Cicognini at ACTI: Analysis of techniques for conspiracies individuation in Italian

Giacomo Cignoni[1], Alessandro Bucci[1]

[1]University of Pisa

Abstract

This report illustrates methods and results for solving SubtaskA (conspiracy detection) and SubtaskB (conspiracy topic classification) of EVALITA 2023 ACTI challenge. We employed different transformer-based models and an original method based on tf-idf. Results shows top performance scores over 80% for both subtasks.

Keywords

Conspiracy Theory, Content Moderation, Large Language Models, Computational Social Science

1. Introduction

We decided to cover the EVALITA 2023 challenge "Automatic Conspiracy Theory Identification" or ACTI for short [2]. This challenge is about classifying whenever an Italian message is conspiratorial or not and, if positive, what type of conspiracy is about. Therefore the challenge is subdivided is 2 subtasks:

- **Conspiratorial Content Classification**: the model must recognize if a telegram post is conspiratorial or not.
- **Conspiracy Category Classification**: the model must discriminate to which conspiracy theory a post belongs from a list of 4 possible conspiracy topics:
  1. Covid-Conspiracy
  2. Qanon-Conspiracy
  3. Flat Earth-Conspiracy
  4. Pro-Russia Conspiracy

2. Related works

Conspiratorial content has been raising on the internet over the past years such that some has define it as a "Golden Age of Conspiracy" [3]. Indeed mainstream platforms have tried to moderate the diffusion of online communities with the implementation of content moderation known as deplatforming. However, there have been a lot of discussion regarding the efficacy of such interventions [4, 5, 6].. Indeed, some identified the presence of spillover of toxic behaviour [7] and the the presence of a radicalization process after the application of content moderation [8]. Therefore, the need for automatic models that can detect the diffusion of troublesome (or more specifically) conspiratorial content has become crucial. Transformer based models have revolutionized modern natural language processing [9, 10, 11, 12]. Indeed, they are the current state of the art models in most NLP tasks spanning different fields from politics [13, 14], conflict prediction [15], and, of course, hate speech detection [16, 17, 18].[19]. In particular finetuning of BERT[20] based models for classification tasks such as sentiment analysis or topic detection has been widely studied and its effectiveness proved with multiple benchmarks [21]. The usage of machine learning techniques for detecting conspiracy theories has been studied mainly in regard to social media texts extracted in the English language, although also classification on different topic of the conspiracies has been considered [22, 23].

3. Datasets

The 2 provided datasets are a collection of labeled Italian Telegram's messages. Both datasets were relatively clean in regard to the text, so heavy preprocessing was not needed.

3.1. Subtask A dataset

More specifically for Subtask A, the training dataset is a .csv file containing:

- **id**: unique post identifier.
- **comment_text**: the text of the telegram's message.
- **conspiratorial**: a binary label that indicates if the message is conspiratorial or not.

The training dataset is composed by 1842 samples, of which 925 with a positive conspiratorial label and 917

EVALITA 2023: 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Sep 7 – 8, Parma, IT [1]
g.cignoni3@studenti.unipi.it (G. Cignoni); a.bucci12@studenti.unipi.it (A. Bucci)

with a negative conspiratorial label. The hidden test set is composed by 460 samples instead.

3.2. Subtask B dataset

And for Subtask B, the training dataset is a .csv file containing:

- **id**: unique post identifier.
- **comment_text**: the text of the telegram's message.
- **conspiracy**: a label going from 0 to 3 indicating which conspiracy topic the message is about.

The training dataset is composed by 810 samples, with the following conspiracy label distribution: 435 Covid-Conspiracy, 242 Qanon-Conspiracy, 76 Flat Earth-Conspiracy, 57 Pro-Russia Conspiracy. The hidden test set is composed by 300 samples instead.

4. Models

Due to the nature of the tasks, we mainly decided to try different types of transformers based models for both subtasks, in order to capture the semantics and the general matter of the message itself. This is concatenated with a densely connected neural network in order to classify what the specific task is asking.

More specifically a Transformer as described in "Attention is all you need" [10], is composed of encoder-decoder structure composed by multiple modules stacked Nx times on top of each other like in Figure 1 where each module is mainly consisted of Multi Head Attentions and Feed Forward layers. In this architecture, the inputs and the outputs (target sentences) are embedded (the outputs need a right shift before usage) into an n-dimensional space because we cannot use the strings directly.

Here we present the selected transformer-based models for the tasks. Those were selected after a preliminary exploratory phase based on their performance on the validation set.

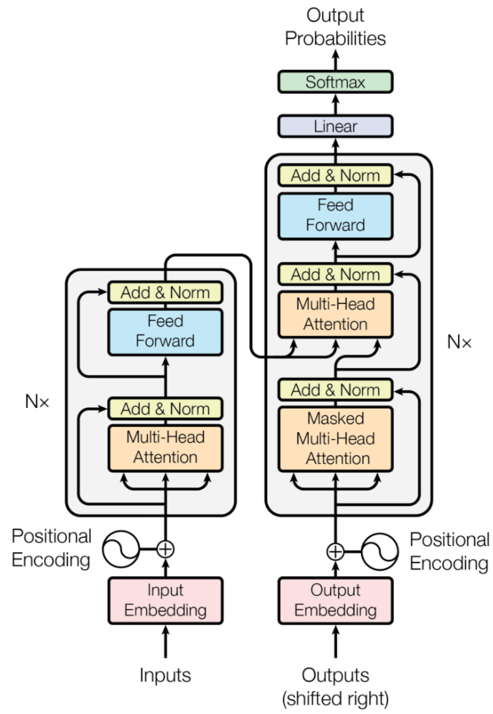

Figure 1: The transformer architecture.

4.1. BERT-xxl

We used the *bert-base-italian-xxl-cased* model[24], which is an Italian pretrained BERT, an encoder-only transformer, variant developed by MDZ Digital Library team. It was pretrained using as source data a Wikipedia dump of various texts from the OPUS corpora collection with a size of 13 GB and more than 2 billion tokens. With the XXL variant, the corpus was extended with the Italian part of the OSCAR corpus, reaching a size of 81 GB and more than 13 billion tokens. This BERT-xxl model has 12 hidden layers, 12 attention heads and a hidden size of 768. We executed fine tuning on the transformer. Classification is executed on the first special output token [CLS] of the transformer

4.2. XLM-RoBERTa

XLM-RoBERTa [25] is a multilingual version of RoBERTa, a transformers model pre-trained in a self-supervised fashion, similarly to BERT, but with a larger corpus and no next sentecnce prediction. *XLM-RoBERTa* was pretrained on 2.5TB of filtered CommonCrawl data containing 100 languages. Specifically, we used the *xlm-roberta-large* variant, which has 24 hidden layers, 16 attention heads and a hidden size of 1024. We executed fine tuning on the transformer. Classification is executed on the first special output token [CLS] of the transformer.

4.3. Llama

LLaMA is an autoregressive language model developed by Meta AI [26], based on a decoder only transformer architecture. We used the 7B variant, the smallest one, which has 7 billions parameters. It was pretrained on 1 trillion tokens from CCNet [67%], C4 [15%], GitHub

[4.5%], Wikipedia [4.5%], Books [4.5%], ArXiv [2.5%], Stack Exchange[2%] sources. The Wikipedia and Books sources are multilingual. Classification is executed on the last output token of the transformer.

We don't use fine tuning on this model due to its size, but only use it to generate sentence embeddings; trainining was only executed on the classification head.

4.4. Topic-specific tf-idf baseline

For Subtask B, considered its nature of topic classification and observing the presence of specific and unique words in each topic, we also developed an original heuristic baseline based on this assumptions. In short, it tries to retrieve the most specific keywords to each topic and extract their distribution in input texts. We recall that the definition of tf-idf for each word i in a set of documents $d \in D$ (in our case each document corresponds to each Telegram message in the dataset) is:

$$tf - idf_{i,j} = tf_{i,j} \times idf_i$$

, with $tf_{i,j} = \frac{n_{i,j}}{|d_j|}$ ($n_{i,j}$ being the number of occurrencies of word i in document j) and $idf_i = log_{10} \frac{|D|}{d:i \in D}$

This method makes use of *topic-specific tf-idf*, which is basically the normalized average tf-idf for each word in respect to the documents of each topic, then divided by the average tf-idf of the same word in the other topics. In mathematical terms, defining T as the set of topics, $avg\_tfidf_{i,t}$ as the average tf-idf for word i and topic t, and $norm\_avg\_tfidf_{i,t}$ as the normalized $avg\_tfidf_{i,t}$ in [0, 100] range, we have:

$$topic\_specific\_tfidf_{i,t} =$$

$$\frac{norm\_avg\_tfidf_{i,t}}{\sum_{t' \in T \setminus t} norm\_avg\_tfidf_{i,t'}}$$

This sccore is calculated only for the training set; for each topic t then we extract the top K $topic\_specific\_tfidf_{k,t}$ words and store them (K is an hyperparameter). Figure 2 shows the top 10 keywords for each topic with their respective score.

Finally, for each input text, we extract the distribution of the previously stored words, thus we obtain a $num\_topics \times K$ distribution vector. This vector is then fed into a Random Forest (RF) model for the final classification. This model is trained with 6-fold Cross-Validation (CV) on the training set.

4.5. Preprocessing

For the transformer-based models, only light preprocessing was applied, only substituting break line characters with spaces and using each transformer tokenizer.

avg_tfidf_covid

	words	avg_tfidf_covid	avg_tfidf_qanon	avg_tfidf_russia	avg_tfidf_flat_earth
992	dos	23.939568	0.000000	0.0	0.0
316	avvers	15.754102	0.000000	0.0	0.0
1602	inocul	15.027083	0.000000	0.0	0.0
1967	mrna	14.798967	0.000000	0.0	0.0
1372	giovan	14.456688	0.000000	0.0	0.0
1039	efficac	13.687523	0.000000	0.0	0.0
1195	farmac	13.044677	0.000000	0.0	0.0
2055	obbligator	12.993569	0.000000	0.0	0.0
2964	sperimental	11.597360	0.000000	0.0	0.0
1832	mascherin	11.391805	0.000000	0.0	0.0

avg_tfidf_qanon

	words	avg_tfidf_covid	avg_tfidf_qanon	avg_tfidf_russia	avg_tfidf_flat_earth
3223	trump	0.020017	98.037534	0.0	0.0
958	dittatur	0.285784	48.449649	0.0	0.0
3822	elezion	0.000000	46.367399	0.0	0.0
1223	femmin	0.032511	39.571809	0.0	0.0
2737	sar	0.245433	30.175185	0.0	0.0
3821	elettoral	0.000000	29.785769	0.0	0.0
3871	fbi	0.000000	27.942181	0.0	0.0
3643	clinton	0.000000	27.854850	0.0	0.0
1252	finc	0.197767	27.317816	0.0	0.0
658	condivid	0.292505	27.184331	0.0	0.0

avg_tfidf_russia

	words	avg_tfidf_covid	avg_tfidf_qanon	avg_tfidf_russia	avg_tfidf_flat_earth
3239	ucrain	0.040250	0.338406	70.648510	0.0
2707	russ	0.025138	0.438085	68.342292	0.0
1992	nat	0.050197	0.191845	37.563651	0.0
4740	zelensky	0.000000	0.252306	30.421714	0.0
1910	milit	0.045818	0.232245	25.686416	0.0
2139	pac	0.051818	0.150653	24.721023	0.0
263	attacc	0.028973	0.412274	24.385013	0.0
4781	azov	0.000000	0.000000	20.561859	0.0
4197	nazist	0.000000	0.639016	18.741174	0.0
4990	polacc	0.000000	0.000000	17.956986	0.0

avg_tfidf_flat_earth

	words	avg_tfidf_covid	avg_tfidf_qanon	avg_tfidf_russia	avg_tfidf_flat_earth
3131	terr	0.051687	0.088669	0.062931	83.105710
4472	satell	0.000000	0.028737	0.000000	59.635025
2262	piatt	0.035781	0.000000	0.000000	58.062935
5160	lun	0.000000	0.000000	0.000000	57.843479
5173	orbit	0.000000	0.000000	0.000000	52.471631
3898	fot	0.000000	0.111569	0.185828	46.121322
4955	nas	0.000000	0.000000	0.048892	45.341877
2278	piu	0.051631	0.187217	0.050303	43.793724
1027	ecco	0.147374	0.230873	0.180290	43.415115
3933	glob	0.000000	0.045826	0.000000	42.738272

Figure 2: This figure reports the top 10 keywords for each of the topics (Covid, Qanon, pro-Russia and Flat-earth in descending order). Keywords are obtained using the *Topic-specific tf-idf* model. For each set of top words of each topic, also the score for the same words of the other topics is shown. It is easy to note that all top words have an high score for their respective topic, but very low ones for other topics.

Instead, for the *Topic-specific tf-idf* model, as the focus are topic specific relevant words, we apply stop word and short words (less than 3 characters) removal, number and punctuation elimination and stemming.

5. Implementation

We used the Python environment for developing the models, using mainly PyTorch, Scikit-Learn and Transformers libraries.

6. Experiments and results

We used an hold-out approach for both subtasks, reserving 20% of the training set for validation for hyperparameter tuning (split with labels ratio preservation). We experimented with retrain on validation found hyperparameters, but with worse results, so we decided to keep the model tested on the validation set as the final model for each configuration.

Model	Learning Rate	LR warmup ratio	Gamma	Head layers sizes
BERT-xxl	[1e-6, 2e-6, **3e-6**]	[**0.1**, 0.05]	-	[{128,2}, {256,2}, **{512,32,2}**]
RoBERTa-XLM	[6e-6, **8e-6**]	[**0.05**]	-	[{768,2}, **{1024,64,2}**]
Llama 7B	[**1e-5**, 5e-5, 1e-4, 5e-4]	-	[0.95, **0.99**]	[{4096,128,2}, {4096,512,32,2}, **{8192,1024,64,2}**]

Table 1
Subtask A hyperparameters, best found hyperparameters in bold.

Model	Learning Rate	LR warmup ratio	Gamma	Head layers sizes
BERT-xxl	[3e-6, **4e-6**, 5e-6]	[0.08, **0.04**]	-	[{256,4}, **{512,4}**, {1024,4}]
RoBERTa-XLM	[6e-6, **8e-6**]	[**0.05**]	-	[**{1536,4}**, {1536,64,4}]
Llama 7B	[**1e-4**, 3e-4, 5e-4, 8e-4, 1e-3]	-	[0.99, **0.999**]	[{4096,128,4}, **{4096,768,64,4}**, {8192,1024,128,4}]

Table 2
Subtask B hyperparameters, best found hyperparameters in bold.

For the Topic-specific tf-idf baseline, the validation set was used for finding the best K. After this we used a retrain strategy, in order to obtain a more general *topic_specific_tfidf* for words in each topic (RF classifier was also retrained with same CV hyperparameters)

The performance score of choice is macro-averaged F1 score, as it is the one also used to evaluate the challenge.

6.1. Hyperparameters grid search

Tables 1, 2 and 3 display the explored hyperparameters respectively for transformer-based models in SubtaskA, transformer-based models in SubtaskB and *Topic-specific tf-idf baseline* model. The final chosen hyperparameters are those which yield the best score on the validation set and are highlighted in bold.

6.2. Results

Tables 4 and 5 display the scores on both the internal validation set (the score used to choose the model with the best hyperparameters) and the hidden test set, respectively for SubtaskA and SubtaskB. Only macro-averaged F1 score is reported in the tables.

The whole hidden test set is split in public and private test sets by the competition rules; the final test score is obtained by weighted average (proportional each of the 2 test set sizes) of the public and private sets.

7. Discussions

For both tasks, the best performing models are the BERT-based ones, both the Italian BERT-xxl and XLM-RoBERTa, as their performance is close in F1 terms and are the top-2 performers in both subtasks. These results are a probable cause of the benefits of finetuning or of the encoder-only transformer architecture, versus the decoder and not finetuned Llama.

Among the relevant findings we include also that the transformer dimension does not influence the performance score; for example, although XLM-RoBERTa employs a larger architecture than BERT, they are comparable. The same reasoning applies when confronting with Llama 7B, which has at least an order of magnitude more parameters than the other transformers.

This indicates that the pre-training dataset (we recall that BERT-xxl is not multilingual and trained only in Italian) and the choice of finetuning have the greatest impact on performances.

In regard to the *Topic-specific tf-idf* model, it provides solid results in exchange for a lower computational cost, thanks to its strong assumptions of the importance of topic specific keywords in Subtask B.

It is also important to note that the samples correctly identified by *Topic-specific tf-idf* are not a strict subset of correctly identified samples by the BERT model, as the predictions on the test set have a divergence ratio of almost 25%, while there is a performance difference of less than 7%, meaning that a substantial set of "hard" (wrongly classified) samples for the transformer model are instead "easy" (correctly classified) for the *Topic-specific tf-idf* and vice versa. This implies that combining the 2 models in a meaningful way could result in a more robust model.

References

[1] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi, Evalita 2023: Overview of the 8th evaluation campaign of natural language process-

K	[10, 20, 30, 40, 50, 60, **70**, 80, 90, 100]
Random Forest max_depth	[5, 15, **None**]
Random Forest max_features	[**log2**, None]
Random Forest min_samples_leaf	[[1, **2**, 4]
Random Forest n_estimators	[[64, 128, **256**]

Table 3
Subtask B *Topic-specific tf-idf baseline* hyperparameters, best found hyperparameters in bold.

	Validation score	Test score
BERT-xxl	0.8184	**0.8257**
XLM-RoBERTa	0.7989	0.8203
Llama 7B	0.7954	0.8022

Table 4
Subtask A validation and test scores. Best test model is in bold.

	Validation score	Test score
BERT-xxl	0.8651	0.8265
XLM-RoBERTa	0.8776	**0.8532**
Llama 7B	0.8123	0.7389
Topic-specific tf-idf	0.7400	0.7520

Table 5
Subtask B validation and test scores. Best test model is in bold.

ing 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.

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

[3] H. W. Hanley, D. Kumar, Z. Durumeric, A golden age: Conspiracy theories' relationship with misinformation outlets, news media, and the wider internet (preprint) (2023).

[4] E. Chandrasekharan, U. Pavalanathan, A. Srinivasan, A. Glynn, J. Eisenstein, E. Gilbert, You can't stay here: The efficacy of reddit's 2015 ban examined through hate speech, Proc. ACM Hum.-Comput. Interact. 1 (2017). URL: https://doi.org/10.1145/3134666. doi:10.1145/3134666.

[5] E. Chandrasekharan, S. Jhaver, A. Bruckman, E. Gilbert, Quarantined! examining the effects of a community-wide moderation intervention on reddit, ACM Transactions on Computer-Human Interaction (TOCHI) 29 (2022) 1–26.

[6] A. Trujillo, S. Cresci, Make reddit great again: assessing community effects of moderation interventions on r/the_donald, Proceedings of the ACM on Human-Computer Interaction 6 (2022) 1–28.

[7] 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.

[8] G. Russo, M. H. Ribeiro, G. Casiraghi, L. Verginer, Understanding online migration decisions following the banning of radical communities, arXiv preprint arXiv:2212.04765 (2022).

[9] 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).

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.

[11] T. Sun, A. Gaut, S. Tang, Y. Huang, M. ElSherief, J. Zhao, D. Mirza, E. Belding, K.-W. Chang, W. Y. Wang, Mitigating gender bias in natural language processing: Literature review, arXiv preprint arXiv:1906.08976 (2019).

[12] 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.

[13] G. Russo, C. Gote, L. Brandenberger, S. Schlosser, F. Schweitzer, Disentangling active and passive cosponsorship in the u.s. congress, ArXiv abs/2205.09674 (2022).

[14] J. Valvoda, T. Pimentel, N. Stoehr, R. Cotterell, S. Teufel, What about the precedent: An information-theoretic analysis of common law, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 2275–2288. URL: https://aclanthology.org/2021.naacl-main.181. doi:10.18653/v1/2021.naacl-main.181.

[15] M. Zhong, S. Dhuliawala, N. Stoehr, Extract-

ing victim counts from text, arXiv preprint arXiv:2302.12367 (2023).
[16] P. Alonso, R. Saini, G. Kovács, Hate speech detection using transformer ensembles on the hasoc dataset, in: Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings, Springer, 2020, pp. 13–21.
[17] R. T. Mutanga, N. Naicker, O. O. Olugbara, Hate speech detection in twitter using transformer methods, International Journal of Advanced Computer Science and Applications 11 (2020).
[18] L. Stappen, F. Brunn, B. Schuller, Cross-lingual zero-and few-shot hate speech detection utilising frozen transformer language models and axel, arXiv preprint arXiv:2004.13850 (2020).
[19] 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, Huggingface's transformers: State-of-the-art natural language processing, 2020. arXiv:1910.03771.
[20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.
[21] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification?, 2020. arXiv:1905.05583.
[22] H. Guo, A. Ash, D. Chung, G. Friedland, Detecting conspiracy theories from tweets: Textual and structural approaches., in: MediaEval, 2020.
[23] W. Marcellino, T. C. Helmus, J. Kerrigan, H. Reininger, R. I. Karimov, R. A. Lawrence, Detecting Conspiracy Theories on Social Media: Improving Machine Learning to Detect and Understand Online Conspiracy Theories, RAND Corporation, Santa Monica, CA, 2021. doi:10.7249/RR-A676-1.
[24] DBMDZ, bert-base-italian-xxl-cased, 2020. URL: https://huggingface.co/dbmdz/bert-base-italian-xxl-cased.
[25] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020. arXiv:1911.02116.
[26] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.