A Study on NLP Model Ensembles and Data Augmentation Techniques for Separating Critical Thinking from Conspiracy Theories in English Texts

Notebook for PAN at CLEF 2024

Iñaki del Campo Sánchez-Hermosilla[1], Angel Panizo-Lledot[1] and David Camacho[1]

[1] *Department of Computer System Engineering, Universidad Politécnica de Madrid, Calle de Alan Turing 28031, Madrid, Spain*

Abstract

Conspiracy theories propose that significant events are orchestrated by secretive, powerful groups, gaining traction especially during social upheaval and spreading rapidly via social media. These theories have real-world consequences, as seen in incidents like Pizzagate, where false claims led to a violent attack in a pizzeria in Washington, D.C., and COVID-19 vaccine conspiracies, which fueled public distrust and made the vaccination campaign more difficult. In the age of social media, distinguishing between conspiracy theories and critical thinking is crucial for accurate content moderation because misidentification can push individuals questioning legitimate issues towards conspiracy communities, highlighting the importance of developing effective methods for identifying conspiratorial content. Our study focuses on addressing this challenge by leveraging advanced NLP models. Specifically, we build ensembles using variations of the BERT model, including BERT-base, BERT-large, and RoBERTa. We experimented with different loss functions, such as cross-entropy, Mix-Up, and Supervised Contrastive Loss, and data augmentation techniques like synonym replacement and random word insertions. Our final model achieved a Matthews correlation coefficient (MCC) of 0.8149 on the competition set, securing 8th place in the ranking and demonstrating a considerable level of effectiveness in identifying conspiratorial content.

Keywords

PAN 2024, Oppositional thinking analysis: Conspiracy theories vs critical thinking narratives, NLP, BERT, data augmentation

1. Introduction

Conspiracy theories are intricate narratives that suggest major events are the result of covert actions by secret, powerful, and malicious groups. Conspiracy theories have a long history, often surfacing during times of social upheaval, but their spread has accelerated with the advent of social media. Conspiracy theories have become a social issue; in recent years, we have seen them provoke real-world consequences. For example, incidents like PizzaGate [1], where a man fired a gun in a pizzeria in Washington, D.C. while attempting to investigate a fake child trafficking ring, or the COVID-19 vaccine conspiracy theories [2], which claim that Bill Gates was introducing microchips with the COVID-19 vaccine to spy on people, raising doubts among the population and leading to vaccine hesitancy. Therefore, identifying conspiracy content is more important than ever; however, this is a challenging task. There is a fine line between conspiracy theory and critical thinking, and identifying this distinction is crucial because mislabeling a critical message as conspiratorial could inadvertently push individuals who are merely questioning into conspiracy communities. This highlights the importance of developing effective methods for identifying conspiratorial content.

This edition of PAN 2024 [3] includes a challenge [4] to tackle the aformentioned problem. The challenge include two tasks, one for binary classification of full messages that differentiate between critical thinking and conspiracy theories; and another for token-level classification of text spans that correspons to the key elements of the narratives. In this work we tackle the first task, i.e. the binary classification task where we decide whether a message published in English follows a conspiracy theory framework or, instead, is simply engaging in critical thinking. Our study focuses on addressing this

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

i.delcampo@upm.es (I. del Campo Sánchez-Hermosilla); angel.panizo@upm.es (A. Panizo-Lledot); david.camacho@upm.es (D. Camacho)

challenge by leveraging advanced NLP techniques, specifically using foundational models like BERT [5] and RoBERTa [6]. These models have been trained with millions of data points in a self-supervised manner and are capable of performing well in a wide variety of NLP tasks, especially in tasks where few labeled examples are available. In this article, we focus on creating ensembles of various BERT models. We fine-tune these models using the classical Cross-Entropy loss function and other alternative ones, such as Mix-Up and Supervised Contrastive Loss; and, we employed data augmentation techniques such as sentence rephrasing, translation, and contextual word replacement.

2. Experimental Design

2.1. Data Processing

To ensure a fair comparison, the original dataset was split into two sets, one for testing (10%) and another for training (90%). A stratified split was used due to the unbalanced nature of the dataset.

2.2. Models

We fine-tuned several pre-trained models using 90% of the data reserved for training. All the models tested were variations of the BERT model [5], featuring a transformer-based, encoder-only architecture. As a baseline, the smaller version of the BERT model, "bert-base," was used to conduct a sufficient number of experiments given the limitations of time and computation. Experiments were also conducted with the larger version of BERT, "bert-large," and an optimized version called RoBERTa [6]. Additionally, tests were conducted with a BERT-Large model pre-trained on texts related to the SARS-CoV-2 pandemic [7], which yielded the best results. Finally, we tested whether creating ensembles of these models yielded better results. To create the ensembles, we followed a 5-fold cross-validation approach, where the training dataset was split into 5 folds. An ensemble was created by combining 5 models, each trained with a different combination of 4 out of the 5 folds.

2.3. Loss Functions

As a baseline, the Binary Cross-Entropy (CE) loss function was selected. In addition, we tested two more exotic fitness functions to fine-tune the models: Mix-Up [8] and Supervised Contrastive Loss (SCL) [9]. The latter, SCL, adds a new term to the cross-entropy that penalizes sparse representations of embeddings for examples within the same class. This is achieved by calculating the distance between embeddings of the same class within the batch. Thus, both the batch size and the weighting of each term in the new loss function affect the final loss. Additionally, we decided to try an alternative approach to the hybrid objective proposed in the original paper, consisting of an initial training phase using only the Supervised Contrastive Loss function and a final training phase with binary cross-entropy. The former, Mix-Up, lies midway between data augmentation and a loss function. The idea behind Mix-Up is that a classifier may perform better if, instead of being trained with discrete examples (i.e., 0 or 1), the model is trained with interpolated examples (i.e., X% of a label 0 example and Y% of a label 1 example). Therefore, instead of predicting 0 or 1, the model must predict the percentage of each label in the sample. This technique has shown great utility, particularly in the field of computer vision, as it promotes more linear behavior in the classifier. However, applying this loss directly to the field of NLP is not trivial; thus, we implemented Mix-Up at the embedding level, inspired by the paper "Mixup-Transformer" [10].

2.4. Data Augmentation

Regarding data augmentation, we tested several different approaches. First, we tried rewriting the dataset's sentences using Llama3 8B. In addition, we also used Llama for translating the dataset from the task 2 of this challenge from Spanish to English to increase the trainning data. Finally, we tested more common augmentation strategies using the nlpaug library [11]. Specifically, we decided to apply: word replacement (WR), random word insertion (WI), and synonym replacement (SR). On the one hand, for WR and RI configurations, we used a bert-base model, assigning a percentage of words to insert or replace, while ensuring that the replaced words were not stop-words. On the other hand, for SR we use WordNet [12].

2.5. Validation Framework

We used the 10% of data points reserved for testing to measure the performance of the models, which was measured using the Matthews Correlation Coefficient (MCC). Additionally, due to the large variability in the results between experiments with the same configuration and different seeds, we used 5-fold cross-validation with 10 different random seeds, resulting in a total of 50 single models and 10 5-model ensembles per experiment. Moreover, for the comparison of the ensemble versus the single model, we evaluated each of the 50 models against the test set, as well as the 10 ensembles resulting from each 5-fold, and calculated the median and IQR of the MCCs obtained.

3. Experimental results

Table 1
Common Parameters for Training

Parameter	Value
Learning Rate	2e-5
Scheduler	Triangular 0-0.1-1
Batch Size	16
Epochs	5
Optimizer	Adam

For fine-tuning the models, we followed the proposals by [13]. They recommend a batch size of 16, the Adam optimizer with bias correction, a learning rate of 2e-5, and a triangular scheduler with a linear increase during the first 10% of steps, followed by a linear decay to zero. Although the paper suggests training the models for up to 20 epochs, noting that overtraining does not seem to have a negative impact, we observed no improvement beyond epoch 3. Thus, to expedite testing for the competition, we decided to train all models for only 5 epochs. The hyperparameter selected are available at Table 1.

Table 2
Results on BERT base

	Model	Data Augmentation	Loss	50-fold Test MCC		10 5-Ensemble MCC	
				Median	IQR	Median	IQR
1	*bert-base*	*None*	*CE*	*0.7934*	*0.0222*	*0.8156*	*0.0105*
2	bert-base	Mix-Up γ(0.1, 0.1)	Mix-Up	0.7987	0.0156	0.8101	0.0080
3	bert-base	Mix-Up γ(0.2, 0.2)	Mix-Up	0.7987	0.0281	0.8102	0.0056
4	bert-base	None	SCL lam 0.9 temp 0.3	0.7934	0.0178	0.8045	0.0155
5	bert-base	None	SCL swap 2	0.7849	0.0175	0.7823	0.0156
6	bert-base	Oversampling	CE	0.7939	0.0300	0.8077	0.0193
7	bert-base	Llama_aug	CE	0.7771	0.0242	0.7916	0.0085
8	**bert-base**	**sp_into_en**	**CE**	**0.8185**	**0.0211**	**0.8337**	**0.0103**

For baseline comparison, we have selected a small BERT model (bert-base) with no data augmentation, fine-tuned using the training dataset with a cross-entropy loss function. We will consider a technique worthy if it can improve the performance of this baseline model. Table 2 shows the results of the experiments for the small BERT model (bert-base). The first row shows the results of the baseline model, i.e., small BERT with no data augmentation and cross-entropy loss function. The baseline model achieves a median MCC of 0.7934 when testing the 50 models and a median MCC of 0.8156 when testing the 10 ensembles of 5 folds. As we can see from the results in this table, only experiment number 8 improves the baseline. Each experiment is described in detail below.

Rows 2 and 3 show the results for the Mix-Up training loss. These experiments show a slight improvement over the baseline of +0.005 in the median MCC of the 50 models but a performance drop of -0.005 when ensemble models are used. This occurred in both experiments tested: one using a mixing distribution $\gamma(0.1, 0.1)$ and another using $\gamma(0.2, 0.2)$. These results lead us to discard this technique as it does not present a clear improvement over the baseline and adds considerable complexity and overload to the training process.

Similarly, rows 4-5 show the results for the Supervised Contrastive Learning (SCL) loss. Row 4 shows the results for the original version proposed by the authors, using their recommended configuration of a 0.3 temperature and a weighting in the objective function of 0.9 to the distance between embeddings and 0.1 to cross-entropy. Row 5 shows our alternative, where the embedding distance was used for 2 batches and cross-entropy for the remaining 3. The results were poor. On the one hand, although the original approach showed a very slight improvement in the results of the 50 models, with an increase of 0.001, it produced a drop of -0.01 in MCC when evaluating the ensembles. On the other hand, the new version proposed by us performed significantly worse than the baseline, with a -0.008 decrease in the median MCC evaluation of the 50 models and a drop of -0.03 when evaluating the ensembles. Considering these results, the effectiveness of this method cannot be assured for this problem. Nevertheless, to fully test the method, an exhaustive search for optimal parameters would be necessary. However, given the nature of the challenge, we decided to explore other more promising avenues.

The sixth row involved training with a balanced training dataset that ensures 50% positive and 50% negative examples. This balanced dataset was created by oversampling the minority class. As we can see, the individual models performed similarly to the baseline. However, there was a -0.004 decrease in performance in the ensembles. Since no significant improvement was observed, this modification was discarded for future iterations.

Row seven shows the results of augmenting the training dataset by asking Llama-3 8B to rewrite the original sentences (llama_aug). The results showed a clear detriment to the performance of both individual models and ensembles, leading to the decision to abandon this line of experimentation for future iterations.

Finally, the last row shows the result of extending the dataset with additional data obtained by translating the Spanish dataset from Task 1 of this competition into English (sp_into_en). As we can see, this approach is undoubtedly the only successful addition to the model so far, providing an average improvement of +0.025 in the evaluation of individual models and +0.018 in the ensembles.

Table 3
Results over bigger Models

	Model	Data Augmentation	Loss	50-fold Test MCC		10 5-Ensemble MCC	
				Median	IQR	Median	IQR
1	*bert-large*	*None*	*CE*	*0.8161*	*0.0184*	*0.8362*	*0.0148*
2	bert-large	sp_into_en	CE	0.8282	0.0257	0.8509	0.0140
3	Roberta-large	None	CE	0.8268	0.0269	0.8385	0.0058
4	bert-large-Covid	None	CE	0.8504	0.0135	0.8730	0.0057
5	bert-large-Covid	sp_into_en	CE	0.8193	0.0222	0.8288	0.0219
6	bert-large-Covid	SR 0.1	CE	0.8557	0.0254	0.8730	0.0164
7	**bert-large-Covid**	**SR 0.5**	**CE**	**0.8504**	**0.0153**	**0.8727**	**0.0134**
8	bert-large-Covid	WR 0.1	CE	0.8399	0.0225	0.8615	0.0070
9	bert-large-Covid	SR 0.2, Ri 0.1	CE	0.8499	0.0178	0.8672	0.0105

Once the different configurations were tested on the BERT-base model, we developed a series of experiments to test these configurations on larger models. The results are available in Table 3. The first row shows the baseline for this round of experiments. When compared with the results in Table 2, we can see that increasing the BERT model size provides a significant performance boost, with an improvement in MCC of 0.023 in individual models and 0.026 in the ensembles. Given the success of the large model, we tried adding the sp_into_en data augmentation, which yielded good results on BERT-base. Row 2 shows these results; as we can see, this combination achieves a solid improvement, raising the median by 0.012 and 0.0147 in individual models and ensembles, respectively.

Next, rows 3 and 4 show the results of the baseline configuration with two new models, RoBERTa-large and a BERT-large model pre-trained with texts related to the COVID-19 pandemic (bert-large-covid). The former shows a mild improvement over the BERT-large benchmark, with an increase of 0.01 in the median of the individual models; however, it only shows an improvement of 0.002 in the ensembles. Meanwhile, the latter model, bert-large-covid, presents a significant improvement over all previously tested models, with an improvement of 0.034 in the single model and 0.036 in the ensembles. Given these good results, the rest of the experiments will focus on the bert-large-covid model.

Row 5 shows the results of incorporating the sp_into_en data augmentation onto the bert-large-covid model. However, to our surprise, this caused a significant performance drop in the model, leading to a loss of -0.03 and -0.04 in the individual models and ensembles, respectively.

Finally, rows 6-9 show experiments with simple data augmentation techniques such as synonym replacement (SR), word replacement (WR), and insertions with BERT-base (WI). As we can see, synonym replacement was the technique that yielded the best results, providing a slight improvement, while word replacement and random insertion negatively impacted the models.

Based on these results, the final model used for the submission of task 1 in its English version was an ensemble averaging the predictions of all the trained bert-large-covid SR 0.5 models. This model obtained an MCC of 0.8149, F1-MACRO of 0.9072, F1-CONSPIRACY of 0.8770, and F1-CRITICAL of 0.9374, resulting in 8th place in the ranking.

4. Conclusions and future work

In this work, we tackle the challenge of distinguishing between conspiracy theories and critical thinking using advanced NLP models. Specifically, we build ensembles using variations of the BERT model, including BERT-base, BERT-large, and RoBERTa. We experimented with different loss functions, such as cross-entropy, Mix-Up, and Supervised Contrastive Loss, and used data augmentation techniques like synonym replacement and random word insertions. From our experimentation, we can conclude that increasing the BERT model size significantly boosts performance, with BERT-large-covid showing the best results for future experiments. Additionally, our experimentation shows that classic cross-entropy loss achieves better results than more complex techniques like Mix-Up and Supervised Contrastive Loss. Finally, we conclude that applying simpler data augmentation techniques like word replacement or word insertion works better than more sophisticated techniques involving state-of-the-art LLMs. Nevertheless, more experimentation is needed with the prompts used for the LLMs, such as including some examples in them. Additionally, it would be interesting to try more models, for example, pre-training RoBERTa on a large COVID corpus and then applying fine-tuning for classification.

Acknowledgements

This work has been supported by MICINN under FightDIS (PID2020-117263GB-I00); by MCIN/AEI/10.13039/501100011033/ and European Union NextGenerationEU/PRTR for XAI-Disinfodemics (PLEC2021-007681) grant; and by the project PCI2022-134990-2 (MARTINI) of the CHISTERA IV Cofund 2021 program, funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR".

References

[1] M. Fisher, J. W. Cox, P. Hermann, Pizzagate: From rumor, to hashtag, to gunfire in dc, Washington Post 6 (2016) 8410–8415.

[2] S. K. Lee, J. Sun, S. Jang, S. Connelly, Misinformation of covid-19 vaccines and vaccine hesitancy, Scientific Reports 12 (2022) 13681.

[3] J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, A. Mukherjee, et al., Overview of pan 2024: multi-author writing style analysis, multilingual text detoxification, oppositional thinking analysis, and generative ai authorship verification, in: European Conference on Information Retrieval, Springer, 2024, pp. 3–10.

[4] K. Damir, C. Berta, B. C. Xavier, T. Mariona, R. Paolo, R. Francisco, Overview of the oppositional thinking analysis PAN task at CLEF 2024, 2024.

[5] 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), 2019, pp. 4171–4186.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[7] M. Müller, M. Salathé, P. E. Kummervold, Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter, Frontiers in artificial intelligence 6 (2023) 1023281.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[9] B. Gunel, J. Du, A. Conneau, V. Stoyanov, Supervised contrastive learning for pre-trained language model fine-tuning, in: International Conference on Learning Representations, 2021.

[10] L. Sun, C. Xia, W. Yin, T. Liang, S. Y. Philip, L. He, Mixup-transformer: Dynamic data augmentation for nlp tasks, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 3436–3440.

[11] E. Ma, Nlp augmentation, https://github.com/makcedward/nlpaug, 2019.

[12] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (1995) 39–41.

[13] M. Mosbach, M. Andriushchenko, D. Klakow, On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines, in: 9th International Conference on Learning Representations, 2021.