Vitali at ACTI – Transformer-based Conspiracy Theory Identification

Michael Vitali[1], Vincenzo Scotti[1] and Mark James Carman[1]

[1]DEIB, Politecnico di Milano, Via Ponzio 34/5, 20133, Milano (MI), Italy

Abstract

English. In this work, we participated in the Automatic Conspiracy Theory Identification (ACTI) competition, which involved two sub-tasks: identifying whether an input text is a conspiracy theory and recognising the specific conspiracy theory it discusses. Our approach involved fine-tuning two BERT models, one in Italian and one multilingual, and combining them in an ensemble. The results were promising, and we achieved a position among the top participants in the challenge. This work contributes to the advancement of automatic conspiracy theory identification and highlights the effectiveness of fine-tuned BERT models in this domain.

Italiano. In questo lavoro, abbiamo partecipato alla competizione di Identificazione Automatica delle Teorie Cospiratorie (Automatic Conspiracy Theory Identification, ACTI), che si compone due sotto-problemi: identificare se un dato testo riguarda una teoria del complotto e riconoscere a quale teoria del complotto in particolare si fa riferimento nel testo. Il nostro approccio prevedeva l'adattamento di due modelli BERT, uno in italiano e uno multilingue, e la loro combinazione in un ensemble. I risultati sono stati promettenti e abbiamo raggiunto una posizione tra i primi partecipanti nella sfida. Questo lavoro contribuisce allo sviluppo dell'identificazione automatica delle teorie del complotto e mette in evidenza l'efficacia dei modelli BERT adattati in questo ambito.

Keywords

Natural Language Processing, Transformer Network, BERT, Ensemble, Conspiracy Theory Identification

1. Introduction

The *Automatic Conspiracy Theory Identification* (ACTI) task [1], organised as part of the *Evaluation of Natural Language Processing* (NLP) *and Speech Tools for Italian* (EVALITA) campaign[1] [2], tackles the problem of automating the identification and classification of conspiracy theories. This task arose from the need to address the growing concern surrounding the spread of conspiracy theories on various online platforms, including social media. In the age of *Large Language Models* (LLMs) [3] and widespread disinformation on social media, the ability to track and recognise these theories automatically is highly relevant [4]. In fact, the dissemination of conspiracy theories has the potential to manipulate public opinion, disrupting societal harmony and trust in institutions.

Moreover, while mainstream platforms apply community-level moderation policies, there is a strong ongoing discussion regarding the effectiveness of these methods, as highlighted in recent works [5, 6]. This discussion points out the need for the development of ad-hoc methods to identify rapidly changing and evolving content, such as conspiracy theories. These methods aim to complement and enhance the existing moderation strategies to better address the challenges posed by the proliferation of conspiracy theories in online spaces.

NLP and Deep Learning techniques offer promising solutions that can ease the analysis of large volumes of textual data [7, 8]. Deep Learning-powered NLP models can analyse text data by looking for semantic and lexical cues that can discriminate regular documents from conspiracy theories narratives, and, within the umbrella of conspiracy theories, which one is the subject of the document. In our work, we relied on state-of-the-art transformer neural networks to tackle the problem, combining single classifiers into ensembles to improve the performances as much as possible.

We organised this report into the following sections. In Section 2, we describe the two subtasks composing the ACTI task, the available data, and the preprocessing steps we applied. In Section 3, we present the classification models we built to solve the tasks and we trained such models. In Section 4, we explain the evaluation approach, report our results, and comment on these results. Finally, in Section 5, we summarise our contribution and suggest possible future extensions.

2. Task

In this section, we present the data sets that constitute the two different sub-tasks of the ACTI task (hereafter

EVALITA 2023: Overview of the 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, September 07–08, 2023, Parma, Italy

michael.vitali@mail.polimi.it (M. Vitali); vincenzo.scotti@polimi.it (V. Scotti); mark.carman@polimi.it (M. J. Carman)

0000-0002-8765-604X (V. Scotti); 0000-0001-6575-9737 (M. J. Carman)

CEUR Workshop Proceedings (CEUR-WS.org)

[1]Website: https://www.evalita.it

Table 1
Sub-task data sets main statistics.

Sub-task	Split	No. of samples	Avg. tokens per sample
Original			
A	Train	1842	59.17±73.89
	Test	460	63.67±78.67
B	Train	810	80.09±90.74
	Test	300	77.81±83.95
Pre-processed			
A	Train	3684	51.35±60.17
	Test	460	56.31±66.42
B	Train	810	52.43±65.33
	Test	300	69.34±73.63

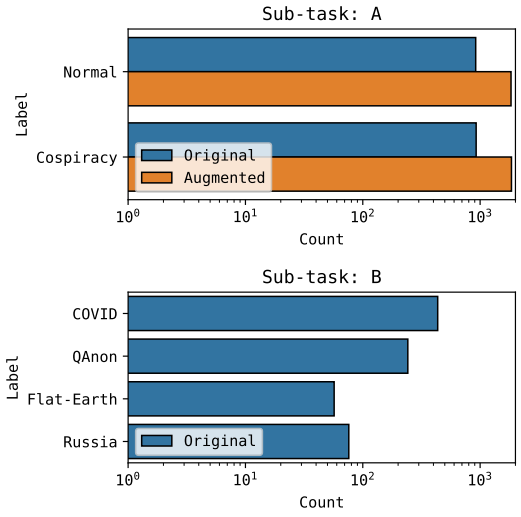


Figure 1: Sub-tasks label distributions.

referred to as *sub-task A* and *sub-task B*). We also describe the pre-processing steps applied to clean and prepare the data sets for training our text-based classifiers.

2.1. Data

The ACTI task comprises two sub-tasks: sub-task A, a binary classification task to determine if a given text piece is about a conspiracy theory or not, and sub-task B, a multi-class classification task to recognise specific conspiracy theories. Both sub-tasks use separate data sets consisting of Italian text samples. Table 1 provides an overview of the main statistics for the text samples in each corpus, we used *NLTK* [9] tokeniser to compute the number of tokens. Figure 1 illustrates the label distributions.

Sub-task A involves binary classification to identify whether a text sample relates to a conspiracy theory or not. Samples contain noise like emoticons or spelling errors, hence we assumed they had not been pre-processed. Label distribution is well balanced between the two classes, as can be seen in the top part of Figure 1.

Sub-task B extends sub-task A by introducing a multi-class classification aspect. The goal is to identify if a text pertains to one of the following conspiracy theories: *COVID*, *QAnon*, *Flat-Earth*, and *Russia*. As for sub-task A, samples have not been pre-processed. Differently from sub-task A, the label distribution is unbalanced, COVID and QAnon are more frequent than Flat-Earth and Russia.

2.2. Pre-processing

Text samples in the data sets contain a lot of noise, like emoticons, slang, or spelling errors; thus, we applied some cleaning and pre-processing steps. Moreover, the two data sets do not contain a large number of samples, and sub-task B presents an unbalanced label distribution, introducing the risks of overfitting and learning biased models. To cope with this issue we considered applying data augmentation to the data sets. Initial results on the training set showed that augmentation was relevant to obtain good results on sub-task A, while we did not need to apply it to sub-task B, despite the class unbalance.

To clean the data sets, we employed basic text transformation and regular expressions to:

- Convert all text to lowercase to ensure consistency and reduce the vocabulary size.
- Clean the data by removing noise such as emoticons, slang, and special characters with regular expressions. This step helps to improve the quality of the text samples.
- Clean data by removing specific patterns from the text, including dates, texts between brackets, links, emails, and multiple spaces, using regular expressions.

Additionally, we applied data augmentation to sub-task A, to increase the number of samples. The method of choice was *back-translation*, which involves translating a text sample from the source language to another language and then translating it back. This process preserves the original text's semantics while potentially altering the syntax, generating synthetic samples.

3. Model

In this section, we describe the architecture of the classifiers we built using *Transformer neural networks* [10] and the training process we followed to prepare our models for evaluation.

3.1. Architecture

To solve both ACTI sub-tasks, we used the same Transformer-based classification architecture, changing only the target classes from one task to the other. We explored different *Transformer Encoder* neural networks, namely BERT [11], pre-trained on different data sets. We also explored individual and combined applications of the classifiers. We visualise the single-model and ensemble-model pipelines in Figure 2.

Each BERT-based classifier takes as input a sequence of tokens, extracted from the pre-processed text. The sequence starts with a classification token `[CLS]` and is concluded by an end-of-sequence token `[SEP]`, introduced during the tokenisation process. To classify the input piece of text, we retrieve the *contextual embedding* computed by the transformer *hidden layers* in correspondence of the `[CLS]` input token and use it to feed a linear classifier. The final classification layer outputs the probability distribution of the input piece of text to belong to one of the possible classes. We reported the entire process in Figure 2a.

To improve the classification results and take the best from the trained models, we considered also creating an *ensemble* [12, Chapter 16]. For each task, we aggregated the predictions of the individual models. To aggregate the predictions, we froze the fine-tuned classifiers and learned a separate *Logistic Regression* classifier on top of the Transformer models. The Logistic Regression classifier takes as input the probability distributions predicted by individual models and compute a new output probability combining the previous. The entire ensemble pipeline is represented in Figure 2b.

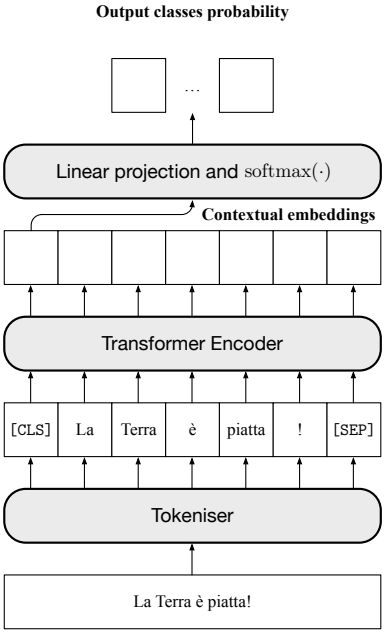
3.2. Training

To effectively train our models, we adopted *5-fold cross-validation* to find the best hyperparameters for each of the considered models and each task. We preferred this approach to the usual train-validation split to make the best out of the available data. Given the best hyperparameters combination, we retrained the model on the entire training data set.

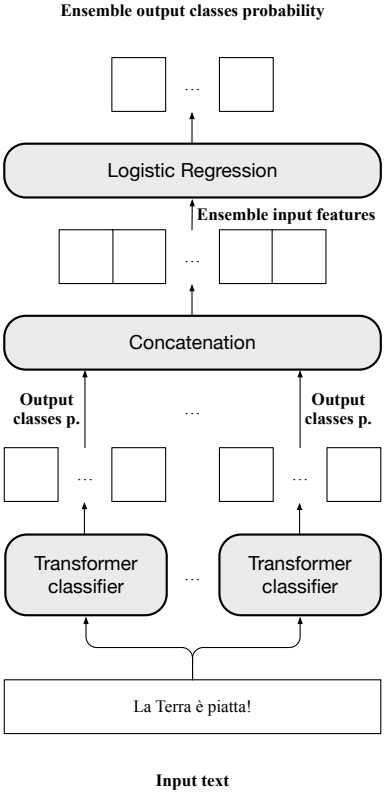
We fine-tuned two variants of BERT base (110M parameters):

- Italian BERT (uncased)[2], pre-trained on Italian.

[2]Model card:
https://huggingface.co/dbmdz/bert-base-italian-xxl-uncased



(a) Single Transformer classifier.



(b) Ensemble classifier.

Figure 2: BERT-based classifiers architecture.

- Multilingual BERT (uncased)[3]; pre-trained on several languages, including Italian.

We used the implementations available in the *Transformers* library from *Hugging Face* [13]. Each configuration was trained using the *Adam* optimiser [14], a linear learning rate schedule and a batch size of 8.

For each of these models, during cross-validation, we varied the learning rate and the number of epochs. Additionally, we explored regularisation: we evaluated models with and without *dynamic masking*. We varied the learning rate in $\{1\times10^{-5},2\times10^{-5},3\times10^{-5},5\times10^{-5}\}$, and the number of epochs in $\{2,3,4\}$. Dynamic masking applies the same kind of masking BERT uses during pre-training, randomly corrupting the input sequence by masking the tokens.

We adopted 5-fold cross-validation with the ensemble as well. Referring to the implementation of Logistic Regression available in the *Scikit-Learn* library [15], we explored values for the following hyperparameters: regularisation strength (L_2 regularisation), number of iterations, and solver. We varied the inverse of the regularisation strength in $\{10^{-3},10^{-2},10^{-1},1,10,10^{2},10^{3}\}$, the maximum number of iterations in $\{20,50,100,200,500,1000\}$, and we tried all solvers apart from the Newton-Cholesky one. Additionally, we weighed the classes with a weight inversely proportional to class frequencies, to obtain a more balanced classifier.

4. Results

In this section, we explain how we evaluated the models proposed for each sub-task, present the results obtained on each task, and provide comments on these results. In both cases, we focus in the results of the ensembles, since they peeform better than individual models in both cases.

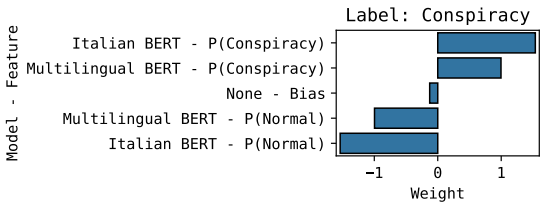
4.1. Evaluation

We evaluated the classification models using the F_1 score. For the multitask settings, we computed the macro average of the scores to account for potential class distribution imbalances.

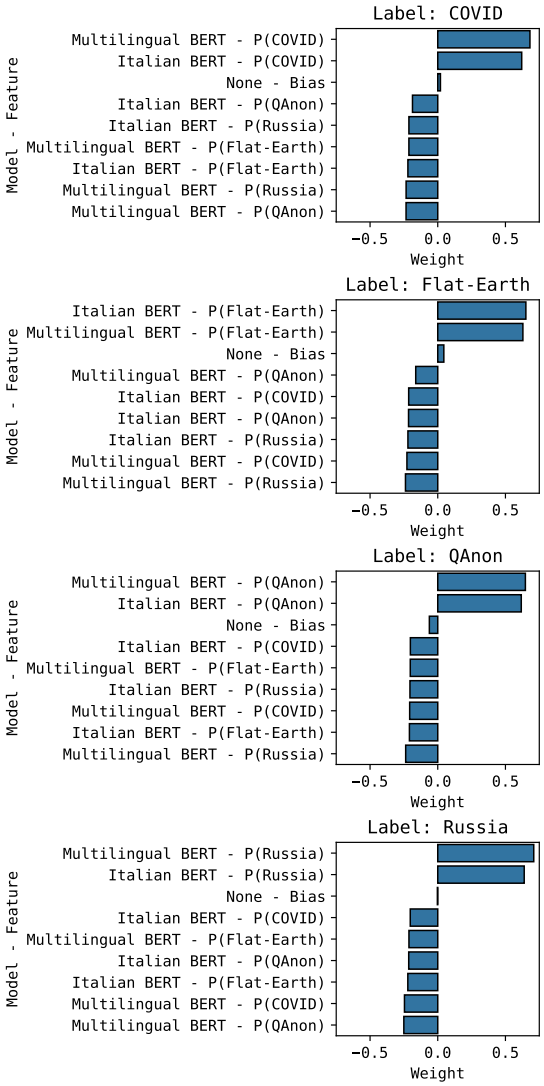
We reported the F_1 scores on the test set in table Table 2. In addition to the results of the submitted models, we included some additional scores for comparison and to provide further insight into the results. The F_1 scores are computed on 70% of the test data for sub-task A 50% of the test data for sub-task B via the *Kaggle* platform[4] (which hosted the competition), as determined by the authors of the ACTI task for the private leaderboard.

[3]Model card:
https://huggingface.co/bert-base-multilingual-uncased

[4]Website: https://www.kaggle.com



(a) Sub-task A.



(b) Sub-task B.

Figure 3: Feature relevance in the ensembles. Relevance scores are given by the weights learned by the Logistic Regression classifier.

Table 2
Ensemble results on ACTI task. Scores are from the private Kaggle evaluation on 70% of the test data. Baselines were provided by task organisers via Kaggle.

Sub-task	Augmentation	Dynamic Masking	$\mathbf{F}_1$
Submitted			
A	✓	✓	**82.30**
B		✓	**89.83**
Other			
A	✓		77.04
A			81.67
B			87.67
Baselines			
A	n.a.	n.a.	51.07
B	n.a.	n.a.	68.37

Additionally, to get better insights on the behaviour of the two Transformer-based classifiers on the sub-tasks, we analysed the weights learned by the Logistic Regression during the training of the ensemble. The higher the weight, the stronger the contribution of the probability predicted by a classifier to the prediction and, thus, the stronger the relevance of that classifier in the ensemble. To this end, we visualised all the weights of the two Logistic Regression models in Figure 3.

4.2. Sub-task A

For Sub-task A, the ensemble model achieved a test F_1 score of 82.30% (see Table 2). This result highlights the effectiveness of combining the predictions from individual models to improve overall performance.

The best model configuration for sub-task A, involved fine-tuning with the following hyperparameters. Italian BERT: learning rate of 2×10^{-5}, number of epochs of 4, and dynamic masking enabled. Multilingual BERT: learning rate of 2×10^{-5}, number of epochs of 4, and dynamic masking enabled. Logistic Regression (ensemble): inverse of the regularisation strength of 10^{-2}, maximum number of iterations of 20, *Newton-CG* solver.

Comparing it with other configurations, we observe that the ensemble model outperformed other approaches, such as using augmentation alone (77.04%) or not applying any augmentation nor masking (81.67%). Furthermore, compared to the provided baseline F_1 score of 51.07%, the proposed ensemble model shows a substantial improvement, highlighting the effectiveness of our approach.

Concerning feature relevance analysis in the ensemble, from Figure 3a, we can see that the ensemble gives a higher weight to the prediction of the Italian BERT model, rather than the multilingual one. This hints that for this specific task of detecting whether the text concerns a conspiracy theory or not, having a language-specific model may be the better solution. However, the ensemble improves over both single models, thus the multilingual model is contributing to the correct classification as well.

4.3. Sub-task B

For Sub-task B, the ensemble model achieved a test accuracy score of 89.83% (see Table 2). This result highlights again the effectiveness of the ensemble approach in capturing task-specific patterns and making accurate predictions.

The best model configuration for sub-task B, involved fine-tuning with the following hyperparameters. Italian BERT: learning rate of 3×10^{-5}, number of epochs of 2, and dynamic masking enabled. Multilingual BERT: learning rate of 3×10^{-5}, number of epochs of 2, and dynamic masking enabled. Logistic Regression (ensemble): inverse of the regularisation strength of 10^{-3}, maximum number of iterations of 20, *Newton-CG* solver.

Comparing it with other configurations, we observe that the ensemble model outperformed the configuration without dynamic masking, which achieved an accuracy score of 87.67%. This indicates that dynamic masking played a crucial role in improving the model's performance. When comparing the ensemble model's accuracy score with the provided baseline accuracy score of 68.37%, we observe a significant performance boost, underscoring the effectiveness of our approach.

Concerning feature relevance analysis in the ensemble, from Figure 3b), we can see that, differently from sub-task A, here both models contribute equally to the prediction. In fact, the values of the weights associated with the same input probability and the same output class models are close for the different models.

5. Discussion

In this report, we described our approach to training Transformer-based classification models for conspiracy theory identification. We trained and evaluated our models on the two sub-tasks of the ACTI data set, an Italian benchmark for conspiracy theory identification. The first task involved binary text classification to determine whether a piece of text is about a conspiracy theory or not, while the second task focused on multi-class classification to identify the specific conspiracy theory referenced in a piece of text.

Given the limited resources available, including the data set size and computational power, we were unable to explore all possible avenues. Additionally, the availability of pre-trained Italian Language Models is also limited. Most Italian models are part of multilingual models rather than dedicated Italian models, and the available Italian-only models are smaller if compared to English ones (for example). However, this provided us with the opportunity to train potentially multilingual models for conspiracy theory identification, although we did not test this approach on other languages in our current study.

The results obtained from our models are promising. For Sub-task A, our ensemble model achieved a test F_1 score of 82.30%, outperforming both the other configurations we explored and the provided baseline F_1 score of 51.07%. Regarding Sub-task B, our ensemble model achieved a test F_1 score of 89.83%, surpassing the other configurations we tested and the provided baseline F_1 score of 68.37%. This highlights the effectiveness of combining the predictions from individual models to improve overall performance on this task.

Moving forward, we aim to explore the application of end-to-end text generation models for conspiracy theory identification. Current research suggests that LLMs can be effectively employed for text classification tasks by concatenating the text to classify with a question asking for the class and triggering text generation. We plan to leverage one of these multilingual LLMs with a combination of *prompting* and *in-context learning*, enabling *zero-shot* to *few-shots* learning.

Overall, our study contributes to the understanding of conspiracy theory identification using Transformer-based models. The achieved results show the potential of these models in accurately classifying conspiracy-related texts, and future investigations can explore additional approaches to further enhance performance.

References

[1] G. Russo, N. Stoehr, M. H. Ribeiro, Acti at evalita 2023: Overview of the conspiracy theory identification task, arXiv preprint arXiv:2307.06954 (2023).

[2] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi, Evalita 2023: Overview of the 8th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[3] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, J. Wen, A survey of large language models, CoRR abs/2303.18223 (2023). URL: https://doi.org/10.48550/arXiv.2303.18223. doi:10.48550/arXiv.2303.18223. arXiv:2303.18223.

[4] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, Z. Kenton, S. Brown, W. Hawkins, T. Stepleton, C. Biles, A. Birhane, J. Haas, L. Rimell, L. A. Hendricks, W. Isaac, S. Legassick, G. Irving, I. Gabriel, Ethical and social risks of harm from language models, CoRR abs/2112.04359 (2021). URL: https://arxiv.org/abs/2112.04359. arXiv:2112.04359.

[5] G. Russo, M. Horta Ribeiro, G. Casiraghi, L. Verginer, Understanding online migration decisions following the banning of radical communities, in: Proceedings of the 15th ACM Web Science Conference 2023, WebSci '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 251–259. URL: https://doi.org/10.1145/3578503.3583608. doi:10.1145/3578503.3583608.

[6] G. Russo, L. Verginer, M. H. Ribeiro, G. Casiraghi, Spillover of antisocial behavior from fringe platforms: The unintended consequences of community banning, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 17, 2023, pp. 742–753.

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[8] G. Russo, C. Gote, L. Brandenberger, S. Schlosser, F. Schweitzer, Helping a friend or supporting a cause? disentangling active and passive cosponsorship in the U.S. congress, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 2952–2969. URL: https://aclanthology.org/2023.acl-long.166.

[9] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, O'Reilly, 2009. URL: http://www.oreilly.de/catalog/9780596516499/index.html.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017,

Long Beach, CA, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[11] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[12] T. Hastie, R. Tibshirani, J. H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer Series in Statistics, Springer, 2009. URL: https://doi.org/10.1007/978-0-387-84858-7. doi:10.1007/978-0-387-84858-7.

[13] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 38–45. URL: https://doi.org/10.18653/v1/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[14] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830. URL: https://dl.acm.org/doi/10.5555/1953048.2078195. doi:10.5555/1953048.2078195.

A. Source Code

The source code developed for the challenge is available via *GitHub* at the following link: https://github.com/MichaelVitali/Evalita2023.