Small Language Models are Diligent Learners: Evaluation of Embedding- and Transformer-based Language Models for Text Classification and Recommendation

Dimitrios A. Koutsomitropoulos, *member IEEE*
Computer Engineering and Informatics Dpt.
School of Engineering, University of Patras
Patras, Greece
koutsomi@ceid.upatras.gr

Savvas Skoulidis
Computer Engineering and Informatics Dpt.
School of Engineering, University of Patras
Patras, Greece
skoulidis@ceid.upatras.gr

2025 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) | 979-8-3315-7024-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/INISTA68122.2025.11249552

***Abstract*— Size and requirements proliferation of LLMs restrain their proprietary usage and often present an overkill for natural language tasks such as text classification and recommendation. In this paper we review and evaluate a series of smaller yet diligent models on two datasets corresponding to single-label multi-class classification problems. For once, we confirm the suitability and advantage of transformer architectures over earlier approaches, even with their base variants. In addition, we show that it is not necessary to employ even larger generative models for such tasks, as their performance improvement does not go on par with their prohibiting costs for everyday users.**

Keywords—LLM, SLM, NLP, NLU, LLaMA, GPT, BERT

I. INTRODUCTION

The recent advancements and sheer volume of Large Language Models (LLMs) have revolutionized the way people perceive the entire spectrum of AI applications into everyday life. A long path has been driven in NLP since the advent of vector embeddings and transformer architectures leading to even larger and more powerful models that match or even surpass human perception in several tasks [1, 2]. Still, such models are often difficult to train and manage for proprietary purposes either because they are "closed source", are behind paywalls or simply are beyond the capabilities of standard commodity hardware.

While smaller models and simpler architectures like BERT and DistilGPT-2 are frequently deemed obsolete, their wide availability and lower impact appear as an advantage that renders them attractive for everyday tasks. These models are still capable of learning effectively and can perform competitively especially in cases where one needs to avoid overshooting a language processing problem with vast amounts of trained parameters and datasets; rather, they can focus on small, specific knowledge domains and possibly pay off in fine-grained tasks.

In this paper we review and evaluate a series of pretrained, trained and finetuned language models, each focusing on different downstream tasks such as embedding creation, classification, and text generation. We argue that small- and medium-sized language models are capable and adequate to support text classification and recommendation tasks for specific knowledge domains and corresponding datasets. We show that such models can remain cost-effective, when compared to, for example, widely popular generative large language models which may be difficult to manage and impractical to train from scratch.

We try the models on two different tasks: One involving classification within a wines production domain; and another about sentiment analysis of social feeds related to the COVID-19 pandemic. Finally, our efforts culminate in comparing these with an indicative large generative language model, namely LLaMA-2 and reflect on their performance and applicability. Source code and experiments are openly available at: https://github.com/savskoul/thesis-

The rest of this paper is organized as follows: In the next section we introduce the background for language models, include related work on text-based classification and recommendation and review the models tested further on. Section 3 discusses our evaluation methodology and describes the datasets and tasks evaluated. Section 4 presents the evaluation results and compares the various models based on their performance on the corresponding tasks. Lastly, section 5 summarizes our conclusions and future work.

II. BACKGROUND AND RELATED WORK

Learning algorithms have been successfully applied in the past for NLP-based classification and recommendation. Indicative examples include sentiment analysis of product reviews, social posts [3], news feeds classification, product recommendation [4], fraud detection [5] etc.

Traditional NLP approaches had the problem of capturing the semantic information contained in the texts. Recognizing this weakness has led to the development of more advanced word representation techniques. These methods are based on the principle that words appearing in similar texts have a similar meaning [6]. Thus, words are encoded as vectors within a multidimensional space, where those with similar meanings are close to each other, allowing algorithms to understand their semantic relationship.

Word2vec is an algorithm that relies on neural networks to understand the correlations between words [7]. Its main goal is to convert words into numerical vectors, which retain semantic and syntactic information. In this way, the model can recognize words with similar meanings or predict possible words in a given context. The effectiveness of word2vec lies in its ability to capture complex linguistic relationships through large datasets.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Authorized licensed use limited to: University of Patras. Downloaded on December 30,2025 at 16:37:27 UTC from IEEE Xplore. Restrictions apply.

ELMo (Embeddings from Language Models) is a model developed to create dynamic vector representations of words [8]. Its main goal is to turn words into contextual dependent vectors, taking into account the syntax and semantics of each word depending on the context in which it appears. ELMo's innovation lies in its ability to capture richer linguistic information, as it does not use static embeddings like Word2Vec, but learns dynamic word representations.

BERT (Bidirectional Encoder Representations from Transformers) is a model that is based on the transformer architecture [9]. Its main goal is to create vector representations of words that depend on their context, allowing the model to understand the meaning of words depending on the environment in which they appear. The effectiveness of BERT lies in its two-way training, since, unlike previous language models that only read the text from left to right or vice versa, BERT takes into account the entire context of the sentence at the same time. In this way, it can capture deeper semantic relationships and dependencies between words.

XLNet is a language model developed by Yang et al. [10]. Unlike BERT, which masks specific words and predicts them independently, XLNet learns to predict words based on all possible sequences of words in a sentence. This randomly rearranged training framework allows the model to learn all possible dependencies between words, thereby improving natural language comprehension and reducing dependence on specific positions of words in the sentence.

GPT (Generative Pre-trained Transformer) is a language model developed by Radford et al. [11]. The effectiveness of GPT lies in its ability to learn linguistic patterns through large datasets, without supervision, and then generalize to different natural language processing tasks. GPT has evolved through various iterations (GPT-2, GPT-3, GPT-4), each of which has increased capacity and greater understanding of the language.

LLaMA is a family of large language models released by Meta and ranging from 7B to 65B parameters [12]. These models are focused on efficient inference by training a smaller model on more tokens. The Llama model is based on the GPT architecture, but it uses pre-normalization and certain optimizations to improve performance and better handling of longer sequence lengths. LLaMA-2 follows mostly the same architecture as the oringal LLaMA but is pretrained on more tokens and integrates reinforcement learning with human feedback (RLHF) on the fine-tuned model for chat purposes.

III. METHODOLOGY

A. Tasks and Datasets

We evaluate our models in two tasks: *a) wine recommendation* where, given wine reviews and descriptions, the model predicts the appropriate wine variety and *b) sentiment analysis* on tweets regarding the COVID-19 pandemic.

For the problem of wine recommendation, the public dataset Wine Reviews by Kaggle was used [13]. The initial set initially comprised around 130,000 records, with detailed wine descriptions and information such as wine variety, wine description, country of origin, price and rating. Class descriptions are highly unbalanced with some varieties having many more descriptions than others. Therefore, the top 10 most frequent wine varieties were selected, with a balanced sampling of 1,000 entries per variety to create a balanced subset. Post-processing, the average length of descriptions was approximately 40 words, involving removal of special characters, stopwords, and digits. Figure 1 shows a sample of the original dataset.

Wine Variety	Description
Merlot	A blocky, toasty, chewy wine with a flavor mix of herb and earth and berries. The tannins are rough and leave a slightly bitter impression.
Chardonnay	Reserved aromas of lemon zest, shaved butter and wet stone greet the nose on this ready-to-drink bottling. The palate shows poached apple and cooked pear tones, with a slightly sour tang keeping it lively.
Cabernet Sauvignon	A good everyday Cabernet with some aspirations to quality. It's a little rugged and harsh in tannins, but dry, with oak-inspired flavors of blackberries, black currants, cola and herbs. Drink now.
Bordeaux-style Red Blend	Jammy, overripe fruit does not help this wine. Its soft tannins and acidity provide some pleasure, but that clumsy fruit gets in the way.
Pinot Noir	This is a medium-bodied, somewhat chunky Pinot Noir, combining plummy aromas with distinctively Pinot-like sous-bois character. The flavors turn tart and savory, ending on a pomegranate note.

Fig. 1. A sample of the processed wine variety dataset.

For sentiment analysis, the public dataset Coronavirus tweets NLP - Text Classification by Kaggle was used [14]. The initial set contained about 45,000 Tweets along with their source, the date they were created, and their classification as Extremely Negative, Negative, Neutral, Positive, and Extremely Positive. The dataset was cleaned and preprocessed by removing URLs, hashtags, stopwords etc, discarding tweets with fewer than 5 words, and collating sentiment categories into 3: 0 – Negative & Extremely Negative, 1 – Neutral, 2 – Positive & Extremely Positive. Each category was represented by randomly selecting 3,000 tweets, averaging 15 words per tweet after extensive preprocessing, including URL, hashtag, stopwords removal, and discarding tweets shorter than five words. A sample of the dataset is shown in Figure 2.

Sentiment	Clean Tweet
0	Food banks shift their distribution model in response to COVID-19 pandemic and skyrocketing demand
1	Even Freddy Krueger has to go to the grocery store sometimes #COVID19 #CoronavirusUSA
1	Coronavirus hangs around even after symptoms subside #COVID2019
0	There's psychology behind the foods we don't buy in a crisis #COVID19
2	You're doing a great job, thank you — next step: following D.C. in a rent freeze
2	Thank you to our truckers and grocery store workers for all they are doing to ensure essential supplies for Californians during this very challenging time

Fig. 2. A sample of the processed COVID-19 tweets dataset.

B. Models Configuration and Metrics

1) Word2Vec

For the evaluation of Word2Vec, the following procedure was followed. Initially, the text of the wine descriptions underwent a tokenization process, removing the stopwords using the NLTK library. The result was a list of tokens for each description, which was then used to train the Word2Vec model. Word2Vec was trained on the set of tokenized descriptions, with embedding dimensions equal to 100, a word window of 5, and a minimum word occurrence of 2. After training the model, a text representation was created for each sample through the average of its word vectors, thus creating a fixed size embedding vector per observation.

Authorized licensed use limited to: University of Patras. Downloaded on December 30,2025 at 16:37:27 UTC from IEEE Xplore. Restrictions apply.

The dataset was then divided into a training and test set with an 80-20 ratio and maintaining the ratio of categories. For the classification, a Logistic Regression model with balanced class weights was used. The model's performance was evaluated using accuracy, recall, and F1-score metrics for each of the ten wine variety categories. A similar procedure was followed for sentiment analysis.

2) BERT

The pre-trained BERT model was then applied to the problem of classification of wine varieties. We used the *bert-base-uncased* variant with 110M parameters. The dataset was split into training and test sets with a ratio of 80%-20%, maintaining the distribution of categories. The data was converted into Dataset objects of the Hugging Face library, and tokenization was applied with the BERT tokenizer, with padding and truncation so that the texts had a uniform length, up to a maximum of 512 tokens.

The BERT model was configured for a downstream classification task with 10 outputs and then trained with a small number of epochs to investigate the capability of the pretrained model to finetune and adapt to the problem data. Training hyperparameters were defined, such as learning rate 2e-5 and batch size 8. To avoid overfitting, early stopping was applied with 2 epochs patience and a threshold of improvement of 0.001. Throughout training, an evaluation of the model was carried out at the end of each epoch, using the F1-score as a key metric for selecting the best model. The final test of the model was carried out on the test set, using metrics such as accuracy, F1-score and recall, in order to assess the model's ability to correctly classify the descriptions of the wines as well as the sentiment categories.

3) ELMo

To evaluate ELMo, the pre-trained ELMo model was used through the TensorFlow Hub, with 93M parameters. Initially, the wine descriptions were converted into fixed-length vectors of 1024 dimensions via the ELMo embedding layer. ELMo has the advantage of producing contextualized word embeddings, taking into account the context of each word within the text, unlike models like Word2Vec that produce static embeddings. The classification problem was then addressed using logistic regression, as in previous experiments, to compare the results.

4) GPT

For evaluating GPT architectures we opted for the small language model variant DistilGPT-2 with 82M parameters vs 124M of the original GPT-2. DistilGPT-2 is a lighter version of the well-known GPT-2, offering faster training and reduced computational resource requirements while retaining a significant portion of the performance of the full model.

Tokenization was carried out with DistilGPT-2's tokenizer, with padding and truncation so that all texts have a uniform length, up to 128 tokens, taking into account the relatively short length of wine descriptions. As GPT models do not include a predefined pad token, the model's end-of-sequence token was used for padding. The model's training was set with a learning rate of 2e-5, batch size 8, and weight decay of 0.01. Early stopping was also used with two epochs patience, to avoid overfitting. The number of finetuning epochs was set to 10, and the best model was based on the macro F1-score.

5) XLNet

The pre-trained model XLNet (*xlnet-base-cased*), with 110M parameters, was applied. Initially, the wine varieties were numerically encoded using the Label Encoding technique for compatibility with the model. The dataset was then separated into a training and test set, maintaining the balance between the categories via stratified split. The tokenization process followed with XLNet's tokenizer, applying padding and truncation so that all texts have a uniform length of up to 512 tokens, taking full advantage of the model's potential on large sequences.

For finetuning the model, the following hyperparameters were used: learning rate 2e-5, batch size 8, weight decay 0.01 and number of epochs 3, as well as early stopping with 2 epochs patience to avoid overfitting. Fewer epochs were chosen for finetuning because the model required a lot of training time that could not be allocated due to resource limitations. The best model was chosen based on the macro F1-score.

6) LLaMA-2

We opted for the smaller pretrained 7B variant of LLaMA-2 (*Llama-2-7b-chat-hf*) from the Hugging Face library. We applied the QLoRA finetuning approach with 4bit quantization for efficient memory usage (8x less) along with gradient checkpointing. We finetuned the model on both datasets using the quantized weights and a small number of epochs (up to 5), with learning rate 1e-4 and early stopping with 2 epochs patience. Due to the model's high resource requirements as compared to the previous models, we used the high tier of Google Colab offering an A100 GPU with 84GB of RAM.

IV. Results and Discussion

All experiments, with the exception of LLaMA, were conducted on the Google Colab environment using a T4 GPU with 16GB of RAM. For the metrics reported below, since both datasets are balanced and the tasks are in fact *single-label* multi-class classification problems, the average F1-score equals accuracy, as expected. Average computational times per epoch were around 30 minutes for BERT and XLNet, while LLaMA-2 required approximately one hour per epoch.

A. Wine Recommendations

Table I summarizes evaluation results across all models for the wine variety task.

TABLE I. Models Evaluation for the Wine Dataset

Model	Accuracy	Macro F1-score
Word2Vec + Logistic Regression	47%	0.47
ELMo + Logistic Regression	49%	0.49
BERT (bert-base-uncased)	**69%**	**0.69**
DistilGPT-2	64%	0.64
XLNet	67%	0.67
LLaMA-2	**75%**	**0.75**

Word2Vec is useful as a baseline model and, along with logistic regression, is capable of predicting the appropriate variety 5x times better than a random selection (10%). BERT improves significantly over Word2Vec exhibiting 69% accuracy and excelling in wine varieties where Word2Vec performed poorly thus lowering its overall score. ELMo improved slightly over Word2Vec but remained below BERT. As in Word2Vec, some varieties like Riesling (F1-score 0.65)

Authorized licensed use limited to: University of Patras. Downloaded on December 30,2025 at 16:37:27 UTC from IEEE Xplore. Restrictions apply.

and Rosé (F1-score 0.62) perform better than others, confirming that such varieties have more characteristic descriptions that facilitate their identification by the model. The improvement over Word2Vec confirms that the utilization of context-aware embeddings offers advantages over traditional static approaches.

Overall, the DistilGPT-2 model achieved better performance than the Word2Vec and ELMo models but lagged behind BERT. This result is to be expected, given that DistilGPT-2, although a transformer-based model, is lighter and smaller than BERT, with a smaller number of parameters and less variety in the patterns it can learn. However, the results confirm the usefulness of GPT-type language models in text classification.

The application of the XLNet model to the problem of classification of wine varieties led to an accuracy of 67% that approached that of BERT. XLNet exhibits high scores in certain varieties as before, indicating its capability to capture the fine-grained descriptions and their characteristic taste and flavor aspects. LLaMA has also shown higher values for characteristic varieties, while others like Merlot and Chardonnay had lower, possibly due to similar descriptions and overlap of features. Figure 3 shows the learning curve of the models for the wine classification task.

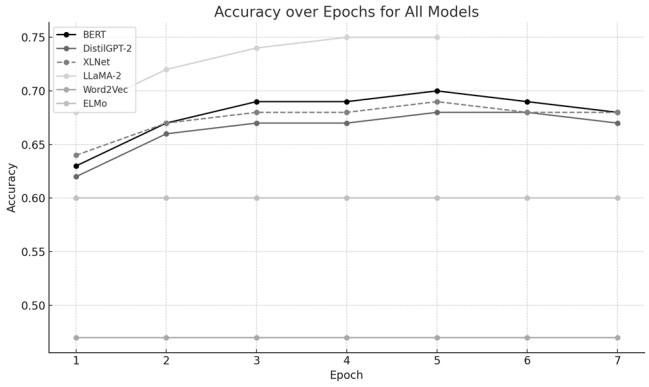

Fig. 3. Accuracy over epochs for the wine variety dataset.

Figure 3 compares the accuracy of the six models, over seven training epochs. All four transformer-based models surge in performance during the first two epochs, then either plateau or improve slightly, with LLaMA-2 ultimately topping out around 0.75. By contrast, ELMo remains locked at roughly 0.60 and Word2Vec at about 0.47 for every epoch. Those horizontal lines reflect the fact that both Word2Vec and ELMo are deployed as fixed feature extractors rather than fully fine-tuned models, so as long as their output embeddings are computed once, only a lightweight classifier on top is trained. Once that classifier converges during the initial pass, there's no further adaptation—so accuracy stays exactly the same.

B. Sentiment Analysis

Table II summarizes evaluation results across all models for the sentiment analysis task.

TABLE II. MODELS EVALUATION FOR THE TWEETS DATASET

Model	Accuracy	Macro F1-score
Word2Vec + Logistic Regression	47%	0.47
ELMo + Logistic Regression	60%	0.6
BERT (bert-base-uncased)	**71%**	**0.71**
DistilGPT-2	66%	0.65
XLNet	67%	0.67
LLaMA-2	**76%**	**0.76**

For this task, Word2Vec remains a baseline for improvement comparison. In itself, the model exhibited difficulty in identifying tweets with positive sentiment. As expected, BERT shows considerable improvement, excelling especially in generally negative and neutral tweets (F1-score of 0.74 and 0.71 respectively). ELMo on the other hand performed lower, but had uniform performance across all sentiment categories, thus showing adequate generalization and capability of cross-identifying different sentiments.

This was also demonstrated by DistilGPT-2, with an overall improved accuracy of 66%. XLNet achieved 67% sentiment recognition. The result demonstrates that XLNet successfully manages the variety of expressions and the short length of tweets, leveraging its potential in understanding the complex linguistic structure and sequence of words. LLaMA-2 has also shown balanced scores in all sentiment categories, thus being relatively unbiased to certain sentiments. Figure 4 shows the learning curve of the models for the sentiment analysis task.

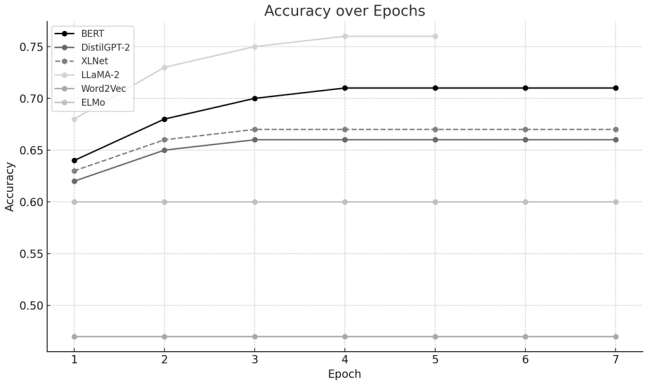

Fig. 4. Accuracy over epochs for the wine variety dataset.

Figure 4 compares the accuracy of the six models over seven training epochs for the sentiment analysis task. The four transformer-based models—BERT, DistilGPT-2, XLNet, and LLaMA-2—show rapid performance gains within the first two epochs, after which they either plateau or exhibit marginal improvements. LLaMA-2 ultimately reaches the highest accuracy, stabilizing around 0.76. In contrast, ELMo consistently maintains accuracy at approximately 0.60, while Word2Vec remains steady around 0.47 across all epochs. These flat performance curves for ELMo and Word2Vec highlight their use as fixed feature extractors, implying that once their embeddings are generated, the subsequent training involves only a lightweight classifier.

C. Discussion

It becomes evident that modern language models like BERT and XLNet clearly outperform traditional Word2Vec and ELMo techniques in both datasets, both in the problem of wine recommendation and in the sentiment analysis of tweets. BERT achieved overall the 2nd-best performance in the tweet problem (71% macro F1-score), demonstrating the power of bidirectional transformers in handling short, informal texts. XLNet performed highly competitively in both datasets, with slightly better results in the wine classification problem than GPT and balanced performance in the sentiment analysis of tweets.

Authorized licensed use limited to: University of Patras. Downloaded on December 30,2025 at 16:37:27 UTC from IEEE Xplore. Restrictions apply.

DistilGPT-2, although a lighter model, has achieved satisfactory results, making it an efficient choice when there are limitations of computing resources. ELMo, as an intermediate approach, clearly outperformed Word2Vec and proved to be particularly effective in the problem of tweets, thanks to its ability to take into account the context of words. Finally, Word2Vec, as expected, was the lowest-performing baseline model, providing the benchmark for comparing advanced models.

Due to its size, LLaMA-2 requires significant resources, even with compression techniques such as QLoRA. Although it offers top-notch accuracy, the cost of computing resources is higher compared to smaller models such as BERT or DistilGPT-2, making it suitable only for applications where maximum performance is critical.

V. Conclusions and Future Work

Besides the dominance of large generative language models, results highlight the effectiveness of modern transformer-based architectures, such as BERT and XLNet, which achieved the second-highest F1-score and accuracy values in both problems. In contrast, traditional embeddings, such as Word2Vec, served as a useful baseline, reinforcing the substantial gap in performance between static and contextual language representations. Notably, the results remained consistent across both datasets, indicating that these models are capable of handling both complex descriptive texts and shorter, informal messages, such as tweets. Even though LLaMA-2 performs best among all models, its higher scores are not justified by its increased size: It exceeds BERT by almost two orders of magnitude but improves only by 5% on accuracy. Given the forbidding computational costs of such models for commodity users, small and medium language models still have an apt room for applicability with a more favorable balance between accuracy, resource consumption, and deployment feasibility.

Additional experiments are in order to assess the suitability of smaller models for certain tasks. We intend to further investigate the balance between performance, cost, model size and dataset characteristics by involving more model variants and designing an end-to-end application targeted for everyday use. A task-oriented chatbot capable of recognizing user intent to provide support and guidance with their queries is a promising research goal as our next step.

References

[1] Wang, Alex, et al. Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems, 2019, 32.

[2] White, Colin, et al. Livebench: A challenging, contamination-free llm benchmark. In Proc. of the 13th Int. Conference on Learning Representations (ICLR 2025), to appear.

[3] Pak, A., & Paroubek, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. International Conference on Language Resources and Evaluation, 2010.

[4] Gomez-Uribe, C., & Hunt, N. The Netflix Recommender System. ACM Transactions on Management Information Systems (TMIS), 6, 1 – 19, 2015.

[5] Chandola, V., Banerjee, A., & Kumar, V. Anomaly detection: A survey. ACM Comput. Surv., 41, 15:1-15:58, 2009.

[6] Ferrone, L., & Zanzotto, F.M. Symbolic, Distributed, and Distributional Representations for Natural Language Processing in the Era of Deep Learning: A Survey. Frontiers in Robotics and AI, 6, 2017.

[7] Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. Efficient Estimation of Word Representations in Vector Space. International Conference on Learning Representations, 2013.

[8] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. ArXiv, abs/1802.05365.

[9] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics.

[10] 44. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., & Le, Q.V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. Neural Information Processing Systems.

[11] Radford, A., & Narasimhan, K. Improving Language Understanding by Generative Pre-Training. Open-AI Blog, 2018.

[12] Touvron, Hugo, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

[13] Zynicide. Wine Reviews [Data set]. Kaggle. https://www.kaggle.com/datasets/zynicide/wine-reviews

[14] Datatattle. COVID-19 NLP Text Classification [Data set]. Kaggle. https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification

Authorized licensed use limited to: University of Patras. Downloaded on December 30,2025 at 16:37:27 UTC from IEEE Xplore. Restrictions apply.