

Платформа за локално обучение на големи езикови модели

С. Стойкова

Platform for Local Fine-Tuning of Large Language Models

S. Stoykova

Department of Control Systems, Faculty of Electronics and Automation, Technical University of Sofia, Plovdiv Branch, 25 Tsanko Dyustabanov St., 4000 Plovdiv, Bulgaria, stoykova@tu-plovdiv.bg

Key Words: Large language models; management information systems; artificial intelligence; machine learning; local fine-tuning.

Abstract. The study presents a platform designed for local training and inference of popular large language models (LLMs), addressing the growing need for solutions that preserve the privacy of training data. The platform enables researchers and developers to further fine-tune base models using locally stored and processed data through a modular software architecture. It supports a wide range of models and tasks, offering seamless integration with existing machine learning workflows and tools within enterprise management information systems. Two alternative approaches to contextualization of LLMs' knowledge bases are implemented within the platform i.e., indirectly through model tuning (parameter-efficient fine-tuning, reinforcement learning with human feedback and tutorial fine-tuning) and directly through extending the original knowledge base via retrieval-augmented generation. The proposed hybrid solution for knowledge contextualization addresses existing challenges within the generative artificial intelligence field, namely accuracy of generated information (hallucinations), as well as transparency and traceability of generated information to source data. Benchmark results from the platform utilization demonstrate competitive performance in terms of latency, context accuracy, and resource utilization compared to cloud-based alternatives. By decentralizing model development and inference, the platform empowers institutions to maintain data sovereignty, reduce operational costs, and promote sustainable practices in the field of artificial intelligence.

Въведение

Осъществено литературно проучване по темата “Приложения на изкуствен интелект в информационните системи за управление” [1] показва, че бизнес организациите генерират стойност чрез интегрирането на приложения, базирани на изкуствен интелект в три основни направления.

- **Автоматизация на процесите**, т.е. стойността, генерирана поради автоматизацията на бизнес процесите и прехвърлянето на изпълнението на задачи с ниска добавена стойност от служители към интелигентни инструменти, базирани на ИИ.

- **Интелигентен бизнес и оперативен анализ**, т.е. стойността, генерирана благодарение на анализ на бизнес информация и данни, базиран на методи и техники от областта на машинното обучение.
- **Когнитивно взаимодействие**, т.е. стойността, генерирана поради взаимодействия на служители или клиенти с дигитални асистенти, чатботове и базирани на генеративен изкуствен интелект инструменти и приложения.

В идентифицираните основни литературни източници преобладава внедряването на приложения, базирани на ИИ в областта на автоматизацията на процеси, като по-малко от половината от разгледаните изследвания се концентрират върху извличането на стойност чрез интегриране на ИИ в направленията на бизнес оперативен анализ и когнитивно взаимодействие. Това показва необходимостта от допълнително изследване на приложимостта на ИИ в тези две направления.

Задачите, основаващи се на когнитивно взаимодействие, представляват значителен дял от ежедневните операции, осъществявани в информационните системи за управление в индустрията, което ги превръща във фокус на редица научни изследвания и разработки както в държавния, така и в частния сектор [2, 3]. Тези задачи включват динамичен обмен на информация между потребители/служители и машинен интерфейс, софтуерни платформи и интелигентни дигитални агенти за целите на извличане на познание, систематизиране и обобщаване на информация, генериране на съдържание и подпомагане на процеса по вземане на решения. Това означава, че е необходим интердисциплинарен подход в разработването на модели и изследването на възможни решения в областта.

Въпреки че взаимодействията човек-машина не са нова област на изследване, а всъщност са обект на засилен научноизследователски интерес още от зората на компютърните технологии и информационните науки, развитието на тази област е повлияно драстично през последните няколко години от приоритизираното разра-

ботване и масово интегриране на технологии, базирани на методите на изкуствен интелект, специфично тези с генеративен характер като големите езикови модели (Large Language Models, LLMs).

Масовата комерсиализация на големи езикови модели (LLMs) като ChatGPT, Copilot, Gemini в периода от 2022 г. до момента доведе до преориентиране на редица дистрибутори на интегрирани информационни системи за управление към внедряването на т. нар. интелигентни дигитални агенти (Joule за SAP, Copilot за Dynamics 365, Oracle AI Assistant), базирани на LLM модели към облачните им платформи. Прилагането на LLM базирани решения носи редица предизвикателства по отношение на точност на генерираното съдържание и конфиденциалност на обработваните данни. Отсъствието на прозрачност (explainability) на компоненти, базирани на машинно обучение, и уязвимостта на приложения, базирани на генеративен ИИ, към кибератаки поради голямата свързаност и отвореност на базовите модели (LLMs) се явяват сериозни предизвикателства пред успешното интегриране на технологиите, базирани на изкуствен интелект, в съществуващите системи за индустриална автоматизация [4]. В този контекст, осигуряването на прозрачност, проследимост и обяснимост в процесите на последващо обучение (fine-tuning) и инференция на големи езикови модели се превръщат в основни цели при разработката на приложения, базирани на генеративен изкуствен интелект и при тяхното внедряване към съществуващи информационни системи за управление в индустрията. Основно предизвикателство пред успешното внедряване на големи езикови модели в информационните системи за управление в индустрията е осъществяването на балансирано контекстуализиране на базата знания на модела към съответната приложна област, а в някои случаи към конкретна задача, изискваща специализирано познание [5]. Съществуват различни техники за контекстуализация в рамките на последващото обучение на такива модели:

- **SFT (Supervised Fine-Tuning).** Основният принцип на SFT е използването на общите езикови представяния, усвоени от предварително обучение базов модел, в комбинация с последващо обучение на неговите параметри върху специализиран за съответната приложна област набор от анотирани данни. При техники за контекстуализация на знание от този вид етапът на последващо обучение разчита на наличието на анотирани данни за обучение, т.е. осъществява се обучение с учител, което ограничава приложимостта на такива подходи до задачи, за които съществуват или могат да бъдат съставени анотирани обучителни бази данни. Към този клас от техники за контекстуализация принадлежат FFT (Full Fine-Tuning) и PEFT (Parameter-Efficient Fine-Tuning), като разликата е в обхвата на обучението: FFT осъществява актуализация на теглата на всички параметри на модела в процеса на обучение, а PEFT

само на малък поднабор от параметри на модела в базовата му конфигурация или в допълнително добавени в архитектурата му слоеве. Макар и да е изчерпателен FFT подходът е изчислително скъп и податлив на свръхнапасване, особено когато анотирани данни са ограничени [6]. PEFT [7], включително негови варианти като LoRA [8] и Adapter [7], оптимизира само малък поднабор от параметри, намалявайки изчислителните разходи и подобрявайки ефективността на обучението без компромис в производителността.

- **RLHF (Reinforcement Learning with Human Feedback).** Подход за контекстуализация, който разчита на обратна връзка от потребителя в процеса на генериране на съдържание по отношение на оценяване на качеството на това съдържание спрямо поставената заявка. В този случай оценката на потребителите се използва като инструмент за подобряване на безопасността и управляемостта на генерираното съдържание. Необходимо е да се подчертае, че такива подходи са зависими от съвместното оценяване от страна на потребителите [9].
- **TFT (Tutorial Fine-Tuning).** Включва използването на самия целеви модел едновременно като обучаем и като оценител, като по този начин той функционира като „учител“ в собственото си обучение. Този подход позволява на моделите да генерират собствени данни за обучение (т. нар. симулационни данни), намалявайки зависимостта на процеса на обучение от наличието на набори с анотирани данни [10]. Рискът при такава реализация е в отдалечаването на симулационните данни от условията на решаваната специализирана задача и като резултат влошаване на качеството на генерираното съдържание (например халюцинации, конфабулации).

Алтернатива в областта на контекстуализиране на познание за целите на разширяване на способността на големи езикови модели да генерират по-точно съдържание е т. нар. генериране с добавено извеждане (Retrieval-Augmented Generation, RAG). Това е подход, при който съдържанието, генерирано от голям езиков модел, се позовава на данни, извлечени от външни за модела източници като файлове, бази данни, уеб страници. За разлика от техниките за контекстуализация в процеса на последващо обучение (SFT, RLFT, TFT), които въздействат върху големия езиков модел, актуализирайки теглата на всички или част от неговите параметри, без пряко да разширяват неговата основна база знания, генерирането с добавено извеждане отваря процеса на генериране на съдържание към външни за модела източници на знание. Този подход за контекстуализиране на познание има способността директно да въздейства върху един от основните проблеми при приложението на генеративен изкуствен интелект: неточност или неактуалност в ге-

нерираното съдържание (халюцинации, конфабулации), като оценява генерираното съдържание на базата съответствие на потвърждаваща информация във външни за основополагащия модел бази знания.

Настоящото изследване има за цел да представи разработена платформа, предназначена за локално последващо обучение (fine-tuning) и инференция на популярни големи езикови модели (LLMs), позволяваща комбиниране на подходи за контекстуализация в процеса на последващо обучение (PEFT, RLHF и TTF) на големите езикови модели с алтернативен подход за контекстуализация чрез разширяване на базата знания на модела с външни източници (RAG). Този хибриден подход на контекстуализация на знание отличава разработената платформа от съществуващи платформи за последващо обучение на големи езикови модели като Axolotl, LLaMA-Factory, Unsloth, които не поддържат генериране с добавено извеждане (RAG). Предложеният хибриден подход на контекстуализация на познание, благодарение на RAG, представя адекватна компенсация на основни слабости на приложенията, базирани на големи езикови модели като прозрачност и проследимост на истинността на генерираното съдържание.

Платформата е разработена в отговор на нарастващата нужда от решения, които да съхраняват поверителността и конфиденциалността на използваните за обучение данни и на генерираното в резултат на инференция съдържание. Тя позволява на изследователи и разработчици да осъществяват допълнително обучение на базовите модели върху локално съхранявани и обработвани данни посредством модулна софтуерна архитектура. Платформата поддържа широк спектър от базови модели и задачи, като предлага улеснена интеграция със съществуващи в информационните системи за управление в индустрията работни процеси и инструменти за събиране и обработка на данни.

Разработената платформа Private Edge LLMs:

- Позволява обучение с локален ресурс (локални модели: llama, deepseek-R1, gemma с наличния изчислителен ресурс на периферното устройство: компютър, лаптоп) и локално съхранение и обработка на данните. Подход, който е предпочитан при задачи, изискващи конфиденциалност на данните за обучение, подходящ за приложения в научноизследователски институции.
- Позволява връзка с API към нелокални изчислителни ресурси (GPU, TPU масиви за обучение, достъпни в облака) и/или големи езикови модели в облака (GPT).
- Внедрява функция за генериране с добавено извеждане (Retrieval-Augmented Generation, RAG), която подобрява възможностите на големите езикови модели (LLM), като ги комбинира с външни за модела източници на знания, предоставяйки допълнителен контекст за изпълнение на промпта (тези източници могат да бъдат локални за периферното устройство, на което се

осъществява допълнителното обучение или споделени в облака). Тази функция адресира някои от ограниченията на LLM, като например остаряването на информацията, използвана в първоначалното обучение и помага за редуцирането на т. нар. халюцинации (генериране на неточна информация), като позволява на големите езикови модели да имат достъп и да включват информация от конкретни документи или бази данни при генериране на отговори.

- Предоставя подобрение в проследимостта на източника, използван за генериране на текстово съдържание чрез функция цитиране на източник.

За да бъде осъществено едновременно внедряване на множество големи езикови модели и да бъде управлявано тяхното последващо обучение върху локални ресурси, е необходимо да бъде изградена локална софтуерна платформа, основаваща се на модулна софтуерна архитектура, която да дава необходимата гъвкавост и адаптивност към разнообразието от архитектури и подходи за обучение на големи езикови модели.

Модулна софтуерна архитектура

Модулната софтуерна архитектура, интегрирана в разработената платформа Private Edge LLMs, включва следните елементи:

- **Потребителски интерфейс на приложния слой.** Позволява селекция на базовия голям езиков модел, който ще бъде обучаван от страна на потребителя; селекция на източници на данни за допълнителното обучение и селекция на режима на работа: RAG (генериране на съдържание с добавено извеждане), Search (контекстово генериране на съдържание, където се извеждат като резултат само четирите най-релевантни за потребителската заявка информационни блока) и Basic (неконтекстова инференция на дадения голям езиков модел). Имплементирането на три режима на работа насърчава осъществяването на контрол от страна на потребителя на платформата върху процеса на контекстуализиране на познание на големите езикови модели. Потребителският интерфейс е изграден на базата на отворената библиотека на Python за създаване на уеб интерфейси за модели, базирани на машинно обучение, Gradio. Библиотеката Gradio е предпочетена като основа за изграждане на потребителския интерфейс заради улесненото интегриране на стандартни библиотеки за машинно обучение и обработка на данни в Python, което позволява разширяване и надграждане на разработената платформа с цел адаптирането и внедряването ѝ в работните потоци за обработка на данни на съществуващи информационни системи за управление.

- **Модул за оркестрация на задачи спрямо изборения режим на работа.** Модулът за оркестрация е реализиран на базата на оркестрационната рамка LlamaIndex, която осъществява управление на процесите по обработка на външни за модела източници на знания, индексирание на обработената информация, извличане на информационни блокове на база контекст и генериране на промпт (допълнена с контекст потребителска заявка). Оркестрационната рамка LlamaIndex е избрана заради нейната гъвкавост по отношение на използването на различни модели за векторизация и възможността за интегриране на векторни бази данни с отворен код.
- **Модул за планиране на етапи на изпълнение на заявката.** Задава последователността на изпълнение на потребителската заявка въз основа на шаблон за поведение, отговарящ на изборения режим на работа: RAG и Search режимите разчитат на допълнително контекстуализиране на потребителската заявка, докато Basic режимът позволява инференция на оригиналния голям езиков модел без контекст от външна за модела база знания.
- **Модул за контекстуализиране на потребителските заявки.** Подготвя външната за големия езиков модел база знания (може да бъде локална или разположена в облака). Изходните данни, представени в различни формати (.pdf, .docx, .csv, .pptx, .ipynb, .json, .epub), преминават през процеси на почистване и извличане на текстово съдържание. Полученият текст се сегментира на информационни блокове на база контекст (context chunks), които се преобразуват във векторни представяния (embeddings) с помощта на предварително обучен модел и се съхраняват в динамичната памет.
- **Динамична памет.** Концепция, която надгражда традиционната архитектура на големите езикови модели (LLMs), като им позволява да запомнят потребителски предпочитания и контекст от предишни разговори, да адаптират поведението си спрямо натрупаната информация, да извличат релевантни спрямо потребителската заявка данни от външни или вътрешни хранилища при нужда. Тази функционалност е особено значима за приложения като дигитални асистенти, персонализирани чатботове и други системи, които трябва да поддържат дългосрочна и последователна комуникация. Динамичната памет за разработваната платформа е реализирана посредством векторна база данни Qdrant. Qdrant е високопроизводителна векторна база данни с отворен код, предназначена за семантично търсене в приложения с изкуствен интелект. Тя е оптимизирана за работа с векторни представяния (embeddings), които се използват широко в системи за търсене и генериране с до-

бавено извеждане (RAG). Qdrant е предпочетена пред налични алтернативи като Chroma (само за локално интегриране на RAG) и Pinecone (само за облачно внедряване) заради възможността за комбинирано индексирание на локални ресурси и ресурси в облака. Това позволява платформата да бъде адаптирана към условията на имплементация на информационната система за управление на отделно предприятие или научноизследователска институция, без да е необходимо преструктуриране на самата платформа с цел преодоляване на ограниченията, наложени от вида реализация на системата (в облака, с локален център за обработка и съхранение на данни или хибридно). В допълнение Qdrant предлага нативно хибридно търсене (по вектори и ключови думи), т.е. за осъществяване на търсене за векторни сходства в базата данни не е необходимо интегриране на външна търсачка, както е при популярни алтернативни решения като Weaviate и Milvus. Независимостта на Qdrant от външни услуги за търсене и индексирание опростява процеса на внедряване в съществуващата архитектура за обработка на данни на дадена информационна система за управление и елиминира натрупването на допълнителни разходи за утилизация на външни услуги за индексирание. Допълнително преимущество на избраната векторна база данни Qdrant е ниската предвидима латентност при векторно търсене, която е от критично значение за приложения с генериране с добавено извеждане. Ниската латентност се дължи на прилагането на оптимизиран HNSW (Hierarchical Navigable Small World) алгоритъм за векторно търсене като част от самата Qdrant архитектура, а не като външен за нея елемент, в комбинация с асоцииране на векторни представяния със съответстващите им адресни пространства от паметта (аналогично на memory mapped storage за файлове).

- **Модул за оценяване.** Определя релеванността на индексирания контекстово сегментирани информационни блокове спрямо поставената потребителска заявка на базата на сходство между векторните им представяния, т.е. извършва търсене във векторната база, изчислявайки метрика на сходство (например косинусова близост или скаларно произведение) между вектора на потребителската заявка и векторите на индексирания контекстово сегментирани информационни блокове (context chunks).
- **Модул за генериране на отговор.** Формира допълнена с контекст потребителска заявка (контекстов промпт) за големия езиков модел. Този промпт обикновено включва оригиналната потребителска заявка и текстовото съдържание на извлечените в предната стъпка най-релевантни контекстово сегментирани информационни блокове.

- **Промпт мениджър.** Подава формулирания от модела за генериране на отговор контекстов промпт на LLM доставчика.
- **LLM доставчик.** Поддържа локална база с големи езикови модели, както и интеграция с частни или публични модели, реализирани в облака посредством API. За управление на зависимости, параметри и инсталационни пакети на големите езикови модели е използван Poetry (съвременен инструмент за управление на зависимости и пакиране на Python проекти, който улеснява създаването, изграждането и публикуването на Python библиотеки и приложения). Управлението на пакети и зависимости е от значение за поддръжката и надграждането на платформата във всички етапи от нейния жизнен цикъл.

На *фиг. 1* е представена обобщената структура на разработената платформа за допълнително обучение на големи езикови модели.



Фиг. 1. Модулна архитектура на разработената платформа за допълнително обучение на големи езикови модели

Внедряване на алтернативна схема за контекстуализация на знание RAG

Генериране с добавено извеждане (RAG) е подход, при който съдържанието, генерирано от голям езиков модел, се позовава на данни, извлечени от външни за модела източници с цел подобряване на точността и проследимостта на източниците на информацията, генерирана от такива модели. Схемата за обучение RAG осъществява допълнително обучение в два основни етапа: първо се извличат релевантни документи или техни фрагменти от външна за големия езиков модел база знания въз основа на потребителската заявка; след това тази информация се комбинира със специални контекстови подсказки (т. нар контекстови векторни съответствия), които насочват модела как да използва данните, за да се формира контекстът за генериране на крайния

отговор.

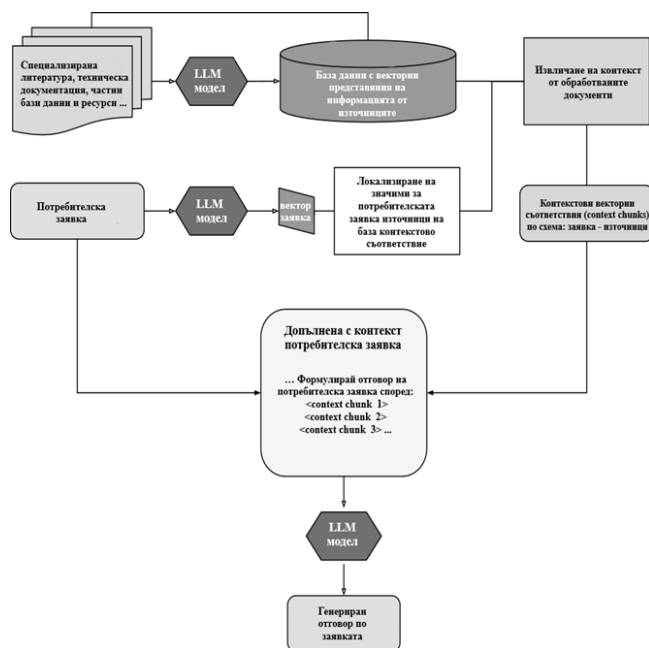
В зависимост от инструкциите в контекстовите подсказки генерираният отговор може да включва цитати или препратки към оригиналните документи, което повишава прозрачността и доверието на потребителите, като им позволява да проверят информацията. RAG помага да се преодолеят ограниченията на LLM като остаряла информация, неточности или халюцинации. Тази схема за обучение разширява базата знания на LLM до практически неограничени размери, осигурявайки бърз достъп до специализирани области или вътрешни данни на организации, научноизследователски институции, университетски библиотеки и т.н.

Функционалните стадии на обработка на информация при внедряване на архитектура за обучение тип RAG са:

- **Индексиране (Indexing).** На този етап се подготвя външната за големия езиков модел база знания. Изходните данни, представени в различни формати, преминават през процеси на почистване и извличане на текстово съдържание. Полученият текст се сегментира на информационни блокове на база контекст (context chunks), които се преобразуват във векторни представяния (embeddings) с помощта на предварително обучен модел. Генерираните вектори, заедно със съответните текстови фрагменти, се индексират и съхраняват в специализирана векторна база данни, оптимизирана за търсене по сходство.
- **Извличане (Retrieval).** При получаване на потребителска заявка, тя се векторизира с помощта на същия модел, използван при индексирането за създаване на векторните представяния (embeddings) на контекстово сегментирани информационни блокове (context chunks). След това системата извършва търсене във векторната база, изчислявайки метрика на сходство (например косинусова близост или скаларно произведение) между вектора на потребителската заявка и векторите на индексираните контекстово сегментирани информационни блокове (context chunks). Извлича се предварително определен брой (K) най-релевантни контекстово сегментирани информационни блокове, които имат най-висока степен на сходство.
- **Генериране (Generation).** В последния етап се формира допълнена с контекст потребителска заявка (контекстов промпт, модифицирана потребителска заявка) за големия езиков модел. Този промпт обикновено включва оригиналната потребителска заявка и текстовото съдържание на извлечените в предната стъпка най-релевантни контекстово сегментирани информационни блокове. LLM моделът обработва този контекстов промпт и генерира финален отговор на потребителската заявка. По време на генерирането моделът може да използва както информацията от предоставе-

ния контекст, така и своите вътрешни параметрични знания, или да бъде инструктиран да се придържа стриктно само към извлечените данни.

На *фиг. 2* е представена диаграма на внедрената архитектура за обучение тип RAG, представяща подробно трите функционални стадия на обработка на потребителската заявка и осъществяване на последващо обучение.



Фиг. 2. Представяне на внедрената схема за обучение тип RAG

Внедрената схема за обучение RAG съчетава силните страни на традиционните системи за извличане на информация като търсачки и бази данни (проследимост и прозрачност на източниците на информация) с информация на генеративните големи езикови модели.

Сравнително изследване на ефективността на платформата при решаване на общи и специализирани задачи

Осъществено е изследване на ефективността на платформата в два вида сценарии спрямо естеството на поставената към големия езиков модел задача:

- **Задача за общо познание (означена като тип А):** според стандартизираните тестове за оценяване на когнитивна способност при LLM: GLUE и MMLU.
- **Задача от специализирана област на познание (означена като тип Б):** разпознаване и контекстуализиране на команди на програмен език MELFA Basic в програмен код за управление на

индустриален робот RH-6CH7020-D за изпълнение на описана траектория.

Сравнението на режимите на обучение се осъществява на базата на стандартизиран параметър за процентна оценка на контекстова точност (КТ, %), който представлява процентно представяне на степента на разпознаване на контекста от потребителския промпт при обработката на заявката за генериране на съдържание от модела.

Табл. 1. Стандартизиран бенчмаркинг

Модел, дата на достъпване	Задача тип А		
	Хардуер за базово обучение	КТ след локално последващо обучение, %	КТ на модел в облака, %
GPT 4.5, февруари 2025	GPU DGX H200/H100	86.2	85.1
Qwen 3-8B, април 2025	GPU DGX A100	86.5	85
Gemma 3-27B, март 2025	TPUv4p, TPUv5p	67.3	66.9
DeepSeek-R1, януари 2025	GPU DGX H800	89.7	88
LLaMA-3.3-70 B, декември, 2024	GPU DGX H200/H100	73	71.3

Табл. 2. Специализиран бенчмаркинг

Модел, дата на достъпване	Задача тип Б			
	Хардуер за базово обучение	КТ при локално обучение, %	КТ на базов модел, %	КТ при хибридно обучение с RAG, %
GPT 4.5, февруари 2025	GPU DGX H200/H100	81	64	87
Qwen 3-8B, април 2025	GPU DGX A100	83	63.2	87
Gemma 3-27B, март 2025	TPUv4p, TPUv5p	62	38.5	65
DeepSeek-R1, януари 2025	GPU DGX H800	73	73	75
LLaMA-3.3-70 B, декември, 2024	GPU DGX H200/H100	66	62.7	87

Резултатите, представени в *табл. 1*, показват, че при задачи за общо познание локалното последващо обучение в платформата Private Edge LLMs постига съизмерима точност на разпознаване на контекст на потреби-

телските заявки с точността, постижима при обучение на съответните модели в алтернативна облачна платформа Hugging Face. Обучението в алтернативната облачна платформа може да бъде асоциирано с т. нар. скрити загуби, произтичащи от ограниченията на платформата по отношение на отделяне на изчислителен ресурс за обучение във времеви диапазони на засилена употреба на платформата. Тези загуби могат да се проявят като удължаване на времето за обучение или натрупване на такси за наднормено използване на изчислителен ресурс спрямо класа потребител в платформата и техния абонаментен план. Локалното обучение предоставя контрол на потребителя върху ограничаване на скритите загуби в процеса на обучение и същевременно гарантира конфиденциалността на обработваните данни, тъй като те не се споделят с външни доставчици на услуги.

Резултатите, представени в табл. 2, показват, че при задачи от специализирана област на познание локалното последващо обучение в платформата Private Edge LLMs постига значимо повишаване в точността на разпознаване на контекст на потребителските заявки в сравнение с точността, постижима с базовите големи езикови модели. Причината е, че базовите модели са обучавани върху огромни бази знание, които съдържат основна информация от множество приложни области, но рядко засягат експертно или специализирано познание за конкретни строго дефинирани приложни сценарии в тези области. За такъв тип задачи е подходящо да се осъществи допълнително контекстуализиране на потребителската заявка с техники за последващо локално обучение (PEFT, RLHF) и/или разширяване на базата знания на модела чрез генериране с допълнително извеждане (RAG).

Заклучение

Представената в изследването разработена платформа Private Edge LLMs за локално последващо обучение (fine-tuning) и инференция на популярни големи езикови модели (LLMs) е подходяща за обработка на данни в научноизследователски институции и в развойните отдели на предприятия, където конфиденциалността на обработваните данни е с висок приоритет. Внедрената схема за обучение RAG подобрява възможностите на големите езикови модели, като ги комбинира с външни за модела източници на знания, предоставяйки допълнителен контекст за изпълнение на потребителските заявки. Реализацията на схема за обучение RAG адресира и предоставя адекватна компенсация на основни слабости на приложенията, базирани на големи езикови модели, като прозрачност и проследимост на източниците на генерирано съдържание и конфиденциалност на обработваните данни за обучение. Платформата се позиционира като иновативно решение в сравнение със съществуващите, тъй като предоставя комбиниран подход към контекстуализацията на знания както в процеса на обучение, така и посредством генериране с добавено извеждане.

Благодарности

Авторът изказва благодарност на сектор „Научно-изследователска и развойна дейност“ при Техническият университет – София, за финансовата подкрепа по проект 242ПД0033-19.

Литература

1. Stoykova, S., N. Shakev. Artificial Intelligence for Management Information Systems: Opportunities, Challenges, and Future Directions. – *Algorithms*, 16, 2023, 357. <https://doi.org/10.3390/a16080357>.
2. Wessel, M., M. Adam, et al. Generative AI and its Transformative Value for Digital Platforms. – *Journal of Management Information Systems*, 42, June 2025, 2, 346–369. <https://doi.org/10.1080/07421222.2025.2487315>.
3. Storey, Veda C., et al. Generative Artificial Intelligence: Evolving Technology, Growing Societal Impact, and Opportunities for Information Systems Research. – *Information Systems Frontiers*, 2025, 1-22. <https://doi.org/10.1007/s10796-025-10581-7>.
4. Хаджийски, М. Разширяване на обхвата на индустриалната автоматизация с ползване на технологиите на изкуствения интелект. – *Автоматика и информатика*, год. LVIII, 2025, № 1.
5. Wu, X.-K., M. Chen, et al. LLM Fine-Tuning: Concepts, Opportunities, and Challenges. – *Big Data and Cognitive Computation*, April 2025, 9, 87. <https://doi.org/10.3390/bdcc9040087>.
6. Lv, K. et al. Full Parameter Fine-Tuning For Large Language Models with Limited Resources. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Aug 2024, 8187-8198. <https://doi.org/10.48550/arXiv.2306.09782>.
7. Houlsby, N. et al. Parameter Efficient Transfer Learning for NLP. Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9-15 June 2019, 2790–2799. <https://doi.org/10.48550/arXiv.1902.00751>.
8. Valipour, Mojtaba, et al. DyLoRA: Parameter-Efficient Tuning of Pre-Trained Models Using Dynamic Search-Free Low-Rank Adaptation. Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, May 2023, 3274-3287. <https://doi.org/10.48550/arXiv.2210.07558>.
9. Chaudhari, Shreyas, et al. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs. – *ACM Computing Surveys*, 58, 2025, 2, 1-37. <https://doi.org/10.1145/374312>.
10. Zamani, I. and A. Jalaly Bidgoly. A New Approach for Lightweight Fine-Tuning Large Language Model by Adding a New Head as an Adapter. University of Qom, Nov 2025. <http://dx.doi.org/10.2139/ssrn.5719769>.

За контакти:

Ас. маг.-инж. **Сте́ла Стойкова**

Катедра „Системи за управление“

Факултет по електроника и автоматика

Технически университет – София, филиал Пловдив

stoykova@tu-plovdiv.bg