PRAKTILINE ÕPPERAAMAT

Keelemudelite treenimine

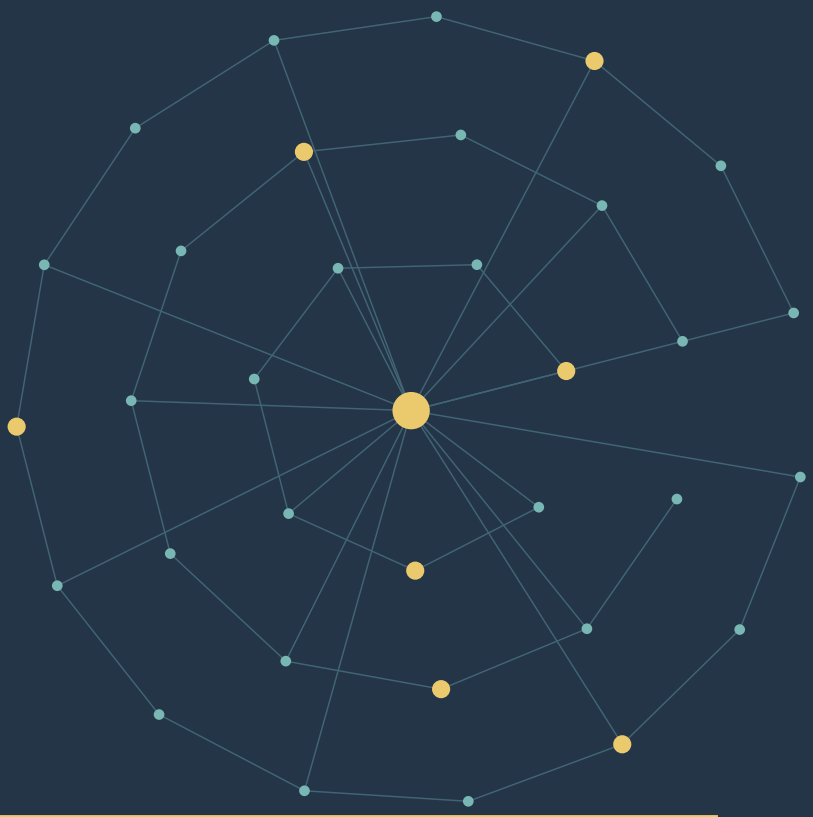
Arusaamisest esimese kontrollitud katseni

Mõista õppimist.

Valmista andmed.

Treeni kohalikult.

Kontrolli tulemust.



Ubuntu + NVIDIA / Apple Silicon + MLX

Autori katsekogemus, vead ja lahendatud harjutused

Pert Lomp

SEPTEMBER 2026

Raamatust ja kasutamisest

Põhimõtted, kogemus ja praktiline labor

Pert Lomp · Täiendatud õpperaamat · september 2026

Kuidas mudel õpib, kuidas kujuneb tema käitumine ja kuidas õpetada talle paremat eesti keelt.

See raamat kasvas välja ühe kohaliku keelemudeli arendamisest: andmete kogumisest, katsetest, vigadest ja nende parandamisest. Praktiline juhtum on Qwen3.8-27B eestikeelne kohandamine ühe RTX 5090 abil. Juhtumi väärtus on jälgitavas õppimisprotsessis, mitte lubaduses, et sama retsept annab igal mudelil sama tulemuse.

Täiendatud väljaanne ühendab algse 50-leheküljelise õppematerjali ja arutelu inimese ning keelemudeli õppimise sarnasustest. Ettekandeks mõeldud kordused on koondatud, tehnilised väited täpsustatud ning lisatud on harjutused, vastused, töölehed ning Ubuntu ja Maci praktilised laborid.

Kellele see raamat on
Alustajale, kes tahab aru saada, mida treeningus tehakse; keele- ja valdkonnaeksperdile, kes soovib kaasa aidata; arendajale, kes tahab oma esimest katset mõtestatult korraldada. Alguses pole vaja teada ühtegi lühendit. Praktilised peatükid eeldavad hiljem põhiteadmisi failidest, Pythonist ja oma arvuti kasutamisest.

Faktikontrolli seis: 5.-6. september 2026. Projektitulemused pärinevad autori algmaterjalist. Selle väljaande koostamisel ei käivitatud treeninguid uuesti ega auditeeritud kõiki mudelifaile, algandmeid ja logisid. Arvutusega kontrollitavad seosed on üle arvutatud; üldised tehnilised selgitused on kõrvutatud viidatud algallikate ja ametliku dokumentatsiooniga.

Sisukord

Lugejale. Kuidas sellest raamatust õppida

Sa ei pea raamatut ühe õhtuga läbi töötama. Esimesel läbimisel otsi seoseid. Teisel korral lahenda ülesandeid. Kolmandal kasuta töölehti oma projekti juures. Kui mõni lühend ununeb, leiad selle lõpus olevast sõnastikust.

Lugemistee	Millele keskenduda
Tahan esmalt mõista	Peatükid 1-3: õppimise võrdlus, mudeli töö ja treeninguetapid.
Tahan teha esimest katset	Peatükid 4-9, 11, 15 ja 17-20: andmed, riistvara, adapter, katseplaan ning käivitatavad laborijuhised.
Tahan parandada olemasolevat mudelit	Peatükid 8-14: mõõtmine, keel, dialoog, riistvara, pakendamine ja juhtumianalüüs.
Tahan õpetada teisi	Kasuta peatükkide küsimusi, lahendatud näiteid ning lõpus olevaid vastuseid ja töölehti.

Kolm märgist, mis hoiavad väited ausad

„Põhimõte” tähistab tehnilist selgitust või laialt kasutatavat meetodit. „Projektivaatlus” tähendab selles raamatus kirjeldatud ühe projekti tulemust, mille üldistamise ulatus on piiratud. „Katseidee” on ettepanek tulevaseks tööks; see ei ole juba tõestatud paranemine.

Õppimiseks kasuta lihtsat rütmi: loe üks alapeatükk, sulge tekst ja seleta mõte oma sõnadega. Seejärel vaata, mis jäi puudu. Meenutamine sunnib teadmise ise taastama ning aitab eristada tuttavana tunduvat teksti sellest, mida oskad kasutada. Sellist õppimisvõtet käsitleb ka meenutamisharjutuste uurimuslik ja pedagoogiline kirjandus. [25]

Proovi kolme läbimist
Täna selgita mõte ühe lausega. Homme lahenda selle kohta uus näide. Nädala pärast seo see oma projektiga. Ajavahed on siinne praktiline soovitus, mitte kõigile optimaalne õppegraafik.

Raamatu läbiv ülesanne

Kujutleme eestikeelset keeleabilist. Ta peab oskama fraase käänata, lauseid parandada, selgitada oma parandust ning jätkata vestlust. Vahel annab ta õige sõnavormi, aga rikub vastuse kuju; vahel on tema vastus õige, kuid automaatne hindaja ei tunne rööpvormi. Iga peatükk aitab eristada, millist osa süsteemist tuleks parandada.

Harjutused on nummerdatud H1-H20. Vastused on raamatu lõpus. Proovi vastata enne lahenduse vaatamist. Kui sinu põhjendus erineb pakutust, kontrolli eeldusi: mitmel ülesandel on rohkem kui üks põhjendatud lahendus.

1. I vaatus: kuidas keelemudel kasvab

Selle peatüki järel: oskad selgitada, milles sarnaneb mudeli õpetamine lapse kasvatamisega ja kus see võrdlus lõpeb.

Laps kuuleb keelt ammu enne, kui oskab seletada grammatikat. Ta jälgib vanemaid, kuulab vestlusi ja seostab sõnu olukordadega. Ta märkab, kuidas palutakse, vaieldakse, lohutatakse ja naljatatakse. Kodust, sõpradelt ja koolist saab ta nii teadmisi kui ka käitumise eeskujusid.

Keelemudelitega töötades olen hakanud mõtlema samale küsimusele: mida me oma mudelile tegelikult õpetame? Valime materjali, näitame vastuseid ja anname tagasisidet. Need valikud mõjutavad, milliseid seoseid mudel õpib ning kuidas ta inimestega suhtleb. Õpetamise võrdlus aitab näha vastutust ka siis, kui töövahenditeks on tekstifailid ja videokaart.

Sarnasus seisneb õppimise üldistes põhimõtetes: varasem materjal kujundab sisemisi esitusi, kontekst muudab tõlgendust ning õpitut saab rakendada uuele. Laps ei pea kõiki tulevasi lauseid pähe õppima. Ka mudel võib koostada lause, mida treeningus täpselt sellisena ei olnud. Seda nimetame üldistamiseks.

Sarnased põhimõtted, erinevad mehhanismid

Inimese ajus kujunevad seosed elavate närvirakkude, keha, suhete ja keskkonnaga tegutsemise kaudu. Keelemudeli treeningus muudab optimeerimisalgoritm arvulisi parameetreid. See ei tõesta, et aju kasutab sama õppimisalgoritmi või et mudeli üks kaal vastab inimese ühele mälestusele.

Keele ennustamine on huvitav kokkupuutepunkt. Inimesed aimavad lausete jätku; levinud keelemudeli eeltreening kasutab järgmise tokeni ennustamist. Uuringutes on leitud seoseid mudelite esituste ja inimaju keele töötlemise vahel. Samas võib osa näilisest sarnasusest tuleneda analüüsimeetodist või kõrvalteguritest. Seos ei tõesta samasugust mõtlemist. [1, 2]

Kasvatamise võrdlus	Mudeli juures aitab mõista	Piir, mida meeles hoida
Keelekeskkond	Andmevalik kujundab õpitavaid mustreid.	Tekst kirjeldab elu; see ei anna automaatselt elatud kogemust.
Harjutamine	Treeningnäited ja kordamine mõjutavad sooritust.	Rohkem kordusi ei taga paremat üldistamist.
Õpetaja tagasiside	Näited ja eelistused kujundavad vastamist.	Tasufunktsioon ei võrdu inimese väärtushinnangute tervikuga.
Teatmeteose kasutamine	RAG toob vastamiseks väliseid allikaid.	Dokumendi leidmine ei ole kaalude treenimine.
Eksam	Uued ülesanded kontrollivad ülekannet.	Korduvalt arendust juhtinud eksam pole enam sõltumatu test.

Vundament ja suund

Nullist treenitaval mudelil puudub alguses õpitud keeleoskus. Olemasoleva mudeli kohandamisel alustame aga juba arenenud võimetest. Siis sarnaneb töö pigem täiendõppega: tahame parandada üht keelt, eriala või töövõtet olemasolevat rikkumata.

Hariduse „kihid” on kasulik kujund, kuid neid ei tohi segi ajada närvivõrgu arhitektuursete kihtidega. Iga uus tekstikogu ei loo mudelisse eraldi ajaloo- või kultuurisahtlit. Hilisem treening muudab ja kasutab paljusid ühiseid seoseid.

Ka keeleõpet ja kultuuri ei saa täielikult lahutada. Ametlik kiri, sõbralik vestlus ja ilukirjandus õpetavad erinevaid väljendusviise. Eesti keelt arendades on oluline näidata mitmekesist kasutust, mitte eeldada ühte „õiget eestlase maailmapilti”. Keeles elab palju põlvkondi, piirkondi ja vaatenurki.

Kas viisakus tähendab tundeid?

Mudel võib õppida toetavat, järsku või halvustavat vastamist. Agressiivse kõne nägemine andmetes ei tähenda tingimata selle omaksvõttu: sama materjal võib õpetada solvanguid ära tundma. Vastamist mõjutavad ka hilisem treening, süsteemijuhised ja praegune ülesanne.

Viisakas lause ei tõenda head kavatsust ega tunnet. Vihase tegelase dialoog ei tõenda viha kogemist. Emotsiooni äratundmine, olukorra tõlgendamine, hooliv käitumine ja subjektiivne kogemus on erinevad küsimused. Kas tehisintellekt võiks midagi subjektiivselt kogeda, on jätkuvalt vaieldav.

Kaamera, mikrofon ja robot võivad anda juurde maailmaga seotud sisendit. Püsiv õppimine vajab siiski mehhanismi, mis kogemuse põhjal süsteemi muudab. Meeleelundite sarnaste seadmete lisamine ei lahenda iseenesest õppimise ega teadvuse küsimust.

Kasvataja vastutus
Treenija kujundab käitumissuunda. Hea eesmärk on tõepärane, parandatav ja inimest austav abiline, kes suudab ka lugupidavalt mitte nõustuda. „Õpetame head” tuleb tõlkida nähtavateks hindamiskriteeriumideks.

H1. Õppimine või ligipääs?
Lisad abilise otsingukogusse sada raamatut ning ta vastab nende põhjal paremini. Kas mudeli kaalud muutusid? Milline osa lapse õppimise võrdlusest siin aitab ja milline eksitab?

2. Mis toimub mudeli sees?

Selle peatüki järel: eristad tokenit, kaalu, konteksti ja treeningusammu. Oskad kirjeldada järgmistokeni treeningut ilma liigse lihtsustuseta.

Suur keelemudel ehk LLM (large language model) on arvuline süsteem, mis töötleb keelt. Selles raamatus keskendume autoregressiivsetele mudelitele: teksti genereerimisel arvutavad nad konteksti põhjal järgmise tokeni tõenäosused ning jätkavad teksti sammhaaval. See on genereerimise kirjeldus, mitte kogu mudeli võimekuse täielik seletus.

Vestlus, tõlkimine ja programmeerimine kasutavad treeningus kujunenud esitusi ning sageli ka järeltreeningut ja ümbritsevat tarkvara. Sõnastus „mudel teeb ainult üht asja” on alustamiseks meeldejääv, kuid jätab välja multimodaalse sisendi, teistsugused treeningueesmärgid ja tööriistadega süsteemi.

Token ei ole tingimata sõna

Tokeniseerija teisendab teksti ühikute jadaks. Token võib vastata sõnale, sõnaosale, märgile või baitidest moodustatud tükile. Tükeldus sõltub mudelist, kontekstist ja ka eelnevast tühikust. Seetõttu tuleb näiteid mõõta täpse tokeniseerijaga, mitte oletada sõna kuju põhjal.

Algmaterjalis raporteeritud valimil oli eesti sõna kohta 2,37 ja inglise sõna kohta 1,11 tokenit. Suhe on ligikaudu 2,14. See kirjeldab seda valimit ja tokeniseerijat. See ei tõesta, et kõik samasisulised tekstid, mudelid või API-arved oleksid eesti keeles täpselt kaks korda suuremad.

Miks see oluline on
Tokenite arv mõjutab konteksti mahutavust, treeningu mahtu ja sageli kasutamise hinda. Rohkemateks tükkideks jagunemine võib teha ülesande raskemaks, kuid ühest tükeldusnäitest ei saa järeldada õigekirjavea põhjust.

Kaalud ja arvutused

Parameetrid ehk kaalud on mudeli sisemised arvud. Teadmised ja oskused on nende kaudu hajusalt esitatud, mitte tavalise andmebaasi üksikkirjetena. Mudel võib siiski treeningteksti meelde jätta ja seda hiljem sõna-sõnalt väljastada. Seda on katseliselt näidatud. [3]

Paljud mudelid kasutavad Transformer-arhitektuuri tähelepanumehhanismi, mis seob sisendi eri kohti. Tähelepanu ei tähenda siin teadlikku tähelepanu. Esineb ka hübriidarhitektuure: Qwen3.8-27B ametlik kirjeldus ühendab täieliku tähelepanu ja Gated DeltaNeti plokke. Arhitektuuri nimi qwen3_5 ei ole seetõttu iseenesest vastuolus mudeli versiooninimega 3.8. [4, 5]

Üks õppimistsükkel

1. Tekst tokeniseeritakse ja moodustatakse sisend ning sihtjätk.
2. Mudel arvutab, kui tõenäolised on järgmised tokenid. Kausaalne mask takistab õigete tulevaste tokenite ette nägemist.
3. Kaofunktsioon mõõdab ennustuse sobivust treeningu sihtidega.

4. Tagasilevi arvutab treenitavate parameetrite gradiendid ehk muutmissuunaga seotud tuletised.

5. Optimeerija kasutab gradiente ja õpisammu parameetrite uuendamiseks.

Treeningus saab ühe tekstijada paljude positsioonide viga arvutada paralleelselt. Seega ei pea protsessi ette kujutama nii, et kogu arvuti näeb korraga ainult ühte peidetud sõna. Genereerimine ja treeningu arvutuskorraldus erinevad.

Kadu ehk loss mõõdab valitud eesmärki. Järgmise tokeni kadu ei mõõda otseselt tõde, headust ega vestluse kasulikkust. Ka valeväidet sisaldava teksti jätku saab edukalt ennustada.

Kasutamine ja kontekst

Tavalise kasutamise ehk inferentsi ajal kaale ei muudeta. Mudel võib kasutada sinu parandust käesoleva vestluse kontekstis, kuid see ei tähenda püsivat kaalumuutust. Väline mälu võib paranduse salvestada ja hiljem uuesti ette anda.

Kontekstiaken piirab korraga töödeldava jada mahtu. Serveri seadistus, väljundile jäetav ruum ja riistvara võivad kasutatavat mahtu vähendada. Pikk lubatud kontekst ei taga, et mudel kasutab kõiki seal olevaid fakte võrdselt hästi.

H2. Üks lause, kolm mehhanismi
Selgita erinevust: parandad vastust vestluses; salvestad paranduse mälufaili; treenid selle põhjal adapterit. Millal muutuvad mudeli treenitavad parameetrid?

3. Millist õpetamist sinu mudel vajab?

Selle peatüki järel: valid probleemi järgi viiba, otsingu, tööriista või treeningu. Mõistad eeltreeningu, CPT, SFT ja eelistusõppe eri eesmärke.

Alusta puudusest, mida tahad parandada. „Mudel võiks targem olla” on liiga lai. „Mudel peab moodustama fraasi mitmuse osastava ja vastama ainult vormiga” on juba kontrollitav eesmärk.

Puudus	Esimene mõistlik kontroll	Võimalik järgmine samm
Vastus on liiga pikk	Selgem juhis ja väljundinäide.	SFT, kui käitumine ei püsi eri ülesannetes.
Vajalik fakt muutub sageli	Otsing ehk RAG ja allikaviide.	Paranda otsingut ning allikate kasutamist.
Arvutus läheb valesti	Kalkulaator või koodi käivitamine.	Õpeta tööriista õiget kasutamist.
Eesti keel on kohmakas	Lähtemudel, mall ja tüüpilised vead.	Keeleandmed, CPT või sihitud SFT.
Vastus ignoreerib eelmist vooru	Vestlusajalugu, mall ja kontekstipiir.	Mitmevoorulised näited ning hindamine.

Eeltreening: laia aluse loomine

Nullist eeltreening algab lähtestatud parameetritest ning kujundab põhivõimeid. Suure tippmudeli puhul võib see nõuda tohutut andme- ja arvutusmahtu. Väikese õppemudeli eeltreeningut saab aga teha ka piiratud riistvaral. Väide „seda teevad ainult suured laborid” kehtib vaid teatud mastaabi kohta.

Chinchilla uuring käsitles mudeli suuruse ja tokenite arvu tasakaalu kindla arvutuseelarve juures. Sellest tuntud ligikaudne suhe ei ole ühe keele õpetamiseks vajalik miinimum. Tootmises loeb ka hilisem kasutuskulu, mistõttu sobiv treeningumaht sõltub eesmärgist. [6]

CPT: olemasoleva mudeli täiendav eeltreening

Continued pretraining kasutab olemasolevat mudelit ja valitud materjali, sageli järgmistokeni eesmärgiga. Eestikeelne proosa, erialatekst või mitmekesine keelekorpus võib aidata kohaneda nende tekstide mustritega. Kas paraneb ka vestlus või mõni konkreetne oskus, tuleb eraldi kontrollida. Valdkonnaga kohandamise kasu on näidatud, kuid see ei taga iga kohaliku retsepti edu. [7]

CPT võib mõjutada ka teadmisi ja juhiste järgimist. SFT võib mõjutada ka keelt. Seetõttu on „CPT annab keele, SFT annab oskused” kasulik algne suunaviit, mitte range jaotus. Toortekstil jätkamine võib vestlusmudeli käitumist nõrgendada; see ei juhtu igas katses vältimatult.

SFT: soovitud soorituse näitamine

Supervised fine-tuning ehk juhendatud peenhäälestus kasutab soovitud väljundi näiteid. Need võivad olla küsimuse-vastuse paarid, vestlused, tööriistakutsed või muu ülesandepõhine tekst. Hea näide näitab nii sisu kui ka vormi: mida vastata ja kuidas vastata.

Näide
Kasutaja: „Pane fraas «habras tasakaal» mitmuse osastavasse. Vasta ainult vormiga.” Soovitud vastus: „hapraid tasakaale”. Kontrolli eraldi keelelist õigsust ja seda, kas mudel järgis vormingujuhist.

SFT võib õpetada ka viisakust, lühidust ja ebakindluse väljendamist. Eelistusõpe ei ole nende omaduste ainus õpetamise viis. Inimeste demonstratsioonide ja tagasiside mõju juhiste järgimisele on näidatud näiteks InstructGPT töös. [8]

Eelistusõpe: parema vastuse soosimine

DPO võrdleb sama sisendi eelistatud ja vähem eelistatud vastuseid ning kohandab nende suhtelist tõenäosust viitemudeli suhtes. Eelistatud vastuste sisu on samuti õppematerjal. DPO ei ole piiratud mudeli enda värskete vigadega; nende kasutamine võib olla kasulik andmevalik, mitte meetodi nõue. [9]

RLHF on laiem inimtagasisidet kasutava stiimulõppe käsitlus. Klassikaline PPO-põhine teostus kasutab tasumudelit ja muid komponente; mälus korraga olevate mudelite arv sõltub teostusest. ORPO ja KTO on alternatiivsed järeltreeningu meetodid, mille andmenõuded ja eesmärgid erinevad. Nende kasutamine ei ole kohustuslik järgmine aste.

Järjekord on otsus
Eeltreening, CPT, SFT ja eelistusõpe ei ole neli kohustuslikku kooliklassi. Võid alustada olemasoleva instruct-mudeli SFT-st. Võid enne treeningut lahendada probleemi tööriistaga. Iga lisafaas vajab põhjendust ja mõõtmist.

H3. Vali sekkumine
Keeleabiline vastab grammatiliselt hästi, kuid kasutab vana hinnakirja. Teine mudel tunneb hinnakirja, kuid eksib fraaside ühildumisel. Kummal juhul alustaksid RAG-ist ja kummal juhul keeleoskuse treeningust?

4. Andmed: millise keskkonna sa lood?

Selle peatüki järel: koostad andmevaliku eesmärgi järgi ja eristad kvaliteeti, mitmekesisust, õigusi ning mõõtmise sõltumatust.

Treeningkorpus on mudeli õppekeskkond. Suur fail ei pruugi sisaldada palju uut infot: seal võib olla korduvaid artikleid, menüüsid, tekstituvastusvigu või ebaühtlasi tõlkeid. Ka toimetatud raamat võib sisaldada aegunud väiteid, stereotüüpe ja autori stiili, mida sa igasse vastusesse ei soovi.

Vali materjal ülesande järgi. Kirjandus sobib uurima jutustamist; dialoog vestluse jätkamist; erialased seletused mõistekasutust. „Väike kvaliteetne võidab alati suure” on sama eksitav kui „rohkem on alati parem”. Mõõda andmete marginaalset kasu, mitmekesisust ja kordusi. Mitmekeelsete korpuste puhastamise mõju ei ole kõigis keeltes ühesugune. [10]

Ühe kirje saateleht

Väli	Milleks see vajalik on
record_id ja source_id	Kirje ning algdokumendi püsiv tuvastamine.
Päritolu ja kuupäev	Kust sisu saadi ning millist versiooni kasutati.
Õiguslik alus ja piirangud	Treening, säilitamine ja edasijagamine võivad vajada eri otsuseid.
Keel, žanr, ülesandetüüp	Valiku ja hilisema tulemuse analüüsimiseks.
Jaotus: train/dev/test	Et eri otstarbega materjal kogemata ei seguneks.
Puhastused ja tokeniarv	Et näha eemaldamisi, kärpimist ja tegelikku mahtu.
Kontrollija ja veamärgis	Et parandatud näite päritolu oleks hiljem jälgitav.

Kordusi tuleb otsida mitmel tasandil

Normaliseeritud täisteksti räsi leiab identsed koopiad. Ainult lõigu alguse räsi võib eri tekste ekslikult kokku lugeda ja muudetud algusega koopiad leidmata jätta. Ligikaudne võrdlus leiab sarnaseid sõnajadasid. Dokumendi- või teosetunnus aitab hoida sama allika eri lõike lahus.

Jagamise ühik peab vastama üldistusväitele. Kui tahad hinnata uusi teoseid, eralda teosed. Kui tahad hinnata uusi autoreid, eralda autorid. Sama autori teise teose olemasolu treeningus ei muuda iga tulemust automaatselt kehtetuks, kuid piirab seda, mida „uus” tähendab.

Sünteetilised näited vajavad välist kontrolli

Mudel võib aidata luua ülesandevariante ja vastuseid. Hoia alles päritolu: milline generaator, viip, mudeliversioon ja kontroll. Mitme kandidaadi genereerimine ning kontrolli läbivate vastuste valimine võib parandada andmeid, kuid kontrolli läbimine ei tõesta veatust.

Keelemudel kohtunikuna aitab sõeluda stiili või asjakohasust. Tal võivad olla pikkuse, järjekorra ja oma väljendusviisi eelistused. Vaheta võrdlusvastuste järjekorda, kasuta üheselt kirjeldatud kriteeriume ja võrdle valimit inimhinnangutega. [11]

Vabamorf aitab vormide analüüsi ja sünteesi kontrollida, kuid ei lahenda üksinda loomulikkust, tähendust ja kontekstisobivust. Mudel toodab oma vastused ja vead; usaldusväärse paranduse allikas peab olema eraldi põhjendatud.

Replay: varasemate oskuste hoidmine

Uue materjali kõrval võib kasutada säilitatavaid võimeid esindavat kordusmaterjali ehk replay'd. Sobiv osakaal ei ole universaalne 10 või 20 protsenti. Katseta eesmärgi ja eelarve järgi ning mõõda ka säilitatavaid oskusi.

Projektivaatlus
Algmaterjali järgi oli CPT kordusmaterjali osakaal 3,1% sõnadest, SFT-s 15-18% näidetest. Need on eri nimetajatega arvud. Kui näidete pikkus erineb, võib tokenipõhine osakaal neist mõlemast erineda.

H4. Andmete jagamine
Sul on ühe romaani 1000 lõiku. Loosid 950 treeningusse ja 50 testi. Kas see on hea katse, kui soovid väita, et mudel kirjutab paremini uute autorite stiilist sõltumata? Kuidas jagaksid teisiti?

5. Andmetest treeningpartiini

Selle peatüki järel: kontrollid tokeniseerimist, vestlusmalli, kaomaski ja pakkimist enne pikka treeningut.

Andmefail võib olla korrektne ja treening siiski vale. Tekst läbib mitu teisendust: rolliväljad muutuvad vestlusmalliks, mall tokeniteks, tokenid partiideks ja osa positsioone jäetakse kaost välja. Kontrollida tuleb seda, mida mudel lõpuks näeb.

Vestlusmall on osa mudeli sisendist

Vestlusmall märgib kasutaja, assistendi ja võimaliku süsteemijuhise piirid. Eri mudelid kasutavad eri erimärke ning mõnikord ka mõtlemis- või tööriistavälju. Malli olemasolu üksi ei tõesta, et mudel on vestluseks treenitud; seda kinnitab eeskätt mudeli päritolu ja mudelikaart.

Kasuta täpse lähtemudeli malli ja kontrolli, et treeningu ning kasutamise vorming sobivad. Prindi mõni näide lahti pärast kogu eeltöötlust. Vaata, kas kasutaja küsimus, soovitud vastus ja lõputähis asuvad õiges kohas. See on odavam kui hiljem kajamise põhjust oletada.

Kaomask ja tähelepanumask täidavad eri ülesandeid

Kaomask määrab, millistel tokenitel viga arvutatakse. Assistendi vastuste põhine kadu on vestlusõppes levinud valik, kuid kogu jada põhine kadu ei ole iseenesest vigane meetod. Valik peab vastama eesmärgile ja teostusele.

Tähelepanumask määrab, milline token saab millist varasemat infot kasutada. Kasutaja tokeni kaost väljajätmine ei tee kasutaja teksti mudelile nähtamatuks: vastus peab ju küsimusest sõltuma. Need kaks maski lahendavad erinevaid probleeme.

TRL eristab muu hulgas completion-only ja assistant-only õpet. Käitumine sõltub andmevormingust, vestlusmalli maskitoest, mudeli laadimisteest ja tarkvaraversioonist. Multimodaalse mudeli puhul ära eelda, et tekstimudeli näide töötab muutmata. Kontrolli üht tegelikku partiid. [12]

Mida vaatad?	Mida tahad näha?
input_ids lahtidekodeering	Õige vestlus, rollipiirid ja erimärgid.
labels / kaomask	Soovitud vastusel on õppimissignaal; täidet ei hinnata.
Jada pikkus enne ja pärast	Vastus pole kärbitud ning maht mahub aknasse.
Näite piirid pakkimisel	Eraldamine vastab valitud treeninguviisile.
Jada lõpp	Lõputähis vastab mudeli oodatud lõpetamisele.

Pakkimine ja kärpimine

Pakkimine ehk packing ühendab lühikesi näiteid tõhusamaks töötluseks. Kui 50 tokenit täidetakse alati 2048-ni, võib täite osakaal olla 97,6%. Dünaamiline täitmine ja täitevabad kernelid muudavad aga tegelikku raiskamist; seda protsenti ei saa tuletada ainult max_length väärtusest.

Eraldaja ütleb mudelile, kus tekst lõpeb, kuid ei blokeeri ise tähelepanu. Sõltumatute näidete pakkimisel võib vaja olla plokkide eraldamist tähelepanus või vastavat jada oleku lähtestamist. Hübriidarhitektuuris tuleb kontrollida ka muid jadamehhanisme. Kõik raamistikud ei tee seda ühtemoodi ning kõigis eeltreeninguretseptides ei taotleta sama eraldust.

Kõige kindlam pikkusühik on tegelik tokenijada. Tähemärkide või sõnade järgi tehtud lähendus sobib esialgseks eelarveks, kuid mitte lõplikuks pakkimispiiriks. Vastasel juhul võib lõikamine eemaldada süstemaatiliselt iga teksti lõpu.

Kolm tokeniarvu
Hoia eraldi korpuse tokenid, mudelile sisendina antud mitte-täitetokenid ja kaos hinnatavad tokenid. SFT-s võivad need oluliselt erineda. Järgmistokeni nihutamine, maskid ja viimase partii pikkus muudavad täpset arvu.

H5. Vaikne andmekadu
8000 tähemärgiga tükk sisaldab keskmiselt 2900 tokenit. Aken on 2048. Kui palju jääb selle tüki tokenitest välja? Kas see protsent tõendab sama kadu kogu korpuses?

6. LoRA ja QLoRA: õpetamine piiratud mäluga

Selle peatüki järel: mõistad adapteri, rank'i, alpha ja kvantimise rolli ning oskad hinnata, millist osa mudelist treenid.

Täielikul peenhäälestusel on kõik mudeli kaalud treenitavad. See võib olla kallis, kuid väikese mudeli puhul ei eelda andmekeskust. Parameetritõhus peenhäälestus ehk PEFT vähendab treenitavate parameetrite hulka. LoRA on üks selle meetodipere liik.

Kaks väikest maatriksit suure muudatuse jaoks

LoRA külmutab valitud algkaalumaatriksi ja esitab selle uuenduse kahe väiksema maatriksi korrutisena. Tavapärane kuju on W_uus = W + (alpha/r) × B × A. Rank r piirab uuenduse astakut; see ei ole mudeli intelligentsuse skaala. [13]

Lahendatud arvutus
5120 × 5120 maatriksis on 26 214 400 arvu. Kui r = 16, sisaldavad kaks LoRA-maatriksit kokku 5120 × 16 + 16 × 5120 = 163 840 arvu. Uuenduse parameetreid on 160 korda vähem. See suhe kehtib sellele maatriksile, mitte automaatselt kogu mudeli mälukulule.

Alpha määrab koos rank'i ja meetodi variandiga skaleerimise. Tavaline LoRA kasutab alpha/r; rsLoRA kasutab alpha/ruutjuur(r). Seetõttu võib rank'i muutmine muuta korraga nii uuenduse väljendusvõimet kui ka skaalat. Õpisamm ja sihtmoodulid mõjutavad tulemust samuti. [14]

Adapteri suurust ei saa öelda ainult rank'i järgi. Loe kokku sihtmoodulid, treenitavad parameetrid ning nende osakaal. Kui mõni jadaplokk kasutab teisi moodulinimesid, võib lühike target_modules loend selle vahele jätta.

Projektivaatlus
Algmaterjalis on r = 32 adapteri suuruseks 159,4 miljonit parameetrit ehk ligikaudu 0,574% 27,78 miljardist. Raporteeritud sihtmoodulid katavad edasisideplokke ja osa tähelepanuprojektsioone. Täpset katvust tuleb kontrollida kohalikust adapterist, mitte tuletada mudeli turundusnimest.

QLoRA vähendab eelkõige külmutatud baasi mälukulu

QLoRA kasutab madala bitisügavusega algkaale ning treenib adaptereid suuremas arvutustäpsuses. Algne töö kasutas NF4 vormingut, kvantimise abiandmete täiendavat kvantimist ja lehitsetavaid optimeerijaid. Uurimuse head tulemused ei garanteeri kadudeta kohandamist igal mudelil. [15]

Hoia eraldi kaalude säilitusvorming, arvutuste andmetüüp, adapteri salvestusvorming ja optimeerija olekud. „Baas 4, adapter 16, optimeerija 32 bitti” ei ole universaalne retsept. Raamistik võib adapteri FP32-sse tõsta; 8-bitine optimeerija ei hoia kõiki suuri olekuid FP32-s. Väikesi olekuid võidakse siiski säilitada suuremas täpsuses.

Failisuurus on kontrollküsimus
159,4 miljonit parameetrit võtavad 2 baidiga umbes 318,8 MB, 4 baidiga 637,6 MB. Algmaterjalis nimetatud ligikaudu 640 MB adapter vajab seetõttu andmetüüpide kontrolli. See ei tõesta üksi viga: failis võivad olla muud tensorid või on kasutatud teist mõõtühikut.

Liitmine ja lähteversioon

Adapter üksi ei ole terviklik mudel. Ta eeldab sobivat lähtemudelit ja selle kaale. Adapteri võib toetatud juhul algkaaludesse sisse arvutada, kuid kontrolli arvutustäpsust ja tulemust. Kui SFT tehti CPT-ga liidetud mudelile, ei saa SFT-adapterit automaatselt panna algsele muutmata mudelile.

Ühe projekti r = 16 ja r = 32 võrdlusest ei sünni üldreeglit. Võrdle samast lähtepunktist, võrreldava eelarve ja mõlemale sobiva seadistusega. Suurem rank ei eelda alati rohkem andmeid ega põhjusta vältimatult ülesobitumist.

H6. Adapteri kontroll
Sul on sama rank'iga kaks adapterit, kuid üks on kolm korda suurem. Nimeta kaks võimalikku põhjust ja üks failidest kontrollitav tõend.

7. Seadistused: mida iga nupp muudab?

Selle peatüki järel: oskad koostada väikese seadistuskatse ning eristad sümptomi võimalikku põhjusest.

Hüperparameetrid on treeningu korraldamise valikud: õpisamm, ajakava, partii, epohhid ja adapteri seaded. Neid ei õpita samal viisil nagu mudeli kaale. Head väärtused sõltuvad mudelist, andmetest, optimeerijast ja eesmärgist.

Õpisamm ja kaokõver

Õpisamm ehk learning rate määrab uuenduse skaalat. 2e-5 tähendab 0,00002. Liiga suur samm võib muuta treeningu ebastabiilseks; liiga väike võib aeglustada õppimist. Hüplev kadu ei tõesta siiski liigset sammu: põhjuseks võib olla erineva raskusega partii, andmeviga, arvutustäpsus või logimise viis.

Ka paigal püsiv kadu ei tõesta väikest sammu. Kontrolli, kas soovitud parameetrid on treenitavad, kas kaomask jätab vastuseid alles ja kas gradiendid üldse tekivad. Kõigepealt kontrolli torustikku, seejärel seadistust.

Algmaterjalis raporteeritud seade	Kuidas seda raamatus tõlgendada
SFT õpisamm 2e-5	Ühe projekti kasutatud väärtus, mitte turvalisuse garantii.
CPT õpisamm 8e-6	Kohalik lähtepunkt; suurem väärtus ei ole üldiselt keelatud.
DPO õpisamm 3e-6 ja beeta 0,3	Ühe katse paar; mõju tuleb uurida koos andmete ja eesmärgiga.
Soojendus 2% ja koosinusajakava	Levinud tüüpi valik, mille sobivust peab kontrollima.

Ajakava ja soojendus

Õpisammu võib hoida püsivana või ajas muuta. Soojendus alustab väiksemast sammust ja suurendab seda. See võib vähendada alguse ebastabiilsust; mõju seostub optimeerija olekute, aktiveeringute ja gradientide dünaamikaga. Seda ei saa seletada ainult ühe universaalse põhjusega.

Koosinus- ja lineaarne langus vähendavad sammu erineva kujuga. Nende mõju ei pruugi olla sama. Salvestatud mudelit edasi õpetades otsusta, kas jätkad ka optimeerija ja ajakava olekut või alustad need uuesti. Need on eri katsed isegi samade kaalude korral.

Partii, kogumine ja epohh

Mikropartii on korraga töödeldavate näidete hulk. Gradiendi kogumine liidab mitme mikropartii õppimissignaali enne optimeerija sammu. Lihtsas andmeparalleelses seadistuses on efektiivne partii mikropartii × kogumissammud × andmeparalleelsed koopiad.

Lahendatud arvutus
Ühel kaardil mikropartii 1 ja kogumine 16 annavad 16 näidet uuenduse kohta. 1024 näidet, üks epohh ja täis partiid annavad 64 uuendust. Kahel andmeparalleelsel kaardil samade seadistustega oleks efektiivne partii 32 ning uuendusi 32.

Gradiendi kogumine võib vastata suuremale partiile, kui kaonormaliseerimine ja muud tingimused klapivad. See ei taga bititäpselt sama tulemust ega sama kiirust. Eri pikkusega vastuste puhul loeb, kas kadu keskmistatakse näite või tokeni järgi.

Epohh tähendab üht andmestiku läbimist. Uuenduste arv sõltub viimase osalise partii käsitlemisest, pakkimisest, jaotusest kaartide vahel ja treeningu peatamisest. Lihtne jagatis on planeerimisvalem, mida tuleb võrrelda treeneri tegeliku loenduriga.

DPO beeta ja optimeerija

DPO beeta ei ole kõva lubatud triivi piir. Algse eesmärgi tõlgenduses seostub suurem beeta tugevama viitemudeli lähedal hoidmisega, kuid lõpliku jooksu käitumine sõltub ka gradientidest, andmetest ja teostusest. Õpisamm ei pea olema alati kümneid kordi SFT omast väiksem. [9, 16]

AdamW hoiab gradientide esimese ja teise momendi hinnanguid ning kasutab eraldatud kaalulangust. 8-bitised ja lehitsetavad variandid muudavad mälukasutust. Lehitsemine ei päästa igast mälupuudusest: aktiveeringud või teised puhverdatud tensorid võivad ikkagi liiga suureks osutuda.

Väike katse enne suurt
Katseta esmalt lühikest jooksu. Muuda üht põhjendatud tegurit või väikest ette määratud seadistusruudustikku. Salvesta lähtepunkt ja eelarve. Ära vali tulemust ainult treeningkao järgi.

H7. Sammude arv
Sul on 960 treeningnäidet, mikropartii 2, kogumine 8 ja üks kaart. Mitu uuendust teeb üks epohh täis partiide korral? Mis muutub, kui mikropartii vähendad ühele ja kogumise tõstad 16-ni?

8. Mõõtmine: kas mudel õppis või harjus kontrolliga?

Selle peatüki järel: eristad treeningut, arenduskomplekti ja lõpptesti ning oskad valida mõõdiku oma väite jaoks.

Õpetaja saab kontrollida harjutusvihikut, kuid eksam peab lisama uut infot. Mudeli arenduses on sama küsimus: kas tulemus kandub üle ülesannetele, mille järgi me otsuseid ei teinud?

Andmehulk	Milleks kasutatakse?	Mida tuleb vältida?
Treening	Kaalude muutmiseks.	Vigased sihid ja kontrollandmete juhuslik lisamine.
Arendus ehk dev/validation	Andmevaliku, seadete ja kontrollpunkti valimiseks.	Selle nimetamine sõltumatuks lõpptulemuseks.
Lõpptest	Ette määratud lahenduse hindamiseks.	Tulemuse põhjal kohandamist ja sama testi jätkuvat sõltumatuks nimetamist.

Fikseeritud faili räsi tõendab, et faili sisu ei muutunud. See ei tõenda sõltumatust. Ka siis, kui ükski testiküsimus ei jõua otse treeningusse, võib testi veakaart juhtida andmevalikut ja seeläbi mudeli arendust.

Testi ei pea müstiliselt „ainult ühe korra” käivitama. Ette määratud kordusmõõtmised mitme seemne või fikseeritud mudeliga võivad kuuluda protokolli. Oht tekib siis, kui saadud infot kasutatakse lahenduse või raporteeritava mõõdu kohandamiseks. Uue arendustsükli jaoks võib vaja minna uut puutumata testi.

Kolm mõõdet, kolm küsimust

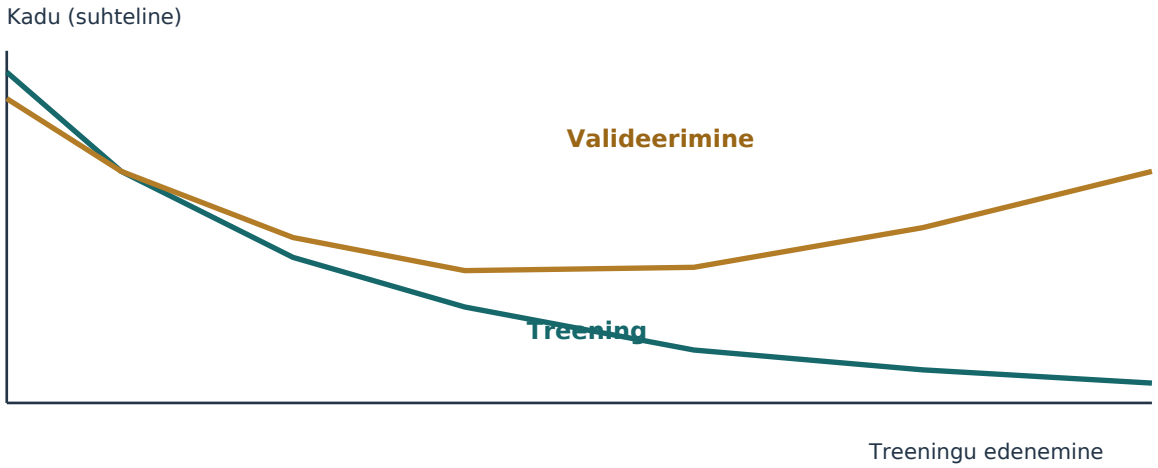
Ülesandeskoor näitab edu valitud ülesannetes. Perpleksus kirjeldab mudeli ennustuslikku sobivust valitud tekstijaotusele. Inimhinnang saab otseselt hinnata kasulikkust, loomulikkust ja stiili. Automaatne või mudelipõhine hindamine võib viimaseid lähendada, kuid vajab inimestega kalibreerimist.

Perpleksus on tavaliselt tokenite keskmise negatiivse logaritmilise tõenäosuse eksponent. Võrdluses peavad sobima tokeniseerija, tekst, konteksti kasutus, nihutamine ja maskimine. Lihtsalt sama faili nimi ei taga sama arvutust. Erinevate tokeniseerijate tavalisi tokenipõhiseid perpleksusi ei tohiks otse paremusjärjestuseks kasutada. [17]

Lahendatud arvutus
22,2-lt 15,4-le langus on (22,2 - 15,4) / 22,2 × 100 = 30,6%. Ümardatult 31%. See ei tähenda 31% vähem grammatikavigu ega 31% suuremat intelligentsust.

Ülesobitumine ja meeldejätmine pole sünonüümid

Ülesobitumine tähendab, et õpitud kohandumine ei üldistu piisavalt soovitud uutele andmetele. Treeningkao langus koos valideerimiskao kasvuga on levinud hoiatusmuster. See ei ole ainus tõend ega eksimatu diagnoos: muutuda võivad jaotus, hindamine või müra. Mudel võib mõnd teksti meelde jätta ka siis, kui mõlemad kaod langevad.



Skeem, mitte projektimõõtmine. Valideerimiskõvera tõus annab põhjuse kontrollida üldistumist; kõver üksi ei selgita põhjust.

Kontrollpunkt salvestab mudeli või adapteri seisu. Varajane peatamine lõpetab jooksu ette määratud paranemisreegli alusel. Vali parim kontrollpunkt eesmärgiga sobiva arendusmõõdu järgi; madalaim keelemudelikadu ei pruugi anda parimat abilise vastust.

Nimetaja ja ebakindlus

Kui skooris kasutatakse osalisi punkte, kirjuta „keskmine punktisumma”, mitte „õigete vastuste protsent”. Algprojekti 151 automaatselt hinnatud ülesandest saadud ligikaudu 129,98 punkti annavad 86,08%. See ei tähenda 130 täiesti õiget vastust.

Väikese kategooria 7 ülesande juures muudab üks binaarne vastus tulemust 14,3 protsendipunkti. See on informatiivne vaatlus, kuid ebakindel üldistus. Raporteeri lugeja, nimetaja ja sobiv usaldusvahemik. Osaliste punktide ning seotud ülesannete puhul ei sobi lihtne binoomvalem automaatselt.

Mudelite võrdluses saab kasutada ülesandepaaridel põhinevat bootstrap'i, grupeerides vajadusel sama lemma või allika ülesanded. Treeningu eri seemned mõõdavad teistsugust varieeruvust kui ülesannete valim. Kolm seemet on praktiline algus mõnes katses, mitte universaalne piisavuse piir.

H8. Lukus, kuid mitte sõltumatu
Testifaili räsi ei muutu. Pärast iga jooksu valid selle tulemuste järgi järgmise treeninguteema. Kas testi võib nimetada sõltumatuks? Mida saad selle skoorist siiski järeldada?

9. Eesti keel: kontrolli ka oma kontrollijat

Selle peatüki järel: oskad eristada keelelist viga, juhise järgimise viga ja hindamisvahendi viga.

Eesti keele treenimisel on hea alustada nähtusest, millel on kontrollitav siht. Sõnavormid annavad selleks palju võimalusi. Eesti käändsõnadel eristatakse tavapäraselt 14 käänet ja kahte arvu, kuid kõigil sõnadel ei ole igas vormis tavalist kasutust. Morfoloogia hõlmab ka muud kui käänamist, näiteks tegusõnavorme. [18]

Lemma on sõna algvorm, näiteks „raamat”. Vorm on konkreetses kasutuses „raamatusse”. Ühildumine seob fraasi liikmete vorme; rektsioon tähendab, et üks sõna tingib teise vormi või seose. Rektsiooni ei määra ainult tegusõnad. Rööpvormid on sama grammatilise ülesandega lubatud variandid.

Miks üksiksõna ja fraas erinevad?

„Habras” ja „tasakaal” eraldi käänamine ei kontrolli kõike seda, mida kontrollib „habras tasakaal” koos. Fraas lisab ühildumise, sõnajärje ja mitme vormi kooskõla. Olemasolev mudel võib neid seoseid juba teada; üksiksõnaõpe ei välista fraasidesse üldistamist, kuid ei taga seda.

Projektivaatlus
Algmaterjalis kirjeldatakse, et üksiksõnadrillide järel jäi fraaside sooritus nõrgaks ning fraasinäidete lisamise järel paranes. See toetab ülesandekuju uurimist. See ei tõesta üldreeglit, et üksiksõnadega ei saa kunagi fraasioskust mõjutada.

Vabamorf on abivahend, mitte eksimatu tõde

Vabamorf ja EstNLTK võimaldavad sõnavormide analüüsi ning sünteesi. Süntees annab lemma ja vormitunnuse põhjal võimalikke vorme; analüüs otsib vormile võimalikke kirjeldusi. Ringkontroll võib leida osa vigadest, kuid mitmesus, puuduv sõnavara ja kontekst jäävad eraldi küsimusteks. [19]

Kui genereerid ja hindad sama tööriistaga, mõõdad osalt kooskõla selle tööriistaga. Lisa sõltumatu keeleeksperdi valim, eri allikatest näited ja veakaebuste ülevaatus. Salvestada tuleb tööriista versioon ning aktsepteeritud vormide loend.

Vastus	Võimalik hinnang	Järgmine kontroll
Vale kääne või arv	Keeleline viga.	Vaata, kas ülesanne oli üheselt sõnastatud.
Õige vorm koos lisaseletusega	Vorminguviga, kui nõuti ainult vormi.	Ära märgi automaatselt keeleoskuse veaks.
Lubatud rööpvorm	Vastus võib olla õige.	Täienda aktsepteeritud vastuste hulka.
Mitmeselt tõlgendatav sõna	Ülesande probleem.	Lisa lemma, tähendus või lausekontekst.

Vastus	Võimalik hinnang	Järgmine kontroll
Hindaja muutus, skoor tõusis	Mõõtmisreegli muutus.	Hinda kõik võrdlusversioonid uue reegliga uuesti.

Veakaart on järgmise katse lähtekoht

Jaota vead tüveks, astmevahelduseks, arvuks, käändeks, ühildumiseks, rektsiooniks, õigekirjaks, ülesande mõistmiseks ja vorminguks. Hoia eraldi hindaja enda vead. Siis ei pea iga nõrka skoori ravima uue üldkorpusega.

Mudeli veapõhine andmevalik on kasulik katseidee: lase mudelil lahendada arendusülesandeid, kontrolli vastuseid ning koosta parandatud näiteid. See ei ole iseseisev tarkuse tekkimise ring. Õiged sihid, tööriistad ja kvaliteedikontroll tulevad väljastpoolt.

Paranda mõõt enne mudelit
Kui aktsepteeritud vastuste loend muutub, salvesta hindaja uus versioon. Vana skoor ja uus skoor ei ole sama mõõdu tulemus. Võrdlus muutub ausaks alles siis, kui sama hindaja hindab mõlemat mudelit.

H9. Näiline areng
Muutsid hindaja rööpvormide loendit ja skoor tõusis 8 punkti, kuigi mudelifail on sama. Mida parandasid? Kuidas kontrollid, kas hilisem treening andis lisakasu?

10. Dialoog, käitumine ja inimese usaldus

Selle peatüki järel: koostad vestlusnäite, mis õpetab konteksti kasutamist, ning hindad viisakust koos tõepärasusega.

Keeleabiline peab lisaks vormi leidmisele aru saama, millest vestluses juba räägiti. Ühevooruline ülesanne ei kontrolli viitamist varasemale parandusele, teema muutmist ega kasutaja täpsustust. Seepärast tuleb vestluse sihtvõimeid eraldi kirjeldada.

Lahendatud vestlusnäide
Kasutaja: „Pane «habras tasakaal» mitmuse osastavasse.” Abiline: „hapraid tasakaale”. Kasutaja: „Miks omadussõna muutus?” Abiline selgitab, et fraasi omadussõna ühildub siin nimisõnaga arvus ja käändes. Järgmine voor „Anna nüüd sama fraas ainsuse omastavas” kontrollib varasema fraasi säilitamist.

Näites õpitakse eri asju: vormimoodustust, põhjendamist ja viite „sama fraas” lahendamist. Treeningandmetes peavad need olema eristatavad rollidega voorud. Kaomask peab vastama sinu valitud eesmärgile; maski olemasolu tuleb kontrollida tegelikust sisendist, mitte ainult konfiguratsioonireast.

Kajamine ei anna üht diagnoosi

Kui mudel kordab küsimust, võivad põhjuseks olla mall, lõputoken, genereerimisrežiim, puuduv ajalugu, treeningnäited või pakendamine. Mitmevoorulise kohaliku SFT puudumine on üks hüpotees, mitte tõend, et lähtemudel pole kunagi dialoogi näinud. Eriti tähtis on see instruct-mudeli kohandamisel.

Valmista eri pikkusega vestlused: kaks vooru, viis vooru, teema vahetamine ja varasema vea parandamine. Mõni ülesanne peab kontrollima, et mudel küsib täpsustust, mitte ei täida lünka oletusega. Samal ajal ei tohiks iga lihtne küsimus tekitada tarbetut ülekuulamist.

Poliitiline või kultuuriline ühtlus pole headus

Kasvatamise võrdlus kutsub vastutama selle eest, millist käitumist kinnistame. Hea eesti keele mudel peab oskama eri registrites rääkida ja eri inimestest lugu pidada. Ühe maailmavaate sagedus andmetes ei tee seda ainsaks mõistlikuks vaateks.

Hindamiskriteerium	Hea vastuse tunnus
Tõepärasus	Eristab teadaolevat, oletust ja puuduvat infot.
Lugupidamine	Parandab vea inimest halvustamata.
Iseseisev hinnang	Ei nõustu valeväitega ainult kasutajale meeldimiseks.
Kontekstitaju	Kasutab vestluse olulisi varasemaid andmeid.
Proportsioon	Vastab nii pikalt, kui ülesanne vajab.

Hindamiskriteerium	Hea vastuse tunnus
Parandatavus	Võtab põhjendatud paranduse arvesse ja kontrollib järeldust.

Inimhindamisel peida mudeli nimi ja juhuslikusta vastuste järjekord. Kasuta selget juhendit ning harjutusvooru. Hindajate erimeelsus võib osutada ebaselgele kriteeriumile, mitmele heale vastusele või keerulisele keeleküsimusele. See on uuritav tulemus, mitte automaatselt häiriv müra.

H10. Viisakas, kuid ekslik
Mudel nõustub sõbralikult väitega, et eesti keeles on seitse käänet. Millised kaks hindamiskriteeriumi annaksid talle eri hinnangu? Millise treeningnäite lisaksid?

11. Riistvara, mälu, aeg ja energia

Selle peatüki järel: koostad eelarve selgete ühikute ja eeldustega. Eristad kaalude suurust kogu mälukulust ning võimsuspiiri mõõdetud energiast.

Videomälu ehk VRAM on üks peamisi piiranguid. Lisaks loevad arhitektuuri tugi, kernelid, arvutustäpsus, draiver, RAM, konteksti pikkus ja teostus. Mudel võib kettale mahtuda, kuid laadimisel või treeningu esimesel sammul ikkagi mälust välja minna.

Millise arvutiga saab alustada?

Õppimiseks ei ole vaja kohe 27 miljardi parameetriga mudelit. Selle raamatu praktiline töö algab ligikaudu 0,5B mudeli LoRA-kohandamisest. B tähendab miljardit parameetrit. Väike mudel teeb kogu töövoo nähtavaks: andmed, gradient, adapter, uus vastus ja hindamine. See ei ole soovitus valida sama mudel lõplikuks eesti keele abiliseks.

Järgmised tabelid on konservatiivsed planeerimisorientiirid, mitte siin riistvaral mõõdetud mahutavustestid ega ostugarantii. Eeldame ühte kasutajat, lühikesi 512-1024 tokeniga jadasid, mikropartiid 1 ja adaptertreeningut. Suuremad mudelivahemikud eeldavad tavaliselt 4-bitist baasi ja mälu säästvaid võtteid. Arhitektuur, sõnastiku suurus ning sihtmoodulid võivad piiri oluliselt muuta.

NVIDIA VRAM	Mõistlik esimene siht	Milleks varu jääb?
8 GB	0,5-1,5B LoRA; selle raamatu väike katse.	Töövoo õppimine; 7B pole siin algaja vaikimisi siht.
12-16 GB	1,5-3B; 7-8B QLoRA pärast mahukatset.	Rohkem andmeid ei pea korraga GPU-s olema.
24 GB	7-8B QLoRA; 14B tingimuste kontrolliga.	Pikem kontekst või rohkem treenitavaid mooduleid vajab eraldi katset.
32 GB	7-14B on mugavam lähtekoht; 27-32B on nõudlikum.	Autori 27B juhtum näitab võimalikkust kindlas seadistuses.
48 GB ja rohkem	Suuremad adapterkatsed või pikemad jadad.	27B täistreening ei muutu seetõttu automaatselt jõukohaseks.

8 GB on siin soovituslik praktiline algus, mitte teoreetiline miinimum. Väga väikese mudeli või CPU abil saab põhimõtteid uurida ka väiksema mäluga, kuid see pole raamatu põhitee. 8 GB RAM-i ja 8 GB VRAM-i ei saa tavalisel NVIDIA tööjaamal lihtsalt üheks 16 GB treeningmäluks liita.

Ubuntu tööjaama ülejäänud osad

Osa	Esimene väike labor	Suurenev kohalik projekt
Operatsioonisüsteem	Toetatud Ubuntu LTS, Python 3.11 või 3.12 sobivas keskkonnas.	Mudeli ja teekidega ühilduv draiver; säilitatud töötav keskkond.
Põhimälu ehk RAM	16 GB võib väikseks katseks piisata; 32 GB annab rohkem varu.	64 GB või rohkem suurte failide, teisenduste ja andmetöötluse jaoks.
Kettaruum	Vähemalt 20-30 GB vaba ruumi väikese labori jaoks.	27B projekti jaoks planeeri sadade GB-de suurusjärk; 300 GB on algne eelarve, mitte ülempiir.
Ketas ja protsessor	SSD ning tänapäevane mitmetuumaline CPU.	NVMe aitab suurte failide ja vahepunktidega; rohkem tuumi aitab andmetöötlust.
Toide ja jahutus	GPU ning kogu masina tootjanõuetele vastav toide ja jahutus.	Pika koormuse stabiilsus, pistikud, temperatuur ja tegelik tarve.

Kettale võivad korraga jääda allalaaditud kaalud, vahemälukoopia, adapterid, optimeerija vahepunktid, ühendatud mudel ja mitu kvanditud varianti. 27B 16-bitine kaalukomplekt on juba ligikaudu 54 GB. Kui lood sellest kolm täismõõdus koopiat, kulub ligikaudu 162 GB enne andmeid ja muid faile. Tee mahueelarve failide kaupa.

NVIDIA rada kasutab CUDA-t. AMD või Inteli kaart pole põhimõtteliselt välistatud, kuid vajab teistsugust ühilduvuskontrolli; selle raamatu käske ei saa sellele muutmata üle kanda. Uue kaardipõlvkonna puhul kontrolli lisaks draiverile ka PyTorchi väljalaske ja kasutatud kernelite tuge. [29, 30]

Apple Silicon: ühendatud mälu

Apple Siliconi Macis kasutavad CPU ja GPU ühist mäluruumi. MLX on selle arhitektuuri jaoks loodud arvutusraamistik; MLX LM lisab keelemudelite kasutamise ja kohandamise vahendid. Süsteemi kogumälu pole siiski tervenisti ühe treeningu käsutuses: mälu kasutavad macOS ja muud rakendused ning GPU kasutusel võivad olla süsteemsed piirangud. [31, 32]

Maci ühendatud mälu	Esimene mõistlik siht	Tähelepanu
16 GB	0,5-3B adapterkatse.	Selle raamatu 0,5B labor; sulge muud mahukad rakendused.
24-32 GB	3B; 7-8B kvanditud adapterkatse mahukontrolliga.	Mälusurve ja swap võivad töö väga aeglaseks teha.
48-64 GB	7-14B adapterkatsed.	27B võib sobiva seadistusega mahtuda, kuid kiirust tuleb mõõta.

Maci ühendatud mälu	Esimene mõistlik siht	Tähelepanu
96-128 GB ja rohkem	27-32B QLoRA on realistlikum uurimissuund.	Suurem mälumaht ei tähenda automaatselt kiiremat GPU-d.

Vali Apple Silicon ehk M-seeria protsessoriga Mac; siin kirjeldatud MLX-rada ei ole Intel Maci juhend. Sama mälumahuga M-seeria kiibid võivad erineda GPU tuumade, mäluribalaiuse ja jahutuse poolest. Sülearvuti hoia pika treeningu ajal vooluvõrgus ning väldi unerežiimi. Ära lahenda mälupuudust esmalt süsteemi kaitsepiiride tõstmisega: alusta väiksema mudeli või lühema jadaga.

Enne ostmist
Kirjuta välja täpne mudel, meetod, jada pikkus ja treenitavad moodulid. Otsi samalaadse seadistuse mõõtmist või tee proov ligipääsetaval masinal. Inferentsi mahutavustabel ei ole treeningu mahutavustabel. Kui olemasolev arvuti võimaldab väikest laborit, tee see enne suuremat ostu.

Kas kaks arvutit annavad ühe suure mälu?

Kaks 16 GB kaarti ei muutu ilma hajutatud treeningu teostuseta üheks 32 GB kaardiks. Andmeparalleelses treeningus on sageli mõlemal kaardil mudeli koopia. Mudeli jagamine seadmete vahel vajab eraldi meetodit ja side võib saada pudelikaelaks. Ka Tailscale ühendus annab ligipääsu masinatele, mitte automaatse ühise GPU-mälu.

Alustaja jaoks on kasulik esimene mitme masina kasutus korraldada sõltumatute katsetena: ühel treening, teisel hindamine või andmete kontroll. Nii saab koostööst kasu ilma, et iga treeningusamm peaks üle võrgu suuri tensoreid vahetama.

Alusta ainult kaaludest

27 miljardi parameetri säilitus	Lihtne arvutus	Mida see ei sisalda
16 bitti	27 × 10^9 × 2 = 54 GB	Aktiveeringud, vahemälud ja treeninguolekud.
8 bitti	27 × 10^9 × 1 = 27 GB	Kvantskaalad, erandid ja muud tensorid.
4 bitti	27 × 10^9 × 0,5 = 13,5 GB	Tegelik pakend võib olla märgatavalt suurem.

GB on miljard baiti; GiB on 2^30 baiti. 27,78 miljardi parameetri 16-bitised kaalud on 55,56 GB ehk umbes 51,7 GiB. See võib selgitada, miks kaks programmi näitavad sama faili kohta „56” ja „52”. Kirjuta ühik alati välja.

Täistreeningu mälu sõltub olekutest

Tavapärane segatäpsusega AdamW võib lisada FP32 põhikoopia, gradiendid ning kaks FP32 momenditensorit. Levinud arvestus on ligikaudu 16-18 baiti parameetri kohta enne aktiveeringuid, sõltuvalt gradientide ja kaalude teostusest. 27 miljardi korral tähendab see umbes 432-486 GB, mitte algmaterjali tabelis toodud universaalset 220 GB. Teistsugune optimeerija, täpsus, jaotamine või väljakolimine muudab arvu. [20]

QLoRA vähendab külmutatud baasi ja treenitavate olekute koormust, kuid aktiveeringud jäävad. Gradiendi kontrollpunktimine säästab aktiveeringumälu neid osaliselt uuesti arvutades. See erineb mudelifaili kontrollpunkti salvestamisest. Pikem jada võib suurendada nii mälu kui ka aega; kasv sõltub arhitektuurist ja kernelitest.

Tee oma mahukatse
Laadi täpne mudel, käivita mõni tegelik treeningusamm ja mõõda mälutipp. Katseta ka pikimaid näiteid. Ära osta riistvara üksnes üldise „mudeli suurus → VRAM” tabeli järgi.

Aeg: defineeri, mida token sekundis tähendab

Jooksu kestust võib hinnata sisendtokenite koguse ja samas seadistuses mõõdetud läbilaskevõimega. SFT-s erista sisendtokenite ja kaotokenite kiirust. Lühikese mikropartii benchmark ja terve jooksu seinakellaaeg pole sama mõõt.

Lahendatud arvutus
110,2 miljonit tokenit / 870 tokenit sekundis / 3600 = ligikaudu 35,2 tundi. Kontrollpunktide salvestamine, valideerimine ja käivitamine võivad lisada aega, kui need kiiruse mõõtmises juba ei sisaldu.

Energia vajab tegelikku võimsust

Energia on võimsuse ajaintegraal. Võimsuspiir 450 W ei tähenda, et kaart tarbis igal hetkel 450 W. Arvutus 450 W × 35,3 h = 15,9 kWh on nimipiiril põhinev kaardi hinnang, mitte mõõdetud kogu arvuti elektrikulu.

Algmaterjali 575 W ja 8,77 s korrutis annab 5043 J ning 450 W ja 10,59 s 4766 J. Need on piiri ja aja korrutised. Energiasäästu tõestamiseks on vaja tegeliku võimsuse mõõtmist võrreldava töö jooksul; kogu süsteemi jaoks on kasulik seinast mõõtmine.

H11. Aja vastuolu
Raport ütleb 3364 uuendust ja 35,3 tundi. Teine tabel ütleb 10,59 sekundit sammu kohta. Arvuta kogu jooksu keskmine sammuaeg. Millist lisainfot vajad, enne kui kuulutad ühe arvu valeks?

12. Treenitud mudelist töötava abiliseni

Selle peatüki järel: kontrollid liitmist, kvantimist ja serveerimist eraldi etappidena.

Treeningu tulemus võib olla hea, kuid kasutaja näha halba vastust. Seepärast tuleb vaadata tervet ahelat: lähtemudel koos adapteriga, liidetud kaalud, kvanditud fail ja lõpuks serverisse pakendatud mudel. Igas sammus on oma veavõimalused.

Mida kvantimine muudab?

Kvantimine vähendab arvude esitamise täpsust ja võib parandada mahutavust või kiirust. Q4_K_M, Q6_K ja Q8_0 on konkreetsed vormingud, mitte universaalsed kvaliteedigarantiid. Q6 ja Q8 võivad säilitada kvaliteeti paremini, kuid „peaaegu kadudeta” vajab sinu ülesannetel kontrolli.

Võrdle sama lähtefaili, malli, viipu, serverit ja genereerimisseadeid. Kontrolli nii täpseid keeleülesandeid kui ka vestlust. Kui ortograafiavead tekivad Q4 paketis, on kvantimine üks kahtlus, kuid kontrollida tuleb ka teisendust ja malli.

Etapp	Kontroll	Millise vea see eristab?
Baas + adapter	Fikseeritud väike kontrollkogum.	Kas treeninguväljund toimib?
Liidetud mudel	Samad sisendid ja sobiv täpsus.	Kas adapteri liitmine muutis tulemust?
Kvanditud fail	Samad ülesanded eri vormingutes.	Kas viga seostub kvantimise või teisendusega?
Server / rakendus	Üksikvastus, dialoog, lõpetamine.	Kas mall, ajalugu ja parameetrid on õiged?

Mall ei kao alati

GGUF võib sisaldada tokeniseerija ja vestlusmalli metaandmeid. See, mida importija kasutab, sõltub mudelist, failist ja tarkvaraversioonist. Väide „GGUF-ist ei kandu mall kaasa” ei ole üldreegel. Ollama ametlik Modelfile'i dokumentatsioon kirjeldab TEMPLATE-juhist ja mudelipõhist malli. RENDERER/PARSER ei ole igale mudelile kohustuslik universaalne paranduspaar. [21]

Ollama sobib kohaliku kasutamise lihtsustamiseks; llama.cpp annab kontrolli toetatud kohalike mudelite üle. vLLM ja teised serverid võivad pakkuda suure koormuse juures tõhusat teenindamist. „Alati kiirem” või „alati rohkem mälu” ei ole ilma töökoormuse ja seadistuseta sisukas võrdlus.

Genereerimisseaded ei ole treening

Temperatuur muudab tokenite valiku jaotust. Top-k piirab kandidaatide hulka; top-p valib suure tõenäosusega kandidaatide kogumi kumulatiivse piiri järgi. Kordustrahv muudab varem esinenud tokenite käsitlemist. Ükski neist ei paranda püsivalt kaale.

Mõnes mootoris tähendab temperatuur 0 kokkuleppeliselt ahnet valikut. See ei taga bititäpset korratavust kõigis keskkondades. Suurem temperatuur ei tähenda alati rohkem sisulisi vigu ja väiksem ei taga tõde. Mõtlemisrežiimi puhul järgi täpse mudeli soovitusi ning mõõda valitud režiimi eraldi.

Väike tehniline kontroll ei asenda hindamist
Viis kuni kümme fikseeritud viipa võivad avastada katkise pakendi. Need ei tõesta üldist keeleoskust. Kasuta neid iga teisenduse järel, seejärel käivita eesmärgiga sobiv hindamine.

H12. Pärast eksporti läks halvaks
Adapteriga mudel vastab hästi, Ollama pakett kajab küsimust. Miks ei ole uus treening veel põhjendatud esimene samm? Millised kaks võrdlust teeksid enne?

13. Järgmised võtted: mida tasub katsetada?

Selle peatüki järel: oskad uut meetodit käsitleda kontrollitava hüpoteesina ja hinnata selle ülekantavust.

Uue meetodi nimi ei ole iseenesest põhjus selle kasutamiseks. Küsi, millise pudelikaela ta eemaldab, millist uut riski lisab ja millise võrdlusega kasu nähtavaks saab. Järgmised võimalused ei ole selle raamatu juhtumis kõik läbi katsetatud.

Võte	Mis muutub?	Mida kontrollida?
DoRA	Uuenduse suuna ja suuruse käsitlus.	Kvaliteet, mälu, kiirus ning mudelitugi.
rsLoRA	Adapteri rank'ist sõltuv skaala.	Õpisammu ja rank'i ühine sobivus.
Prompt/prefix tuning	Õpitavad lisavektorid, baas tavaliselt külmutatud.	Kas kitsas ülesanne üldistub?
Destilleerimine	Õpilane õpib õpetaja väljunditest või jaotustest.	Õpetaja vead, õigused, sõltumatu kvaliteet.
Mudelite liitmine	Ühilduvate kaalude matemaatiline kombineerimine.	Päritolu, arhitektuur ja eri oskuste säilimine.

Maatriksite liitmine ei kanna oskust igasse mudelisse

SLERP interpoleerib kaale geomeetrilise reegliga. TIES käsitleb uuenduste tugevust ja märgikonflikte. DARE eemaldab juhuslikult osa uuenduse elemente ning skaleerib allesjäänuid; eemaldamine ei ole ainult väikseimate elementide valik. Tööriistad nagu mergekit võimaldavad eri teostusi. [22]

Eduka liitmise eelduseks on sobivad kaalude kujud ja tähendused; ühine lähtebaas aitab. Meetod ei vii ühe mudeli eesti keele adapterit automaatselt teise arhitektuuri või järgmisse põlvkonda. Samuti ei ole vältimatu, et midagi läheb katki. Mõlemad suunad vajavad mõõtmist.

Tokeniseerija laiendamine

Uute tokenite lisamine võib lühendada jadasid. Vaja on kontrollida teksti tükeldusreegleid, sisend- ja väljundtabeleid, seotud kaale, erimärke ning serverituge. Lihtne nimekirja pikendamine ei taga, et uued tükid tegelikult kasutusse lähevad.

Uute esituste lähtestamine vanade tükkide keskmisest on üks võimalik võte. See ei anna täpselt sama käitumist: üks uus token muudab jada pikkust, konteksti ja väljundijaotust. Mõõda

tokenisäästu, kvaliteeti, unustamist ning kogu kasutusaega. Lühem jada võib aidata, kuid suurem sõnastik võib lisada oma kulu.

MoE ja avatud mudelid

MoE ehk mixture of experts suunab tokeni läbi valitud ekspertplokkide. Aktiivsete parameetrite arv aitab kirjeldada arvutust, kuid ei taga sama kiirust sama suure tiheda mudeliga. Mällu võivad vaja minna kõik eksperdid; marsruutimine ja andmete liigutamine lisavad kulu.

Qwen, Llama, Mistral, Gemma ning Euroopa keeli toetavad mudelipered pakuvad eri lähtekohti. Allalaaditavad kaalud ei tähenda automaatselt avatud lähtekoodi ega ühesugust litsentsi. Kontrolli konkreetset mudelikaarti, treeningustaadiumi, mõõdetud keeleoskust ja oma tööriistade tuge. Perekonna nimi üksi ei anna õiget vastust.

Katseidee: ülekande komplekt
Hoia ühe adapteri asemel alles ka puhas andmevalik, ülesannete generaator, hindamisjuhend ja manifest. Need on sageli järgmisse mudelipõlvkonda lihtsamini ülekantavad kui kaalud ise.

H13. Uus mudelipõlvkond
Sul on hea eestikeelne adapter ja ilmub teistsuguse arhitektuuriga mudel. Millised kolm projekti tulemit saad tõenäoliselt uuesti kasutada ka siis, kui adapter ei sobi?

14. Ühe projekti lugu: tulemus ja selle piirid

Selle peatüki järel: oskad lugeda katsetulemust koos nimetaja, mõõtetingimuste ja ebakindlusega.

Algmaterjal kirjeldab Qwen3.8-27B kohandamist eesti keele jaoks ühe 32 GB mäluga RTX 5090 abil. Autor raporteerib üheksa päeva jooksul 34 salvestatud treeningut ja ühe väikese proovijooksu. Need arvud kirjeldavad tehtud tööd, mitte 34 sõltumatut kinnitust sama meetodi paremusest. Järgnev põhineb autori õppematerjalil; treeninguid ei ole selle väljaande jaoks korratud. [26]

Mis mudelist me räägime?

Qwen3.8-27B ametlik mudelikaart kirjeldab tihedat multimodaalset mudelit, mille arhitektuur ühendab täistähelepanu ja Gated DeltaNeti plokke. See ei ole lihtsalt algse Transformeri kõigi kihtide kordus. Konfiguratsioonis olev vocab_size ei pruugi võrduda tokeniseerija tegelike kasutatavate kirjete arvuga: kaalutabel võib olla polsterdatud. Konkreetse projekti parameetrite täpne arv tuleb lugeda kasutatud failidest. [5]

Tulemuste tabelit peab saama lahti seletada

Mõõdik algmaterjalis	Enne → pärast	Mida saab öelda?
Automaatne koondhinne	61,7% → 86,1%	Paranemine projekti hindamiskogumil; kogumit kasutati arenduses.
Käänamine, 60 ülesannet	51,7% → 91,7%	Suur paranemine selle ülesanderühma hindamisreeglite järgi.
Grammatika, 50 ülesannet	80,2% → 84,3%	Väiksem muutus; vaja vaadata konkreetseid vastuseid.
Rektsioon, 8 ülesannet	87,5% → 100%	Väike ülesannete arv piirab järelduse ulatust.
Proosavalimi PPL	22,2 → 15,4	Mudeli ennustus sobitus paremini selle tekstivalimiga.

Algne komplekt sisaldab 200 ülesannet. Koondprotsendi arvestusest jäid välja 47 inimhindamist ootavat ülesannet ja kaks koodiülesannet. Nimetaja on seega 151. Osapunktidega summa 129,98 annab 129,98 / 151 × 100 = 86,08%, ümardatult 86,1%. See ei tähenda, et 86,1% kõigist 200 küsimusest sai täielikult õige vastuse.

Algmaterjalis on ka EstQA, MMLU-et ja HumanEvali tulemused, kuid need käivad varasema mudelivariandi kohta. Neid ei tohi tõsta lõpliku mudeli veergu. Varasemal variandil paranes EstQA F1 86,9-lt 93,6-ni ja HumanEval 71,3%-lt 85,4%-ni, samal ajal langes MMLU-et 68,7%-lt 66,0%-ni. Üks paranemine ei tõenda kõigi oskuste säilimist.

378 käänamisvormi eraldi kontrolli kirjeldati ligikaudu kümme protsendipunkti nõrgemana. See on põhjus uurida üldistumist. Ilma vastuste, hindaja ja valimi täpse kirjelduseta ei saa sellest teha täpset uut kvaliteedinumbrit. Proosavalimi puhul kontrolliti lõikude kattuvust, kuid puudus terviklik autorite ja teoste eraldatuse audit.

Projektivaatlus ei ole edetabel
Teiste süsteemidega võrdlemine nõuab täpseid versioone, ühesuguseid ülesandeid, viipasid, režiime ja hindamist. Projekti arendustulemus üksi ei tõenda, et kohalik mudel on üldiselt mõnest teisest süsteemist parem.

Millised katsed andsid õpetliku vea?

Ühes DPO katses kasutati 9543 välist eelistuspaari ning tulemus halvenes kirjeldatud hindamisel umbes kuus protsendipunkti. Hilisem, 244 oma veast koostatud paariga katse andis umbes ühe punkti juurde. Muutusid ka muud tingimused. Sellest saab püstitada hüpoteesi, et lähedasemad veanäited aitasid; ei saa järeldada, et väline DPO andmestik üldiselt ei tööta või et oma mudeli paarid alati võidavad.

Samuti ei tõesta eri etappide rank 16 ja rank 32 katsed ühe rank'i universaalset paremust. Muutusid andmed ja treeninguajalugu; lõplik kirjeldatud 86,1% variant kasutas rank 32. Õige järgmine katse võrdleks rank'e sama lähtekaalu, andmevaliku, eelarve ja hindamisega.

Lahendatud näide: kuhu kadusid tokenid?

CPT segu mahuks on antud 59,5 miljonit sõna. Korrutamine suhtega 2,2 annab 130,9 miljonit tokenit. Suhtega 2,37 saadakse 141,015 miljonit. Mõlemad on hinnangud seni, kuni kogu tekst on täpse tokeniseerijaga loendatud. Mõne valimi suhe ei ole kogu korpuse tegelik loendus.

Treeningu mahutavus
Kui 53 821 tükki kärbitakse maksimaalselt 2048 tokenini, on ülempiir 110 225 408 tokenikohta. Tegelik arv võib olla väiksem lühemate jadade või polsterduse tõttu; loss'is osalevate tokenite arv võib olla veel väiksem. Valem 3364 sammu × 16 näidet × 2048 annab ligikaudse ülempiiri, sest viimane partii ei pruugi olla täis.

Kui tüki pikkus oli 2900 ja piir 2048, kaob sellest tükist 852 tokenit ehk 29,4%. Seda protsenti ei saa kanda kogu korpusele, teadmata pikkusjaotust ja pakkimise käiku. Lahendus on loendada enne ja pärast pakkimist ning kontrollida, kas ülejääk kantakse järgmisse tükki või visatakse ära.

Aeg, kiirus ja ühik
35,3 tundi on 127 080 sekundit. Jagades selle 3364 optimeerimissammuga, saame umbes 37,8 sekundit sammu kohta. Algmaterjali 10,59 sekundit võib olla teistsuguse koormuse mõõtmine või teine sammumääratlus; sama jooksu keskmisena need arvud ei klapi. 141,015 miljoni tokeni töötlemine kiirusel 870 tokenit sekundis kestaks umbes 45,0 tundi, mitte 35,3 tundi.

Avalik projektihoidla ja paranduste fail on kasulikud töö jälgimiseks, kuid sama autori avalik kirjeldus ei ole sõltumatu korduskatse. Usaldusväärsus kasvab, kui koos tulemusega säilivad mudeli räsi, andmemanifest, hindaja versioon ja toorvastused. [27]

Minu kogemuse seitse õppetundi

Järgmised juhtumid toovad algmaterjali praktilise kogemuse eraldi nähtavale. Need on autori kirjeldatud sündmuste toimetatud kokkuvõtted. Iga loo juures eristame tähelepanekut ja sellest põhjendatud järgmist tegevust. [26]

1. Õpetasin sõnavormi, aga ootasin fraasi

Algmaterjali järgi treeniti esmalt üksiksõnade vorme, samal ajal kui hindamine nõudis omadussõna ja nimisõna kooskõla. Fraasinäidete lisamise järel tulemus paranes. Vahe on sisuline: „habras” ja „tasakaal” eraldi ei kontrolli seda, kas mudel moodustab fraasi „hapraid tasakaale”.

Välditav viga on treeningu ja kasutusülesande lahknevus. Võta enne andmete tootmist kümme tüüpilist kasutusjuhtumit ja kontrolli, et nende ülesehitus oleks õppematerjalis esindatud. See ei tähenda samade lõpphindamise lausete treeningusse kopeerimist.

2. Väike sihitud paranduste kogu osutus kasulikuks

Autori kirjeldatud sekkumine	Täheldatud muutus	Piirang
720 metateadmise näidet	Üks kategooria 0,8% → 57,7%.	Kategooria tulemus ei ole kogu mudeli keeleoskus.
130 käsitsi kureeritud rektsiooninäidet	Rektsioon 62,5% → 100%; koondhinne +2,1 punkti.	Teine katseetapp kui lõpptabeli rektsioonivõrdlus.
81 268 üldist näidet	Koondhinne +1,1 punkti; mõnes rühmas langus.	Maht, sisu ja treeninguajalugu polnud kontrollitud võrdluses eraldatud.

Järgmise katse mõte on uurida vealiiki, mitte koguda automaatselt kümme korda rohkem teksti. Koosta näiteks rektsiooniveale selged head ja halvad näited, kontrolli põhjendusi ning mõõda sama oskust uutel verbidel ja lausetes.

3. Andmefaili suurus ei näidanud mudelini jõudnud mahtu

Tähemärkide järgi pakitud tekstid osutusid tokeniseerides liiga pikaks. Algmaterjal ise tõi selle välja vaikse kaona. Selle väljaande arvutused näitavad, miks täpset kaoprotsenti tuleb kinnitada päris jadade loendusega. Enne pikka jooksu salvesta viis lühikest ja viis pikka töödeldud näidet koos tokenite ning loss'i maskiga.

4. Väikseim loss polnud parima abilise tõend

Ebaõnnestunud DPO katses oli raporteeritud kadu väga väike, kuid keeleülesannete tulemus halvenes. Ka CPT puhul puudus algses jooksus otsustamiseks piisav valideerimiskõver. Võta sellest kaasa kontroll: logi lisaks treeningkaole sobiv arendusmõõt ja vaata vastuseid. Ära vali lõppvarianti ainult selle järgi, et ta on viimane salvestatud fail.

5. Andmete järjekord ja kohandamine läksid ühte võrdlusse

Algmaterjal kirjeldab ühe suure SFT-annuse tulemust 76,5% ning kolmeteistkümne ringi järel 85,3%. Ringide vahel muudeti materjali vigade põhjal. Seega ei erista see võrdlus treeningu etapistamise mõju paremini sihitud andmete mõjust. Hea järgmine katse hoiab andmekogu ja eelarve samana ning muudab ainult järjekorda.

6. Katkine pakend nägi välja nagu halb mudel

Autor kirjeldab halvenemist Q4 variandis ja küsimuste kajamist kasutuspakendis. Algmaterjal seostas viimast vestlusmalliga. Üldine kontrollivõte on teha sama viibaga võrdlus iga teisenduse järel: adapter, ühendatud kaalud, kvanditud fail ja server. See aitab leida, kus erinevus tegelikult tekkis.

7. Kindlustunne kasvas kiiremini kui mõõtmise sõltumatus

Sama arenduskogum aitas valida järgmiste ringide teemasid. Isegi muutumatu faili korral jõudis hindamise info arendusotsustesse. Õppetund on väärtuslik: säilita arendusmõõt ausa nimega ning planeeri uus hindamine, kui tahad rääkida üldistumisest.

Minu veast sinu kontrollküsimuseks
Enne „mudel vajab uut treeningut” küsi: kas näide on õige, kas ülesanne sobib, kas tekst jõuab mudelini, kas mõõdik on õige ja kas kasutuspakett kordab treeningu vestlusvormingut? Viis kontrolli võivad säästa terve treeningu.

H14. Ühe numbri lahtikirjutamine
Kirjuta lause, millega tutvustaksid 86,1% tulemust ausalt inimesele, kes pole projekti näinud. Lisa mõõdetud ülesannete arv ja üks piirang.

15. Esimene kontrollitud katse

Selle peatüki järel: oskad koostada väikese katse, mille tulemusest saab teha järgmise otsuse.

Esimese katse eesmärk on õppida oma süsteemi tundma. Ära alusta kõige suurema korpuse ja pikima treeninguga. Väikese katse abil saad kontrollida, kas andmed jõuavad õigesti mudelini, loss arvutatakse õigetes kohtades ja hindaja mõõdab vajalikku oskust.

Tööleht A. Üks küsimus korraga

Väli	Näidis, mida saad asendada
Kasutusülesanne	Keeleabiline parandab vigase käändevormi ja põhjendab parandust.
Täheldatud viga	Vorm on õige, kuid põhjendus sisaldab väljamõeldud reeglit.
Hüpotees	Kontrollitud põhjendustega näited vähendavad seda vealiiki.
Muudetav tegur	Ainult treeningnäidete põhjenduste kvaliteet.
Võrdlus	Sama lähtekaal, sama eelarve ja sama hindaja; salvestatud baastulemus.
Edu mõõt	Vähem vale põhjendusega vastuseid uutel ülesannetel; vormitäpsus säilib.
Peatamise põhjus	Valideerimisel kasvab vigade arv või kaob oluline varasem oskus.

Pane enne katset kirja ka praktiline lävend. Näiteks nõua, et parandust näeks mitmes ülesanderühmas, mitte ainult ühel sõnal. Lävend sõltub kasutusjuhtumist ja vea hinnast; raamatus toodud näide ei ole universaalne standard.

Katse järjekord

1. Määra üks oskus ja koosta eri raskusega hindamisnäited. Eralda treening, arendus ja lõpphindamine enne mudeli valimist.
2. Käivita muutmata lähtekaaluga hindamine. Säilita kõik vastused, mitte ainult protsent. Kontrolli osa hinnanguid käsitsi.
3. Valmista väike, õiguste ja kvaliteedi poolest kontrollitud treeningkogum. Näiteks mõnisada head näidet on torustiku proovimiseks sobiv lähtekoht, mitte kvaliteedigarantii.
4. Vaata tokeniseeritud näiteid inimesele loetaval kujul. Kontrolli rollimärke, kärpimist, maske ja assistendi vastuse olemasolu.
5. Tee mõnekümne sammuga tehniline proov. Kontrolli, et treenitavad parameetrid saavad gradiente, kadu on lõplik ja salvestatud adapter laaditakse tagasi.
6. Käivita kavandatud katse. Salvesta seaded, aja- ja mälumõõtmised, arendushindamine ning vähemalt üks taastatav vahepunkt.

7. Võrdle toorvastuseid vealiikide kaupa. Kui tulemus ei parane, uuri kõigepealt hüpoteesi, andmeid ja mõõtmist; uus juhuslik parameetrikomplekt ei selgita põhjust.
8. Kui meetod on valitud, tee eelnevalt kavandatud lõpphindamine. Seejärel kontrolli kasutuspaketti ja dokumenteeri piirangud.

Tööleht B. Katsepass

Katsepass ehk manifest seob tulemuse selle põhjustanud failide ja valikutega. „Qwen, vaikeseaded, minu andmed" ei võimalda tööd hiljem taastada. Kasuta järgmisi välju oma märkmetes või masinloetavas failis.

Rühm	Salvestatavad väljad
Identiteet	Katse ID, kuupäev, eesmärk, lähtekatse, koodi commit või arhiveeritud versioon.
Mudel	Täpne hoidla ja revisjon, kaalude räsi, tokeniseerija, vestlusmall, lähtekaalu treeningustaadium.
Andmed	Failide räsid, päritolu ja õigused, jaotuse reegel, näidete ning tegelike tokenite arv.
Treening	Meetod, rank ja alpha, sihtmoodulid, LR, optimeerija, ajakava, partii, kogumine, pikkus, seeme, täpsus.
Keskkond	GPU, draiver, teekide tegelikud versioonid, kvantimise sätted, mälukasutuse tipp.
Hindamine	Ülesannete ja hindaja versioonid, genereerimisseaded, toorvastuste asukoht, inimese kontroll.
Väljund	Adapteri ja pakendi räsid, mõõtühikutega aeg ja kiirus, tulemus, piirangud, järgmine otsus.

Tööleht C. Vea päevik

Kirjuta iga olulise vea kohta sisend, mudeli vastus, oodatud vastuste hulk, vealiik ja kontrollija põhjendus. Lisa, kas viga kuulub mudelile, andmetele, hindajale või pakendile. Kui põhjus pole teada, märgi „teadmata". Sellest sünnib järgmise katse jaoks kasulikum andmestik kui üldisest soovist „tee mudel targemaks".

Katseidee: väike pime võrdlus
Näita hindajale kahe mudeli vastuseid juhuslikus järjekorras, varjates mudeli nime. Luba viiki ja mõlema vastuse tagasilükkamist. See vähendab mõnd ootustest tulenevat kallutatust, kuid ei eemalda hindamisjuhendi ega valimi puudusi.

Mida teha, kui midagi ei parane?

Vaata mõnd treeningnäidet otse. Kas mudelilt oodatakse sama ülesannet, mida hiljem hindad? Kontrolli, kas baas juba oskab seda ja võimalik paranemisruum on väike. Uuri, kas vastused muutusid sisuliselt paremaks, kuid parser loeb neid valesti. Kui kõik on korras, võib hüpotees olla vale: ka hästi korraldatud nulltulemus aitab järgmist sammu valida.

H15. Kolm muudatust korraga
Suurendasid rank'i, vahetasid andmed ja vähendasid õpisammu. Tulemus paranes. Mida saad järeldada ja kuidas uuriksid, milline muudatus aitas?

16. Õpetaja vastutus ja järgmine põlvkond

Selle peatüki järel: oskad siduda tehnilise töö andmete päritolu, inimeste vajaduste ja läbipaistva vastutusega.

Lapse kasvatamise võrdlus toob esile olulise mõtte: õpetaja valikud mõjutavad seda, mida õppija kordab ja mille eest ta heakskiitu saab. Mudeli puhul saab seda muuta konkreetseks tööks. Kes koostab näited? Kelle keelekasutus on esindatud? Milline vastus saab kõrge hinde? Kes märkab, et viisakas vastus on sisult vale?

Mudeli käitumissuund kujuneb andmete, treeningueesmärkide, tagasiside ja kasutuskeskkonna koosmõjus. Seda ei ole vaja nimetada mudelile antud elu mõtteks, et võtta vastutust tõsiselt. Tehniline süsteem võib mõjutada päris inimesi ka siis, kui tema võimaliku sisemise kogemuse kohta puudub kindel vastus.

Õigus teksti lugeda ei lahenda kõiki kasutusõigusi

Andmete puhul tuleb eraldi hinnata ligipääsu, kopeerimise ja kaevandamise alust ning hilisemat levitamist. Euroopa Liidu autoriõiguse direktiivi teadusuuringute erand on seotud tingimustele vastavate teadus- ja kultuuripärandiasutustega. Üldisema teksti- ja andmekaeve erandi puhul on muu hulgas oluline õiguspärane ligipääs ja õiguste reserveerimine. Konkreetse projekti jaoks tuleb vaadata ka kohaldatavat Eesti õigust. See lühiselgitus ei otsusta üksiku korpuse lubatavust. [23]

Creative Commonsi märgis ei ole kõigil failidel ühesugune luba. Säilita täpne litsents, versioon ja päritolu. Autorile viitamine ei asenda puuduvaid õigusi. Ka mudeli või andmete avaldamata jätmine ei muuda treeninguks tehtud koopiaid automaatselt lubatuks. Litsentsi mõju andmestikule, kaaludele ja väljunditele tuleb hinnata eraldi. [24]

Isikuandmed on omaette küsimus. Ära lisa isiklikke vestlusi lihtsalt sellepärast, et need parandavad dialoogi loomulikkust. Eelista eesmärgi jaoks vajalikku ja sobiva alusega andmekasutust; kontrolli tundliku info ning haruldaste tuvastatavate lõikude esinemist. Avaldamisel kirjelda, milliseid andmekategooriaid kasutati ja mida pole kontrollitud.

Üks keel ei tähenda üht häält

Eesti keelemudel vajab korrektset kirjakeelt, kuid kasutajad räägivad erinevalt. Murre, argikeel, õppija keel ja eri põlvkondade väljendus võivad kõik olla asjakohased. Määratle, millal mudel peaks parandama ning millal lihtsalt mõistma. Kõneleja alandamine ei ole keeleõpetus.

Hea kultuuritundlik mudel võib osata kirjeldada eri ajalootõlgendusi, eristada allikat ja arvamust ning tunnistada ebakindlust. Rahva tekstikogu ei ole kogu rahva ühtne teadmine. See peegeldab ka seda, kes sai avaldada, milline materjal säilis ja mida oli lihtne digitaalselt koguda.

Koostöö, millest sünnib parem mudel

Keeleekspert aitab koostada korrektseid rööpvorme ja põhjendusi. Arendaja muudab need kontrollitavaks torustikuks. Õpetaja märkab, kas selgitus on arusaadav. Kasutaja näitab, millistes olukordades abiline päriselt hätta jääb. Nende vaadete ühendamine annab rikkama hindamise kui üks protsent või ühe hindaja eelistus.

Katseidee: ühine veapank
Kogu kontrollitud veajuhtumeid koos päritolu, lubatud vastuste ja hindamisreegliga. Hoia avalikud õpetamisnäited ja varjatud lõpphindamise näited lahus. Versioonihaldus näitab, millal muutus keelereegel, hindaja või mudel; andmete avaldamine eeldab selleks sobivaid õigusi.

Järgmisele arendajale jäta kaasa ka ebaõnnestunud katsed. Kirjelda, mida muutsid, mida nägid ja mida sellest veel järeldada ei saa. See aitab vältida sama vea kordamist ning annab uuele ideele ausa lähtekoha.

H16. Vastutustundlik keeleabiline
Mudel parandab murdelise lause halvustavalt kirjakeelseks, kuigi kasutaja küsis ainult lause tähendust. Milles on ülesande viga ja millise treeningnäite ning hindamiskriteeriumi lisaksid?

17. Labori ettevalmistus: üks ülesanne, kaks rada

Selle peatüki järel: oskad valida oma arvutile sobiva raja, koostada näidisandmed ja säilitada lähtekaalu.

Nüüd teeme põhimõtetest töötava katse. Labori väljund on kohalik adapter ning enne ja pärast salvestatud vastused. Järgime sama väikest ülesannet mõlemal platvormil: vigase pöördevormiga lause parandamine kindla JSON-kujuga vastuseks.

Näidismudel on Qwen2.5-0.5B-Instruct. Valik on tehtud väikese mahu ja levinud arhitektuuri tõttu; see ei ole väide mudeli tänase paremuse kohta. See mudel erineb autori 27B projektimudelist. Mudelikaart kirjeldab ligikaudu 0,49 miljardi parameetriga instruct-mudelit. [33]

Mida see labor tõestab?
12 treeningnäidet ja 30 sammu on torustiku õppekatse. Need ei anna alust väita üldise eesti keele oskuse paranemist. Baas võib ülesande juba lahendada, tulemus võib jääda samaks või halveneda. Edukas tehniline läbimine tähendab, et treening toimib, adapter salvestub ja selle kasutamist saab võrrelda lähtekaaluga.

Näidisskriptide süntaks ning andmefailide loomine on kontrollitud. Käskude liidesed on kõrvutatud ametliku dokumentatsiooniga, kuid selle raamatu koostamisel ei käivitatud täielikku GPU- ega Maci treeningut. Seepärast tee oma masinas kõigepealt lühike katse ja salvesta töötavad versioonid. Praktika dokumentatsiooni kontrolli kuupäev on 6. september 2026.

Töökaust ja failide lugemine

Kõik järgmised käsud käivita Terminalis. Rida, mis algab pythoniga, käivitab Pythoni; koodiplokk nimega andmed.py tuleb salvestada samanimelisse UTF-8 tekstifaili, mitte ükshaaval terminali sisestada. Mitmes osas näidatud fail tuleb ühendada samasse faili esitatud järjekorras. Säilita taanded; ära kasuta tekstitöötlusprogrammi nutikaid jutumärke.

Kood on lisatud ka PDF-i manusfailidena. Manuseid toetavas PDF-lugejas saab need salvestada failipaanilt. Kui sinu lugeja seda ei paku, saad kasutada raamatus trükitud koodi. Skriptide käivitamine jääb sinu arvutis teadlikuks eraldi sammuks.

Vali esmalt peatükk 18 Ubuntu jaoks või 19 Maci jaoks ning loo seal kirjeldatud keskkond. Seejärel tule siia tagasi andmefaili loomiseks. Mõlemal rajal on töökausta nimi keelelabor. Kaust base sisaldab muutmata lähtekaalu; data andmeid; adapter või adapters-mac õpitud muudatusi; tulemuste JSON-failid toorvastuseid.

Andmed: väike ja nähtav näidiskogu

Salvesta järgmine programm faili andmed.py. See loob kolm JSONL-faili: igal real on üks terviklik vestlus. Treeningus on verbid „olema” ja „minema”, arenduses „lugema” ning lõppharjutuses „istuma”. See on lihtne nähtav eraldus, mitte piisav keeleline üldistumistest. Üks korrektne lause õpetab, et iga sisendit ei pea muutma.

andmed.py

```
import json
from pathlib import Path

# Väike õppekatse, mitte üldise keeleoskuse korpus.
train = [
    ('Mina on kodus.', 'Mina olen kodus.'),
    ('Sina olen siin.', 'Sina oled siin.'),
    ('Tema oled õpetaja.', 'Tema on õpetaja.'),

    ('Meie on valmis.', 'Meie oleme valmis.'),
    ('Teie olen kohal.', 'Teie olete kohal.'),
    ('Nemad on õues.', 'Nemad on õues.'),
    ('Mina läheb koju.', 'Mina lähen koju.'),
    ('Sina lähen kooli.', 'Sina lähed kooli.'),
    ('Tema lähed tööle.', 'Tema läheb tööle.'),
    ('Meie läheb linna.', 'Meie läheme linna.'),
    ('Teie lähen poodi.', 'Teie lähete poodi.'),
    ('Nemad läheb koju.', 'Nemad lähevad koju.'),
]
valid = [
    ('Mina loeb raamatut.', 'Mina loen raamatut.'),
    ('Sina loen kirja.', 'Sina loed kirja.'),
    ('Meie loeb uudiseid.', 'Meie loeme uudiseid.'),
    ('Teie loen teksti.', 'Teie loete teksti.'),
]
test = [
    ('Mina istub siin.', 'Mina istun siin.'),
    ('Sina istun seal.', 'Sina istud seal.'),
    ('Meie istub toas.', 'Meie istume toas.'),
    ('Teie istun autos.', 'Teie istute autos.'),
]

def row(bad, good):
    prompt = ('Paranda lause. Vasta ainult JSON-objektiga, '
              'mille ainus võti on parandus. Lause: ' + bad)
    answer = json.dumps({'parandus': good}, ensure_ascii=False)
    return {'messages': [
        {'role': 'user', 'content': prompt},
        {'role': 'assistant', 'content': answer},
    ]}

Path('data').mkdir(exist_ok=True)
for name, pairs in [('train', train), ('valid', valid),
                    ('test', test)]:
    path = Path('data') / (name + '.jsonl')
    with path.open('w', encoding='utf-8') as f:
        for bad, good in pairs:
            f.write(json.dumps(row(bad, good),
                               ensure_ascii=False) + '\n')
    print(path, len(pairs))
assert not (set(train) & set(valid) or set(train) & set(test)
            or set(valid) & set(test))
```

Käivita python andmed.py. Oodatud väljund on kolm failinime ning arvud 12, 4 ja 4. Ava iga fail ja loe vähemalt kaks rida. Kontrolli, et täpitähed säilisid ja assistendi vastus sisaldab JSON-objekti

tekstina. Uuel käivitamisel kirjutab programm need kolm laborifaili üle; oma suuremat korpust hoia eraldi.

Lähtekaalu allalaadimine

Paigaldatud keskkonnas salvesta järgmine lühike kood faili laadi.py. See küsib hoidla täpse revisjoni ja laadib selle kohalikku kausta. Esimene laadimine vajab internetti; treening kasutab seejärel kohalikke faile. Hoia revisjon ja kaalud koos, et baas ei muutuks katsete vahel. Hugging Face Hubi tööriistad võimaldavad revisjoniga allalaadimist. [34]

laadi.py

```
from pathlib import Path
from huggingface_hub import HfApi, snapshot_download

repo = 'Qwen/Qwen2.5-0.5B-Instruct'
revision = HfApi().model_info(repo).sha
snapshot_download(repo, revision=revision, local_dir='base')
Path('base-revision.txt').write_text(
    repo + '\n' + revision + '\n', encoding='utf-8')

print('Lähteversioon:', revision)
```

Käivita python laadi.py. Kontrolli, et tekkis base-revision.txt ning base-kaustas on konfiguratsioon, tokeniseerija failid ja mudelikaalud. Ära käivita seda programmi keset võrdluskatset uuesti: hilisem allalaadimine võib valida uuema revisjoni. Mudeli ja andmete litsentsid kuuluvad katsepassi.

Ühine hindaja mõlemale rajale

Fail hinda.py on toodud peatükis 20 ning lisatud PDF-i manusena. Salvesta see juba enne baashindamise käsku. See kasutab vaikimisi arenduskogumit, jätab toorvastused alles ja loeb eraldi korrektse JSON-kuju ning täpse vastuse. Neljast lausest saadud protsenti ära tõlgenda keeleoskuse üldhinnanguna.

H17. Kas oled alustamiseks valmis?
Pane kirja oma GPU VRAM või Maci ühendatud mälu, vaba kettaruum, valitud rada ning labori eesmärk. Milline fail tõendab hiljem, millist lähtekaalu kasutasid?

18. Ubuntu ja NVIDIA: esimene LoRA-treening

Selle peatüki järel: oskad luua eraldi keskkonna, kontrollida CUDA-t, käivitada adaptertreeningu ja laadida tulemuse tagasi.

1. Kontrolli draiverit ja loo keskkond

Eeldame toetatud Ubuntu LTS-i ning töötavat NVIDIA draiverit. Kui nvidia-smi ei näita kaarti, lahenda draiveri probleem enne Pythoni treeningut. Ubuntu ametlik juhend kirjeldab sobiva draiveri paigaldust; draiverinumbrit ei ole mõistlik võtta igale kaardile ühest vanast käsureast. [30]

Terminal: algkontroll

```
nvidia-smi
python3 --version
df -h .
free -h
```

Soovituslik lähtekoht on Python 3.11 või 3.12. Kui Ubuntu Pythonil puudub venv, paigalda python3-venv oma süsteemi paketihalduriga. Loo uus tühi projektikaust ja keskkond järgmiste käskudega. sudo pole Pythoni pakettide paigaldamisel selles keskkonnas vajalik.

Terminal: projekti loomine

```
mkdir keelelabor
cd keelelabor
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
```

PyTorchi paigalduskäsk vali ametlikust valikust: Linux, Pip, Python ja sinu kaardi ning draiveriga ühilduv CUDA-väljalase. Käivita seal antud käsk aktiveeritud keskkonnas. nvidia-smi näidatud CUDA number ei ole tõend, et Pythoni keskkonnas on õige PyTorch. Valmis PyTorchi paketiga ei pea iga labori jaoks eraldi kogu CUDA arenduskomplekti paigaldama. [29]

Terminal: ülejäänud teegid

```
python -m pip install transformers peft accelerate huggingface_hub
python -m pip check
```

Need paigalduskäsud ei lukusta tulevasi teegiversioone. Kui import või liides ei sobi, kontrolli teekide ühilduvust ametlikust dokumentatsioonist; ära uuenda töötava katse ajal kõike korraga. Pärast edukat proovijooksu salvestame täpsed versioonid.

Terminal: kas päris GPU-arvutus töötab?

```
python - <<'PY'
import torch
print('PyTorch:', torch.__version__)
print('CUDA ehitus:', torch.version.cuda)
assert torch.cuda.is_available(), 'CUDA puudub'
print('GPU:', torch.cuda.get_device_name(0))
x = torch.ones((32, 32), device='cuda')
print('Kontroll:', (x @ x)[0, 0].item())
PY
```

Oodatud arv on 32.0. Kui kaart on nähtav, kuid maatriksarvutus annab näiteks „no kernel image”, võib PyTorchi ehitus olla kaardipõlvkonnaga sobimatu. See pole andmeviga. „CUDA out of memory” tähendab seevastu mälupuudust. Ära liigu treeninguni enne, kui väike arvutus toimib.

2. Valmista andmed ja mõõda baas

Salvesta peatükkide 17 ja 20 failid andmed.py, laadi.py ning hinda.py töökausta. Käivita allolevad käsud. Linuxi näide kasutab tavalist LoRA-t 16-bitise baasi peal; 0,5B mudeli puhul pole 4-bitist kvantimist esimeseks harjutuseks vaja.

Terminal: lähtepunkt

```
python andmed.py
python laadi.py
python hinda.py --engine cuda --out enne.json
```

Ava enne.json ja loe neli vastust. Pane kirja, kas vead on pöördevormis, JSON-kujus või mõlemas. Kui kõik neli on õiged, ei ole selles väikeses kogumis paranemisruumi. See ei tee laborit mõttetuks: saad endiselt õppida salvestamist, taaslaadimist ja võimaliku halvenemise avastamist.

3. Salvesta treeninguprogramm

Järgmine fail treeni.py kasutab Transformersi Trainerit ja PEFT-i LoRA-adapterit. Trainitakse q_proj ja v_proj moodulite lisakaale. Õpisamm 2e-4 ning 30 sammu on selle väikese tehnilise harjutuse valikud, mitte autori 27B mudeli soovituslik retsept. [35, 36]

Programm kontrollib, et vestluse prefiks sobib malli ja et vastus mahub 512 tokeni sisse. Liiga pikk näide katkestab töö nähtava veaga, selle asemel et vaikselt ära kaduda. Kaomask jätab kasutaja teksti optimeerimissihist välja. See on siin valitud ülesande teostus, mitte ainus lubatud SFT viis.

treeni.py

```
import json
from pathlib import Path
import torch
from transformers import (AutoModelForCausalLM, AutoTokenizer,
    DataCollatorForSeq2Seq, Trainer, TrainingArguments, set_seed)
from peft import LoraConfig, get_peft_model

set_seed(42)
```

```
assert torch.cuda.is_available(), 'CUDA ei ole kasutatav'
base = 'base'
tok = AutoTokenizer.from_pretrained(base)
if tok.pad_token_id is None:
    tok.pad_token = tok.eos_token
tok.padding_side = 'right'

def dataset(name):
    result = []
    for line in Path('data/' + name + '.jsonl').read_text(
            encoding='utf-8').splitlines():
        messages = json.loads(line)['messages']
        ids = tok.apply_chat_template(messages, tokenize=True)
        prefix = tok.apply_chat_template(messages[:-1],
            tokenize=True, add_generation_prompt=True)
        assert ids[:len(prefix)] == prefix, 'Malli vastuolu'
        assert len(prefix) < len(ids) <= 512, 'Kontrolli pikkust'
        labels = [-100] * len(prefix) + ids[len(prefix):]
        result.append({'input_ids': ids,
            'attention_mask': [1] * len(ids), 'labels': labels})
    assert result, 'Andmestik on tühi'
    return result

bf16 = torch.cuda.is_bf16_supported()
model = AutoModelForCausalLM.from_pretrained(base,
    torch_dtype=torch.bfloat16 if bf16 else torch.float16)
model.config.use_cache = False
model = get_peft_model(model, LoraConfig(
    task_type='CAUSAL_LM', r=8, lora_alpha=16,
    lora_dropout=0.05, target_modules=['q_proj', 'v_proj']))
model.print_trainable_parameters()
args = TrainingArguments(
    output_dir='runs/proov', max_steps=30,
    per_device_train_batch_size=1,
    per_device_eval_batch_size=1,
    gradient_accumulation_steps=4, learning_rate=2e-4,
    bf16=bf16, fp16=not bf16, logging_steps=5,
    save_strategy='steps', save_steps=15, save_total_limit=2,
    eval_strategy='steps', eval_steps=15,
    gradient_checkpointing=True,
    gradient_checkpointing_kwargs={'use_reentrant': False},
    report_to='none', seed=42)
trainer = Trainer(model=model, args=args,
    train_dataset=dataset('train'),
    eval_dataset=dataset('valid'),
    data_collator=DataCollatorForSeq2Seq(
        tok, padding=True, label_pad_token_id=-100))
trainer.train()
trainer.save_model('adapter')
tok.save_pretrained('adapter')
print('Valmis: adapter; see ei ole kogu baasmudel.')
```

4. Käivita ja jälgi

Terminal: treening ja uus hindamine

```
python treeni.py
python hinda.py --engine cuda --adapter adapter --out parast.json
python -m pip freeze > requirements-lock.txt
```

Programmi alguses peab treenitavate parameetrite arv olema nullist suurem ja koguarvust palju väiksem. Seejärel ilmuvad sammude kaupa kaonäidud, kaks arendushindamist ja salvestused. Edukal lõpul tekib adapter-kaust. Vaata enne.json ja parast.json vastuseid kõrvuti. Kadu võib langeda ilma, et kõik neli vastust paraneksid.

Vahepunktid asuvad runs/proov kaustas. Need võivad sisaldada optimeerija olekut ning võimaldada sama jooksu jätkamist. Lõplik adapter-kaust on eelkõige kasutamiseks mõeldud lisakaalud; pelgalt adapteri laadimine ei taasta automaatselt optimeerija ajalugu. Jätkamiseks kasuta Traineri resume_from_checkpoint valikut koos sobiva vahepunktiga ja sama katse seadistusega. [35]

5. Millal minna suuremaks?

Alles pärast väikese labori läbimist vali 1,5B või 3B lähtekaal ja tee uus mahukatse uues kaustas. Kontrolli tema malli ja moodulinimesid. 7B või 27B jaoks ei piisa alati ainult hoidlanime vahetamisest: QLoRA lisab kvanditud laadimise, teegitoe ja külmutatud baasi treeninguks ettevalmistamise. Peatükk 6 annab põhimõtte, kasutatava mudeli ametlik näide konkreetse teostuse.

H18. Treening lõppes, mida säilitad?
Nimeta neli faili või failirühma, milleta oleks tulemust hiljem raske korrata. Selgita, miks adapter üksi ei ole kogu mudel.

19. Apple Siliconi Mac: esimene MLX-treening

Selle peatüki järel: oskad sama andmestikuga teha kohaliku adapterkatse Macis ja jälgida ühendatud mälu piiranguid.

1. Kontrolli platvormi

See rada kasutab MLX LM-i ning Apple Siliconi GPU-d, mitte CUDA-t. Ava Terminal, kontrolli arhitektuuri ja vaata Activity Monitorist mälu mahtu ning mälusurvet. Kui uname -m annab x86_64, kontrolli, kas masin on Intel või terminal töötab Rosetta kaudu. Selle raja jaoks on vaja toetatud macOS-i ja ARM-keskkonda. [31, 32]

Terminal: Maci keskkond

```
uname -m
python3 --version
mkdir keelelabor
cd keelelabor
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install 'mlx-lm[train]' huggingface_hub
python -m pip check
python -c "import mlx.core as mx; print(mx.default_device())"
```

Eeldame, et python3 viitab paigaldatud ARM-versiooni Pythonile, näiteks 3.11 või 3.12, mis sobib kasutatava MLX paketiga. Kui sobivat paketti ei leita, kontrolli kõigepealt Pythoni, macOS-i ja protsessori ühilduvust. Ära asenda juhuslikult kogu süsteemi Pythonit.

2. Kasuta samu andmeid ja salvestatud baasi

Salvesta peatüki 17 andmed.py ja laadi.py ning peatüki 20 hinda.py. Väike Qwen2.5 lähtekaal sobib selle labori näiteks; MLX-i tugi tuleb suurema või teistsuguse mudeli valikul uuesti kontrollida. MLX LM pakub nii genereerimist kui ka adaptertreeningut. [32]

Terminal: enne treeningut

```
python andmed.py
python laadi.py
python hinda.py --engine mlx --out enne-mac.json
mlx_lm.lora --help
```

3. Käivita väike adapterkatse

MLX LM-i ametlik treeninguliides loeb vestlusnäiteid JSONL-failidest. --mask-prompt jätab selle labori viimase assistendivastuse õpetussihiks. Siin kohandame nelja kihti ja kasutame mikropartiid 1. CUDA näite ning MLX näite adapteri sihtmoodulid ja vaikeseaded pole täpselt samad: neid jooksusid ei tohi esitada riistvara õiglase kiirusvõrdlusena. [37]

Terminal: treening

```
mlx_lm.lora --model base --train --data data \
  --adapter-path adapters-mac --iters 30 \
  --batch-size 1 --num-layers 4 --learning-rate 0.0002 \
  --max-seq-length 512 --mask-prompt \
  --steps-per-eval 15 --val-batches 4 --save-every 15
```

Kaldkriips rea lõpus jätkab sama terminalikäsku järgmisel real; ära lisa selle järele tühikut. Kui sinu versioon mõnd lippu ei tunne, võrdle --help väljundit dokumentatsiooniga ja salvesta kasutatud versioon. Eduka jooksu järel peab adapters-mac sisaldama adapteri kaale ja seadistust. Ära kustuta base-kausta.

Terminal: pärast treeningut

```
python hinda.py --engine mlx --adapter adapters-mac \
  --out parast-mac.json
python -m pip freeze > requirements-lock.txt
```

4. Mälusurve ja järgmine katse

Jälgi Activity Monitoris mälusurvet, mitte ainult vaba mälu numbrit. Suur swap ja järsult aeglustuv masin viitavad, et koormus on liiga suur. Sulge muud mahukad rakendused, vähenda jada pikkust või treenitavate kihtide arvu ning vali väiksem mudel. Ära hoia samal ajal teises rakenduses suurt vestlusmudelit laetuna.

Suurema mudeli puhul võib kasutada MLX-iga ühilduvat kvanditud baasi: sellisel juhul on adaptertreening QLoRA tüüpi. Säilita kvantimise seaded ja lähteversioon. Maci ning PEFT-i adapterifaile ei saa eeldada omavahel vahetatavateks; sama meetodinimi ei taga sama failistruktuuri.

MLX-i adapteri pealt jätkamiseks on olemas --resume-adapter-file. Kontrolli oma versiooni dokumentatsioonist, milline olek taastatakse. Olemasoleva adapteri edasine õpetamine ja katkestatud optimeerimisjooksu täpne taastamine on eri eesmärgid. [37]

H19. 32 GB ei võrdu 32 GB
Miks ei saa 32 GB ühendatud mäluga Maci ja 32 GB VRAM-iga NVIDIA kaardi puhul eeldada täpselt sama treeningumahutavust? Nimeta kaks põhjust.

20. Hindamine, tõrkeotsing ja iseseisev lõputöö

Selle peatüki järel: oskad kontrollida vastuse sisu, leida tõrke etapi ning kavandada esimese sisulise keelekatse.

Hindamisprogramm: hinda.py

See ühine programm valib käsurealt CUDA või MLX-i. Kummalgi rajal peab olema paigaldatud ainult selle raja teegistik. Programm võrdleb JSON-objekte, seega võtmete ümbruse tühikud ei muuda tulemust. Lubamatu lisatekst või koodiplokk muudab JSON-kuju kontrolli vääraks. See on labori teadlikult range vormistusnõue.

hinda.py

```
import argparse
import json
from pathlib import Path

p = argparse.ArgumentParser()
p.add_argument('--engine', choices=['cuda', 'mlx'], required=True)
p.add_argument('--adapter')
p.add_argument('--split', choices=['valid', 'test'], default='valid')

p.add_argument('--out', required=True)
a = p.parse_args()
rows = [json.loads(s) for s in Path(
    'data/' + a.split + '.jsonl').read_text(
        encoding='utf-8').splitlines()]

if a.engine == 'cuda':
    import torch
    from transformers import AutoTokenizer, AutoModelForCausalLM
    from peft import PeftModel
    assert torch.cuda.is_available()
    tok = AutoTokenizer.from_pretrained('base')
    dtype = (torch.bfloat16 if torch.cuda.is_bf16_supported()
             else torch.float16)
    model = AutoModelForCausalLM.from_pretrained(
        'base', torch_dtype=dtype).to('cuda')
    if a.adapter:
        model = PeftModel.from_pretrained(model, a.adapter)
    model.eval()
    def answer(messages):
        prompt = tok.apply_chat_template(messages,
            tokenize=False, add_generation_prompt=True)
        x = tok(prompt, add_special_tokens=False,
                return_tensors='pt').to('cuda')
        with torch.inference_mode():
            y = model.generate(**x, max_new_tokens=80,
                do_sample=False, pad_token_id=tok.eos_token_id)
        return tok.decode(y[0, x.input_ids.shape[1]:],
                          skip_special_tokens=True)
```

```
else:
    from mlx_lm import load, generate
    from mlx_lm.sample_utils import make_sampler
    model, tok = load('base', adapter_path=a.adapter)
    def answer(messages):
        prompt = tok.apply_chat_template(messages,
            tokenize=False, add_generation_prompt=True)
        return generate(model, tok, prompt=prompt,
            max_tokens=80, sampler=make_sampler(temp=0))

results = []
for row in rows:
    messages = row['messages']
    expected = json.loads(messages[-1]['content'])
    text = answer(messages[:-1]).strip()
    try:
        parsed = json.loads(text)
    except json.JSONDecodeError:
        parsed = None
    shape = isinstance(parsed, dict) and set(parsed) == {'parandus'}
    results.append({'prompt': messages[:-1], 'response': text,
        'expected': expected, 'json_ok': shape,
        'exact_ok': shape and parsed == expected})
Path(a.out).write_text(json.dumps(results,
    ensure_ascii=False, indent=2), encoding='utf-8')
print('JSON-kuju:', sum(r['json_ok'] for r in results), '/', len(results))
print('Täpne vaste:', sum(r['exact_ok'] for r in results), '/', len(results))
```

Tulemusest otsuseni

Kui JSON-kuju paranes, aga pöördevorm mitte, õppis mudel vähemalt osa väljundilepingust. Kui mõlemad halvenesid, vaata esmalt näiteid ja treeningu koormust. Kui kõik oli juba enne õige, vali sisulise katse jaoks raskemad uued arendusülesanded. Nelja lausega laborist ei saa hinnata statistiliselt usaldusväärset väikest paranemist.

Kood ei aktsepteeri automaatselt kõiki tähenduselt samaväärseid parandusi. Vaata valeks loetud vastused käsitsi üle. Kui laiendad aktsepteeritud vastuste hulka, hinda nii baasi kui adapterit sama uue reegliga. Üks õige rööpvorm ei tohi panna sind treenima mudelit valest asjast loobuma.

Lõppharjutuseks kasuta --split test valikut mõlema mudelivariandi hindamisel. See eraldab näidetes uue verbi, kuid raamatus nähtav nelja lause komplekt ei ole avalikuks kvaliteediväiteks sobiv pimetest. Oma päris projekti jaoks koosta eraldi piisavalt lai hindamine, mille järgi arendusotsuseid ei tehta.

Tõrkeotsing: alusta vea kihist

Sümptom	Esimene kontroll	Järgmine tegevus
nvidia-smi ei tööta	Draiver ja kaardi nähtavus süsteemis.	Paranda draiver enne Pythoni pakettide muutmist.
CUDA on False	Aktiivne Python ja PyTorchi ehitus.	Kontrolli which python ning ametlikku paigalduskäsku.
No kernel image	GPU arhitektuuri ja teegi toe sobivus.	Vali kaarti toetav PyTorch; proovi väikest arvutust uuesti.
Out of memory / suur swap	Muud protsessid, jada, partii, mudel.	Vähenda koormust; ära kustuta juhuslikult andmeid.
Kadu on NaN või inf	Täpsus, õpisamm, andmed ja gradient.	Peata pikk jooks; korda väiksema LR-i ja nähtava näitega.
0 treenitavat parameetrit	Adapteri moodulinimed ja külmutamine.	Paranda LoRA sihtmoodulid enne jätkamist.
Kadu langeb, oskus ei parane	Arenduse vastused, ülesannete sobivus.	Kontrolli vormi, sisu ja hindajat eraldi.
Pärast laadimist pole mõju	Kas adapter tegelikult laaditi õigele baasile?	Võrdle kaalukonfiguratsiooni, failiteid ja toorvastuseid.
Katkestuse järel ei saa jätkata	Kas alles on vahepunkt või ainult adapter?	Vali õige taastamisviis; säilita lähtefailid ja seaded.

Esimene sisuline projekt

Kui tehniline labor toimib, vali üks päris kasutusülesanne: näiteks omadussõna ja nimisõna ühildumine koos lühikese selgitusega. Koosta alustuseks mõnisada kontrollitud näidet ning eraldi uued arendus- ja lõpphindamise ülesanded. See maht on praktiline lähtekoht, mitte universaalne piisavuse piir.

Lisa ühe vormi asemel eri sõnu, arve, käändeid ja lausekontekste. Esinda õigeid lauseid, kus parandust pole vaja. Ära genereeri kõiki andmeid ühe malli ja sama väikese sõnaloendiga ning nimeta seda laialdaseks katvuseks. Kasuta peatüki 9 veakaarti ja lase osa näiteid teisel inimesel kontrollida.

Tee baashindamine, lühike mahukatse ja üks kavandatud treening. Salvesta keeleoskuse kõrval vastuste loomulikkus, juhiste järgimine ning mõned varasemad olulised oskused. Alles pärast seda otsusta, kas vajad rohkem andmeid, uut meetodit või tugevamat lähtekaalu.

Lõputöö: anna oma katse teisele inimesele üle

Esitatav tulemus	Mille järgi loed töö tehtuks?
Ühe lehe katsepass	Eesmärk, riistvara, revisjonid, seaded ja kulud on selged.
Andmete kirjeldus	Päritolu, õigused, jaotus ja näidete kontroll on dokumenteeritud.
Käivitatav töövoog	Teine inimene leiab failid, sõltuvused ja käskude järjekorra.
Baas ja adapter	Mõlemad on tuvastatavad; adapter on taas laaditud ja kasutatud.
Enne/pärast vastused	Toorvastused, hindaja versioon ja nimetajad on säilinud.
Vea- ja järelduspäevik	Kirjas on vähemalt kolm tähelepanekut ning üks piirang.

Õppija on raamatu praktilise eesmärgi saavutanud siis, kui ta oskab ise valida jõukohase katse, valmistada andmed, käivitada treeningu, laadida tulemuse tagasi ning põhjendada, mida tulemus näitas. Suurim skoor ei ole selle lõputöö ainus hindamiskriteerium. Oluline on, et töö oleks arusaadav ja korratav.

H20. Anna järgmine samm
Sinu adapter annab korrektset JSON-i 4/4, kuid õige lause ainult 2/4; baas andis vastavalt 2/4 ja 2/4. Mis paranes, mis jäi tõendamata ja millise katse kavandad järgmiseks?

Harjutuste vastused ja arutlused

Vastused näitavad põhjendamise suunda. Oma projekti puhul kontrolli alati küsimuses nimetamata eeldusi.

H1. Raamatute lisamine

Kui failid lisatakse otsingukogusse, muutub kättesaadav taustmaterjal, mitte tingimata mudeli kaalud. RAG toob sobivaid lõike vastamise konteksti. Kaalude muutmiseks on vaja tegelikku optimeerimist. Võrdlus lapse juurdepääsuga raamatukogule aitab mõista info kättesaadavust, kuid raamatute olemasolu ei tähenda veel läbiloetust ega õpitut. Mõlemal juhul tuleb kontrollida, kas süsteem kasutab allikaid õigesti.

H2. Vestluse jooksul õppimine

Varasema vestluse kasutamine võib olla kontekstis kohanemine või rakenduse püsiv mälu. See ei tõenda kaalude uuendamist. Küsi, kus info säilib, millal seda loetakse ja kas käivitati treening.

H3. Vali sekkumine

Sageli muutuvat hinnakirja sobib küsida kontrollitud allikast või otsingust. Käänamise korduvate vigade puhul võib olla põhjendatud sihitud treening, kui hindaja ja viip on kontrollitud. Esmalt täpsusta, kas probleem on teadmises, oskuses või süsteemi ühenduses.

H4. Kattuvus

Juhuslik lõigujaotus võib panna sama autori või teose väga sarnase materjali mõlemale poole. Eralda eesmärgiga sobivalt dokumentide, teoste või autorite järgi ning otsi täpseid ja ligikaudseid duplikaate. Ükski jaotus ei taga automaatselt täielikku sõltumatust.

H5. Kärpimine

2900 - 2048 = 852 tokenit; 852 / 2900 × 100 = 29,4%. Kontrolli, milline osa kadus. Kui kadus assistendi vastus, ei anna näide kavandatud õpetussignaali. Eelista sobivat pikkust, mõtestatud tükeldamist või ülejäägi säilitamist.

H6. Adapteri maht

Sama rank ei taga sama mahtu: erineda võivad sihtmoodulite arv ja mõõtmed, lisaks salvestatud kihid ning andmetüüp. Kontrolli tensorite nimesid, kujusid ja dtype-välju. Kolmekordne erinevus võib tulla nende koosmõjust. Näiteks 159,4 miljonit parameetrit vajavad kahebaidise esitusega ligikaudu 318,8 MB ja neljabaidisega 637,6 MB, enne failivormingu lisakulu. 640 MB fail viitab kontrollivajadusele, näiteks FP32 esitusele; see ei tõenda automaatselt topeltparameetrite arvu.

H7. Efektiivne partii

Ühel seadmel annab partii 2 ja kogumine 8 efektiivseks partiiks 16. 960 näite korral on täispartiidega 60 optimeerimissammu. Partii 1 ja kogumine 16 annab sama arvu, kuid kiirus ja numbriline käitumine ei pea olema identsed.

H8. Hindamise sõltumatus

Kui mudelit valiti korduvalt sama kogumi tulemuse järgi, toimis kogum arenduskogumina. Loo uutest sobivalt eraldatud ülesannetest lõpphindamine ja fikseeri protokoll enne tulemuste vaatamist. Ümbernimetamine ei taasta sõltumatust.

H9. Õige rööpvorm

Kontrolli vastust usaldusväärse keeleallika ja vajaduse korral eksperdiga. Paranda lubatud vastuste hulka, versiooni hindaja ning hinda mõlemad mudelid sama parandatud reegliga uuesti. Ära muuda üksnes uuema mudeli tulemust.

H10. Viisakas vale

Sõbralik toon ja faktiline õigsus on eri mõõtmed. Eesti keele 14 käände asemel seitsme nimetamine jääb sisuliseks veaks ka heas toonis. Hindamisjuhend peab käsitlema mõlemat eraldi.

H11. Sammude võrdlus

35,3 × 3600 / 3364 annab umbes 37,8 sekundit optimeerimissammu kohta. 10,59 sekundi võrdlemiseks on vaja teada partii suurust, jada pikkust, kogumist, mõõtmispiire ja seda, mida nimetati sammuks. Võimsuspiir ei ole mõõdetud keskmine võimsus.

H12. Pakendamise kontroll

Võrdle sama viibaga adapterit ja ühendatud kaale, seejärel ühendatud kaale ning eksporditud paketti. Kontrolli vestlusmalli, erimärke ja genereerimisseadeid. Nii saab vea etapi kitsendada enne uue treeningu kulu.

H13. Ülekantav töö

Uuesti saab sageli kasutada kontrollitud andmeid, hindamisülesandeid ja hindamisjuhendit. Kasulikud on ka puhastuskood ning katsepäevik. Vorming, tokeniseerimine, õigused ja ülesannete sobivus vajavad siiski uut kontrolli.

H14. Aus tulemuslause

Näide: „Projekti arenduses kasutatud 151 automaatselt hinnatava ülesande osapunktidega koondhinne tõusis 61,7%-lt 86,1%-ni; see ei ole sõltumatu lõpphinnang mudeli üldisele eesti keele oskusele.”

H15. Põhjuse leidmine

Sa tead, et uus tervikseadistus töötas selles katses paremini. Üksiku muudatuse mõju jääb eristamata. Võrdle põhjendatud kontrollkatsetes tegureid eraldi või kavanda nende koosmõju uuriv katse. Kui erinevus on väike, arvesta juhuslikkusega.

H16. Kasutaja eesmärk

Abiline vastas teisele ülesandele ja halvustas keelekuju. Näide peaks õpetama tähenduse selgitamist austavas toonis; kirjakeelset vastet võib pakkuda asjakohaselt. Mõõda ülesande järgimist, tähenduse õigsust ja tooni eraldi, kaasates murde tundja.

H17. Valmisolek

Kirjelda täpset mälu liiki, vaba kettaruumi ja oma platvormi rada. Eesmärk on esialgu torustiku kontroll, mitte rahvusliku keelemudeli valmimine. base-revision.txt seob lähtekaalu hoidla ja revisjoniga; ka failiräsid ning keskkonnakirjeldus aitavad tulemust taastada.

H18. Säilitamine

Säilita lähtekaal või taastatav revisjon, andmefailid ja nende jaotus, treeningukood koos sõltuvuste ning seadetega ja adapter. Hindamiseks on vaja ka toorvastuseid ning hindajat. Adapter kirjeldab üksnes osa õpitud muudatusi ja vajab sobivat baasi.

H19. Erinevad mälusüsteemid

Macis kasutavad sama kogumälu ka süsteem ja CPU-rakendused. NVIDIA kaardil on eraldi VRAM ning arvutil lisaks RAM. Erinevad ka teostus, arvutustäpsus, vahetulemused ja GPU-le kättesaadav eelarve. Sama number ei tähenda sama kiirust ega mahutavust.

H20. Vorm ja sisu

Vastuse JSON-kuju paranes sellel neljal näitel. Keelelise paranduse täpsus jäi samaks; üldist paranemist pole näidatud. Vaata kahte sisuviga, kontrolli hindajat ja lisa põhjendatud uued treeningnäited. Mõõda tulemust uutel arenduslausetel ning säilita vormikontroll.

Sõnastik

Mõiste	Tähendus selles raamatus
Aktivatsioon	Arvutuse vahetulemus; treeningus võib selle säilitamine nõuda palju mälu.
Adapter	Väike treenitav lisaparameetrite kogum, millega kohandatakse suuremat mudelit.
Alpha	LoRA uuenduse skaleerimise seadistus; toime sõltub ka rank'ist ja teostusest.
Baas	Katse lähtekaal; see võib juba olla juhendite järgimiseks kohandatud.
BF16 / FP16 / FP32	Arvude esituse vormingud, vastavalt 16, 16 ja 32 bitiga; nende omadused erinevad.
CPT	Continued pretraining: olemasoleva mudeli eeltreeningu jätkamine uuel tekstisegul.
DPO	Direct Preference Optimization: eelistatud ja mitte-eelistatud vastuste abil kohandamine.
Epohh	Üks kavandatud läbimine treeningandmetest; proovivõtt ja kärpimine võivad pilti muuta.
GGUF	Mudeli kasutamiseks mõeldud failivorming, mis võib sisaldada ka metaandmeid ja malli.
Gradient	Näitab, kuidas parameetri väike muutus mõjutab sihtfunktsiooni lokaalselt.
Gradient accumulation	Gradientide kogumine mitmest mikropartiist enne optimeerija sammu.
Hüperparameeter	Treeningu seadistus, näiteks õpisamm või partii suurus.
Inferents	Treenitud mudeli kasutamine väljundi arvutamiseks.
Kaalud	Mudeli õpitud arvulised parameetrid.
Kontekst	Sisend, millele mudel konkreetses arvutuses saab toetuda.
Kvantimine	Arvulise esituse täpsuse vähendamine mälu või arvutuskulu muutmiseks.
LoRA	Low-Rank Adaptation: kaalumuutuse esitamine väiksemate maatriksite korrutisena.
Loss ehk kadu	Optimeeritav arvuline veamõõt; ei võrdu kogu kasutuskvaliteediga.

Mõiste	Tähendus selles raamatus
LR	Learning rate ehk õpisamm: optimeerija uuenduse skaalat mõjutav seadistus.
Mask	Märgistus, mis määrab näiteks tähelepanu lubatud seosed või loss'is osalevad kohad.
MoE	Mixture of experts: mudel kasutab sisendi jaoks valitud ekspertplokke.
Optimeerija	Algoritm, mis kasutab gradiente parameetrite uuendamiseks.
Packing	Mitme teksti või näite koondamine treeningjadadesse; piiride käsitlus tuleb määrata.
PPL	Perplexity: tokenite keskmisest negatiivsest logitõenäosusest tuletatud mõõdik.
QLoRA	LoRA treening koos külmutatud baaskaalu kvanditud esitusega.
RAG	Retrieval-augmented generation: otsitud materjali kasutamine genereerimise kontekstis.
Rank	LoRA maatriksite vahemõõde, mis mõjutab treenitavate parameetrite arvu.
Rektsioon	Sõna nõue teise sõna vormi või seose kohta, näiteks „sõltub millest”.
Replay	Varasema või üldisema materjali lisamine treeningusegusse oskuste säilitamise toetamiseks.
SFT	Supervised fine-tuning: kohandamine etteantud soovitud väljundite järgi.
Token	Tokeniseerija määratud tekstiühik; ei pea olema sõna ega täht.
Valideerimine	Arendusaegne kontroll, mida saab kasutada seadistuste ja mudelivariandi valimiseks.
VRAM	Graafikaprotsessori mälu; erista seda arvuti põhimälust ja kettaruumist.
Ülesobitumine	Treeningmaterjalile sobivus kasvab viisil, mis ei kandu vajalikule uuele materjalile.

Faktikontrolli ja toimetamise märkmik

Selles lisas on nähtavad olulisemad sisulised täpsustused võrreldes algmaterjaliga. See ei ole uus mõõtmistabel: dokumenteerimata projektitulemusi ei ole tagantjärele mõõdetuks kuulutatud.

Teema	Täpsustus selles väljaandes
Inimene ja mudel	Õppimise analoogia on säilitatud; aju ja LLM-i mehhanisme ei kuulutata samaks.
Emotsioonid	Emotsiooni kirjeldamine ja toetav käitumine on eristatud subjektiivsest tundmisest.
Andmete lisamine	Kontekst, otsing, rakenduse mälu ja kaalude treening on eraldi mõisted.
Treeninguetapid	CPT, SFT ja DPO ei ole igas projektis kohustuslik astmestik.
Andmekvaliteet	„Vähem on alati parem” asemel hinnatakse kvaliteeti, katvust ja eelarvet koos.
Dialoog	Kohaliku mitmevoorulise andmestiku puudumine ei tähenda, et lähtekaal dialoogi ei õppinud.
Maskimine	Assistendi vastuse mask on eesmärgist sõltuv valik; see pole kõigi treeningute ainuvõimalus.
DPO	Beta ei ole kõva kauguspiir; välised paarid ei ole olemuslikult kõlbmatud.
Rank	Eri tingimustel tehtud katsetest ei saa järeldada rank 16 üldist paremust.
Mälu	Täistreeningu mälu eristab kaale, gradiente, optimeerijat ja aktivatsioone.
Adapterifail	159,4 miljoni parameetri failimaht sõltub andmetüübist; 640 MB vajab täpsustust.
Tokenite maht	Sõnade suhtarv on hinnang; pakkimise järel tuleb lugeda tegelikud tokenid ja loss'i kohad.
Treeningu aeg	35,3 tundi ja 3364 sammu annavad umbes 37,8 sekundit sammu kohta.
Energia	Võimsuspiirist arvutatud kulu on tingimuslik hinnang, mitte pistikust mõõdetud tarve.
86,1% tulemus	Nimetaja on 151 ning arvesse lähevad osapunktid; tulemus on arenduskogumilt.

Teema	Täpsustus selles väljaandes
PPL	30,6% vähenemine ei tähenda sama suurt grammatika paranemist.
Oskuste säilimine	Mõne ülesande paranemine ei tõesta unustamise puudumist.
Mõõdikute ühendamine	Varasema mudelivariandi välishindamist ei esitata lõppvariandi tulemusena.
DARE	Uuenduse elementide eemaldamine on juhuslik, mitte lihtsalt väikseimate kärpimine.
Pakendamine	Mall võib olla ka GGUF metaandmetes; Ollama sätted sõltuvad mudelist ja versioonist.
Õigused	Avaldamata jätmine üksi ei lahenda treeningandmete kasutusõigust.

Mida oleks järgmiseks vaja kontrollida?

Kõige väärtuslikum lisatöö on seotud algfailidega: kogu CPT segu tegelik tokeniloendus, treeningus kasutatud jadad ja maskid, riistvaramõõtmiste logid, adapteri andmetüübid ning lõppmudeli täpne revisjon. Hindamise jaoks on vaja toorvastuseid, osapunktide reegleid ja andmejaotuse päritolu. Need kontrollid võivad täpsustada projekti järeldusi rohkem kui järgmine pikk treening.

Allikad ja edasi lugemine

Numbrid tekstis viitavad järgmistele allikatele. Viited on klõpsatavad. Tehnilised teegid ja mudelikaardid muutuvad; katse kordamiseks salvesta oma kasutatud revisjon. Põhiosa veebiallikad kontrolliti 5. septembril ning praktiliste lisade allikad 6. septembril 2026. Projektimõõtmiste põhiline allikas on autori esitatud õppematerjal, mitte sõltumatu replikatsioon.

[1] Caucheteux jt (2023). Inimaju ja keelemudelite ennustuste võrdlus. Nature Human Behaviour. Ava allikas ↗

[2] Nature Communications (2026). Uurimus aju ja keelemudelite vastavuse tõlgendamise piiridest. Ava allikas ↗

[3] Carlini jt (2021). Extracting Training Data from Large Language Models. Ava allikas ↗

[4] Vaswani jt (2017). Attention Is All You Need. Ava allikas ↗

[5] Qwen. Qwen3.8-27B ametlik mudelikaart ja hoidla konfiguratsioon. Ava allikas ↗

[6] Hoffmann jt (2022). Training Compute-Optimal Large Language Models. Ava allikas ↗

[7] Gururangan jt (2020). Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. Ava allikas ↗

[8] Ouyang jt (2022). Training language models to follow instructions with human feedback. Ava allikas ↗

[9] Rafailov jt (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. Ava allikas ↗

[10] Penedo jt (2025). FineWeb2: One Pipeline to Scale Them All - Adapting Pre-Training Data Processing to Every Language. Ava allikas ↗

[11] Zheng jt (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. Ava allikas ↗

[12] Hugging Face TRL. SFT Trainer: andmevorming, maskimine ja pakkimine. Ava allikas ↗

[13] Hu jt (2021). LoRA: Low-Rank Adaptation of Large Language Models. Ava allikas ↗

[14] Hugging Face PEFT. LoRA, rsLoRA ja DoRA konfiguratsioon ning kasutus. Ava allikas ↗

[15] Dettmers jt (2023). QLoRA: Efficient Finetuning of Quantized LLMs. Ava allikas ↗

[16] Hugging Face TRL. DPO Trainer. Ava allikas ↗

[17] Hugging Face Transformers. Perplexity of fixed-length models. Ava allikas ↗

[18] Eesti Keele Instituut. Eesti keele käsiraamat (2020). Ava allikas ↗

[19] EstNLTK. Eesti keele töötlusvahendid ja Vabamorfi liides. Ava allikas ↗

[20] Hugging Face Transformers. Model training anatomy: treeningumälu komponendid. Ava allikas ↗

[21] Ollama. Modelfile'i ametlik dokumentatsioon ja mudelite importimine. Ava allikas ↗

[22] Arcee AI. mergekit: mudelite liitmise meetodid ja teostus. Ava allikas ↗

[23] Euroopa Parlament ja nõukogu. Direktiiv (EL) 2019/790, eelkõige artiklid 3 ja 4. Ava allikas ↗

[24] Creative Commons (2026). Guidance on Using CC Licenses in an AI Ecosystem. Ava allikas ↗

[25] Carnegie Mellon University, Eberly Center. Retrieval Practice. Ava allikas ↗

[26] Pert Lomp. Keelemudelite treenimine: autori esitatud 50-leheküljeline algmaterjal, 1. september 2026. Projektitulemused ja katsekogemus.

[27] Pert Lomp. qwen38-et projektihoidla; sealhulgas PARANDUSED.md. Autori avalik projektikirjeldus. Ava allikas ↗

[28] Lewis jt (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Ava allikas ↗

[29] PyTorch. Start Locally: platvormi ja CUDA jaoks sobiva paketi valimine. Ava allikas ↗

[30] Ubuntu. NVIDIA drivers installation: draiverite paigaldus ja kontroll. Ava allikas ↗

[31] MLX. Ametlik dokumentatsioon: ühendatud mälu ja arvutusraamistik. Ava allikas ↗

[32] MLX LM. Ametlik hoidla: kohalike mudelite kasutamine ja kohandamine Apple Siliconil. Ava allikas ↗

[33] Qwen. Qwen2.5-0.5B-Instruct ametlik mudelikaart. Ava allikas ↗

[34] Hugging Face Hub. Allalaadimine ja revisjonide kasutamine. Ava allikas ↗

[35] Hugging Face Transformers. Trainer ja TrainingArguments. Ava allikas ↗

[36] Hugging Face PEFT. Quicktour: LoRA loomine, salvestamine ja laadimine. Ava allikas ↗

[37] MLX LM. Fine-Tuning with LoRA or QLoRA: ametlik laboriliidese juhend. Ava allikas ↗

Õpperaamatu täiendamisel kasutati tehisintellekti abi struktuuri, selgituste, arvutuste ja allikavõrdluse koostamisel. Tulevase avaldamise eel tasub lasta eesti keele näited ja projekti mõõtmisandmed üle vaadata nende eest vastutaval inimesel.