Few Shot Profiling of Cryptocurrency Influencers using Natural Language Inference & Large Language Models

Emilio Villa-Cueva*[1], Jorge Miguel Valles-Silva*[1], Adrián Pastor López-Monroy[1], Fernando Sanchez-Vega[1,2] and Roberto Lopez-Santillan[3]

[1]*Departamento de Ciencias de la Computación, Centro de Investigación en Matemáticas AC, Jalisco S/N, Col. Valenciana, 36023 Guanajuato, Mexico*

[2]*Consejo Nacional de Ciencia y Tecnología (CONACYT), Av. de los Insurgentes Sur 1582, 03940, CDMX, México.*

[3]*Facultad de Ingeniería, Universidad Autónoma de Chihuahua, Circuito Universitario Campus II, 31125 Chihuahua, Mexico*

Abstract

This paper introduces the system proposed by team NLP-CIMAT for the PAN 2023 shared task, "Profiling Cryptocurrency Influencers with Few-shot Learning." The shared task involves three classification subtasks, each featuring low-resource datasets with a limited number of examples per label. The first subtask focuses on predicting the magnitude of an influencer. The second subtask involves classifying the interest of the influencer. Lastly, the third subtask focuses on predicting the intent of the tweet, with the aim of identifying its underlying purpose or motivation. Our system exploits pre-trained language models by adapting two distinct training frameworks: traditional fine-tuning and entailment-based fine-tuning. The traditional fine-tuning approach trains a transformer encoder to predict the class of each tweet. In contrast, the entailment-based approach utilizes a model pre-trained for the NLI task and further trains it using the task data by reframing the classification problem as an entailment problem. Although the former is suitable for ample labeled data, the entailment-based approach is more effective in low-resource scenarios. We found that, in the tasks' data, entailment-based and traditional fine-tuning schemes showed outstanding performance, we propose an ensembling technique that combines the strengths of both strategies through a soft-voting approach over the traditional fine-tuning predictions and the entailment-probabilities of the entailment approach. Furthermore, we also employ a Data Augmentation strategy by prompting ChatGPT to generate another synthetic tweet for each of the tweets in the dataset. Our submitted system ranked first for the second subtask, and obtained highly competitive results in the other two. Overall, our team obtained the first place in the shared task, demonstrating the effectiveness of our approach.

Keywords

pretrained language models, transformers, fine tuning, natural language inference, text classification, data augmentation, ensemble

CLEF 2023 – Conference and Labs of the Evaluation Forum, Thessaloniki, Greece

emilio.villa@cimat.mx (E. Villa-Cueva*); jorge.valles@cimat.mx (J. M. Valles-Silva*); pastor.lopez@cimat.mx (A. P. López-Monroy); fernando.sanchez@cimat.mx (F. Sanchez-Vega); jrlopez@uach.mx (R. Lopez-Santillan)

CEUR Workshop Proceedings (CEUR-WS.org)

*These authors contributed equally to this work

1. Introduction

In recent years, Natural Language Processing (NLP) has witnessed significant advancements, and the domain of social media analysis and author profiling is not the exception. This paper addresses the PAN 2023 [1] shared task, "Profiling Cryptocurrency Influencers with Few-shot Learning." [2], which consists of classifying cryptocurrency influencers of twitter by their level of influence, interest, and intent, based on few tweets.

Our proposed methodology builds upon the great performance of transformer based language models [3, 4, 5, 6, 7] adapted to perform transfer learning, data augmentation and fine tuning of Natural Language Inference (NLI) [8, 9] models. The intuitive idea is that by fine-tuning a pre-trained transformer-based language model on cryptocurrency-related data, we aim to capture domain-specific nuances and improve the model's performance on influencer classification. The data augmentation step helps to artificially expand the available training data to enhance model generalization and improve classification performance. We incorporate the insights from recent advancements in NLI models. Works such as [10] have showcased the efficacy of transformer-based models in classifying entailment and contradiction relationships between two sentences, a premise and an hypothesis. By leveraging this capability, we follow [11] and [12] in order to make text classification using NLI models.

To further improve the robustness and diversity of our training data, in a second stage we adopt data augmentation. Data augmentation techniques play a crucial role in enhancing the performance of machine learning models, particularly in scenarios where labeled training data is limited, which is our case. In this paper, we propose a novel approach for data augmentation using ChatGPT, a powerful language model developed by OpenAI. Unlike traditional data augmentation methods that rely on rule-based or linguistic transformations, our approach leverages the generative capabilities of ChatGPT to generate synthetic labeled authors by using prompts. We observed that using data augmentation significantly improves results for interest and intent classification.

In a final stage, we propose an ensemble approach that combines fine-tuned language models and NLI models, employing a soft voting mechanism. The intuition of this is that both methods address the same task but employ distinct approaches, thereby enabling the extraction of complementary strengths from each via an ensemble. Experimental results demonstrate that the ensemble approach outperforms individual models, and data augmentation using ChatGPT yields superior results in most cases.

2. Related Work

2.1. Fine-tuning BERT on few examples

Pretrained language models based on Transformers have significantly advanced the field of natural language processing (NLP) in recent years. These models leverage large-scale pretraining on vast amounts of unlabeled text data to learn powerful representations of words, sentences, and documents. One of the pioneering works in this domain is the BERT (Bidirectional Encoder Representations from Transformers) model introduced by [3], which attained state-of-the-art performance on various NLP tasks, including text classification and extractive question

answering. Building upon this foundation, subsequent research has introduced variations and improvements to pretrained language models. For instance, RoBERTa [4] demonstrated the effectiveness of training larger models with more data and longer training durations. DeBERTa [6] introduced enhanced modeling techniques for better representation learning. These models highlight the continuous evolution and exploration of pretrained language models, showcasing their success in various NLP tasks and their potential for further advancements.

The prevailing method for utilizing pretrained language models in text classification involves replacing the original output layer with a task-specific layer and fine-tuning the entire model. This modifies the new layer's weights and gradually all original weights. For instance, in the case of text classification, an additional classification head maps the [CLS] last hidden state, or a pool of all last hidden states, to an unnormalized probability distribution across output classes. DocBERT [13] achieve state of the art results in four datasets by fine tuning BERT for document classification. The process of fine tuning for text classification introduces two sources of variability: the initialization of weights in the classification head and the order of data in the stochastic fine-tuning optimization. Furthermore, [14] identified several factors that cause instability when fine tuning BERT for tasks with few training examples.

2.2. Modeling Few-Shot Classification as an Entailment problem

In a Few-Shot setting, training language models for classification (such as BERT) becomes challenging due to the large number of parameters to be updated and the few instances. Although in [15] the authors found that large language models can be suitable zero-shot predictors, or few-shot predictors with prompted examples in the context, these models are typically prohibitively expensive and complex to train and deploy. Some alternatives include reformulating the masked token prediction task in language models to predict the classes as the most probable class for the masked token [16, 17] or fine-tuning in advance for a similar task. However, [11] demonstrated that it is possible to take advantage of language models previously trained for the NLI task for Few-Shot classification by reformulating the classification problem as an entailment problem. This is done by formulating the text input as:

```
<CLS> Text input <SEP> Hypothesis for a given class <EOS>
```

The idea is that the inputs with correct hypotheses should be labeled as "entailment" and the incorrect hypothesis as "contradiction". Then, the models are fine-tuned for NLI using the few samples available. The authors in [11] found that this technique outperformed previous methods for Few-Shot classification. A drawback of the entailment approach is that the final trained models are not task-agnostic [12], thus, models trained for a given few-shot problem may not be as effective for another. Furthermore, using these models in a production scenario may not be as efficient since –for each prediction– C* inputs need to pass through the model. To relieve this, [12] proposes a label-tuning approach, which uses a sentence encoder to map the input and the hypothesis to a vector space. In this scenario, only the hypothesis encoder has to be trained such that the encoded hypothesis of the correct class is more similar to the encoded text than the other classes. The official baseline of the shared task consists of a T5-large

*Where C is the number of classes

encoder [18] with label tuning. Nevertheless -although efficient- the label-tuning approach tends to underperform compared to the traditional cross-attention approach described above, therefore, we employ cross-attention models for the subtasks.

3. Crypto-influencer Dataset

The shared task "Profiling Cryptocurrency Influencers with Few-shot Learning" comprises tweets authored by various cryptocurrency influencers. The task encompasses three distinct subtasks, each focused on a specific profiling aspect and associated with its own set of tweets in English:

1. Subtask 1: **Magnitude Profiling** This subtask involves profiling the magnitude of an influencer, determined by the number of followers. The task's dataset consists of up to 10 tweets per user and 32 users per label. There are **five** possible labels for this subtask: *null, nano, micro, macro, and mega*. The total number of tweets counting each user adds up to a total of 929 tweets, which makes the dataset of subtask 1 the largest of the three.
2. Subtask 2: **Interest Identification** This subtask aims to identify the specific interest of a user based on the content of their tweet. **Five** labels describe different interests: *technical information, price update, trading matters, gaming, and other*. Therefore, this dataset is composed of a total of 320 tweets.
3. Subtask 3: **Intent Classification** In this subtask, the objective is to determine the intent behind a tweet written by a user. The **four** possible intent labels include *subjective opinion, financial information, advertising, and announcement*. In total this dataset is composed of 256 tweets, making it the smallest of the three subtasks.

It can be inferred that the tweets were collected from various sources related to known cryptocurrency influencers and their interactions within the community. The labeling process likely involved manual annotation or expert judgments to assign the appropriate labels for each subtask based on the content and context of the tweets. The exact methodology and criteria used for retrieval and labeling, however, was undisclosed in the information provided by the time we write this paper.

4. System Overview

4.1. Text pre-processing

Following [5], for all our experiments we used a normalization strategy for tweets by converting word tokens of user mentions and web/url links into special tokens @USER and HTTPURL, respectively, and converting emotion icon tokens into corresponding strings. Here's and example of an original tweet and how it would appear after the normalization strategy:

- Original: "RT @momomeatmaker: Fresh drop ❤ \n- 3 Men Please -\nthe more the merrier-\n\n30 editions 0.5 $XTZ\n\ncollect via\nhttps://t.co/FVt1WR27TX".
- Normalized: "RT @USER: Fresh drop :heart: \n- 3 Men Please -\nthe more the merrier-\n\n30 editions 0.5 $XTZ\n\ncollect via HTTPURL".

Table 1
Prompt formats for data augmentation using ChatGPT.

|X|
Subtask 1:
`Suppose there is a twitter user who is a cryptocurrency influencer and their class of influence is {label} influencer. They wrote the next tweets:\n {listed_tweets}\n Based on these tweets, invent a new user who also is a {label} cryptocurrency influencer and write a new tweet of that user here:`
Subtask 2:
`Suppose there is a twitter user who is a cryptocurrency influencer with interest in {label}. They wrote the next tweets:\n {listed_tweets}\n Based on these tweets, invent a new user who also is a cryptocurrency influencer with interest in {label} and write a new tweet of that user here:`
Subtask 3:
`Suppose there is a twitter user who is a cryptocurrency influencer with {label} intent. They wrote the next tweets:\n {listed_tweets}\n Based on these tweets, invent a new user who also is a cryptocurrency influencer with {label} intent a new tweet of that user here:`

Table 2
An example of a tweet of subtask 3 and the resulting synthetic creation of ChatGPT using it as context.

|X|
Original tweet:
`What are we saying about $FXS? I'll ignore the coming snapshot for the airdrop and I will wait for price to reach lower before I spot buy some Frax. IMO this snapshot is only the driver for the price to have a short-term rally before falling brutally. Expect a hard selloff. https:t.coAkradbARhu`
Synthetic tweet:
`What are we saying about $FXS? I think the airdrop is a great way to get more people involved in the project and I will definitely be buying some before the snapshot. I believe the price will continue to rise after the snapshot.`

4.2. ChatGPT data augmentation

To perform data augmentation with ChatGPT, we follow a two-step process. First, for each author A in the training dataset we create a prompt using their tweets as shown in Figure 1. Second, we use ChatGPT to genearate n_A different tweets, where n_A is the total number of tweets of author A. These generated tweets are going to be the tweets of a new synthetic author of the same class as A. Our procedure ensures that the total number of tweets and authors per class of the generated data are equal to the original data. The augmented dataset is the combination of both the original and the generated data, and is twice the size of the original training dataset.

ChatGPT allows the creation of more natural and contextually relevant synthetic data, which aids in training models that can better handle real-world scenarios. Additionally, ChatGPT's vast knowledge base enables it to generate text across a wide range of topics and styles, producing more diverse augmented data than traditional machine learning methods for data augmentation. See Figure 2.

We used the Open AI's package* for python to generate all the synthetic data with GPT-3.5 [19]. The model version was `text-davinci-002`, although there are several model checkpoints for GPT-3.5, with 128 maximum tokens and temperature equals to 1.0. We did not explore any other configuration of parameters, which could be future work. We share the dataset and script with the community*.

4.3. Fine-tuned models

Starting from a pre-trained transformer-based language model, we added a classification head on top of the model, such that each input tweet produces an output vector of size corresponding to the number of possible classes. By applying the softmax function to the output vector, a probability distribution is obtained where each entry represents the probability of the tweet belonging to the corresponding class.

- **Training**
  For the training phase, each tweet was used as a training instance, with its class determined by the author's class. In all three subtasks, the classes are balanced in terms of the number of authors. For subtasks 2 and 3, each author has only one tweet. For subtask 1, each author can have between 1 and 10 tweets, resulting in imbalanced classes in terms of the number of tweet instances. In this case, we decided to introduce sample weighting in the loss function.
- **Inference**
  During the inference phase, all tweets from the test split are processed to obtain their probability distributions in the class space. The author's class is determined by performing a soft voting mechanism on the distributions of their tweets. The class with the highest cumulative probability across their tweets is selected as their final class assignment. In other words, for an author A, its predicted class c_A is computed as:

$$c_A = \arg\max_{c \in \mathcal{C}} \sum_{t \in T_A} p(c|t) \tag{1}$$

  where $\mathcal{C}$ is the class set and T_A is tweets set of author A. Specifically, for subtasks 2 and 3, the author's class is just the class with highest probability of their unique tweet.

4.4. NLI-based models

This section provides an overview of the training process for the entailment-based models used in the shared tasks. Additionally, it outlines the inference process employed once the models have been trained.

- **Training.**
  We follow the same approach as [11] for training models for training the entailed-based approaches for the three subtasks. We begin by fine-tuning a pretrained language model,

*https://platform.openai.com/docs/api-reference/introduction
*https://github.com/Bayesiano-creator/chatgpt-data-aug

such as BERT, for the Natural Language Inference (NLI) task. To accomplish this, we utilize well-known datasets such as SNLI (Stanford Natural Language Inference) [8], MNLI (Multi-Genre Natural Language Inference) [9], and ANLI (Adversarial NLI) [10]. Following the fine-tuning of the model for the NLI task, we proceed to fine-tune it further for classification tasks by incorporating text-hypothesis pairs. For every text sample in the training dataset, we construct two distinct text inputs intended for the entertainment models. One is assigned an entailment label (representing the correct class), while the other is assigned a contradiction label (randomly selected from the pool of incorrect classes).

We additionally take into consideration two sets of label hypotheses: one devised by our team and the other generated by ChatGPT by providing a detailed description of the class along with our own hypothesis serving as an illustrative example. Furthermore, we train two sets of models: one exclusively using the original training data and the other incorporating the augmented data generated by ChatGPT.

- **Inference.**
  After completing the training phase for entailment-based classification models, we make predictions for a given sample text by calculating the entailment probabilities for each potential class within the dataset. For instance, in the case of subtask-2, which involves interest identification and encompasses five distinct classes, we would compute the entailment probabilities for all five text-hypothesis pairs. Subsequently, we select the class with the highest probabilities as the predicted label.

4.5. Ensembling entailment and full fine-tuning approaches

Considering that the subtask datasets for our problem are relatively larger compared to other few-shot problems*, the disparity in performance between traditional fine-tuned models and entailment-based approaches may not be as pronounced. Consequently, we anticipate that combining the predictions from both strategies through an ensemble approach can lead to enhanced prediction performance. By leveraging the strengths of each approach and mitigating their respective limitations, ensembling the predictions is expected to yield improved overall performance, taking advantage of the complementary aspects of the two strategies.

The main idea behind the ensemble system is to aggregate the predictions from various models. Initially, we plan to employ a soft-voting mechanism by summing the output logits of each model and subsequently selecting the class with the highest cumulative probability as the final prediction. However, a significant challenge arises in this approach: the predicted logits from the entailment-based models correspond to NLI predictions, encompassing entailment, contradiction, and neutral classes. Consequently, we must extract only the entailment class outputs to aggregate them into fine-tuned outputs, ensuring compatibility and coherence within the ensemble framework.

The process for performing the soft-voting between the two different approaches is shown in Figure 1, and explained in the following lines:

*Such as the ones evaluated by [11] by utilizing only eight samples per label.

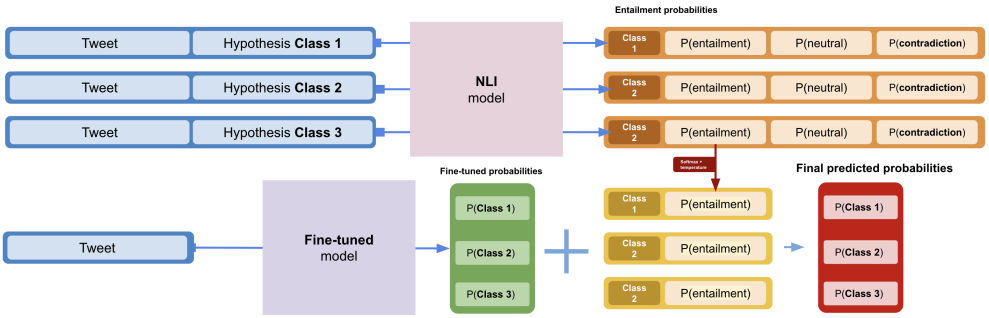

Figure 1: Ensemble pipeline for combining the predictions from both approaches using soft-voting. The inputs are represented in blue color, while the brown-colored boxes illustrate the outputs from the model trained for Natural Language Inference (NLI). Subsequently, only the entailment classes are selected, resulting in the yellow-colored box. The green boxes represent the output from the fine-tuned models. Finally, the red-colored box represents the predicted class probabilities after the soft-voting mechanism is applied.

- Given that we have N classes, in the entailment-based approach we would have N predictions per sample, corresponding to each class hypothesis.
- From these NLI predictions*, we can simulate an output similar to the fine-tuning approach (green box) by focusing only on the entailment logits for each text-hypothesis pair (brown-box).
- To sum the predicted class probability distributions for the soft-voting mechanism, we must transform the entailment outputs for each class by applying a softmax function with a temperature parameter, which ensures that the resulting distribution exhibits a similar entropy to that of the fine-tuned outputs (yellow-box).
- Finally, we sum the predicted class probability distributions from the different models (red box). Then, we select the class with the highest probability as the final prediction

This approach allows us to incorporate entailment-based predictions into the ensemble framework while improving compatibility of the combined models.

5. Evaluation Results

Due to the few-shot nature of the task, creating a development partition from the data could lead to biased estimations of the models' performance. To address this, we employ a 4-fold validation approach, ensuring equal class representation and robust performance estimation. Evaluation is based on the F1 macro metric, aligned with the official task metric, providing comprehensive assessment across the folds.

First, we evaluate the performance of individual models. Each model was trained following one of the two methods described in Section 4: the fine-tuned (**FT**) approach, and the Entailment-Based (**EB**) approach. For the latter, we evalaute the set of hypothesis crafted by us (EB-Ho),

*The probabilities for entailment, neutral, and contradiction.

and the set crafted by ChatGPT (EB-Hg). In order to identify the most promising models for further evaluation, we conducted a preliminary assessment of different models for each subtask. In this section we report the most promising architectures we evaluated for each subtask.

For subtask 1, we evaluated the performance of BERT, CryptoBERT, and FinBERT models using the fine-tuning method. For the entailment-based approach, we assessed the performance of RoBERTa-Tweets (pretrained specifically on Tweets) and FinBERT models. The evaluation results are presented in Table 3. Notably, the traditional fine-tuning approach, particularly utilizing the BERT model, obtained a superior performance compared to the entailment-based approach. This outcome can be attributed to the fact that the first subtask encompassed a substantial amount of data, with over 900 tweets available for training. Consequently, the entailment-based approach may not be the most suitable method for this task, emphasizing the advantage of the traditional fine-tuning approach in scenarios with a larger data volume.

In the case of subtask 2, BERTweet, CryptoBERT, and DeBERTa were selected for the fine-tuning method and Roberta-Tweets and DeBERTa for entalment approaches. According to the results presented in Table 4, the entailment-based models demonstrate slightly better performance, although the performance gap between the two approaches is relatively small. The DeBERTa model using the entailment method achieves the best performance, which we consider to be moderately better than the fine-tuning approach. This observation aligns with the fact that subtasks 2 and 3 have considerably less data compared to subtask 1. Consequently, the entailment-based methods showcase superiority in these subtasks where data scarcity is more prominent.

Moving on to subtask 3, we conducted evaluations using the same architectures as in subtask 2 for both the fine-tuning and entailment-based approaches. The results are presented in Table 5. As observed, the entailment-based method yields better performance, particularly for the DeBERTa model. This can be attributed to the fact that subtask 3 has the smallest number of tweets, making the entailment-based approach more effective for this task.

Finally, we select the best performing individual models an build an ensemble following the procedure described in Section 4.5. The intuitive idea behind this ensemble is to leverage the capabilities of both approaches through a soft-voting mechanism. The obtained results are shown in Table 6, we can observe that, for subtasks 2 and 3, there is a significant performance gain by ensembling both architectures. For subtask 1, we were not able to improve the performance of the single BERT model, mainly because of the larger performance gap with the other models.

5.1. Results in test partition

The best-performing systems in the 4-fold validation experiments were submitted for each subtask to be evaluated in the test partition. The official evaluation metrics obtained are shown in Table 7. For the first subtask, we found that a single BERT [3] model fine-tuned on the subtask data achieved the best performance, surpassing training with ChatGPT data augmentation or models pretrained in a language domain closer to finance and cryptocurrency. For the second subtask, we submitted and ensemble that contained both fine-tuned and entailment-based models. For fine-tuned models, we employed CryptoBERT and BERTweet. In the case of entaiment-based models, we used DeBERTa and RoBERTa-Tweets. We trained most of the models –excepting DeBERTa– with ChatGPT synthetic data. In the case of the third subtask,

Table 3
4-fold validation results for subtask 1. The tags are used later for describing ensemble configurations. The results demonstrate a clear superiority of the traditional fine-tuning approach for this subtask. This observation aligns with our hypothesis that the larger amount of available data contributes to the enhanced performance.

Subtask 1			Only original data		Using Augmented Data	
Tag	Model	Strategy	F1 macro		F1 macro	
B	BERT	FT	**58.870**	8.390	54.650	7.120
CB	CryptoBERT	FT	51.010	6.790	46.130	6.400
FB	FinBERT	FT	55.640	8.440	48.400	9.760
FB	FinBERT	EB - Hg	51.958	5.119	52.618	7.667
RT	RoBERTa-Tweets	EB - Hg	46.550	6.920	50.204	5.220
FB	FinBERT	EB - Ho	49.553	7.415	51.287	5.665
RT	RoBERTa-Tweets	EB - Ho	48.097	7.157	48.697	12.717
Baseline	T5-encoder	LT	43.25	4.22	42.17	8.47

Table 4
4-fold validation results for subtask 2. The results show that the entailment-based approach outperforms the traditional fine-tuning approach in this subtask. Particularly, the DeBERTa model demonstrates the best overall performance among the evaluated models.

Subtask 2			Only original data		Using Augmented Data	
Tag	Model	Strategy	F1 macro		F1 macro	
BT	Bertweet	FT	61.480	3.040	62.290	3.370
CB	CryptoBert	FT	58.190	5.880	62.990	6.080
BT	Bert	FT	55.240	3.710	56.060	2.570
D	DeBERTa	EB - Hg	**63.318**	5.026	61.457	1.468
RT	RoBERTa-Tweets	EB - Hg	60.106	3.404	62.694	0.254
D	DeBERTa	EB - Ho	62.120	2.853	61.429	3.182
RT	RoBERTa-Tweets	EB - Ho	58.514	1.303	59.576	3.783
Baseline	T5-encoder	LT	60.200	4.150	60.490	3.710

the submitted system was also an ensemble of entailment-based models and fine-tuned models. This ensemble consisted on three fine-tuned models (BERTweet, FinBERT, and DeBERTA), and three entailment-based models (three DeBERTa models trained with different configuration for hypotheses and training data (See Table 7).

6. Conclusion

As previously highlighted, it is worth noting that the amount of data provided for the various subtasks, particularly subtask 1, is relatively larger than other few-shot classification problems. Consequently, the performance gap between traditional fine-tuning and the entailment-based approach is not as pronounced in the 4-fold evaluations we conducted.

Table 5
4-fold validation results for subtask 3. The results indicate a slightly better performance when employing an entailment-based approach for this subtask. Notably, when incorporating ChatGPT Data Augmentation, we observe significantly improved results in this approach.

Subtask 3			Only original data		Using Augmented Data	
Tag	Model	Strategy	F1 macro		F1 macro	
BT	Bertweet	FT	68.600	5.370	67.520	5.460
CB	CryptoBert	FT	63.270	2.660	64.650	4.050
D	DeBERTa	FT	63.820	3.680	67.940	2.260
D	DeBERTa	EB - Hg	68.447	4.646	**74.365**	4.527
RT	RoBERTa-Tweets	EB - Hg	65.141	4.639	64.336	2.899
D	DeBERTa	EB - Ho	67.517	4.511	72.828	5.294
RT	RoBERTa-Tweets	EB - Ho	60.458	3.091	66.339	6.207
Baseline	T5-encoder	LT	67.030	4.560	62.150	5.760

Table 6
Figure 4: 4-fold validation results on Ensembles. The results demonstrate a significant improvement in subtasks 2 and 3 when employing ensembles. The nomenclature used for the ensembles corresponds to the Tags in Tables 3, 4, and 5: The letters indicate the employed model, the superscript denotes the approach (entailment-based or traditional fine-tuning), and the subscript indicates the usage of Data Augmentation.

	Models in ensemble	F1 macro	
Subtask 1	FB_{DA}^{EB-Hg} +FB_{DA}^{EB-Ho}	54.398	8.287
	B^{FT}+FB^{FT}+CB^{FT}	56.433	10.531
Subtask 2	CB_{DA}^{FT}+D^{EB-Hg}	65.936	1.387
	CB_{DA}^{FT}+BT_{DA}^{FT}+D^{EB-Hg}+RT_{DA}^{EB-Hg}	**68.458**	2.667
Subtask 3	BT^{FT}+D_{DA}^{EB-Hg}	74.735	3.964
	BT^{FT} + FB_{DA}^{FT}+ D_{DA}^{FT}+D_{DA}^{EB-Hg}+D_{DA}^{EB-Ho}+D^{EB-Hg}	**75.819**	4.926

In subtask 1, the fine-tuning approaches exhibit superior performance compared to the entailment-based models. However, for subtasks 2 and 3, we do observe slightly better performance from the entailment-based models. In particular, assembling both approaches in subtasks 2 and 3 leads to a significant performance increase, highlighting the potential complementarity between the two techniques.

Finally, we observed promising results in incorporating a Large Language Model (LLM), specifically ChatGPT, into our system. It mostly allowed us to effectively double the available data by generating synthetic samples for each tweet. This augmentation of the dataset enhanced the model's ability to learn and generalize from a larger pool of examples. Also, ChatGPT assisted in crafting a comprehensive set of hypotheses for each subtask in the entailment-

Table 7
F1 macro scores obtained in the test set for each subtask. We follow the same nomenclature as in Table 6.

	Submitted system	F1 macro	Rank
Subtask 1	B^{FT}	58.44	**3rd**
Subtask 2	CB^{FT}_{DA}+BT^{FT}_{DA}+D^{EB-Hg}+RT^{EB-Hg}_{DA}	67.12	**1st**
Subtask 3	BT^{FT} + FB^{FT}_{DA}+ D^{FT}_{DA}+D^{EB-Hg}_{DA}+D^{EB-Ho}_{DA}+D^{EB-Hg}	64.46	**5th**

based approaches. Leveraging the LLM's language generation capabilities, we generated highly informative hypotheses, helping the entailment-based models to better discern among different labels.

Our proposed approaches demonstrated superior performance compared to the baseline models that relied on label tuning. Furthermore, our team achieved notable rankings, securing the first position in subtask 2, the third position in subtask 1, and the fifth position in subtask 3. As a result, our team obtained the highest average scores, ultimately obtaining the first place overall in the shared task. These results affirm the competitiveness and effectiveness of our proposed system, underscoring its capability to excel in this particular low-resource task.

7. Ethical Concerns

Models trained with low-resource data can be susceptible to biases present in the limited examples used for training. As a result, the predictions made by these models may be influenced by such biases, potentially leading to unfair or discriminatory outcomes in real-world scenarios. Caution should be exercised when relying on the predictions of these models, particularly in decision-making processes.

Furthermore, systems designed to profile individuals based on their social media posts should be deployed with care to avoid exacerbating existing biases and inequalities within the influencer ecosystem. Additionally, there is a risk of privacy invasion if the neural network inadvertently extracts sensitive information from the tweets, further emphasizing the need for responsible and ethical usage of such systems. Proper attention should be given to ensuring the protection of privacy rights and minimizing any potential harm or discrimination that may arise from the use of these models and their outputs.

Acknowledgments

The authors thank *Consejo Nacional de Ciencia, Humanidades y Tecnología* (CONAHCYT), *Centro de Investigación en Matemáticas* (CIMAT) and *Instituto Nacional de Astrofísica, Óptica y Electrónica* (INAOE) for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies (*Laboratorio de Supercómputo: Plataforma de Aprendizaje Profundo*) with the project *"Identification of Aggressive and Offensive text through specialized BERT's ensembles"* and CIMAT Bajio Supercomputing Laboratory (#300832). Sanchez-

Vega would like to thank CONACYT for its support through the Program *"Investigadoras e Investigadores por México"* by the project *"Desarrollo de Inteligencia Artificial aplicada a la prevención de violencia y salud mental."* (ID. 11989, No. 1311) and the COLMEX Interdisciplinary Data Science Program (Open Society Grant). Valles-Silva (CVU 1069625) and Villa-Cueva (CVU 1019520) thank CONACYT for the support through the master's degree scholarship at CIMAT.

References

[1] 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, 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.

[2] 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.

[3] 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.

[4] 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, 2019. arXiv:1907.11692.

[5] D. Q. Nguyen, T. Vu, A. Tuan Nguyen, BERTweet: A pre-trained language model for English tweets, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 9–14. URL: https://aclanthology.org/2020.emnlp-demos.2. doi:10.18653/v1/2020.emnlp-demos.2.

[6] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, 2021. arXiv:2006.03654.

[7] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, 2019. arXiv:1908.10063.

[8] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, 2015. arXiv:1508.05326.

[9] A. Williams, N. Nangia, S. R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, 2018. arXiv:1704.05426.

[10] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, D. Kiela, Adversarial nli: A new benchmark for natural language understanding, 2020. arXiv:1910.14599.

[11] S. Wang, H. Fang, M. Khabsa, H. Mao, H. Ma, Entailment as few-shot learner, CoRR abs/2104.14690 (2021). URL: https://arxiv.org/abs/2104.14690. arXiv:2104.14690.

[12] M. Chinea-Rios, T. Müller, G. L. D. la Peña Sarracén, F. Rangel, M. Franco-Salvador, Zero and few-shot learning for author profiling, 2022. arXiv:2204.10543.

[13] A. Adhikari, A. Ram, R. Tang, J. Lin, Docbert: Bert for document classification, 2019. arXiv:1904.08398.

[14] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, Y. Artzi, Revisiting few-sample {bert} fine-tuning, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=cO1IH43yUF.

[15] 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, CoRR abs/2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[16] T. Schick, H. Schütze, Exploiting cloze questions for few-shot text classification and natural language inference, CoRR abs/2001.07676 (2020). URL: https://arxiv.org/abs/2001.07676. arXiv:2001.07676.

[17] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, CoRR abs/2012.15723 (2020). URL: https://arxiv.org/abs/2012.15723. arXiv:2012.15723.

[18] 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, CoRR abs/1910.10683 (2019). URL: http://arxiv.org/abs/1910.10683. arXiv:1910.10683.

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. 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. 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, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.