ТЕХНИКАЛЫҚ ҒЫЛЫМДАР
ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCE

CENTRAL ASIAN SCIENTIFIC JOURNAL
ISSN 3135-3061
2026. №4 (32), 2-том. 595–600 б.
DOI: https://doi.org/10.5281/zenodo.24459939
Журнал сайты: https://cajournal.kz

ӘОЖ 004.85

ҮЛКЕН ТІЛДІК МОДЕЛЬДЕРДІҢ АРХИТЕКТУРАСЫ ЖӘНЕ ОЛАРДЫ ҚОЛДАНБАЛАРҒА ИНТЕГРАЦИЯЛАУ ТӘСІЛДЕРІ

А.А. Серіков
М.Х. Дулати атындағы Тараз университеті, Тараз, Қазақстан

Серіков Ағаділ Айдарұлы — 4 курс студенті, М.Х. Дулати атындағы Тараз университеті, Тараз, Қазақстан.

Ғылыми жетекші: Шрымбай Дана Абилахатқызы,Туреханова Гулжанар Ильясовна

© А.А. Серіков, 2026

Аңдатпа: Мақалада үлкен тілдік модельдердің (ҮТМ) архитектурасы және оларды қолданбаларға интеграциялау тәсілдері әдебиеттерге шолу негізінде қарастырылады. Дереккөз ретінде іргелі ғылыми жұмыстар мен ресми техникалық құжаттама іріктелді. Transformer архитектурасынан бастап модельді бейімдеу, іздеумен толықтыру (RAG) және сыртқы құралдарды шақыру тәсілдері салыстырылып, қолданбаның жалпы архитектурасы ұсынылады. Талдау көрсеткендей, тәсілді таңдау шешілетін міндетке (білім, мінез-құлық немесе әрекет) байланысты, ал галлюцинация мен prompt injection тәуекелдерін бөлек басқару қажет. Мақала теориялық сипатта және авторлық эксперимент қамтымайды.

Түйін сөздер: үлкен тілдік модельдер, Transformer, RAG, құралдарды шақыру, Model Context Protocol, галлюцинация, prompt injection.

Аннотация: В статье на основе обзора литературы рассматриваются архитектура больших языковых моделей (БЯМ) и способы их интеграции в приложения. В качестве источников отобраны основополагающие научные работы и официальная техническая документация. От архитектуры Transformer до адаптации модели, расширения поиском (RAG) и вызова внешних инструментов сопоставляются подходы, предлагается общая архитектура приложения. Анализ показывает, что выбор подхода зависит от решаемой задачи (знания, поведение или действие), а риски галлюцинаций и prompt injection требуют отдельного управления. Статья носит теоретический характер и не содержит авторского эксперимента.

Ключевые слова: большие языковые модели, Transformer, RAG, вызов инструментов, Model Context Protocol, галлюцинации, prompt injection.

Abstract: The paper reviews the architecture of large language models (LLMs) and the ways of integrating them into applications on the basis of the literature. Foundational research papers and official technical documentation were selected as sources. Starting from the Transformer architecture, the approaches of model adaptation, retrieval-augmented generation (RAG) and external tool calling are compared, and a general application architecture is proposed. The analysis shows that the choice of an approach depends on the task being solved (knowledge, behaviour or action), while the risks of hallucination and prompt injection require separate management. The work is theoretical and contains no original experiment.

Keywords: large language models, Transformer, RAG, tool calling, Model Context Protocol, hallucination, prompt injection.

Кіріспе

Үлкен тілдік модельдер соңғы жылдары мәтінді түсіну мен жасаудан бастап бағдарламалық код жазуға дейінгі кең міндеттерде қолданыла бастады. Әдебиетте ҮТМ деп әдетте мәтіндік деректердің үлкен көлемінде алдын ала оқытылған, көптеген параметрі бар Transformer негізіндегі модельдерді айтады; шолуда олардың дамуы алдын ала оқыту, бейімдеу, қолдану және сапаны бағалау бағыттары бойынша жүйеленген [1]. Дегенмен модельді іс жүзінде қолдану оны сұрақ қою үшін ашып қоюдан күрделірек: нақты қолданбаға модельден бөлек деректер көзі, сыртқы функциялар және қауіпсіздік тексерулері қосылады.

Тақырыптың өзектілігі бағдарламашы үшін басты сұрақтың өзгеруімен байланысты: модельді қалай жасау емес, дайын модельді қолданбаға қалай дұрыс қосу қажет. Модель архитектурасы, бейімдеу әдістері, іздеумен толықтыру, құралдарды шақыру және қауіпсіздік мәселелері әдетте бөлек жұмыстарда қарастырылады, сондықтан оларды қолданбаны жобалау тұрғысынан бір жүйеге келтіру қажет болады.

Зерттеудің мақсаты – ҮТМ архитектурасының негіздерін және оларды қолданбаларға интеграциялаудың негізгі тәсілдерін әдебиеттер негізінде жүйелеп, таңдау өлшемдері мен тәуекелдерін көрсету. Міндеттері: архитектура негіздерін сипаттау; модельді бейімдеу тәсілдерін қарастыру; интеграция тәсілдерін салыстырып, қолданбаның жалпы архитектурасын ұсыну; негізгі тәуекелдерді анықтау. Мақала әдебиеттерге шолу және теориялық талдау сипатында жазылған, авторлық эксперимент не өлшеу жүргізілмеген.

Материалдар мен әдістер

Шолу қолданба жобалаушыға қажетті төрт сұрақ бойынша жүргізілді: модель қалай құрылған, оны қалай бейімдеуге болады, қолданбаға қалай қосады және қандай тәуекелдер бар. Дереккөздер екі топтан іріктелді: тиісті бағыттың негізін салған, NeurIPS, ICLR және NAACL конференцияларында не рецензияланатын журналда жарияланған іргелі жұмыстар; сондай-ақ ресми техникалық құжаттама мен салалық ұйымның қауіпсіздік нұсқаулығы. Әдістің алғашқы авторлық жұмысы болса, соны алдық, ал қайталама шолулар тек ұғымдарды жүйелеуге қолданылды. Әр дереккөзден ұсынылған тәсілдің мәні, қолдану шарты және авторлар көрсеткен шектеулер алынып, шешілетін міндет түрі бойынша салыстырылды. Бұл жүйелі шолу емес және жарияланымдардың барлығын қамтуды мақсат етпейді, сондықтан қорытындылар сапалық сипатта. Электрондық ресурстарға сілтемелер 2026 жылғы 17 қарашада тексерілді.

Нәтижелер және талқылау

ҮТМ архитектурасының негіздері

Қазіргі ҮТМ-нің көпшілігі Transformer архитектурасына негізделген. Оны ұсынған жұмыста тізбекті өңдеуде рекуренттік және конволюциялық блоктардан бас тартып, тек зейін (attention) механизміне сүйену ұсынылған [2]. Авторлар негізгі артықшылық ретінде есептеуді параллель орындау мүмкіндігін көрсетеді. Одан әрі екі үлгі қалыптасты. BERT екі бағытты контексті пайдаланатын кодтаушы ретінде алдын ала оқытылып, әр міндетке қосымша шығыс қабатын қосу арқылы бейімделеді [3]. GPT-3 сияқты авторегрессивті модельдер келесі токенді болжау арқылы мәтін жасайды, ал 175 млрд параметрлі модель параметрлерін өзгертпей-ақ, промптқа бірнеше мысал қосқанда жаңа міндетті орындай алатыны көрсетілген [4]. Қолданбаға интеграциялау тұрғысынан екінші үлгі ыңғайлы: оған тапсырманы мәтінмен беруге болады.

Модельді бейімдеу тәсілдері

Алдын ала оқытылған модель пайдаланушы нұсқауын әрдайым дұрыс орындамайды. Бұл мәселені адам бағалауымен оқыту арқылы шешу ұсынылған: InstructGPT жұмысында модель адамдардың жауап үлгілері мен бағалары негізінде қосымша оқытылған, нәтижесінде әлдеқайда шағын InstructGPT жауаптары үлкен GPT-3 жауаптарынан гөрі артық бағаланған

[5]. Модельді қайта оқытпай-ақ сапаны арттырудың тағы бір жолы – промптың өзін құрастыру: қадамдық ойлау мысалдарын (chain-of-thought) беру көпсатылы есептерде нәтижені жақсартады [6]. Модельді өз деректеріңе бейімдеу керек болса, толық қайта оқыту қымбат; LoRA тәсілі алдын ала оқытылған салмақтарды қатырып, әр қабатқа аз параметрлі төмен рангілі матрицалар қосып, оқытылатын параметр санын айтарлықтай азайтады [7]. Осылайша бейімдеудің үш деңгейі пайда болады: промпт, жеңіл баптау және толық оқыту; қолданба әзірлеушісі үшін алғашқы екеуі қолжетімдірек (автор бағасы).

Қолданбаға интеграциялау тәсілдері

Модель өз параметрлерінде сақталған біліммен шектеледі және ол өздігінен жаңармайды. Мұны айналып өтудің бір жолы – RAG: сұрау бойынша сыртқы құжаттардан сәйкес үзінділерді іздеуші арқылы табады да, оларды жауап жасауға қосады; түпнұсқа жұмыста параметрлік жады (генеративті модель) пен параметрлік емес жады (Wikipedia-ның векторлық индексі) біріктірілген [8]. Іс жүзінде де осы идея қолданылады: құжаттар индекске салынады, ал сұрау келгенде ең жақын үзінділер модельге контекст ретінде беріледі (автор түсіндірмесі; [8]-де модель бірге оқытылған, ал қолданбада әдетте дайын модель алынады).

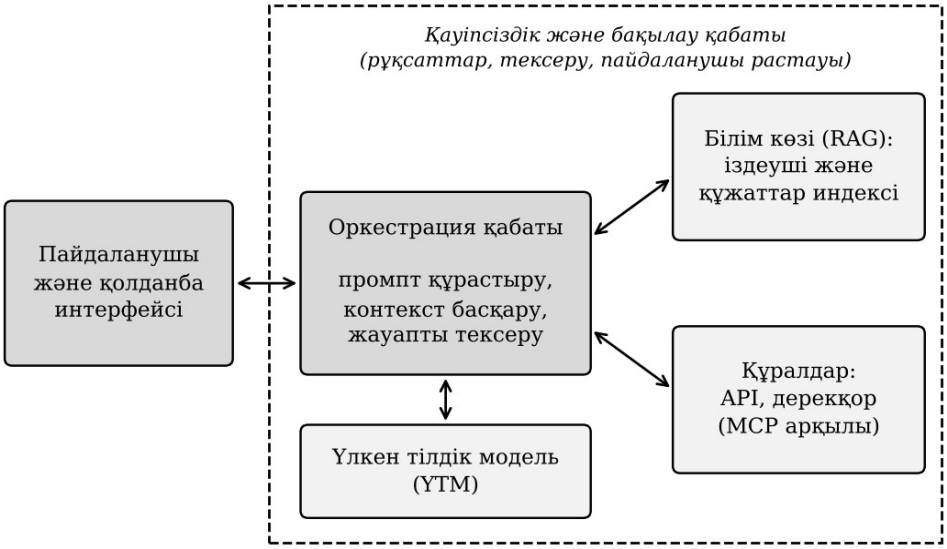
Екінші бағыт – модельге әрекет жасауға мүмкіндік беру. ReAct тәсілінде модель ойлау қадамдарын және сыртқы ортаға (мысалы, Wikipedia API) жасалатын әрекеттерді кезек-кезек орындайды: ойлау әрекетті бағыттайды, ал әрекет нәтижесі ойды жаңа ақпаратпен түзетеді [9]. Toolformer жұмысында модель қандай API-ды қашан, қандай аргументпен шақыруды және нәтижені қалай пайдалануды өзі үйренеді, ол үшін әр API-ға аз ғана мысал жеткілікті [10]. Құралдарды қолданбаға қосуды стандарттауға Model Context Protocol бағытталған: ол хост (ҮТМ қолданбасы), клиент және сервер арасындағы байланысты JSON-RPC 2.0 негізінде сипаттайды, ал сервер ресурстарды, промпт үлгілерін және модель орындайтын құралдарды ұсына алады [11]. Тәсілдердің салыстырмасы 1-кестеде берілген.

Кесте 1 – ҮТМ-ді қолданбаға интеграциялау тәсілдерінің салыстырмасы

Тәсіл	Мәні	Күшті жағы және шектеуі	Дереккөз
Промпт арқылы шақыру	Тапсырма мен мысалдар мәтінмен беріледі, модель параметрлері өзгермейді	Ең қарапайым; білім модель оқуымен шектеледі	[4], [6]
Модельді бейімдеу	Адам бағалауымен немесе жеңіл баптаумен (LoRA) қосымша оқыту	Мінез-құлық пен пішімді түзетеді; деректер мен есептеу қажет	[5], [7]
RAG	Сұрау бойынша сыртқы құжаттардан үзінді табып, жауапқа қосады	Жаңа не жеке деректермен жұмыс істейді; іздеу сапасына тәуелді	[8]
Құралдарды шақыру	Модель API не функцияны шақырып, нәтижені пайдаланады	Әрекет жасауға мүмкіндік береді; қауіпсіздікті талап етеді	[9]–[11]

Ескерту: автор жүйелеуі, [4]–[11] негізінде

Қолданбаның жалпы архитектурасы 1-суретте көрсетілгендей құрылуы мүмкін: пайдаланушы сұрауы оркестрация қабатына келеді, ол промпт құрастырып, қажет болса білім көзінен үзінді алады не құралды шақырады және ҮТМ жауабын тексеріп қайтарады. Бұл сызба автор жүйелеуі болып, [8]–[11] жұмыстарындағы тәсілдерді бір қолданбаға біріктіреді.



Сурет 1 - ҮТМ негізіндегі қолданбаның жалпы архитектурасы (автор жүйелеуі)

Қолданбадағы сұраудың жолын бірнеше қадамға бөлуге болады (автор сипаттамасы). Алдымен оркестрация қабаты кіріс мәтінді тексеріп, сеанстың алдыңғы мәліметімен біріктіреді. Содан кейін қажет болса, білім көзінен үзінділер алынып, промптқа қосылады. Модель жауап береді не құралды шақыру туралы құрылымдалған сұрау қайтарады; бұл сұрауды модельдің өзі орындамайды, оны қолданба қабылдап, рұқсатты тексеріп, орындайды да, нәтижені модельге қайта береді. Соңында жауап пайдаланушыға жеткізілмес бұрын тексеріледі. Осылай модельдің әрекеті әрқашан қолданба бақылайтын қадам болып қалады, ал мұндай бөлу қауіпсіздік тексерулерін қоятын орынды анық көрсетеді.

Мысал ретінде тұрмыстық қызметтерге тапсырыс беру веб-жүйесін алайық (бұл ойша сценарий, іске асырылған жүйе емес). Пайдаланушы «ертең сағат 10-ға сантехник керек» деп жазады. Оркестрация қабаты RAG арқылы қызмет каталогы мен тарифтерден сәйкес үзінділерді табады, модель қызмет түрі мен уақытын анықтап, «тапсырыс жасау» құралын шақыруды ұсынады. Құрал орындалмас бұрын қолданба пайдаланушыдан растау сұрайды, содан кейін ғана тапсырыс дерекқорға жазылады.

Тәуекелдер

Бірінші тәуекел – галлюцинация: Ji және авторлар оны жасалған мазмұнның мағынасыз не берілген дереккөзге сәйкес келмеуі ретінде анықтайды және дереккөзге қайшы (intrinsic) және дереккөзден тексеруге болмайтын (extrinsic) түрлерге бөледі [12]. Қолданбада оны азайту үшін жауапты іздеу нәтижелеріне сүйеніп беру, дереккөзді көрсету және маңызды әрекеттерде пайдаланушы растауын талап ету ұсынылады (автор ұсынысы). Екінші тәуекел – prompt injection: OWASP ұйымының LLM қолданбаларына арналған Top 10 тізімінде ол бөлек қауіп ретінде қарастырылады [13]. RAG-та индекске түскен құжат, құралдарда оның сипаттамасы модельге нұсқау ретінде әсер етуі мүмкін, сондықтан Model Context Protocol ережесінде құрал сипаттамасы сенімді серверден алынбаса, сенімсіз деректер деп есептелуі және құралды шақыру алдында пайдаланушының нақты келісімі алынуы тиіс екені көрсетілген [11].

Негізгі тәуекелдер мен оларды азайту шаралары 2-кестеде жинақталған.

Кесте 2 – ҮТМ негізіндегі қолданбалардың негізгі тәуекелдері және оларды азайту шаралары

Тәуекел	Қайдан пайда болады	Азайту шарасы (автор ұсынысы)	Дереккөз
Галлюцинация	Модель дереккөзге сәйкес келмейтін не тексерілмейтін мазмұн жасайды	Жауапты іздеу нәтижелеріне сүйеу, дереккөзді көрсету, маңызды әрекетте адам растауы	[12]

Тәуекел	Қайдан пайда болады	Азайту шарасы (автор ұсынысы)	Дереккөз
Prompt injection	Құжаттағы не құрал сипаттамасындағы мәтін нұсқау ретінде қабылданады	Сыртқы мәтінді сенімсіз деп санау, шығысты тексеру, құқықты шектеу	[11], [13]
Рұқсатсыз әрекет	Құрал қосымша тексерусіз орындалады	Әр құралды шақыру алдында пайдаланушы келісімін алу	[11]
Жеке деректердің сыртқа шығуы	Пайдаланушы деректері серверге не сыртқы қызметке беріледі	Деректі беру алдында нақты келісім алу, рұқсатты шектеу	[11]

Ескерту: автор жүйелеуі, [11]–[13] негізінде

Талқылау

Талдау үш тұжырымды көрсетеді (автор қорытындысы). Біріншіден, тәсілді таңдау қолданбадағы кемшілікке байланысты: білім жетіспесе – RAG; мінез-құлық не пішім дұрыс болмаса – промпт немесе бейімдеу; әрекет жасау керек болса – құралдар. Екіншіден, тәсілдер бір-бірін алмастырмай, бір қолданбада бірге жұмыс істейді, мұны жоғарыдағы сценарий көрсетеді. Үшіншіден, мүмкіндік артқан сайын тәуекел де артады: RAG-та тәуекел көзі құжаттар, құралдарда орындалатын әрекет. Практикалық таңдау үшін мына өлшемдерді ұсынуға болады (автор бағасы): деректердің өзгеру жиілігі (жиі өзгерсе, индексті жаңарту модельді қайта оқытудан жеңіл болғандықтан RAG қолайлы); деректің құпиялылығы (құпия құжатты сұрау кезінде контекст ретінде беру басқаруға оңай, бірақ оның қайда жіберілетінін көрсету қажет); әрекеттің қайтарымсыздығы (төлем не тапсырыс сияқты әрекеттерге адам растауы керек); күрделілік (әр қосымша қабат сынау мен қолдауды қиындатады). Бұл өлшемдер әдебиеттегі тәсілдердің сипаттамасынан шығарылған және сандық түрде тексерілмеген. Шолудың шектеулері: ол жүйелі емес, сала тез өзгереді, ал тәсілдерді нақты өлшемдермен (дәлдік, кідіріс, құны) салыстыру мақаланың міндетіне кірмейді және эксперименттік зерттеуді қажет етеді.

Қорытынды

Әдебиеттерге шолу ҮТМ-ді қолданбаға интеграциялауды төрт тәсіл ретінде жүйелеуге мүмкіндік берді: промпт арқылы шақыру, модельді бейімдеу, іздеумен толықтыру (RAG) және құралдарды шақыру. Тәсілді таңдау модельдің жетіспейтін қасиетіне – біліміне, мінез-құлқына не әрекет жасау қабілетіне – байланысты, ал нақты қолданбада олар әдетте біріктіріледі. Осы бірлестіктің негізгі қауіпті жерлері – галлюцинация мен prompt injection, сондықтан қауіпсіздік және бақылау қабаты қолданба архитектурасының бөлігі болуы тиіс. Алынған тұжырымдар әдебиеттерге негізделген сапалық сипатта; оларды нақты қолданбада өлшеммен тексеру болашақ жұмыстың міндеті болып қалады.

Әдебиеттер тізімі:

1. Zhao W.X. et al. A Survey of Large Language Models // Frontiers of Computer Science. – 2026. – Vol. 20, No. 12. – Art. 2012627. – DOI: 10.1007/s11704-026-60308-3.

2. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need // Advances in Neural Information Processing Systems 30 (NeurIPS 2017). – 2017. – P. 5998–6008. – URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (қаралған күні: 17.11.2026).

3. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Vol. 1. – Minneapolis: ACL, 2019. – P. 4171–4186. – DOI: 10.18653/v1/N19-1423.

4. Brown T., Mann B., Ryder N. et al. Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). – 2020. – P. 1877–1901. – URL: https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (қаралған күні: 17.11.2026).

5. Ouyang L., Wu J., Jiang X. et al. Training language models to follow instructions with human feedback // Advances in Neural Information Processing Systems 35 (NeurIPS 2022). – 2022. – URL:

https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html (қаралған күні: 17.11.2026).

6. Wei J., Wang X., Schuurmans D. et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models // Advances in Neural Information Processing Systems 35 (NeurIPS 2022). – 2022. – P. 24824–24837.

7. Hu E.J. et al. LoRA: Low-Rank Adaptation of Large Language Models // International Conference on Learning Representations (ICLR 2022). – 2022. – URL: https://openreview.net/forum?id=nZeVKeeFYf9 (қаралған күні: 17.11.2026).

8. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). – 2020. – P. 9459–9474.

9. Yao S. et al. ReAct: Synergizing Reasoning and Acting in Language Models // International Conference on Learning Representations (ICLR 2023). – 2023. – URL: https://openreview.net/forum?id=WE_vluYUL-X (қаралған күні: 17.11.2026).

10. Schick T., Dwivedi-Yu J., Dessi R. et al. Toolformer: Language Models Can Teach Themselves to Use Tools // Advances in Neural Information Processing Systems 36 (NeurIPS 2023). – 2023. – P. 68539–68551.

11. Model Context Protocol Specification [Электронный ресурс]. – Version 2026-07-28. – URL: https://modelcontextprotocol.io/specification/2026-07-28 (қаралған күні: 17.11.2026).

12. Ji Z., Lee N., Frieske R. et al. Survey of Hallucination in Natural Language Generation // arXiv:2202.03629 [cs.CL]. – 2022 (v7, 2024). – URL: https://arxiv.org/abs/2202.03629 (қаралған күні: 17.11.2026).

13. OWASP Top 10 for LLM Applications 2025 [Электронный ресурс] / OWASP Gen AI Security Project. – URL: https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/ (қаралған күні: 17.11.2026).