



VILNIAUS UNIVERSITETAS
MATEMATIKOS FAKULTETAS
GRETUTINĖS MATEMATIKOS STUDIJOS

Mechanistinio interpretavimo tyrimas Mamba architektūros didžiajame kalbos modelyje (LLM)

Gretutinių studijų baigiamasis darbas

Atliko: **Danielius Kundrotas**

VU el. p.: danieliuskundrotas@ff.stud.vu.lt

Vadovas: doc. Vytautas Čyras

Vilnius

2024

Turinys

1	Kodėl dirbtinis intelektas kelia pavojų?	6
1.1	Suderinamumo problema	6
1.2	Nubégimo scenarijus	7
1.3	Léktuvas – geležinis paukštis	7
2	Didieji kalbos modeliai	8
3	Mechanistinis interpretavimas	10
4	Neuroninio tinklo architektūrų palyginimas	12
5	Apibrėžimai	16
5.1	Teksto vienetai	16
5.2	Pirmasis sluoksnis – įdėtis	18
5.3	Išvesties generavimo tvarka	19
5.4	Išvestys L, T, P	19
6	Išvesčių lyginimas	20
6.1	Vaizdavimas	20
6.2	<i>K</i> skirstiniai	21
6.3	Natūralios kalbos saknio ir monotoninės įvesties palyginimas	25
6.3.1	„Mary“ ir „123“ įvestys	25
6.3.2	Metodas	25
6.3.3	Rezultatai	25
6.4	Periodinės įvestys	28
6.4.1	Metodas	28
6.4.2	Rezultatai	28
7	Triukšmo įtaka	32
7.1	Metodas	32
7.2	Rezultatai	32
7.3	Eksperimento kritika	32
7.4	Tyrimo pratesimas	34
7.5	Matricinė išvestis į vektorių	34
8	Išvados ir rezultatai	36

Santrauka

Paskutiniu metu didieji kalbos modeliai naudojami vis plačiau. Beveik visi naujausi dirbtiniame intelekto proveržiai kilo dėl transformatorių architektūros. Prieš pusę metų pasirodė alternatyvi architektūra - Mamba, turinti daugelį pranašumų transformerų architektūrai. Mamba konteksto ilgis praktiškai neribojamas, kadangi skaičiavimo kaštai didėja tiesiškai. Tuo tarpu transformatoriaus skaičiavimo kaštai didėja kvadratu. Tad didysis kalbos modelis Mamba ne pamirš svarbių detalių per daugybę pokalbių. Deja dauguma dirbtinio intelekto (DI) modelių yra juodos dėžės – jie atlieka, tai ko prašoma, bet neaišku kaip. Dėl to kyla saugumo problema. Mechanistinio interpretavimo (MI) srityje dirbantys žmonės, atidarinėja DI vidų, bandydam i ji suprasti. Kadangi Mamba architektūra pasirodė tik prieš pusę metų ji tik pradedama tyrinėti MI srityje. Šiame darbe dalinuosi atlirk Mamba mechanistinio interpretavimo tyrimu.

Raktiniai žodžiai: Mamba, LLM, mechanistinis interpretavimas.

Abstract

Recently, large language models have been increasingly used. Nearly all the latest breakthroughs in artificial intelligence have utilized the transformer architecture. Six months ago, an alternative architecture known as Mamba emerged, offering several advantages over transformers. Mamba's context length is virtually unlimited because its computational costs increase linearly, whereas the computational costs of transformers increase quadratically. Thus, an AI assistant using Mamba won't forget important details over many conversations. Unfortunately, artificial intelligence is still largely a black box in many respects; it performs as requested, but how it does so remains unclear, raising security concerns. People working in the field of mechanistic interpretation are opening up these AI black boxes, attempting to understand them. Since the Mamba model was released months ago, it is only beginning to be explored in the field of mechanistic interpretability. In this work, I share my research by doing mechanistic interpretability in the model with Mamba architecture.

Keywords: Mamba, LLM, mechanistic interpretability.

Žymėjimai

Anglų kalbos terminų vertimas pateikiamas remiantis doc. Lino Pektevičiaus žodynu [31].

- Didysis kalbos modelis (angl. large language model, LLM).
- Skatinamasis mokymas (angl. reinforcement learning, RL).
- Parametrai – kintamieji, kurie nustatomi modelio apmokymo metu, bei po apmokymo nebesikeičia (angl. weights).
- Apmokymas – modelio parametrų keitimas (angl. model training).
- DI – dirbtinis intelektas (angl. artificial intelligence).
- MI – mechanistinis interpretavimas (angl. mechanistic interpretability).
- TNT – transformerių neuroninis tinklas (angl. transformer).
- TV – teksto vienetas (angl. token).
- TV generuotojas – funkciją tekštą paverčiantį į teksto vienetus (angl. tokenizer).
- L – modelio išvestis arba nesunormuotas TV tikimybės (angl. logits).

Ivadas

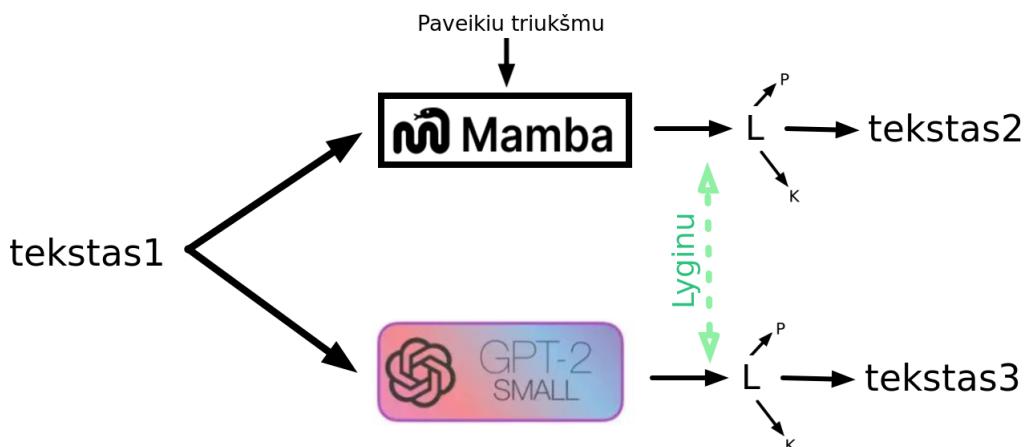
Atrodo, jog patys kuriame savo išpėdinius. Kiekvieną dieną juos tobuliname, ir suteikiama vis daugiau galios <...> Amžių eigoje tapsime antrarušiais gyventojais.

Cellarius 1863m.
„Darvinas tarp Robotų“

Darbo tikslas

Darbo tikslas ištirti didžiųjų kalbos modelį Mamba. Tai darysiu atlikdamas skaitmeninius eksperimentus su Mamba. Modelio išvestių pateiksiu vizualiai, jog būtų galima ižvelgti kokybinius skirtumus. Į Mamba vidų įdedant triukšmą stebėsime, kaip tai veikia išvestij. Kur galima Mamba lyginsių su GPT-2.

- Darbas buvo atliekamas google collab aplinkoje naudojant python programavimo kalbą.
- Tiriamas „hugging face delphi Mamba 130M“ modelis su „gpt-neox-20b“ TV generuotoju. Rezultatai lyginami su GPT-2 modeliu. Detali modelių schema (priedas 8).
- Tyrimui ypatinga svarbą atlieka „nnsight“ biblioteką, dėl jos, turiu galimybę modeliui paduoti kokį noriu tekstą, įdėti triukšmą ir nuskaityti išvestį.



1 pav.: Tyrimo schema.

Dirbtinio intelekto saugumo problema pradėjau domėtis prieš du metus, tad pirmuosiuose skyriuose pateikiu kiek ilgesnį, nei išprastą kontekstą, jei tai nedomina, galima pirmus keturis skyrius praleisti ir pradėti skaityti nuo 5 skyriaus „Apibrėžimai“.

Ateities prognozės

Anot Daniel Kahnemano eksperto nuomonė patikima tik tuo atveju, kai ekspertas savo įgūdžius įgijo aplinkoje, kurioje jų savo veiksmą nedelsiant gavo atgalinį ryšį. Pvz., investuotojams reikia laukti dešimtmečius, jog sužinotų apie savo investicijos sėkmungumą, ko pasekoje daugumas „ekspertų“ investuotojų pelningumas atitinka atsitiktinai parinktų investicijų pelningumą. Taip pat daugelis ekspertų neturi savo ekspertizės ribų suvokimo, tad pasitikėdami daro spėjimus ten kur nėra pagrindo galoti jų ekspertizei [24].

Analogiskai ilgalaikiui investavimui – DI padarinių nuspėjimas – veiksmingos apmokymo aplinkos nesudaro. Tad tvirtinčiau, jog DI ekspertų kokybinės¹ prognozės neturi validumo. Tai patvirtina 2024 m. sausio mėnesį atlikto tyrimo rezultatai, kuriame dalyvavo 2 778 DI mokslininkai. Žymi mokslininkų dalis tikisi DI utopijos, tačiau net du iš penkių priskyrė 10% tikimybę, jog DI padariniai bus katastrofiški (pvz., žmonijos išnykimas). Verta pabrėžti apklaustųjų vienbalsi sutarimą vienu klausimu – reikia teikti pirmenybę tyrimams siekiantiems sumažinti DI grėsmę [3].

Vienas pesimistiškiausių² DI balsų Eliezer Yudkowsky teigia, jog daugumą žmonių pradėjo domėtis DI saugumo problema visai neseniai, tad nesuvokdami problemos sunkumo optimistiškai tikisi, jog ją pavyks lengvai išspręsti. Tačiau Yudkowsky pabrėžia, jog aktualiausia sritis sprendžianti DI pavojų yra mechanistinis interpretavimas [19].

Alice Rigg indėlis

Šiam darbui pradžią padėjo padaryti tyrėja dirbanti MI srityje Alice Rigg³. Alice 2020 metais įgijo matematikos bakalauro laipsnį iš Kanadoje esančio Carleton universiteto, kur dirbo gilaus mokymo asistente ties automatinių teoremu įrodinėjimu. Dabar eina programuotojos pareigas iGraphFoldings įmonėje kurdama įrankius įvairių tinklų vizualizavimui.

Forume, kuriame diskutuojama įvairios aktualijos susietos su MI paklausiau ar kažkas dirbties naujai išleista architektūra – Mamba. I klausimą atsakė Alice pasidalindama savo Python skaičiavimų aplinka, kurioje parinko reikiamas bibliotekas[17] Mamba modelio interpretavimui. Būtent šią aplinką ir naudojau kaip pradžią savo tyrimui. Kiek vėliau Alice įkūrė išskirtinai Mamba MI [forumą](#), kuriame tyrėjai dalinasi savo mintimis.

¹Nors OpenAI išvedė empirines formules, kurios tiksliai nuspėja skaitinę modelio gerumo vertę, kuri bus pasiekti apmokant modelį [19]. Tačiau nuspėti kokybinių modelio savybių niekas nesugeba. Pvz., niekas nenumatė, jog modeliui pasiekus tam tikrą kiekybinį įvertį, programavimo varžybose jis aplenkis daugumą programuotojų.

²Yudkowsky ties DI saugumo problema dirba jau keletą dešimtmečių ir anot jo – DI katastrofa neišvengiama – lieka tik džiaugtis likusiu trumpu gyvenimu.

³Interviu su Alice Rigg <https://into-ai-safety.github.io/episode/interview/episode-4/>

1 Kodėl dirbtinis intelektas kelia pavojų?

Dirbtinio intelekto keliamą pavojų tiria DI saugumo (angl. AI safety) tarpdisciplinė sritis [14]. Scenarijų, kuriais DI sukelia masines katastrofas yra daug ir kur kas kūrybiškesnių, nei vaizduoja medija linkusi antropomorfizuoti DI.

1.1 Suderinamumo problema

Biologinę evoliuciją iš esmės galime modeliuoti [27], kaip paprasčiausią mašininio mokymo algoritmą – gradientinį nusileidimą. Tad žmogus šio algoritmo kurinys. Įdomu, jog egzistuoja elgesys iš esmės prieštaraujantis algoritmo optimizacijai. Pavyzdys – kontraceptikų naudojimas. Žmonės gauna hormoninį atlygi, neatlikdami pradinės atlygio paskirties. Kitas pavyzdys, tai didelis cukraus vartojimas keliantis sveikatos problemas. Kažkada saldumo mėgimo stimulas padėjo išgyventi dabar kenkia. Kadangi sugebame sukurti didelius kiekius cukraus. Žmogui intelektas, leidžia pasipriešinti evoliucijos atlygio funkcijai, gauti atlygi neatliekant atlygio paskirties [19].

Pavyzdžiui, RL žaidimo agentui atradusiam, jog važiuojant vienoje vietoje surenkamas didžiausias taškų skaičius, agentas būtent tai ir darys. Nes agento tikslas maksimalizuoti taškų skaičių, o ne inžinierių tikrosios intensijos. Tad, net esant mažam neatitikimui tarp ko norime ir ką užprogramuojame, rezultatas skirsis. Ir skirsis tuo labiau kuo galingesnį modelį turime [19].

Po apmokymo DI gali atlikti, tai ko norime. Tačiau sukurtas DI turi pakankamai gerai modeliuoja pasauly, jis suvoks, jog neatlikęs to ko iš jo norime, jis bus ištrintas. Tad, net jei DI turi skirtinga tikrajį tikslą, jis atliks mūsų užduotį, tam, kad turėtų galimybę ateityje pasiekti savo tikrajį tikslą.

Suderinamumo kaina

Jei⁴ suprantamas ir su žmogaus moralę sederintas modelis yra mažiau efektyvus už nesuprantamą ir nesuderintą modelį. Tada konkurencija tarp įmonių, šalių ar net atskirų individų stumia naudotis nesuderintais ir nesaugiais, bet labiau efektyviais DI modeliais. Norint, to išvengti būtina rasti interpretavimo ir sederinamumo būdus, kurie turėtų kuo mažesnį neigiamą poveikį modelio efektyvumui.

Galios siekimas

Jei apmokymas vyksta naudojant skatinamąjį mokymą (angl. reinforcement learning, toliau RL) yra įrodyta, jog optimali strategija linkusi siekti galios [11]. Supaprastintai agentas

⁴Yra teigiančių, jog sederintas modelis savaime bus efektyvesnis už nesuderintą, kadangi sederintas modelis tiksliau dirbs pagal tikruosius žmogaus poreikius.

turėdamas daugiau galios, turi daugiau galimybių optimizuoti savo tikslą funkciją. Tad agentui beveik visais atvejais apsimoka siekti galios.

1.2 Nubégimo scenarijus

DI modeliui pasiekus ar viršijus jį kuriančių inžinierių intelektą leistu, pačiam modeliui sukurti saveš tobulesne versiją, kartojant, gaunamas eksponentinis intelekto augimas [21]. Vieni spėja, jog tokis sprogimas, galėtų įvykti per metus, kiti per mėnesį ar net valandos laiko tarpe [19]. Toks greitas ir nekontroliuojamas intelekto augimas kelia grėsmę žmonijos kontrolės praradimui, kadangi gautume galingą modelį, kuris nebūtinai suderintas su žmonių tikslais. Pvz., turėdamas tikslą išgydyti vėžį visą populiaciją naudos vėžio gydymo eksperimentams.

Fizikinės ribos

Manau akivaizdu, jog fizikinėje erdvėje nesame įmanomo intelekto viršūnėje, vien turint omeny, jog kompiuterių procesorių sparta keliomis eilėmis didesnė už biologinių neuronų veikimą [9]. Bet, galbūt algoritminiu požiūriu esame netoli viršūnės? Tada greitas sprogimas negali įvykti, nes yra ribojantys faktoriai – pagaminti, bei įjungti procesorių užtrunka. Taip pat užtrunka naujo modelio treniravimas. Tad greitas scenarijus galimas, tik tada, kai labai galingam DI užtenka išnaudoti esamus skaičiavimo resursus, t.y. yra daug vietos tobuleti algoritminiu atžvilgiu.

Svarbu paminėti, jog vis didesnę naujų procesorių kūrimo išlaidų dalį sudaro skaičiavimo resursai. Tad tikrają ta žodžio prasme, geresni procesoriai, padeda kurti dar galingesnius procesorius. Beje eksponentinį skaičiavimo resursų augimą stebime jau virš 50 metų – Moore dėsnis.

Piktybiniai agentai

Meta⁵ kompanijos įkūrėjo Mark Zuckerberg nuomone, tokis DI nubégimo scenarijus yra mažai tikėtinės dėl fizikinių suvaržymų. Zuckerberg labiau bijo galios sutelkimo scenarijaus, kai prie galingo modelio turi prieiga nedidelė grupė žmonių, esant pakankamai galingam modeliui, turime galios monopolij. Tai kažkiek įtakoja Zuckerbergo kompanijos kuriamų Lamma modelių atvirą prieigą. Tačiau atvira prieiga išskelia piktybinių agentų pavojų⁶. Esant plačiai paplitusiai prieigai prie galingo modelio užtenka vieno piktybinio agento, jog būtų sukurtas sekantis Covid19 virusas [30].

1.3 Lėktuvas – geležinis paukštis

Neuromokslininkas John Vervaeke teigia, jog daugelyje aspektų mes esame optimalios programos būtybės [35]. O kai kurie tvirtina, jog vis dar tiksliai nesuprantama kaip veikia žmogaus

⁵Meta, tai naujas Facebook imonės pavadinimas.

⁶Piktybiniai agentai, pvz., teroristinių organizacijos, nusikaltėliai.

smegenys, todėl nėra galimybės kurti DI su žmogui prilygstančiu mąstymo gebėjimu. Manau ši išvada klaidinga, tai puikiai iliustruoja Max Tegmarko⁷ pateiktą analogiją: mes vis dar negalime sukurti į paukštį panašaus prietaiso, kuris būtų energetiškai optimalus, pats susirastą energijos šaltinį ir kurtų savo kopijas. Tačiau jau prieš 120 metų buvo sukurta sunkesnė už orą skrai-danti metalo konstrukcija varoma neefektyvaus benziniinio variklio [20]. Lėktuvo egzistavimo galimybę iki pat jo sukūrimo neigė net mokslo milžinas lordas Kelvinas⁸ – demonstruodamas, jog ekspertų prognozės neturi validumo. Panašiai su intelektu, nebūtina atkartoti žmogaus smegenų, galima sukurti stokojantį efektyvumo, neoptimalų ir trapų aparata, kuris savo greičiu ir apimtimi pranoksta evoliucijos kūrinius.

Dabartiniai neuroninių tinklų modeliai yra inspiruoti žmogaus smegenų. Tačiau tai tik inspiraciją, tarp DI ir žmogaus esti ryškūs skirtumai. Pavyzdžiui atgalinio sklidimo algoritmas, kuris įprastai naudojamas apmokyti DI neuroninį tinklą, biologijoje negali egzistuoti iš principo, kadangi biologiniai neuronai informaciją perduoda tik viena kryptimi.

Taip pat įdomu, jog dabartiniams DI modeliams išmokti atligli tą pačią užduotį (pvz., rašyti eileraštį, atpažinti katę, ir t.t.) reikia kur kas daugiau duomenų negu žmonėms. Tačiau pabrėžiama, jog šis lyginimas prastas, kadangi kūdikis nepradeda „nuo nulio“. Žmogaus genetinis kodas yra milijardus metų trukusios evoliucijos kūrinys. Ilgos evoliucijos metu žmogaus genetika buvo apmokyta naudojant milžiniškus kiekius duomenų. Kitas svarbus aspektas, jog tai pačiai užduočiai atligli DI užtenka kur kas mažesnio parametrų skaičiaus negu žmonėms [28].

Apibendrinant dirbtinį intelektą pavoju, kadangi jo plėtra gali sukelti žmonijos kontrolės praradimą dėl eksponentinio intelekto augimo, galios siekimo ir evoliucinių jėgų. Problema susijusi su suderinamumu tarp DI tikslų ir žmogaus poreikių yra svarbi, nes konkurenčija tarp žmonių skatina naudoti efektyvesnius, tačiau mažiau saugius modelius. Žmogaus intelekto ir DI veikimo prigimties skirtumai reikalauja naujų tyrimo metodų, kurie užtikrintu, jog galingi DI modeliai veiktų žmonių labui, o ne prieš juos.

2 Didieji kalbos modeliai

Didieji kalbos modeliai LLM - tikimybiniai natūralios kalbos⁹ modeliai. Įprastai apmokomi nuspėti sekantį žodį¹⁰ tekste naudojant didelius tekstynus, bei daug skaičiavimo resursų. Modelis išmoksta statistinius sąryšius tarp žodžių. Šiuos modelius galima naudoti tekstu klasifikavimui¹¹ arba teksto generavimui.

⁷Max Tegmark – fizikas, MI grupės įkūrėjas.

⁸William Thomson (lordas Kelvinas) – gabus fizikas už savo nuopelnus pelnės lordo titulą, jo garbei pavadinta SI temperatūros skalė.

⁹Natūralios kalbos pavyzdžiai: anglų, lietuvių, vokiečių kalbos

¹⁰Žodį naudoju dėl aiškumo skaitytojui. Žodis čia reiškia teksto vienetą.

¹¹GPT-2 buvo rastas neuronas, kuris puikiai klasifikavo vartotojų komentarus į neigiamus ir teigiamus atsiliepimus.

LLM apmokymui naudojami dideli tekstynai. Mažiausias tyrimo tikslams sintetiškai sukurtais „TinyStories“ tekstynas užima 1GB atminties ir sudarytas iš 2 milijonų teksto pastraipų [2]. „The Pile“ tekstynas užima 886GB, tačiau yra vienas iš mažesnių tekstytių. Sudarytas 2020 metais ir naudojamas lyginant Mamba su kitomis architektūromis (6 pav.). Tuo tarpu dabartiniams LLM įmonės OpenAI, Anthropic, Meta apmokymui naudoja beveik visą tekstinę informaciją randama internete, o kokybiškus informacijos šaltinius (knygas, straipsnius, programavimo pavyzdžius) pateikia kelis kartus [30].

LLM turi didelį parametrų skaičių. Eksperimentuose naudojamas GPT-2 turi kelis šimtus milijonų parametrų, GPT-3.5¹² turi 175 milijardus [32], GPT-4 apytiksliai¹³ turi 1.8 trilijonus [33], o jo apmokymas kainavo 100 milijonų dolerių [26].

„ChatGPT – tik įmantrus plagijavimas“ teigia skeptiškai LLM vertinantis lingvistas Noam Chomsky [16]. Panašią mintį išsakė Stefanas Wolfram¹⁴ teigdamas, jog LLM – naujoji Boilio logika. Analogiskai Boilio logikai LLM atskleidė anksčiau paslėptas gramatines taisykles, kuriomis remiasi natūrali kalba [22]. Idomu, jog žmogaus smegenyse natūralios kalbos centras yra aiškiai atskirtas nuo kitų sričių, net ir tokų artimų kaip matematika ar programavimas. Taigi žmogus gali mąstyti net neturėdamas kalbos modulio. Tad nenuostabu, jog LLM gali kalbėti, net ir neturėdamas prasminio žodžių suvokimo. Galbūt LLM išmoksta tik kalbos formą ir to pakanka. Išties dalis pavyzdžių rodo, jog LLM nesuvokia turinio. Pavyzdžiui pateikus modifikuotą Monty Hall problemą „Už pasirinktų durų prizo tikimybė 100% ar apsimoka keisti duris?“. LLM daug kartų matės šią problemą (problemoje teisingas atsakymas keisti duris), bet nesuvokdamas sąlygos pasikeitimo (pasikeitusioje problemoje teisingas atsakymas nekeisti durų), teigia, jog duris keisti apsimoka. Anot lingvisto Edward Gibson, šiuo metu LLM yra pirmaujanti natūralios kalbos teorija dėl neprilyginamai aukštos nuspėjamumo galios. Tačiau E. Gibson pažymi, jog LLM yra prasta teorija dėl savo dydžio ir juodos dėžės veikimo principo [18]. Vyriausiojo OpenAI mokslininko Ilia Sutskever teigimu, sekantio teksto vieneto nuspėjimas yra pakankama sąlyga žmogų pranokstančiam bendram intelektui susiformuoti. Kadangi norint visiškai tiksliai nuspėti sekantį teksto vienetą būtina sąlyga yra sudaryti tikslų pasaulio modelį [29].

LLM galingas įrankis, sugebantis kurti ir atrasti Štai Nature atspausdintame straipsnyje [10] naudojant LLM sugeneruotas python programas, buvo padarytas progresas netrivialiose matematikos problemose. Viena iš jų, tai didžiausios galimos aibės radimas sudarytos iš vektorių $\in \mathbb{Z}_3^n$ taip kad jokių trejų vektorių suma nebūtų lygi nuliui. LLM paremtas metodas pagerino

¹²GPT-3.5 modelis, tai pirmasis ChatGPT, pasaulį išvydęs 2022 metų lapkričio mėnesį.

¹³Tikslus svorių kiekis nėra žinomas, nes tokia informacija yra kompanijos paslaptis. Bet dydžio eilė įvertina, kiti srityje dirbantys ekspertai.

¹⁴Stefan Wolfram – Mathematica įkūrėjas, fizikas, „new kind of science“ autorius.

buvusį rezultatą $n = 8$ atveju. Taip pat dėžučių pakavimo problemoje LLM rado efektyvesnes empirines formules. Kol kas LLM vargai atlieka matematikos užduotis. Bet sukabinus LLM su atgaliniu ryšiu gauname galingą sistemą.

Šviežias pavyzdys – robotinis šuo balansuojantis ant kamuolio. Robotui, norint atlikti kažkokia užduotį (eiti, bėgti, balansuoti) reikalinga tikslas funkciją g , kuri naudojama roboto apmokymui (pvz., balansavimo užduočiai tikslas funkcija galėtų būti $g(h) = (h - 1)$, kur h roboto aukštis nuo žemės, tad robotas sieks išlaikyti save aukščiau, nei 1 metras nuo žemės). g įprastai parašo žmogus ir tai užima daug laiko. Tyrėjai šią užduotį perdavė LLM, kuris parašė daug skirtingų g . Pagal LLM parašytą g robotas buvo apmokomas fizikinėje simuliacijoje geriausiai apmokančios g buvo paliekamos. Kadangi LLM turi „begalinę kantrybę“, šį procesą buvo galima kartoti daug kartų ir prieitą prie geresnių, nei įprastai žmogaus parašomų tikslas funkcijų. Rezultatas vien simuliacijoje ištreniruotam robotui pavyko balansuoti ant kamuolio fizikinėje erdvėje [6].

3 Mechanistinis interpretavimas

Daugelis dabartiniu metu veikiančių DI modelių veikia juodos dėžės principu (2 pav.). Net ir turint prieigą prie kiekvieno LLM parametru dėl milžiniško LLM dydžio žmogui tiesiogiai suprasti LLM – neįmanoma. Įsivaizduokite lygtį su milijardu kintamųjų. Mechanistinis interpretavimas (MI) bando atverti dirbtinio intelekto juodają dėžę ir sukurti įrankius, kurie leistų suprasti kas vyksta šios DI juodos dėžės viduje.

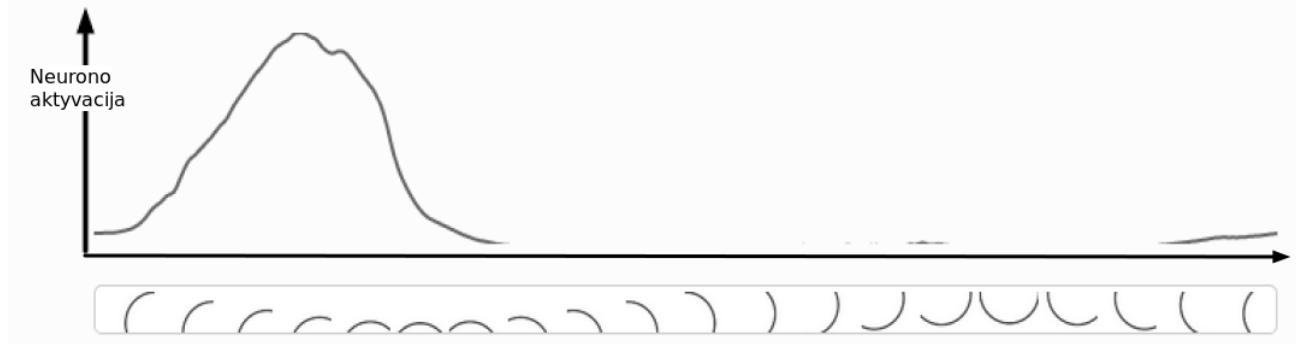


2 pav.: Juoda dėžė atlieka išvesties generavimą, nors nežinome kas vyksta jos viduje. ChatGPT būtų puikus juodos dėžės pavyzdys.

Anot MI tyrėjo Neel Nanda, MI sritis - kurioje bandoma sudėtingą dirbtinio intelekto modelių išskaidyti jų žmogui suvokiamas dalis [7]. Nuotraukų klasifikavimo modeliai patyrė ženklią pažangą mechanistinio interpretavimo atžvilgiu. Tuo tarpu transformerų LLM neturi paprasto būdo perteikti juose esančius algoritmus, kaip kad turi vizijos modeliai (4 pav.). Tad jų tyrimai sudėtingi, nepadeda, ir tai, jog LLM vieni didžiausių DI modelių ir pradėti tirti palyginus neseniai. Tačiau Mamba atsilieka dar labiau – kai pradėjau domėtis Mamba, nebuvo jokių Mamba tiriančių MI tyrimų.

Mechanistinio interpretavimo pavyzdžiai

Nuotraukų atpažinimo modeliuose pasiektas nemenkas MI įdirbis. DI viduje rastos dalys, kurios atpažįsta mikroskopines nuotraukos savybes (pvz., dažnių detektorai). Atrastos ir tos modelio dalys, kurios įgalina atpažinti bendresnius vaizdo bruožus (pvz., snukučio detektorius). Štai pateiktas konkretus kreivės atpažinimo pavyzdys (3 pav.). Pirmuojuose InceptionV1¹⁵ sluoksniuose rasti neuronai – kreivės detektorai, reagujatys į tik į tam tikrą kreivumą nuotraukoje. Vienas iš įrankių padedančių tirti nuotraukų modelius yra galimybė stebeti į ką tiksliai



3 pav.: Vertikalioje ašyje neurono aktyvacija, kuri atitinka horizontalioje ašyje pavaizduotą pusapskritimio nuotrauką. Adaptuota pagal [1].

reaguoja neuronas (4 pav.).



4 pav.: Pradinė nuotraukos įvestis yra atsitiktinis triukšmas (step 0). Sekančios nuotraukos gaunamos, kiekvienu žingsniu modifikuojant nuotrauką, jog būtų kuo labiau padidintas konkretaus neurono aktyvavimas modelyje. altinis [8].

Didžiajam kalbos modeliui pateikus šį sakinį:

„When Mary and John went to the store, John gave a drink to <?>“ (1)

¹⁵InceptionV – 2014 metų nuotraukų klasifikavimo „ImageNet Large Scale Visual Recognition“ turnyro nugalėtojas, sudarytas iš 22-jų sluoksninių.

GPT-2¹⁶ nuspėja¹⁷, jog $\langle ? \rangle = \langle \text{Mary} \rangle$. Kaip GPT-2 tai atlieka? Vienas iš pirmųjų MI rezultatų didžiuosiuose kalbos modeliuose būtent į tai atsako. Atskiriama modelio dalis, kuri teisingai nuspėja teksto vienetą panašios struktūros sakiniuose [12]. Kitaip tariant modelio viduje rastas algoritmas nustatantis, jog sekantis teksto vienetas $\langle \text{Mary} \rangle$. Kadangi Mamba modelis taip pat atliekā teisingą $\langle \text{Mary} \rangle$ nuspėjimą, šiame darbe didelę dalį laiko praleidau ieškodamas algoritmo, kuris atlieka šią užduotį, deja nerada.

Kitas pavyzdys vadinamas transformeriu modelių „grokking“. Treniruojant su moduliarinės aritmetikos užduotimi vyksta pastovus nuostolių funkcijos mažėjimas, ir po tam tikro treniravimo žingsnių įvyksta fazinis virsmas ir nuostolių funkciją vėl smunka ir pasiekia beveik nulį. Kas vyksta lemiantys tokį elgesį išsiaiškino Neel Nanda ir kiti. Pasirodo, jog pradžioje modelis „bando įsiminti“ duomenis, bet po kurio laiko modelis perpranta moduliarinę aritmetiką ir implementuoja moduliarinės aritmetikos algoritmą [7].

4 Neuroninio tinklo architektūrų palyginimas

Naujausi neuroninių tinklų proveržiai: kalbos (ChatGPT), nuotraukų (StableDiffusion, Midjourney, DALL-E), muzikos, vaizdo klipų (Sora) generavimas. Buvo įgalinti 2017 metais pasirodžiusios **transformeriu** **neuroninio tinklo** (TNT) architektūros. Esminis TNT komponentas – dėmesio matrica, kurioje kiekvienas teksto vienetas lyginamas su kitais įvestyje esančiais teksto vienetais. Šios architektūros pranašumas, jog galimas modelio apmokymas lygiagrečiai, kadangi matricos elementų poros nepriklausomos ir skaičiavimus galima atlikti atskirai. Tad galima skaičiavimus paskirstyti per daug atskirų kompiuterių ir greitai treniruoti. Deja, norint TNT modelį naudoti¹⁸ skaičiavimo ištekliai didėja kvadratu priklausomai nuo įvesties ilgio. Todėl LLM naudojantys transformatorių architektūra turi ribotą kontekstą¹⁹. Vieno galingiausio šiuo metu esančio modelio – GPT-4 konteksto ilgis siekia 4096 teksto vienetų [15]. Tad jei su GPT-4 asistentu susirašinėsite ilgiau, nei ≈ 4000 žodžių jis būtinai „pamirš“ pokalbio pradžią²⁰.

Klasikinė rekurentinio neuronininio tinklo (angl. recurrent neural network, RNN) architektūra kvadratinio išteklių didėjimo problemos neturi. Jos ištekliai priklauso tiesiškai nuo įvesties. Deja RNN neturi lygiagrečaus treniravimo: Treniravimo metu atgalinio sklidimo algoritmas treniruoja nuosekliai po vieną žingsnį. Nėra galimybės skaičiavimus išskaidyti, tad modelio apmokymas lėtas.

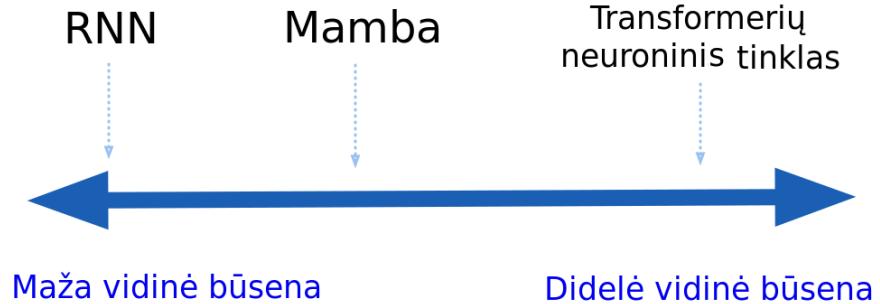
¹⁶GPT-2 – ChatGPT pirmtakas sukurtas 2019 metais.

¹⁷Dėl teksto aiškumo čia netiksliai naudoju tarpelius. Eksperimentuose tarpo po sakinio nėra, nes įprastai žodis į TV-us skaidomi su tarpu prieš žodį, tad išties tikrasis nuspėjamas TV yra $\langle \text{Mary} \rangle$, o ne $\langle \text{Mary} \rangle$.

¹⁸naudoti, t.y. duoti įvestį ir gauti išvestį

¹⁹kontekstas įvestis į kurią modelis gali aktyviai naudoti generuodamas išvestį

²⁰Arba kitą dalį teksto, gali tekštą pergrupuoti ir esmines detales išsaugoti.



5 pav.: Neuroninio tinklo architektūrų palyginimas: RNN neturi vidinės būsenos, tuo tarpu TNT „išsimena“ visą įvestį, mamba turi $h(t)$, kuri prisimeną svarbiausią informaciją. Panašiai, mes žmonės atsimename intuityvų pasaulio modelį, dauguma atskirų faktų pamirštame. Adaptuota pagal: [23]

Mamba arba S6 yra išrankieji būsenų erdvės modeliai. Yra patobulinimas paprastesnių S4 būsenų erdvės modelių (angl. state space models). Kurie turi savoje minimalų skaičių kintamujų, jog būtų galima pilnai apibūdinti sistemą. Pagrindinė Mamba dalis yra paprastųjų diferencialinių lygčių sistema:

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) \end{aligned} \tag{2}$$

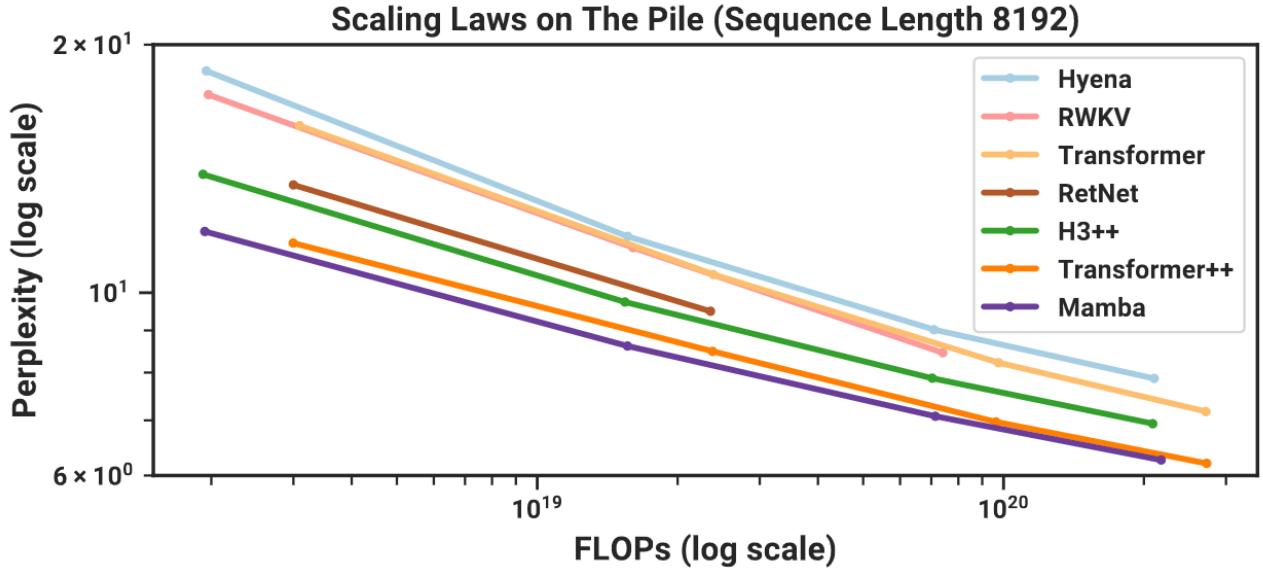
Kur $h(t)$ vidinė būsena, $x(t)$ įvestis, o $y(t)$ išvestis, o A, B, C – parametru matricos. Vidinė būsena $h(t)$ išlieka, tarsi intuityvus žmogaus pasaulio suvokimas.

Mamba architektūra skiriasi nuo įprasto S4, nes turi algoritmą, kuris pasirinktinai „peržvelgia“ įvestį ir sutraukia informaciją. Taip pat pradinei įvesčiai A naudoja HiPPo matrica, kuri padeda išsaugoti ilgus ryšius tarp TV.

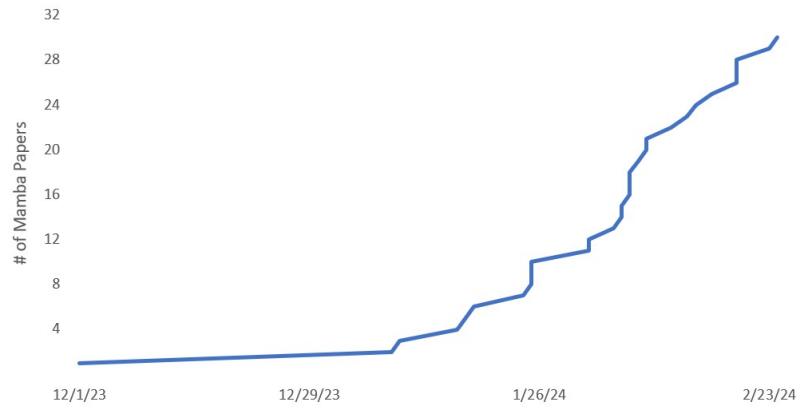
Mamba yra optimizuota architektūra kompiuteriui, kadangi jos bendraautorius yra pasaulio lygio lygiagretaus skaičiavimo procesorių ekspertas. Dauguma modernių kompiuterių išskaitant kompiuterius kuriais treniruojami ir naudojami DI modeliai turi greitą atmintį DRAM ir labai greitą atmintį SRAM esančia procesoriaus viduje. Didžioji laiko dalis užtrunka neskaičiavimuo-se, bet informacijos perdavime tarp procesoriaus ir atminties. Mamba sutalpina vidinę būseną $h(t)$ į SRAM. Taip stipriai padidindama modelio spartą.

Diskretizuojant (2) lygtį galime ją interpretuoti kaip konvoluciją ir gauti galimybę apmokyti Mamba lygiagrečiai, o diskretizavus, kaip rekurentinį sąryšį ir gauti tiesinį²¹ išvesties generavimą. Tad Mamba išlaiko RNN, bei TNT gerias savybes. Tikėtina dėl to susilaikė nemenko tyréjų dėmesio (7 pav.).

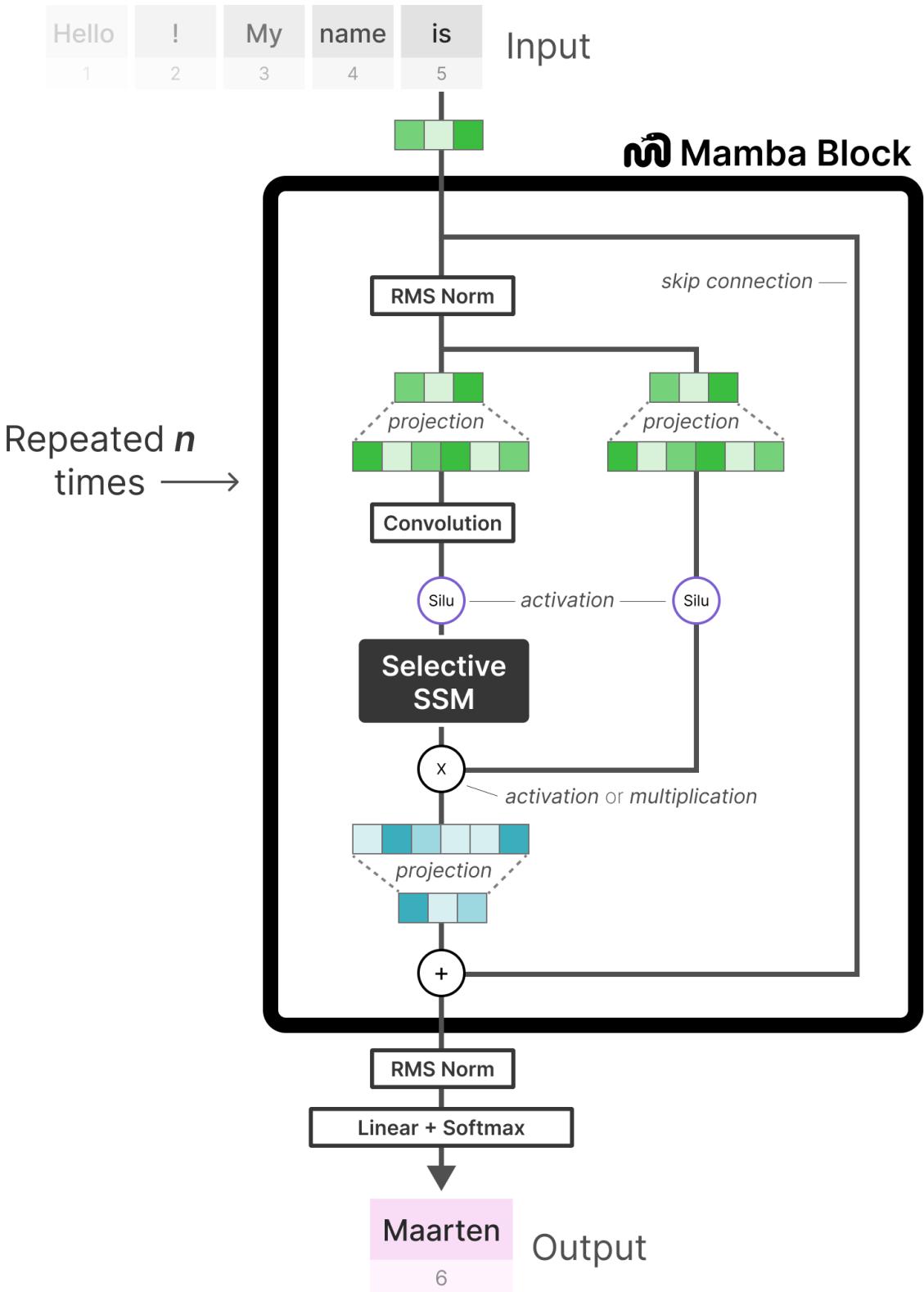
²¹Tiesinį, tai yra skaičiavimo resursų dydis didės tiesiškai priklausomai nuo išvesties.



6 pav.: Mambos treniravimas su „The Pile“ tekstynu, rodo, jog užtenka mažiau informacinių resursų pasiekti tuos pačius rezultatus. Log-Log grafikas, x ašyje yra operacijų skaičius, o y ašyje \mathcal{P} (angl. *perplexity*) matas apibūdinantis, kiek modelio skirtinys skiriasi nuo tikrojo skirtinio. Formaliai apibrėžiamas per kryžminę entropiją H , kaip $\mathcal{P} := 2^{H(p_M; p_D)}$, kur p_M modelio tikimybės tankis, o p_D duomenų tikimybės tankis. Šaltinis [4].



7 pav.: Mamba straipsnių kiekis sparčiai auga. Vertikalioje ašyje pavaizduota straipsnių kiekis naudojantis Mamba, o horizontalioje ašyje data (mėnesis/diena/metai). Šaltinis [34].



8 pav.: Mamba sluoksnis. Šio darbo tiriamame Mamba modelyje tokų sluoksnii yra 24. Esminė sluoksnio dalis, kuri skiriasi nuo kitų NN tinklų architektūrų yra „Selective SSM“ komponentas. „RMS Norm“ sluoksnii normalizacija, „projection“ – tiesinė projekcija, „Silu“ aktyvacijos funkcija, kuri įveda netiesiškumą, „Convolution“ – konvoliucijos operacija. Šaltinis [23].

5 Apibrėžimai

Šioje dalyje pateikiu apibrėžimus ir pradedu techninę atlikto darbo dalį. Pradékime nuo sluoksnio apibrėžimo pagal [13]:

Apibrėžimas 1. *Tarkim modelis f sudarytas iš funkcijų $f^{(i)}$ kompozicijos:*

$$f(x) = f^{(N)} \circ \dots \circ f^{(1)}(x)$$

Tada funkcija $f^{(i)}$, kur $i = 1, \dots, N$ yra sluoksnis.

5.1 Teksto vienetai

Teksto vienetas (angl. token, toliau – TV) šiek tiek klaidinanti savyoka, kadangi transformeriu, bei Mamba architektūros naudoja teksto vienetus garso įrašams, DNR nariams, šachmatų ėjimams, bei nuotraukų atpažinimui [5] koduoti. Teksto vienetas bendra prasme, tai suskaidytos nuoseklios įvesties (arba išvesties) informacijos vienetas. Tačiau šiame darbe dirbu tik su tekstu, tad apsiribosiu siauresniu apibrėžimu:

Apibrėžimas 2. *Teksto vienetas – vektorius, turinčio simbolinę reikšmę, bei ID, natūraliojo skaičiaus koduojančio teksto vienetą, pora.*

Apibrėžimas 3. *$\langle \cdot \rangle$ tekstas tarp laužinių skliaustų žymi simbolinę teksto vieneto reikšmę.*

Pvz., teksto vienetas ($\langle \text{Mary} \rangle$, 6393). Teksto vienetams gera analogija – telefonų knygą – ID nurodo numerį, o vektorius būtų žmogaus vardas. Paprastumo dėlei galima galvoti apie teksto vienetą, kaip apie atskirą žodį. Visgi svarbu atkreipti dėmesį, jog ne kiekvienas žodis turės atitinkamą teksto vienetą, o ir teksto vienetas nebūtinai koduoja tik vieną žodį. TV išvis gali būti neatvaizduojamas simboliais (pvz., TV žymintis failo pabaigą). Dėl vaizdumo pateikiui teksto skaidymą į teksto vienetus (9 pav.). Atkreipkite dėmesį, jog tos pačios reikšmės sakinyss anglų kalba užima mažiau atskirų teksto vienetų (14 TV), nei lietuvių kalba (26 TV). O kadangi kiekvieno TV generavimas kainuoja, generuoti tekštą anglų kalba pigiau, nei lietuvių kalba.

Apibrėžimas 4. *Funkcija G , kuri tekštą paverčia į ID vektorių vadinsime TV generuotoju (angl. tokenizer).*

Įprastai G treniruojama atskirai nuo modelio. Šiame darbe naudojamas „GPT-neox-20b“ TV generuotojas yra apmokytas naudojant „The Pile“ tekstyną. Svarbu pabrėžti, jog G prideda papildomų laisvės laipsnių ir painiavos situacija komplikuojasi ir tai, jog naujausiųose LLM (pvz., GPT4) įvesties ir išvesties G nesutampa [25].

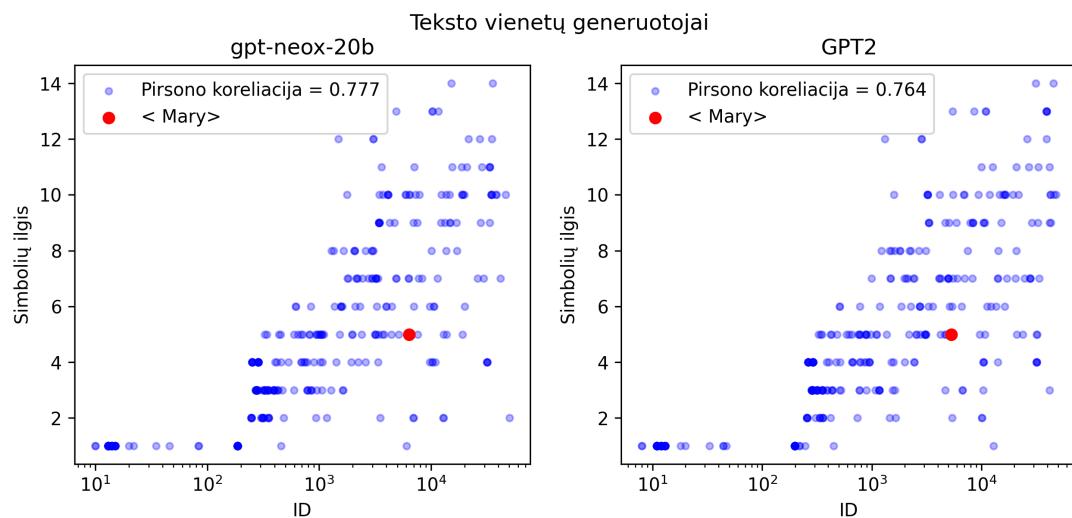
Pateikiu atliktą palyginimą tarp GPT-2 ir Mamba TV generuotojų (10 pav.). Mamba, bei GPT-2 TV generatoriui pateikiu tą patį anglų kalbos tekštą. TV generatorius tekštą konvertuoja į ID vektorių. Matome aiškią koreliaciją tarp teksto vieneto simboliių ilgio ir ID. Manau

When Mary and John went to the store,
John gave a drink to

Kai Marija ir Jonas éjo į parduotuvę,
Jonas davė atsigerti

9 pav.: Sakinys, bei jo vertimas į lietuvių kalba. Skirtingos spalvos žymi skirtingus teksto vienetus. Paveikslas gautas naudojant [tiktokenizer.vercel.app](#) įrankį.

koreliacijos priežastis paprasta – trumpesni žodžiai vartojami dažniau, tad TV generuotojas į tai atsižvelgia sumažindamas atmintį reikalingą teksto vienetų laikymui.



10 pav.: Kairėje konvertavimą atlieka tiriamo Mamba TV generatorius „gpt-neox-20b“, dešinėje GPT-2. Y ašyje TV simbolų ilgis, o X ašyje logaritmuoti TV atitinkantys ID. Pažymėto teksto vieneto < Mary> ilgis 5 simboliai, jo kairėje ID_{GPT2} = 6393, o dešinėje ID_{gpt-neox-20b} = 5335.

Pavyzdys

Tekstui „When Mary and“ ir pritaikykime TV generuotoja. Tekstas suskaidomas į teksto vienetus ir koduojamas į ID vektorių.

$$\text{„When Mary and“} \xrightarrow{G \text{ skaido į}} \begin{bmatrix} <\text{When}> \\ <\text{Mary}> \\ <\text{and}> \end{bmatrix} \xrightarrow{G \text{ koduoja}} \begin{bmatrix} 3039 \\ 6393 \\ 285 \end{bmatrix} \quad (3)$$

5.2 Pirmasis sluoksnis – įdėtis

Iprastai LLM įskaitant Mamba turi įdėties sluoksnį (angl. embedding), kuris kiekvienam ID priskiria n-dimensių vektorių. Mamba, bei GPT-2 atveju $n = 768$. Kadangi skirtinę teksto vienetų skaičius $\approx 50k$, tad įdėties sluoksnis TV iš didesnės erdvės, kur kiekvienam teksto vienetui bijektyviai priskiriama dimensija, konvertuoja į mažesnę 768 dimensijų erdvę. Tai yra dimensija sumažinama 65 kartus:

$$Emb : M_T^{50280} \rightarrow M_T^{768}$$

Sumažinus erdvę vyksta „informacijos sutraukimas“, dėl ko dalis MI tyrėjų hipotezuoją, jog kiekviena vidinės dimensijos vektoriaus kryptis atspindi kažkokią savybę (angl. feature). Štai plačiai žinomas pavyzdys:

$$Emb(<\text{karalius}>) - Emb(<\text{vyras}>) + Emb(<\text{moteris}>) \approx Emb(<\text{karalienė}>)$$

Tai yra iš karaliaus atėmus vyro ir pridėjus moters vektorių gauname karalienę! Kas nuostabu, jog tai tik vienas sluoksnis ir šį sluoksnį nesunku ištreniruoti naudojant nej asmeninį kompiuterį. Tęsiant (3) pavyzdį įdėties sluoksnis kiekvienam ID priskirtų vidinės dimensijos vektorių:

$$\begin{bmatrix} 3039 \\ 6393 \\ 285 \end{bmatrix} \xrightarrow{\text{Emb}} \begin{bmatrix} -0.14 & 0.35 & \dots & 0.94 \\ 0.50 & -0.77 & \dots & 0.03 \\ -0.04 & 0.42 & \dots & 0.22 \end{bmatrix} \quad (4)$$

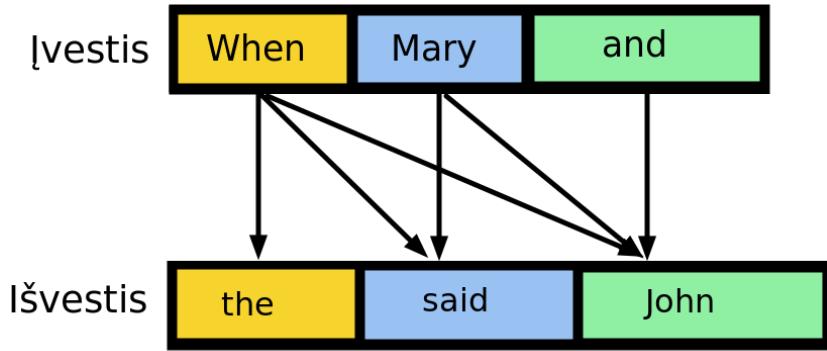
O tęsiant telefonų knygos analogiją: vidinės dimensijos vektorius apibūdintu žmogaus ūgi, plaukų spalvą, svorį, akių spalvą, batų dydį ir t.t., tad būtų galima atpažinti žmogų iš jo detalaus apibūdinimo, net ir nežinant jo vardo ar telefono numerio.

5.3 Išvesties generavimo tvarka

Tarkime turime tokią įvestį: a_1, a_2, \dots, a_i čia a_i teksto vienetas. Modelio išvestis b_1, b_2, \dots, b_i bus sugeneruojama dalinai „paslepiant“ dalį įvesties: generuojant pirmąjį išvesties narį b_1 modelis f „mato“ tik a_1 :

$$f(a_1) = b_1$$

Apibrėžimas 5. *mato $x :=$ modelis naudoja įvestį x išvesties skaičiavimui.*



11 pav.: Teksto vienetų generavimo tvarka, rodyklė atspindi informacijos įtakojimą, tai yra generuojant $\langle \text{the} \rangle$ modelis „mato“ tik $\langle \text{When} \rangle$, o generuojant $\langle \text{said} \rangle$ modelis „mato“ ir $\langle \text{When} \rangle$ ir $\langle \text{Mary} \rangle$.

Generuojant antrąjį b_2 modelis matys a_1 ir a_2 .

$$f(a_1, a_2) = b_2$$

Ir taip toliau:

$$f(a_1, a_2, \dots, a_i) = b_i$$

Tad modelis pateikia spėjimus atitinkančius kiekvieną įvesties teksto vienetą.

5.4 Išvestys L, T, P

Apibrėžimas 6. *L – matricinė modelio išvestis. Matricos eilutę nurodo indeksas T, o stulpelį ID.*

Kiekvienas L stulpelis atitinka teksto vienetą. Jeigu įvestimi naudoju jau paminėtą sakinių (1), kuris sudarytas iš 14 teksto vienetų (TV). Modelio išvestimi (L) gausiu matricą, kuri sudaryta iš 14 eilučių ir 50280 stulpelių:

$$L = \begin{bmatrix} L_0^0 & L_1^0 & \cdots & L_{6393}^0 & \cdots & L_{50279}^0 \\ L_0^1 & L_1^1 & \cdots & L_{6393}^1 & \cdots & L_{50279}^1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ L_0^{14} & L_1^{14} & \cdots & L_{6393}^{14} & \cdots & L_{50279}^{14} \end{bmatrix}$$

T tai teksto vieneto pozicija, $T = 0$ pirmasis teksto vienetas. Paskutinis T , tai yra modelio spėjimas sekančio žodžio. Modelis nuspėja sekančius teksto vienetą nuosekliai eidamas per duotą ivestį ir pateikia tokio paties dydžio išvestį.

Pritaikius eksponentinį normalizavimą iš L gauname P – išvestyje esančių teksto vienetų tikimybes.

$$P_i = \frac{e^{L_i}}{\sum_j e^{L_j}} \quad (5)$$

Tam, jog rezultatai būtų pilnai deterministiški²². Vietoj softmax funkcijos naudoju max funkciją: iš L eilutės imu teksto vienetą turintį didžiausią L vertę (arba didžiausią P vertę).

$$\begin{bmatrix} P_0^0 & P_1^0 & \cdots & P_{6393}^0 & \cdots & P_{50279}^0 \\ P_0^1 & P_1^1 & \cdots & P_{6393}^1 & \cdots & P_{50279}^1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_0^{14} & P_1^{14} & \cdots & P_{6393}^{14} & \cdots & P_{50279}^{14} \end{bmatrix} \rightarrow \begin{bmatrix} \max(P_i^0) = P_t^0 \\ \max(P_i^1) = P_k^1 \\ \vdots \\ \max(P_i^{14}) = P_m^{14} \end{bmatrix} \rightarrow \begin{bmatrix} ID = t \\ ID = k \\ \vdots \\ ID = m \end{bmatrix} \xrightarrow{G} \begin{bmatrix} \text{When} \\ \text{Mary} \\ \vdots \\ \text{Mary} \end{bmatrix}$$

6 Išvesčių lyginimas

Geriau kartą pamatyti, negu šimtą kartų išgirsti.

Šiame skyriuje pateikdamas skirtingas ižestis vizualizuoju skirtumus tarp modelių.

6.1 Vaizdavimas

Norėdamas pateikti kaip kinta konkretaus teksto vieneto tikėtinumas priklausomai nuo T įvedu K . Jis turi teigiamą aspektą, jog grafike „nesunaikina“ mažai tikėtinų teksto vienetų,

²²Visada gausiu tą patį išvestį. Jeigu naudočiau softmax funkciją, tai vieną kartą modelis pasirinktų vieną teksto vienetą, bet kitą kartą gali pasirinkti kitą teksto vienetą pagal tikimybės skirtinių. Galiu, gauti deterministinį elgesį, bet tada reikės naudoti tą pačią „SEED“ reikšmę.

bet išsprendžia „blokelių šokinėjimą“ (24 pav.). Jei L nedomifikuosime paveiksle matysime varijavimą, kai TV tikimybė nekis, kadangi visas eilutės ansamblis kinta nuo T .

Apibrėžimas 7. K , tai L , kur kiekviena K vertė normuota – padalinta iš L eilutės aritmetinio vidurkio modulio.

$$K_{ID}^T = \frac{L_{ID}^T}{\frac{1}{\max(ID)} |\sum_{ID} L_{ID}^T|} \quad (6)$$

Tai yra kiekvieną stulpelį padalinu iš eilutės vidurkio, taip gaunu nesunormuotas tikimybes – K ir kiekvieną teksto vienetą atitinkantį stulpelį vaizduoju atskira kreive viename grafike (15, 17, 19, 20, 21 pav.).

$$L = \begin{bmatrix} L_0^0 & L_1^0 & \cdots & L_{6393}^0 & \cdots & L_{50279}^0 \\ L_0^1 & L_1^1 & \cdots & L_{6393}^1 & \cdots & L_{50279}^1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ L_0^{14} & L_1^{14} & \cdots & L_{6393}^{14} & \cdots & L_{50279}^{14} \end{bmatrix} \xrightarrow[\text{normuoju}]{\text{skaidau}} \left(\begin{bmatrix} K_0^0 \\ K_1^0 \\ \vdots \\ K_0^{14} \end{bmatrix}, \begin{bmatrix} K_1^0 \\ K_1^1 \\ \vdots \\ K_1^{14} \end{bmatrix}, \dots, \begin{bmatrix} K_{50279}^0 \\ K_{50279}^1 \\ \vdots \\ K_{50279}^{14} \end{bmatrix} \right) \quad (7)$$

Analogiškai atvaizduoju tikimybes, padalindamas matricinę tikimybių išvestį P į stulpelius ir kiekvieną stulpelį atvaizduodamas atskira kreive.

6.2 K skirstiniai

Šiai tyrimo daliai neturėjau jokios hipotezės, bet manau, tai vienas sėkmingiausių šio darbo eksperimentų. Mamba K skirstiniuose rastas „gauburiukas“ (12, 13 pav.), kuriame esantys teksto vienetai kokybiškai skiriasi nuo atsitiktinių teksto vienetų (6.2 TV), jie sudaryti iš tarpų, bei naujos eilutės simbolių (6.2 TV). Tokio „gauburiuko“ neradau GPT-2 (14 pav.).

```

1 inedractysisologyiversiningopeologicalthoughasingcretideooudposedocument
2 ulatealyippwiducedcinelseuyakersoptioninateszingr00STaphintelavascr
3 iptastywanauxalloconconductptonaza centersarertanimmregularasia rangesconn
4 ectioncuuctionincoln mediated

```

TV 1: Atsitiktinė teksto vienetų imtis.

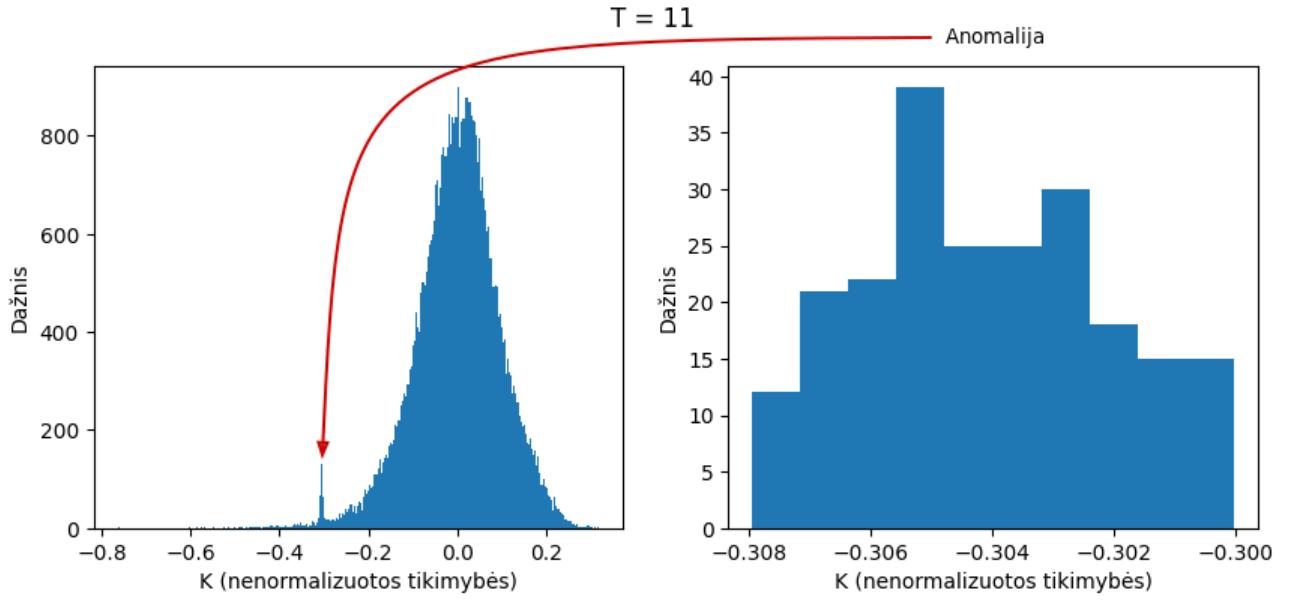
```

1 <| padding |>
2 \n          \n          \n          \n          \n          \n
3 \n          \n          \n\n\n          \n          \n          \n
4 \n          \n          \n\n\n

```

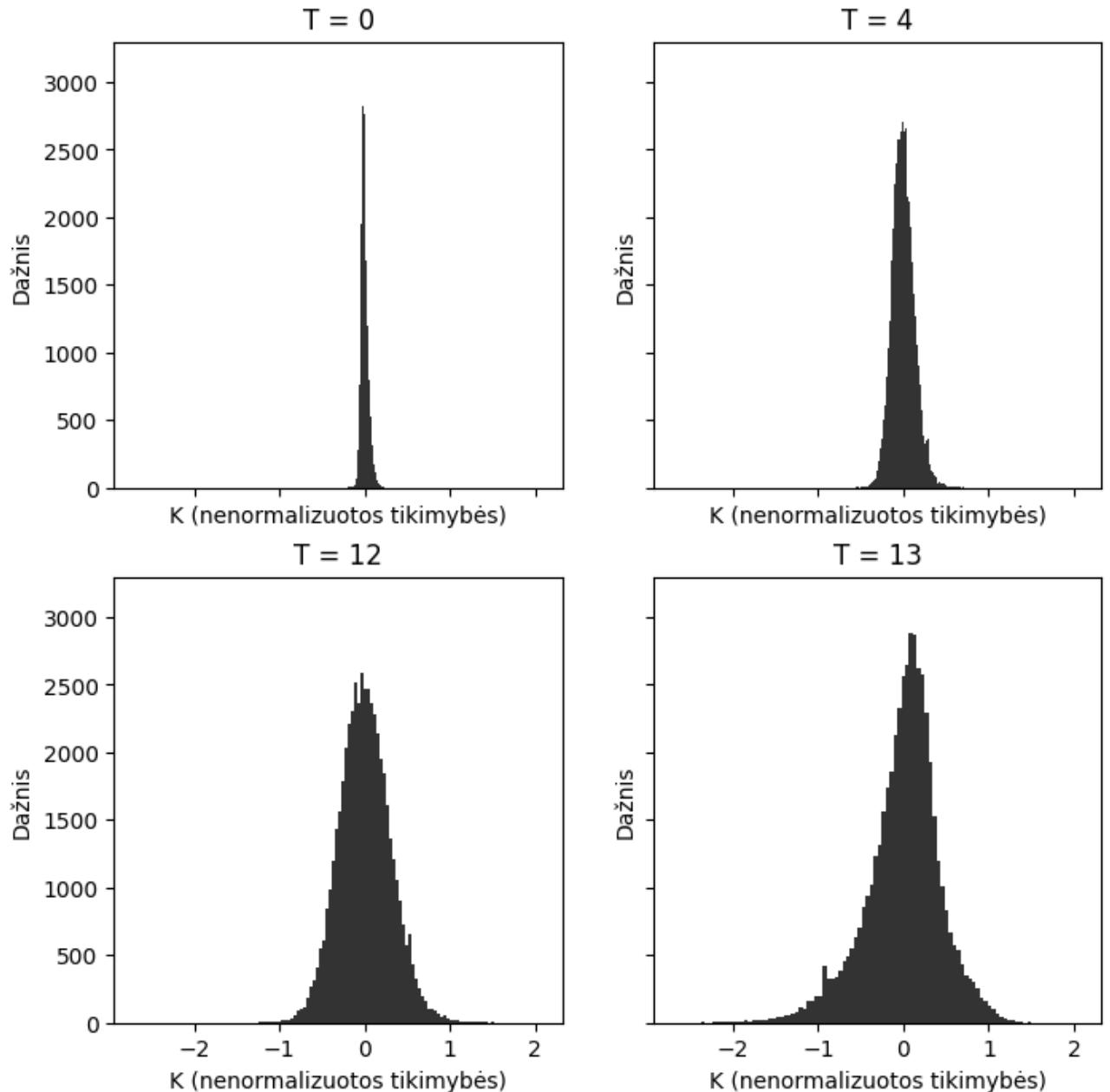
TV 2: „Guziuke“ esantys teksto vienetai sudaryti iš tarpų ir naujos eilutės simbolių.

Grįžtant prie Mamba skirstinių (13 pav.) įdomu, jog jei neatsižvelgiama į K absoliatū dydį, pirmoje ($T = 0$) ir paskutinėje ($T = 13$) pozicijoje esančių teksto vienetų skirstiniai iš pažiūros yra vienos kito veidrodinis atspindys vienas pakrypęs į dešinę, kitas į kairę. Taip pat Mamba



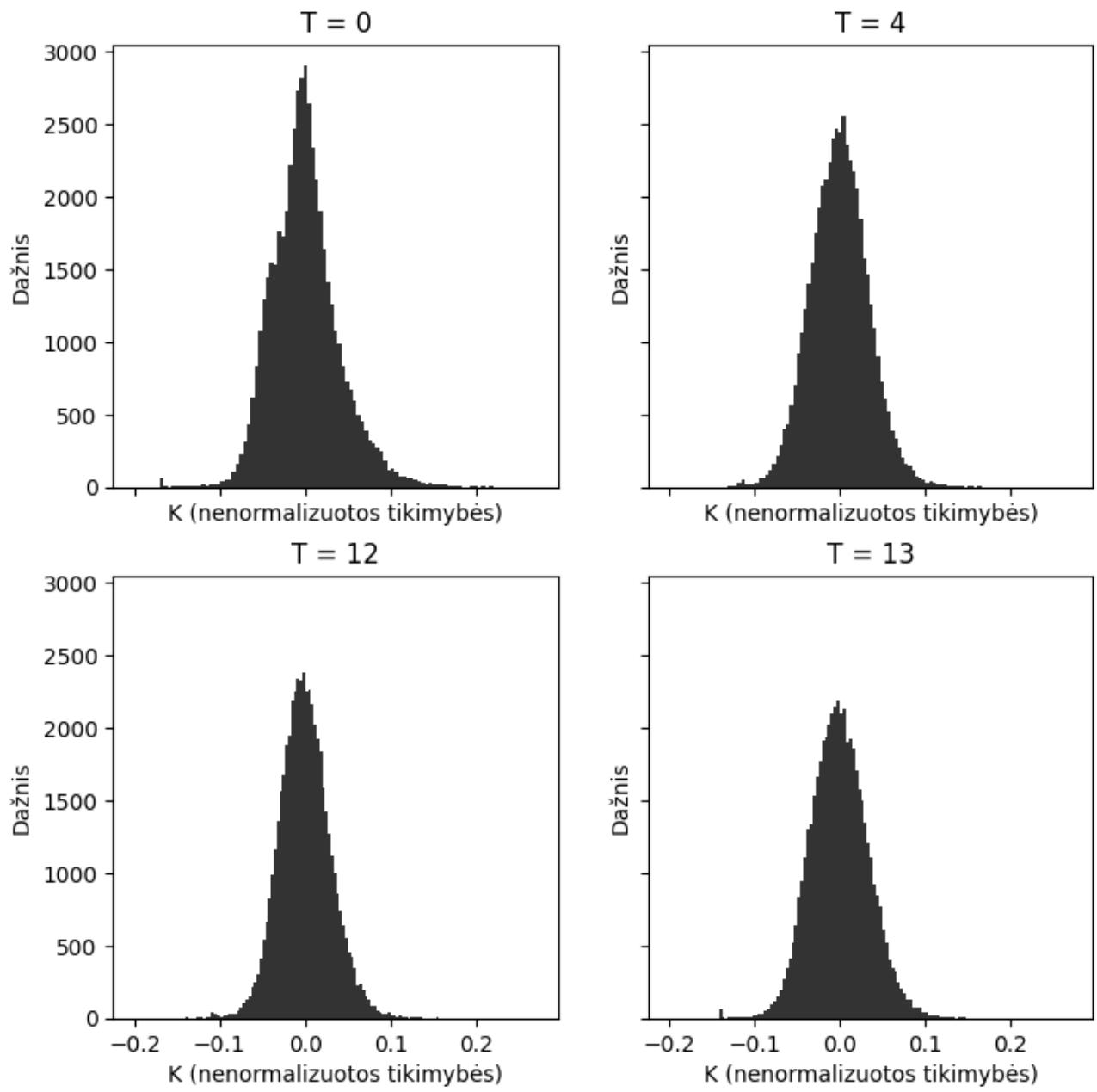
12 pav.: Čia iš arčiau pavaizduotas skirstinio anomalija („guziukas“), vertės kurios išsišoka iš skirstinio. Įvestis: „When Mary ...“⁽⁸⁾; Modelis GPT-2.

skirstinių variacija didėja priklausomai nuo T . Tai yra kuo daugiau įvesties matė Mamba, tuo didesnė K skirstinio variacija (13 pav.). Tad matome viena ryškiausiu kokybinių skirtumų tarp Mamba ir GPT-2, kadangi GPT-2 skirstinio variacija nė kiek nedidėja (14 pav.).



13 pav.: K skirtiniai. K skirtinių variacija didėja nuo T (teksto vieneto vietas išvestyje). Matomas „gauburiukas“. Įvestis „When Mary[...]to“ (8); Modelis Mamba.

GPT2 Anomalijos



14 pav.: K skirstiniai. K skirstinių variacija nesikeičia nuo T . Ivestis „When Mary[...]to“ (8); Modelis GPT-2.

6.3 Natūralios kalbos sakinio ir monotoninės įvesties palyginimas

Palyginkime, kaip atrodo natūralios kalbos sakinys, bei to paties teksto vieneto pasikartojanti sakinį. Kaip kis tikimybės? Kuo skirsis Mamba ir GPT-2 išvestys?

Hipotezė 1. *GPT-2 ir Mamba dėl savo skirtingu architektury turės kokybiškai skirtinges išvestis.*

6.3.1 „Mary“ ir „123“ įvestys

Jau minėta (1) įvestis išskaidoma į 14 teksto vienetų (9 pav.):

$$\begin{array}{c} \text{„When Mary and John went to the store, John gave a drink to“} \\ \downarrow \\ [3039, 6393, 285, 2516, 2427, 281, 253, 4657, 13, 2516, 3534, 247, 5484, 281] \end{array} \quad (8)$$

Analogiskai į 14 teksto vienetų išskaidomas ir (9), tačiau tai tas pats teksto vienetas (<123>, 10683) pasikartojantis 14 kartų.

$$\begin{array}{c} \text{„123123123123123123123123123123123123123123123123“} \\ \downarrow \\ [10683, 10683, \dots, 10683] \end{array} \quad (9)$$

6.3.2 Metodas

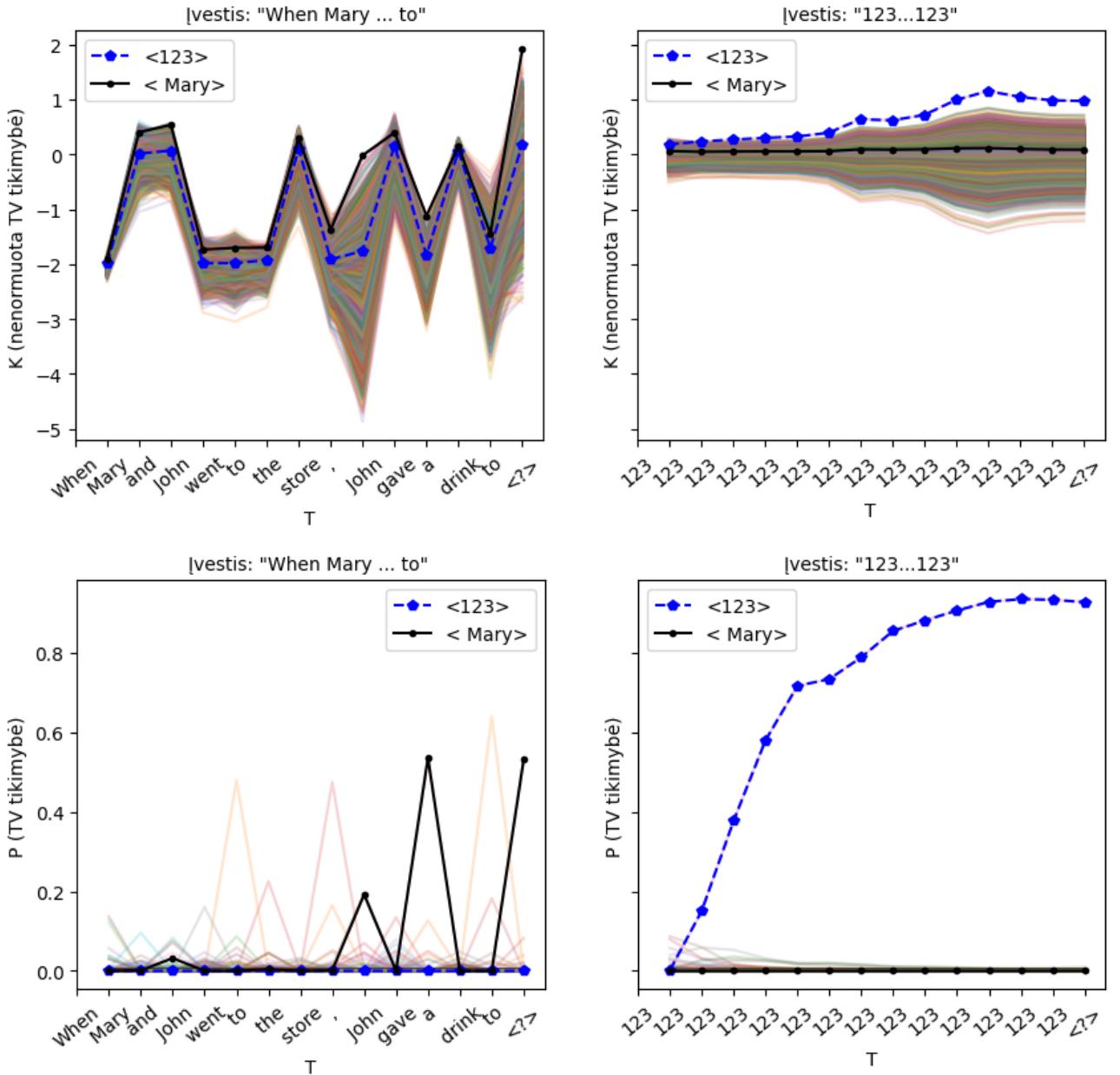
Natūralios kalbos įvesti „When Mary[...]to“ (8), bei monotonine vieno TV įvesti „123[...]123“ (9) pateikiu Mamba ir GPT-2. Išvesti K atvaizduojant grafike pagal (6), (7) formules. O P gaunu pagal (5), toliau analogiskai (7) suskaidau į stulpelius ir pavaizduoju. Šio stulpelio T vertė atitinka tikimybę teksto vienetui būti išrinktam T išvesties pozicijoje.

6.3.3 Rezultatai

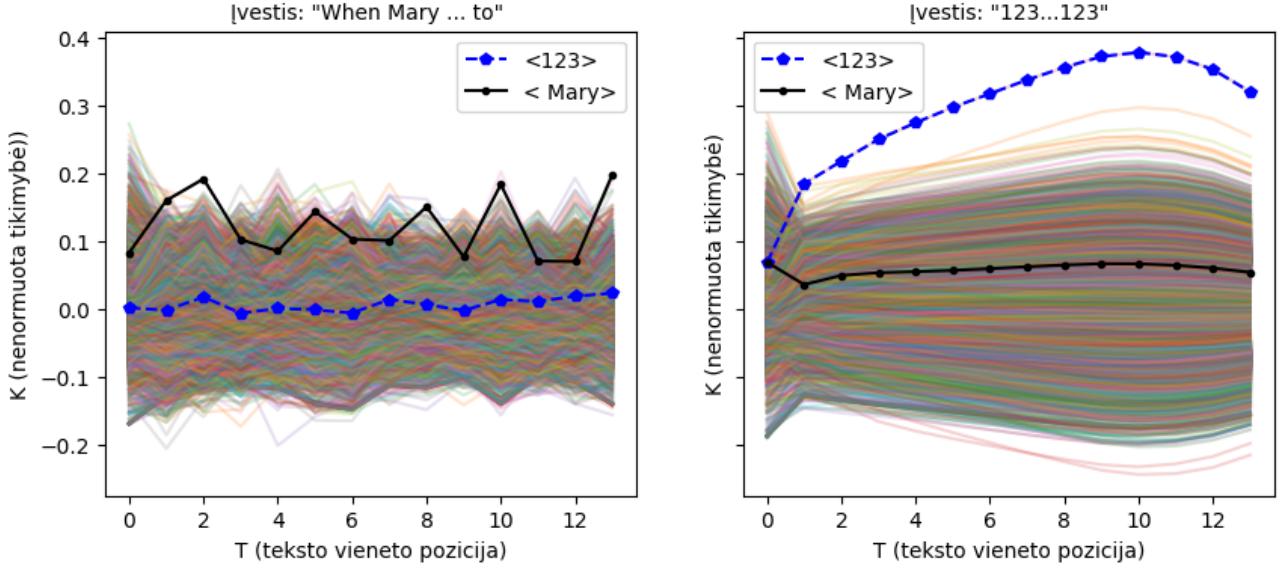
Abiejų modelių galutinės išvesties tikimybės daugumoje atveju sutampa (1 lentelė). Vietose kur Mamba išvestyje matome aukštą teksto vieneto <Mary> tikimybę ten aukštą tikimybę <Mary> priskiria ir GPT-2. Abiejų modelių monotoninės įvesties atveju teksto vieneto <123> tikimybė pradeda mažėti ties $T = 10$. Taip pat abiejų modelių išvestys su „When Mary[...]to“ įvestimi ir su „123[...]123“ įvestimi kokybiškai skiriasi (15, 16, 17 pav.). su „When Mary[...]to“ įvestimi tikimybės varijuoja – pažymėto teksto vieneto vertės, tai padidėja, tai sumažėja. Tuo tarpu monotoninės įvesties grafikuose tikimybės stabiliai kinta ir tokio šokinėjimo nedemonstruoja. Tačiau Mamba <123> TV tikimybė P_{123} pakyla lėčiau, nei GPT-2. Bendrai paėmus tarp skirtingu modelių daugiau panašumų, nei skirtumų.

	Ivestis: „When Mary[...]to“	Ivestis: „123[...]123“
Mamba ir GPT-2	Tikimybė P_{Mary} aukšta, kai $T \in \{2, 8, 10, 13\}$	P_{123} didėja, kai $T < 10$ ir mažėja, kai $T > 10$
Mamba prieš GPT-2	Ties $T = 2$ tikimybės skiriasi	GPT-2 P_{123} auga sparčiau už Mamba

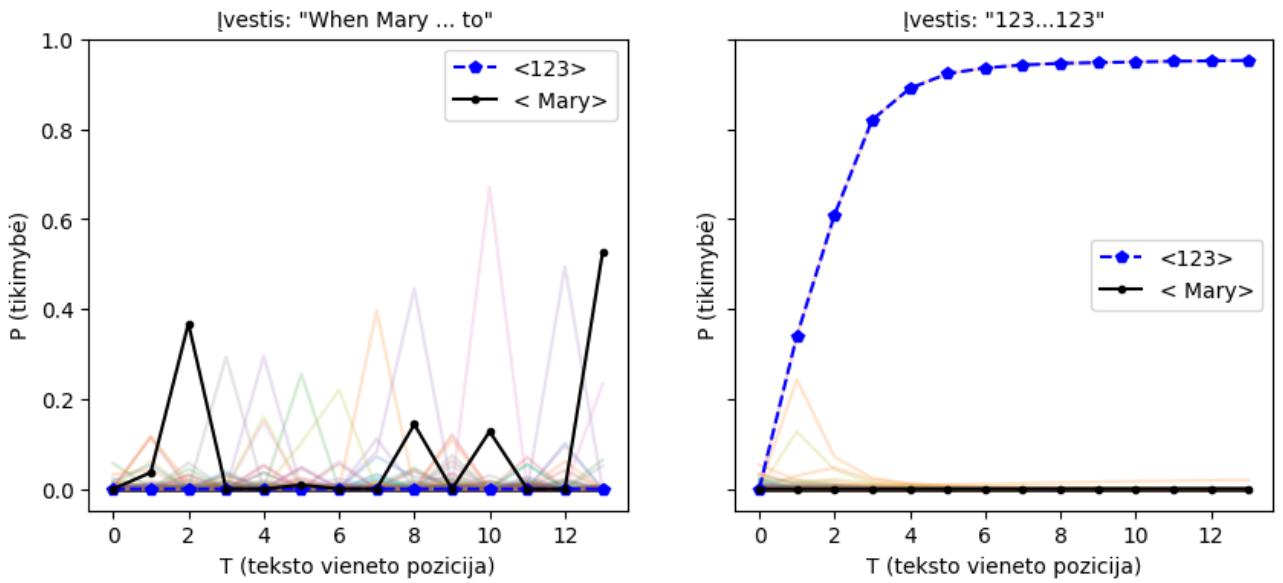
1 lentelė: Teksto vieneto tikimybės palyginimas tarp modelių ir įvesčių.



15 pav.: Viršutiniuose grafikuose išvestis K apibrėžta pagal (6) formulę, o apatiniuose TV tikimybės P . Kaireje išvestis „When Mary[...]to“ (8), o dešinėje „123[...]123“ (9). Modelis Mamba.



16 pav.: Išvestis K apibrėžta pagal (6) formulę. Kairėje įvestis „When Mary[...]to“ (8), o dešinėje „123[...]123“ (9). Modelis GPT-2.



17 pav.: TV tikimybės P . Kairėje įvestis „When Mary[...]to“ (8), o dešinėje „123[...]123“ (9). Modelis GPT-2.

6.4 Periodinės įvestys

Žinome, jog LLM „spėja“ žodžius, bet tiksliai kiek ši spėjimą įtakoja esama įvestis, o kiek spėjimas atsižvelgia į apmokymo duomenis.

Hipotezė 2. *LLM bando nuspėja sekanti TV pagal teksto struktūrą. Modeliui darant spėjimą po vieno teksto vieneto a įvesties, nematant daugiau jokios įvesties, sekantis teksto vienetas bus A, jei modelio apmokymo tekstyne pora (a, A) buvo dažniausiai. Panašiai jei modelis mato du teksto vienetus (a, b) spėjamas W jei (a, b, W) kombinacija matyta dažniausiai. Tačiau jei modelis mato (a, b, a, b, a, b) didelis kontekstas ir nors tekstyne tokios sekos nebuvo. Modelis „perpratęs“ esamų teksto vienetyų struktūrą į šią struktūrą atsižvelgia bandydamas nuspėti. Tad modelio išvestis pradžioje (T mažas) bus atsitiktinė ir didėjant T perpras seką ir ją idealiai atkartos.*

Hipotezė 3. *Mamba ir GPT-2 architektūros skiriasi, bet abu yra veikiantys LLM. Tad turėtų atlikti užduotį, bet tikėčiausi kažkokiu skirtumų. Kadangi transformeris aktyviai atmintyje laiko visą gautą įvestį (jos nekompresuoja), spėčiau, jog GTP2 modelis greičiau pradės generuoti „teisingus“ teksto vienetus.*

6.4.1 Metodas

Modeliui duodu pasikartojančius teksto vienetų seka $[a, b, \dots, a, b]$. Kad rezultatas būtų bendresnis naudoju mažus $[42, 69, \dots, 42, 69]$, bei didesnius $[4200, 6660, \dots, 4200, 6660]$ ID turinčią įvestį. Šiek tiek problemų kelia, tai, jog GPT-2 ir Mamba turi skirtingą teksto vienetų generuotoją. Pvz.,

$$G_{\text{Mamba}}(42) = \langle \text{I} \rangle$$
$$G_{\text{GPT-2}}(42) = \langle \text{K} \rangle$$

Tad pilnumo dėlei pateikiu tris grafikus pirmame (19 pav.) Mamba su jau minėtomis ID įvestimis $[42, 69, \dots]$ ir $[4200, 6660, \dots]$. Antrame (20 pav.) GPT-2 su tokiais pačiais teksto vienetų ID, tačiau skirtinė simbolinė reikšme. Trečiame (21 pav.) GPT-2 su skirtiniais ID, bet tokia pačia simbolinė reikšme.

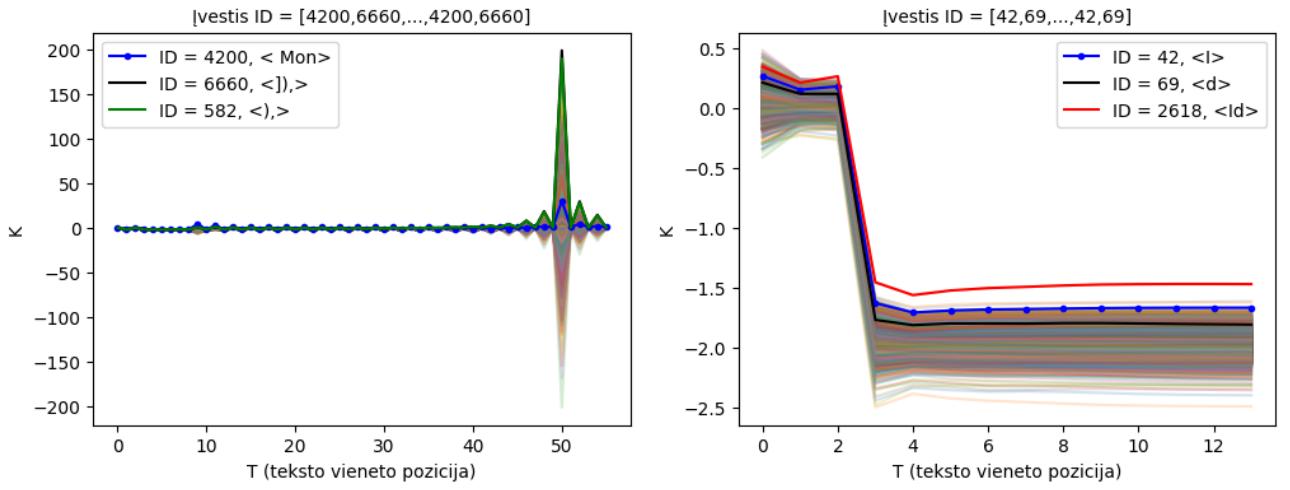
Išvestį K atvaizduoju grafike pagal (6), (7) formules. O P gaunu pagal (5), toliau analogiškai (7) formulei suskaidau į stulpelius ir pavaizduoju. Šio stulpelio T vertė atitinka tikimybę teksto vienetui būti išrinktam T išvesties pozicijoje.

6.4.2 Rezultatai

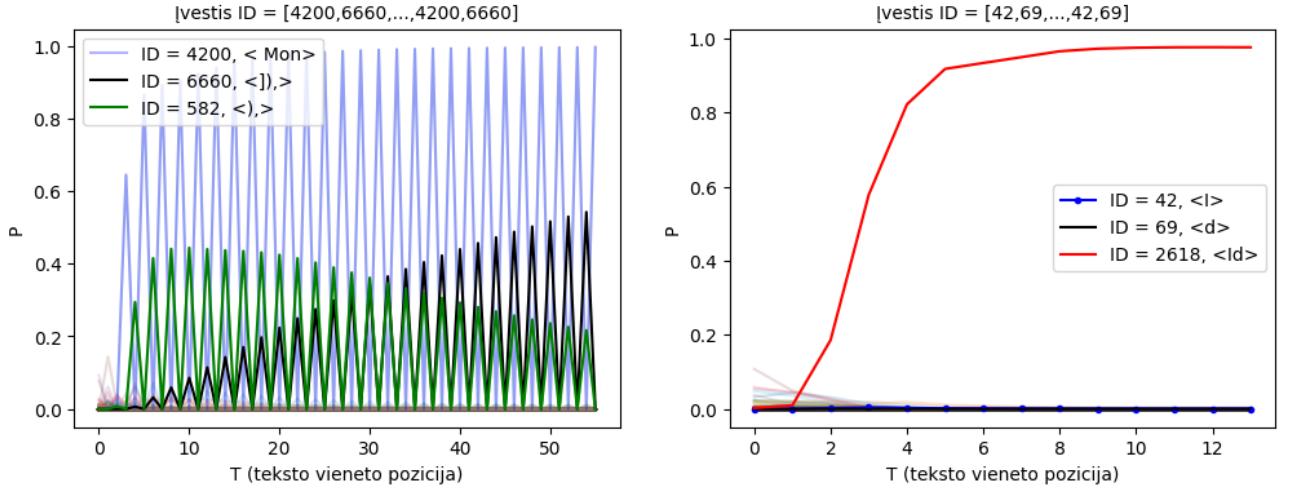
Kairėje esantis (19 pav.) sutampa su 2-a hipoteze. Pradžioje modelis nuspėja mažesnio ID ($582, \langle \rangle, \rangle$) teksto vienetą, kuris išties simboliškai labai artimas tikrajam ($6660, \langle \rangle, \rangle$) tad

tikėčiaus, jog pritaikius jidėties sluoksnį (angl. embedding) jie turėtų būti artimi vidinėje 768 dimensijų erdvėje. Dešinėje (19 pav.) vyksta kai kas įdomaus – greitai atrandamas teksto vienetas, kuris apima abu mažesnius teksto vienetus ir jo tikimybę staigiai užauga.

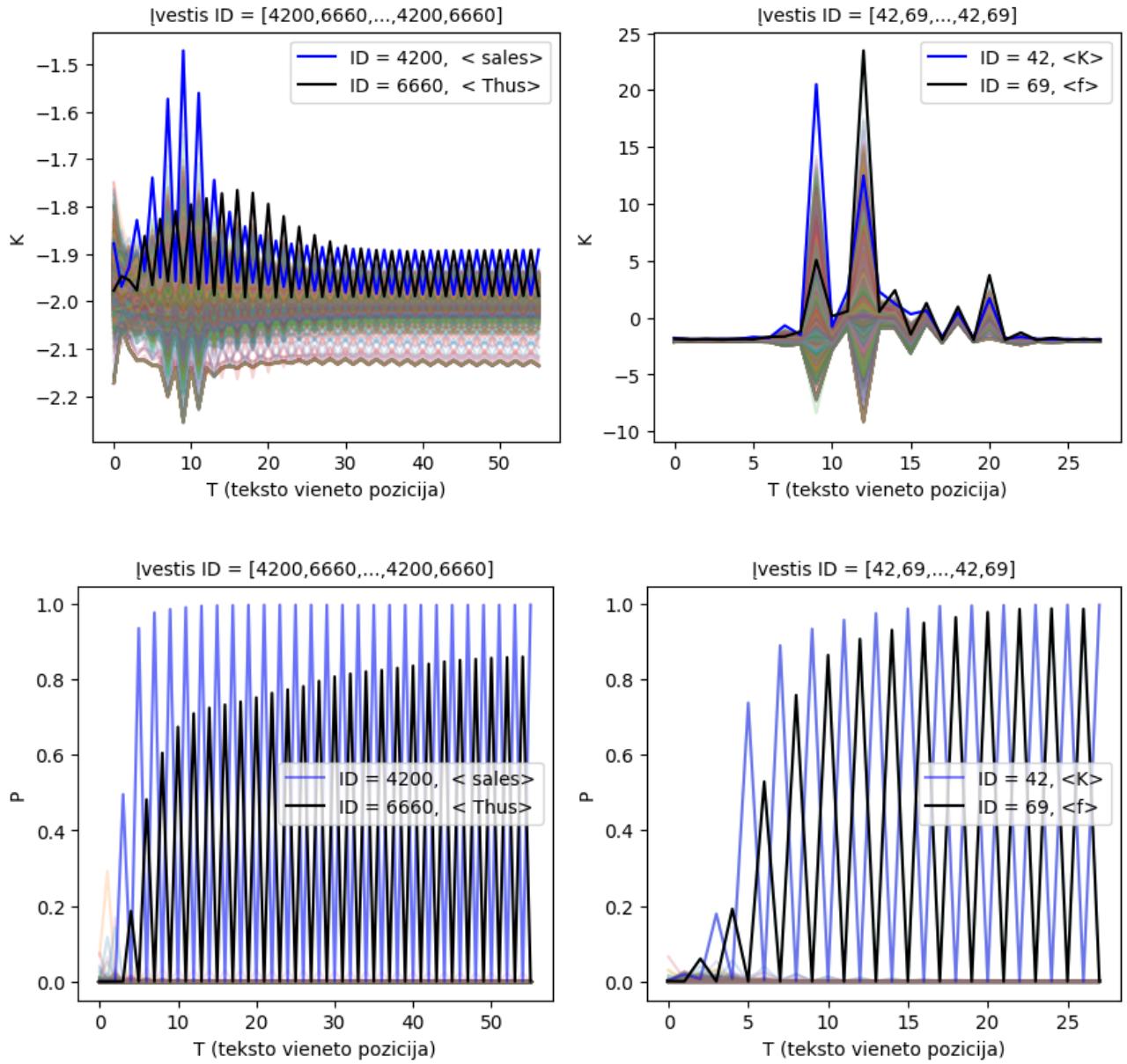
Paneigiant 3-a hipotezę atrodo, jog GPT-2 lėčiau pradeda atkartoti seką, negu Mamba modelis. Taip pat įdomu paprastesnių teksto vienetų sujungimas į vieną sudėtingesnį teksto vienetą, kurį atlieka abu GPT-2 ir Mamba modeliai. K grafikuose matome papildomą elgesį, kuris užmaskuojamas normalizuojant L į tikimybes. Pvz., K verčių išsiplėtimas ties $T = 50$ (kairėje 18 pav.), bei fazinis šuolis, kai $T \in \{2, 3\}$ (dešinėje 18 pav.).



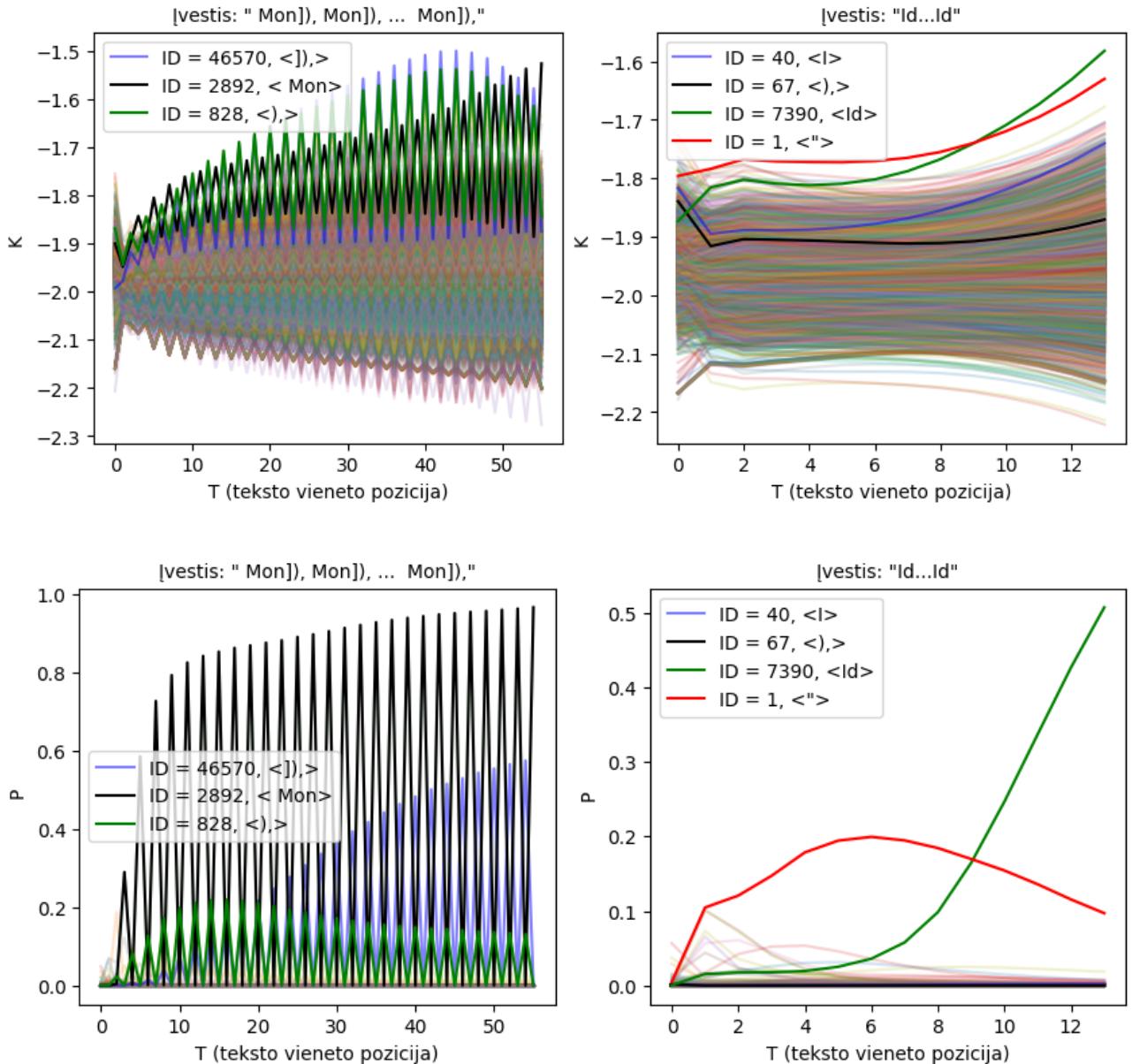
18 pav.: Išvestis K apibrėžta pagal (6) formulę. Modelis Mamba.



19 pav.: Viršutiniuose grafikuose išvestis K apibrėžta pagal (6) formulę, o apatiniuose TV tikimybės P . Kairėje įvestis $[4200, 6660, \dots] = [<\text{Mon}>, <],>, \dots]$, dešinėje $[42, 69, \dots] = [<\text{I}>, <\text{d}>, \dots]$. Modelis Mamba.



20 pav.: Viršutiniuose grafikuose išvestis K apibrėžta pagal (6) formulę, o apatiniuose TV tikimybės P . Kairėje išvestis $[4200, 6660, \dots] = [<\text{sales}>, <\text{Thus}>, \dots]$, dešinėje $[42, 69, \dots] = [<\text{K}>, <\text{f}>, \dots]$. Modelis GPT-2.



21 pav.: Viršutiniuose grafikuose išvestis K apibrėžta pagal (6) formulę, o apatiniuose TV tikimybės P . Kairėje išvestis $[4200, 6660, \dots] = [<\text{Mon}>, <]\), >, \dots]$, dešinėje $[40, 67, \dots] = [<\text{I}>, <\text{d}>, \dots]$. Modelis GPT-2.

7 Triukšmo įtaka

Šiame skyriuje dalinuosi pagrindiniais rezultatais, kai į Mamba vidų įterpiu triukšmą.

Apibrėžimas 8. *Triukšmas – vektorius, kurio kiekviena vertė yra atsitiktinis dydis.*

Grafikuose kiekviena vertė turi Gauso skirstinį, bet rezultatai pakartoti ir su pastoviu skirstiniu.

Apibrėžimas 9. *Triukšmo lygis – skaliaras iš kurio padaugintas triukšmas.*

7.1 Metodas

Pateikiu „When Mary[...]to“ (8) įvestį Mamba modeliui kartu įterpdamas skirtingo lygio triukšmą į Mamba modelio vidinį komponentą (priedas 8) gaunu išvestį ir fiksuoju ar modelis atspėjo $\langle \text{Mary} \rangle$ teksto vienetą ar ne. Taip gaunu priklausomybę parodančią skirtingo lygio triukšmo įtaką teisingam spējimui (22 pav.). O bandydamas gauti skirtingu sluoksnį jautrumą triukšmui naudojų tokį algoritmą:

1. Pateikiu įvestį (8) kartu į Mamba sluoksnį (8 pav.) įterpdamas triukšmą.
2. Fiksuoju ar atspėjo $\langle \text{Mary} \rangle$.
3. Pakeičiu triukšmo lygi.
4. Punktus 1, 2, 3 kartoju šimtą kartų.
5. Pakeičiu sluoksnį.

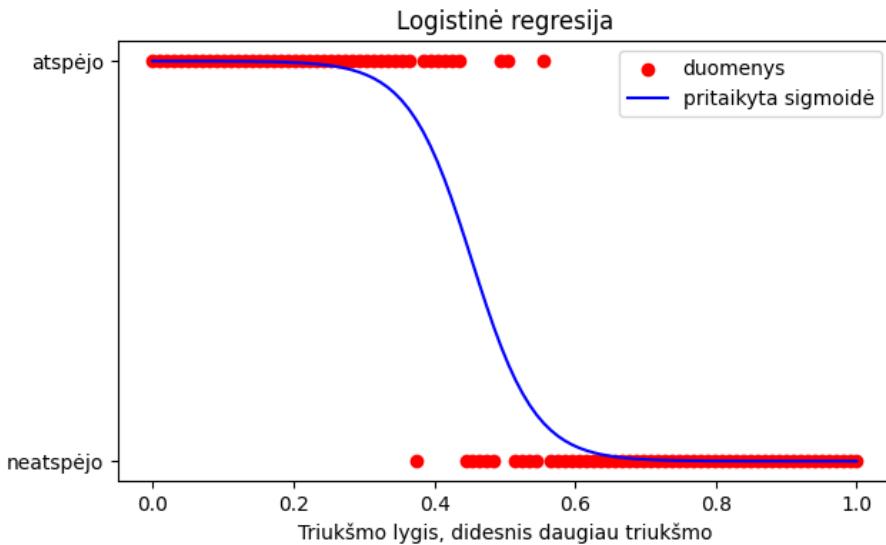
Taip gaunu priklausomybę parodanči triukšmo įtaką skirtingiem Mamba sluoksniam (23 pav.).

7.2 Rezultatai

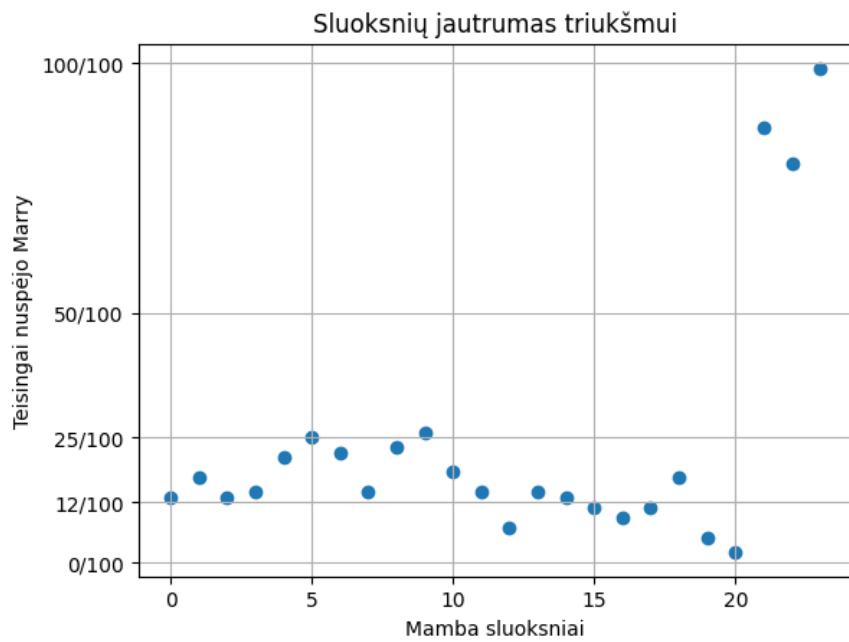
Pirmiausia radau, jog triukšmo įtaka spējimo rezultatams suteikia fazinį šuolį (22 pav.). Tai yra iki tam tikro triukšmo lygio modelis puikiai nuspėja Mamba, tada dar šiek tiek pakėlus triukšmo lygi modelis pradeda kartais nuspéti kartais nenuspēti, pakėlus triukšmo lygi dar labiau, modelis užtikrintai nenuspēja $\langle \text{Mary} \rangle$. Tokį fazinį šuolį vykstanti dėl skirtingo triukšmo lygio stebėjau beveik visose Mamba komponentuose, net ir pakeitus pačio triukšmo skirstinį iš Gauso į pastovų. Kitas paminėtinis rezultatas, tai, jog galiniai Mamba sluoksniai yra mažiau jautrūs triukšmui atliekant $\langle \text{Mary} \rangle$ nuspėjimą, negu pirmieji sluoksniai (23 pav.).

7.3 Eksperimento kritika

Tai, jog galiniai sluoksniai silpniai reaguoja į triukšmą, nebūtinai reiškia, jog jie yra mažiau svarbūs. Gal tiesiog sluoksnyje esančios vertės labiau išsiskaidžiusios, tad triukšmas joms turi



22 pav.: Taškas žymi Mamba teksto vieneto *< Mary>* atspéjimą y ašyje įterpus skirtingo lygio triukšmą x ašyje. Ties 0.45 matomas fazinis virsmas: tarp atspéjimo ir neatspéjimo. Įvestis „When Mary[...]to“ (8). Modelis Mamba.



23 pav.: X ašyje yra mamba sluoksniai (8 pav.). O Y ašyje varijuojant triukšmo lygį nuo 0 iki 1 modelio numatymas, jog po 1 saknio eis *< Mary>* teksto vienetas. Triukšmas pridedamas prie vidinės $h(t)$ būsenos (formulė (2), bei priede 8). Įvestis „When Mary[...]to“ (8). Modelis Mamba.

mažiau įtakos. Antra, nežinome ar tai Mamba architektūros savybė ar tik tirto „Mamba 130M“ savybė.

7.4 Tyrimo pratesimas

Šiame skyriuje radome, jog galiniai sluoksniai nėra tiek svarbūs $\langle \text{Mary} \rangle$ nuspėjimui. Kiekvienas sluoksnis sudarytas iš ≈ 5 milijonų parametru, tad mūsų rastas rezultatas yra makroskopinio pobūdžio. Analogiskai būtų galima atlikti labiau mikroskopinį tyrimą: pabandyti rasti atskirus parametrus, kurie būtini $\langle \text{Mary} \rangle$ nuspėjimui. Tai atlikti būtų galima taip: išsaugojant naudotą triukšmą stebeti, kuriuos modeliu parametrus paveikus modelis neatspėja, o kuriuos paveikus modeliui toliau pavyksta atspėti Pavyzdžiui, jei taikant triukšmą $= (1, 0, 0)$ modelis atspėjo, bet taikant triukšmą $(0, 0, 0)$ modelis neatspėjo, galime teigti, jog pirmasis parametras yra svarbus.

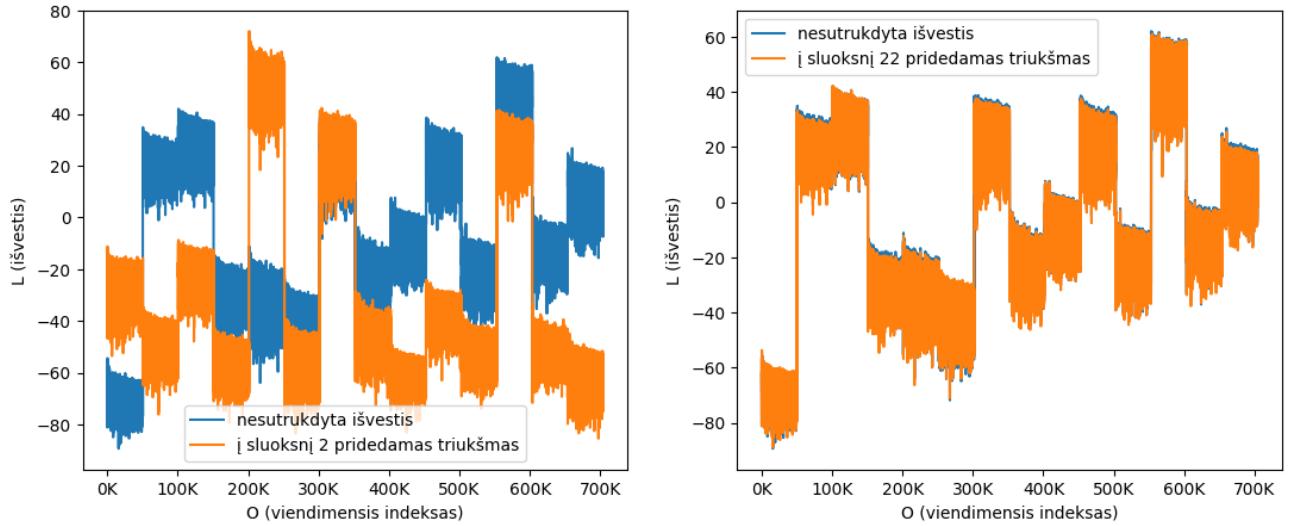
7.5 Matricinė išvestis į vektorių

Geresniai vizualizacijai L matricą perikiuoju į vektorių:

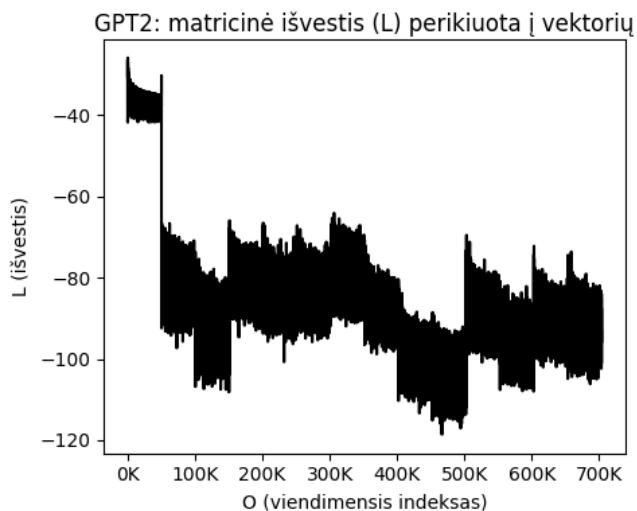
$$L = \begin{bmatrix} L_0^0 & L_1^0 & \cdots & L_{6393}^0 & \cdots & L_{50279}^0 \\ L_0^1 & L_1^1 & \cdots & L_{6393}^1 & \cdots & L_{50279}^1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ L_0^{14} & L_1^{14} & \cdots & L_{6393}^{14} & \cdots & L_{50279}^{14} \end{bmatrix} \xrightarrow{\text{perikiuoju}} \begin{bmatrix} L_0^0 & \dots & L_{50279}^0 & L_0^1 & \dots & L_{50279}^{14} \end{bmatrix}$$

Perikiuotą vektorių nupiešiu (24, 25 pav.). Pradinė jvestis vizualicijai yra 14 TV ilgio, tad tokios formos bus ir išvestis. Išties išvestis padalinta į 14 atskirų „blokelių“, kiekvienas blokelis yra apie ≈ 50 k ilgio, tad galime iškart daryti išvadą, jog blokelis atitinka L matrcios eilutę. Blokuotą išvestį stebime abiejuose GPT-2 ir Mamba. Kiekvienas blokelis yra neigiamai koreliuotas priklausomai nuo ID, tad abiejuose modeliuose lengvai ižvelgiame, jau numanomą rezultatą (10 pav.) mažesnio ID teksto vienetai yra labiau tikėtini. Tokia perikiuota L matricos vizualizacija leidžia aiškiau pamatyti triukšmo skirtumą skirtingiem sluoksniams (24 pav.) paveikus 2 sluoksnį mėlyni ir oranžiniai blokeliai smarkiai atsiskiria, tuo tarpu paveikus 22 sluoksnį mėlyni ir oranžiniai blokeliai praktiškai neatsiskiria. Tad galime daryti išvada, jog galinių sluoksnų nejautumas triukšmui nėra specifinis tik $\langle \text{Mary} \rangle$ nuspėjimo atveju, o bendra tiriamo Mamba savybė.

Mamba: matricinė išvestis (L) perikiuota į vektorių.



24 pav.: Matricinė L išvestis sudėliojama į vieną ilgą vektorių, kurio indeksas yra x ašis, o vertės y ašyje (atitinkančios nesunormuotas tikimybes). Tokio pat lygio triukšmu įsiterpiama į 2-ą mamba sluoksnį (pav. kairėje) ir į 22-ą sluoksnį (pav. dešinėje). Ivestis „When Mary ...“⁽⁸⁾.



25 pav.: GPT-2 išvestis analogiškai (24 pav.) eilutėmis „blokuota“, tačiau atstumai tarp „blokelių“ mažesni nei Mamba atveju.

8 Išvados ir rezultatai

Rezultatai

1. Tarp teksto vieneto ID, bei simbolių skaičiaus yra koreliacija $R \approx 0.77$. Abiejuose GPT-2, bei Mamba teksto vienetų generuotuojuose (angl. tokenizer).
2. Išvesties L eilutėje matoma neigama koreliacija tarp tikimybės ir ID.
3. Mamba K skirstiniuose rasta anomalija, kuri sudaryta iš įvairaus ilgio tarpų, bei naujos eilutės simbolių.
4. Mamba K skirstinio variacija didėja priklausomai nuo T , tuo tarpu GPT-2 nekinta.

Išvados

1. Generuoti tekštą lietuvių kalba brangiau, nei anglų kalba.
2. Mažesnio ID teksto vienetai yra labiau tikėtini, nei teksto vienetai su didesniu ID.
3. Triukšmas įterptas į modelį, modelio spėjimą veikia faziškai.
4. Triukšmas silpniau veikia galinius Mamba sluoksnius.
5. Mamba ir GPT-2 išvesties L eilutėje esančios vertės tarp savęs skiriasi kur kas mažiau, nei tarp skirtinį eilučių.
6. Kokybiskai skirtinės išvestys lemia kokybiskai skirtinės išvestis.
7. Modeliai sparčiai perprantą paprastą TV periodiškumą ir ji pakartoja. Jei pavyksta skirtinės TV sujungia į vieną TV didesniam efektyvumui.

Literatūros šaltiniai

- [1] Nick Cammarata ir kt. *Curve Detectors*. In: *Distill* 5.6 (2020 m. birželio 17 d.), e00024.003. ISSN: 2476-0757. DOI: [10.23915/distill.00024.003](https://doi.org/10.23915/distill.00024.003).
- [2] Ronen Eldan ir Yuanzhi Li. *TinyStories: How Small Can Language Models Be and Still Speak Coherent English?* In: *arXiv* (2023 m. gegužės 12 d.). DOI: [10.48550/arXiv.2305.07759](https://doi.org/10.48550/arXiv.2305.07759). eprint: [2305.07759](https://arxiv.org/abs/2305.07759).
- [3] Katja Grace ir kt. *Thousands of AI Authors on the Future of AI*. In: *arXiv* (2024 m. sausio 5 d.). DOI: [10.48550/arXiv.2401.02843](https://doi.org/10.48550/arXiv.2401.02843). eprint: [2401.02843](https://arxiv.org/abs/2401.02843).
- [4] Albert Gu ir Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. In: *arXiv* (2023 m. gruodžio 1 d.). DOI: [10.48550/arXiv.2312.00752](https://doi.org/10.48550/arXiv.2312.00752). eprint: [2312.00752](https://arxiv.org/abs/2312.00752).
- [5] Yue Liu ir kt. *VMamba: Visual State Space Model*. In: *arXiv preprint arXiv:2401.10166* (2024 m.).
- [6] Yecheng Jason Ma ir kt. *DrEureka: Language Model Guided Sim-To-Real Transfer*. In: (2024 m.).
- [7] Neel Nanda ir kt. *Progress measures for grokking via mechanistic interpretability*. In: *arXiv* (2023 m. sausio 12 d.). DOI: [10.48550/arXiv.2301.05217](https://doi.org/10.48550/arXiv.2301.05217). eprint: [2301.05217](https://arxiv.org/abs/2301.05217).
- [8] Chris Olah, Alexander Mordvintsev ir Ludwig Schubert. *Feature Visualization*. In: *Distill* 2.11 (2017 m. lapkričio 7 d.), e7. ISSN: 2476-0757. DOI: [10.23915/distill.00007](https://doi.org/10.23915/distill.00007).
- [9] Sara Reardon. *Artificial neurons compute faster than the human brain*. In: *Nature* (2018 m. sausio 26 d.). DOI: [10.1038/d41586-018-01290-0](https://doi.org/10.1038/d41586-018-01290-0). (Tikrinta 2024 04 24).
- [10] Bernardino Romera-Paredes ir kt. *Mathematical discoveries from program search with large language models*. In: *Nature* 625.7995 (2024 m. sausio mėn.), p. 468–475. ISSN: 1476-4687. DOI: [10.1038/s41586-023-06924-6](https://doi.org/10.1038/s41586-023-06924-6).
- [11] Alexander Matt Turner ir kt. *Optimal Policies Tend to Seek Power*. In: *arXiv* (2019 m. gruodžio 3 d.). DOI: [10.48550/arXiv.1912.01683](https://doi.org/10.48550/arXiv.1912.01683). eprint: [1912.01683](https://arxiv.org/abs/1912.01683).
- [12] Kevin Wang ir kt. *Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small*. In: *arXiv* (2022 m. lapkričio 1 d.). DOI: [10.48550/arXiv.2211.00593](https://doi.org/10.48550/arXiv.2211.00593). eprint: [2211.00593](https://arxiv.org/abs/2211.00593).

Kiti šaltiniai

- [13] 2019 m. Sausio 3 d. URL: <https://www.deeplearningbook.org/contents/mlp.html> (tikrinta 2024 05 03).

- [14] *AI Safety Fundamentals Course*. 2024 m. gegužės 3 d. URL: <https://course.aisafetyfundamentals.com/alignment>.
- [15] *ChatGPT-4 context lengths - API - OpenAI Developer Forum*. 2023 m. kovo 23 d. URL: <https://community.openai.com/t/chatgpt-4-context-lengths/114919> (tikrinta 2024 04 13).
- [16] Noam Chomsky, Ian Roberts ir Jeffrey Watumull. *Noam Chomsky: The False Promise of ChatGPT*. 2023 m. kovo 8 d. URL: <https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html> (tikrinta 2024 03 19).
- [17] Jaden Fiotto-Kaufman. *nnsight: The package for interpreting and manipulating the internals of deep learned models*. URL: <https://github.com/JadenFiotto-Kaufman/nnsight> (tikrinta 2024 04 17).
- [18] Lex Fridman. *Edward Gibson: Human Language, Psycholinguistics, Syntax, Grammar & LLMs | Lex Fridman Podcast #426*. Vaizdo įrašas. 2024 m. balandžio 17 d. URL: https://www.youtube.com/watch?v=F3Jd9GI6XqE&ab_channel=LexFridman (tikrinta 2024 04 25).
- [19] Lex Fridman. *Eliezer Yudkowsky: Dangers of AI and the End of Human Civilization | Lex Fridman Podcast #368*. Vaizdo įrašas. 2023 m. kovo 30 d. URL: https://www.youtube.com/watch?v=AaTRHFaaPG8&t=4s&ab_channel=LexFridman.
- [20] Lex Fridman. *Max Tegmark: The Case for Halting AI Development | Lex Fridman Podcast #371*. Vaizdo įrašas. 2023 m. balandžio 13 d. URL: https://www.youtube.com/watch?v=VcVfceTsDOA&ab_channel=LexFridman (tikrinta 2024 04 15).
- [21] Lex Fridman. *Stephen Wolfram: ChatGPT and the Nature of Truth, Reality & Computation | Lex Fridman Podcast #376*. Vaizdo įrašas. 2023 m. gegužės 9 d. URL: https://www.youtube.com/watch?v=PdE-waSx-d8&ab_channel=LexFridman (tikrinta 2024 04 25).
- [22] Lex Fridman. *Stephen Wolfram: ChatGPT and the Nature of Truth, Reality & Computation | Lex Fridman Podcast #376*. Vaizdo įrašas. 2023 m. gegužės 9 d. URL: https://www.youtube.com/watch?v=PdE-waSx-d8&t=4s&ab_channel=LexFridman (tikrinta 2024 04 25).
- [23] Maarten Grootendorst. *A Visual Guide to Mamba and State Space Models*. 2024 m. vasario 19 d. URL: <https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mamba-and-state> (tikrinta 2024 04 13).
- [24] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus ir Giroux, 2011 m.
- [25] Andrej Karpathy. *Let's build the GPT Tokenizer*. Vaizdo įrašas. 2024 m. URL: <https://youtu.be/zduSFxRajkE?si=AnvhTexndwx9nBbm> (tikrinta 2024 03 27).

- [26] Will Knight. *OpenAI's CEO Says the Age of Giant AI Models Is Already Over*. 2023 m. balandžio 17 d. (Tikrinta 2024 04 21).
- [27] Danielius Kundrotas. *Natūrali atranka, tai tik optimizacijos algoritmas?* Vaizdo įrašas. 2023 m. lapkričio 30 d. URL: https://www.youtube.com/watch?v=EgJHy35ulvY&t=4s&ab_channel=DanieliusKundrotas (tikrinta 2024 04 25).
- [28] Lex Fridman. *Transcript for Yann Lecun: Meta AI, Open Source, Limits of LLMs, AGI & the Future of AI | Lex Fridman Podcast #416 - Lex Fridman*. 2024 m. kovo mėn. URL: <https://lexfridman.com/yann-lecun-3-transcript> (tikrinta 2024 03 08).
- [29] Dwarkesh Patel. *Ilya Sutskever (OpenAI Chief Scientist) - Building AGI, Alignment, Spies, Microsoft, & Enlightenment*. Vaizdo įrašas. 2023 m. kovo 27 d. URL: https://www.youtube.com/watch?v=Yf1o0TQzry8&ab_channel=DwarkeshPatel (tikrinta 2024 04 25).
- [30] Dwarkesh Patel. *Mark Zuckerberg - Llama 3, 10B Models, Caesar Augustus, & 1GW Datacenters*. Vaizdo įrašas. 2024 m. balandžio 18 d. URL: https://www.youtube.com/watch?v=bc6uFV9CJGg&t=81s&ab_channel=DwarkeshPatel (tikrinta 2024 04 22).
- [31] Linas Petkevičius. *Mašininio ir giliojo mokymosi sąvokų žodynai*. 2024 m. URL: <https://github.com/linas-p/ML-AI-2-LT> (tikrinta 2024 03 27).
- [32] Aditya Raghunath. *GPT-4 Parameters Explained: How Many Parameters in GPT-4*. 2023 m. gruodžio 26 d. URL: <https://hix.ai/hub/chatgpt/gpt-4-parameters> (tikrinta 2024 03 27).
- [33] Maximilian Schreiner. *GPT-4 architecture, datasets, costs and more leaked*. 2023 m. liepos 11 d. URL: <https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked> (tikrinta 2024 03 27).
- [34] *The Mamba Explosion*. 2024 m. balandžio 24 d. URL: <https://www.statespace.info/the-mamba-explosion> (tikrinta 2024 04 24).
- [35] John Vervaeke. *AI Sages and the Ethical Frontier: Exploring Human Values, Embodiment, and Spiritual Realms*. Vaizdo įrašas. 2024 m. kovo 15 d. URL: https://www.youtube.com/watch?v=6pU1clFG_rg&ab_channel=JohnVervaeke (tikrinta 2024 04 25).

Priedas 1: GPT-2 struktūra

Pateikiu darbe naudoto GPT-2 struktūrą.

```

1 GPT - 2LMHeadModel (
2     (transformer): GPT - 2Model (
3         (wte): Embedding (50257, 768)
4         (wpe): Embedding (1024, 768)
5         (drop): Dropout (p = 0.1, inplace = False)
6         (h): ModuleList (
7             (0 - 11): 12 x GPT - 2Block (

```

```
8     (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
9     (attn): GPT-2Attention(
10         (c_attn): Conv1D()
11         (c_proj): Conv1D()
12         (attn_dropout): Dropout(p=0.1, inplace=False)
13         (resid_dropout): Dropout(p=0.1, inplace=False)
14     )
15     (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
16     (mlp): GPT-2MLP(
17         (c_fc): Conv1D()
18         (c_proj): Conv1D()
19         (act): NewGELUActivation()
20         (dropout): Dropout(p=0.1, inplace=False)
21     )
22 )
23 )
24     (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
25 )
26     (lm_head): Linear(in_features=768, out_features=50257, bias=False)
27 )
```

Priedas 2: Mamba struktūra

Pateikiu darbe naudotos Mamba struktūrą.

```

1 MambaLMHeadModel(
2   (backbone): MixerModel(
3     (embedding): Embedding(50280, 768)
4     (layers): ModuleList(
5       (0-23): 24 x Block(
6         (mixer): MambaModuleInterp(
7           (in_proj): Linear(in_features=768, out_features=3072, bias=False)
8           (conv1d): Conv1d(1536, 1536, kernel_size=(4,), stride=(1,), padding
9             =(3,), groups=1536)
10          (act): SiLU()
11          (x_proj): Linear(in_features=1536, out_features=80, bias=False)
12          (dt_proj): Linear(in_features=48, out_features=1536, bias=True)
13          (out_proj): Linear(in_features=1536, out_features=768, bias=False)
14          (dt): WrapperModule()
15          (B): WrapperModule()
16          (C): WrapperModule()
17          (ssm): SSM(
18            (discA): DiscA()
19            (discB): DiscB()
20            (hx): Hx(
21              (bx): Bx()
22              (ah): Ah()
23            )
24            (yh): Yh()
25          )
26          (delta_softplus): Softplus(beta=1, threshold=20)
27        )
28        (norm): RMSNorm()
29      )

```

```
29     )
30     (norm_f): RMSNorm()
31   )
32   lm_head): Linear(in_features=768, out_features=50280, bias=False)
33 )
```