Experimental chatbot for accelerated digital transformation in public institutions

Răzvan-Alexandru DUȚESCU, Mihai APOSTOL, Monica BARBU

National Institute for Research and Development in Informatics – ICI Bucharest, Romania

razvan.dutescu@ici.ro, mihai.apostol@ici.ro, monica.barbu@ici.ro

Abstract: For a large number of citizens dealing with public authorities can feel like a harrowing experience. With a large amount of bureaucracy and ever-changing rules and procedures, it can be quite confusing for people to achieve their goals. The aim of this paper is to prove that a chatbot can be trained with minimal coding experience, by having a robust dataset created by people with domain-specific knowledge. This would empower every public institution to customise a helper chatbot. This chatbot would not only provide common information about required documents for whatever needs to be done, but also more general information pertaining to the specific aspects of that institution, such as its location, how to get to it, or public transportation connections nearby. The aim is to provide the necessary steps so that any public institution willing to undertake this digital transformation step can do so without much outside help or technical know-how in the field of Artificial Intelligence.

Keywords: Metaverse, Artificial Intelligence, Large Language Model (LLM), Digital transformation.

Chatbot experimental pentru transformare digitală accelerată în instituțiile publice

Rezumat: Pentru un număr mare de cetățeni, interacțiunea cu autoritățile publice poate să fie o experiență chinuitoare. Din cauza birocrației excesive și a regulilor și procedurilor aflate într-o continuă schimbare, poate fi destul de greu pentru oameni să-și îndeplinească obiectivele. Scopul acestei lucrări este acela de a dovedi că un chatbot poate fi instruit cu o experiență minimă de codificare, pe baza unui set de date robust creat de oameni care au cunoștințe specifice domeniului. Astfel, instituțiile publice ar fi împuternicite să își creeze propriul chatbot personalizat. Acest chatbot ar oferi nu doar informațiile punctuale despre actele necesare pentru orice ar trebui făcut, ci și informații mai generale referitoare la aspecte individuale ale instituției, precum locația acesteia, direcții de ajuns la locație sau conexiuni la transportul public. Obiectivul este de a oferi pașii necesari astfel încât orice instituție publică dispusă să întreprindă acest pas de transformare digitală să poată face acest lucru fără prea mult ajutor extern sau fără prea multe cunoștințe tehnice în domeniul Inteligenței Artificiale.

Cuvinte cheie: Metavers, Inteligență Artificială, Modele Lingvistice Mari (LLM), Transformare digitală.

1. Introduction

Large Language Models (LLMs) have revolutionised various sectors including public and private institutions. LLMs have the potential to transform how government agencies, educational institutions, healthcare providers, and other public entities operate, enhancing efficiency and improving services. LLMs have emerged as cutting-edge artificial intelligence systems designed to process and generate text, aiming to communicate coherently (Naveed et al., 2023). This paper aims to introduce the concept of an LLM in the process of digital transformation within institutions that work extensively with the public (Barbu et al., 2024; Floroiu & Timisică, 2024).

A Large Language Model is a type of Artificial Intelligence (AI) designed to understand and generate text in a human-like manner, based on a vast amount of data. These models are built using deep learning techniques, particularly neural networks, which allow them to process and learn from large datasets. The term "large" in LLM refers to the size of the model which includes millions to billions of parameters, and the extensive volume of training data, which can encompass a diverse range of text sources such as books, articles, websites, source code and more.

Today's modern LLMs are capable of performing various tasks like text and code generation, tool manipulation, reasoning, understanding, language translation, sentiment analysis and question

https://doi.org/10.33436/v34i3y202408

answering in zero-shot and few-shot settings in diverse domains, even without requiring any fine-tuning on downstream tasks (Naveed et al., 2023).

Text-to-speech provides great accessibility for user interaction alongside the simple text, especially in the metaverse (Petre et al., 2023). Google Text To Speech API provides text conversion into audio file formats such as MP3, WAV, and others (Marinescu & Iordache, 2023). It is also capable of SSML (Speech Synthesis Markup Language) which provides pronunciation, specific pauses, and date and time formatting.

This paper will explore the elaboration, training, setup, and integration of an LLM, in particular the GPT2 (Radford et al., 2019), in public institutions, in the form of a chatbot capable of text-to-speech, which can assist citizens in different ways such as information regarding a procedure (How do I register a property acquired through a sale-purchase contract?) or a piece of particular information about the institution itself (What are the working hours for Tuesday?). The result of this experimental model can be further integrated into a future metaverse application.

1.2. Literature review

Generative AI having the ability to analyse vast amounts of data can uncover patterns and trends regarding bureaucrats. It can examine how public service workers interpret and respond to policy guidelines and it can identify factors that drive their decision-making processes. Furthermore, ChatGPT and Bard can simulate conversations between street-level bureaucrats and citizens, providing a unique opportunity to explore the dynamics of these interactions and their implications for policy implementation. These can advance the understanding of factors that influence public service workers (Salah et al., 2023).

Zheng et al. (2023) conducted an experiment using a BERT model with improvements such as embedded layer decomposition, cross-layer parameter sharing, and SOP sentence order prediction task. Also, the training was optimised using gradient accumulation and 16-bit precision training. The experiment focused on the impact of different batch sizes and accuracy, showing that the best effect is when the gradient is accumulated 2 or 3 times with very little difference between them (1 time has an accuracy of 0.794, 2 times has 0.815, 3 times has 0.816 and 4 times has 0.813). The highest accuracy obtained was 0.815, when the attenuation coefficient was 0.95, with a learning rate of 2*10-5 and 2 times gradient accumulation.

A question-answering system has been developed by Vijayan et al. (2024) based on BERT and unsupervised Smoothed Inverse Frequency (uSIF) in which the embeddings are sentence-based and not word-based. This produces an accuracy of around 80% with an EM (exact match) score of 54.28% and F1 of 85.53%. Furthermore, BERT+uSIF is capable of longer text sequences, compared to the base BERT model which has a maximum token restriction of 512.

In the paper of Andrews & Witteveen (2019), a small unsupervised AI model was developed for questions and answers, using the pre-trained GPT-2 with 117M parameters, 876k sentences from Wikipedia, and a scheme for incorporating “sentence hints” in context generation. Being in the early stages, the results were not promising, but when using the ‘hint sentences’, GPT-2-117M was able to use extra information without training.

2. Method

2.1. Development environment

In setting up this experiment, it has to be taken into account that it should be easily replicated by any public institution in Romania (Banciu et al., 2020). As such, it was decided to use readily available hardware and software, with an emphasis on ease of access and use for a casual user. To that end, it was decided to use a mid-tier hardware configuration that any public institution would have access to. An Intel NUC computer with an RTX 4070 graphics processor running Arch Linux had to be set up. Standard Python3 libraries were also used for large language model fine-tuning such as Transformers and DataLoader. The used models are readily available on the Huggingface platform, lowering the technical know-how requirements of the end-user.

2.2. Model selection

There are a variety of Large Language Models available, all with different properties, strengths and weaknesses. There are several factors which need to be taken into consideration. First of all, we need to acknowledge that not everyone has access to state-of-the-art hardware, in order to train their own advanced model. There are several publicly available models on the internet, most of which can be found on Huggingface. An appropriate model has to be small enough to fit on an average computer, yet powerful enough to understand nuances. It must also be able to understand the local language, in this case Romanian. A model which does not understand the local language is a lot harder to work with, although it is not impossible to make it work. This would require the user to do further training. However that is not easy and is beyond the scope of this paper. For the purposes of this article, a medium-sized model which was either trained on a Romanian dataset or understands Romanian language was required.

Three viable models which fulfil all the criteria that have been outlined before were designated. All three are variations of the GPT2 model which was trained on a Romanian dataset called RoGPT (Niculescu et al., 2021).

2.3. Gathering data

The present goal of creating a robust framework for training a large language model also involves gathering data. This data can be split into two sections. The first one involves technical or legal knowledge and relates to document requirements, legal procedures, or payment details. The second set of data is more customised and tailor-made for a specific institution. This can involve, but is not limited to, information on the physical location of the institution such as where the entrance is, working hours, public transportation options in the vicinity or parking availability. This also means that the people in charge of said institution can easily update their language model, in order to fit their way of conducting their business.

In the case of this project, a public institution whose activity ranges from simple payments to submitting stacks of documents was chosen, since it meets a lot of the criteria of the data types mentioned above. In order to create this dataset, several steps were followed. Firstly, all the legal documents were gathered from their website and a list of generic and basic questions was compiled based on them. The answers were designed to be short and to the point, in order to avoid adding unnecessary extra information. This was done by a two-person team, over the course of several days. First step was to look for an FAQ on the website, since these questions are the likeliest to come up. The answers from the website were not chosen, since these were quite detailed and included links to outside material, which would not be helpful to the target users. Furthermore, the intention was to include information, from said outside link, into the answers, not to redirect users. The second step was browsing public forums and message boards such as Reddit in order to find out what other users have asked or had trouble with. In this case, some of the answers were synthesised from other users' responses, since they were presenting their lived experience with these processes. Lastly, the final step was compiling a list of questions specific to the institution for which the chatbot is customised. This includes general information such as address, where to find their location, points of interest nearby, but also more specific details such as bank accounts to pay them or the fact that they can photocopy their documents at their location for free and don't need to pay a copying service for this.

In total, 221 questions were compiled, with varying answer lengths, although it was tried to keep them as brief as possible, without losing information. The gathered data has a total of 32137 tokens with 3137 unique tokens and a maximum token length of 18. A visual representation of the token length distribution can be seen in Figure 1.

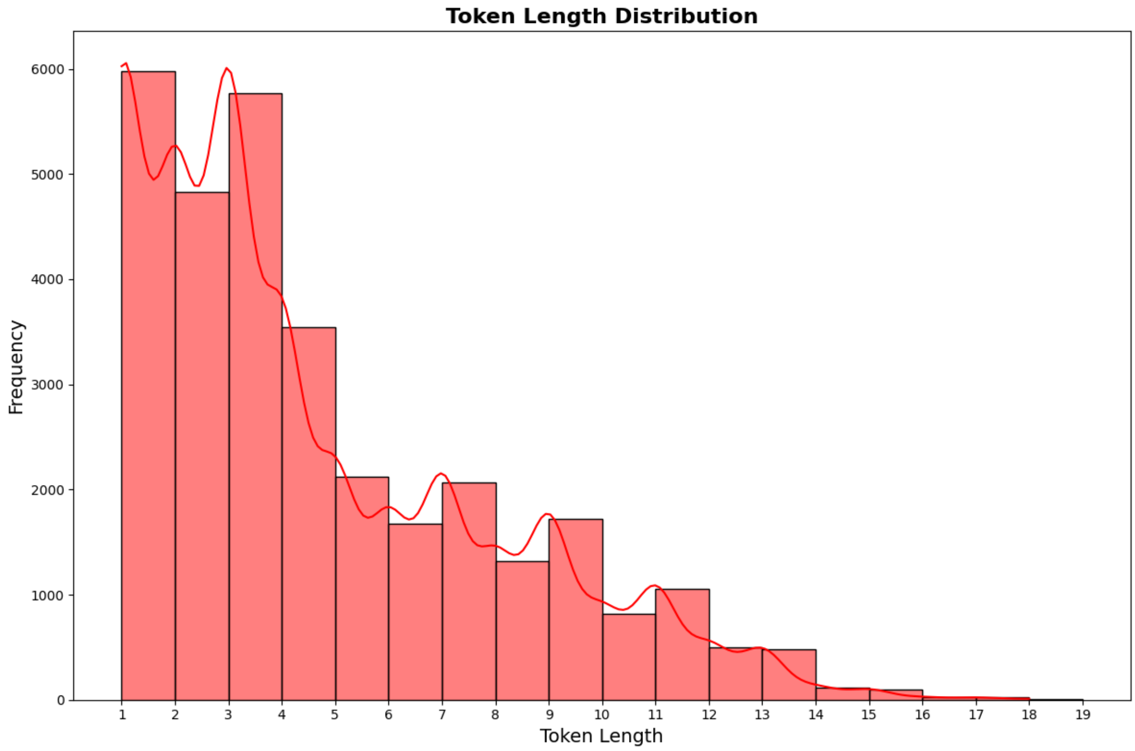

Figure 1. Token length distribution of the gathered data

2.4. Implementation

In order to make this experiment as easy as possible to reproduce for casual users who do not have programming knowledge, the dilemma of using a programming language or some other tool that could achieve the same result had to be faced with. On one hand, the Python programming language could have been used, as it has a number of advantages. First of all, it's easy to learn and write and the number of libraries for natural language processing and deep learning is staggering. It is also very popular among programmers, meaning that new users can easily find resources online on any related topic, if they have an inquiry or encounter a problem. The downside is that while easy, it is still asking of someone with little programming knowledge to learn a new skill, which could turn off institutions from adopting such a service.

The second solution was to use a program designed specifically for training machine learning algorithms. It would also have to be intuitive for new users to understand and have a friendly interface, so as to not frustrate anyone with cumbersome details. It was decided that Altair AI Studio is perfect for this endeavour. It has several features that are of great benefit to the present work and to any new potential user. First of all, there is little encoding involved when using it. It features a drag and drop interface where the user can select modules and link them up by clicking on them. Lastly, it features a marketplace of user created modules similar to how Python has a large collection of libraries. All the necessary modules to implement the present experiment were found from this marketplace.

In order to recreate it, there are three major steps that the user has to follow. First of all, the data has to be inserted in the programme. All the questions and answers should be in a file separated by a user selected symbol and the end of a pair of questions and answers should be separated by a newline. When adding the data in Altair AI Studio, the user must mention that the columns are separated by their chosen symbol and the rows are separated by the newline character. They should also make sure that there are no warnings with their data from the programme, as that could affect the performance of the algorithm further down the pipeline. However, these warnings are user specific and are based on the data being used, but through accumulated experience, the most common ones are choosing the wrong symbol to separate columns, having said separator symbol twice in a row, including characters that are unsupported and having a different number of questions to the number of answers.

After the data is uploaded, it cannot be used straight away to train the language model. This intermediary step between having the data ready and using it as intended to is called preprocessing. This preprocessing is a necessary step in order to shape the data in a way that the learning algorithm will extract the maximum information from it and perform as well as expected. This pipeline is shown in its entirety in Figure 2. The first step in any language processing task is to tokenize the text. This will split the text from one continuous block of information into a fragmented one. Some tokens are not necessary to include such as certain symbols or punctuation, so they will also have to be removed. Since all words have a root form, every single word in the text needs to be reduced to its root. For each language, the rules of stemming words will be different, so the user has to provide a dictionary for Altair AI Studio to know. This dictionary can be compiled in multiple ways. One way is to find one available online, although for some languages it can be rather difficult. Another way is to create a list of every single word used in the training data and stem it manually. Although a daunting task, this can be automated using a programming language such as Python and its powerful natural language toolkit libraries available. With the text stemmed, the next step in the preprocessing pipeline is to filter stop words. In this case, stop words are words that are common and bring no extra information to the training algorithm. These include prepositions, conjunctions, articles and verb auxiliaries. There are numerous online repositories where lists of these common stop words can be found and downloaded, such as Github or Bitbucket. The penultimate step in preprocessing the data is to transform the cases of the original data. While not mandatory to be the penultimate step, it was decided to perform it here, in order to emphasise that all the other steps need to be done in the order presented, whereas this one can be inserted anywhere in the pipeline and it would not affect the final result. What this step does is simply transform any word starting with an uppercase letter to a lower case. The final step is to generate n-grams of the tokens. In essence, this will transform the data from human readable to machine readable.

With the data preprocessed, the final step is to train the model. Figure 3 shows this process in detail. Note that the preprocessing step described above is contained within the “Process documents” module. Training the present model consists in two steps and those are downloading the model and fine tuning it. The “Download model” module downloads a pretrained model from Hugging face. The user needs only to provide the model card from the website and specify the type of task it is aiming to solve. They should also provide the target folder where the data can be found and a Conda environment which should be set up during the initial setup of Altair AI Studio. The second module named “Fine tune model” is the actual training. Here, the users will specify all the parameters that they need, in order to maximise their results. While some parameters are not necessary in order to successfully train the model, there are some that the user should pay attention to. First of all, if they have the means, they should make sure that they are using the GPU for training. They should also make sure that they select the right column and the input and select a number of epochs that will complete the training. By default, it is set to 5, but in most cases the user will want to increase this number by a factor of 3 or 4. The other parameters, while not crucial, can influence the outcome of the training. The three most important ones are the batch size, learning rate and train-test ratio. Batch size should be increased in case the user has multiple GPUs available on hand and decreased should the user not benefit from having computing resources. The learning rate determines how big the steps are when an epoch is completed and the test-train ration determines how many examples from the dataset will not be used in training, but rather in testing the outcome of the model.

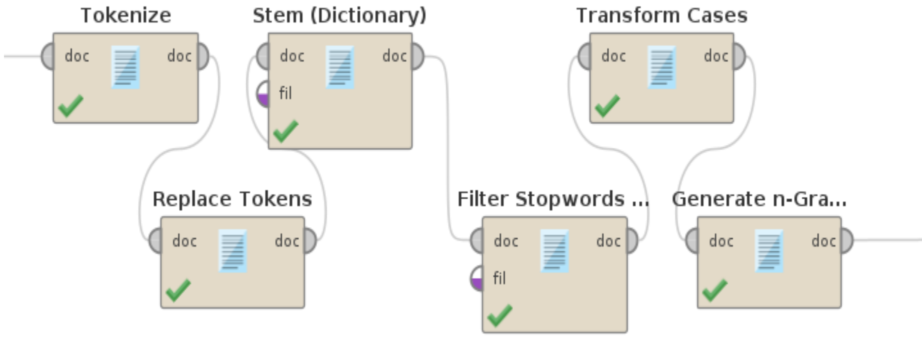

Figure 2. The preprocessing pipeline

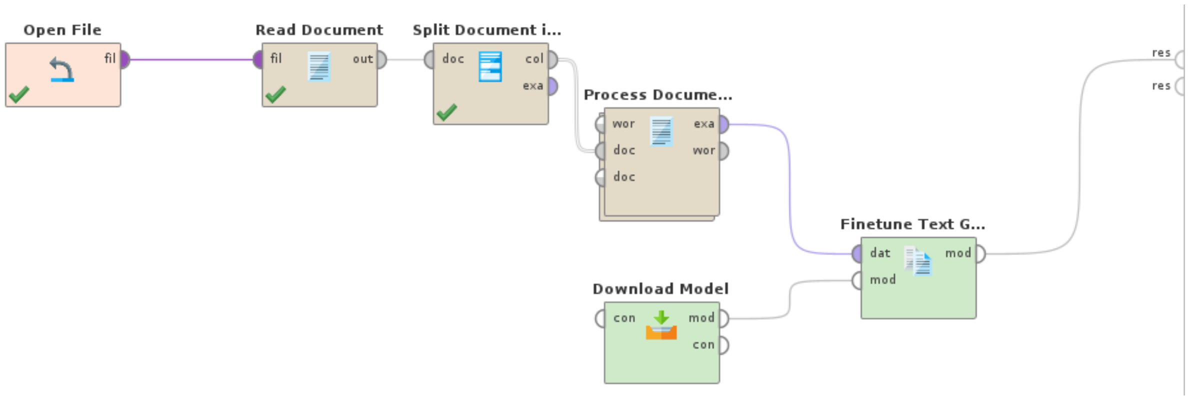

Figure 3. The complete training pipeline

3. Experiment results

It was decided to run the experiments on all three available RoGPT2 models in order to observe the differences in performance. One of the main obstacles was the disjointed nature of the information available to compose a comprehensive dataset that can encompass the whole range of issues that a public institution can encounter.

The experiment was set up by using publicly available data from a local public institution. In order to compile this dataset, their website was analysed and pairs of questions and answers were formed. This dataset was incomplete, as the in-depth knowledge required to capture all the nuances that such a public institution can have was missing. In its stead, the focus was on the procedures and required documents that a common citizen might need while interacting with the institution. Furthermore, the answers were made as accurate as possible, in order to easily convert them into a context, in case they need to be included in training the model. It was decided to test a model based on GPT2, as it is powerful enough to achieve the goal that was set out for it, but not so resource-heavy as to require a powerful machine that may be outside the budget range of any institution.

The three models were trained with the same parameters across the board, in order to get a fair comparative assessment of their performance. It was decided to train the model for 10 epochs. Following the training of the models, it was noticed that, for all three of them, the results were very similar, when plotting out the training and testing loss, as shown in Figure 4. While using the model, there are two important parameters that need to be kept in mind. First of all, it is the model's temperature. This parameter sets the creativity of the model when used. It was set lower, so that the model is not necessarily a creative one, but rather gives out concrete information about the user's inquiry. Setting it higher would mean that the model has more creative freedom, thus leading to hallucinations and the model inventing new types of documents and procedures. The second parameter is the answer's length. It is not intended for the model to tell a long story about relatively simple tasks, but rather to keep the answers short and to the point. However, an increase in the time required to train the model was observed, as the number of parameters increased, as shown in Table 1.

Table 1. Train results for three types of GPT2 using 32137 tokens

	RoGPT2-base	RoGPT2-medium	RoGPT2-large
Total parameters	124M	354M	774M
Train runtime	426.6665	1099.3313	2346.0604
Train samples/second	37.781	14.663	6.871
Train steps/second	1.195	0.464	0.217
Final train loss	1.7348909705292945	1.5284638862983853	1.4733923790501613

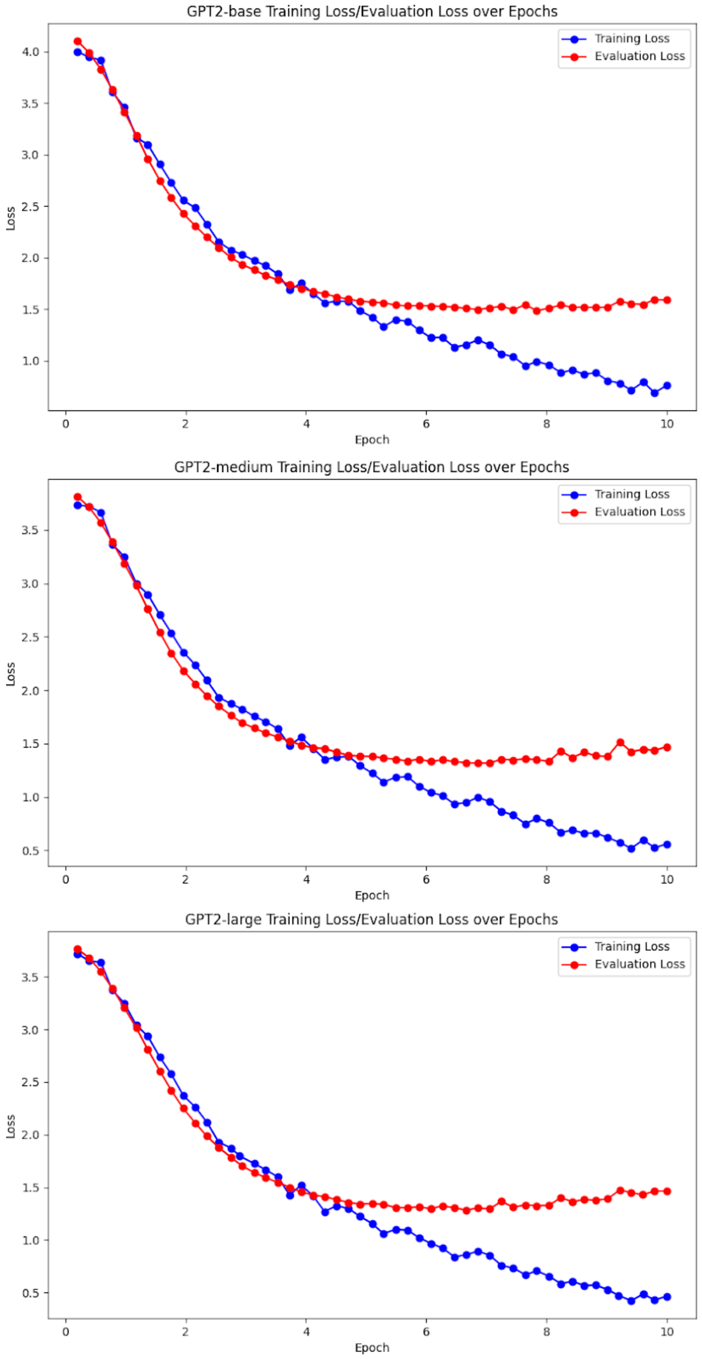

Figure 4. Train loss and evaluation loss for GPT2-base, medium and large

4. Discussion and future work

This small-scale experiment was used to show that it is indeed possible for a language model trained by people with limited resources to be of great use. In order to bring this from the experimental stage to a workable framework, there are several steps ahead that need to be undertaken. First of all, not all public institutions have the same intricate systems, as some are more straightforward when dealing with the public, whereas some are significantly more intricate. Further experimentation is also required with different models. The present work identified two models which could fit the tasks set. First of all, there is RoLlama (Masala et al., 2024), a Romanian version of Llama2 developed by Facebook. The second model which could be useful is RoBer (Ceausu & Nisioi, 2022), a version of the Bert model also, trained on a Romanian dataset, by using legal documents. A qualitative comparison between the present base model, a more powerful model that needs more memory to train, and a model with knowledge of the legal system would provide greater insight into the direction this project should take.

Furthermore, while it is easy to train such a model, it can still be intimidating for someone with no programming knowledge. As such, there needs to be a centralised online platform dedicated to training a large language model specifically designed for this task where users can just upload the training data and select the model they need, with the whole training process being handled by the backend (Nicolau & Barbu, 2014).

A well-known issue is that the training data can quickly become outdated, as the legislation can change and the information that it was trained on cannot be "untrained". This "undo" process can be implemented by saving the models' weights at certain points. Eldan & Russinovich (2023) presented a fine-tuned technique for achieving unlearning, by using a reinforced model trained on the targeted data, in order to identify the tokens that are most related to the targeted data, by comparing it's-logits with the base model ones. The idiosyncratic expressions were then replaced with generic counterparts and alternative labels were generated for every token which approximated the next token prediction. Then it could be fine-tuned with these alternative labels that effectively achieved the unlearning.

This experiment is only a small part of what could be a larger metaverse application designed for public administration. While important in defining the interaction between a human operator and a computer, it is not the only factor. There are further considerations when designing this type of application, such as where it will be accessed from, if it will be a web page or a programme, if it can be accessed from a phone or if specialised equipment is needed to access it. These are all questions that need to be considered further when developing beyond this experimental chatbot.

Further work can be done by creating tools that can simplify the process even further. There are several steps in the presented training pipeline, where we had to write our own code in order to compile a list. While these tasks were not impossible to achieve without any additional code, it would have been a tedious task and, as a result, it can be concluded that it’s tedious for anyone that would have to do it. The first such programme that would help anyone is the stemmer and the common stop words compiler. The end goal would be a complete software in which the users introduce the data and it automatically deploys the chatbot, cutting out the middle part there the users have to do the training themselves.

5. Conclusion

While this experiment may seem simple, it is important to remember that it is meant for people with next to zero experience in coding and machine learning to be able to replicate it. The goal is for them to be able to create such a model themselves with minimal outside help, whether it’s from an experienced coder or the internet. This is only a small part of a larger theoretical metaverse application in which each user can introduce their own chatbot. Streamlining this process and empowering individuals within public institutions to do this themselves accelerate digital transformation, while minimising costs. Moreover, it is important for institutions to be aware of the fact that there isn't one dataset that will train a universal model that will fit all, but rather they will have to create a customised one for their needs. It is of paramount importance for the dataset to be compiled by a person with domain-specific knowledge, in order to cover all the nuances that differentiate each public institution from one another.

Acknowledgements

The work presented in this paper was supported by the Core Program within the National Research Development and Innovation Plan 2022-2027, financed by the Ministry of Research, Innovation and Digitalization of Romania, project no. 23380601.

REFERENCES

Andrews M. & Witteveen S. (2019) Unsupervised Natural Question Answering with a Small Model. In: *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), 3 November 2019, Hong Kong*. Association for Computational Linguistics. pp. 34-38.

Banciu, D., Radoi, M. & Belloiu, S. (2020) Information Security Awareness in Romanian Public Administration: An Exploratory Case Study. *Studies in Informatics and Control.* 29(1), 121-129. doi: 10.24846/v29i1y202012.

Barbu, M., Vevera, A. V. & Barbu, D. C. (2024) Standardization and Interoperability – Key Elements of Digital Transformation. In: Cioca, L. I., Ivascu, L., Filip, F. G. & Doina, B. (eds.) *Digital Transformation. Intelligent Systems Reference Library*. vol 253. Cham, Springer, pp. 87-94. doi: 10.1007/978-3-031-55952-5_5.

Ceausu, C. & Nisioi, S. (2022) Identifying Draft Bills Impacting Existing Legislation: a Case Study on Romanian. In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022), 20-25 June 2022, Marseille, France*. European Language Resources Association. pp. 3670-3674 .

Eldan, R. & Russinovich, M. (2023) Who's Harry Potter? Approximate unlearning in LLMs. *arXiv*. [Preprint] https://arxiv.org/abs/2310.02238

Floroiu, I. & Timisică, D. (2024) A Heideggerian analysis of generative pretrained transformer models. *Romanian Journal of Information Technology and Automatic Control [Revista Română de Informatică și Automatică]*. 34(1), 13-22. doi: 10.33436/v34i1y202402.

Marinescu, I. M. & Iordache, D. D. (2023) Exploring relevant technologies for simulating user interaction in Metaverse virtual spaces. *Romanian Journal of Information Technology and Automatic Control [Revista Română de Informatică și Automatică]*. 33(3), 129-142. doi: 10.33436/v33i3y202310.

Masala, M., Ilie-Ablachim, D. C., Dima, A., Corlatescu, D., Zavelca, M., Olaru, O., Terian, S., Terian, A., Leordeanu, M., Velicu, H., Popescu, M., Dascalu, M. & Rebedea, T. (2024) "Vorbeşti Româneşte?" A Recipe to Train Powerful Romanian LLMs with English Instructions. *arXiv*. [Preprint] https://arxiv.org/abs/2406.18266

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N. & Mian, A. (2023) A comprehensive overview of large language models. *arXiv*. [Preprint] https://arxiv.org/abs/2307.06435

Nicolau, D. & Barbu, D. (2014) Reflections on Moving and Aggregating in Cloud of the Databases Belonging to Public Administration. *Romanian Journal of Information Technology and Automatic Control [Revista Română de Informatică și Automatică]*. 24(2), 43-48.

Niculescu, M., Ruseti, S. & Dascalu, M. (2021) RoGPT2: Romanian GPT2 for Text Generation. In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), 01-03 November 2021, Washington, DC, USA*. IEEE. pp. 1154-1161. doi: 10.1109/ICTAI52525.2021.00183.

Petre, I., Iordache, D., Zamfir, M., Barbu, M., Duţescu, R. & Marinescu, I. A. (2023) Virtual worlds, real technologies: an insight into Metaverse and its principles. In: *Proceedings of the 2023 24th International Conference on Control Systems and Computer Science (CSCS), 24-26 May 2023, Bucharest, Romania*. IEEE. pp. 551-556. doi: 10.1109/CSCS59211.2023.00093.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. (2019) Language models are unsupervised multitask learners. *OpenAI blog*. 1(8), 9.

Salah, M., Abdelfattah, F. & Al Halbusi, H. (2023) Generative Artificial Intelligence (ChatGPT & Bard) in Public Administration Research: A Double-Edged Sword for Street-Level Bureaucracy Studies. *International Journal of Public Administration*. 1–7. doi: 10.1080/01900692.2023.2274801.

Vijayan, R. V., M., Prince, E. & Kumar, C. (2024) Improving the BERT model for long text sequences in question answering domain. *International Journal of Advances in Applied Sciences*. 13(1), 106-115. doi: 10.11591/ijaas.v13.i1.pp106-115.

Zheng, W., Lu, S., Cai, Z., Wang, R., Wang, L. & Yin, L. (2023) PAL-BERT: an improved question answering model. *Computer Modeling in Engineering & Sciences*. 139(3), 2729-2745. doi: 10.32604/cmes.2023.046692.

Mihai APOSTOL graduated from the Faculty of Applied Sciences within The National University of Science and Technology POLITEHNICA Bucharest, with an engineering degree, and pursued a master's degree in the same institution, specialising in cyber security. The main areas of interest and activity are the development of web applications, web technologies and cyber security. Currently, he is working at the National Institute for Research and Development in Informatics – ICI Bucharest, as a Researcher within the Digital Transformation Department.

Mihai APOSTOL a absolvit Facultatea de Științe Aplicate din cadrul Universității Naționale de Știință și Tehnologie POLITEHNICA București, cu diplomă de inginer, și a urmat un program de master în cadrul aceleiași instituții, cu specializare în securitate cibernetică. Principalele sale domenii de interes și de activitate sunt dezvoltarea de aplicații web, tehnologii web și securitate cibernetică. În prezent, lucrează la Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București, având funcția de cercetător științific în cadrul Departamentului de Transformare Digitală.

Răzvan-Alexandru DUȚESCU graduated from the University of Manchester (United Kingdom), with a bachelor's degree in Computer Science, and pursued a master's degree in Artificial Intelligence, at the University of Aberdeen (United Kingdom), in 2019. His main areas of expertise include image processing and unsupervised machine learning. He is currently working at the National Institute for Research & Development in Informatics – ICI Bucharest, as a Researcher within the Digital Transformation Department.

Răzvan-Alexandru DUȚESCU a absolvit Universitatea din Manchester (Regatul Unit al Marii Britanii și Irlandei de Nord) cu o diplomă de licență în Informatică și a urmat un program de master în Inteligența Artificială în cadrul Universității din Aberdeen (Regatul Unit al Marii Britanii și Irlandei de Nord), în 2019. Principalele sale arii de expertiză includ procesarea de imagini și învățarea automată nesupervizată. În prezent, lucrează la Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București, având funcția de cercetător științific în cadrul Departamentului de Transformare Digitală.

Monica BARBU is Scientific Researcher III and Head of Resilience, Interoperability and Standardisation Department at National Institute for Research and Development in Informatics – ICI Bucharest. She graduated from the Bucharest University of Economic Studies – the Faculty of Administration and Public Management, she holds a master's degree in Information and Document Management from the University of Bucharest – the Faculty of Letters and a master's degree in International Relations and European Integration from the National School of Political and Administrative Sciences in Bucharest – Department for International Relations. She has a Ph.D. in Management from "Lucian Blaga" University in Sibiu – Faculty of Economic Studies. Her main research interests are: the interoperability of informatic systems, e-Services, e-Government and e-Learning.

Monica BARBU este cercetător științific gradul III și Șef al Serviciului Reziliență, Interoperabilitate și Standardizare din cadrul Institutului Național de Cercetare-Dezvoltare în Informatică – ICI București. A absolvit Facultatea de Administrație și Management Public din cadrul Academiei de Studii Economice din București, deține diplomă de master în Managementul Informațiilor și Documentelor din cadrul Facultății de Litere, Universitatea din București și diplomă de master în Relații Internaționale și Integrare Europeană din cadrul Școlii Naționale de Științe Politice și Administrative din București, Departamentul pentru Relații Internaționale. A obținut titlul de doctor în Management la Facultatea de Studii Economice din cadrul Universității „Lucian Blaga" din Sibiu. Principalele sale domenii de interes pentru activitatea de cercetare sunt: interoperabilitatea sistemelor informatice, servicii publice electronice, e-Guvernare și e-Learning.