.

3.6 Efficient GPU Training of a Diversified Model Ensemble for the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022)

Efficient GPU Training of a Diversified Model Ensemble for the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022)

Jakub Pokrywka
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Poznań, Poland
jakub.pokrywka@amu.edu.pl

Abstract—Road maintenance inspection may be performed with the use of low-budget smartphones mounted inside a car, instead of expensive specialized vehicles with dedicated equipment. This approach, though, requires high-quality computer vision systems for processing the images. This paper describes a method developed during the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022). The method focuses on efficient GPU usage during model training and creating a diversified ensemble with different architectures and data augmentation settings. My approach achieves a good result, with a 0.60 F1-score for all images and an average 0.53 F1-score across all leaderboards, which is the competition’s final metric.

Index Terms—Object Detection, Urban Street Analysis, Road Damage Detection and Classification, Ensemble Learning

I. INTRODUCTION

The maintenance of roads in urban areas is crucial for safe transportation. The condition of the roads requires regular inspection. This may be done with specialized vehicles equipped with laser linescan and 3D cameras. This method allows the collection of the best possible data but is very expensive. The high cost of these professional systems has led to the emergence of low-cost solutions, which may be used by low-budget road maintenance agencies, for example in developing countries. One solution is the use of smartphones mounted inside an ordinary car. However, this requires developing a computer system for processing images collected from a smartphone. In this paper, I present an object detection solution for the detection of road damage, which was developed in the Crowdsensing-based Road Damage Detection Challenge (CRDDC2022). The shared task was to construct models for several countries. The method is a deep neural network object detection model, which achieved a 0.53 F-Score in the final ranking. The described approach focuses on efficient GPU utilization during model training, but without sacrificing the use of large object detection architectures and the possibility of creating many ensembles. Research in efficient GPU utilization is the AI community’s contribution to caring for the environment and bringing research opportunities closer

to the wider community. Although the competition allowed the construction of country-specific models, my experiments showed that a single model trained jointly on all of the data available for the competition achieves better results than country-specific models. The paper includes a discussion of the approach of creating a single model for all countries versus many country-specific models, with a relevant experiment.

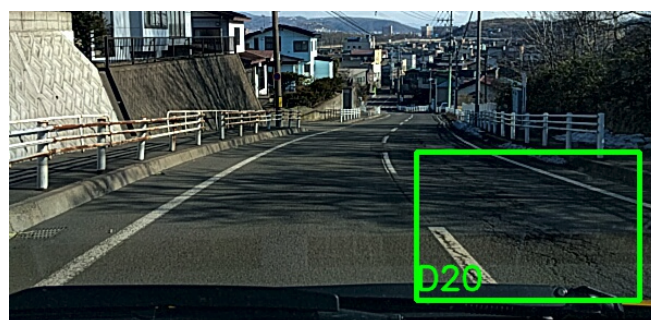


Fig. 1. Sample image with annotation from the competition page

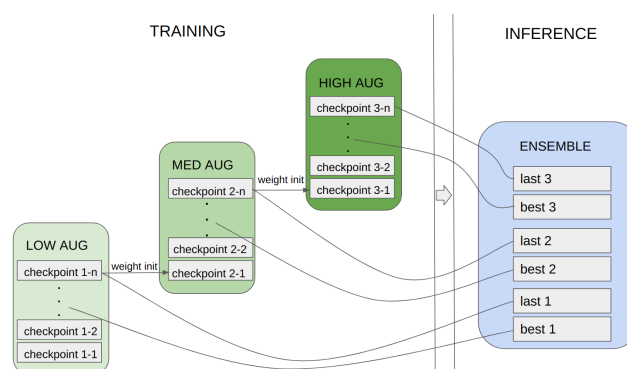


Fig. 2. Training and inference diagram

II. RELATED WORK

A. Object detection

The task of localizing objects in a picture and assigning relevant classes is called object detection. Nowadays, this task is dominated by deep-learning neural networks, which are superior to other methods. There are multiple works on these subjects, and many software frameworks have been released. Among them, there are two main approaches: one-stage single object detectors, and two-stage object detectors. Two-stage object detectors try to make object region proposals and then classify and refine those proposals. One-stage object detectors treat region proposals and their classification jointly and execute them in conjunction. Some of the two-stage detectors appreciated by the machine learning community are R-CNN [1], Mask R-CNN [2], FPN [3], and Libra R-CNN [4]. Single-stage detectors include YOLO [5], YOLOv4 [6], RetinaNet [7], and SSD [8]. A very popular and efficient single-stage object detection framework is YOLOv5 [9].

B. Road damage detection

The Road Damage Detection Dataset was first introduced in [10] as RDD2018. This dataset consists of 9,053 road images and 15,435 road damage instances. It was used for the Road Image Detection Challenge in 2018. The improved version with corrected annotations and augmentations with a Generative Adversarial Network is RDD2019 [11]. It consists of 13,133 images and 30,989 road damage instances. The works [12], [13] utilized the RDD2019 dataset in order to create a model working on images from several countries. Next, the RDD2020 [14] dataset was proposed, including images for India, Japan, and the Czech Republic, and it was utilized for the Global Road Damage Detection Challenge (GRDDC2020) [15].

C. Utilizing additional data sources

Recent studies show the benefit of utilizing additional data sources in large deep-learning models. In the domain of Natural Language Processing, [16] improves the Nepali language model tested on the Nepali test dataset using additional English and Hindi training datasets, while [17] shows surprisingly good performance in the zero-shot cross-language model transfer of a BERT model [18] trained on a monolingual corpus. There are effective large language models that have been trained on multiple languages simultaneously [19], [20].

In computer vision, the Road Damage Detection task [13] shows that adding data from other countries helps in the generalization of a model for any country. Many studies enrich a training data set with synthetic data images for object detection, for example, [21], [22].

III. CROWDSENSING-BASED ROAD DAMAGE DETECTION CHALLENGE

CRDDC2022, available at the site <https://crddc2022.sekilab.global/overview/>, consists of two phases. One is the contribution of datasets from different countries, as described in [23]. The other is the construction of an object detector for

TABLE I
CLASS COUNT IN COUNTRIES' DATASETS

Country	D00	D10	D20	D40
China Motorbike	2678	1096	641	235
China Drone	1426	1263	293	86
Czech Republic	988	399	161	197
India	1555	68	2021	3187
Japan	4049	3979	6199	2243
Norway	8570	1730	468	461
United States	6750	3295	834	135
Total	26016	11830	10617	6544

road damage instances from the following countries: China, the Czech Republic, India, Japan, Norway, and the United States. Images are generally collected by a smartphone placed behind the vehicle's front windscreen. This is not the case with the pictures from China, where training and testing images are collected from a motorbike, with additional training images collected from a drone. Sample images are shown in Figure 3, and a sample image from a drone in Figure 4. There are four road damage classes: D00 (Longitudinal Crack), D10 (Transverse Crack), D20 (Alligator Crack), and D40 (Potholes). A sample image with annotations is shown in Figure 1. The competition allows the creation of a single model for all countries or multiple models for individual countries. Pre-trained model weights are allowed, but the use of additional data is not.

A. Evaluation metric

Solutions in the competition are evaluated using the F1-Score metric. A prediction is correct if the Intersection over Union (IoU) between the predicted bounding box and the ground-truth bounding box is 0.5 or higher, and the predicted label matches the ground-truth label.

Using these auxiliary definitions:

- True Positive (TP) – the predicted label is in the ground-truth and $\text{IoU} \geq 0.5$
- False Positive (FP) – the predicted label is not in the ground-truth
- False Negative (FN) – there is a label in the ground-truth, but an object is not predicted

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

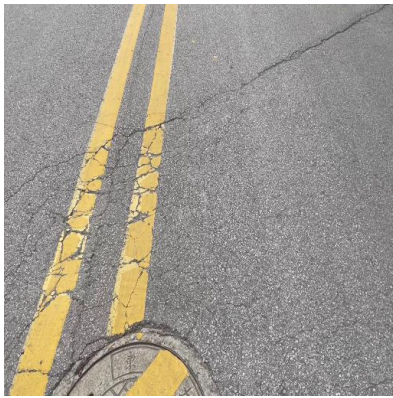
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The F1-Score metric is defined as follows:

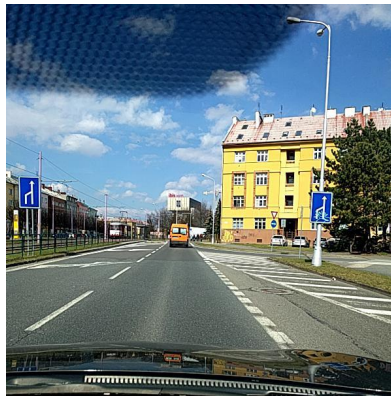
$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The metric is described in detail in [12].

The final competition result is an average of five F1-scores. These are the scores for India, Japan, Norway, and the United States, and a score for all of those countries plus the Czech Republic and China.



(a) China Motorbike



(b) Czech



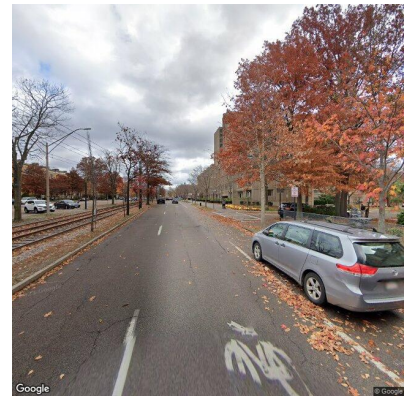
(c) India



(d) Japan



(e) Norway



(f) United States

Fig. 3. Sample images from countries' datasets



(a) China Drone

Fig. 4. Sample image from China Drone dataset

Classes per 100 images in countries' datasets

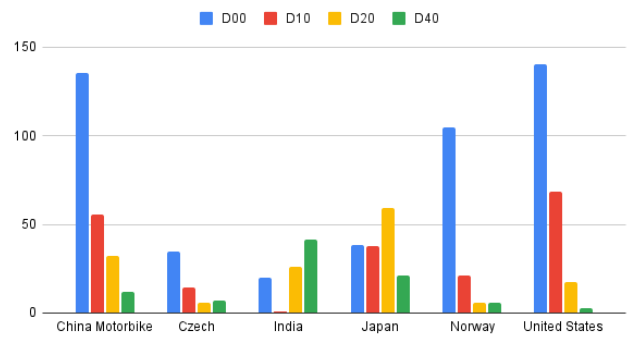


Fig. 5. Classes per 100 images in countries' datasets

IV. DATASET ANALYSIS

Dataset analysis is an important factor in the discussion on whether to use single or multiple models for object detection. It seems probable that the more similar are countries' datasets to each other, the more useful will be the approach of constructing one model trained jointly on all of the countries. A

comprehensive description is given in [23]. For sample images of datasets, see Figure 3, which includes all of the data sources. All the pictures were taken in good weather conditions. The dataset sizes are significantly different (see Table II). For example, in the case of the training datasets, there are 4–5 times more images from Japan than from the China Motorbike or Czech sets, and there are five times more images from Japan

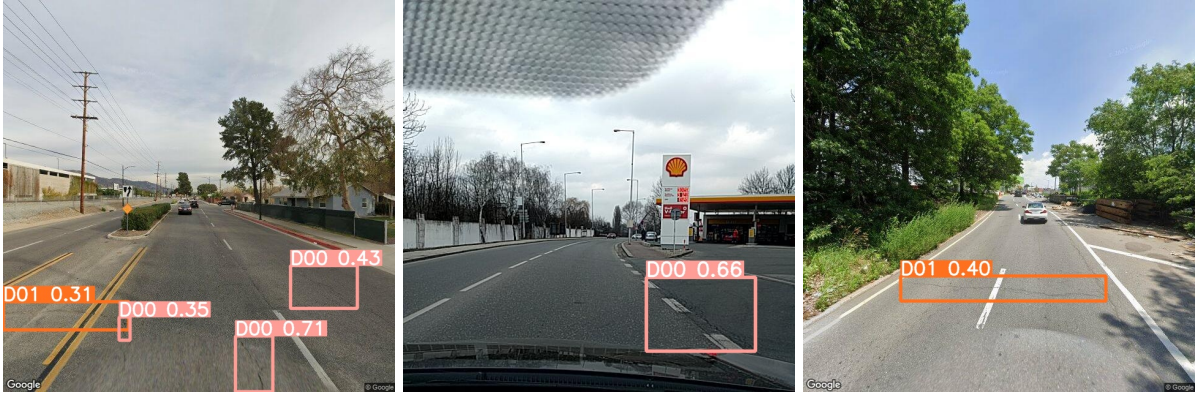


Fig. 6. Example correct predictions of the final model

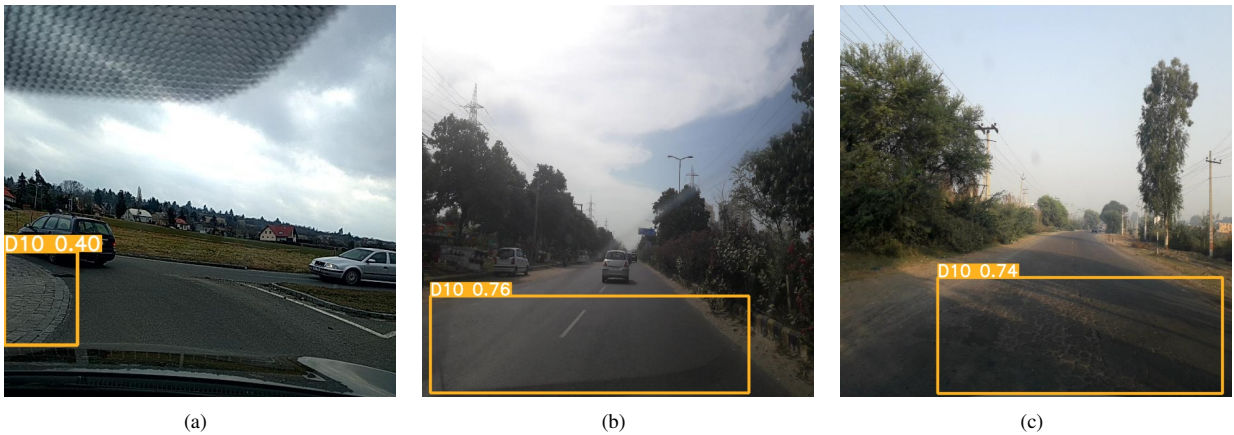


Fig. 7. Example erroneous predictions of the final model

than from China Motorbike in the test dataset. Table I and Figure 5 give the class distributions in each country. There are far more road damage instances per 100 images in China, Norway, and the United States than in the Czech Republic, India, and Japan. The distributions of classes are also different; for example, in the United States, there are eight times more D00 instances than D20 instances, but in Japan there are fewer D00 instances than D20 instances. For box size and placement on the image, see the labels correlogram in Figure 9. Width and height are generally exponentially distributed, and x and y are normally distributed, with the exception of Norway, where most of the boxes are located on the left part of an image. This may be due to the camera's viewing directions being more to the right. The correlogram for the China Motorbike dataset is not very different despite the camera's viewing directions being solely on the road, not including a horizon line.

V. PROPOSED APPROACH

A. Efficient training

My approach is to maximize the efficient usage of GPU without sacrificing model performance. Large deep neural architectures are generally more effective than smaller ones. It is also clear that ensembling techniques improve the final

TABLE II
DATASET SIZES

Country	Train size	Test size
China Motorbike	1977	500
China Drone	2401	0
Czech	2829	709
India	7706	1959
Japan	10506	2627
Norway	8161	2040
United States	4805	1200

result. The more diversified are the models contributing to the ensemble, the better its result. However, training many large models requires a large amount of GPU resources. The proposed solution is depicted in Figure 2. It consists of training a model firstly with low data augmentation settings until validation loss is flattened. Then a new model is trained with medium augmentation settings. Its weights are initialized with the last checkpoint of the previous training (with low data augmentation settings). It is trained until validation loss is flattened. Then a new model is trained with high augmentation settings. Its weight is initialized with the last checkpoint of the previous training, which used the medium data augmentation

settings. For inference, the ensemble of the last and best checkpoints of each training is used – this is six checkpoints in total. However, more checkpoints may be used. The strength of this approach is that it produces three models with different training settings, but the GPU time is the same as it would be for three separate training runs. This is because two of the three models’ weights are initialized not randomly, but based on an effective model.

The object detection library used is [9], with standard low, medium, and high data augmentation parameters obtained from <https://github.com/ultralytics/yolov5/tree/master/data/hyps>. I used two model architectures: YOLOv5l6 (76.8M params, 111.4B FLOPS) and YOLOv5x6 (140.7M params, 209.8B FLOPS). I used Test Time Augmentations (TTA) for the final solution; the score was slightly better than without TTA. The low data augmentation setting model was trained for 90 epochs, the medium setting model for 90 epochs, and the high setting model for 100 epochs. The YOLOv5x6 model was trained on a 4xA100 80 GB RAM GPU server with a batch size of 96. Each epoch took approximately 7.5 minutes, so that the full training took about 35 hours. The training time of the YOLOv5l6 model is negligible due to its smaller size and larger batch sizes. The training and inference image resolution was 1280p for maximum model performance.

The dataset split was 37,785 images for the training set and only 600 images for the validation set. This means that only about 1.5 percent of images are allocated to the validation dataset, which is quite a small number. This dataset split allows a model to see a large number of images, but the score on validation may not well illustrate the model’s result. However, a training set and test set images are very similar, which reduces overfitting, therefore, I chose the larger training set over the dev set. The best way to bypass this issue is to use k-fold cross-validation (with ensembling models trained on different folds). However, this would require more GPU resources, which I wished to avoid. Apart from D00, D10, D20, and D40, the training dataset contained more road damage instance classes with labels. I used them for training the object detector, for two reasons. The first was to ensure that the model would not mistake road image instances outside the assessed classes. The second was to help the model in the detection rather than the classification part of the training. Obviously, during inference, I restricted the possible classes to D00, D10, D20, and D40 using the YOLOv5 prediction settings. Similarly, the maximum number of detected objects was set to 5 due to the competition restrictions. No additional image post-processing was applied.

The source code is available at https://github.com/kubapok/T22_031_Jakub_Pokrywka_CRDCC22_solution

B. Multiple models for each country vs. a single model trained on all of the data

The recommendation in [12] suggests that mixing training data from outside a domain country may improve the results. The [12] authors trained a model mixing datasets from multiple countries.

However, my experiment described in this section has two stages. In the first stage, I trained a model on all the data. In the second stage, I further fine-tuned the model on each country dataset (separately for India, Japan, Norway, United States). I selected the best performing YOLOv5l6 model with high data augmentation settings trained on all the data for further fine-tuning. The fine-tuning on each country dataset employed low augmentation settings. My idea was to enable a model to see all the available data in the first stage and let the model learn the class distribution in the second stage. Even though the model was further fine-tuned solely on a specific country’s dataset, the performance on the test dataset was not better but slightly worse. The F1-scores of the final model and the best country-specific models are given in Table III. This may mean that despite a huge distribution class difference between countries’ datasets; the model trained on all the data is usually able to predict the correct class. For this reason, I used one single-model ensemble for the final solution.



Fig. 8. Confusion matrix on validation dataset including all classes contained in the training dataset

C. Results

The final model performs well, achieving a 0.60 F1-score on all images and an average 0.53 F1-score on all countries’ leaderboards. Nonetheless, the result could be improved. Sample correct predictions are shown in Figure 6 and sample incorrect predictions are shown in Figure 7. The 7(a) prediction is not road damage, but a different type of road. The 7(b) is not road damage as well but includes reflection on a windscreen. The 7(c) prediction should be class D20 (Alligator Crack), not D10 (Transverse Crack). The results of the final model are given in Table III in the Final model column. The confusion matrix for all classes in the training dataset is shown in Figure 8. As the diagram shows, the model does not rather confuse classes with each other, but often confuses the background with road damage instances.

TABLE III
F1-SCORE ON COMPETITION TEST DATA IN DIFFERENT COUNTRIES

Country	Best country-specific	Final model
India	0.41	0.42
Japan	0.58	0.60
Norway	0.36	0.38
United States	0.64	0.65
all	0.58	0.60
average	0.51	0.53

VI. CONCLUSION

In this paper, I have presented a GPU-efficient way to train an ensemble of large object detection models with different data augmentation hyperparameters. Even though the models did not require much computation time, they used YOLOv5l6 (76.8M params, 111.4B FLOPS) and YOLOv5x6 (140.7M params, 209.8B FLOPS) architecture. The ensemble consists of models trained with different data augmentation hyperparameter settings. The training took about 35 hours on a 4xA100 GPU server. The model performs well in the Crowdsensing-based Road Damage Detection Challenge. According to my research, training a model jointly on all of the countries' data is superior to fine-tuning to a specific country in the CRDCC2022 dataset. The developed approach achieved a 0.60 F1-Score on all images and an average 0.53 F1-score across all competition leaderboards. It is hoped that this research will contribute to more efficient GPU training in the future.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[3] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[4] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 821–830, 2019.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[8] A. Del Signore, A. J. Hendriks, H. R. Lenders, R. S. Leuven, and A. Breure, "Development and application of the ssd approach in scientific case studies for ecological risk assessment," *Environmental Toxicology and Chemistry*, vol. 35, no. 9, pp. 2149–2161, 2016.

[9] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomamma, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, m15ah, Doug, F. Ingham, Frederik, Guillhen, Hatovix, J. Poznanski, J. Fang, L. Y. , changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai,

"ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.

[10] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," *arXiv preprint arXiv:1801.09454*, 2018.

[11] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, 2021.

[12] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv preprint arXiv:2008.13101*, 2020.

[13] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," *Automation in Construction*, vol. 132, p. 103935, 2021.

[14] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "Rdd2020: An annotated image dataset for automatic road damage detection using deep learning," *Data in Brief*, vol. 36, p. 107133, 2021.

[15] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Global road damage detection: State-of-the-art solutions," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5533–5539, IEEE, 2020.

[16] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[17] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4996–5001, Association for Computational Linguistics, July 2019.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

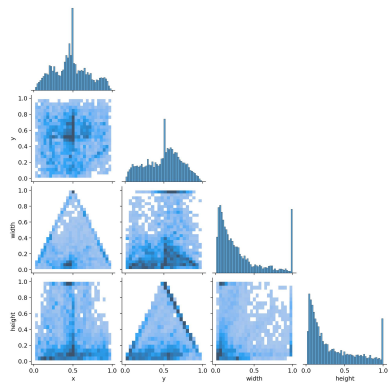
[19] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *arXiv preprint arXiv:2010.11934*, 2020.

[20] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 8440–8451, Association for Computational Linguistics, July 2020.

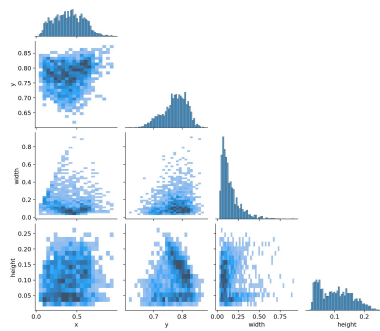
[21] A. Rozantsev, V. Lepetit, and P. Fua, "On rendering synthetic images for training an object detector," *Computer Vision and Image Understanding*, vol. 137, pp. 24–37, 2015.

[22] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

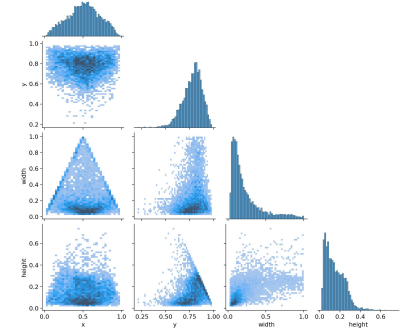
[23] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "Rdd2022: A multi-national image dataset for automatic road damage detection," *arXiv preprint arXiv:2209.08538*, 2022.



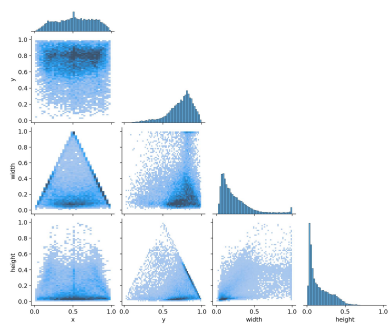
(a) China Motorbike labels correlogram



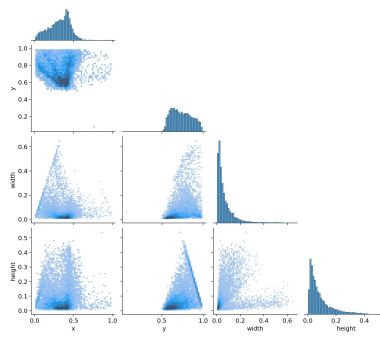
(b) Czech labels correlogram



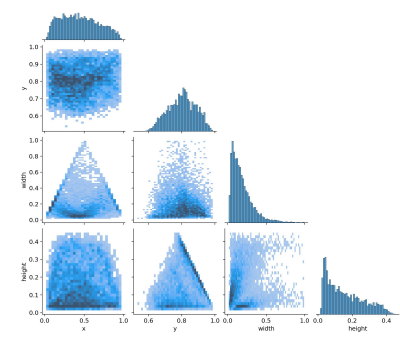
(c) India labels correlogram



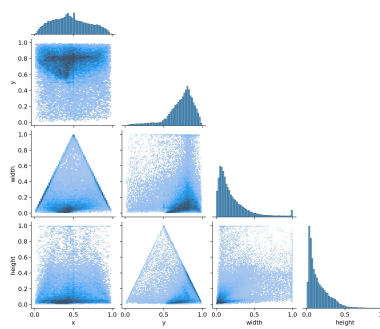
(d) Japan labels correlogram



(e) Norway labels correlogram



(f) United States labels correlogram



(g) All labels correlogram

Fig. 9. Training and validation dataset labels correlogram. Note that the x and y axes do not always start and end at 0 and 1.

3.7 Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images

Gradient Boosted Trees for Privacy-Preserving Matching of Encrypted Images

Jakub Pokrywka

Faculty of Mathematics and Computer Science

Adam Mickiewicz University

Poznań, Poland

jakub.pokrywka@amu.edu.pl

Abstract—A huge amount of data of various types is stored on servers and transferred over the Internet. This may pose a threat to the privacy of individuals. Encryption algorithms are developed to protect privacy. However, not all encryption mechanisms are effective. The easiest way of proving such ineffectiveness is by presenting a method of breaking the encryption scheme. In this paper, one method for matching original and encrypted images is proposed. The method was developed under the Privacy-preserving Matching of Encrypted Images shared task in the IEEE BigData 2022 Cup, and achieved second place with an accuracy of 0.6915, trailing the winning solution by only 0.0031. The proposed solution is based on Gradient Boosted Trees as implemented in CatBoost, and feature extraction of pixel value aggregates regardless of their positions. This is done using Arnold’s cat map obfuscation scheme in the encryption algorithm. Arnold’s cat map shuffles pixel positions, but leaves the image histogram unchanged. The method described in this paper almost solves two of the subtasks in the competition, reaching 0.98 accuracy for both of them; however, it does not propose a solution to the third subtask.

Index Terms—Image Encryption, Ensemble Learning, Gradient Boosted Trees, Cryptanalysis, Arnold’s Cat Map

I. INTRODUCTION

For large AI-based online platforms, it is common nowadays to collect as much information as possible. This is done in order to feed machine learning algorithms, which utilize huge amounts of data for superior performance [1]–[4]. Usually, the data encompass various formats – images, audio, text, etc. – and thus consume large amounts of storage space. Additionally, some collected data may be sensitive for users. This may include personal information such as name, address, gender, and age, but also personal pictures and voice messages. This entails the requirement to create high-quality encryption algorithms, which are not possible to break and are also efficient in terms of hardware resources. Attempts to break encryption methods may, in some cases, verify whether those methods are effective. In this paper, a machine learning method for breaking an image encryption scheme is presented. The encryption mechanism is based on Arnold’s cat map. The task was presented as a shared task [5], and the proposed solution achieved good results for two of the total three subtasks in the competition. The rest of this paper is organized as follows. The next section concerns related work. Section III describes

the IEEE BigData 2022 Cup Privacy-preserving Matching of Encrypted Images competition, with all of the subtasks. Section IV describes Arnold’s cat map, the algorithm used in the encryption scheme for two subtasks of the competition. Solutions for the first two subtasks are given in section V, and an attempt at a solution for the third subtask is described in section VI. The last section contains conclusions.

II. RELATED WORK

Many image encryption algorithms have been developed [6]–[10]. Some of them utilize Arnold’s cat map [6]. The most recent comprehensive review of image encryption algorithms is [11]. Systems developed in work on breaking image encryption obfuscation schemes include CatBoost [12], XGBoost [13], and LightGBM [14]. Randomness test models have been developed especially for testing image encryption algorithms [15].

Gradient boosted decision trees [16] are used in many machine learning competitions due to their superior performance in applications based on tabular data. The most common implementations are reported in [12]–[14]. In this work, I chose the CatBoost library due to its ease of use and superior performance. An important part of the presented solutions is ensemble learning, which is described extensively in [17].

III. IEEE BIGDATA 2022 CUP: PRIVACY-PRESERVING MATCHING OF ENCRYPTED IMAGES

The competition consists of subtasks S1, S2, and S3. For each task, 10,000 training pairs are provided, each consisting of an original image and the same image after encryption. The test set consists of 10,000 pairs with a source image and an encrypted image. In the test set, the encrypted image may or may not be from the source image. The task is to determine whether this is a relevant pair. In total, 30,000 training images and 30,000 test images are delivered. A summary is given in Table I. The difference between the subtasks is the encryption algorithm applied. In subtasks S1 and S2, the encrypted data is an image, and in subtask S3, the encrypted data is binary.

A. Evaluation

Evaluation is based on a weighted piece-wise accuracy measure:

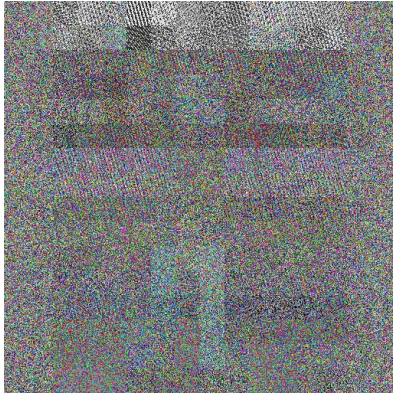
$$Acc = w_1 \cdot Acc_1 + w_2 \cdot Acc_2 + w_3 \cdot Acc_3,$$

Subtask	Train size	Test size
S1	10000	10000
S2	10000	10000
S3	10000	10000

TABLE I
DATASET SIZES



(a) Original image



(b) Encrypted image

Fig. 1. Example of original and encoded image in subtask S1

where the weights are $w_1 = 0.1$, $w_2 = 0.3$ and $w_3 = 0.6$, Acc_1 is accuracy for subtask S1, Acc_2 is accuracy for subtask S2, and Acc_3 is accuracy for subtask S3.

B. Subtask S1

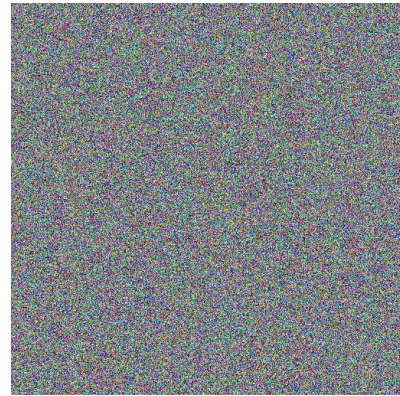
The encoding algorithm in subtask S1 is an obfuscation scheme based on Arnold's cat map. The details of the algorithm and its parameters are not revealed to competition participants. Original images are collected from the public domain and then preprocessed, resulting in 512×512 image resolution. Then the images are divided into tiles of 32×32 pixels. Each tile is individually encoded by means of the obfuscation scheme. Examples of an original and an encoded image are shown in Figure 1.

C. Subtask S2

The encoding algorithm in subtask S2 is the same obfuscation scheme as in subtask S1. The only difference is that



(a) Original image



(b) Encrypted image

Fig. 2. Example of original and encoded image in subtask S2

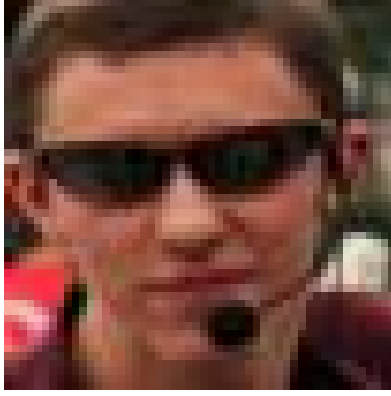
in subtask S1 the encryption algorithm is applied to each 32×32 pixel tile individually, but in subtask S2 the encryption algorithm is applied to the whole image. Examples of an original and an encoded image are shown in Figure 2.

D. Subtask S3

The encryption mechanism in subtask S3 is different from those described in subtasks S1 and S2. Here, the obfuscation scheme is based on Brakerski–Fan–Vercauteren Homomorphic Encryption. A similar algorithm is described in [18]. The result of the encoding is a binary string. The strength of this scheme is based on the hard computation problem Learning With Errors. Original images have a size of 52×52 pixels, and the encoded image is binary, with a size of about 42 KB. Examples of an original and an encoded image are shown in Figure 3.

IV. ARNOLD'S CAT MAPS

Arnold's cat map is an algorithm that shuffles the position of pixels in an image. The procedure is described in [6] as follows. We assume that the size of the original image I is $N \times N$. The coordinates of the pixels are $S = \{(x, y) \mid x, y = 0, 1, 2, \dots, N - 1\}$. The new positions of the pixels (x', y') may be calculated using:



(a) Original image

```

00000000: 01011110 10100001 00010000 00000011 00000111 00000010 0...
00000000: 00000000 00000000 01110010 10011001 00000110 00000000 ...r...
00000000: 00000000 00000000 00000000 00000000 00011000 10110101 ...t...
00000012: 00101111 11111101 10100000 01011001 00000000 00001000 /...Y...
00000018: 00000000 00101100 00010100 00001101 10011110 11111101 ...r...
0000001e: 10011111 10011011 01101000 00101001 00010000 10110000 ...h)...
00000024: 00011100 00100100 11101000 00110111 00001101 10111011 ...$...7...
0000002a: 01101000 01000000 10001000 11100101 00001010 10101010 h0...
00000030: 01000000 10100001 10011010 01100100 01111011 01011001 0...d...y...
00000036: 00101011 11101001 01111101 10111111 10000110 00100101 ...+)...%...
0000003c: 01101101 00001011 10001111 10101001 10111010 10111100 m...
00000042: 11110111 01110110 00100000 11111110 11111111 11111111 ...v...
00000048: 11111111 11101111 00000011 10000111 01101000 10000111 ...h...
0000004e: 01101000 10011110 01101000 00011011 00011100 10000001 h...h...
00000054: 01110010 10001000 11100000 01010101 00110101 01101011 r...USk...
0000005a: 00001010 10110010 11110110 01111110 00001100 10010101 ...-...
00000060: 01000100 11011000 01110000 11111010 11010101 10000101 D...p...w...
00000066: 00010010 00011010 01101010 10010011 01010111 11111011 ...j...M...
0000006c: 11011000 11000001 10101100 01110000 11100010 11010111 ...p...
00000072: 00011111 10101011 00100011 00000010 10100011 11111111 ...#...

```

(b) Beginning of binary string of encrypted image

Fig. 3. Example of original and encoded image in subtask S3. Note that the encryption algorithms output a binary file, not an image.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} = \begin{bmatrix} 1 & p \\ q & pq + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} \quad (1)$$

for some positive integers p, q ; $\det(A) = 1$. This procedure is usually performed multiple times. Since Arnold's cat map only shuffles pixel positions, the histogram of an image should not be changed, even if it is applied many times.

V. SOLUTION FOR SUBTASKS S1 AND S2

This section describes a solution for subtasks S1 and S2. The source code is available at <https://github.com/kubapok/IEEE-BigData-22-Privacy>. I addressed subtasks S1 and S2 using exactly the same method, since both of them use the same encryption method. Since the standard Arnold's cat map algorithm does not change images' histograms, it seems reasonable in this case to compare original and encrypted histograms.

Sample grayscale histograms are presented in Figure 4, and sample RGB histograms in Figure 5. As is clearly visible in the diagrams, the obfuscation scheme significantly changes the histograms, meaning that Arnold's cat map is not the only encryption mechanism applied. However, the pixel value distribution in the encoded image does not seem to come from a uniform distribution. It is rather a Gaussian distribution, and in the RGB histogram in the encoded image in Figure 5 there is a peak of high-value pixels in the blue channel, while there are fewer middle-value pixels in the green channel.

Additionally, there are visual differences between image tiles in the encoded image in Figure 1, which can be observed by exploring different encoded images and their histograms. This leads to the conclusion that extracting features of pixel values in the R, G, B, and grayscale channels, regardless of their positions, may provide good input for a discriminant.

Since this input data is in tabular form, I chose Gradient Boosted Trees [12], which usually achieves the best results, as confirmed in many machine learning competitions. The dataset preparations are described in subsection V-A, feature extraction in subsection V-B, and the machine learning algorithm in subsection V-C.

A. Dataset preparation

The training dataset provided by the competition organizers contained pairs consisting of an image and the encoded form of the same image. Usually, the classification supervised machine learning approach uses not only positive but also negative samples. For negative pairs, I took all of the encrypted images paired with randomly selected original images. This approach allows the creation of a huge number of negative pairs, because for each encrypted image, there are 9,999 images available to form a negative pair. In my experiments, I verified that having more negative pairs improves accuracy, but above the level of 10 times more negative than positive pairs, the difference is insignificant. Having regard to computational resources, I decided to retain the 10:1 ratio.

I used a dataset split of 3/10 for the validation set and the remainder for the training data set. For the final solution, I created ten folds of splits, trained ten models with the same hyperparameters, and averaged their output probabilities.

B. Feature extraction

This subsection describes the transformation of an image into a feature vector. The features were derived from the original and encoded image in the same procedure. For each of the Red, Green, and Blue channels and Grayscale, I extracted the following features:

- mean of pixels values
- minimum of pixel values
- maximum of pixel values
- variance of pixel values
- sum of pixel values
- percentiles of pixel values for $p = 0.5, 0.1, 0.15, \dots, 0.95$

I also created a histogram for the Red, Green, and Blue channels and Grayscale and obtained the features listed above from those histograms.

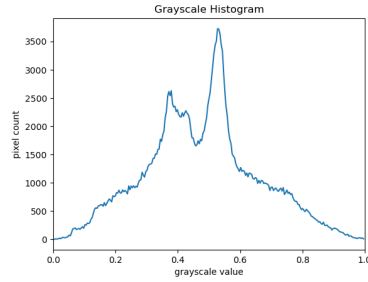
Additionally, I created the following features:

$$\begin{aligned} &\text{sum}(a \leq \text{pixels in color channel} < a + \text{interval and} \\ &\quad b \leq \text{pixels in color channel} < b + \text{interval and} \\ &\quad c \leq \text{pixels in color channel} < c + \text{interval}) \end{aligned}$$

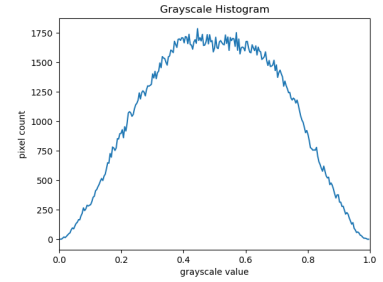
for color in (Red, Green, Blue), interval in (32, 64, 128), and a, b, c in $(0, 1 \cdot \text{interval}, 2 \cdot \text{interval}, \dots, 256 - 2 \cdot \text{interval}, 256 - 1 \cdot \text{interval})$.



(a) Original image in grayscale



(b) Original image grayscale histogram

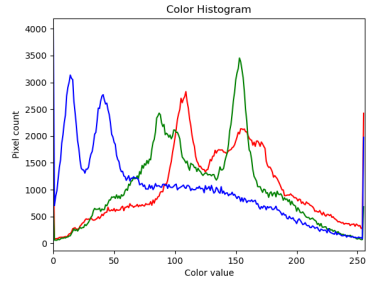


(c) Encrypted image grayscale histogram

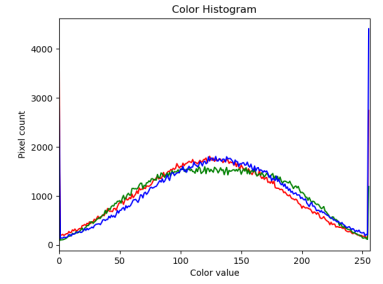
Fig. 4. Example of the grayscale histograms of original and encrypted images



(a) Original image



(b) Original image RGB histogram



(c) Encrypted image RGB histogram

Fig. 5. Example of the RGB histograms of original and encrypted images

In the same way, I created:

$$\text{sum}(a \leq \text{pixels in grayscale} < a + \text{interval})$$

with all the same parameters but with intervals in (8, 16, 32, 64, 128).

Reducing the interval length would result in too large an input vector, especially for all R, G, and B features, preventing training from executing in a reasonable time.

All of the above operations lead to the processing of a 512×512 image into a vector of length 2,019. Assuming that e is the vector of an encrypted image and o is the vector of an original image, I constructed vectors:

- $e + o$
- $e - o$
- $o - e$
- $e / (o + \epsilon)$
- $o / (e + \epsilon)$

for some small ϵ , e.g. $\epsilon = 0.0001$, and concatenated all of them with the vectors e and o .

In total, one input image pair is transformed into a vector of length 14,133. The time taken to process one image into a

vector is about 15 seconds. Therefore, it is beneficial to run the processing computations in parallel.

C. Machine learning algorithm

The extracted input vector is fed into the CatBoost algorithm. All of the model hyperparameters are default, except for the number of estimators. The model is trained until convergence using early stopping at 500 rounds, and then the best model is used for inference. I trained the model using a CPU on 20 threads. However, it is possible to speed up the training using a GPU. RAM usage during training is about 70 GB. Some model statistics are given in Table II and Table III. The used CPU was AMD EPYC 7402 2,8 GHz.

It may be beneficial to perform hyperparameter tuning. However, this requires huge computational resources due to the long single-model training time, so I did not include this step.

VI. SOLUTION FOR SUBTASK S3

I tried a similar approach for subtask S3 as in subtasks S1 and S2. Each encrypted image in subtask S3 was a binary string. Therefore, instead of extracting the features described

fold	best iteration	training time	train accuracy	val accuracy
1	1908	1h 19min	0.999	0.981
2	1601	1h 10min	0.998	0.984
3	2136	1h 27min	0.999	0.987
4	2378	1h 32min	0.999	0.986
5	2215	1h 27min	0.999	0.987
6	1948	1h 20min	0.999	0.982
7	3290	2h 00min	1.000	0.985
8	3734	2h 12min	1.000	0.984
9	3014	1h 52min	1.000	0.986
10	3505	2h 06min	1.000	0.987
avg	2573	1h 29min	0.999	0.985
std dev	750	0h 23min	0.000	0.002

TABLE II
TRAINING STATISTICS FOR SUBTASK S1

fold	best iteration	training time	train accuracy	val accuracy
1	1029	0h 51min	0.994	0.976
2	1259	0h 58min	0.996	0.976
3	1940	1h 19min	0.997	0.976
4	2759	1h 42min	1.000	0.978
5	2216	1h 26min	0.999	0.978
6	3880	2h 14min	1.000	0.979
7	2193	1h 27min	0.999	0.975
8	1837	1h 15min	0.998	0.978
9	1720	1h 12min	0.998	0.978
10	1617	1h 09min	0.998	0.975
avg	2045	1h 21min	0.998	0.977
std dev	811	0h 24min	0.002	0.001

TABLE III
TRAINING STATISTICS FOR SUBTASK S2

in subsection V-B, I created similar features, treating the binary string as image data in a way that handles a binary number as a pixel value. Apart from this, I added position-related features, such as the first one hundred bytes, the last one hundred bytes, and file size. Despite this, I was not able to propose any method achieving better than random results on the validation datasets. For this reason, I used a dummy model which always predicts 0 (not a match) for all pairs in the test dataset.

VII. CONCLUSION

This paper presents a solution for the Privacy-preserving Matching of Encrypted Images competition in the IEEE Big-Data 2022 Cup. The proposed solution achieves very good results for two subtasks out of a total of three. These two subtasks are to identify whether an encrypted image comes from a presented original image. The encryption algorithm is an obfuscation scheme based on Arnold’s cat map. The proposed classifier is the gradient boosted trees model implemented in the CatBoost library. The key factor is feature extraction, which is an aggregate of pixel values regardless of their position in images, due to the pixel-shuffling property of Arnold’s cat map. The ensembling method takes a 10-fold data train/validation split, and average model probabilities. Besides this, creating additional negative pairs in the training datasets elevates the score. The method achieves 0.985 accuracy for subtask S1 and 0.977 accuracy for subtask S2 (average over ten folds). Unfortunately, I was not able to propose any better than random solution for subtask S3. Despite this, the method scored 0.691500 for accuracy in the competition’s final

results, which is only 0.0031 behind the best solution in the competition (0.694600).

The present work leads to the conclusion that the encryption method applied in subtasks S1 and S2 is not safe. The author was not able to prove the same for the encryption method applied in subtask S3.

REFERENCES

- [1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [2] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhume, G. Zerveas, V. Korthikanti, *et al.*, “Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model,” *arXiv preprint arXiv:2201.11990*, 2022.
- [3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.
- [4] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” *arXiv preprint arXiv:2205.11487*, 2022.
- [5] A. Janusz, M. Szczuka, B. Cyganek, J. Grabek, Ł. Przebinda, A. Zalewska, A. Buwała, and D. Ślęzak, “IEEE Big Data Cup 2022 Report: Privacy preserving matching of encrypted images,” in *2022 IEEE International Conference on Big Data, BigData 2022, Osaka, Japan, December 17-20, 2022*, 2022.
- [6] Z.-H. Guan, F. Huang, and W. Guan, “Chaos-based image encryption algorithm,” *Physics Letters A*, vol. 346, no. 1, pp. 153–157, 2005.
- [7] Y. Zhou, W. Cao, and C. P. Chen, “Image encryption using binary bitplane,” *Signal Processing*, vol. 100, pp. 197–207, 2014.
- [8] S. J. Shyu, “Image encryption by random grids,” *Pattern Recognition*, vol. 40, no. 3, pp. 1014–1031, 2007.
- [9] R. Rhouma, S. Meherzi, and S. Belghith, “Ocml-based colour image encryption,” *Chaos, Solitons & Fractals*, vol. 40, no. 1, pp. 309–318, 2009.
- [10] Y. Mao and G. Chen, “Chaos-based image encryption,” in *Handbook of Geometric Computing*, pp. 231–265, Springer, 2005.
- [11] M. Kaur and V. Kumar, “A comprehensive review on image encryption techniques,” *Archives of Computational Methods in Engineering*, vol. 27, no. 1, pp. 15–43, 2020.
- [12] A. V. Dorogush, V. Ershov, and A. Gulin, “Catboost: gradient boosting with categorical features support,” *arXiv preprint arXiv:1810.11363*, 2018.
- [13] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16*, (New York, NY, USA), pp. 785–794, ACM, 2016.
- [14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.
- [15] Y. Wu, J. P. Noonan, S. Aghaian, *et al.*, “Npcr and uaci randomness tests for image encryption,” *Cyber journals: multidisciplinary journals in science and technology. Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.
- [16] L. Breiman, “Arcing the edge,” 1997.
- [17] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.
- [18] J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption.” *Cryptology ePrint Archive*, Paper 2012/144, 2012. <https://eprint.iacr.org/2012/144>.

Chapter 4

Declarations of Contribution

Poznań, April 4, 2023

Declaration of Contribution

I hereby declare that the contribution to the following paper:

Jakub Pokrywka, Filip Graliński, Krzysztof Jassem, Karol Kaczmarek, Krzysztof Jurkiewicz, Piotr Wierchoń
Challenging America: Modeling language in longer time scales
Findings of the Association for Computational Linguistics: NAACL 2022

is correctly characterized in the table below.

Contributor	Task description
Jakub Pokrywka	Implementation of the algorithm for generating machine learning challenges according to the methodology proposed by Prof. Filip Graliński. Idea and implementation of the algorithm for selecting images and text fragments from newspapers. Preparation of scripts for pretraining the temporal language model. Proposal of Haversine metric for geo coordinate task. Creation of the baseline models for challenges. Results analysis. Writing of the article.
Filip Graliński	Conceptualization. The idea of methodology for creating machine learning challenges. Scientific supervision. Preprocessing and collection of the text corpora. Result analysis. Writing of the article.
Krzysztof Jassem	Scientific supervision. Writing of the article.
Karol Kaczmarek	Implementation of Haversine metric for the geo coordinates task. Supervising pretraining of the RoBERTa temporal language models.
Krzysztof Jurkiewicz	Creation of corpus statistics. Some parts of the implementation of the algorithm for generating machine learning challenges according to the methodology proposed by Prof. Filip Graliński. Writing of the article.
Piotr Wierchoń	Data analysis from the historical and linguistic perspective. Writing of the article.



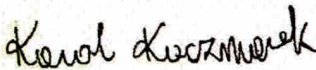
Jakub Pokrywka



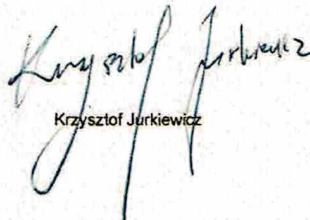
Filip Graliński



Krzysztof Jassem



Karol Kaczmarek



Krzysztof Jurkiewicz



Piotr Wierchoń

Poznań, April 4, 2023

Declaration of Contribution

I hereby declare that the contribution to the following paper:

Jakub Pokrywka, Filip Graliński
Temporal Language Modeling for Short Text Document Classification with Transformers
2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)

is correctly characterized in the table below.

Contributor	Task description
Jakub Pokrywka	Conceptualization and methodology. Selecting and preparing the text corpora. Creating diachronic challenges. Implementation of the machine learning models (especially based on temporal embeddings). Running experiments. Analyzing data and results. Writing most of the article.
Filip Graliński	Running experiments. Scientific supervision. Writing and correction of the article.

Jakub Pokrywka



Filip Graliński



Declaration of Contribution

I hereby declare that the contribution to the following paper:

Jakub Pokrywka, Marcin Biedalok, Filip Grański, Krzysztof Biedalok

Modeling Spaced Repetition with LSTMs

in Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023), In Print.

is correctly characterized in the table below.

Contributor	Task description
Jakub Pokrywka	Implementation of the part of ML methods, including the idea and implementation of the XGBoost method with exponential decay. Analyzing the results on the general test data regarding the various metrics. Writing of the article.
Marcin Biedalok	Preparation of the data. Creating the synthetic test set and analysis of the results of the methods on the synthetic test set. Creating a challenge based on the data. Implementation of non-ML methods and implementation of some of the ML methods. Writing of the article.
Filip Grański	Conceptualization and methodology. Scientific supervision. Proposal and analysis of metrics. Writing of the article.
Krzysztof Biedalok	Research idea. Implementation supervision. Provision of the data. Analysis of results in terms of implementation application. Provision of the knowledge on SuperMemo expert methods. Writing of the article, particularly in the field of SuperMemo and SuperMemo expert methods.

Jakub Pokrywka



Marcin Biedalok



Filip Grański



Krzysztof Biedalok



Poznań, April 4, 2023

Declaration of Contribution

I hereby declare that the contribution to the following paper:

Karol Kaczmarek, Jakub Pokrywka, Filip Graliński
Using Transformer models for gender attribution in Polish
2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)

is correctly characterized in the table below.

Contributor	Task description
Karol Kaczmarek	Implementation of some of the models using Transformer architecture (Polish RoBERTa, RoBERTa, XLM), tuning hyperparameters on them, and scaling experiments. Designing and implementation of data annotation mechanism. Preparing methodology for evaluating model outputs against human baselines. Supervising data annotation. Writing of the article.
Jakub Pokrywka	Implementation of TFIDF, fastText, and LSTM methods. Implementation of some of the models using Transformer architecture (Polish RoBERTa, Polish RoBERTa with Monte-Carlo model averaging). Acquisition and supervision of annotators. Data annotation. Partial data preparation for contamination analysis. Writing of the article.
Filip Graliński	Scientific supervision. Research idea. Model contamination analysis. Writing of the article.


Karol Kaczmarek


Jakub Pokrywka


Filip Graliński