TARTU ÜLIKOOL
Loodus- ja täppisteaduste valdkond
Arvutiteaduse instituut
Informaatika õppekava

Anna Maria Tammin

GPT-3.5 peenhäälestamine terviseandmete märgendamiseks

Bakalaureusetöö (9 EAP)

Juhendaja: Hendrik Šuvalov, MSc

Tartu 2024

GPT-3.5 peenhäälestamine terviseandmete märgendamiseks

Lühikokkuvõte: Töö eesmärk oli uurida suure keelemudeli GPT-3.5 Turbo võimekust terviseandmetes nimeolemite märgendamiseks. Terviseandmed sisaldavad vabatekstina olulist teavet patsientide kohta. Selleks et seda teavet oleks võimalik kasutada statistilistes analüüsides, tuleb tekstidest oluline info eraldada, näiteks nimeolemeid märgendades. Masinõppel põhinevad lahendused vajavad nimeolemite märgendamise ülesandega toimetulekuks suurt märgendatud andmestikku, kuid keelemudel GPT-3.5 Turbo on võimeline kohanema erinevatele loomuliku keele töötluse ülesannetele väheste näidete põhjal. Seega võib keelemudeli üldistamisvõime tulla kasuks ka nimeolemite märgendamiseks. Töö raames peenhäälestati erinevate suurustega andmehulkadel mudeleid, et näha, kuidas peenhäälestamine mudeli märgendamise tulemusi mõjutab. Tulemused näitasid, et peenhäälestamine parandab mudeli võimet terviseandmeid märgendada ning ingliskeelsetel tekstidel peenhäälestatud mudelid saavad ka eestikeelsete epikriiside tekstide märgendamises paremini hakkama kui nende baasmudel.

Võtmesõnad:
GPT, keelemudelid, nimeolemite märgendamine, loomuliku keele töötlus, terviseandmed

CERCS: P176 Tehisintellekt

Fine-tuning GPT-3.5 for Medical Named Entity Recognition

Abstract: The aim of this thesis was to explore how well can GPT-3.5 Turbo label named entities. Patient health data contains a lot of useful information in free text form. In order to use this for statistical analyses, structured information has to be extracted from them, for example by annotating named entities. Machine learning based approaches require a lot of annotated data for this, however, a large language model such as GPT-3.5 Turbo has been shown to adapt to different tasks on only a few examples. This general understanding can be leveraged to label named entities. In this thesis, models were fine-tuned with different amounts of data to see how it would benefit labelling. Results showed that fine-tuning does enhance the model's proficiency in recognising entities in health data. Additionally, it is found that the models fine-tuned on English electronic health records outperform their base counterpart at annotating synthetic Estonian electronic health records.

Keywords:
GPT, language models, named entity recognition, natural language processing, health records

CERCS: P176 Artificial Intelligence

Sisukord

Sissejuhatus

Terviseandmed sisaldavad patsientide kohta olulist informatsiooni, mida saab kasutada statistilisteks analüüsideks. Suur osa sellest informatsioonist on aga vabateksti kujul, mida on vaja töödelda, et struktureeritud kujul andmeid saada. Üheks infoeraldusmeetodiks, mida antud töös käsitletakse, on nimeolemite märgendamine. Selle käigus eraldatakse otsitavatele klassidele vastavad lõigud tekstist. Ülesande lahendamiseks kasutatakse laialdaselt masinõppepõhiseid lähenemisi, kuid mudelite treenimine nõuab andmeid, mille pealt ülesannet õppida. Kuna eesti keeles pole treenimiseks ressursse nii palju saadaval kui levinumates keeltes ning terviseandmed on delikaatsed ja nende jagamine on raskendatud, siis on antud mudelite treenimine keeruline.

Viimasel ajal on keeletehnoloogias üha paremaid tulemusi saavutanud suured keelemudelid, mis on eeltreenitud vabateksti korpustel, et vähendada mudelite loomisel vajadust struktureeritud andmestikule [3]. Mudelid nagu GPT-3.5 Turbo suudavad näidete põhjal ülesannetega kohaneda ning tulevad nimeolemite märgendamises paremini toime kui piiratud kogusel märgendatud andmetel treenitud mudelid [2]. Töö eesmärk on katsetada, kas suurte keelemudelite üldistusvõime on piisav terviseandmetes nimeolemite märgendamiseks juhul, kui mudeli treenimiseks puuduvad piisavad andmed, näiteks kasutades ülesandele kohanemiseks teises keeles andmestikku.

Töö esimeses peatükis kirjeldatakse olemasolevaid lahendusi meditsiinilistest tekstides nimeolemite märgendamiseks. Teises peatükis tutvustatakse suurt keelemudelit GPT-3.5 ning selgitatakse kuidas võib mudel terviseandmete märgendamisega toime tulla. Töö raames peenhäälestatakse keelemudelit GPT-3.5 erinevatel andmehulkadel ning hinnatakse, kas mudeli peenhäälestamise tulemused kanduvad edasi ka teises keeles andmetele ning muudele tervisevaldkonna olemitele.

1 Terviseandmete märgendamine

Epikriiside[1] ehk möödunud haigusjuhtude kokkuvõtete tekstid on väärtuslikud allikad teadustöös kasutatavate meditsiinilistele andmete hankimiseks. Ent suurem osa epikriise on vabateksti kujul, mistõttu on nende kasutamine statistiliseks analüüsiks raskendatud. Selleks, et epikriisides sisalduvat teavet oleks võimalik analüüsida, tuleks see struktureerida. Kuna aga manuaalselt teabe eraldamine on suuremahuliste andmehulkade puhul ebarealistlik, rakendatakse protsessi lihtsustamiseks loomuliku keele töötlust [6].

Üheks viisiks vabateksti struktureerida on nimeolemite märgendamine (NER, ingl *named-entity recognition*). NER käigus märgendatakse ja kategoriseeritakse vabatekstis olemeid. Nimeolemite märgendamine võimaldab tekste olemite põhjal filtreerida ning nende sisu ja esinevaid mustreid statistiliselt analüüsida [6]. NER ülesannet on võimalik lahendada näiteks reeglipõhiselt või masinõpet kasutades. Reeglipõhisel lähenemisel kasutatakse nimeolemite märgendamiseks manuaalselt loodud reegleid, mustreid ja sõnastikke [13].

Masinõppe kasutamine nimeolemite märgendamiseks seineb mudeli struktureeritud andmestikul treenimisel [6]. Mudel õpib ära keelemustrid, mille põhjal tekste märgendada. Nende mudelite tulemus sõltub põhiliselt treeningandmete mahust ja kvaliteedist ning enamasti on heade tulemuste saavutamiseks vaja väga suurt märgendatud andmestikku. Paljudes valdkondades on aga suur puudus struktureeritud andmestikkudest, millega mudeleid treenida [13]. García-Barragán jt [14] võrdlesid peenhäälestatud BERT mudelit näidete põhjal juhendatud GPT-3.5 mudeliga hispaaniakeelsete meditsiiliste tekstide märgendamiseks. Nad saavutasid mudelil GPT-3.5 peenhäälestatud BERT mudelist paremad tulemused kasutades palju vähem andmeid kui peenhäälestamiseks kulus.

[1] Epikriis. EKI ühendsõnasik 2023. Eesti Keele Instituut, Sõnaveeb 2023. `https://sonaveeb.ee/search/unif/dlall/dsall/epikriis/1`

2 Suur keelemudel GPT-3

Suur keelemudel on keelemudel[2], mis on treenitud suurel hulgal tekstiandmetel, et luua loomuliku keele väljundeid ja sooritada erinevaid loomuliku keele töötlemise ülesandeid. Generative pre-trained transformer 3 [3] (edaspidi GPT-3) on üks tipptasemel keelemudelitest, mis mõistab ja sünteesib loomuliku keelt. Läbi eeltreenimise on GPT-3 omandanud arusaama keelest, õppides ära keelemustrid ja sõnade vahelised seosed. See võimaldab mudelil kontekstipõhiselt ehk näidete abil uutele ülesannetele kohaneda [3].

GPT-3 mudeli baasiks on Transformer [5] arhitektuur, mis põhineb enesetähelepanu mehhanismil. Enesetähelepanu mehhanism teisendab mudelis sõned sõnavektoriteks vastavalt seostele teiste sõnadega. Vektorite põhjal suudab mudel eristada tekstist olulist teavet ning luua keelelisi seoseid [5]. GPT-3 mudelis on kasutusel kaetud enesetähelepanu mehhanism [3], mis nii-öelda peidab ennustamise käigus mudeli eest edasise olemasoleva teksti, et see tulemust ei mõjutaks. Sõnavektoritesse on lisatud ka teave sõne positsioonist tekstis [5].

GPT-3 on autoregressiivne keelemudel. See tähendab, et väljundi genereerimiseks ennustatakse järjest olemasoleva teksti põhjal kõige tõenäolisemat järgnevat sõne. Erinevalt Transformer mudelist koosneb GPT-3 mudel ainult dekodeerivast osast [3], mitte kodeerivast ja dekodeerivast, sest selline ülesehitus on optimaalsem pikemate tekstide genereerimiseks. Sisendi ja väljundi eraldi töötlemise asemel võtab mudel sisendi teksti alguseks ning hakkab sellele teksti juurde genereerima [4].

GPT-3.5 on mudel, mis põhineb GPT-3 arhitektuuril, kuid mida on edasi treenitud inimese poolt antud juhiste ja tagasisidega (ingl *reinforcement learning from human feedback* ehk RLHF) [12]. See võimaldab mudelil paremini kasutaja poolt antud juhiseid

[2]Suur keelemudel. EKI ühendsõnasik 2023. Eesti Keele Instituut, Sõnaveeb 2023. `https://sonaveeb.ee/search/unif/dlall/dsall/suur%20keelemudel/1`

järgida. Antud töös kasutatakse mudelit GPT-3.5-turbo-0613[3], mida saab kasutaja poolt peenhäälestada.

2.1 Peenhäälestamine

Peenhäälestamine (ingl *fine-tuning*) võimaldab ülesande või valdkonna spetsiifilise mudeli loomist ilma, et see tuleks nullist treenida. GPT mudelite peenhäälestamise käigus kohendatakse mudeli parameetreid [1] etteantud sisendi ja väljundi paaride põhjal. GPT-3 on näidete põhjal õppiv mudel, seega soovitatakse OpenAI dokumentatsioonis[4] ka GPT-3.5-turbo põhiliseks optimeerimise viisiks mudeli viiba ehk juhise täpsustamist (ingl *prompt engineering*). Dokumentatsiooni järgi võib peenhäälestamisest kasu olla juhtudel nagu vastuste stiili ja vormi täpsustamine või viibas raskesti selgitatava ülesande täitmine.

2.2 GPT märgendamiseks

Töö teises peatükis kirjeldati olemasolevaid lahendusi andmete märgendamiseks. Üheks neist oli masinõpe, mille miinuseks toodi välja vajadus suurele ülesandele spetsiifilisele märgendatud andmestikule. Sellistest andmestikudest on tihtilugu puudus, eriti vähem levinud keeltes nagu eesti keel. GPT-3 vajab tänu eeltreenimisest tulenevale üldistusvõimele ülesannetega toimetulekuks vähem näiteid kui tavaline peenhäälestamine [3]. Seega võib mudel olla kasulik olukordades kus puudub piisav kogus ülesande spetsiifilisi treeningandmeid. Kuigi GPT-3 on mõeldud teksti genereerimiseks tuleb see toime ka loomuliku keele töötluse ülesannetega [3].

Kuna GPT-3 on tekstirobot, ei ole seda võimalik rakendada täpselt samal viisil nagu

[3]Azure OpenAI Service models. https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models#gpt-35 (15.05.2024)

[4]Fine-tuning. https://platform.openai.com/docs/guides/fine-tuning?ref=mlq.ai (05.05.2024)

märgendamiseks mõeldud mudeleid, mis märgendavad kõik sõned tekstis. Mudel tagastab märgendid tekstilisel kujul, seega sõltub ülesande lahendus lisaks mudeli vastusest ka selle tekstiga sidumise viisist. Tuleb tagada, et märgendusi kaduma ei läheks ning need saaksid kõik õige olemi alla kategoriseeritud. Wang jt [2] lasid näiteks mudelil tagastada kogu sisendteksti, milles olid olemid sümbolitega markeeritud. Sel viisil pidi aga iga olemi puhul mudelit eraldi viipama. Lisaks andsid nad igas viibas ette ka tekstile vastavad näiteid lahendustest. Pikemate tekstide korral ei pruugi aga kogu tekst koos näidete ning märgendus sümbolitega konteksti limiiti mahtuda. Peenhäälestamine võimaldab anda mudelile korraga rohkem näiteid ette ilma, et neid tuleks viipa lisada. Käesolevas töös pakuti välja lahendus, kus mudel tagastab märgendid JSON-kujul ilma originaalse tekstita, kus igale nimeolemile vastab järjend märgendustest. See lähenemine võimaldab kõik olemid ühe viipamisega märgendada ning pikemaid tekste viipa mahutada.

Lisaks tuleb arvetada ka mudelile seatud piirangutega. Põhiliseks neist on konteksti limiit, mis määrab vestlusesse mahtuvate sõnede arvu. Limiidi suurus oleneb mudelist, töös kasutataval mudelil GPT-3.5-turbo-0613 on selleks 4096 sõne. Mudelile esitatavale viipadele on seatud ka filter[5], mis piirab teatud graafiliste väljendite kasutamist. Valdkonnas nagu meditsiin võib see tulla probleemiks, sest tekstid võivad sisaldada graafilist sisu.

[5]Content filtering. `https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/content-filter?tabs=warning%2Cpython-new` (13.04.2024)

3 Metoodika

Keelemudeli terviseandmete märgendamise võimekuse ning ülesandele peenhäälestamise testimiseks peenhäälestati 3 mudelit. Mudelite tulemusi hinnati baasmudeliga ning omavahel. Lisaks hinnati ka peenhäälestamise tulemuse laienemist sama valdkonna teises keeles andmestikule.

Töö praktiline osa koosneb neljast etapist: andmete eeltöötlus, mudelite peenhäälestamine, mudelite viipamine ja tulemuste hindamine. Enamus nendest sammudest viidi läbi rakenduses Jupyter Notebook [10] ning kirjutati programmeerimiskeeles Python. Kogu töö kood on lisatud GitHubi repositooriumisse [6], millest leiab ka fail kõiki vajalikke teeke sisaldava keskkonna ülesseadmiseks. Töö valmis Tartu Ülikooli teadusarvutuste keskuses [11].

3.1 Kasutatud andmestikud

Töös kasutati kahte andmestikku - ühte mudeli peenhäälestamiseks ning teist testimiseks. Mõlemad andmestikud koosnevad märgendatud ja anonümiseeritud epikriisidest, aga nende sisu erineb üksteisest keele, päritolu ja märgendatud olemite poolest.

Töö kirjutamise hetkel ei olnud GPT-3.5 mudelit võimalik lokaalselt jooksutada ning see oli kasutatav vaid läbi OpenAi API[7]. Seetõttu oli piiratud mudeli viipamine sensitiivsete andmetega, mis vähendas ka töös kasutatavate andmestikude valikut.

Esimeseks andmestikuks oli National NLP Clinical Challenges 2018 ADE and Medication Extraction Challenge (edaspidi n2c2) andmestik [8]. See koosneb 505 ingliskeelsest epikriisist ning selles on märgendatud olemid ravimite kasutuse kohta. Nendeks olemiteks on ravim, tugevus, doos, kestus, sagedus, vorm, viis, põhjus ja kõrvaltoime. Epikriisid on märgendatud kahe sõltumatu isiku poolt ning märgendite konflikte lahendas

[6] Lõputöö GitHubi repositoorium. `https://github.com/annammariaa/FT-GPT-NER`

[7] Introduction. `https://platform.openai.com/docs/introduction` (15.05.2024)

kolmas isik. Andmestik ei ole avalikult kättesaadav, aga sellele on võimalik taodelda töös kasutamiseks ligipääsu eeldusel, et töö käigus andmeid ei levitata. Seda andmestikku kasutati mudelite peenhäälestamiseks ja testimiseks.

Teise andmestiku moodustasid sünteetilised ehk keelemudeli poolt genereeritud epikriisi tekstid. Andmestik on loodud kasutades mudelil GPT-2 põhinevat keelemudelit [9]. Mudelit suunati genereerima tekste, mis sisaldavad viit kõige sagedamini esinenud diagnoosi RITA MAITT [7] andmestikus. Sünteetiline andmestik koosneb 500 eestikeelsest tekstist, milles on märgendatud olemid haigus, ravim, suitsetamine ja protseduur. Tekstid olid märgendatud käsitsi meditsiinilise ekspertiisiga inimese poolt. Seda andmestikku kasutati ainult mudeli testimiseks.

3.2 Andmete eeltöötlus

Andmestikud töödeldi OpenAI peenhäälestamise juhendis[8] nõutud kujule. Peenhäälestamiseks loodi kaks eraldi JSONL-faili: treeninghulga fail ja valideerimishulga fail. Failid pidid sisaldama igal real ühte json-vormingus näidet mudeliga vestlustest. Nende näidete juures tuli arvestada mudeli konteksti limiidiga, milleks GPT-3.5-turbo-0613 puhul on 4096 sõne. Sellesse limiiti pidi mahtuma kogu näite sisu ehk viibad koos mudelilt oodatud vastusega.

Enne näidete kokkupanekut tuli luua süsteemi ja mudeli viip. Viibad hoiti võimalikult lihtsad ja lühikesed, et need tulemust võimalikult vähe mõjutaksid. Loodud viibad olid ingliskeelsed ning neid kasutati nii mudeli peenhäälestamisel kui ka hiljem mudelit testides. Näite vorming koos viipadega on toodud lisas 1.

n2c2 andmetest eraldati 15% ehk 76 teksti testhulka, ülejäänud 429 teksti moodustasid treeninghulga. Suur osa n2c2 andmestiku tekste olid liiga pikad, et koos viipadega mudeli konteksti limiiti mahtuda. Seetõttu tuli neid lühendada ning andmestiku võimalikult

[8] Fine-tuning. https://platform.openai.com/docs/guides/fine-tuning (15.05.2024)

suurena hoidmise eesmärgil otsustati jagada tekstid mitme näite peale laiali. Esmalt aga lihtsustati andmete lühendamiseks tekstisisesed anonümiseerimised, näiteks kuupäevade märgise „[**yyyy-mm-dd**]” asenduseks oli „*DATE*”. Edasi jagati üle limiidi olevad näited väiksemateks osadeks. Limiiti mahtumist kontrolliti sõnestades kogu näide kasutades OpenAI sõnestajat Tiktoken[9]. Tulemuseks oli 704 näitest koosnev treeningandmestik ja 125 näitest koosnev testandmestik.

Testimisel täpsemaks tulemuste hindamiseks loodi eraldi fail, millest oleks võimalik indeksite põhjal märgendite asukohad tekstis leida. Fail sisaldas testhulga tekste, märgendeid ja märgendite positsioone teksts, mis olid kohandatud vastavalt tekstide jagamisele. Erinevalt teisest testandmestikust ei olnud selles failis tekstides anonümiseerimised lühendatud, et n2c2 andmestikus antud märgendite indeksid tekstile vastaks.

Sünteetiliste eestikeelsete andmete töötlus seisnes neist genereerimist suunava teabe eemaldamisel. Neid tekste ei lühendatud, sest kõik tekstid jäid konteksti limiidi piiri. Tekste töödeldes leiti ka märgendite positsioonid indeksite paaride kujul. Erinevalt n2c2 andmestikust salvestati sünteetiliste andmete positsioonid kogu muu informatsiooniga samasse faili, sest need vastasid töödeldud tekstidele. Mudelite viipamisel neid viipa, et lisatud.

Vormistatud näited kirjutati JSONL-failidesse. Töödeldud n2c2 treeninghulgast loodi peenhäälestamisteks 3 treeningandmestikku ja 3 valideerimisandmestikku. Andmestikke luues suurendati järjest kaasavate näidete arvu ehk andmestikkude vahel oli mõningane kattuvus. Treeningandmestikud koosnesid 150, 300 ja 704 näitest, lisaks eraldati igast treeningandmestikust 20% vastavasse valideerimise andmestikku.

[9]Tiktoken. https://github.com/openai/tiktoken (15.05.2024)

3.3 Keelemudeli peenhäälestamine

Keelemudelid peenhäälestati Azure OpenAI stuudios[10], kus on võimalik kasutada ja peenhäälestada OpenAI mudeleid. Peenhäälestati 3 mudelit, mille baasiks oli OpenAI mudel GPT-3.5-turbo-0613 (edaspidi peenhäälestamata GPT) ning peenhäälestamiseks kasutati töödeldud n2c2 andmestikust loodud treening- ja valideerimisandmestikke.

Tabel 1. Mudelite peenhäälestamiseks kasutatud andmestikkude suurused ja peenhäälestamise maksumus.

Mudel	Näiteid treeninghulgas	Näiteid valideerimishulgas	Maksumus
GPT-150	120	30	21.07€
GPT-300	240	60	21.07€
GPT-700	564	140	42.13€

Mudelite määrati töö raames nimetused GPT-150, GPT-300 ja GPT-700. Tabelis 1 on toodud mudelitele vastavate treeningandmestikude ja valideerimiseandmestkude suurused. Mudelile GPT-300 vastasid kõige vähesemate näidetega andmestikud ning mudelile GPT-700 vastasid kõige rohkemate näidetega andmestikud. Tabelis 1 on toodud lisaks välja ka mudelite peenhäälestamiste maksumused.

3.4 Keelemudeli viipamine

Mudelite viipamiseks kasutati OpenAI Pythoni teeki[11], mis võimaldab API kaudu mudelit kasutada. Mudelite testimisel kasutati mõlemal testhulgal samu viipu, mida ka mudeli peenhäälestamiseks. Erandlikult tuli peenhäälestamata mudelit sünteetiliste eestikeelsete andmetega viibates ette anda ka soovitud vastuse kuju täpsustus, sest ilma selleta ei

[10] Azure OpenAI Studio. https://oai.azure.com/portal (13.05.2024)

[11] OpenAI API. `https://platform.openai.com/docs/api-reference/introduction?lang=python` (15.05.2024)

tagastanud mudel enamus kordadel märgendeid JSON-kujul. Selleks lisati mudeli viiba lõppu tühjalt JSON-kuju. Mudeli vastused salvestati koos originaalsete märgenditega uude JSONL-faili.

Enne faili lisamist kontrolliti, kas vastus on JSON-vormingus. Vales vormingus vastuseid faili ei lisatud, kuid mudelit viibati ebaõnnestumise juhul veel ühe korra sama tekstiga uuesti. Enamustel kordadel tagastasid mudelid teisel korral vastuse õigel kujul. Mudelit viibati täiesti uuesti, mitte ei jätkatud eelmist vestlust, sest see ei oleks mudeli konteksti limiiti ära mahtunud. Lisaks tuli parandada vastustes ära ka jutumärgid, sest mudelid tagastasid tigtilugu json-vormi ühekordsete jutumärkidega, aga JSON-vorm nõuab kahekordseid jutumärke. Ühekordsed jutumärgid asendati paaride kaupa kasutades regurlaaravaldist.

3.5 Tulemuste hindamine

Tulemuste hindamiseks arvutati Seqeval teeki[12] kasutades mudelite täpsus, saagis ja F1-skoor. Teegi funktsioonid võtavad argumentideks tõeliste ja ennustatud märgendite järjendid ning arvutavad nimetatud tulemused nii olemi põhiselt kui ka olemite peale kokku. Täpsus (valem 1) näitab kui suur osa kõigist ennustatud märgenditest osutusid õigeks. Täpsus on 0 kui kõik ennustatud märgendid olid valed ning 1 kui kõik ennustatud märgendid olid õiged. Saagis (valem 2) näitab kui suur osa päris märgendustest ennustatud said. Saagis on väärtuselt 0 kui mitte ühtegi päris märgendit ei ennustatud ja 1 kui kõik päris märgendused said ennustatud. F1-skoor (valem 3) on täpsuse ja saagise harmooniline kesmine. Valemitel tähistab TP tõeliste posiivsete ehk õigesti ennustatud märgendite arvu, FP valepositiivsete ehk valesti ennustatud märgendite arvu ning FN ehk valenegatiivne tähistab ennustamata jäänud märgendite arvu.

[12] seqeval. `https://github.com/chakki-works/seqeval` (15.05.2024)

$$Täpsus = \frac{TP}{TP + FP} \tag{1}$$

$$Saagis = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - skoor = \frac{2 \cdot Täpsus \cdot Saagis}{Täpsus + Saagis} \tag{3}$$

Hindamise võimaldamiseks vormistati märgendid tekstiliselt kujult ümber järjenditeks. Järjendite loomiseks sõnestati esmalt kogu algne tekst kasutades NLTK teegi sõnestajat TreebankWordTokenizer[13], et saada sõned koos nende positsioonidega tekstis. Edasi teisendati need sõnastiku kujule, kus võtmeteks olid sõnede positsioonid ning väärtusteks vastavad sõned. See kuju võimaldas hiljem lihtsamini järjendid tõeliste märgenditega siduda.

Tõeliste märgendite järjendi kujule viimiseks kasutati andmete eeltöötlusel salvestatud alg- ja lõppindeksite paare. Iga paar vastas ühele märgendile. Märgendused tuli sõnestada, sest nad võisid koosneda mitmest sõnast. Need andmed teisendati sõnastiku kujule sarnaselt teksti sõnedele, kusjuures väärtusteks olid sõnede asemel nendele vastavad nimeolemid.

Ennustatud märgendid viidi samale sõnastiku kujule, mis tõelised. Nende positsioonid tuli aga tekstist eraldi leida. Selleks loodi igast märgendist regulaaravaldis, millega leiti kõigi avaldisele vastavate sõnede positsioonid tekstis. Esmalt aga väiketähestati nii tekst kui ka märgendid, sest puudus veendumus mudelite väike ja suure algustähe täpsustes. Märgendused sõnestati ning lisati koos sõnedele vastavate positsioonidega sõnastikku.

[13] TreebankWordTokenizer. `https://www.nltk.org/api/nltk.tokenize.TreebankWordTokenizer.html`

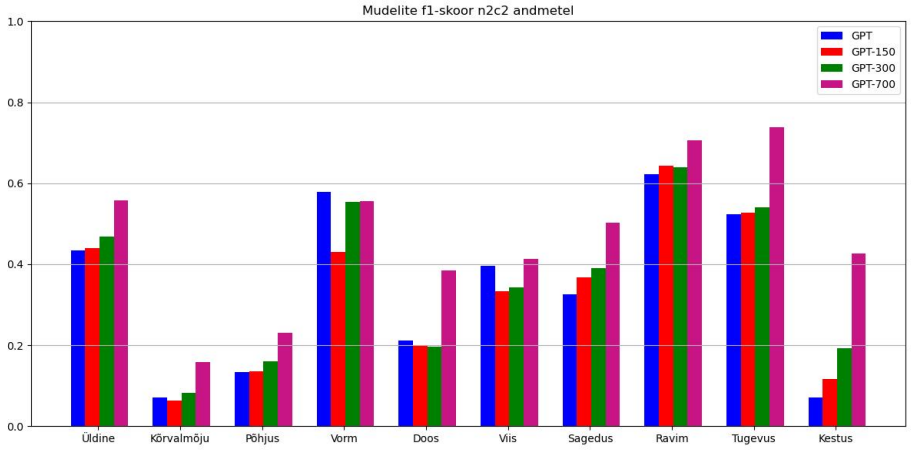
Oluline on märkida, et ennustatud märgendite asukohtade leidmisel võeti arvesse kõik tekstist leitud sõnad, mis mustrile vastasid olenemata sellest mitu korda mudel neid tagastanud oli. Märgenduste tagastamise lähenemine ei võimalda automaatselt tuvastada, milliseid kohti mudelid tekstis tegelikult märgendasid. Seetõttu jäeti ennustustest välja ühe tähe pikkuselised märgendused kuna need põhjustaks liigselt valepositiivseid märgendusi. Kui mudel oli tagastanud märgendid valel kujul, näiteks iga märgendi eraldi sõnestikuna, kust oli olemeid raske automaatselt tuvastada, siis ühtegi märgendit arvesse ei võetud.

Loodud sõnastikkude abil teisendati märgendid järjendite kujule. Iga teksti sõnastikus oleva sõne puhul vaadati, kas selle positsioonile leidub märgendite hulgas vaste. Kuigi sõnestamiseks kasutati alati sama sõnestajat, võis märgendite indeksite ümber arvutamisel tekkida väikseid erinevusi. Selleks, et need siiski vastava sõnega kokku viia, veenduti positsioonide täpses vastavuses ning lisaks ka selles, et märgendi alguse ja lõpu indeksid jääks sõne positsiooni indeksite vahemikku. Kui vaste leiti lisati järjendis sõnele vastavale kohale märgendatud olemi nimetus. Kui vaste puudus, ei kuulunud sõne ühegi olemi alla ning sellele lisati märgendiks „O”. Selliselt loodi järjendid nii tõelistest kui ka iga mudeli ennustatud märgenditest ning nende põhjal arvutati mudelite täpsused, saagised ja F1-skoorid.

4 Tulemuste valideerimine

Tulemusi hinnati baasmudelil GPT-3.5-turbo-0613 ja igal peengäälestatud mudelil. Kõigepealt hinnati tulemusi n2c2 andmestikul, et näha kas peenhäälestamine parandab mudeli tulemusi tervise andmete märgendamisel ning kuidas mõjutab seda kasutatud andmestiku suurendamine. Järgmisena hinnati peenhäälestatud mudelite tulemusi sünteetilistel eestikeelsetel andmetel, mida võrreldi samuti baasmudeli tulemusega, et näha kui palju märgendamise ülesandele peenhäälestamine ka teistes keeltes tekstide märgendamist mõjutab.

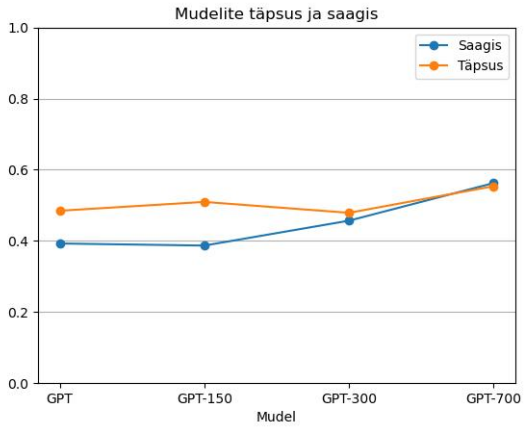
4.1 Tulemused inglise keelsel andmestikul



Joonis 1. Mudelite F1-skoorid n2c2 andmestikul olemite kaupa.

Joonisel 1 on toodud mudelite F1-skoorid n2c2 testhulgal olemite kaupa, joonisele on lisatud ka F1-skoor kõigi olemite peale kokku sildiga „Üldine". Mudelite F1-skooride põhjal paranes üldine tulemus iga peenhäälestamiseks kasutatud andmestiku suurendamisega. Võrreldes omavahel mudelite täpsust ja saagist (joonis 2) on näha, et üldise tulemuse paranemine tuleneb põhiliselt saagise tõusust, sest täpsus muutus peenhäälestamistega vähe. See tähendab, et peenhäälestatud mudelid said suurema osa tõelistest märgenditest

kätte. Näiteks on peenhäälestamata GPT mudelil ravimi olemil kõige kõrgem täpsus ning peenhäälestamistega täpsus langeb (tabel 2), aga see-eest samal olemil saagis tõuseb peaaegu sama palju kui täpsus langes.



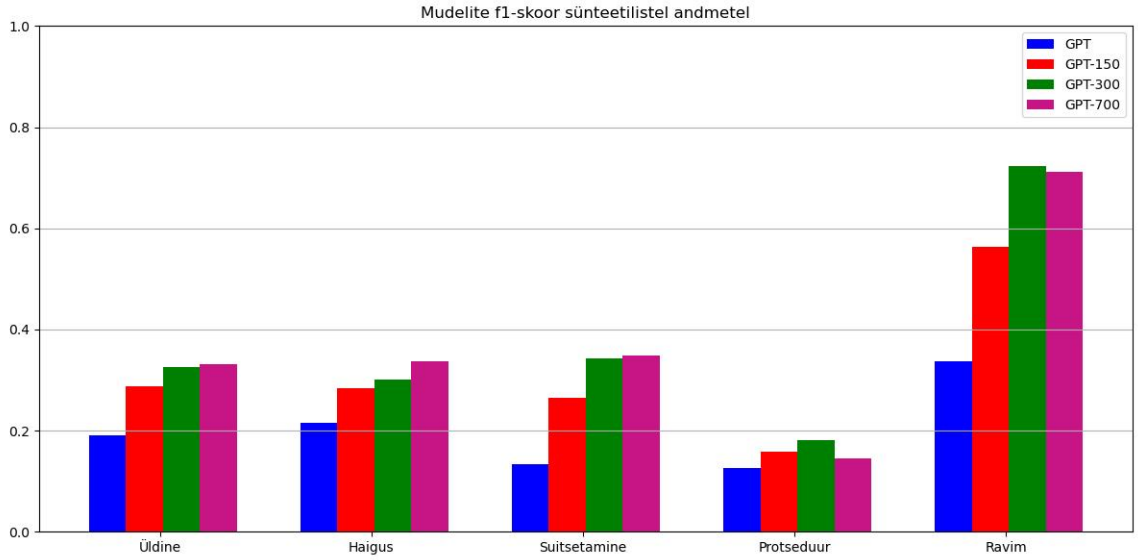
Joonis 2. Mudelite täpsuste ja saagiste võrdlus n2c2 andmestikul.

Mudelil GPT-700 tõusid nii saagis kui ka täpsus võrreldes kõigi väiksematel andmehulkadel peenhäälestatud mudelitega (joonis 2). Kõige rohkem mõjutas peenhäälestamine just neid olemeid, millel alguses väga madal tulemus oli. Nendeks olemiteks olid kõrvalmõju, põhjus, doos ja kestus (joonis 1). Lisas 2 on toodud joonised mudelite saagistest ja täpsustest olemite kaupa.

Tabel 2. Mudelite tulemused n2c2 andmestiku olemil ravim.

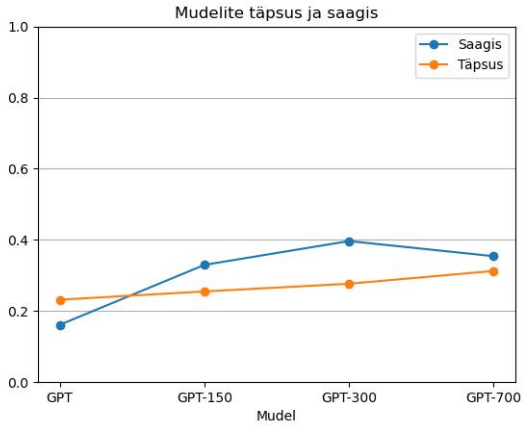
Mudel	Täpsus	Saagis	F1-skoor
GPT	**0.907**	0.506	0.649
GPT-150	0.763	0.557	0.644
GPT-300	0.652	0.629	0.640
GPT-700	0.750	**0.666**	**0.706**

4.2 Tulemused sünteetilisel eesti keelsel andmestikul



Joonis 3. Mudelite F1-skoorid sünteetilisel eestikeelsel andmestikul olemite kaupa.

Joonisel 3 on toodud mudelite F1-skoorid sünteetilstel eestikeelsetel andmetel. F1-skooride järgi tõusid tulemused mudelite peenhäälestamisega ka nende tekstide märgendamisel. Kõige rohkem paranesid tulemused olemil ravim, mille saagis peaaegu kolmekordistus peenhäälestamistega (lisa 3 joonis 8).



Joonis 4. Mudelite täpsused ja saagised sünteetilisel eestikeelsel andmestikul.

Sünteetilistel eestikeelsetel tekstidel paranes saagis rohkem kui täpsus (joonis 4). Nende tõus oli nii saagisel kui ka täpsusel enamjaolt ühtlane. Küll aga Võrreldes langes saagis lõpuks mudelil GPT-700 kuid tulemus oli ikka kõrgem kui peenhäälestamata mudelil.

4.3 Diskussion

Peenhäälestamisega paranes mudeli GPT-3.5 terviseandmete märgendamise tulemus. Eriti oli peenhäälestamisest kasu nendel olemitel, millel peenhäälestamat GPT väga madalad tulemused sai (joonis 1). Põhiliselt tõusis mudeli saagis, mis on parem kui täpsuse tõus, sest tervise valdkonnas on olemi märgendamata jätmisel palju suurem kaal kui valesti märgendamisel. Lisaks on tekstide kontrollimine ja valesti märgendatud tekstide parandamine lihtsam kui manuaalne olemite tekstist otsimine.

Peenhäälestamise tulemused laienesid uuele andmestikule, eriti saagise tõusu näol. Kõige rohkem oli peenhäälestamisest kasu olemil ravim, mis oli ainsana ka peenhäälestamise andmestikus märgendatud. Seega õpib mudel ingliskeelsetel näidetel olulist konteksti, mis aitab ka eesti keelel olemeid leida.

Mudelite peenhäälestusega ei jõutud punkti, kus andmestiku suurendamine tulemust enam ei parandaks. Tõenäoliselt tõstaks andmete hulga suurendamine märgendamise tulemusi veelgi, aga selleks puudusid vajalikud andmed. Mudelite peenhäälestamine on lisaks ka tasuline teenus. Üha enam andmehulkade suurendamisel tõusevad häälestamiseks kuluvad summad liiga kõrgeks, et mudeleid töö raames rohkem edasi luua.

Kuigi sünteetilistel eestikeelsetel andmetel olid mudeli GPT-700 tulemused veidi nõrgemad kui mudelil GPT-300 (joonis 3), siis edasi peenhäälestamiseks kasutatava andmehulga suurendamisel tulemus ilmselt paraneks, mitte ei langeks veelgi. Keelemudeli puhul tuleb arvestada, et tekstidel võivad tulemused märkimisväärselt sama viibaga mitu korda viibates omavahel erineda. Töös viibati testimisel mudelit sama tekstiga uuesti juhul kui eelnev vastus ei sisaldanud korrektset JSON-vormi. Enamusel kordadest tagastasid mudelid teisel katsel korrektses vormis vastuse. See näitab kui erinevad võivad ühel mudelil vastused tulla, isegi kui viip on täpselt sama. Tulemuste langus mudelil GPT-700 võiski tuleneda juhuslikult halbadest vastustest, mis uuesti katsetamisel võivad erineda.

Lisaks mudelite märgendamise tulemustele paranes ka vastuste vormi täpsus. Peenhäälestatud mudelite poolt tagastatud täiuslik JSON-kuju oli iga kord õige ning ei sisaldanud soovimatut lisainformatsiooni. Küllaga esines kõigil mudelitel vigu JSON-vormingu kokkupanekul. Näiteks võisid olla sulud või jutumärgid paigast ära, mistõttu ei olnud vastust võimalik kasutada. Võis ka juhtuda, et mudel siiski genereeris lisa infot märgendite kohta, aga JSON-vorm jäi lõpetamata, sest vastus ei mahtunud konteksti limiiti ning genereerimine jäeti pooleli.

Kuigi enamasti tagastas peenhäälestamata GPT ennustusi täiuslikul JSON-kujul, siis mitmel korral ei vastanud mudeli tagastus oodatud JSON-vormile. Näiteks tagastas mudel iga märgendatud olemi kohta eraldi sõnastiku. Ka need sõnastikud erinesid üksteisest viipamiset, misõttu ei olnud neist võimalik märgendeid koos vastava nimiolemiga automaatselt välja lugeda.

Kokkuvõte

Töö eesmärk oli katsetada terviseandmetes nimeolemite märgendamist suure keelemudeliga GPT-3.5 ning uurida, kuidas mõjutab peenhäälestamine mudeli tulemusi. Olemasolevad masinõppel põhinevad andmete nimeolemite märgendajad vajavad heade tulemuste saavutamiseks suurt ülesandele vastavat märgendatud andmestikku. Selliseid andmeid on mudelite treenimiseks vähe kättesaadaval, eriti vähem levinud keeltest nagu eesti keel. Kuigi GPT-3.5 on loodud loomuliku teksti genereerimiseks, mitte nimeolemite märgendamiseks, siis on mudel saavutanud läbi eeltreenimise üldistusvõime, mis võimaldab mudelil saavutada häid tulemusi erinevates keeletehnoloogia ülesannetes.

Töö raames peenhäälestati GPT-3.5 Turbo baasil kolm mudelit, mis erinesid üksteisest kasutatud andmehulga poolest. Peenhäälestamise andmetena kasutati märgendatud ingliskeelseid epikriisi tekste. Olenemata sellest, et andmete puuduse tõttu ei saavutatud punkti, milles peenhäälestamiseks kasutatud andmehulga suurendamine mudeli tulemusi enam ei parandanud, siis näidati, et peenhäälestamine parandab mudeli GPT-3.5 Turbo võimet terviseandmeid märgendada. Peenhäälestamine muutis mudelid täpsemaks ning tulemused paranesid põhiliselt saagise näol, mis tähendab, et mudelid leidsid tekstist üha rohkem nimeolemeid üles. Inglise keelsetel epikriisi tekstidel peenhäälestatud mudelid tulid peenhäälestamata mudelist paremini toime ka eestikeelsete epikriisi tekstide märgendamisega. Tulemused laienesid vähesel määral isegi nimeolemitele, mida peenhäälestamiseks kasutatud andmestikus märgendatud ei olnud.

Viidatud kirjandus

[1] Radford, A., Narasimhan, K., Salimans, T. Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (08.05.2024)

[2] Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J. Wang, G. GPT-NER: Named Entity Recognition via Large Language Models. 2023. https://doi.org/10.48550/arXiv.2304.10428

[3] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C. ... Amodeo, D. Language Models are Few-Shot Learners. 2020. https://doi.org/10.48550/arXiv.2005.14165

[4] Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N. Generating Wikipedia by Summarizing Long Sequences. 2018. https://doi.org/10.48550/arXiv.1801.10198

[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention Is All You Need. 2023. https://doi.org/10.48550/arXiv.1706.03762

[6] Kundeti, S. R., Vijayananda, J., Mujjiga, S., Kalyan, M. Clinical named entity recognition: Challenges and opportunities. 2016 IEEE International Conference on Big Data. IEEE, 2016, 1937-1945. https://doi.org/10.1109/BigData.2016.7840814

[7] Oja, M., Tamm, S., Mooses, K., Pajusalu, M., Talvik, H.-A., Ott, A., Laht, M., Malk, M., Lõo, M., Holm, J., Haug, M., Šuvalov, H., Särg, D., Vilo, J., Laur, S., Kolde,

R., Reisberg, S. Transforming Estonian health data to the Observational Medical Outcomes Partnership. 2023. https://doi.org/10.1093/jamiaopen/ooad100

[8] Henry, S., Buchan, K., Filannino, M., Stubbs, A., Uzuner, O. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. 2019. https://doi.org/10.1093/jamia/ocz166

[9] Lepson, M. Epikriisi tekstide genereerimine GPT-2 mudeliga. 2023. https://dspace.ut.ee/items/517f9375-7c0d-4094-865b-de91d7782840 (15.05.2024)

[10] Project Jupyter. Jupyter Notebook. https://jupyter.org/ (15.05.2024)

[11] Tartu Ülikool. UT Rocket. https://doi.org/10.23673/PH6N-0144

[12] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R. Training language models to follow instructions with human feedback. 2022. https://doi.org/10.48550/arXiv.2203.02155

[13] Anandika, A., Mishra, S. P. A study on machine learning approaches for named entity recognition. 2019. https://doi.org/10.1109/ICAML48257.2019.00037

[14] García-Barragán, Á., Calatayud, A. G., Solarte-Pabóna, O., Provencio, M., Menasalvas, E., Robles, V. GPT for medical entity recognition in Spanish. 2024. https://doi.org/10.1007/s11042-024-19209-5

[15] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. https://doi.org/10.48550/arXiv.1810.04805

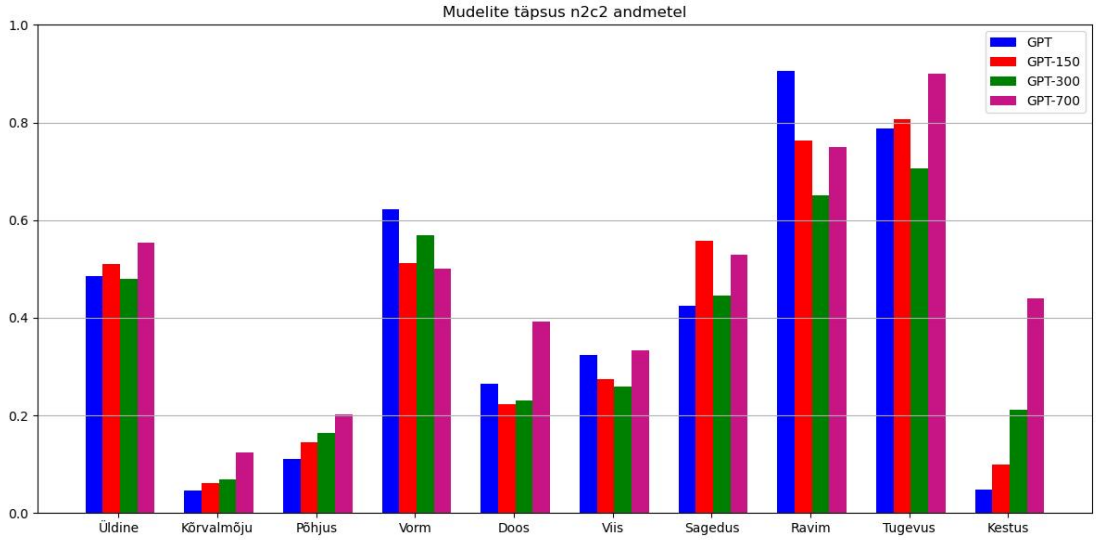
Lisad

I. JSONL-faili näidis koos viipadega

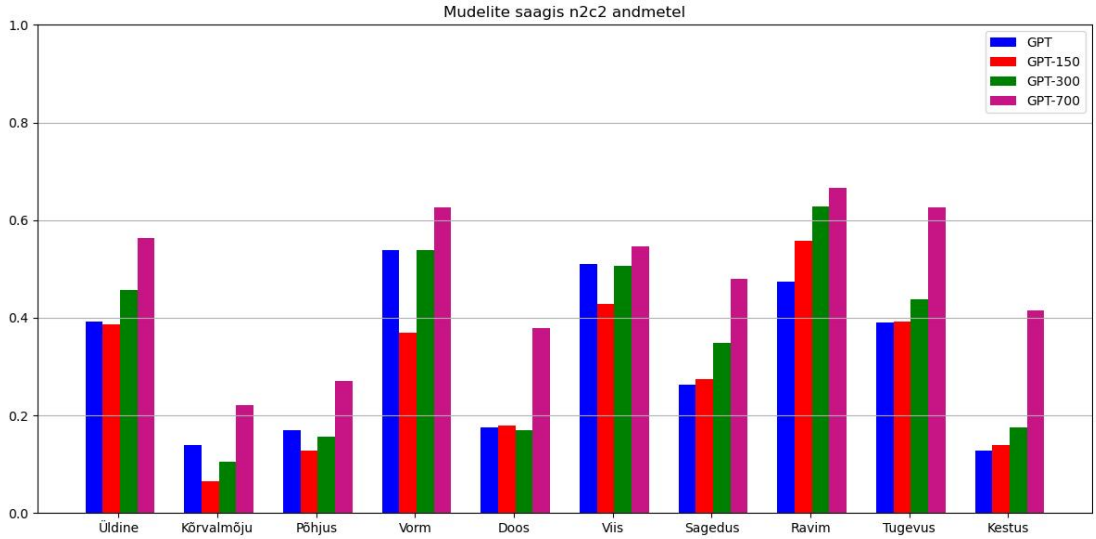
Ühe JSONL-faili rea näidis, millel on näha süsteemiviip ja mudeli viip. Muutuja $TAGS vastab nimeolemitele, mida sooviti tekstis märgendada ja muutuja $TEXT märgendatavale tekstile.

```
{"messages" : [
    {
        "role": "system"
        "content" : "This model extracts entities from text,
        returning JSON-formatted output for tags $TAGS."
    },
    {
        "role": "user"
        "content" : "Extract entities $TAGS from the following
        text and return the output in JSON format. $TEXT."
    }
]}
```

II. Mudelite saagis ja täpsus n2c2 testhulgal

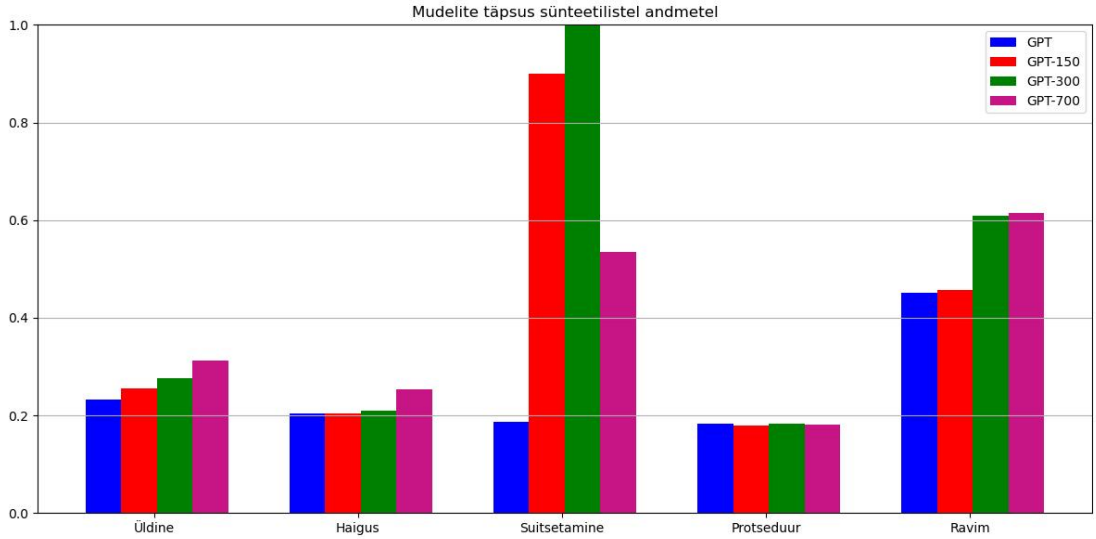


Joonis 5. Mudelite täpsused n2c2 andmestikul olemite kaupa.

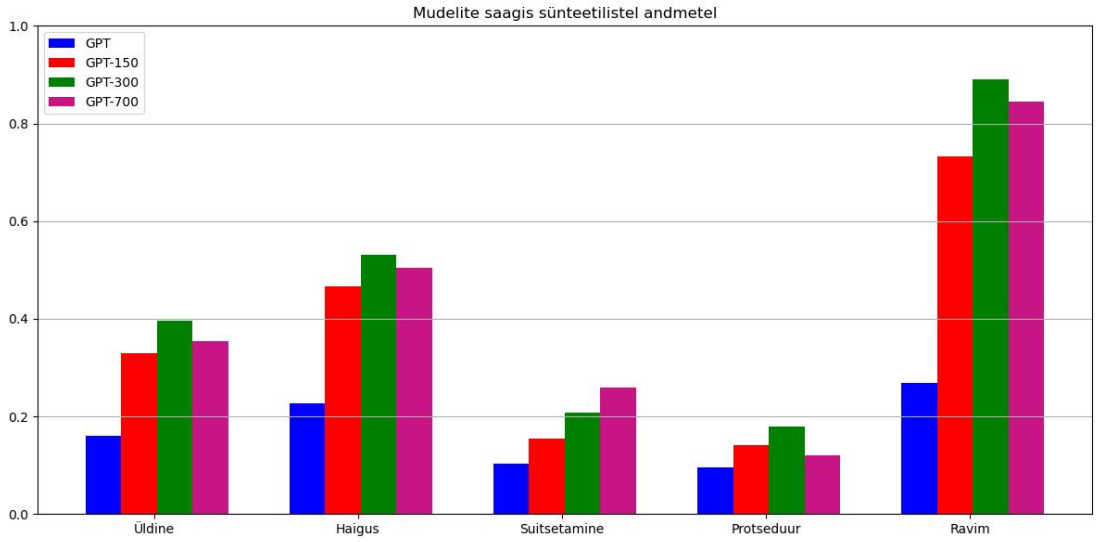


Joonis 6. Mudelite saagised n2c2 andmestikul olemite kaupa.

III. Mudelite saagis ja täpsus sünteetilistel eestikeelsetel andmetel



Joonis 7. Mudelite täpsused sünteetilisel eesti keeles andmestikul olemite kaupa.



Joonis 8. Mudelite saagised sünteetilisel eesti keeles andmestikul olemite kaupa.

IV. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Anna Maria Tammin**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

   GPT-3.5 peenhäälestamine terviseandmete märgendamiseks,

   mille juhendaja on Hendrik Šuvalov,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Anna Maria Tammin

15.05.2024