SMM4H'24 Task6 : Extracting Self-Reported Age with LLM and BERTweet: Fine-Grained Approaches for Social Media Text

Jaskaran Singh, Jatin Bedi,Maninder Kaur
Thapar Institute of Engineering and Technology
jsingh7_be21@thapar.edu, jatin.bedi@thapar.edu, manindersohal@thapar.edu

Abstract

The paper presents two distinct approaches to Task 6 of the SMM4H'24 workshop: extracting self-reported exact age information from social media posts across platforms. This research task focuses on developing methods for automatically extracting self-reported ages from posts on two prominent social media platforms: Twitter (now X) and Reddit. The work leverages two ways, one Mistral-7B-Instruct-v0.2 Large Language Model (LLM) and another pre-trained language model BERTweet, to achieve robust and generalizable age classification, surpassing limitations of existing methods that rely on predefined age groups. The proposed models aim to advance the automatic extraction of self-reported exact ages from social media posts, enabling more nuanced analyses and insights into user demographics across different platforms.

1 Introduction

The widespread use of social media platforms by individuals across demographics offers a unique opportunity to gain valuable insights into their health experiences and perspectives. Effectively leveraging social media data for research purposes necessitates the development of methods for automatically extracting demographic information, such as user age, with high accuracy. Existing methods for identifying user age on social media platforms often rely on categorizing users into predefined age groups. As computational analysis offers new possibilities for investigating complex subjects through social media data, models are being created to automatically identify demographic information, such as the age of users(Klein et al., 2022) (Sadeghi et al., 2024). Many studies have tackled age detection through automated methods, primarily using binary or multi-class classification of predetermined age categories(Chew et al., 2021). These approaches typically involve identifying users' ages from their posts, profile information, or external data sources and subsequently predicting their age groups based on various factors such as their social media activity profile details or a combination of both (Morgan-Lopez et al., 2017); however, the diversity and inconsistency in the number and scope of age groups used across studies indicate that these methods may not be universally applicable to all scenarios. Accurately pinpointing the exact age of social media users, instead of placing them into broad age categories, would enable the extensive use of social media data for applications needing precise age information. This would be particularly beneficial for identifying specific age-related risk factors in observational studies, which current binary or multi-class models fail to address. This work applies two strategies, one using Mistral-7B-Instruct-v0.2 Large Language Model (LLM) and another pre-trained language model BERTweet for automatically extracting self-reported exact age information from social media posts on two prominent platforms: Twitter and Reddit. The proposed models aim to address the issues of existing methods by directly identifying the user's exact age as expressed in the text, enabling the large-scale utilization of social media data for a wider range of research applications. \documentclass declaration and before \begin{document}) using \usepackage{graphicx}.

2 Data description

The training dataset consists of 8,800 labeled tweets and 100,000 unlabeled Reddit posts, primarily focused on age-related information. The labeled tweets indicate whether the user's exact age could be determined from the text, with "1" indicating explicit or inferred age and "0" otherwise. Validation set includes 2,200 labeled tweets and 1,000 Reddit posts about dry eye disease, while the testing set includes 2,200 labeled tweets, 2,000

Proceedings of The 9th Social Media Mining for Health Applications (#SMM4H) Workshop and Shared Tasks, pages 106–109
August 15, 2024 ©2024 Association for Computational Linguistics

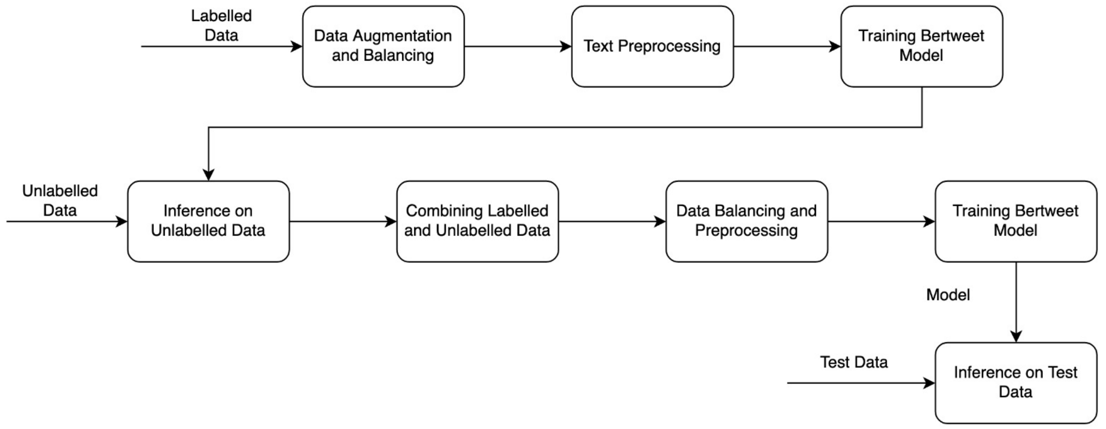
Labelled Data
Data Augmentation and Balancing
Text Preprocessing
Training Bertweet Model
Unlabelled Data
Inference on Unlabelled Data
Combining Labelled and Unlabelled Data
Data Balancing and Preprocessing
Training Bertweet Model
Model
Test Data
Inference on Test Data

Figure 1: Flowchart of the methodology using BERTweet

Reddit dry eye disease posts, and 12,482 Reddit posts about social anxiety among individuals aged 13-25.

3 Methodology using LLM

This code utilizes a pre-trained LLM called "mistralai/Mistral-7B-Instruct-v0.2" (Jiang et al., 2023)from the Hugging Face library to classify social media posts based on whether the user's age can be inferred from the text content. The main steps are:

3.1 Preprocessing

Initially, preprocess the unlabeled_testing data by applying the *normalizeTweet()* function to each entry in the "text" column. This step ensured uniformity and prepared the text content for subsequent analysis.

3.2 Defining the Instructions for the LLM

The LLM instructions were defined through a string variable named "prompt." This string contains the instructions for the LLM. It explains the task (predicting age from social media posts) and provides examples of positive and negative cases for age reveal.

3.3 Building the Prediction Function

Define a function named y_pred.It creates a conversation-like structure with three parts: <User role: Provides the prompt explaining the task, Assistant role: Acknowledges understanding of the task. User role again: Provides the actual social media post enclosed in square brackets ([start] and [end]).> Apply the tokenizer to the conversation structure to convert it into a format suitable for the LLM (numerical representations). Generating a response from the LLM using the encoded conversation: max_new_tokens=1000 limits the generated response length. do_sample=True enables random sampling for potentially diverse outputs. Decodes the generated tokens back into human-readable text. Extracts the predicted label ("positive" or "negative") from the generated text, likely by splitting by "[/INST]." Use a try-except block to handle potential errors. Loop through each text entry (t) in the "text" column inside the try block. Append the predicted label from the LLM to a list y_pred.Inside the except block: If an error occurs, print the error message and "error occurred."

3.4 Creating the Output DataFrame

Create a DataFrame test with a " label " column containing the predicted labels from y_pred. Saves the DataFrame test to a CSV file named "output-test.csv."

4 Methodology using BERTweet

For the second attempt, a BERTweetbase (Nguyen et al., 2020) implemented using the Huggingface toolkit(Wolf et al., 2019) was exploited to extract the exact age from social media posts. The flow chart of the methodology using BERTweet is presented in Figure 1. The main steps of this approach are:

4.1 Balance the given labeled training dataset

The given labeled training data named "labeled_training.csv" comprising 8800 samples is imbalanced with 5965 for 0 labels and 2834 for 1 label. The following sequence of operations is applied to balance this data.

Model Used	Dataset	F1-score	Precision	Recall
Mistral-7B-Instruct-v0.2 (LLM)	validation data	0.725	0.773	0.835
	test data	0.793	0.716	0.889
BERTweet	Validation data	0.880	0.920	0.850
	test data	0.900	0.916	0.884

Table 1: Performance of our models on the validation and test sets for Task 6

4.1.1 Data augmentation

The *augmentation()* function is defined to perform data augmentation. In this case, it uses back-translation augmentation from English to German and back to English using the *naw.back_translation.BackTranslationAug* module from nlpaug. The code performs data augmentation using back translation to increase the diversity of the labeled training data, especially for the positive class (label = 1). this augmented text data is saved to a CSV file named augmented_text.csv

4.1.2 Concatenating Data

The two datasets, i.e., from CSV files named "labeled_training.csv" and augmented_text.csv, are concatenated together. This concatenates the original labeled training dataset and the augmented text dataset vertically (stacking them on top of each other) to create a single concatenated dataset.

4.1.3 Undersampling

Undersample the concatenated dataset using *RandomUnderSampler()* from the *learn* library to balance the class distribution by randomly removing instances from the majority class (label = 0) until both classes have an equal number of samples. The sampling strategy ensures that both classes have 5000 samples each. The balanced dataset is saved to a CSV file named balanced_data.csv.
Preprocess balanced labeled training dataset and labeled validation dataset

4.1.4 Tweet Normalization

The next step first preprocess the text data in both the training and validation sets (i.e., CSV files, 'balanced_data.csv' and 'labeled_validation.csv') using the *normalizeTweet()* function from the TweetNormalizer module. This function cleans up and standardizes the text data by handling tasks such as lowercasing, URL removal, punctuation removal, and other text normalization tasks specific to social media text.

4.1.5 Tokenization

After Tweet Normalization, the text data in both the training and validation sets is tokenized using the BERTweet tokenizer (AutoTokenizer.from_pretrained (checkpoint)). This tokenizer is specifically designed for tweet text and handles the tokenization of tweets, including special characters, hashtags, and mentions.

4.1.6 Preprocess unlabelled dataset

Preprocess unlabeled dataset using Tweet Normalization and tokenization as done previously.

4.1.7 Model Training

The model was trained using the balanced labeled training and validation datasets.
The trained model is applied to the tokenized unlabeled dataset for labeling it. The result is stored in the csv file 'unlabeled_labeled.csv'

4.1.8 Random undersampling

Random undersampling is performed on the CSV file 'unlabeled_labeled.csv' dataset to balance using RandomUnderSampler from imblearn.

4.1.9 Concatenating the data

The next step involves the concatenation of data of original "labeled_training.csv" and 'unlabeled_labeled.csv' and is saved as final_train.csv

4.1.10 Preprocessing

Preprocess this final training dataset('final_train.csv') and the validation dataset (labeled_validation.csv).using Tweet Normalization and tokenization as done previously

4.1.11 Train model and Model Evaluation

The BERTweet model is loaded, and the tokenizer is initialized. The model is trained on the training dataset and is used to make predictions on the validation dataset. The predictions are then evaluated using classification_report from sklearn.metrics.The trained model is evaluated on the validation dataset to assess its performance.

5 System description

For the first methodology, the pre-trained LLM ’Australia/ Mistral-7B-Instruct-v0.2’ was configured to reduce memory usage (potentially using 4-bit weights) by setting load_in_4bit=True.". The Language Model (LLM) generates a response from the encoded conversation, with the maximum length of the response limited to 1000 tokens by setting max_new_tokens=1000. Additionally, enabling do_sample=True allows for random sampling, potentially yielding diverse outputs.

For the second strategy, the textual data was pre-processed using the *normalizeTweet()* function from the TweetNormalizer module. Following Tweet Normalization, text data in both the training and validation sets undergo tokenization using the BERTweet tokenizer (AutoTokenizer.from_pretrained(checkpoint)) with a maximum sequence length of 512 tokens. We optimized the model using the BERTweet model’s training parameters as *Experiment name:* between-test, *learning rate*: 2e-5, *number of training epochs*: 5, *Checkpoint saving strategy*: No checkpoints saved, and *batch size per device as* 64. The experiment used Nvidia A100 GPU with Python as the programming language.

6 Results

The performance of the final trained models was assessed using the Task 6 validation set prior to evaluating and submitting the prediction file on the test set. The performance of the models was evaluated using the F1-score for the positive class (i.e., posts annotated as "1"). The results of the validation set for Mistral-7B-Instruct-v0.2 (LLM) and BERTweet–base are reported in Table 1.As indicated in Table 1, the top-performing model on the Task 6 validation set is the BERTweet-base model, which achieved an F1-score of 0.880. BERTweet-base model showed better results in predicting positive class(1) than Mistral-7B-Instruct-v0.2 (LLM). When evaluating the performance of the models on the test set, the BERTweet–base model achieved an F1-score of 0.90 on the test set, with precision and recall values of 0.916 and 0.884, as seen in Table 1. These findings highlight the superior performance of the BERTweet-base model in accurately predicting the positive class (label "1") compared to Mistral-7B-Instruct-v0.2 (LLM) on both the validation and test datasets for Task 6.

7 Conclusion

This work uses Mistral-7B-Instruct and BERTweet models for precise age extraction from social media posts. The limitations of traditional approaches were overcome by directly inferring user age from text. Evaluation of validation and test sets reveals significant performance disparities between the two models. BERTweetbase outperforms Mistral-7B-Instruct with an F1-score of 0.90 for the positive class on the test set, showcasing its superior efficacy in age prediction. These results underscore the efficacy of advanced language models in enhancing demographic analysis and research applications. In future work, refining prompt engineering techniques and utilizing advanced models such as Llama 3 can be exploited to enhance performance. Additionally, exploring ensembling methods with models like Bernie can offer even greater accuracy and robustness in demographic predictions.

References

Robert Chew, Caroline Kery, Laura Baum, Thomas Bukowski, Annice Kim, Mario Navarro, et al. 2021. Predicting age groups of reddit users based on posting behavior and metadata: classification model development and validation. *JMIR Public Health and Surveillance*, 7(3):e25807.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ari Z Klein, Arjun Magge, and Graciela Gonzalez-Hernandez. 2022. Reportage: Automatically extracting the exact age of twitter users based on self-reports in tweets. *PloS one*, 17(1):e0262087.

Antonio A Morgan-Lopez, Annice E Kim, Robert F Chew, and Paul Ruddle. 2017. Predicting age groups of twitter users based on language and metadata features. *PloS one*, 12(8):e0183537.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Reyhaneh Sadeghi, Ahmad Akbari, and Mohammad Mehdi Jaziriyan. 2024. Exaauac: Arabic twitter user age prediction corpus based on language and metadata features.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.