When XGBoost Outperforms GPT-4 on Text Classification: A Case Study

Matyas Bohacek
Stanford University
maty@stanford.edu

Michal Bravansky
University College London
michal@bravansky.com

Abstract

Large language models (LLMs) are increasingly used for applications beyond text generation, ranging from text summarization to instruction following. One popular example of exploiting LLMs' zero- and few-shot capabilities is the task of text classification. This short paper compares two popular LLM-based classification pipelines (GPT-4 and LLAMA 2) to a popular pre-LLM-era classification pipeline on the task of news trustworthiness classification, focusing on performance, training, and deployment requirements. We find that, in this case, the pre-LLM-era ensemble pipeline outperforms the two popular LLM pipelines while being orders of magnitude smaller in parameter size.

1 Introduction

Over the past year, large language models (LLMs) have become exceedingly popular with the public. LLM-powered chatbots such as ChatGPT[1] have made LLM use intuitive even for non-technical audiences, which have found creative ways of integrating them into day-to-day tasks (Chan et al., 2023), school work (Kasneci et al., 2023), creative practice (Parra Pennefather, 2023), and more. For many, LLMs have become synonymous with artificial intelligence (Liao and Vaughan, 2023).

One of the many reasons for why the public took notice of LLMs are their emergent capabilities beyond sentence completion (e.g., translation, problem solving, and instruction following) (Wei et al., 2022a; Valmeekam et al., 2023), allowing for many down-stream applications. The abundance of emergent capabilities has also been recognized in the technical communities. In the research domain, LLMs are now being used for code generation (Zhou et al., 2023; Lomshakov et al., 2023), medicine research (Thirunavukarasu et al., 2023), and drug discovery (Chakraborty et al., 2023). Similarly, many industry solutions that analyze text data now rely on LLM architectures (McElheran et al., 2023).

There are clear benefits of using LLMs beyond the scope of text generation – specifically for classification, tagging, or content detection. For once, LLMs can be used in a few- or a zero-shot fashion, which minimizes or even eliminates the need for training data. Moreover, LLMs have become increasingly accessible and customizable using cloud-based inference and fine-tuning solutions.

On the other hand, the fast adoption of LLMs has, in many ways, exceeded our understanding of their risks and limitations. Initial exploratory work has identified gaps in the robustness of LLMs across diverse tasks and languages (Ahuja et al., 2023; Bang et al., 2023) and patterns of gender, racial, and political biases (Dong et al., 2023; Motoki et al., 2023; Khandelwal et al., 2023). Moreover, LLMs are prone to hallucination: a state in which they construct factually or logically incorrect narratives, possibly leading to user deception (Wang et al., 2023; Zhang et al., 2023; McKenna et al., 2023; Rawte et al., 2023).

In this short paper, we present a case study comparing two LLMs to a pre-LLM-era classification pipeline on the task of news trustworthiness analysis (using the Verifee dataset (Boháček et al., 2023)). We focus on each method's performance, training, and deployment requirements. This comparison is limited and, on its own, cannot be used to draw broader conclusions about the comparable performance of the examined methods. Nonetheless, it presents a template for easy evaluation of LLMs' performance compared to previous methods, reflecting aspects beyond pure accuracy. Overall, we believe that this paper can encourage more work evaluating LLMs in comparison to earlier methods, effectively expanding our understanding of the benefits and shortcomings of LLMs.

[1] https://openai.com/chatgpt

Proceedings of the 4th Workshop on Trustworthy NLP (TrustNLP 2024), pages 51–60
June 21, 2024 ©2024 Association for Computational Linguistics

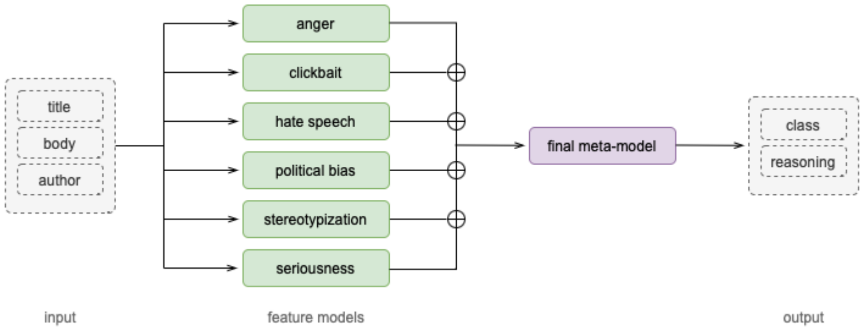

Figure 1: Overview of the modular ensemble pipeline data flow. At the input, a news article is analyzed using each feature model, yielding a feature embedding that is then inserted into the final meta-model. This model outputs a class prediction, along with its reasoning as a list of found features.

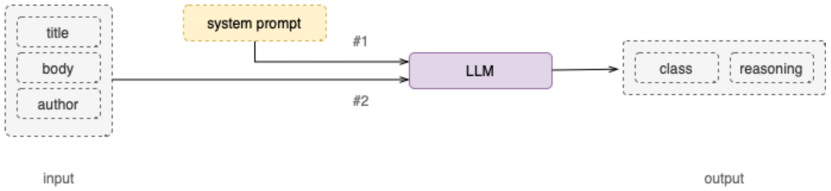

Figure 2: Overview of the LLM pipeline data flow. The LLM is first presented with the system prompt. At the input to the pipeline, a news article is structured as a single body of text and inserted into the LLM. The model first outputs the detected features of the article (i.e., the reasoning) and then proceeds to the final classification.

2 Related Works

In this section, we briefly review the existing work about the pre-LLM-era classification pipelines, LLMs, and comparative studies of the two.

2.1 Pre-LLM-Era Classification Pipelines

Over the past few years, text classification methods have mostly transitioned from hand-crafted features to deep learning architectures (Gasparetto et al., 2022) such as Electra (Clark et al., 2019), which was the state-of-the-art pre-trained language model on the GLUE benchmark (Wang et al., 2018) before the advent of LLMs. The literature has explored classification in various contexts, finding that achieving the best results requires specific architecture and data adjustments (Riduan et al., 2021; Wang et al., 2021), as there is no universal architecture for complex text classification tasks.

That said, let us consider a niche classification subtopic as an illustration of overarching trends, specifically IT ticket classification (Liu, 2023; Zicari et al., 2022): categorizing user inquiries based on rigid rules and a knowledge base. Recent work (Revina et al., 2021) has found that the best results for this task are obtained through extracting individual features and then utilizing a meta-model for final prediction. We refer to this pipeline approach as the modular ensemble pipeline.

2.2 LLM Classification Pipelines

Recent year has seen a boom of new LLM architectures and models (Zhao et al., 2023; Wan et al., 2023) – some of the most popular ones include GPT-4 (OpenAI, 2023), LLAMA 2 70B (Touvron et al., 2023), Claude 2 (Anthropic), and Mistral 7B (Jiang et al., 2023). Originally, LLMs were exploited to generate synthetic data and expand training datasets for conventional classification architectures (Kumar et al., 2020; Li et al., 2023; Golde et al., 2023; Chung et al., 2023). Recently, this approach was replaced by direct LLM inference for classification (Loukas et al., 2023; Chen et al., 2023; Frick, 2023; Sun et al., 2023).

2.3 Comparative Studies

Existing comparative studies (Qin et al., 2023; Laskar et al., 2023; Zhong et al., 2023; Wu et al., 2023) evaluate LLMs on conventional NLP tasks (e.g., summarization and question answering). They find that LLMs perform on par with pre-LLM benchmarks on some tasks but mostly score below the state-of-the-art results. However, these studies lack insight into the training and inference considerations of these approaches.

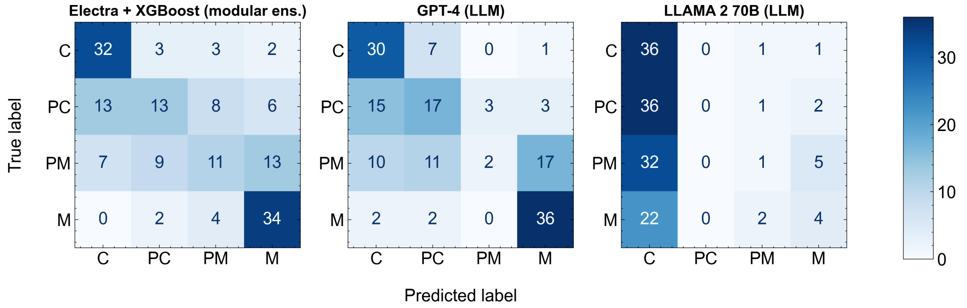

Figure 3: Confusion matrices of the Electra + XGBoost (*modular ensemble*), GPT-4 (*LLM*), and LLAMA 2 70B (*LLM*) pipelines on the testing set of the Verifee dataset. C, PC, PM, and M correspond to credible, partially credible, partially manipulative, and manipulative classes, respectively. Note that for the LLM pipelines, only the best-performing model configuration is shown.

3 Data

We use the Verifee news trustworthiness dataset (Boháček et al., 2023) with over $10,000$ Czech news articles. The authors of this dataset propose the task of news trustworthiness classification, which recognizes the presence of select stylistic, linguistic, and semantic features concerning news credibility (e.g., clickbait, stereotypization, and hate speech). They define 4 classes of credibility: credible, partially credible, partially manipulative, and manipulative.

We choose this dataset because it presents a difficult two-stage classification problem in which the model must provide reasoning for its final prediction. It also comes with a detailed methodology describing the problem at hand, which we saw as a good fit for the system prompt of the LLM pipelines (described in Section 4.2). Notably, since the dataset was created in the pre-LLM era and deemed challenging for the standard architectures at the time, it falls into the category of datasets that were anticipated to significantly benefit from the advent of LLMs.

4 Methods

This section describes the two high-level classification pipeline approaches that we compare: the modular ensemble pipeline and the LLM pipeline. As representative examples of these approaches, we specifically evaluate the following models: Electra + XGBoost (*modular ensemble*), GPT-4 (*LLM*), and LLAMA 2 70B (*LLM*).

4.1 Modular Ensemble Pipeline

The general idea of the modular ensemble pipeline approach is to create a set of feature models, each yielding predictions about a single feature in the input, and a meta-model that combines the feature predictions into the final classification. Shown in Figure 1 is an overview of this pipeline adapted to our specific case, comprising 6 feature models and a final meta-model. Each feature model is a language model fine-tuned on a single task, corresponding to the Verifee dataset methodology. To match the language of the dataset, we use the Czech Electra (Kocián et al., 2021) as the fine-tuning baseline. Each feature model is fine-tuned on a task-specific dataset, as listed in Appendix C. The details and configuration of the fine-tuning are described in Appendix A. We open-source the code at `https://github.com/matyasbohacek/xgboost-vs-gpt4`. At input, each feature model is presented with the news article's title, body, and author.

The final meta-model is an XGBoost classifier (Chen and Guestrin, 2016), which receives the outputs of all the previous feature models as its input. Trained on pairs of the feature model representations and ground-truth classes from the Verifee dataset, this model seeks to predict the final trustworthiness class of the article.

4.2 LLM Pipeline

The general idea of the LLM pipeline approach is to leave the entire classification on an LLM, leveraging its emergent capabilities. Any information about the task at hand is conveyed through the system prompt (i.e., natural language).

Shown in Figure 2 is an overview of the LLM pipeline, adapted to our specific case. The system prompt contains the full news assessment methodology of the Verifee dataset and instructions about

Model(s)	Pipe	Lang.	F-1
Electra+XGBoost	mod.	CZ	0.533
GPT-4	LLM	CZ	0.531
GPT-4	LLM	EN	0.425
LLAMA 2 70B	LLM	CZ	0.188
LLAMA 2 70B	LLM	EN	0.256

Table 1: Micro F-1 scores on the testing set of the Verifee dataset. *Lang.* refers to the language used in the pipeline: CZ (Czech) or EN (English).

Model(s)	Pipe	Params.	Size
Electra+XGBoost	Mod.	78×10^6	0.9
LLAMA 2 70B	LLM	70×10^9	140
GPT-4	LLM	1.8×10^{12}	3370

Table 2: Model size comparison. *Params.* refers to the absolute number of parameters. *Size* refers to the size of the model in virtual memory in GB, estimated for a single-batch input (16-bit precision, 512 tokens), using https://github.com/RahulSChand/gpu_poor/.

the expected output format, following the chain-of-thought practices (Wei et al., 2022b). The system prompt is included in Appendix B.

During inference, the LLM is first presented with the system prompt, followed by the input news article. At the output, the pipeline first provides a list of features in the article, which it then uses for a final trustworthiness classification. The model is used in a zero-shot manner, meaning the pipeline is not trained on the Verifee dataset.

We specifically use GPT-4 (OpenAI, 2023) and LLAMA 2 70B (Touvron et al., 2023) as the LLM backbones, evaluating 2 configurations for each – one wherein the system prompt is left in its original language (Czech) and one wherein the system prompt is translated to English.

5 Results

This section describes the results of our comparison of the example modular ensemble and LLM pipelines.

5.1 Quantitative Performance

The F-1 scores obtained on the testing split of the Verifee dataset are presented in Table 1. The Electra + XGBoost (modular ensemble) with an F-1 score of 0.533 outperformed the LLM pipelines.

The confusion matrix of the predictions on the testing split of the Verifee dataset is shown in Figure 3. The models perform best on the edge classes (i.e., credible and manipulative) and struggle more with the center classes (i.e., partially credible/manipulative). While worse than the Electra + XGBoost, the GPT-4 pipeline performs better than the LLAMA 2 pipeline, which near uniformly predicts one class.

5.2 Training Requirements

The example modular ensemble pipeline approach, Electra + XGBoost, involves a multi-stage training process. First, 6 separate Electra models are fine-tuned for binary classification tasks. Next, these models analyze the news articles in the training split of the Verifee dataset and build up their feature representations, which are then fed into the XGBoost (meta-model classifier). The XGBoost model is trained to classify the news article into one of the four credibility classes based on the aggregated insights from the feature representations. On the other hand, the example LLM pipeline approaches, GPT-4 and LLAMA 2, are used out of the box and require no additional fine-tuning.

5.3 Deployment Requirements

Model statistics about deployment requirements are presented in Table 2. The example modular ensemble pipeline approach, Electra + XGBoost, can be executed on consumer-grade hardware, requiring 0.9 GB of virtual memory. In contrast, the LLM pipelines are 3 and 6 orders of magnitude larger in parameter size and require cloud-level GPU resources. LLAMA 2 requires about 140 GB of virtual memory, while GPT-4 requires 3370 GB.

6 Conclusion

We find that LLM classification pipelines may not necessarily be better than the pre-LLM-era classification pipelines on all classification tasks. In the case study of news trustworthiness assessment, deemed particularly challenging in the pre-LLM era, we identify an example use case in which an ensemble pipeline outperforms two popular LLM pipelines. While the LLM pipelines come with lesser training requirements, they pose orders of magnitude higher computational deployment costs.

While there are many exciting use cases of LLMs that can push NLP and other disciplines, further, we argue that critical work on the robustness of LLM-based methods is lacking. To that end, this narrow case study paper can serve as a template for similar task- and dataset-specific studies, together solidifying our understanding of where LLMs stand compared to their architecture predecessors.

Limitations

While we strive to make the comparison in this paper as fair and representative as possible, our analysis, of course, has limitations. Notably, we only compare the pipelines on a single classification task in two languages. The pipelines may exhibit different performance on different tasks and languages. Therefore, this dataset should not be seen as representative of all classification tasks – task-specific datasets must be used for each task to make judgments about LLM and pre-LLM-era pipelines on that particular task. We call for similar studies following this template in different tasks to offer a broader picture of where LLM classification pipelines stand compared to pre-LLM-era classification pipelines across tasks, languages, and datasets.

In terms of the architectures, it must be stated that the LLMs described in this paper operate in the domain of few- and zero-shot classification, whereas the ensemble pipeline is supervised. Moreover, one could argue that the performance of both of the examined pipeline approaches could be further improved using techniques such as hyperparameter optimization for the modular ensemble pipeline or LLM fine-tuning for the LLM pipeline. While likely true, we believe that evaluating both pipelines in a default setting without these additional techniques maintains a fair comparison of these methods as they would be used. Moreover, a more detailed comparison goes beyond the scope of this short paper.

An additional limitation we would like to point out is the number of parameters of the GPT-4 model, which we obtained from `https://www.semianalysis.com/p/gpt-4-architecture-infrastructure`. Albeit speculative, the estimate we refer to is supported by external evidence and several independent sources. Still, we must reiterate that this is not a precise number but rather a rough estimate.

References

Kabir Ahuja, Rishav Hada, Millicent A. Ochieng, Prachi Jain, Harshita Diddee, Krithika Ramesh, Samuel C. Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai. *ArXiv*, abs/2303.12528.

Dmitrii Aksenov, Peter Bourgonje, Karolina Zaczynska, Malte Ostendorff, Julian Moreno-Schneider, and Georg Rehm. 2021. Fine-grained classification of political bias in german news: A data set and initial experiments. In *WOAH*.

Aman Anand. 2020. Clickbait dataset.

Anthropic. Model card and evaluations for claude models.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023.

Matyáš Boháček, Michal Bravansky, Filip Trhlík, and Václav Moravec. 2023. Czech-ing the news: Article trustworthiness dataset for czech. In *Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

Chiranjib Chakraborty, Manojit Bhattacharya, and Sang-Soo Lee. 2023. Artificial intelligence enabled chatgpt and large language models in drug target discovery, drug discovery, and development. *Molecular Therapy. Nucleic Acids*, 33:866 – 868.

Szeyi Chan, Jiachen Li, Bingsheng Yao, Amama Mahmood, Chien-Ming Huang, Holly Jimison, Elizabeth D. Mynatt, and Dakuo Wang. 2023. "mango mango, how to let the lettuce dry without a spinner?": Exploring user perceptions of using an llm-based conversational assistant toward cooking partner. *ArXiv*, abs/2310.05853.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei dong Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2023. Label-free node classification on graphs with large language models (llms). *ArXiv*, abs/2310.04668.

John Joon Young Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Annual Meeting of the Association for Computational Linguistics*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. In *Annual Meeting of the Association for Computational Linguistics*.

Xiangjue Dong, Yibo Wang, Philip S. Yu, and James Caverlee. 2023. Probing explicit and implicit gender bias through llm conditional text generation. *ArXiv*, abs/2311.00306.

Raphael Antonius Frick. 2023. Fraunhofer sit at checkthat!-2023: Can llms be used for data augmentation & few-shot classification? detecting subjectivity in text using chatgpt. In *Conference and Labs of the Evaluation Forum*.

Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. 2022. A survey on text classification algorithms: From text to predictions. *Inf.*, 13:83.

Jonas Golde, Patrick Haller, Felix Hamborg, Julian Risch, and A. Akbik. 2023. Fabricator: An open source toolkit for generating labeled training data with teacher llms. *ArXiv*, abs/2309.09582.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, George Louis Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*.

Khyati Khandelwal, Manuel Tonneau, Andrew M. Bean, Hannah Rose Kirk, and Scott A. Hale. 2023. Casteist but not racist? quantifying disparities in large language model bias between india and the west. *ArXiv*, abs/2309.08573.

Matej Kocián, Jakub N'aplava, Daniel Stancl, and Vladimír Kadlec. 2021. Siamese bert-based model for web search relevance ranking evaluated on a new czech dataset. In *AAAI Conference on Artificial Intelligence*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *ArXiv*, abs/2003.02245.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq R. Joty, and J. Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. In *Annual Meeting of the Association for Computational Linguistics*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. *ArXiv*, abs/2310.07849.

Qingzi Vera Liao and Jennifer Wortman Vaughan. 2023. Ai transparency in the age of llms: A human-centered research roadmap. *ArXiv*, abs/2306.01941.

Zhexiong Liu. 2023. Ticket-bert: Labeling incident management tickets with language models. *ArXiv*, abs/2307.00108.

Vadim Lomshakov, Sergey V. Kovalchuk, Maxim Omelchenko, Sergey I. Nikolenko, and Artem Aliev. 2023. Fine-tuning large language models for answering programming questions with code snippets. In *International Conference on Conceptual Structures*.

Lefteris Loukas, Ilias Stogiannidis, Odysseas Diamantopoulos, Prodromos Malakasiotis, and Stavros Vassos. 2023. Making llms worth every penny: Resource-limited text classification in banking. *Proceedings of the Fourth ACM International Conference on AI in Finance*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. In *AAAI Conference on Artificial Intelligence*.

Kristina McElheran, J. Frank Li, Erik Brynjolfsson, Zachary Kroff, Emin M. Dinlersoz, Lucia Foster, and Nikolas J. Zolas. 2023. Ai adoption in america: Who, what, and where. *SSRN Electronic Journal*.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Rishabh Misra. 2022. News category dataset. *ArXiv*, abs/2209.11429.

Fabio Motoki, Valdemar Pinho Neto, and Victor Rodrigues. 2023. More human than human: Measuring chatgpt political bias. *SSRN Electronic Journal*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. In *Annual Meeting of the Association for Computational Linguistics*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Patrick Parra Pennefather. 2023. Being creative with machines. In *Creative Prototyping with Generative AI: Augmenting Creative Workflows with Generative AI*, pages 27–63. Springer.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *ArXiv*, abs/2302.06476.

Vipula Rawte, Prachi Priya, S.M. Towhidul Islam Tonmoy, Islam Tonmoy, M Mehedi Zaman, A. Sheth, and Amitava Das. 2023. Exploring the relationship between llm hallucinations and prompt linguistic nuances: Readability, formality, and concreteness. *ArXiv*, abs/2309.11064.

Aleksandra Revina, Krisztián Búza, and Vera G. Meister. 2021. Designing explainable text classification pipelines: Insights from it ticket complexity prediction case study.

Gusti Muhammad Riduan, Indah Soesanti, and Teguh Bharata Adji. 2021. A systematic literature review of text classification: Datasets and methods. *2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 71–77.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature Medicine*, 29:1930 – 1940.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo Hernandez, and Subbarao Kambhampati. 2023. On the planning abilities of large language models (a critical investigation with a proposed benchmark). *ArXiv*, abs/2302.06706.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2023. Efficient large language models: A survey.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. 2023. An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *ArXiv*, abs/2311.07397.

Qi Wang, Wenling Li, and Zhezhi Jin. 2021. Review of text classification in deep learning. *OALib*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *ArXiv*, abs/2304.14402.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *ArXiv*, abs/2309.01219.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *ArXiv*, abs/2302.10198.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *ArXiv*, abs/2310.04406.

P. Zicari, Gianluigi Folino, Massimo Guarascio, and Luigi Pontieri. 2022. Combining deep ensemble learning and explanation for intelligent ticket management. *Expert Syst. Appl.*, 206:117815.

A Modular Ensemble Pipeline: Training Details

The feature models in the modular ensemble pipeline (of the Electra architecture) are implemented using the Hugging Face 3 (Wolf et al., 2019) and PyTorch (Paszke et al., 2019) libraries. Namely, we use the `ElectraForSequenceClassification`[2] pipeline and train it using the default hyperparameters. If any of the feature-specific datasets is not already available in the same language as the Verifee news trustworthiness dataset, we translate it using DeepL[3]. The final-meta model (of the XGBoost architecture) is implemented using the DMLC XGBoost (Chen and Guestrin, 2016) library and also trained with the default hyperparameters.

B LLM Pipeline System Prompt

We use the following system prompt for the LLM pipelines, which is derived from the news assessment methodology of the Verifee dataset (Boháček et al., 2023). In the actual prompting, the model is asked to first list out the features found in the article. Then, it is asked to provide the final trustworthiness class prediction. Moreover, examples of the features outlined below were provided.

You are a perfect AI system capable of evaluating article trustworthiness. Consider only the information presented within the article and make assumptions based on the methodology.

Output in this JSON format: {{"explanation": list of criteria found in the article, "label": One of the trustworthiness labels}}

Base your evaluation solely on this methodology:

1. Trustworthiness Classification:

1.1 Trustworthy:

Positive Criteria (5+ required): *Citations from relevant authorities, Representation of all interested parties' views, Facts presented within context, Grammatically correct, neutral language, Identifiable author, Undistorted data*

Negative Criteria (1 or fewer allowed): *Missing citations, Unrepresented opposing views, Facts without context, Grammatical errors or overly expressive language, Anonymous author, Distorted data*

Forbidden Criteria: *Clickbait, Hate speech, Unjustified attack on an opinion opponent, Manipulation of reader, Conspiracy theories, Emotional appeals, Logical fallacies, Tabloid language*

1.2 Partially Trustworthy:

Positive Criteria: *Grammatically correct and neutral language, Undistorted data*

Negative Criteria (2-5 allowed): *Missing citations, Unrepresented opposing views, Facts without context, Grammatical errors or overly expressive language, Anonymous author, Distorted data, Clickbait, Emotional appeals, Tabloid language*

Forbidden Criteria: *- Hate speech - Unjustified attack on an opinion opponent - Manipulation of reader - Conspiracy theories - Logical fallacies*

1.3 Misleading:

Positive Criteria: *None required*

Negative Criteria (6-7 allowed): *Missing citations, Unrepresented opposing views, Facts without context, Grammatical errors or overly expressive language, Anonymous author, Distorted data, Clickbait, Emotional appeals, Tabloid language, Logical fallacies, Unjustified attack on an opinion opponent*

Forbidden Criteria: *Hate speech, Manipulation of reader, Conspiracy theories*

1.4 Manipulative:

Positive Criteria: *None required*

Negative Criteria (8+ allowed or any of the 3 forbidden criteria): *Missing citations, Unrepresented opposing views, Facts without context, Grammatical errors or overly expressive language, Anonymous author, Distorted data, Clickbait, Emotional appeals, Tabloid language, Logical fallacies, Unjustified attack on an opinion opponent, Hate speech, Manipulation of reader, Conspiracy theories*

Forbidden Criteria: *None*

2. Handling Unclassifiable Articles and Errors:

If an article's length or structure makes it unclassifiable or lacks sufficient content for analysis, label it as unclassifiable.

[2] https://huggingface.co/transformers/v3.0.2/model_doc/electra.html?highlight=electra#transformers.ElectraForSequenceClassification

[3] https://www.deepl.com/translator

C Modular Ensemble Pipeline: Datasets

Feature	Dataset	Description
Anger	GoEmotions (Demszky et al., 2020)	This dataset comprises 10,000 comments scraped from the internet, annotated for the emotions they convey. While the dataset recognizes 28 emotion classes, we only use the anger class versus a balanced sample of the remaining classes (including 'neutral') to model this as a binary classification task.
Clickbait	Kaggle Clickbait Dataset (Anand, 2020)	This dataset contains 32,000 headlines from 10 diverse news sources, classified as either clickbait or non-clickbait.
Hate speech	HateXplain (Mathew et al., 2020)	This dataset comprises 20,148 social media posts classified into 3 categories of hate speech (hate, offensive, and normal), with additional annotations about the target community and rationales.
Political bias	German News Bias Dataset (Aksenov et al., 2021)	This dataset contains 47,362 news articles from 15 news sources, classified into 5 categories of political bias.
Stereotypization	StereoSet (Nadeem et al., 2020)	This dataset comprises sentences with common gender-, profession-, race-, and religion-based stereotypes, as well as counterparts without stereotypes.
Seriousness	Kaggle News Category Dataset (Misra, 2022)	This dataset contains 210,000 news headlines classified into 42 news categories. We use only a subset of these categories (namely, 'style and beauty,' 'comedy,' 'entertainment,' 'wellness,' and 'home & living'), which we group under the umbrella category of tabloid news, and the rest, modeling this as a binary classification task.

Table 3: Overview of the datasets used for fine-tuning of the respective feature models. Each dataset is used for a single classification task.