# Leveraging Large Language Models with Multiple Loss Learners for Few-Shot Author Profiling

Notebook for PAN at CLEF 2023

Hamed Babaei Giglou[1,2], Mostafa Rahgouy[3], Jennifer D'Souza[1], Milad Molazadeh Oskuee[4], Hadi Bayrami Asl Tekanlou[5] and Cheryl D Seals[3]

[1]*TIB Leibniz Information Centre for Science and Technology, Hannover, Germany*
[2]*Leibniz Universität Hannover, Hannover, Germany*
[3]*Department of Computer Science, Auburn University, Alabama, USA*
[4]*Department of Computer Science, Iran University of Science and Technology, Tehran, Iran*
[5]*Department of Computer Science, University of Mohaghegh Ardabili, Ardabil, Iran*

## Abstract

The objective of author profiling (AP) is to study the characteristics of authors through the analysis of how language is exchanged among people. Studying these attributes sometimes is challenging due to the lack of annotated data. This indicates the significance of studying AP from a low-resource perspective. This year at AP@PAN 2023 the major interest raised in profiling cryptocurrency influencers with a few-shot learning technique to analyze the effectiveness of advanced approaches in dealing with new tasks from a low-resource perspective. The AP-2023 task consists of 3 subtasks including cryptocurrency influencer analysis, interest identification, and intent identification. In this work, we studied the integration of Bi-Encoders with Large Language Models (LLMs), to enhance the semantic representation of authors by enabling the models to transfer knowledge across domains and adapt to new tasks with a small number of data. We incorporated multi-losses to enforce LLMs to learn the representations of different categories and authors to facilitate similarity-based comparisons among authors and categories. Finally, our approach achieved impressive F1 Macro scores of 52.31 for crypto influencer profiling, 61.21 for crypto influencer interest identification, and 65.83 for crypto influencer intent identification using limited supervised learning data. Overall, the obtained and experimental analysis shows the effectiveness of the integration of multiple-loss learners with LLMs in profiling cryptocurrency influencers using limited resources.

## Keywords

Few-shot Learning, Large Language Models, Low-resource Text Classification, Cryptocurrency, Author Profiling

*CLEF 2023: Conference and Labs of the Evaluation Forum, September 18-21, 2023, Thessaloniki, Greece*
hamed.babaei@tib.eu (H. Babaei Giglou); mzr0108@auburn.edu (M. Rahgouy); jennifer.dsouza@tib.eu (J. D'Souza); molazadeh\protect1_milad@comp.iust.ac.ir (M. Molazadeh Oskuee); hadibayrami@student.uma.ac.ir (H. Bayrami Asl Tekanlou); sealscd@auburn.edu (C. D. Seals)
https://hamedbabaei.github.io/ (H. Babaei Giglou); https://sites.google.com/view/jen-web (J. D'Souza); https://www.eng.auburn.edu/~sealscd/ (C. D. Seals)
0000-0003-3758-1454 (H. Babaei Giglou); 0000-0002-6616-9509 (J. D'Souza)

# 1. Introduction

Cryptocurrencies have gained high popularity in recent years, capturing the attention of many. Factors such as independency from central authorities, the potential offered by different cryptocurrency projects, and the influence of social media influencers have contributed to their trendy status. However, in a real environment where, for instance, traders may want to leverage social media signals to forecast the market, data collection is a challenge and real-time profiling needs to be done in a few milliseconds, which implies processing as little data as possible. However, due to the economic and temporal cost, the psychological and linguistic expertise needed by the annotator, and the congenital subjectivity involved in the data preparation for analysis of social media for cryptocurrency topics, studying this field from a low-resource perspective become an important matter.

In recent years, there has been significant interest in exploring the capabilities of Large Language Models (LLMs) to perform new tasks through inference alone. This emerging approach, known as *in-context learning* or *prompting* technique, relies on utilizing zero-shot or a limited number of input-label pairs, referred to as demonstrations, to enable the model to predict new inputs without explicit training on the specific task. To improve the in-context learning technique, we have three ways; 1) Training an LLM on multiple tasks, thereby offering the LLMs a training exposure to diverse tasks, with the hypothesis that it facilitates its adaptation to novel tasks during testing, 2) Choosing labeled examples for the demonstrations more effectively, and 3) Exploring variants of in-context learning, such as learning to follow instructions or incorporating external knowledge sources [1]. Moreover, LLMs demonstrate their capability as few-shot learners, achieving impressive performance on novel tasks with minimal training examples. By scaling up LLMs, their effectiveness in a task-agnostic, few-shot environment can be significantly enhanced, often surpassing the performance of previous state-of-the-art fine-tuning methods [2]. Task profiling cryptocurrencies influencers can benefit from this since analyzing users' behavior on social media often is challenging due to the low resource perspective of the task.

In this paper, we addressed a PAN-2023 shared task [3] on *Profiling Cryptocurrency Influencers* [4] to profile cryptocurrency influencers in social media, from a low-resource perspective in three subtasks:

- **Subtask 1**: Low-resource influencer profiling
- **Subtask 2**: Low-resource influencer interest identification
- **Subtask 3**: Low-resource influencer intent identification

To identify and analyze cryptocurrency influencers on Twitter, we utilized LLMs and a semantic textual similarity approach with multiple loss learners to leverage LLMs for learning different profiling tasks in a few-shot scenario in the English language. The code for the proposed method has been published in a repository on GitHub[1] for the research community.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 describes the proposed methodology in detail. Section 4 describes the experimental setup including dataset, metrics, and training settings. Next, in section 5 we discuss the results. Finally, section 6 presents our conclusions.

[1]https://github.com/HamedBabaei/author-profiling-pan2023

## 2. Related Work

Transfer Learning (TL) has been successfully applied to many machine learning applications, including text sentiment classification, image classification, human activity classification, software defect classification, and multi-language text classification. TL transfers knowledge from the source domain/task, where training data is abundant, to the target domain/task, where training data is scarce [5]. TL methods are popularly used in Few-Shot Learning (FSL), where the prior knowledge is transferred from the source task to the few-shot task. Where FSL is a type of machine learning problem, that contains only a limited number of examples with supervised information [6]. Bidirectional Encoder Representations from Transformers (BERT) [7] is a powerful transformer-based model pretrained on a large corpus of unlabeled text data. It has been successfully applied in many FSL tasks, where the pretrained BERT model is fine-tuned on a few-shot dataset for tasks like text classification, named entity recognition, and question answering. By leveraging the pretraining knowledge, BERT can effectively generalize and adapt to new few-shot tasks. Similarly, T5: Text-To-Text Transfer Transformer [8] is trained on a large-scale text-to-text dataset, where it learns to map input text to output text. T5 can be fine-tuned on a few-shot task by formulating the task as a text-to-text problem. It has been applied to various FSL tasks, including text classification, summarization, and machine translation. Moreover, Flan-T5 [9] combines the benefits of FSL and T5 to enable effective knowledge transfer and adaptation to new few-shot tasks. It highlights the significant performance boost achieved by incorporating chain-of-thought [10] prompting data for reasoning tasks and providing a comprehensive evaluation of instruction-finetuned models across various setups and benchmarks.

The FASL [11] is a platform that integrates FSL and active learning to facilitate rapid and effective training of text classification models. The authors examine various active learning methods to determine their effectiveness in a few-shot setup and develop a model that predicts the optimal point to stop annotating data. In [12] explores the construction of text classifiers with minimal or zero training data by employing Siamese Networks, which embed both texts and labels to enable model adaptation in few-shot scenarios by solely modifying the label embeddings. The SimSCE [13] is a simple contrastive learning (CL) framework that uses the standard dropout operation to generate high-quality training pairs. It's scalable and better in the regular data augmentation used in NLP tasks. CL [14] is a kind of unsupervised learning that learns how to represent data by comparing pairs that are alike and different. The model is trained to tell apart positive pairs and negative pairs in the input data. The aim is to learn a representation space where similar samples are close to each other, while different samples are far from each other. CL has been shown to be effective for few-shot learning. It has also been used in combination with other techniques such as self-supervised learning and transfer learning to improve performance on downstream tasks with limited labeled data.

## 3. Methodology

In this section, we describe the details of our proposed model. Our proposed approach aims to predict whether the user is keen to be a cryptocurrency influencer's profile, interest, and intent on Twitter. With recent advancements in LLMs, we studied leveraging LLMs for cryptocurrency influencer profiling, we utilized the Flan-T5 model and employed it in the form of text generation, incorporating multiple instructions per user tweet. The primary objective was to train the Flan-T5-Encoder using the whole model architecture. Next, we constructed a bi-encoder model using the finetuned encoder component of Flan-T5. We employed contrastive learning (CL) and Multiple Negative Ranking (MNR) [15] losses during the training. For CL, we created negative samples which allowed us to create a comparative framework for the model to learn from, however, the MNR uses positive pairs and automatically generates negative pairs in low dimensional space. Both losses enhance the ability of Flan-T5 to distinguish between different types of author profiles. In the testing phase, we employed cosine similarity to compare different users and categories them. Figure 1 illustrates our framework.

First, we concatenate all user tweets, next, we conducted some minor preprocessing, i.e.; Removing URLs, @ and # symbols, punctuations, special characters, additional lines, free spaces, and converting text to lowercase. The processed user texts are used for training and testing models. In the following, we will describe the training and testing components of the proposed method separately.

## 3.1. Prompt Templates

Prompt templates [16] are predefined structures or guidelines that assist in generating effective prompts for LMs. They serve as a framework for constructing inputs in a standardized format that the model can understand and respond to appropriately. Prompt templates typically include placeholders or keywords that users can customize with their desired information. By using prompt templates, we can easily obtain accurate and relevant responses from the LM. The standardized format provided by prompt templates helps ensure clarity and consistency in interactions, enabling efficient and effective communication with the model. For these reasons, we manually designed **10** prompt templates per subtask to convert original user tweets to form questions with contexts (tweets) as inputs of an LM. The prompt templates are illustrated as $T_i(U)$, where $i$ is the $i$-th template and $U$ is the combined user tweets. For example, $T_2(U)$ is the second template for querying LMs for user interest and it is defined as follows:

> Analyze the given tweets to identify if the user has a particular purpose in cryptocurrency.
>
> Tweets: {tweets}

Where {tweets} is a placeholder for a user's tweets. All the templates per task are listed in table 3 in the appendix.

## 3.2. Finetuning Flan-T5-Encoder

In this study, we conducted fine-tuning of the Flan-T5 LLM for a multi-class classification task using designed templates (section *A. Flan-T5* in figure 1). Flan-T5, a state-of-the-art language model based on the Transformer architecture, was initially pre-trained on a vast amount of text data to capture the intricacies of language semantics. By fine-tuning Flan-T5 on our specific

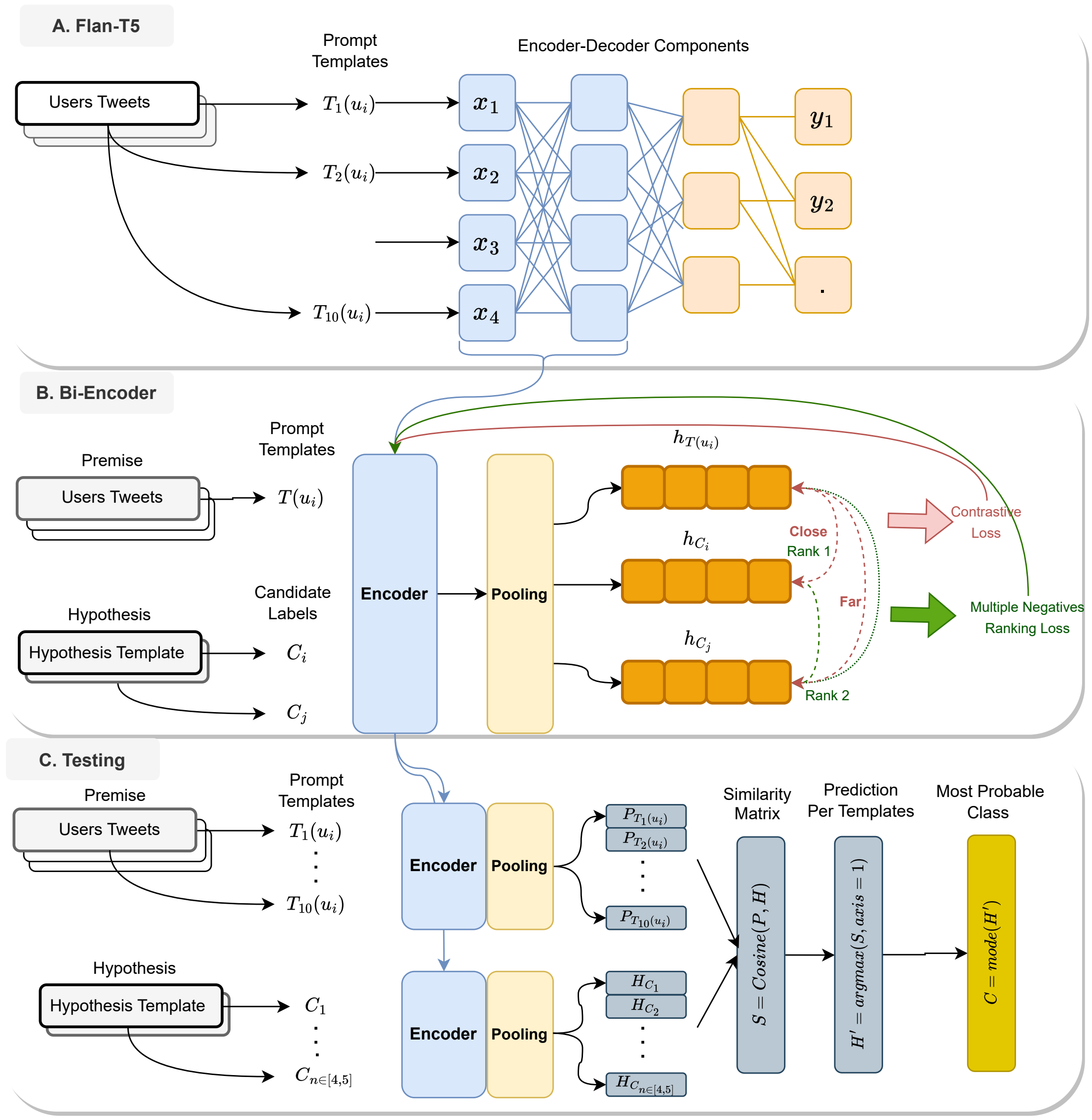


**Figure 1:** An illustration of Flan-T5 and Bi-Encoder finetuning, and testing workflow on profiling influencers.

task, we aimed to enhance its ability to accurately classify text inputs into multiple predefined classes where later only the encoder component will be used. To accomplish this, we followed a systematic approach that involved filling $T_i(U_j)$ (where $j$ is user $j$-th in the training dataset) prompt templates for all the users and feeding the Flan-T5 as input with the respective class as output. By conditioning the Flan-T5 model on these templates, we generated synthetic training examples for each class, augmenting the original labeled data. This approach not only helped address the issue of the low-resource nature of the problem but also enhanced the model's generalization capability by exposing it to diverse and informative examples, which resulted in generalized encoder architecture for the bi-encoder component.

### 3.3. Bi-Encoder Model with Multiple Losses

LLMs hallucination is one of the weaknesses of LLMs which sometimes makes them hard for classification tasks due to the limited available classes automatic classification with LLMs without interfering is tricky, especially when we are performing an FSL, but nevertheless, they contain a large amount of information about different topics which make them very good few-shot learners [17].

The Bi-Encoder model is a type of encoder architecture where commonly used in tasks such as semantic similarity. It is designed to encode pairs of text inputs and measure their semantic similarity. In our case, considering the $T(u_i)$ as a set of filled prompt templates per user as a premise, we created the $C_i$ hypothesis where $C_i$ represents the hypotheses template for the user $i$-th (hypotheses templates per subtask listed in the table 4 at appendix). So per a user, we can create a $\cup_{t=1}^{10}(T_t(u_i), C_i)$ *premise-hypothesis* set as positive samples for training a Bi-Encoder model to correlate users with similar classes into a high-dimensional space, where it will learn premise representations in semantic space that have a similar hypothesis. To further optimize the performance of the Flan-T5, we have explored Bi-Encoder models with the use of multiple losses during training. Incorporating multiple loss functions allows the model to capture different aspects of the data and learn more robust and accurate representations. We applied the following loss during training a Flan-T5-Encoder with a Bi-Encoder strategy.

- **Contrastive Learning Loss (CL):** It encourages similar premise-hypothesis pairs to have higher similarity scores, while dissimilar pairs (premise with incorrect hypothesis) have lower scores. This loss effectively trains the model to distinguish between relevant and irrelevant pairs. This results in identifying the premise (users) with the relevant hypothesis (label) as a similar premise-hypothesis pair and the incorrect hypothesis as a dissimilar pair.
- **Multiple Negative Ranking Loss (MNR):** It is crucial to explicitly optimize the model for ranking relevant user-label (premise-hypothesis) pairs with higher than irrelevant ones. Ranking loss is used to directly optimize the ranking order of user-label (premise-hypothesis) pairs. By incorporating a ranking loss, the Bi-Encoder model can better capture the relative importance of different users for a given label candidate.

The MNR uses positive premise-hypothesis (user-label) pairs for the training and it automatically will rank the respective hypothesis at a higher rank than the other unrelated hypotheses. However, CL except for the positive premise-hypothesis pairs requires negative pairs as well which means a user with an incorrect label (premise with incorrect hypothesis). For this, we created automatically negative samples by corrupting the hypothesis with premises during the training process.

### 3.4. Testing

The BiEncoder model consists of two encoders, referred to as the "premise encoder" and the "hypothesis encoder". These encoders independently transform the premise and hypothesis

**Table 1**
Statistics of the experimental dataset. We split the train set into train-train and train-test datasets for profiling cryptocurrency influencers with few-shot learning.

| Task | No. of Classes | Train-Train | Train-Test | Total-Train | Final-Test |
|---|---|---|---|---|---|
| *Subtask 1* | 5 | 80 | 80 | 160 | 220 |
| *Subtask 2* | 5 | 160 | 160 | 320 | 402 |
| *Subtask 3* | 4 | 128 | 128 | 256 | 292 |

texts into fixed-dimensional representations. As a result of these, we will obtain $P^{10*1024}$ and $H^{n*1024}$ matrixes as a fixed-dimensional representation of premises and hypothesis, respectively. Where $n$ is the number of label candidates and for subtask 1 and subtask 2 it is set to $n = 5$ and for subtask 3 it is set to $n = 4$. Next, we calculated the cosine-similarity matrix between premises $V$ and hypothesis $U$ matrixes as following:

$$S = \sum_{t=1}^{10} \sum_{c=1}^{n} cosine(P_t, H_c)$$

Where $S^{10*n}$ is the obtained similarity matrix, next for premise-based hypothesis identification we chose the maximum probable hypothesis per all 10 premises as follows:

$$H' = argmax(S, axis = 1)$$

Where $H'$ is a predicted hypothesis for all 10 premises and the most appeared hypothesis in $H'$ will be chosen as a final hypothesis prediction as follows:

$$C = mode(H')$$

Where $C$ is the final candidate hypothesis for the input premise.

## 4. Experimental Setup

**Dataset:** Table 1 presents the statistics of the dataset [18], which consists of 380 users in total for subtask 1 with a maximum of 10 tweets per user in 5 categories, 722 users for subtask 2 with a single tweet per in 5 categories, and 548 users for subtask 1 with a single tweet per user in 4 categories. The dataset is balanced in the training where for subtask 1 per each class only 32 users are provided and for subtask 1 and 2 only 64 users per class. We split the train set into the 50/50 proportion for experimental analysis where the stats are presented in the table 1.

**Metrics:** According to the multi-class nature of the subtasks we have used Macro F1 as an evaluation metric.

**Training Setups:** We considered the Flan-T5-Large[2] variant for training models. We utilized a consistent training strategy for all subtasks where Flan-T5 trained using *AdamW* optimizer

[2]https://huggingface.co/google/flan-t5-large

**Table 2**
Reported F1-Macro Results for Subtask 1, Subtask 2, and Subtask 3 in Baseline and Experimental Models with Train-Test Set, and Final Results with Test Set for team **symbol** in the AP@2023.

| Model | Subtask 1 | Subtask 2 | Subtask 3 | Average |
|---|---|---|---|---|
| Experimental results on Train-Test Set | | | | |
| *RandomBaseline* | 23.07 | 17.52 | 31.17 | 23.92 |
| *Flan-T5+Zero-Shot* | 16.61 | 6.66 | 26.07 | 16.45 |
| *Flan-T5+Few-Shot* | 36.47 | **63.11** | 72.68 | 57.42 |
| *Flan-T5+Bi-Encoder+CL+MNR* | **43.36** | 60.35 | 64.75 | 56.15 |
| *Flan-T5+Few-Shot+Bi-Encoder+CL+MNR* | **43.06** | 60.76 | **73.34** | **59.05** |
| The **symbol** team submission results on Test Set | | | | |
| *Flan-T5+Few-Shot+Bi-Encoder+CL+MNR* | **52.31** | **61.21** | **65.83** | **59.78** |

with a learning rate of $1e-5$ For $10$ epoch and batch size of $4$. Also, the Bi-Encoder model is trained to minimize the combined loss, which is a weighted sum of the individual losses. The model's parameters are updated using $AdamW$ optimizer with and learning rate of $1e-5$ and batch size of $4$. The CL requires a margin to put negative samples at least a margin further apart from the anchor than the positive, where we set this metric into $0.5$ during the training with cosine distant metric.

## 5. Results

We experimented using a *train-test* set from the train set as an evaluation dataset to analyze the proposed model. Table 2 shows the experimentally designed models and their results per subtask. In the following, we discussed the experimental models and our analysis for the task.

### 5.1. Experimental Models

In our experimental analysis of the models, we evaluated five different models based on their F1 Macro scores. To compare the proposed methods, we implemented two baseline models, *RandomBaseline* and *Flan-T5 + Zero-Shot*, provided initial performance benchmarks. Subsequently, we designed and tested three additional models:

1. *RandomBaseline*: A random prediction model that predicts based on the random selection of a class from candidate label sets.
2. *Flan-T5 + Zero-Shot*: The Flan-T5 has been trained on NLI task so to use Flan-T5 for inferencing in a zero-shot manner, we used *"xnli: premise: {premise} hypothesis: {hypothesis}"* prompt template to query Flan-T5. Where {premise}, and {hypothesis} are placeholder for premise and hypothesis.
3. *Flan-T5 + Few-Shot*: A text-to-text with a few-shot scenario and instruction tuning for text classification.
4. *Flan-T5 + Bi-Encoder+CL+MNR*: A Bi-Encoder model that uses CL and MNR losses with pretrained Flan-T5 encoder component.

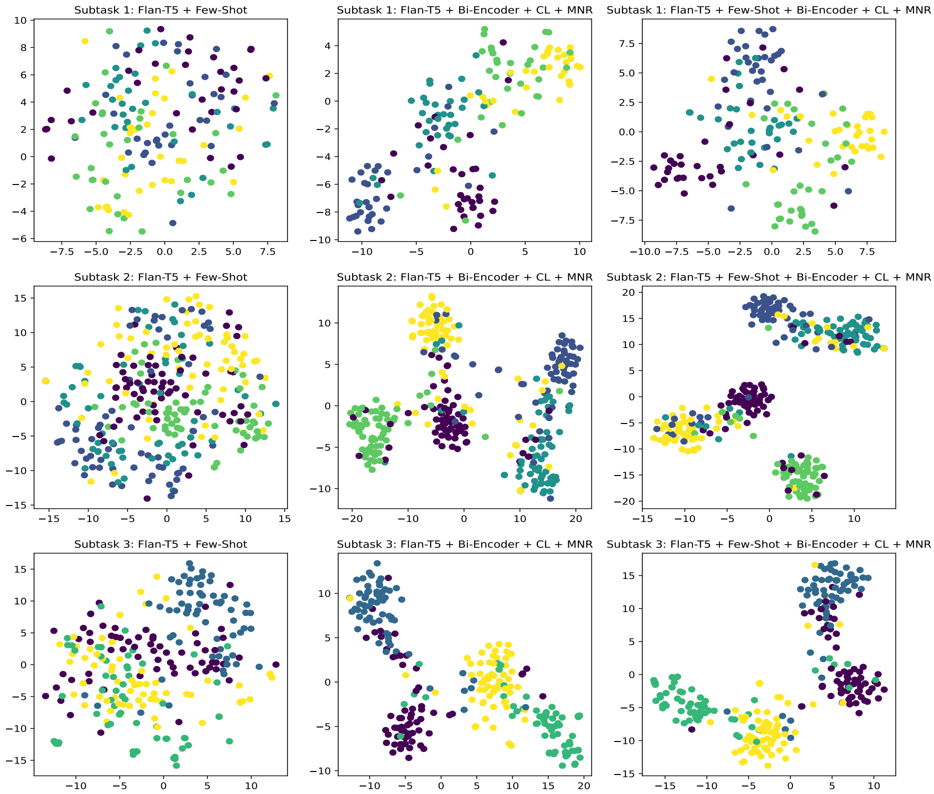


**Figure 2:** The t-SNE visualization of the author embeddings learned from the different models for all three subtasks. The models trained on *Train-Train* set and visualized in all the train data's.

5. *Flan-T5+Few-Shot+Bi-Encoder+CL+MNR*: A Bi-Encoder model that uses CL and MNR losses with finetuned Flan-T5-Encoder model from *Flan-T5+Few-Shot* model.

## 5.2. Quantitative Findings

**Baseline models:** The *RandomBaseline* model exhibited limited performance across all three subtasks, with an average F1 Macro score of 23.92. The *Flan-T5+Zero-Shot* model also demonstrated below average level performance, with an average F1 Macro score of 16.45 on all three tasks. These results indicate that both baseline models struggled to effectively classify the different classes in the given tasks.

**Few-Shot learning:** The *Flan-T5+Few-Shot* model showed substantial improvements in performance regarding the baseline model, with an average F1 Macro score of 57.42. This model successfully leveraged FSL techniques to adapt to the classification tasks, resulting in improved all subtasks by a large margin.

**Bi-Encoder with multiple loss learners:** The *Flan-T5 + Bi-Encoder + CL + MNR* model demonstrated competitive performance, with an average F1 Macro score of 56.15. The inclusion of bi-encoders and multiple-loss learners contributed to enhanced classification accuracy.

**Few-Shot Flan-T5 with Bi-Encoder and multiple loss learners:** The *Flan-T5 + Few-Shot +Bi-Encoder+CL+MNR* model achieved the highest overall performance among all the models tested, with an average F1 Macro score of 59.05. This model combined few-shot learning techniques on Flan-T5 with bi-encoders, contrastive learning, and multiple negative ranging losses resulting in the most accurate classification across the evaluated subtasks.

**Best performers:** The Flan-T5 model with few-shot scenario performed well on subtask 2 by Macro F1 score of **63.11** by 2.35 % better than *Flan-T5 + Few-Shot +Bi-Encoder+CL+MNR*, however, *Flan-T5 + Few-Shot +Bi-Encoder+CL+MNR* performed very well on both subtask 1 and subtask 2.

Our experimental analysis highlights the effectiveness of incorporating advanced techniques such as few-shot learning, bi-encoders, and multiple-loss learners. The *Flan-T5+Few-Shot+Bi-Encoder+CL+MNR* model emerged as the top-performing model, followed closely by the *Flan-T5+Few-Shot* and *Flan-T5+Bi-Encoder+CL+MNR* models. So for the final submission, we submitted *Flan-T5+Few-Shot+Bi-Encoder+CL+MNR* and *Flan-T5+Few-Shot* models where according to the best performer results among them we obtained the best F1 Macro of 43.06 for subtask 1, 63.11 for subtask 2, 73.34 for subtask 3, and average F1 Macro of **59.84** for final results in experimental setups.

## 5.3. Quantitative Analysis

**Bottleneck in terms of performance:** All the experiments reveal that models are suffering from better performance in subtask 1 due to the high input text (10 tweets per user) that overlaps semantically with other classes. It shows that the author's style analysis [19] in finding author profiles is still important in getting good performance in subtask 1.

**Benefits of multiple losses:** By combining multiple losses, the Bi-Encoder model can leverage the strengths of each loss function and achieve improved performance. The CL loss encourages clear separation between relevant and irrelevant pairs, and the MNR ranking loss emphasizes the correct ordering of hypothesis (or label candidates). This combination allows the model to learn more comprehensive representations and enhance its ability to accurately measure semantic similarity and determine the correct hypothesis for premises.

**Visualization:** Given author embedding learned by different models, the t-SNE visualization in figure 2 shows that *Flan-T5+Few-Shot+Bi-Encoder+CL+MNR* learn more discriminative author embeddings that are more distinguishable. Generally, the visualizations show that Bi-Encoder models produce linearly separable features that make them good models than *Flan-T5-Few-shot*, where subtask 1, 2, and 3 clearly shows that similar users scattered all over the semantic space which makes *Flan-T5-Few-shot* not well generalized for the task.

### 5.4. Final Results

In a final submission, our team **symbol** obtained averaged F1 Macro scores of 52.31 for subtask 1, 61.21 for subtask 2, and 65.83 for subtask 3 over an unseen test set. For task 1, the symbol achieved 11th out of 27 participants, for task 2, the symbol acquired 6th place among 20 submissions, and for task 3, the symbol gained 3rd place among 21 teams. In all tasks, our proposed model defeats all the respective baseline models, it again confirms the quality proposed model in *profiling cryptocurrency influencers with few-shot learning* task. Overall, we obtained **2nd** place according to the official ranking [3].

## 6. Conclusion

In conclusion, our experimental analysis focused on three multi-class classification subtasks within the context of cryptocurrency influencer profiling. The results demonstrated the effectiveness of integrating advanced techniques, including few-shot learning, bi-encoders, and multiple-loss learners, to leverage LLMs. Our approach yielded promising outcomes in development by achieving F1 Macro scores of 43.36 for subtask 1, 63.11 for subtask 2, and 73.34 for subtask 3. As a result, our team *symbol* achieved F1 Macro scores of 52.31 for subtask 1, 61.21 for subtask 2, and 65.83 for subtask 3. The *symbol* team achieves 2nd rank among competitors where the results highlight the capability of our approach in accurately determining influencers' profiles, intentions, and interests, as confirmed through manual evaluation. While our findings provide compelling evidence of the effectiveness of the applied techniques, there is still room for further exploration and improvement. Continued research and development can uncover additional avenues to enhance the performance and capabilities of our approach, opening up new possibilities in the field of author profiling.

## References


[1] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, L. Zettlemoyer, Rethinking the role of demonstrations: What makes in-context learning work?, arXiv preprint arXiv:2202.12837 (2022).

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners – special version, Advances in Neural Information Processing Systems 2020-Decem (2020). arXiv:2005.14165.

[3] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pęzik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, A. G. Stefanos Vrochidis, D. Li,


[3] https://pan.webis.de/clef23/pan23-web/author-profiling.html

M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science, Springer, 2023.

[4] M. Chinea-Rios, I. Borrego-Obrador, M. Franco-Salvador, F. Rangel, P. Rosso, Profiling Cryptocurrency Influencers with Few shot Learning at PAN 2023, in: CLEF 2022 Labs and Workshops, Notebook Papers, 2023.

[5] K. Weiss, T. M. Khoshgoftaar, D. D. Wang, A survey of transfer learning, volume 3, Springer International Publishing, 2016. doi:10.1186/s40537-016-0043-6.

[6] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a Few Examples: A Survey on Few-shot Learning, ACM Computing Surveys 53 (2020) 1–34. doi:10.1145/3386252. arXiv:1904.05046.

[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] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[9] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling Instruction-Finetuned Language Models (2022) 1–54. URL: http://arxiv.org/abs/2210.11416. arXiv:2210.11416.

[10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, D. Zhou, Chain of thought prompting elicits reasoning in large language models, arXiv preprint arXiv:2201.11903 (2022).

[11] T. Müller, G. Pérez-Torró, M. Franco-Salvador, Few-Shot Learning with Siamese Networks and Label Tuning, Proceedings of the Annual Meeting of the Association for Computational Linguistics 1 (2022) 8532–8545. doi:10.18653/v1/2022.acl-long.584. arXiv:2203.14655.

[12] T. Müller, G. Pérez-Torró, A. Basile, M. Franco-Salvador, Active Few-Shot Learning with FASL, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 13286 LNCS (2022) 98–110. doi:10.1007/978-3-031-08473-7_9. arXiv:2204.09347.

[13] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 6894–6910.

[14] B. Gunel, J. Du, A. Conneau, V. Stoyanov, Supervised contrastive learning for pre-trained language model fine-tuning, 2021. arXiv:2011.01403.

[15] M. Henderson, R. Al-Rfou, B. Strope, Y. hsuan Sung, L. Lukacs, R. Guo, S. Kumar, B. Mik-

los, R. Kurzweil, Efficient natural language response suggestion for smart reply, 2017. arXiv:1705.00652.

[16] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3560815. doi:10.1145/3560815.

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[18] F. Rangel, M. Chinea-Rios, M. Franco-Salvador, P. Rosso, PAN23 Profiling Cryptocurrency Influencers with Few-shot Learning, 2023. URL: https://doi.org/10.5281/zenodo.7701748. doi:10.5281/zenodo.7701748.

[19] H. B. Giglou, M. Rahgouy, A. Rahmati, T. Rahgooy, C. D. Seals, Profiling Irony and Stereotype Spreaders with Encoding Dependency Information using Graph Convolutional Network, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022. URL: http://ceur-ws.org/Vol-3180/paper-190.pdf.

# A. Apendix

**Table 3**
Designed prompt templates for Subtask 1, Subtask 2, and Subtask 3.

| Subtask | Prompt Templates |
|---|---|
| *Subtask 1* | Identify cryptocurrency influencers profiles from given tweets: \n\n Tweets: {tweets} |
| | User tweets: "{tweets}" \n\n Question: What is the profile of this user on twitter? |
| | {tweets} \n\n What profile is appropriate for this user from a cryptocurrency perspective? |
| | {tweets} \n\n Is this a cryptocurrency influencers? |
| | Given a collection of tweets from a user: "{tweets}" \n\n What is the user profile as a cryptocurrency influencers? |
| | What is the user related aspect of the influencer using the following tweets?? \n\n Tweets:{tweets} |
| | Given the following user tweets, determine the profile of this user as a cryptocurrency influencer: \n\n User tweets: {tweets} |
| | Consider the tweets provided: "{tweets}" \n\n What would be an appropriate profile for this user from a cryptocurrency perspective? |
| | A user has posted the following collection of tweets: "{tweets}" \n\n What is the user's profile as a cryptocurrency influencer? |
| | Evaluate the given tweets to identify cryptocurrency influencers: \n\n Tweets: {tweets} |
| *Subtask 2* | Identify the user interest in cryptocurrency from the given tweets: \n\n Tweets: {tweets} |
| | Analyze the given tweets to identify if the user has a particular interest in cryptocurrency. \n\n Tweets: {tweets} |
| | User tweets: "{tweets}" \n\n Question: What is the user interest in cryptocurrency? |
| | Given collection of tweets from a user: "{tweets}" \n\n What is the user interest in cryptocurrency influencers? |
| | {tweets} \n\n Examine the tweets and determine if the user exhibits an interest in cryptocurrency. |
| | {tweets} \n\n From the provided tweets, ascertain whether the user shows interest in following or engaging with cryptocurrency? |
| | Evaluate the given tweets to identify the user's interest in cryptocurrency:\n\n Tweets: {tweets} |
| | Given the following user tweets, determine the user interest: \n\n User tweets: {tweets} |
| | Consider the tweets provided: "{tweets}" \n\n Identify the user interest? |
| | A user has posted the following collection of tweets: "{tweets}" \n\n What is the user's preference in the cryptocurrency? |
| *Subtask 3* | Identify the user intent in cryptocurrency from the given tweets: \n\n Tweets: {tweets} |
| | Analyze the given tweets to identify if the user has a particular purpose in cryptocurrency. \n\n Tweets:{tweets} |
| | User tweets: "{tweets}" \n\n Question: What is the user intent in cryptocurrency? |
| | Given collection of tweets from a user: "{tweets}" \n\n What is the user purpose of cryptocurrency influencers? |
| | {tweets} \n\n Examine the tweets and determine if the user exhibits an intent in cryptocurrency. |
| | {tweets} \n\n From the provided tweets, ascertain whether the user shows purpose in following or engaging with cryptocurrency? |
| | Evaluate the given tweets to identify the user's intent in cryptocurrency: \n\n Tweets: {tweets} |
| | Given the following user tweets, determine the user aim in cryptocurrency: \n\n User tweets: {tweets} |
| | Consider the tweets provided: "{tweets}" \n\n Identify the user intent? |
| | A user has posted the following collection of tweets: "{tweets}" \n\n What is the user's goal in the cryptocurrency? |

**Table 4**
Hypotheses templates for labels in Subtask 1, Subtask 2, and Subtask 3.

| Subtask | Hypotheses Templates |
|---|---|
| *Subtask 1* | This user profile in cryptocurrency is a no influencer. |
| | This user profile in cryptocurrency is a nano. |
| | This user profile in cryptocurrency is a micro. |
| | This user profile in cryptocurrency is a macro. |
| | This user profile in cryptocurrency is a mega. |
| *Subtask 2* | This influencer interest is a technical information. |
| | This influencer interest is a price update. |
| | This influencer interest is a trading matters. |
| | This influencer interest is a gaming. |
| | This influencer interest is a other. |
| *Subtask 3* | This influencer intent is a subjective opinion. |
| | This influencer intent is a financial information. |
| | This influencer intent is a advertising. |
| | This influencer intent is a announcement. |