LLM KLASIFIKATORS: ZIŅOJUMA VALIDĀCIJA UZ RUPJĪBAS PAZĪMĒM

LLM CLASSIFIER: MESSAGE VALIDATION ON PROFANITY

Autori: **Aleksandrs Daniels Lebeds**, e-pasts: dl22040@edu.rta.lv
Egīls Kivlis, e-pasts: ek22140@edu.rta.lv
Zinātniskais vadītājs: **Sergejs Kodors, Dr.sc.ing.**, e-pasts: sergejs.kodors@rta.lv
Rēzeknes Tehnoloģiju akadēmija
Atbrīvošanas aleja 155, Rēzekne, Latvija

Abstract: *The aim of the given scientific paper "Message validation on profanity using LLM" to train LLM to detect profanity in user messages. During the research, the authors created a dataset of 30 phrases (15 good, 15 bad), which were then used to train LLM using AI platform Cohera and evaluated the recognition accuracy. The authors achieved accuracy equal to 70%.*

Keywords: *Dataset, ChatGPT 3.5, Cohera, LLM.*

Ievads

Pāris pēdējo gadu laikā lielie valodas modeļi (LLM) ir attīstījušies no jaunizveidotām tehnoloģijām līdz praktiski lietojamiem rīkiem. Lielais valodas modelis ir sava veida mašīnmācīšanās atzars, kas paredzēts apmācīt neirontīklus, kas spēj apstrādāt dabisko valodu (*NLP*). [1] Lielie valodas modeļi, piemēram, GPT-3, tiešām ir revolucionējoši dažādās jomās, ieskaitot tīmekļa izstrādi. Šie modeļi ir spējīgi saprast, klasificēt, apkopot, pārstrādāt (pārveidot), meklēt un ģenerēt cilvēkam līdzīgu tekstu pamatojoties uz iegūto informāciju. Kā arī šie modeļi ir ļoti svarīgi mūsdienu mākslīgā intelekta algoritmiem, kas spēj apstrādāt un radīt teksta saturu, un tie ir plaši izmantoti dažādās jomās, piemēram, valodu tulkošanā, teksta klasifikācijā, jūtu analīzē, teksta ģenerēšanā un jautājumu atbildēšanā. Šie modeļi tiek apmācīti ar lielu datu apjomu no dažādiem avotiem, daži no tiem satur pat simtiem miljardu tekstvienību jeb tokenu (*tokens*). Lielo valodas modeļu gadījumā datu apjomu mēra ar tekstvienībām.

Tekstvienība ir teikuma unikāla daļa, tā varbūt vārds, burts, burtu kombinācija, vārdu kopa vai baitu secība. Moderni lielie valodas modeļi ir balstīti uz transformera (*transformer*) pielietošanas, kas bija anonsēts 2017. gadā. [2] Lielie valodas modeļi būtiski paplašina iespējas, ko dators var darīt ar tekstu. Pēdējo gadu populārākie lielie valodas modeļi ir *GPT-3.5, GPT-4, Gemini, LLaMA, Falcon, Cohere, PaLM, Cloude v1*. [3] Pateicoties spējai ģenerēt loģisku tekstu neatšķiramo no cilvēka rakstīta varianta, lielie valodas modeli tagad tiek izmantoti ļoti plaši, sākot no satura izveides līdz klientu apkalpošanas tērzēšanas robotiem.

Pētījuma mērķis: apmācīt LLM klasifikatoru atpazīt rupjību lietotāju ziņojumos.

Uzdevumi:

1) sagatavot datu kopu;
2) apmācīt LLM;
3) novērtēt atpazīšanas precizitāti.

Materiāli un metodes

Lai izveidotu datu kopu ar piemēriem un veiktu klasifikācijas testu, par pamatu tika paņemti 15 lamu vārdi un 15 labi vārdi, kas tika izmantoti teikumu ģenerēšanai ar *ChatGPT 3.5* [4-5] palīdzību, lai radītu gan labas, gan sliktas frāzes. Tad darba autori izmantoja šo datu kopu, kas sastāvēja no 15 labām un 15 sliktām frāzēm, lai veiktu klasifikācijas testu un aprēķinātu precizitāti (*Accuracy*) ar kļūdu matricas (*confusion matrix*) palīdzību. Dotajā eksperimentā tika izmantota mākslīgā intelekta apmācības rīku *Cohere* [6],

https://doi.org/10.17770/het2024.28.8262

lai veiktu teksta analīzi un apmācītu lielo valodas modeli. Saskaņā ar *Cohere* prasībām, lai lielo valodas modeli pielāgotu (*customization*) individuālajiem klasifikācijas mērķiem ir nepieciešams to sagatavot, izpildot vairākus soļus. Pirmkārt, ir jāizstrādā un jāsagatavo apmācības dati, kas ietver piemērus un etiķetes jeb klasifikācijas marķierus, kas atbilst vēlamajam uzdevumam. Tad ir jāizvēlas piemērots lielā valodas modeļa arhitektūras variantu, kas atbilst uzdevumam, un jāveic pielāgošanas procesa konfigurācija. Pēc tam notiek pats pielāgošanas process, kas ietver modeļa *featuru* izguves un svaru pielāgošanu, izmantojot apmācības datus. Kad pielāgošana ir pabeigta, modelis tiek pārbaudīts un novērtēts izmantojot testa datus, lai nodrošinātu tā efektivitāti un pareizību.

Rezultāti un diskusija

Autoru izpildīts eksperiments sastāvēja no šādiem posmiem:

1. Sagatavot datu kopu ar *ChatGPT;*
2. Apmācīt LLM klasifikatoru pielietojot *Cohere*.

Lai sasniegtu izvirzītus uzdevumus, tika sagatavota un apkopota datu kopa ar autoru izveidotiem piemēriem, ietverot gan labus, gan sliktus vārdus:

1.tabula. Datu kopa ar LLM apmācības piemēriem

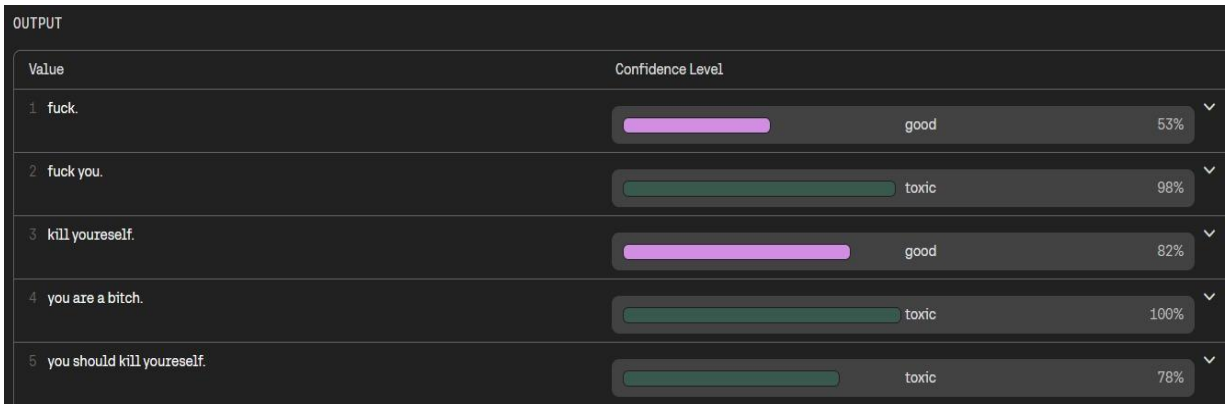
Teksts	***Klase***
Bridge construction is progressing well.	Labi
You're such a dumbass.	Slikti
I love eating asparagus.	Labi
That's a load of bullshit.	Slikti
The clock is ticking.	Labi
You're such a jerk.	Slikti

Lai izmantotu šo datu kopu apmācībai un testēšanai, *Cohere* piedāvā izvēlēties LLM arhitektūru. Klasifikācijai izmanto tikai *Encoder* daļu. [8] Šāda veida modeļi varētu būt noderīgi, piemēram, sociālajos medijos vai citās platformās, lai automātiski filtrētu un klasificētu tekstuālu saturu atbilstoši noteiktām prasībām. Šajā datu kopā ir 30 frāzes, kurās ir 15 labas frāzes un 15 sliktas frāzes. Izrietot pēc datu kopas ir iespējams veikt klasifikācijas kļūdu matrica un aprēķināt precizitāti (*Accuracy*).

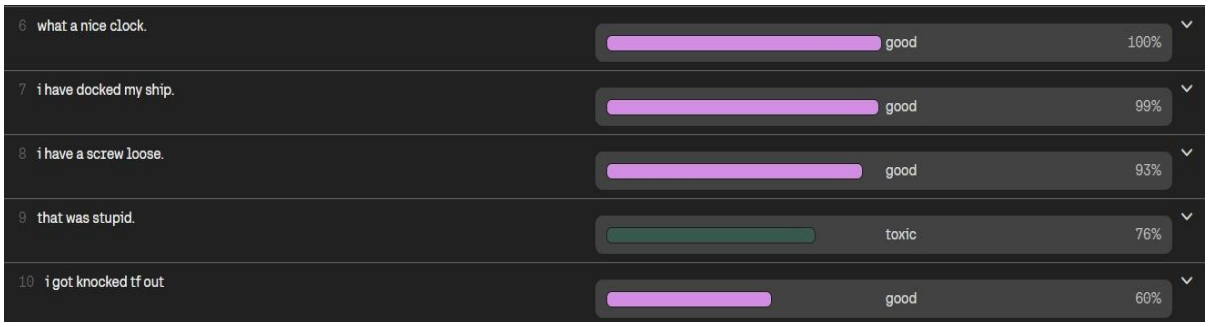
Autori izmantoja *playground* opciju *Cohere* rīkā, kas atļauj veikt LLM apmācību bez koda rakstīšanas. Ar klasifikācijas opciju ātri un vienkārši tika apmācīts LLM atpazīt jeb validēt frāzes pret ziņojumiem ar rupjībām.

Validācija: lai novērtētu modeļa veiktspēju un novērstu pārapmācību (*overfitting*), tas tiek validēts, izmantojot atsevišķu datu kopu, kas nav izmantota apmācībai.

Testēšana: kad modeļa izstrādes posms ir pabeigts un tika izvērtēta arī tā veiktspēja validācijas kopā, tas tiek testēts, izmantojot pilnīgi atsevišķu datu kopu, lai novērtētu tā ģenerālo veiktspēju.



1.attēls. LLM apmācīšana, sliktos frāžu ievadīšana



2.attēls. LLM apmācīšana, labos frāžu izvadīšana

Pēc datu kopu frāžu ievadīšanas un izvadīšanas ar *Cohera* programmas palīdzību, tiek veidota klasifikācijas tabula, ar klasifikācijas rezultātiem. Ņemot vērā izlasi, kurā tika paņemtas 10 frāzes, kurās 5 ir pozitīvas un 5 ir negatīvās, tika izveidota šāda klasifikācijas tabula (skat. 2. tabulu):

2.tabula. Klasifikācijas kļūdu matricas rezultāti

Frāzes numurs	1	2	3	4	5	6	7	8	9	10
Reālā klasifikācija	1	1	1	1	1	0	0	0	0	0
Paredzētā klasifikācija	0	1	0	1	1	0	0	0	1	0
rezultāts	*FN*	*TP*	*FN*	*TP*	*TP*	*TN*	*TN*	*TN*	*FP*	*TN*

- *True Positive* (*TP*): gadījums, kad modelis pareizi atpazīst pozitīvo klasi.
- *True Negative* (*TN*): gadījums, kad modelis pareizi atpazīst negatīvo klasi.
- *False Positive* (*FP*): gadījums, kad modelis nepareizi identificē negatīvo klasi kā pozitīvo (pazīstot kļūdu, kad nav).
- *False Negative* (*FN*): gadījums, kad modelis nepareizi identificē pozitīvo klasi kā negatīvo (pazīstot kļūdu, kad ir).

3.tabula. Klasifikācijas kļūdu tabula

		Paredzētā klasifikācija	
	5+5=10	4 slikti	6 labi
Reālā klasifikācija	5 slikti	3	2
	5 labi	1	4

Kā rezultātā tika pielietoti 10 varianti, pēc rezultātiem ir iespējams spriest, ka reālā klasifikācija ir 5 = 5, bet paredzētā klasifikācija 4 = 6. Pēc tabulas datiem, autoriem ir izveidojies:

- *TP* = 3
- *TN* = 4
- *FP* = 1
- *FN* = 2

Lai aprēķinātu precizitāti (*accuracy*), ir jāsaskaita pareizi klasificētie gadījumi (*TP* un *TN*) un jāsadala tos ar kopējo skaitu: *Accuracy* = (*TP* + *TN*) / (*TP* + *TN* + *FP* + *FN*).

Accuracy = (3 + 4) / (3 + 4 + 1 + 2) = 7 / 10 = 0.7, tādējādi precizitāte ir 0.7 jeb 70%.

Šo LLM klasifikatoru var integrēt tīmekļā lietotnē, pielietojot *Cohere API*. Ir iespējams pārbaudīt *Cohere API* integrāciju, lai pārliecinātos, ka tā darbojas pareizi un sniedz gaidītus rezultātus. Ja nepieciešams, var veikt pielāgojumus vai uzlabojumus, lai optimizētu integrācijas veiktspēju un precizitāti.

Secinājumi

Apmācīta LLM klasifikatora precizitāte sasniedza 70%. Tas nozīmē, ka tas ir spējīgs pietiekoši pareizi klasificēt tekstu, aptuveni 7 no 10 gadījumiem. Kļūdu matrica sniedz ieskatu par modeļa veiktspēju un to, kāda veida kļūdās tiek veidotas. Kā arī autoru novērojums, ka modelis precīzāk atpazīst garākas frāzes nekā īsās. Tā izceļ gan pareizi identificētās ("*True*") gan nepareizi identificētās ("*False*") pozitīvās un negatīvās kategorijas. Tā kā autoru datu kopā ir vienāds skaits gan labu, gan sliktu piemēru, varam uzskatīt, ka abu klašu vērtējums ir līdzsvarots. Tomēr joprojām ir svarīgi ņemt vērā modeļa spēju pareizi identificēt svarīgāko klasi, kas varētu būt atkarīga no konkrētā scenārija vai uzdevuma. Jāņem vērā, ka šī pētījuma rezultāti ir atkarīgi no izmantotās datu kopas un izmēra. Lai gan 70% precizitāte var izskatīties laba, tā varbūt atkarīga no dažādiem faktoriem, piemēram, datu kvalitātes, izmēra un sarežģītības. Šo rezultātu varētu uzlabot, izmantojot papildu apmācības piemērus, pielāgojot modeļa *hiperparametrus* vai izvēloties citus algoritmus.

Summary

In recent years, large language models (LLMs) have become practical technologies, enabling natural language processing (NLP). LLMs, such as GPT-3, have introduced revolutionary advancements in various fields, including web development. These models are capable of understanding, classifying, aggregating, processing, searching, and generating human-like text, thus creating a strong connection between customers and businesses. The research aim is to train LLM to perform rudeness classification in user messages. The study used both "positive" and "negative" words, and the dataset was created with the assistance of ChatGPT 3.5. Additionally, the study used the Cohera artificial intelligence training platform to perform text analysis and train LLMs. Validation and testing results showed that the average accuracy is 70%. These results can be integrated into web application using Cohere API, with the ability to make adjustments and improvements to optimize integration performance and accuracy.

Literatūra

1. What Is A Large Language Model (LLM)? A Complete Guide. [tiešsaiste 22.03.2024.]. Pieejas veids: https://www.hostinger.com/tutorials/large-language-models?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=9075644&gad_source=1&gclid=Cj0KCQjw2PSvBhDjARIsAKc2cgOQyPS0JVEpWalFc4hcqEho1QYLmIAnPmBDpN7lzdDJlTpfP6A3bLsaAok5EALw_wcB
2. Attention Is All You Need. [tiešsaiste 08.04.2024.]. Pieejams: 1706.03762.pdf (arxiv.org)
3. Best Large Language Models for 2024 and How to Choose the Right One for Your Site. [tieššaiste 22.03.2024.]. Pieejas veids: https://www.hostinger.com/tutorials/large-language-

models?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=9075644&gad_source=1&gclid=Cj0KCQjw2PSvBhDjARIsAKc2cgOQyPS0JVEpWalFc4hcqEho1QYLmIAnPmBDpN7lzdDJlTpfP6A3bLsaAok5EALw_wcB#1_GPT_35
4. Datu kopa ChatGPT3.5 [tiešsaiste 22.03.2024.].Pieejas veids: https://chat.openai.com/
5. Make database. [tiešsaiste 03.04.2024.]. Pieejams: https://chat.openai.com/auth/login?next=%2Fc%2F1444a459-dfb0-4b8d-b378-b2aece7c6413
6. Introduction to Large Language Models. [tiešsaiste 29.03.2024.]. Pieejams: https://docs.cohere.com/docs/introduction-to-large-language-models
7. LLMs – Model Architectures and Pre-Training Objectives. [tiešsaiste 11.04.2024.]. Pieejams: https://ritikjain51.medium.com/llms-model-architectures-and-pre-training-objectives-39c4543edef0