Carpe Diem spēle Syntra platformā. Uzdevumu krājums studentiem un AI rīkiem

Sastādījis Dainis Zeps, Latvijas Universitāte

Iesākts 2026. gada 11. augustā

Kopsavilkums

Šis darbs tiek veidots kā uzdevumu krājums, kas papildina mācīblīdzekli [1]. Uzdevumi paredzēti gan studentiem, gan AI rīkiem (LLM), turklāt uzdevuma teksts vienlaikus kalpo arī kā prompts LLM. Tādējādi LLM var palīdzēt studentam un piedāvāt iespējamos risinājumus, savukārt students var kritiski izvērtēt, cik sekmīgi LLM risina doto uzdevumu. Uzdevumi ļauj arī salīdzināt un izvērtēt gadījumus, kad LLM, balstoties uz savu uzdevuma interpretāciju, piedāvā būtiski atšķirīgus risinājumus.

Otrs un viens no galvenajiem mērķiem ir piedāvāt studentiem iespēju mācīties pašu izvēlētas valodas, vienlaikus aktīvi iesaistoties arī tādu uzdevumu risināšanā, kuros aplūkotās valodas viņiem ir maz pazīstamas vai pavisam svešas. Šādās situācijās studentam jāliek lietā savas vispārīgās lingvistiskās zināšanas un valodas intuīcija. Viņam jāmācās orientēties situācijā, kurā konkrēto valodu viņš nezina, tomēr, analizējot tās formas, struktūras, paralēles ar citām valodām un AI piedāvātos risinājumus, spēj izvirzīt pieņēmumus un tos pārbaudīt. Tādējādi uzdevumu mērķis nav tikai konkrētas valodas apguve, bet arī vispārīgas lingvistiskās domāšanas attīstīšana — spēja risināt lingvistiskus jautājumus un orientēties valodas situācijās arī tad, ja pati valoda vēl nav apgūta.

Trešais mērķis ir praktiski pētīt un veidot cilvēka un mašīnas sadarbību lingvistisku uzdevumu risināšanā. Students šeit nav tikai AI sniegtās atbildes saņēmējs: viņš formulē uzdevumu, vērtē AI piedāvāto risinājumu, atrod kļūdas un neskaidrības, labo tās un, ja nepieciešams, liek AI meklēt citu risinājumu. Savukārt AI darbojas ne tikai kā informācijas avots, bet kā partneris kopīgā valodas izpētes procesā. Tādējādi šie uzdevumi vienlaikus kļūst arī par eksperimentiem cilvēka un mašīnas sadarbības formu veidošanā.

Abstract

This work is being developed as a collection of exercises accompanying the learning material [1]. The exercises are intended for both students and AI tools (LLMs), with the text of each exercise also serving as a prompt for an LLM. In this way, an LLM can assist the student and suggest possible solutions, while the student can critically evaluate how successfully the LLM solves the

given task. The exercises also make it possible to compare and assess cases in which LLMs, based on their own interpretation of a task, propose substantially different solutions.

A second and central objective is to offer students the opportunity to learn languages of their own choice while also actively engaging with tasks involving languages that are only slightly familiar or entirely unknown to them. In such situations, students are encouraged to draw on their general linguistic knowledge and linguistic intuition. They learn to orient themselves in a situation in which they do not know the particular language but can nevertheless formulate and test hypotheses by analysing its forms and structures, comparing them with other languages, and examining solutions proposed by AI. The aim of these exercises is therefore not only the acquisition of a particular language, but also the development of general linguistic reasoning: the ability to address linguistic questions and find one's way through unfamiliar linguistic situations even when the language itself has not yet been learned.

A third objective is to explore and develop, in practical terms, forms of human–machine collaboration in solving linguistic problems. The student is not merely a recipient of answers supplied by AI: the student formulates the task, evaluates the proposed solution, detects errors and ambiguities, corrects them, and, where necessary, prompts the AI to seek an alternative solution. AI, in turn, functions not only as a source of information but as a partner in a joint process of linguistic inquiry. In this sense, the exercises also become experiments in developing forms of collaboration between humans and machines.

Uzdevumi

1.uzdevums.

Lai veidojam Latīņu mājas lasīšanas paraugam [] interaktīvu lapu pēc parauga [Lx,1], armēņu pasakām. Veidosim Syntra Latin Latvian LATIN, kur ar lielajiem burtiem būs skaņas kolona.

Lai mums teksts jau izraudzīts – lai šajā pirmajā reizē teksts ir viena rindiņa "Ut panem domum ferant canes", blakus būs latviešu tulkojums "Lai maizi mājās nes suņi". Mums ir jau arī Whitaker vārdnīcai atbilstošās datu bāzes izveidota python bibliotēka, par to atsevišķi.

Vispirms izveidosim ar python prog. build_batch.py. Šo, kā arī citus python elementus lūgsim veidot mums piesaistītajam AI rīkam. Mēs taču darbojamies kopā. Šai procedūrai jāizveido json fails ar pieciem ierakstiem, piemēram, piemēram pirmajam vārdam atbildīs ieraksts,

"ut": {"lemma": "","morphology": "","latvian": "", "english": "","note": "" }

Kas atbildīs pirmajam latīņu vārdam 'ut', bet pārējie lauki tukši, jo tur ir "", kas nozīmē, šī json forma vēl nav aizpildīta ar gramatiskajām ziņām.

Nākamajā solī mums šī json forma jāizpilda, ko izdarīsim ar Whitaker vārdnīcas palīdzību, kur attiecīgais python modulis mums ir jau izveidots. AI lūgsim izveidot python procedūru fill_vocab.py, kuru izpildot mūsu json forma aizpildīsies ar gramatiskajām ziņām, piemēram, pirmajam ierakstam atbildīs jau šāds elements

"ut": {"ut": "","morphology": "**V - SG PRES IMP ACT 2P**","latvian": "", "english": "use, make use of, enjoy","note": "" } . Latviešu tulkojums paliek tukšs, jo Whitaker tulko tikai angļu valodā. Pēdējo lauku arī neizmantojam šeit. Kad mums ir gatava jau aizpildītā json forma, veicam trešo soli, lūdzam AI izveidot python procedūru generate_latin.py, kas izveido .html lapu pēc vēlamā parauga, kur vēl palūdzam pievienot Read pogu, lai mums pieejamais balss aģents pievieno izrunu. Iesākumā šo soli varam arī izlaist.

Izveidojiem json formu visai dotajai frāzei/teikumam no pieciem vārdiem.

Tagad, ja izveidotajam docx ar tabulu, kur kreisajā rindā latīņu un labajā latviešu tulkojumi, pielietojam visas trīs mūsu iegūtās python procedūras, tad iegūstam šo https://lingua.id.lv//alex/Ut_panem_reader.html

Lai šis kopskats, ko redzam, ir arī piederīgs pie kopējā prompta, ko mēs dosim mūsu palīgiem, AI rīkiem, lai no visa kopā ņemot, viņi izveido šo iznākumu, kā testu, šai pieejai, ko te klāstam. Tālāk jau teksta parauga vietā liekam citus tekstus un izmēģinām. Piemēram, https://lingua.id.lv/alex/Poggii_facetiae_reader.html.

Visas trīs procedūras, ko darbināsim, ir šīs zemāk. (Par Whitaker datu bāzes pievienojamo bibliotēku mums vēl jārunā atsevišķi, ko ņemsim no https://lingua.id.lv/alex/lingua-dicttools.html.)

Šīs procedūras darbinām, lai iegūtu rezultātā vēlamo .html lapu.

python.exe build_batch.py

python.exe fill_vocab.py

python.exe generate_latin.py

Jautājums: piedāvātajā risinājumā kļūda ieviesusies, jo ‘ut’ ir divas interpretācijas iespējas, kā verbu un kā saikli. Sistēma izvēlējusies verbu un to mēs varētu uzlūkot kā valodas kļūdu, bet sistēma nav risinājusi jautājumu, ko darīt šādu paralēl-situāciju gadījumā. Vai to vajag risināt? Un vai ir vienkāršs kāds risinājums, kas derētu šajā situācijā?

2.uzdevums

Lai risinām līdzīgu uzdevumu kā iepriekšējā uzdevumā, bet tagad veidosim

Syntra Greek Latin GREEK izkārtojumu

ar dotu tekstu **:** *Ὅπως οἱ κύνες τὸν ἄρτον οἴκαδε φέρωσιν,* t.i. iepriekšējais teikums tikai grieķu valodā.

Bet atšķirībā no iepriekšējā uzdevuma pieņemsim, ka mums nepieciešamās Greek WDB nav, bet mēs to veidosim paši. Kā mums būs tas jādara (šī uzdevuma rāmjos)?

Izveidojam neaizpildītu json tekstam, tā formām pēc parauga

" *Ὅπως* ": {"lemma": "","morphology": "","latin": "" } .

Tā kā mums WDB vēl nav, tad darbinām tikai pēdējo no trīs procedūrām generate.py, ko pasūtīsim izveidot LLM. Rezultātā mums jāiegūst līdzīga kā iepriekšējā uzdevumā html lapa, kur peldošie ekrāniņi uz teikuma formām rādīs neaizpildītu gramatisko informāciju, jo nelietojām WDB. Bet tagad darbosimies pretēji kā iepriekšējā uzdevumā un paši aizpildīsim WDB informāciju, t.i, uzbraucot ar peli uz formas un tur klikšķinot, gramatiskās informācijas laukā pa labi, kas mums veidojās statistiski, neesošās informācijas vietā ierakstīsim Whitaker gramatisko rindu, e.g. šajā piemērā priekš vārdiņa CONJ vai kaut kā tamlīdzīgi, kāda tā parādījās iepriekšējā uzdevumā vārdiņam ‘ut’ . Kad šo esam ierakstījuši gramatiskajos laukos, lai sistēma to atceras, un kad nākamreizi skatīsim ar peli šo formu, tur parādīsies mūsu ievietotā informācija. Šajā solī mēs vēl nepārbaudām, vai mūsu ierakstītā informācija ir pareiza. Jā, mēs it kā veidojam mūsu darinātās WDB no mūsu gramatisko zināšanu krātuves. Kad aizpildīta gramatiskā informācija visiem pieciem vārdiem, uzdevums ir izpildīts un nododams pārbaudei.

Jautājums: kā saglabāt studenta izveidoto WDB? (Meklējiet atbildi iepriekšējā uzdevumā.)

Jautājums: kuru valodu students zina un kuru mācās pēc šī uzdevuma nostādnes?

Literatūra

Mācīšanās platforma Syntra un lingvistiskā spēle Carpe Diem: piemēri, vingrinājumi, eksaminēšanas plāni un MI atbalstīti programmatūras izstrādes piemēri, The Syntra Learning Platform and the Carpe Diem Language-Learning Game: Examples, Exercises, Assessment Plans, and AI-Assisted Software Development, sast. D.Zeps, https://lingua.id.lv/art/war.101.pdf, https://scireprints.lu.lv/569/, 2026