Giobergia at Multi-Task Transformer Tuning for Joint Conspiracy Theory Detection and Classification

Flavio Giobergia

Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy

Abstract

Conspiracy theories, prevalent in contemporary society, often propagate misinformation and distrust, impacting public opinion and decision-making processes. In this paper, we present an automated approach to detect and classify conspiracy theories in Italian to address the Automatic Conspiracy Theory Identification (ACTI) task. Our methodology leverages a transformer-based architecture trained on a multi-task problem to tackle the challenging task of conspiracy theory identification. Through this multi-task learning framework, we aim to build a single model capable of addressing both the detection and the classification tasks simultaneously. We show that tackling both problems in a multi-task setting results in improved performance w.r.t. simple transformer-based solutions.

Keywords

transformers, conspiracy theory, deep learning, natural language processing

1. Introduction

Conspiracy theories have become a pervasive phenomenon, spreading through various communication channels, including social media platforms and online communities. These theories often involve the belief in secret plots or covert actions orchestrated by influential entities, which aim to manipulate events, control narratives, or conceal the truth. While some conspiracy theories may seem harmless or merely speculative, many have far-reaching consequences, potentially eroding trust in institutions, sowing discord among communities, and hindering informed decision-making processes: in recent years some of these conspiracies have been involved in the Capitol Hill attack (QAnon [1]) and hindered the efforts made toward the mitigation of the COVID-19 pandemic (e.g. resulting in lower vaccination and social distancing responses [2]), thus effectively jeopardizing lives.

The prevalence and impact of conspiracy theories necessitate effective methods for their detection and classification. Manual identification and analysis of conspiracy theories are time-consuming, resource-intensive, and subject to bias. Mainstream platforms (e.g. Reddit, Facebook) need to apply moderation policies at the community level. Given the limitations of the currently adopted approaches [3, 4], a more suitable methodology is required to address an efficient identification and classification of conspiracy theories that are constantly evolving. For these reasons, the Automatic Conspiracy Theory Identification (ACTI) [5], one of the tasks of EVALITA 2023 [6], is aimed at advancing the automation of these tasks for the Italian language. In particular, the task identifies two main goals: detecting whether a short piece of text is conspiratorial in nature or not and, if so, to which kind of conspiracy theories it conforms.

The detection of conspiracy theories in online contents is not a new one: other works have focused, for example, on the detection of conspiracies and misinformation related to COVID-19 [7], whereas the authors in [8, 9] propose building an automated pipeline for the detection of conspiracy theories and their diffusion by analyzing a network of actors and the interactions occurring within. To the best of our knowledge, ACTI is the first challenge on the detection of conspiracy theories with a specific focus on the Italian language. It should be pointed out, however, that the problem under study often spans across multiple languages, given the often English-centric origin of such theories. Various works have focused on the cross-lingual detection of adjacent topics, such as hate speech [10] and fake news [11], or sentiment analysis [12, 13], thus offering useful building blocks for future possible applications.

In this paper we propose leveraging Natural Language Processing (NLP) techniques to address the ACTI task. We employ a pre-trained transformer-based model fine-tuned to address both subtasks simultaneously. The source code for the proposed method is openly available on GitHub[1]

The remainder of this paper is organized as follows: Section 2 provides an overview of the subtasks and the data, Section 3 introduces the proposed method to address both subtasks and Section 4 presents the results obtained. Finally, Section 5 draws conclusions based on

EVALITA 2023: 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Sep 7 – 8, Parma, IT

flavio.giobergia@polito.it (F. Giobergia)

0000-0001-8806-7979 (F. Giobergia)

[1] https://github.com/fgiobergia/EVALITA-ACTI-2023

the results achieved.

2. Problem overview

The ACTI task is focused on the detection and classification of conspiracy theories based on short text posts. The data is collected from Telegram Channels and is entirely in Italian, with the exception of some short citations in English. In particular, two subtasks are proposed:

- *Subtask A*: the main goal is to detect whether a post is conspiratorial in nature or not, so the problem is framed as a binary classification one. A training set $\mathcal{D}_A$ containing a total of 1,841 posts is made available, with approximately 50% of the messages within being conspiratorial and 50% not. The metric used for the evaluation of this subtask is the macro F_1 score (i.e. the unweighted average F_1 score for the two classes). The unlabelled test set is instead comprised of a total of 460 posts.
- *Subtask B*: in this task, the posts provided can be classified as conforming to one of four conspiracy theories, namely Covid, QAnon, Flat Earth, Russia (more details on each conspiracy theory are provided in the task paper). Each post should be classified as belonging to either one of these categories[2] based on its contents. The training set $\mathcal{D}_B$ is comprised of 810 samples, with an approximate split of 50/30/10/10 among the Covid, QAnon, Flat Earth, Russia classes. The unlabelled test set contains 300 unlabelled samples. The macro F_1 score is used as the main evaluation metric for this problem.

Finally, we note that there are some overlaps in the posts contained in the two subtasks. While this is not, in general, a problem when the two training sets overlap (although it needs to be kept into account for a multi-task solution, to avoid data leakage during validation), we note that the test set of Subtask A has 164 posts in common with the training set and 66 with the test set of Subtask B, for a total of 230 posts that can be easily assumed to be conspiratorial in nature. In the spirit of a fair competition this information has obviously not been used in any way during the competition.

3. Methods

Let X be the set of all possible inputs (i.e. Telegram posts) and $\mathcal{C}$ be the set of possible conspiracies. We can formalize Subtask A as building a model $f_A : X \to [0, 1]$ that estimates the probabilities for a post to be conspiratorial in nature and Subtask B as a model $f_B : X \to [0, 1]^{|\mathcal{C}|}$ that estimates the distribution of probabilities across conspiracy classes (the following hold for all x: $f_B(x)_i \geq 0$ and $\sum_i f_B(x)_i = 1$). We note that, although the two functions produce different results, they work on the same inputs. We thus propose building an encoding function $e : X \to \mathbb{R}^d$ that projects the inputs into a shared latent space, and two head functions h_A, h_B such that $f_A = h_A \circ e$ and $f_B = h_B \circ e$. In other words, we aim to build robust shared representations by framing the problem as a multi-task one.

By introducing a common encoder, we can use two simple classification models for the heads, deferring the complexity of the entire model to $e(\cdot)$. In particular, we use a pre-trained transformer which is fine-tuned on the conspiracy theory detection and classification tasks simultaneously.

To solve this multi-task problem we adopt a loss function that evaluates the model based on the two separate targets. In particular, detecting whether a message is conspiratorial in nature is a binary problem that can be evaluated in terms of binary cross-entropy. For a given message x_i with binary conspiratorial label $y_i^{(A)}$, we define the conspiratorial loss function as:

$$\mathcal{L}^{(A)}(x_i, y_i^{(A)}) = -y_i^{(A)} log\sigma(f_A(x_i)) - (1 - y^{(A)}) log(1 - \sigma(f_A(x_i))) \quad (1)$$

Where $\sigma(\cdot)$ is the sigmoid function. By contrast, the loss function for the conspiracy theory classification is a multi-class classification problem. As such, we aim to minimize the cross-entropy between the predicted probability distribution and the ground truth value $y_i^{(B)}$, defined as follows:

$$\mathcal{L}^{(B)}(x_i, y_i^{(B)}) = \sum_j y_{ij}^{(B)} softmax(f_B(x_i))_j \quad (2)$$

We note that the datasets available for the two tasks are not the same, although some overlaps occur. Because of this, the two losses $\mathcal{L}^{(A)}$ and $\mathcal{L}^{(B)}$ cannot be computed for all points. In particular, points that are not conspiratorial in nature ($y^{(A)} = 0$) are not associated with any conspiracy theory, thus making the term $\mathcal{L}^{(B)}$ meaningless. Similarly, points that are conspiratorial in nature but only appear as a part of the dataset for Subtask A are not annotated with a ground truth label regarding the conspiracy theory to which they conform. Because of this, all points that belong exclusively to the dataset for Subtask A can only be evaluated in terms of $\mathcal{L}^{(A)}$. Instead, all points exclusively belonging to Subtask B are guaranteed to be conspiratorial in nature. Because of this,

[2]We note that some conspiracy theories are strongly related to one another (e.g. QAnon and Covid), so the identification of a single class may at times lead to ambiguous results.

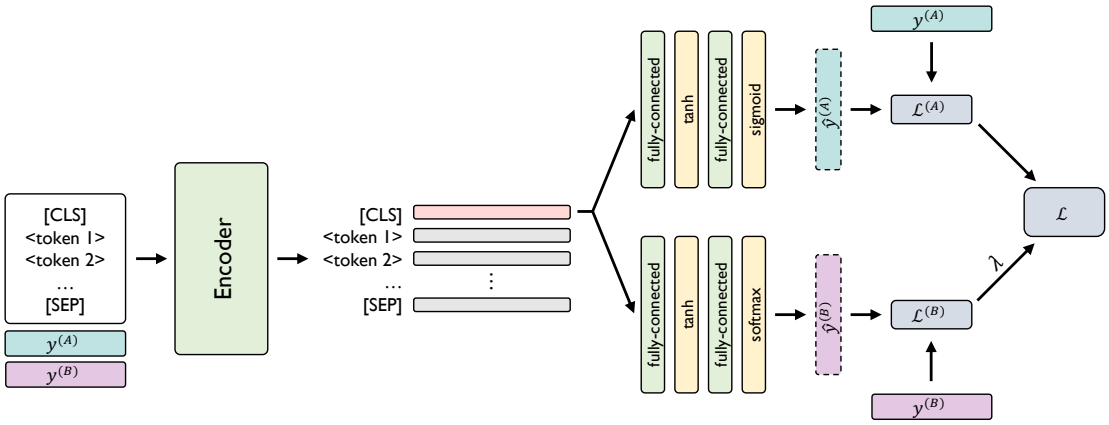

Figure 1: Architecture used for the proposed solution. In green are the trainable portions of the pipeline. The encoder used is transformer-based. The output vector for the [CLS] token is used as representative of the entire input sentence.

we infer for those points that $y^{(A)} = 1$. This consideration makes it reasonable to assume that the multi-task approach proposed should be particularly beneficial in terms of improvements on Subtask A.

The overall loss is obtained as a weighted sum of the above terms:

$$\mathcal{L} = \mathcal{L}^{(A)} + \lambda \mathbb{1}(x_i \in \mathcal{D}_B)\mathcal{L}^{(B)} \tag{3}$$

Where λ is used to balance the important that the two loss terms play in the overall predictions, and $\mathbb{1}(\cdot)$ is a selector that applies the second loss term only for terms where that portion can be applied meaningfully.

Figure 1 summarizes the architecture for the proposed methodology from inputs to the computation of the overall loss. Since we make use of transformer-based encoders, we note that we adopt the final hidden state corresponding to the [CLS] token as representative of the semantic contents of the entire message [14].

4. Results

The experimental section aims to evaluate the effectiveness of our proposed approach for detecting and classifying messages containing conspiracy theories. We present the main choices made in terms of encoder adopted, as well as results in terms of choice of hyperparameters.

4.1. Model evaluation

As already discussed, the models are evaluated on both subtasks by means of the macro F_1 score metric. This metric is particularly useful when evaluating models on unbalanced classes, as is for example the case with Subtask B.

We report some of the main results (i.e. encoders choice and multi-task vs single tasks comparison) in terms of performance on the final test set, whereas the other results on definition of λ in terms of performance on a validation set that has been obtained as a 20% hold out from the available dataset.

The test set made available for the competition is split intro a public and a private subsets (used for different parts of the competition itself). Both scores have been made available, for each submission, at the end of the challenge. For Subtask A, the private/public test is a 70/30 split, whereas for Subtask B the private/public test split is approximately 50/50.

For conciseness, we report an aggregated result that covers both private and public scores simultaneously with the aforementioned weights: more specifically, if $F^{(A)}_{1,public}$ is the score obtained on the public set for subtask A and $F^{(A)}_{1,private}$ is the score obtained on the private set, we report $F^{(A)}_1 = 0.3\, F^{(A)}_{1,public} + 0.7\, F^{(A)}_{1,private}$. Similarly, we report $F^{(B)}_1 = 0.5\, F^{(B)}_{1,public} + 0.7\, F^{(B)}_{1,private}$. We note that, despite assigning the same weights to the same partitions, the reported metrics are not the same as computing the F_1 scores on the entire test set – an operation that cannot be performed with the information at hand.

The overall metric used to evaluate the performance in the competition is a weighted average of the performance obtained on the two tasks, with weights 0.6 and 0.4 for Subtasks A and B respectively. Thus, we additionally report the overall score $F_1 = 0.6\, F^{(A)}_1 + 0.4\, F^{(B)}_1$.

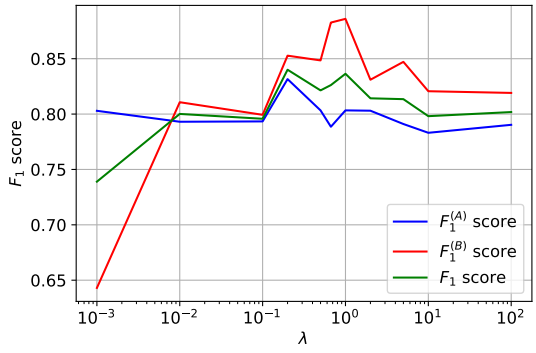

Figure 2: Performance of the model in terms of F_1 score (A, B, overall) as the parameter λ varies.

Model	$\mathbf{F_1^{(A)}}$	$\mathbf{F_1^{(B)}}$	$\mathbf{F_1}$
BERT-Italian-XXL-uncased	0.8930	0.8475	**0.8748**
BERT-Italian-XXL-cased	0.8686	0.8363	0.8556
BERT-Italian-base-uncased	**0.8953**	0.8082	0.8605
BERT-Italian-base-cased	0.8867	0.7920	0.8488
BART-IT-WITS	0.8494	0.8431	0.8469
BART-IT-IlPost	0.8504	0.8373	0.8452
BART-IT-FanPage	0.8327	**0.8490**	0.8393
BART-IT	0.8513	0.8322	0.8437
ELECTRA-XXL-discriminator	0.8886	0.8260	0.8635
ELECTRA-XXL-generator	0.8622	0.7693	0.8251

Table 1
Macro F_1 scores for the various encoding models used. Performance measured on subtasks A and B separately, as well as with the overall score (computed as $F_1 = 0.6\ F_1^{(A)} + 0.4\ F_1^{(B)}$). In **bold** are the best performing models for each metric. Underlined are the second best models.

4.2. Hyperparameters tuning

The main hyperparameters to be configured for the proposed pipeline is the λ coefficient. Other parameters that are generally important, but not specific to this precise context (e.g., number of training epochs, learning rate, layer sizes, optimizer) will not be covered in detail and can be found as a part of the provided source code.

The choice of a valid value for λ is quite specific to the conspiracy theory detection problem, as it represents the trade-off coefficient between the capability of detecting conspiratorial contents and being able to correctly assign them. Figure 2 shows how the performance of the model (based on BERT-Italian-XXL-uncased – as explained in the next subsection) varies as λ increases. Although the best value in terms of overall F_1 score is observed for $\lambda = 0.2$, we identify $\lambda = 1$ as being a more robust choice, considering the overall optimal behavior in the interval centered around that value.

4.3. Encoder choice

In recent years a wide variety of transformer-based models have been introduced for various languages, including Italian. An *a priori* choice regarding the most suitable model is not trivial to make. Therefore, we ran a benchmark study to assess how well various models behave on both tasks. In particular, we compare results obtained for the following models:

- **Italian ELECTRA** [15], an encoder-only architecture based on the ELECTRA model [16]; a method trained on detecting corruptions of the input introduced using a generator network. We test both the generator and discriminator versions of Italian ELECTRA.
- **BART-IT** [17], a sequence-to-sequence model based on the BART [18] architecture that is specifically tailored to the Italian language. BART-IT has been shown to outperform other state-of-the-art architectures on various tasks. We note that, on top of the original BART-IT model, three additional versions have been fine-tuned on abstractive summarization tasks on various datasets: FanPage, IlPost [19] and WITS (Wikipedia for Italian Text Summarization) [20]. Since these three data sources have rather different scopes and styles, we assess the quality of the various fine-tuned versions.
- **Italian BERT** [21], a BERT-based model [14] trained on a recent Wikipedia dump as well as data from the OPUS corpora[3] collection. We use both a cased and uncased version as well as a base and an XXL ones.

All models have been fine-tuned for a total of 11 epochs, considering that no meaningful improvement in performance on the validation set has been observed after this point. To reduce the computational cost, we only attempted a value for $\lambda = 1$ during the training of all models.

Table 1 presents the results obtained on the 10 models that have been tested. The best-performing model, in most cases, is BERT-Italian-XXL-uncased. The rest of the experiments presented will be performed using this encoder only, as a way to reduce the computational cost required.

4.4. Multi-task effect

Table 2 shows the results that are achieved by the multi-task model in contrast to the ones that can be obtained

[3] https://opus.nlpl.eu/

Task	$\mathbf{F_1^{(A)}}$	$\mathbf{F_1^{(B)}}$	$\mathbf{F_1}$
Multi-task	**0.8930**	0.8475	**0.8748**
A only	0.8370	-	0.8491*
B only	-	**0.8672**	0.8491*

Table 2
Results in terms of F_1 score when addressing the problem with a multi-task approach, or as separate subtasks. In **bold** are the best performing models for each metric. (* The overall results for "A only" and "B only" are obtained by merging the results obtained on the two subtasks separately.)

Submission	$\mathbf{F^{(A)}_{1,private}}$	$\mathbf{F^{(A)}_{1,public}}$	$\mathbf{F^{(B)}_{1,private}}$	$\mathbf{F^{(B)}_{1,public}}$
Submitted	0.8371	0.8389	0.8470	0.8360
Proposed	0.8907	0.8984	0.8805	0.8145

Table 3
Results in terms of macro F_1 score on the private and public sets, for both subtasks. Both the performance that have been submitted at the end of the challenge ("Submitted") and the best ones achieved ("Proposed") are shown.

by training the same model on only one of the tasks at a time. As expected, we observe a significant improvement in performance for Subtask A, whereas the performance of Subtask B does not benefit from the introduction of a multi-task approach. This can be explained in terms of benefits that are introduced by the multi-task loss: while points belonging to $\mathcal{D}_B$ could all be labelled as being conspiratorial (thus enhancing the dataset available for subtask A), points in $\mathcal{D}_A$ are not beneficial to the improvement of Subtask B (as they are either non-conspiratorial, or the conspiracy theory to which they conform is unknown).

4.5. Competition performance

The pipeline presented in this work is similar to the one used to take part to the competition, with some minor changes mainly regarding the adoption of BERT-Italian-XXL-uncased instead of BART-IT, as well as *not* adopting a different convolutional model in parallel. For the sake of completeness, the private scores obtained during the challenge are reported in Table 3, along with the best ones obtained with the proposed method.

5. Conclusions

In this work we presented a transformer-based multi-task approach to addressing a joint detection and classification problem. We have shown the importance of choosing the most suitable encoding model, as well as the benefits of adopting a multi-task approach. We highlighted how the current framing of the multi-task problem is only beneficial to one of the subtasks. As a future work, we aim to reframe the problem through a semi-supervised approach that allows for the pseudo-labelling of all points, thus aiming to improve the performance uniformly across subtasks.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013), with partial support from SmartData@PoliTO center on Big Data and Data Science. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] S. Moskalenko, C. McCauley, Qanon, Perspectives on Terrorism 15 (2021) 142–146.

[2] K. Bierwiaczonek, A. B. Gundersen, J. R. Kunst, The role of conspiracy beliefs for covid-19 health responses: A meta-analysis, Current Opinion in Psychology (2022) 101346.

[3] 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.

[4] 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.

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

[6] 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.

[7] S. Shahsavari, P. Holur, T. Wang, T. R. Tangherlini, V. Roychowdhury, Conspiracy in the time of corona: automatic detection of emerging covid-19 conspiracy theories in social media and the news, Journal of computational social science 3 (2020) 279–317.

[8] T. R. Tangherlini, S. Shahsavari, B. Shahbazi, E. Ebrahimzadeh, V. Roychowdhury, An automated pipeline for the discovery of conspiracy and conspiracy theory narrative frameworks: Bridgegate, pizzagate and storytelling on the web, PloS one 15 (2020) e0233879.

[9] 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.

[10] E. W. Pamungkas, V. Basile, V. Patti, A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection, Information Processing & Management 58 (2021) 102544.

[11] J. Köhler, G. K. Shahi, J. M. Struß, M. Wiegand, M. Siegel, T. Mandl, M. Schütz, Overview of the clef-2022 checkthat! lab task 3 on fake news detection, Working Notes of CLEF (2022).

[12] F. Giobergia, L. Cagliero, P. Garza, E. Baralis, Cross-lingual propagation of sentiment information based on bilingual vector space alignment., in: EDBT/ICDT Workshops, 2020, pp. 8–10.

[13] G. Russo, N. Hollenstein, C. C. Musat, C. Zhang, Control, generate, augment: A scalable framework for multi-attribute text generation, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 351–366. URL: https://aclanthology.org/2020.findings-emnlp.33. doi:10.18653/v1/2020.findings-emnlp.33.

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

[15] R. Guarasci, A. Minutolo, E. Damiano, G. De Pietro, H. Fujita, M. Esposito, Electra for neural coreference resolution in italian, IEEE Access 9 (2021) 115643–115654.

[16] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).

[17] M. La Quatra, L. Cagliero, Bart-it: An efficient sequence-to-sequence model for italian text summarization, Future Internet 15 (2023). URL: https://www.mdpi.com/1999-5903/15/1/15. doi:10.3390/fi15010015.

[18] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461 (2019).

[19] N. Landro, I. Gallo, R. La Grassa, E. Federici, Two new datasets for italian-language abstractive text summarization, Information 13 (2022) 228.

[20] S. Casola, A. Lavelli, Wits: Wikipedia for italian text summarization., in: CLiC-it, 2021.

[21] S. Schweter, Italian bert and electra models, 2020. URL: https://doi.org/10.5281/zenodo.4263142. doi:10.5281/zenodo.4263142.