 DOI: https://doi.org/10.26493/scores26.04

Obdelava ogroženih jezikov z velikimi jezikovnimi modeli na osnovi jezikoslovnih virov

Enej Krabonja
enej.krabonja@student.um.si
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Koroška cesta 46
SI-2000 Maribor, Slovenija

Miha Pliberšek
miha.plibersek@student.um.si
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Koroška cesta 46
SI-2000 Maribor, Slovenija

Oskar Forštnarič
oskar.forstnaric@student.um.si
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Koroška cesta 46
SI-2000 Maribor, Slovenija

Tjaša Lah
tjasa.lah1@student.um.si
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Koroška cesta 46
SI-2000 Maribor, Slovenija

Maks Marolt
maks.marolt@student.um.si
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Koroška cesta 46
SI-2000 Maribor, Slovenija

Borko Boškovič
borko.boskovic@um.si
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Koroška cesta 46
SI-2000 Maribor, Slovenija

POVZETEK

Prispevek obravnava obdelavo ogroženih jezikov z velikimi jezikovnimi modeli na osnovi jezikoslovnih virov. Izhodišče je sistem LingoLLM, ki z vključitvijo morfološke analize, slovarja in slovnice omogoča prevajanje jezikov brez vzporednih učnih podatkov. Predlagamo tri optimizacije: uporabo lokalnega modela, dinamično pridobivanje slovničnih odlomkov in stiskanje pozivov z LLMLingua. Eksperiment na jeziku manchu in slovenščini pokaže, da dinamično pridobivanje slovničnih pravil bistveno zmanjša število vhodnih žetonov, hkrati pa ohrani ali izboljša kakovost prevodov.

KLJUČNE BESEDE

ogroženi jeziki, veliki jezikovni modeli, strojno prevajanje, dinamično pridobivanje z jezikovnimi viri, stiskanje pozivov

1 UVOD

V zadnjih letih so veliki jezikovni modeli (angl. *large language models*; LLM) bistveno izboljšali kakovost obdelave naravnega jezika, med drugim tudi na področju strojnega prevajanja. Modeli, kot je ChatGPT, pri prevajanju visoko zastopanih jezikov pogosto dosegajo primerljive ali celo boljše rezultate od specializiranih prevajalskih sistemov. Vendar pa ta prednost ni univerzalna. Avtorji v delu [5] ugotavljajo, da zmogljivost modela ChatGPT izrazito upade pri jezikih z omejenimi viri, saj so ti jeziki v učnih podatkih praviloma slabo zastopani ali povsem odsotni. To predstavlja pomembno omejitev pri uporabi sodobnih jezikovnih modelov za globalno večjezično podporo. Problem je posebej izrazit pri ogroženih jezikih, kjer pogosto ne obstajajo vzporedni korpusi, obsežni digitalni slovarji ali zadostne količine besedil za dodatno učenje modelov.

Po ocenah je danes ogroženih več kot 40 % svetovnih jezikov, pri čemer številni raziskovalci napovedujejo nadaljnje pospešeno izumiranje jezikovne raznolikosti. Zaradi tega postaja razvoj metod za digitalno podporo ogroženim jezikom pomemben tako z vidika računalništva kot jezikovne revitalizacije. Pomemben prispevek na tem področju predstavlja raziskava [12], v kateri avtorji analizirajo možnosti uporabe metod obdelave naravnega jezika pri revitalizaciji čerokejščine. Predlagajo več ključnih nalog, med njimi gradnjo govornih korpusov, razvoj morfoloških analizatorjev ter strojno prevajanje. Posebej poudarijo, da tehnološke rešitve same po sebi niso zadostne, temveč morajo biti tesno povezane z obstoječimi jezikovnimi viri in sodelovanjem z jezikovno skupnostjo.

Na tej osnovi delo [13] predstavi sistem LingoLLM, ki velikim jezikovnim modelom omogoča obdelavo ogroženih jezikov z uporabo eksplicitnih jezikoslovnih virov. Namesto dodatnega učenja model uporablja slovnice, slovarje in rezultate morfološke analize neposredno v pozivu. Poziv je celotno besedilo, ki ga sistem posreduje jezikovnemu modelu kot vhod. Tak pristop omogoča prevajanje tudi za jezike brez vzporednih učnih podatkov. Avtorji pokažejo, da lahko s tem pristopom bistveno izboljšajo kakovost prevajanja v primerjavi z neposredno uporabo velikega jezikovnega modela brez dodatnega konteksta.

Sorodna dela dodatno potrjujejo potencial uporabe strukturiranega jezikovnega znanja pri prevajanju jezikov z malo viri. Dela [1, 3] pokažejo, da je mogoče prevajanje izboljšati že z uporabo slovnic in slovarjev brez klasičnih paralelnih korpusov. Raziskave na področju večjezičnih LLM [4, 7, 8] medtem predlagajo dodatno učenje, medjezikovno sklepanje in optimizacijo vhodnih podatkov za izboljšanje večjezičnih sposobnosti modelov.

Kljub obetavnim rezultatom ima LingoLLM pomembno omejitev: celotna slovnica se pri vsakem klicu ponovno vključuje v poziv modela. To povečuje število vhodnih žetonov, podaljšuje čas obdelave in povečuje računske stroške, zlasti pri večjih slovničnih opisih. Ta omejitev je posebej problematična pri praktični uporabi lokalno gostovanih modelov, kjer so pomnilniške in računske omejitve izrazitejše. V tem prispevku predstavimo optimizacijo sistema LingoLLM s tremi izboljšavami: uporabo lokalnih jezikovnih modelov, dinamičnim pridobivanjem relevantnih slovničnih odlomkov z metodo RAG (angl. *Retrieval Augmented Generation*) in stiskanjem pozivov z orodjem LLMLingua. Cilj predlaganega pristopa je

zmanjšati računsko zahtevnost ob ohranitvi ali izboljšanju kakovosti prevajanja pri jezikih z omejenimi viri.

Prispevek je organiziran tako, da poglavje 2 predstavi predlagane optimizacije sistema LingoLLM, poglavje 3 eksperimentalno metodologijo in rezultate vrednotenja, poglavje 4 pa povzame glavne ugotovitve in možnosti nadaljnjega dela.

2 OPTIMIZACIJA SISTEMA LINGOLLM

Čeprav pristop v delu [13] dosega odlične rezultate pri prevajanju jezikov brez učnih podatkov, je izjemno potraten z vidika računalniških virov. Celotna slovnica se vsakič znova pošlje v LLM, kar povečuje zakasnitev, porabo pomnilnika in pri uporabi zunanjih storitev tudi stroške. Naša strategija za optimizacijo prvotnega pristopa naslavlja omenjene pomanjkljivosti s tremi glavnimi gradniki: lokalni LLM (Ollama) - nadomešča komercialne programske aplikacije za vmesnike (angl. *Application Programming Interface*; API) [9], RAG - dinamično pridobivanje le tistih delov slovnice, ki so relevantni za trenutni stavek [2] in kompresija pozivov (LLMLingua) - dodatno zmanjšanje števila žetonov z ohranjanjem ključnih informacij [6].

2.1 Lokalni jezikovni model

Namesto klica GPT-4 smo uporabili lokalno gostovan model z orodjem Ollama [9]. S tem smo odpravili stroške klica do storitve in zagotovili zasebnost podatkov. Ollama omogoča preprost zagon kvantiziranih modelov, kot so Mistral 7B, Llama 3 ali Mixtral, s prilagodljivim nadzorom porabe pomnilnika.

2.2 RAG - dinamično pridobivanje slovničnih pravil

Namesto celotne slovnice v poziv vključimo le tiste odlomke, ki so pomembni za trenutni vhodni stavek [2]. Postopek poteka v štirih korakih:

- priprava podatkov: slovnico razdelimo na smiselne dele (poglavja, odstavke),
- vložitve (angl. *embeddings*): za vsak del izračunamo vektorsko predstavitev in ga vstavimo v vektorsko bazo,
- pridobivanje: za vsak vhodni stavek izračunamo njegovo vektorsko predstavitev in v bazi poiščemo semantično najbližje odlomke; privzeto vrnemo tri najustreznejše,
- združevanje: pridobljene odlomke vstavimo v poziv pred stavkom in navodilom.

2.3 Kompresija pozivov

Poziv vključuje navodila za prevajanje, slovnične odlomke, pridobljene z metodo RAG, in vhodni stavek. Kljub metodi RAG lahko poziv še vedno vsebuje odvečne besede, primere ali ponavljanja. Zato število žetonov dodatno zmanjšamo z odstranitvijo manj pomembnih delov poziva, pri čemer ohranimo ključne informacije. Uporabimo knjižnico LLMLingua [6], ki z manjšim pomožnim modelom oceni pomembnost vsakega žetona in odstrani tiste z najnižjo oceno. Kompresijo izvedemo po sestavljanju končnega poziva, tako da pridobljeni odlomki in vhodni stavek skupaj tvorijo izhodišče za stiskanje. Rezultat je kompakten poziv, ki ga posredujemo jezikovnemu modelu.

3 EKSPERIMENT

3.1 Pridobivanje podatkov

Za primerljivost s sistemom iz [13] smo uporabili dva nabora podatkov: korpus jezika manchu in lasten korpus za slovenščino.

Za manchu smo prevzeli celoten nabor podatkov iz [13], ki vsebuje 101 par stavkov za prevajanje med manchu in angleščino, 15 stavkov za matematično sklepanje, demonstracijske primere za učenje iz nekaj primerov (*few-shot*), morfološki slovar in označeno slovnico. Gre za isti nabor, na katerem je izhodiščni sistem dosegel vrednost 0,5 metrike BLEU (angl. Bilingual Evaluation Understudy) pri neposrednem GPT-4 in BLEU 10 pri LingoLLM, zato omogoča neposredno primerjavo.

Slovenščino smo vključili kot srednje zastopan jezik s kompleksno morfologijo [11]. Uporabili smo referenčni nabor FLORES-200 [10], demonstracijske primere pa izbrali tako, da pokrivajo sklanjanje samostalnikov, spreganje glagolov, dvojino in naslonko *se*.

3.2 Priprava okolja

Izhodiščni sistem LingoLLM je temeljil na klicih vmesnika OpenAI in statičnem nalaganju celotnih slovničnih opisov v poziv modela. V predstavljeni raziskavi smo sistem razširili s štirimi komponentami, od katerih vsaka zahteva posebno nastavitev: z lokalnim jezikovnim modelom, modulom za dinamično pridobivanje odlomkov, modulom za kompresijo poziva in s podporo za slovenščino kot dodaten testni jezik.

3.3 Postopek

Eksperiment smo izvedli z lokalnimi modeli, s čimer smo odpravili odvisnost od komercialnih storitev. Preizkusili smo lokalno nameščene kvantizirane različice modelov Qwen2.5 in Qwen3.6. Uporabili smo model Qwen2.5 32B s 4-bitno in 5-bitno kvantizacijo ter model Qwen3.6 35B s 5-bitno kvantizacijo. Kvantizacija zmanjša porabo pomnilnika ob minimalnem vplivu na kakovost rezultatov. RAG je bil vključen v vse različice sistema. Namesto celotne slovnice sistem za vsak poziv poišče semantično najbližje odlomke z vektorskim iskanjem in jih vstavi v poziv modela. Na ta način se zmanjša skupna dolžina vhodnega besedila in izboljša ustreznost posredovanih informacij.
Stiskanje poziva je bilo vključeno v različice, ki vsebujejo daljše slovnične opise, saj pri kratkih pozivih stiskanje ni smiselno, ker bi lahko povzročilo izgubo informacij. Kompresija ohranja tiste dele vhodnega besedila, ki so semantično najbližje vhodnemu stavku.

Po zaključenem eksperimentu smo prevode ovrednotili z izračunom metrik BLEU in chrF za vsak stavek posebej ter s povprečjem po celotnem testnem nizu. Metrika chrF je bila glavni kriterij primerjave, ker je za morfološko bogate in jezike z omejenimi viri robustnejša od metrike BLEU.

3.4 Rezultati

Tabeli 1 in 2 prikazujeta primerjavo različnih konfiguracij. Pri manchu slovarski pristop dosledno preseže neposredno prevajanje: najboljša konfiguracija (Gemini z RAG) doseže vrednost BLEU 21,13 in chrF 51,57, najboljši neposredni sistem pa BLEU 12,20 in chrF 37,57.

Tabela 1: Pregled metrik – manchu.

Slovar	Konfiguracija	BLEU	chrF
Da	Gemini (slovar)	18,87	46,81
Da	Gemini + RAG (slovar + slovnica)	21,13	51,57
Da	Gemini + RAG + kompresija	16,82	44,31
Da	Qwen3.6 35B Q5_K_M + RAG + kompresija	15,16	39,63
Da	gpt-4-1106-preview [13]	11,10	38,89
Da	Qwen2.5 32B Q5_K_M	9,79	36,32
Da	Qwen2.5 32B Q4_K_M	7,59	34,88
Ne	Gemini	11,30	37,63
Ne	Gemini + kompresija	11,25	37,18
Ne	Gemini + RAG + kompresija	12,20	37,57
Ne	gpt-4-1106-preview [13]	3,57	23,03
Ne	Qwen2.5 32B Q5_K_M	0,17	14,97
Ne	Qwen2.5 32B Q4_K_M	0,16	14,63

Tabela 2: Pregled metrik – slovenščina.

Slovar	Konfiguracija	BLEU	chrF
Da	Gemini (slovar)	34,03	64,96
Da	Gemini + RAG (slovar + slovnica)	33,83	65,55
Da	Gemini + RAG + kompresija	34,94	64,26
Da	Qwen3.6 35B Q5_K_M + RAG + kompresija	33,80	63,27
Da	Qwen3.6 35B Q5_K_M	0,44	11,61
Ne	Gemini	37,51	66,25
Ne	Gemini + kompresija	37,34	65,65
Ne	Gemini + RAG + kompresija	37,09	65,50
Ne	Qwen3.6 35B Q5_K_M	31,68	63,58

RAG se pri modelu Gemini izkaže kot ključna optimizacija. V primerjavi z izhodiščnim GPT-4 [13] izboljša rezultat za BLEU 10,03 in chrF 12,68. Dodana kompresija nekoliko zmanjša kakovost prevoda, vendar zmanjša število vhodnih žetonov in računske stroške. Pri neposrednem prevajanju brez slovarja vpliv RAG ostane majhen.

Lokalni model Qwen2.5 32B brez slovarskega konteksta pri manchu praktično odpove (vrednost BLEU 0,16–0,17), s slovarjem pa doseže BLEU 7,59–9,79. Prehod s kvantizacije Q4 na Q5 prinese zmerno izboljšanje, večji napredek pa povečanje modela na Qwen3.6 35B, ki doseže BLEU 15,16 in chrF 39,63. Kljub temu ne doseže ravni Gemini z RAG, kar kaže na razliko med lokalnimi in komercialnimi modeli pri ogroženih jezikih.

Pri slovenščini so rezultati višji, saj je jezik dobro zastopan v učnih podatkih modelov. Neposredno prevajanje z modelom Gemini doseže vrednost BLEU 37,51 in preseže vse slovarske konfiguracije, kar kaže, da dodatni jezikovni viri pri poznanem jeziku ne prinesejo enake koristi kot pri manchu. Nasprotno lokalni model Qwen3.6 35B brez slovarskega konteksta doseže le BLEU 0,44, z RAG in stiskanjem pa BLEU 33,80, kar potrjuje pomen jezikoslovnega konteksta tudi pri jeziku s srednje veliko količino virov.

3.4.1 Učinkovitost obdelave. Število vhodnih in izhodnih žetonov je neposredna mera računske zahtevnosti posamezne konfiguracije. Jezikovni modeli porabijo pomnilnik in čas sorazmerno s številom žetonov v vhodu, število izhodnih žetonov pa odraža obsežnost generiranega odgovora. Tabela 3 prikazuje primerjavo štirih konfiguracij za slovenščino. Največji učinek ima RAG, ki zmanjša število vhodnih žetonov s 4536 na 487 (9,3-krat), hkrati pa izboljša kakovost prevoda na BLEU 27,96 in chrF 59,02. To kaže, da vključevanje celotne slovnice modelu ne koristi, saj predolg poziv poveča količino nerelevantnega konteksta..

Tabela 3: Primerjava učinkovitosti konfiguracij – slovenščina (gemini-3.1-flash-lite, slovarski pristop).

Konfiguracija	Povp. čas (s)	V. žetoni	I. žetoni	BLEU/chrF
Osnovna	96,78	4536	678	25,04/56,29
RAG	91,57	487	647	27,96/59,02
Stiskanje	150,91	1012	984	1,53/26,06
RAG + stiskanje	91,88	444	604	27,14/56,94

Samo stiskanje brez RAG se izkaže za neučinkovito: čas obdelave naraste za 56 %, kakovost pa pade na BLEU 1,53 in chrF 26,06, ker se pomembni deli slovnice odstranijo že pred izbiro relevantnega konteksta.

Kombinacija RAG in stiskanja doseže najmanj vhodnih (444) in izhodnih (604) žetonov ob kakovosti, primerljivi z uporabo samega RAG, zato predstavlja računsko najučinkovitejšo konfiguracijo.

Tabela 4: Primerjava učinkovitosti konfiguracij – manchu (gemini-3.1-flash-lite, slovarski pristop).

Konfiguracija	Povp. čas (s)	V. žetoni	I. žetoni	BLEU/chrF
Osnovna	38,73	1527	444	25,91/60,05
RAG	39,53	1491	436	21,50/53,22
Stiskanje	51,84	1186	475	21,28/49,58
RAG + stiskanje	51,87	1199	474	28,40/63,09

Rezultati za manchu (tabela 4) se razlikujejo zaradi bistveno krajše slovnice. RAG zmanjša število vhodnih žetonov le za 36, zato ne izboljša niti časa obdelave niti kakovosti prevoda. Samo stiskanje ponovno poslabša rezultate in podaljša čas izvajanja.

Najboljše rezultate pri manchu doseže šele kombinacija RAG in stiskanja (vrednost BLEU 28,40, chrF 63,09). Čeprav je nekoliko počasnejša, doseže najvišjo kakovost prevoda, saj učinkovito izbere in zgosti le relevantne dele slovnice. Številčne rezultate dopolnjuje kvalitativna primerjava prevodov.

3.4.2 Kvalitativna primerjava prevodov. Tabela 5 prikazuje primer prevoda iz jezika manchu v angleščino. Najboljši rezultat doseže Gemini z RAG in kompresijo ob uporabi slovarja, saj pravilno ohrani vzročno razmerje in vlogo osebka. Izhodiščni GPT-4 napačno interpretira subjekt in časovni okvir, lokalni model Qwen2.5 pa brez ustreznega slovarskega konteksta od referenčnega prevoda opazno odstopa.

4 ZAKLJUČEK

Cilj prispevka je bil optimizirati sistem LingoLLM za prevajanje ogroženih jezikov z zmanjšanjem računske zahtevnosti ob ohranitvi oz. izboljšanju kakovosti prevodov. V ta namen smo sistem nadgradili z lokalnim jezikovnim modelom, dinamičnim pridobivanjem slovničnih odlomkov z metodo RAG ter kompresijo pozivov.

Tabela 5: Primerjava prevodov izbranega stavka iz jezika manchu v angleščino.

Konfiguracija	Prevod
Vhod (manchu)	*emu guqu tutafi jimeofi bi elhexeme aliyakiyame yabure jakade tuttu jime goidaha*
Referenca	As a friend of mine was delayed and coming late, I was travelling slowly to wait for him, and because of this my coming has taken such a long time.
GPT-4 + slovar (izhodišče)	One companion, having remained behind, came to the place, leisurely lingering, going to the vicinity like that, came a long time ago.
Qwen2.5 32B Q5 + slovar	Long ago, my friend remained behind, and I waited leisurely for him to come, thus he came late.
Gemini + RAG + komp. (brez slovarja)	A friend having stopped and come, because I was idling and waiting, it has been a long time since they came like that.
Gemini + RAG + komp. (slovar)	A friend stayed behind and came to join me, but because I was walking slowly and lingering, it took a long time to arrive.

Eksperimente smo izvedli na jeziku manchu in slovenščini ter primerjali različne konfiguracije sistema. Rezultati kažejo, da RAG bistveno zmanjša število vhodnih žetonov (v izbranih konfiguracijah), pri slovenščini z 4536 na 487, hkrati pa izboljša kakovost prevoda (z vrednosti BLEU 25,04 na 27,96 in chrF 56,29 na 59,02). Pri manchu najboljše rezultate doseže kombinacija RAG in kompresije (BLEU 28,40, chrF 63,09). Rezultati kažejo tudi, da učinkovitost metode RAG ni odvisna zgolj od same uporabe pridobivanja konteksta, temveč predvsem od obsega in kompleksnosti razpoložljivih jezikoslovnih virov. Pri daljših slovničnih opisih RAG bistveno zmanjša število vhodnih žetonov in izboljša kakovost prevoda, medtem ko je pri krajših slovnicah njegov prispevek manj izrazit.

V nadaljnjem delu bi bilo smiselno pristop ovrednotiti še na večjem številu ogroženih jezikov ter raziskati naprednejše strategije pridobivanja in izbire jezikoslovnih virov. S tem bi sistem dodatno približali praktični uporabi pri digitalni podpori ogroženim jezikom.

Literatura

[1] Antonios Hus, Jonathan in Anastasopoulos. 2024. Back to School: Translation Using Grammar Books. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Mohit in Chen Yun-Nung Al-Onaizan, Yaser in Bansal (Ed.). Association for Computational Linguistics, Miami, Florida, ZDA, 20207–20219. doi:10.18653/v1/2024.emnlp-main.1127

[2] Patrick Lewis in Ethan Perez in Aleksandra Piktus in Fabio Petroni in Vladimir Karpukhin in Naman Goyal in Heinrich Küttler in Mike Lewis in Wen-tau Yih in Tim Rocktäschel in Sebastian Riedel in Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. doi:10.48550/arXiv.2005.11401

[3] Garrett Tanzer in Mirac Suzgun in Eline Visser in Dan Jurafsky in Luke Melas-Kyriazi. 2024. A Benchmark for Learning to Translate a New Language from One Grammar Book. doi:10.48550/arXiv.2309.16575

[4] Wen Lai in Mohsen Mesgar in Alexander Fraser. 2024. LLMs Beyond English: Scaling the Multilingual Capability of LLMs with Cross-Lingual Feedback. doi:10.18653/v1/2024.findings-acl.488

[5] Nathaniel R. Robinson in Perez Ogayo in David R. Mortensen in Graham Neubig. 2023. ChatGPT MT: Competitive for High- (but not Low-) Resource Languages. doi:10.18653/v1/2023.wmt-1.40

[6] Zhuoshi Pan in Qianhui Wu in Huiqiang Jiang in Menglin Xia in Xufang Luo in Jue Zhang in Qingwei Lin in Victor Rühle in Yuqing Yang in Chin-Yew Lin in H. Vicky Zhao in Lili Qiu in Dongmei Zhang. 2024. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. doi:10.18653/v1/2024.findings-acl.57

[7] Haoyang Huang in Tianyi Tang in Dongdong Zhang in Wayne Xin Zhao in Ting Song in Yan Xia in Furu Wei. 2023. Not All Languages Are Created Equal in LLMs: Improving Multilingual Capability by Cross-Lingual-Thought Prompting. doi:10.48550/arXiv.2305.07004

[8] Xiaoyu Chen Jiaxin Guo in Hao Yang in Zongyao Li Daimeng Wei in Hengchao Shang. 2024. A Novel Paradigm Boosting Translation Capabilities of Large Language Models. doi:10.48550/arXiv.2403.11430

[9] Ollama. n.d.. Ollama Documentation. https://docs.ollama.com/. Dostopano: 22. 3. 2026.

[10] Open Language Data. n.d.. Datasets at Hugging Face. https://huggingface.co/datasets/openlanguagedata/flores_plus Dostopano: 3. 5. 2026.

[11] Marko Ulčar, Matej in Robnik-Šikonja. 2023. Sequence-to-sequence pretraining for a less-resourced Slovenian language. *Frontiers in Artificial Intelligence* Izdaja 6 - 2023 (2023). doi:10.3389/frai.2023.932519

[12] Ben in Bansal Mohit Zhang, Shiyue in Frey. 2022. How can NLP Help Revitalize Endangered Languages? A Case Study and Roadmap for the Cherokee Language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Preslav in Villavicencio Aline Muresan, Smaranda in Nakov (Ed.). Association for Computational Linguistics, Dublin, Irska, 1529–1541. doi:10.18653/v1/2022.acl-long.108

[13] Yee in Song Zhenqiao in He Taiqi in Wang William Yang in Li Lei Zhang, Kexun in Choi. 2024. Hire a Linguist!: Learning Endangered Languages in LLMs with In-Context Linguistic Descriptions. In *Findings of the Association for Computational Linguistics: ACL 2024*, Andre in Srikumar Vivek Ku, Lun-Wei in Martins (Ed.). Association for Computational Linguistics, Bangkok, Tajska, 15654–15669. doi:10.18653/v1/2024.findings-acl.925