Large language models for processing and intellectual large volumes heterogeneous texts analysis with identifying bots in social networks

Nickolay Rudnichenko[1,*,†], Vladimir Vychuzhanin[1,†], Andrii Simanenkov[2], Tetiana Otradskya[1], Denys Shvedov[1,†], Igor Petrov[3]

[1] Odessa Polytechnic National University, Shevchenko Avenue 1, Odessa, 65001, Ukraine
[2] Kherson state marine academy, Ushakov Avenue, 20, Kherson, 73003, Ukraine
[3] National University "Odessa Maritime Academy", Didrichson street 8, Odessa, 65029, Ukraine

Abstract

The paper describes the problems of analyzing and processing large volumes of heterogeneous texts in natural language in the task of identifying bots in social networks based on deep transfer learning methods, in particular large language models. An analysis of the specifics and key aspects of text content structuring, processing and analysis is provided, the relevance of the problem is substantiated, an analysis of existing approaches in the scientific literature is carried out, the advantages and possibilities of using artificial neural networks and machine learning to automate the processes social network users texts posts analyzing are listed. The set of input data selected for research is described, the choice of artificial neural networks language models is justified and the specifics of using transfer learning to adapt models to the bot search task are described. The technical means and services for implementing the work of the created web application are described, object-oriented models of the system are developed using the UML language in the form of use cases and components diagrams web application, software functionality, prototype pages and a user graphical interface are developed. The results of experimental studies of selected language models on an expanded input data set in modes with and without text explanations are presented. At the selected post, an analysis adapted neural network models results and work specifics was performed, promising ways for further research and development of the identified problems were identified.

Keywords

large language models, data mining, big data, data analysis, neural networks, bot detection

1. Introduction

In the modern information society, the Internet can be called an integral part of business, allowing any company to carry out business communications with such target groups as customers, resellers (distribution channels), PR, suppliers, competitors, current and potential employees of the company [1]. When conducting such communications, large volumes of heterogeneous data are generated, processed and stored, including multimedia files (images, videos), as well as not always clearly structured text content [2]. In this context one of the main trends in the Internet development in recent years is the rapid growth in social networks (SN) popularity, which are increasingly used for marketing purposes, to promote a particular product, service, expert, opinion leader, software applications and services, etc. In these conditions, using SN as a source of obtaining data and forming an information base for clients is appropriate and important [3]. For modern SNs, the following characteristic effects and properties can be identified, which are important to consider when using

ICST-2024: Information Control Systems & Technologies, September 23-25, 2023, Odesa, Ukraine.
* Corresponding author.
† These authors contributed equally.
nickolay.rud@gmail.com (N. Rudnichenko); vint532@gmail.com (V.Vychuzhanin); symon2007@ukr.net (A. Simanenkov); tv_61@ukr.net (T. Otradskya); studylearner@gmail.com (D. Shvedov); firmn@gmail.com (I. Petrov);
0000-0002-7343-8076 (N.Rudnichenko); 0000-0002-6302-1832 (V.Vychuzhanin); 0000-0003-0797-5276 (A. Simanenkov); 0000-0002-5808-5647 (T. Otradskya); 0009-0002-4823-8782(D. Shvedov); 0000-0002-8740-6198 (I. Petrov);

them to solve business problems: the presence of system's users own opinions; SN members opinions changes under others influence; different significance (priority or weight) of some users opinions due to their level of expertise; SN members susceptibility changing degree to influence among themselves; the presence of indirect influence or dependencies between users in the entire chain of existing social contacts; experts existence, i.e. "opinion leaders" in a certain topic; sensitivity certain threshold presence to changes others vocabularies; localization of created groups according to characteristics (by interests, with similar opinions, united by social or gender characteristics); "social correlation" factors; the existence of external influencing factors and third-party agents (media, sellers or manufacturers); the impact of SN on the opinions dynamics in a virtual community; the possibility of forming coalitions or teams, interest groups; game or interactive interaction between users in interactive mode [4-6].

At the same time, it should be noted that at present, the problems of information protection and countering information threats in various data exchange systems, including SN, the content of which is formed in various ways and methods, from manual writing of thematic texts to synthetic generation based on the use of various intelligent technologies and tools [7]. In fact, in practice there are often situations when it is necessary to enter into correspondence with other SN users for the purpose of consulting, exchanging opinions or assessing published content nature and quality [8]. Because these processes cannot be fully automated and are often performed manually, as a result of which they are resource-intensive and expensive for business. In such cases, it is necessary to guarantee the correctness of the information received, its targeted nature and minimize the risks of receiving incorrect data. In this regard, an urgent task is to identify and detect, by direct and indirect behavioral, linguistic and semantic characteristics, automated programs (bots) hiding under user profiles that publish unreliable and deliberately false information, seeking to fraudulently obtain personal and commercially valuable information of other users (telephone numbers, document scans, payment numbers details, credit and debit cards, etc.), as well as contributing to the destabilization of sentiment in the SN society, which leads to negative consequences for business when promoting goods and services [9].

A bot or virtual assistant in a broad sense is specialized software capable of simulating real user's actions, in particular generating text content [10]. Usually, in the case of managing a user account through an software in an automated manner, it is considered that this process is implemented by a bot. When an account is controlled partly by a program and partly by a person, the term "cyborg" is used [7]. In practice, there are simple bots that act in a directive manner and execute a set of clearly defined commands, as well as with support for self-learning functions [11].

It should be noted that not all of the existing bots in SN were developed by attackers with the aim of causing informational and commercial harm; many of them were created to support business, for example, to automate a number of tasks associated with routine work in various applied areas, including marketing, sales and technical (operational) customer support industries. Bots can provide processes for attracting and qualifying leads, accounting for sales of goods and services, and accepting payments.

A typical example is chatbots in SN, which are an effective format for communication between individual companies and users in a 24/7/365 format [12].

Within identified problems framework the key difficulty is the laboriousness and non-trivial nature of assessing and analyzing heterogeneous and semi-structured text content obtained in the context of a specific user profile, from his public messages, open publications and correspondence in SN [13]. Manual mode in this context is almost impractical, and therefore a rational approach is to use intelligent technologies, methods and data analysis models that allow automating the process of assessing text content received from SN users to exclude communication specifically with malicious bots, as well as general adequacy and suitability to support business decisions on marketing tasks [14].

Such approaches are based on machine learning (ML) models and algorithms, statistical approaches, deep learning (DL) and artificial neural networks (ANN), which makes it possible to build adaptive pipelines for aggregated data reconnaissance analysis processes, their preprocessing, normalization, and training, testing and assessing generated models accuracy [15,16]. As a result, serialized objects of the created models can be saved as separate dependencies and dynamically integrated into the ecosystem of processes for mining new data streams (text messages from correspondence with SN users) through various information systems and software applications.

It should be noted that in the most popular and developing area of ANN, there is currently active research and implementation of methods and models from the scientific field of texts processing and analysis in natural language (NLP), where, in turn, promising areas are technologies for automatic heterogeneous texts generation and evaluation [8-10].

New DL large language models (LLM) are being actively created and applied for this purpose. However, due to their stochastic principle of operation, various training data, a large hyperparameters number, interpreting complexity and assessing the mutual influence of parameters on each other and quality loss possibility with new versions, possibilities studying problem using LLM different types for applied problems by comparing and tracking changes in such models quality, their configurations and versions using the example malicious bots identifying specifics in SN user's profiles based on text content analysis [11-13].

2. Analysis existing researches

Recent advances in language models can be attributed mainly to deep learning techniques, advances in neural architectures such as transformers, advanced computing capabilities, and the availability of training data obtained from the Internet. These developments have led to a revolutionary transformation, allowing the creation of LLMs capable of approximating human-level performance on certain assessment tests [9, 16].

LLMs, especially pre-trained data models types, according to a number of studies [17], are capable of providing rich capabilities for understanding, analyzing, evaluating and generating textual content in a wide tasks range.

Due to this, the demand for LLM has increased, also due to the growing need for machines to perform complex language tasks such as translation, summarization, information retrieval and conversational interaction. LLMs achieve this mastery by self-learning on large text datasets [18]. After fine-tuning to perform tasks of heterogeneous texts large volumes analyzing, LLMs demonstrate a significant increase in performance, in some cases [19] exceeding the performance of models trained entirely from scratch.

These features of language models contribute to LLM use when training them on large data sets, which allows us to note the fact that scaling models size themselves and data sets volumes used for training and testing leads to their generalization ability further improvement.

It should be noted that, according to a different science approaches [20], the quality of LLM results is influenced not only by the presentation task execution examples in queries, but also by how the task itself was described in natural language in the query.

An important part of working with LLM is engineering queries (pieces of text queries sent to the model input that formalize the task that the LLM must perform, taking into account additional rules, hints, examples and semantic context) to improve the efficiency and accuracy of their use. This process, according to the authors [21], is based on the sequential execution of procedures for changing and optimizing input queries to improve the target result generated by LLM for applied problems.

An important aspect in this case is that the final quality of the model can vary significantly depending on how exactly the query was formed, even in the case where two different queries have the same essence and purpose, but different formation procedures.

As a result of the work of such an ANN model, it is possible to generate a possible tokens wide probability distribution that are continuations of text sentences. The choice of the final token for the model is often determined through the stages of data sampling and its tuning; in this case, an important role is played by hyperparameter values selection that can influence the trade-off between generated text diversity and accuracy [22].

All this can be useful in evaluating users' text posts on SN to analyze their profiles for anomalous behavior and identify bots.

It should be noted that one of the key LLM common architectures disadvantages, for example, transformers, is the inability to model clear query execution logic and model's tendency to make actual errors on large text prompts [11,17,19]. To decide such problems, various methods are being actively developed and studied to improve the accuracy and quality of LLM work for different semantic contexts. An example of the approach used is relational or non-relational databases connection as a source of relevant information and symbolic memory, which makes it easier for models to process data. In this case, by combining the knowledge chain method and the database, it is possible to provide the LMM with the ability to access factual and symbolic information obtained or stored as needed [17].

Thus, analyzing existing works on the study of this topic, despite the identified difficulties in using LMMs and their shortcomings, it should be noted the relevance and feasibility of the development and use of such models for text data analyzing tasks, in particular, in identifying bots SN's context.

3. Models implementation and technical aspects

To implement and apply the functionality of ANN models within the framework of the problem under researching, it is necessary to find or create significant size test data sets. As a basis, it was decided to use existing publicly available data sets fragments, pre-processing, cleaning, and also aggregate a number of adapted samples to give the data greater balance and diversity. For identifying bots task in SN, which is expediently reduced to a classification task, the "PAN19 Author Profiling" data set was used [23].

This dataset was created to help identify bot users online. The data set includes 100 posts from different users on the social network Twitter, as well as an indicator of whether the user is a bot. The data set is balanced. To assess LLM adequacy and accuracy for a given data set, it is advisable to use binary classification metrics, such as accuracy, recall, precision, f1.

To conduct experiments on popular and new LLMs, it is necessary to provide access to their functionality by connecting available APIs. After analyzing the available options, it was found that:

- For proprietary models, access is most often provided thanks to APIs specially developed for them for text generation tasks.
- Open-source models can be accessed through public repositories and implemented drivers to support their implementation.
- To independently develop language models, we need to create our own or use an existing LLM training and activation framework.

As part of this study, it was decided to use the following LLM models (adapting them to our task): GPT2, Bloomz-1b1 and Mistral-7B.

GPT2. Compared to the latest models, GPT2 has significantly fewer parameters and less ability to understand text. But, due to the small size of the model, it was decided to use GPT2 as the basic "zero" quality level when comparing with other models, since its performance can be quickly calculated, which contributes to the implementation models comparative testing concept.

Bloomz-1b1. This open-source LLM accepts about 1.1 billion parameters, which is relatively small compared to other models. This reduces its potential for understanding text, but its use will allow us to measure how flexible LLMs can be for the task of identifying bots in SN with a relatively small parameters number.

This will also allow local experiments to be carried out relatively quickly. This model was initially trained to analyze semantic instructions in the text, which justifies the advisability of its use in the conversational style text posts analysis.

Mistral-7B. Open source model developed by MistralAI. a model designed to solve NLP problems with a high-performance degree. According to the authors [24], Mistral 7B outperforms Llama 2 13B in all evaluation metrics, the model uses attention to grouped queries for faster generation, combined with attention to sliding windows to efficiently process arbitrary-length sequences with reduced generation speed.

Using this model will make it possible to better represent the open source LLM development field; in this case, a larger model is presented than bloomz-1b1 and with the ability to specify instructions.

To adapt models to the problem under consideration, it is proposed to use the concept of inductive type transfer learning (TL) with elements of cross-modality [11, 17, 25].

If $f_{w,s} : X \to Y$ be a pre-trained model on the source dataset D_s where $w_s \in \Re^D$ denotes D-dimensional weight vector of the pre-trained LLM.

Given the target dataset D_t , the fine-tuning method minimizes the standard negative log-likelihood $L_t(w) = \sum_{i=1}^{N_t} \log p_w(\frac{y_t^i}{x_t^i})$ using the stochastic gradient descent $w(t+1) = w(t) - \eta \nabla_w L_t(w), w_0 = w_s$, where η is a step size and $\nabla_w L_t(w)$ denotes a stochastic estimate of the loss gradient using a mini-batch of data. Thus, the fine-tuning is a maximum likelihood estimation whose the log-prior is centered at w_s .

Using the above pre-trained models, we reduce training time by training only the last layer of models with significantly fewer variables. This is due to the fact that if we do not "freeze" the variables of the pre-trained model, then during the training process on a new data set the values of the variables will change (the last layer will be filled with random values); therefore, the models can make large errors when analyzing the text, which, in turn, will entail strong changes in the initial weights in the pretrained model.

The advantage of accessing selected LLMs through a selected API is that it supports the use of the company providing access to the LLM computing capabilities, but the disadvantage of this is that the URLs and API request formats vary depending on the policies and restrictions of the company providing access to the LLM. This complicates the research process, because it is necessary to develop methods for implementing various request formats. To solve this problem, it was decided to use the OpenRouter service. This service allows us to query proprietary LLMs using a single interface, regardless of the specific model or company providing access to it.

For the selected open-source models, it was decided to adapt a public repository for obtaining trained LLMs and datasets for them - "Huggingface", as well as the library developed by this service, "transformers". With this repository, it is possible to index the most popular LLMs over time; through "transformers", it is possible to locally launch most models directly from the repository due to a special shared software interface for their use.

The technical side of performing research on selected LLM models is implemented in the form of a client-server web application with a simplified graphical user interface.

To develop the main functionality of the project, the Python programming language version 3.7.12 was chosen, which allows us to use convenient data collections and integrate libraries for processing and analyzing text data.

To implement a number of functionalities within a web application, it is necessary to create an interactive interaction between the user and the web page; for this purpose, the JavaScript programming language is used. To build a web application framework and improve work with the database, it was decided to use the Django framework.

To store data, it was decided to use a PostgreSQL relational database; a database of 4 tables was created to store metadata about models, experimental results, sets of hyperparameters and datasets. To ensure easier dependency and version management, the ability to run the proposed platform on many platforms and the logical distribution of the system architecture, it was decided to use Docker and Kubernetes. Based on the project concept, a use case diagram was created, the result is shown in Fig. 1.

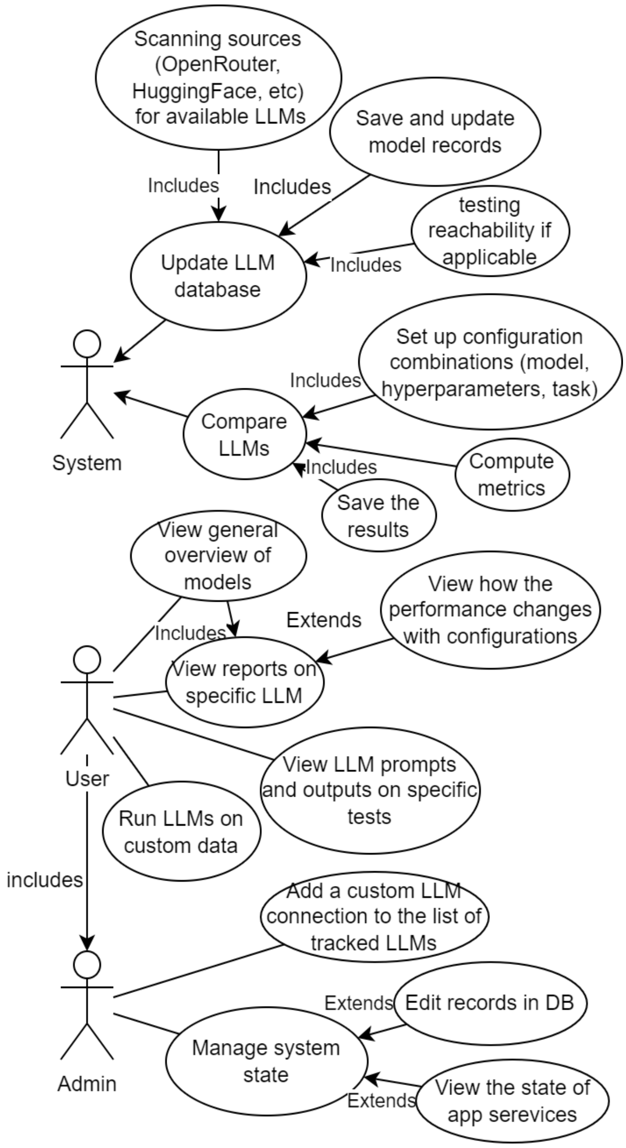

Figure 1: Use case diagram of developed web-application

A project software implementation feature is a wide range of configurations for the task of analyzing text posts to identify bots in SN, integration with the X platform to gain access open posts data and to automate LLM's assessment, as well as the models automatic tracking functionality in open repositories, which is due to the previously described change their qualities with different versions. The web application is used as follows: the user interacts with the system through the web application, he can select different routes, each of which provides the functionality necessary to satisfy the user's functions (Fig.1). The experimentation process is carried out automatically; the system selects combinations of LMMs, configurations and tasks on which to conduct experiments, saving the results in database tables. An administrator is a user who hosts a developed platform for conducting experiments manually by setting configurations, using and testing the latest models and methods, making modifications to experimental methodologies, and also editing the open source code of the developed web application. A diagram of the system components involved in the deployment process is shown on Fig.2. The cluster consists of the following elements (each node is a separate virtual or real machine):

1. Management node. This node performs tasks related to task orchestration, message passing, and cluster management. Specifically, it hosts a kubernetes cluster control ("Control Plane"), a Kafka message broker server for communicating data and messages between individual applications in the cluster, and an Apache Airflow work orchestration server to perform tasks of tracking and evaluating

language models. Since this node is the most important in the cluster, it was decided not to place on it only the code that is taken from trusted libraries (Kubernetes, Kafka, Apache Airflow).

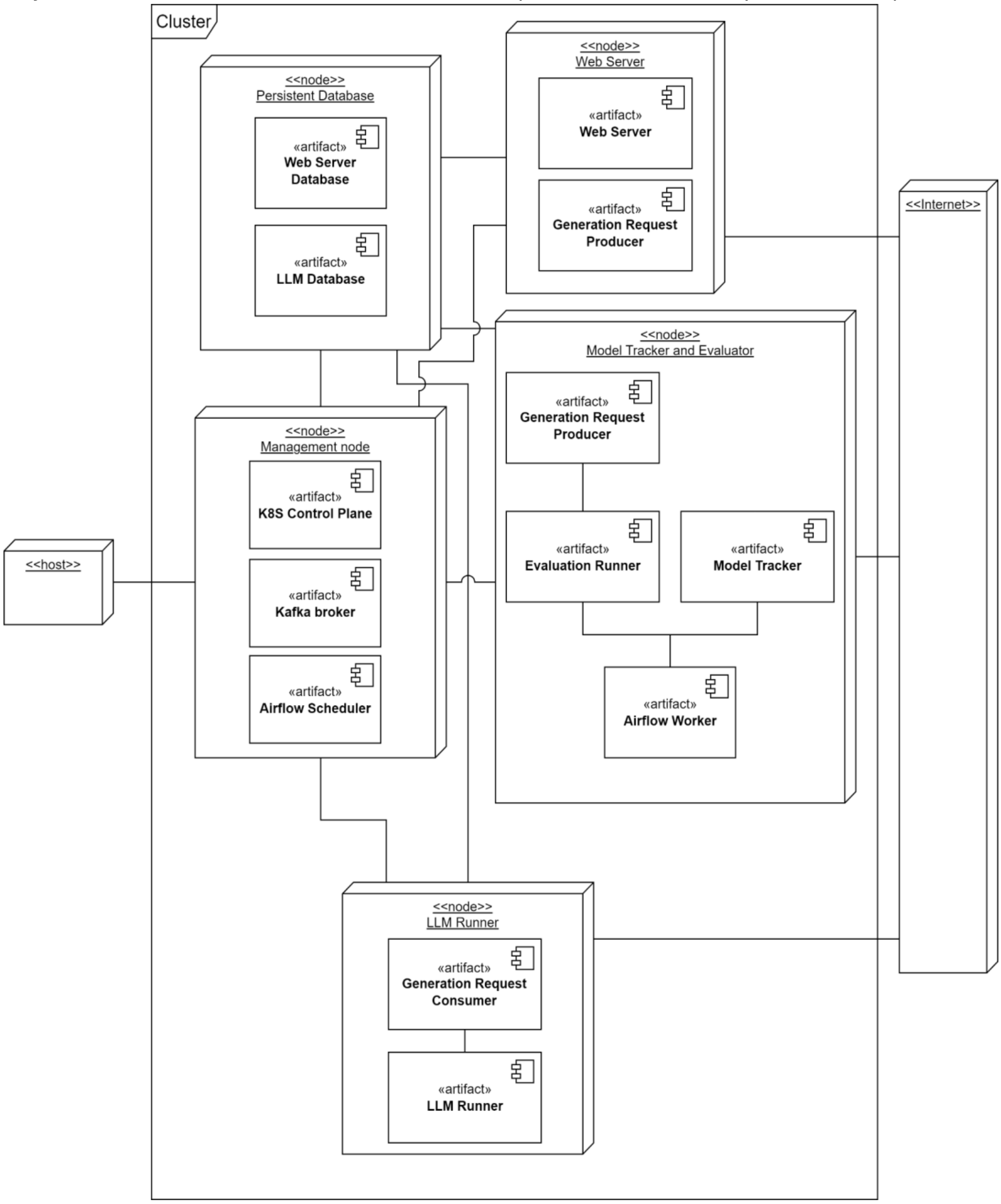

Figure 2: Components diagram of developed web-application

2. Persistent database node. This node hosts the database servers required for the cluster. Placing them on a separate machine allows us to optimize this node for constant data safety.

3. Web server node. This node contains a web server for user interaction with the system, as well as a method for creating Kafka messages (this is necessary for functionality where the user requests generation by some model).

4. Node for tracking and evaluating language models. This node hosts containers and methods for performing language model tracking and evaluating them. In addition, these methods are run by the Airflow task runner if the orchestrator decides to run them, and the node also hosts the Kafka message generation method (this is necessary for functionality where text generation needs to be requested).

5. Language model launch node. This node hosts methods for language models to generate text, as well as a method for receiving Kafka messages about text generation.

6. Internet. The cluster needs to have access to the Internet to serve user requests, find and track language models, as well as links to API requests for text generation to evaluate models.

7. Host. The cluster management node must have access to the host machine, from which it will receive server management commands.

To create the system, it was decided to first implement 4 main pages that meet the functional requirements: a page with navigation, a page with viewing metrics for a specific LLM on text analysis tasks, a page with viewing LLM answers for a set of posts for a classification task, a page with the ability to enter by the user data regarding the task and obtaining the results of the LLM work.

These pages will allow us to obtain the most key results from assessing the performance of models and will be responsible for a larger amount of information that can be reflected in reports. Based on these requirements, an interface mockup was developed (Fig.3).

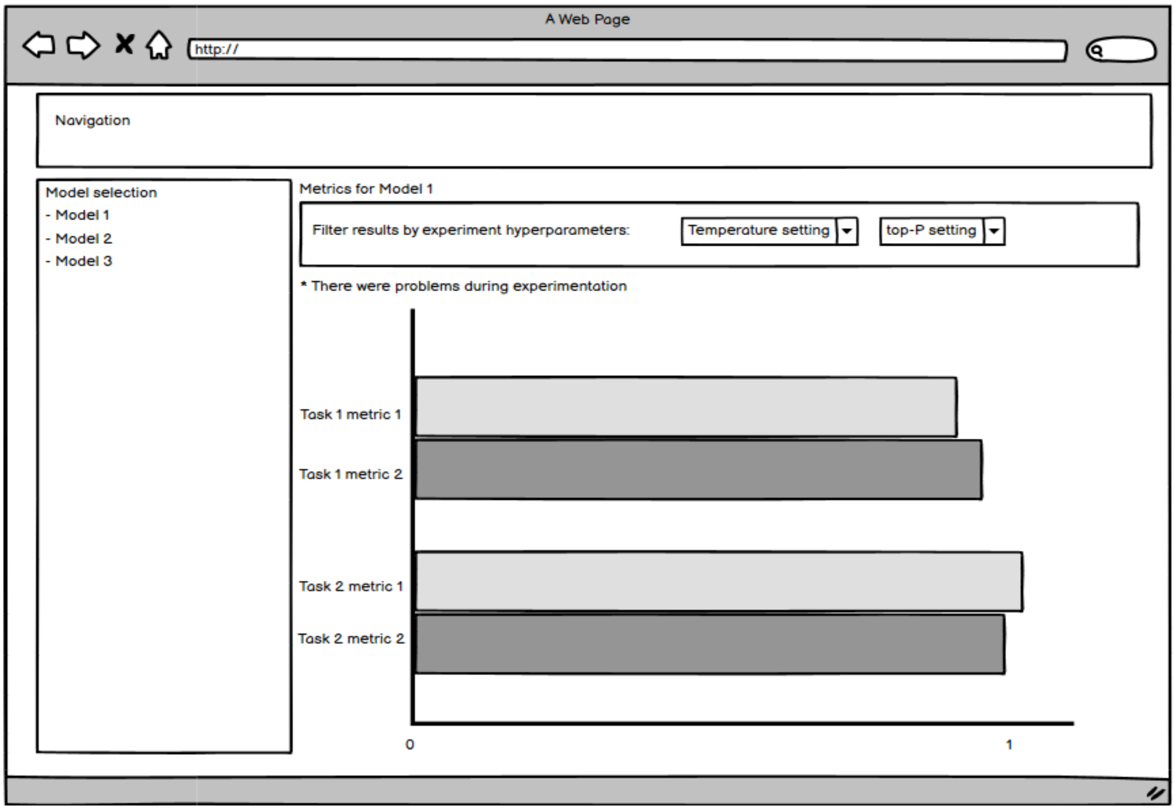

Figure 3: Web-application interface mockup

4. Experiments and results analysis

Table 1 presents the metrics for the task of identifying bots in SN according to the history of their publications based on TL adapted GPT2, bloomz-1b1, mistral-7b LLMs and without it (in the latter case, the results were 3-4 times worse compared to the adapted version).

Experiments were conducted with two types of request ("prompt type") - the request type "with explanation", which contains explanations regarding what publications bots usually make (advertising posts, duplicate posts, links to news, or too monotonous posts), and request "without explanation" in which there is no given explanation. The size of the augmented dataset is 6,760,000 records.

Analyzing the results obtained, it can be concluded that different LLMs have significantly different quality, regardless of their size. For example, the Recall metric shows that bloomz-1b1 flags users as a bot more often than others, so it makes no sense to use it in practice, while the mistral-7b model, which has the same size, showed significantly greater accuracy (more than 0.9) as a bot classifier. Also, mistral-7b in terms of metrics corresponds to better results than gpt2 and bloomz-1b1, while being almost 7 times larger than bloomz-1b1. Analyzing the LMM metrics that most efficiently coped with this task (mistral-7b and bloomz-1b1), it was concluded that it is possible to accurately locate bots in online networks due to LMM, but this requires additional fine-tuning and data preprocessing to obtain a greater degree of adequacy and model generalizing ability, while LLMs sensitivity used to a hyperparameters number in experiments case performed is not great.

Table 1
The results of LLM's metrics on the task of identifying bots in SN

LMM	Prompt Type	Recall	Precision	F1	Accuracy	Unfit Answers
gpt2	Without Explanation	0,87	0.66	0.22	0.75	15
gpt2	With Explanation	0,84	0.64	0.26	0.73	10
bloomz-1b1	Without Explanation	0.79	0.79	0.31	0.7	25
bloomz-1b1	With Explanation	0.77	0.78	0.28	0.72	23
mistral-7b	Without Explanation	0.95	0.9	0.14	0.92	2
istral-7b	With Explanation	0.93	0.87	0.12	0.91	3

Figure 4: Visualizations metrics iteration on GPT2 model

In addition, we note that the presence in the request of an explanation of what publications from bot accounts usually are had a negative impact on LLM quality of for all metrics.

It is therefore concluded that query engineering should take into account that a new modified query format may degrade LLM quality, even if it has information added with the intention of model quality improving, so a separate procedure for evaluating new query formats must be carried out.

Fig.5 shows posts fragment viewing result from one of the users (dataset records), when the model produces different classes depending on the request's length and content.

The user whose post text is given above is actually a bot and often publishes posts - job vacancies and posts about IT topics. It is possible to assume that this user is a robot of some company for recruiting new employees and does not have a negative effect within SN, because controlled by organization's employees. The GPT2 model classified this user as a bot with 77% confidence, bloomz-1b1 with 88%, mistral-7b with 94%. Using this example, we can clearly monitor LLM models correct operation in classifying the text of a user's post as belonging to a bot. This can be explained by the fact that the formalization of the task in the request describes posts publication from bots as those that contain explicit advertising (including links), repeated and close-in-context phrases in posts, news headlines in different registers using highly relevant anchors, or sound too monotonously, without text tone changing signs.

User's post history (user ID = 104eee2839f69f59af377fec70eaaf7b):

- IT Security Analyst: IT Security Analyst – Berkshire – Permanent – SOC, Threat, vulnerability, SIEM, risk, malware Outsource UKs cyber team have an exciting opportunity for an IT Security Analyst to join an organisation that are currently insourcing... https://t.co/sKOc98PIad
- CRM Developer (Dynamics): I have an immediate requirement for a Dynamics CRM Developer to join my client on an initial 6 month contract (Outside of IR35) . You will be working on a major Dynamics 365 project for a leading organisation based in... https://t.co/NEwKWpVel5 https://t.co/mTipJar3V4
- The election overseer for critical Palm Beach County says there is no way the recount for 3 races will be finished by Thursday's deadline: The election overseer for a critical county in Florida confirmed to CNN on Sunday what observers in both parties... https://t.co/aXmntXQhBN https://t.co/VoI9MzVkpQ
- Software Engineer – C++ / C# / UML: Software Engineer – C++ / C# / UML – Various Levels Commutable from Uxbridge / Slough / Watford / High Wycombe / Staines / Twickenham A market leading Global manufacturer is seeking to recruit a Software Engineer to... https://t.co/wA0S8fmqZU https://t.co/RmSKLPX8ME

Figure 5: Bot post example collected to the dataset

It should also be noted that most of the post's publications from this user sound more energetic.

5. Conclusion

As a result of the research, the application and adaptation of existing large language models for processing and intellectual analysis large volumes heterogeneous texts was carried out when identifying bots in social networks. The developed web application allows us to connect, select, track updates to the API and adapt selected LLMs to input data sets, automating all data analysis stages in individual pipelines using AirFlow and other technologies. The GPT2, bloomz-1b1, mistral-7b models adapted on the basis of TL generally successfully cope with identifying bots task in SN based on their text posts; the greatest accuracy is achieved by the mistral-7b model without the function of issuing explanations. Based on the research results obtained, we can conclude that the presence of additional explanations when analyzing aggregated heterogeneous user texts, the volume of which exceeds the length of an individual post, often imposes additional restrictions on ANNs work and limits their generalizing ability. In the absence of this parameter, the performance of ANN models is more weighted, however, this effect may be due to data set specifics. In the future, a rational way of research is to empirically search for the model operation parameter influence degree and identify its relative value in fractional form, taking into account the TL approach.

References

[1] N. Rudnichenko, V. Vychuzhanin, N. Shibaeva, I. Petrov, T. Otradskya, Intelligent Data Clustering System for Searching Hidden Regularities in Financial Transactions, in: 11-th International Conference «Information Control Systems & Technologies» (ICST-2023) CEUR-WS, 3513, 2023, pp. 163-176.

[2] V. Vychuzhanin, N. Rudnichenko, Z. Sagova, M. Smieszek, V. V. Cherniavskyi, A. I. Golovan, M. V. Volodarets, Analysis and structuring diagnostic large volume data of technical condition of complex equipment in transport. IOP Conference Series: Materials Science and Engineering, Volume 776, 24th Slovak-Polish International Scientific Conference on Machine Modelling and Simulations - MMS 2019, 3-6 September 2019, Liptovský Ján, Slovakia, 2019 pp.1-11. DOI:10.1088/1757-899X/776/1/012049.

[3] N. Rudnichenko, V. Vychuzhanin, I. Petrov, D. Shibaev, Decision Support System for the Machine Learning Methods Selection in Big Data Mining, in: Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020), CEUR-WS, 2608, 2020, pp. 872-885.

[4] C. Segalina, D. Cheng, M. Cristani, Social profiling through image understanding: Personality inference using convolutional neural networks, Computer Vision and Image Understanding 156 (2017) 34–50.
[5] F. Liu, Zh. Li, Ch. Yang, D. Gong, H. Lu, F. Liu, SEGCN: a subgraph encoding based graph convolutional network model for social bot detection, Scientific Reports 14 (2024). DOI: 10.1038/s41598-024-54809-z.
[6] M. Zhou, W. Feng, Y. Zhu, D. Zhang, D. Yuxiao, J. Tang, Semi-Supervised Social Bot Detection with Initial Residual Relation Attention Networks, Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track (2023) 207-224. DOI: 10.1007/978-3-031-43427-3_13.
[7] S. Gera, A. Sinha, T-Bot: AI-based social media bot detection model for trend-centric twitter network, Social Network Analysis and Mining 12 (2022). DOI: 10.1007/s13278-022-00897-6.
[8] Z. Ellaky, F. Benabbou, S. Ouahabi, N. Sael, A Survey of Spam Bots Detection in Online Social Networks, in: Conference: 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), 2021, pp. 58-65. DOI: 10.1109/ICDATA52997.2021.00021.
[9] N. Sadeghi, N. Riahi, Comparison of the effect of the generative model on the performance of deep neural networks and transformer in contextual social bot detection (2023). DOI: 10.21203/rs.3.rs-2556040/v1.
[10] E. Kheir, R. Daouadi, R. Rebaï, I. Amous, Bot Detection on Online Social Networks Using Deep Forest. Artificial Intelligence Methods in Intelligent Algorithms (2019) 307-315. DOI: 10.1007/978-3-030-19810-7_30.
[11] S. Pulipati, Malicious Social Bots Detection in Online Social Networks with Using Ensemble Model 14 2510 (2022). DOI:10.9756/INT-JECSE/V14I2.232.
[12] P. Pham, L. Nguyen, B. Vo, U. Yun, Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. Information Systems 103 (2021). DOI:10.1016/j.is.2021.101771.
[13] M. Duddu, D. Mahesh, Detection of Social Bots in Twitter Network, in: Proceedings of International Joint Conference on Advances in Computational Intelligence, 2023, pp.655-668. DOI:10.1007/978-981-99-1435-7_53.
[14] M. Mendoza, E. Providel, M.L. Santos, S. Valenzuela, Detection and impact estimation of social bots in the Chilean Twitter network, Scientific reports 14 6525 (2024). DOI:10.1038/s41598-024-57227-3.
[15] N. Rudnichenko, V.Vychuzhanin, T. Otradskya, I. Petrov, I. Shpinareva, Hybrid Intelligent System for Recognizing Biometric Personal Data, in: Proceedings of the 3rd International Workshop on Computational & Information Technologies for Risk-Informed Systems (CITRisk 2022) the co-located with XXII International scientific and technical conference on Information Technologies in Education and Management (ITEM 2022), CEUR-WS, 3422, 2023. pp. 74-85.
[16] M. Zhou, D. Zhang, W. Dan, G. Yuandong, Y. Yangli-ao, T.J. Dong, LGB: Language Model and Graph Neural Network-Driven Social Bot Detection (2024). DOI: 10.48550/arXiv.2406.08762
[17] S Ozdemir. Quick Start Guide to Large Language Models: Strategies and Best Practices for Using ChatGPT and Other LLMs (2023).
[18] A. Wang, SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems (2019). DOI:10.48550/arXiv.1905.00537
[19] J. Devlin, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language 4 (2018) 29. DOI:10.48550/arXiv.1810.04805
[20] H. A. Naveed, Comprehensive Overview of Large Language Models (2023). DOI: 10.48550/arXiv.2307.06435
[21] M. Grohs, Large Language Models can accomplish Business Process Management Tasks (2023). DOI:10.48550/arXiv.2307.09923
[22] R.E. Turner, An Introduction to Transformers (2023). DOI:10.48550/arXiv.2304.10557
[23] Data set for bot user classification. URL: https://zenodo.org/records/3692340

[24] T. Wolf Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp.38-45. DOI:10.48550/arXiv.1910.03771

[25] M. Sanghoon, H. In, J. Wonik, C.M. Jae, R. Jisu, S. K. Dae, K. Kee-Eung, J. Changwook, PAC-Net: A Model Pruning Approach to Inductive Transfer Learning, in: Proceedings of the 39th International Conference on Machine Learning, PMLR 162, 2022. DOI:10.48550/arXiv.2206.05703