თბილისის თავისუფალი უნივერსიტეტი

მათემატიკის, კომპიუტერული მეცნიერების და ინჟინერიის სკოლა
(MACS[E])

კომპიუტერული მეცნიერებისა და მათემატიკის პროგრამა

შოშიაშვილი საბა

პაიჭაძე ლუკა

კლდიაშვილი გიორგი

საბაკალავრო პროექტი

“ქართული ენის კორპუსი”

ხელმძღვანელი: სტურუა საბა

თბილისი
2024

ანოტაცია

დღესდღეობით კომპიუტერულ მეცნიერებაში ყველაზე აქტუალური თემაა ხელოვნური ინტელექტი. მრავალი დიდი კომპანია უშველებელ რესურსს დებს მანქანური სწავლების სხვადასხვა ალგორითმისა და მოდელის შექმნაში, განვითარებასა და შესწავლაში. ამას სჭირდება უფრო და უფრო მეტი მონაცემი. ბოლო დროს განსაკუთრებით აქტუალური გახდა ე.წ. LLM-ები (დიდი ენის მოდელები), რომლებმაც უკვე კომპიუტერული მეცნიერების სფეროს გარეთაც ფართო გამოყენება ჰპოვა. მათ შორის ყველაზე ცნობილი მოდელები, როგორიცაა ChatGPT, Llama და Claude, ფართოდ გამოიყენება მრავალი ადამიანის მიერ და სრულიად გასაგებ დიალოგს ამყარებს მის მომხმარებლებთან მრავალ ენაზე. აღსანიშნავია, რომ ეს მიღწევები უმეტესწილად ეხება მსოფლიოში გავრცელებულ ენებს, როგორიცაა ინგლისური, ესპანური, რუსული და ა.შ., რომლებისთვისაც მოიპოვება დიდი ენობრივი რესურსი. ნაკლებად გავრცელებულ ენებს, რომლებისთვისაც არც თუ ისე ბევრი ხარისხიანი ტექსტი მოიპოვება, ეწოდება დაბალრესურსიანი ენები. ქართული ენაც ამ კატეგორიას განეკუთვნება. რესურსების ნაკლებობა ზღუდავს ქართულენოვანი მოდელების განვითარებას, რის გამოც დღესდღეობით NLP (ბუნებრივი ენის დამუშავება) სფეროში კარგი შედეგები არ გვაქვს ქართულ ენაზე.

ენის კორპუსი არის ტექსტურ მონაცემთა დიდი რაოდენობა. ენის მოდელის შესაქმნელად საჭიროა ისეთი კორპუსის ქონა, რომელზეც მოდელის გაწვრთნა შესაძლებელი იქნება. ენის კორპუსი უნდა იყოს მოცულობითი და უნდა შეიცავდეს ხარისხიან ტექსტებს, რომელიც შინაარსით მრავალფეროვანია. ამგვარი კორპუსის შექმნისას ვხვდებით გამოწვევებს, როგორიცაა მონაცემთა წყაროების პოვნა, ამ წყაროებიდან არარელევანტური, არასაჭირო მონაცემების გაფილტვრა, და მონაცემების დასამუშავებელი ეფექტური ალგორითმების შემუშავება.

ჩვენი პროექტის მიზანია ამ პრობლემების გადალახვა და დიდი მოცულობის მქონე, ხარისხიანი კორპუსის შექმნა, რომელიც გამოყენებადი უნდა იყოს ქართულ NLP სფეროში.

თავდაპირველად ჩვენ განვიხილეთ რამდენიმე განხვავებული წყარო და საბოლოოდ ამოვარჩიეთ Common Crawl-ის საჯაროდ ხელმისაწვდომი არქივი, რომელიც მოიცავს

წლების განმავლობაში გაშვებული Crawl-ების შეგროვებულ ვებ-გვერდებს. ეს არქივი, მისი დიდი მოცულობის მიუხედავად, საჭიროებდა მნიშვნელოვან დამუშავებას, რათა მრავალენოვანი მონაცემებიდან შინაარსიანი ქართული ტექსტები მიგვეღო. ჩვენ ამისთვის რამდენიმე მნიშვნელოვან ნაბიჯს ვიყენებდით, რომლებიცაა: ტექსტის ექსტრაქცია, გაფილტვრა და დედუპლიკაცია. ამ ნაბიჯებსა და მათ იმპლემენტაციას დეტალურად ამ ნაშრომში განვიხილავთ.

ჩვენ შევქმენით ქართული ენის კორპუსი, რომელიც დიდი მოცულობითა და ხარისხით გამოირჩევა და ამ დროისთვის იქნება ყველაზე დიდი კორპუსი ქართული ენისთვის. ჩვენ იმედი გვაქვს, რომ ეს კორპუსი მნიშვნელოვან შედეგს იქონიებს ქართულ ენაზე ბუნებრივი ენის დამუშავების სფეროში. დიდი მოცულობის ტექსტური მონაცემების ქონა შესაძლებელს გახდის LLM-ებისა და სხვა NLP სისტემების შექმნას ისეთი დაბალრესურსიანი ენისთვის, როგორიც ქართულია.

Abstract

Nowadays, artificial intelligence is one of the most prominent topics in computer science. Many large companies are putting massive resources into creating, improving and researching various machine learning algorithms and models. This requires more and more data. Lately, LLMs(Large Language Models) have gained particular prominence and found wide usage even outside of the field of computer science. Some of the most notable models, such as ChatGPT, Llama and Claude, are being used widely by various people and can hold coherent dialogues in multiple languages with its users. It is worth noting, however, that it's the most widely used languages such as English, Spanish, Russian, etc., that benefit most from these advancements, the reason being that these languages have a lot of linguistic resources. Less widely used languages, for which there's not a lot of high-quality texts available, are called low-resource languages. Georgian language belongs to this category. The lack of resources limits the advancement of Georgian language models, which is why nowadays we don't have great results for Georgian language in the field of NLP(Natural Language Processing).

A language corpus is a large set of textual data. In order to create a language model, we need to have a corpus on which the model can be trained. Such a corpus needs to be large in volume and contain high-quality texts which are diverse in content. In creating a large corpus, we encounter challenges such as finding sources of data, filtering out irrelevant, unnecessary data from these sources, and implementing effective algorithms for processing the data.

The goal of our project is to overcome these challenges and create a large, high-quality corpus, which can be used in the Georgian NLP field.

We considered several different sources of data and finally settled with the publicly available Common Crawl dataset, which contains webpage contents collected over years of crawls. This dataset, despite its large volume, needed a significant amount of processing in order to extract meaningful Georgian texts from multilingual data. For this we utilized several key steps, such as: text extraction, filtering and deduplication. These steps and their implementation will be discussed in detail in this paper.

We have created a Georgian Language Corpus, high in volume and quality, which will be the largest corpus for Georgian language to date. We hope that this corpus will have a big impact on the Georgian language in the field of NLP. The availability of large textual data will make creation of LLMs and other NLP systems possible for a low-resource language such as Georgian.

სარჩევი

საილუსტრაციო მასალა

შესავალი

თავიდანვე ვიყავით დაინტერესებული ბუნებრივი ენის დამუშავების პროექტის გაკეთებით. განსაკუთრებით გვაინტერესებდა როგორ კეთდება დიდი ენობრივი მოდელები (LLM) და რამდენად შესაძლებელია ასეთი რამის გაკეთება ქართული ენისათვის, ვინაიდან ვამჩნევდით ამ მხრივ მიღწევების ნაკლებობას. პირველ რიგში, აშკარა პრობლემა იყო, რომ LLM-ის გაწვრთნას სჭირდება ძალიან დიდი გამოთვლითი რესურსი და რომც გვქონოდა ეს რესურსი, დაგვჭირდებოდა ტექსტური მონაცემების დიდი მოცულობა, რომელიც ასევე თითქმის არ არსებობს ქართული ენისთვის. აქედან გამომდინარე, გადავწყვიტეთ, რომ შეგვეგროვებინა ეს მონაცემები, შეძლებისდაგვარად ჩვენითაც გაგვეწვრთნა მოდელი, მაგრამ, რაც მთავარია, საჯაროდ ხელმისაწვდომი გაგვეხადა ჩვენი მონაცემები, რათა სამომავლოდ ქართული ენისთვის ენობრივი მოდელის შექმნა უფრო მარტივი ყოფილიყო. მაშასადამე, ჩვენი პროექტის მთავარი მიზანი იყო ქართული ენის კორპუსის შექმნა.

NLP სფეროში ენის კორპუსი არის ფუნდამენტური რესურსი. კორპუსი არის ტექსტების დიდი და სტრუქტურირებული კოლექცია, რომელიც სისტემურად არის შეგროვებული ენის მოდელების შესაქმნელად, კვლევისა თუ სხვა მიზნისთვის. მაღალი ხარისხის კორპუსების არსებობა შესაძლებელს ხდის ისეთი ენის ტექნოლოგიების განვითარებას, როგორიცაა დიდი ენის მოდელები, მანქანური თარგმნა, ხმის აღქმა და მისთ.

ისეთი დაბალრესურსიანი ენებისთვის, როგორიც ქართულია, ამგვარი დიდი ზომის კორპუსის არარსებობა სერიოზული გამოწვევაა.

ჩვენი კორპუსის მთავარი ინსპირაცია იყო 2023 წელს გამოქვეყნებული დატასეტი The RefinedWeb Dataset. ამ სტატიაში აჩვენეს, რომ ინტერნეტიდან შეგროვილი ტექსტური მასალის მკაცრად გაფილტვრითა და შემდეგ დამუშავებით შესაძლებელია ექსპერტების მიერ ხელით ამორჩეული დატასეტის ხარისხთან მიახლოება(Penedo et al., 2023). ჩვენ ვეცადეთ შეგვემუშავებინა პროცესებისა და ალგორითმების ისეთი მიმდევრობა, რომელიც ვებგვერდების გაუსუფთავებელ ტექსტს გარდაქმნიდა სრულყოფილ, აზრიან ქართულ ტექსტებად, ისევე როგორც ზემოთხსენებულ დატასეტზე გააკეთეს ინგლისური ენისთვის. ამისათვის საკმაოდ დიდი დრო და რესურსი იყო საჭირო, რისთვისაც გამოვიყენეთ აზურისა (Azure) და ორაკლის (Oracle) ქლაუდზე არსებული ვირტუალური მანქანები, რომლებსაც გაძმოწერის მაღალი სიჩქარე, დიდი მეხსიერება და დიდი ადგილი ჰქონდათ. ტექსტების დასამუშავებლად და გასასუფთავებლად გამოვიყენეთ ალგორითმები, რომლის შთაგონებაც Scaling Language Models: Methods, Analysis & Insights from Training Gopher სტატია იყო(Rae et al., 2021).

ამ პროექტს რამდენიმე რეალური შეზღუდვა აქვს. პირველ რიგში, ტექსტი მოგვაქვს ინტერნეტიდან, რაც ნიშნავს, რომ ჩვენი კორპუსის ზომა პირდაპირ კავშირშია ინტერნეტში არსებული ქართულენოვანი ვებსაიტების რაოდენობასთან. ქართული ენა მოსაუბრეთა რაოდენობით ბევრად პატარაა სხვა ენებთან შედარებით, შესაბამისად ინტერნეტში არც ისე დიდი რაოდენობით ტექსტი მოიპოვება სხვა ენებთან შედარებით. ხშირად ვებსაიტები ქართულის მაგივრად ინგლისურსაც იყენებენ მეტი ხალხის მოსაზიდად. მეორე რაც არის აღსანიშნავი, არის ხარისხი. ინტერნეტში ყველას ყველაფრის დაწერა შეუძლია, შესაბამისად კორპუსის ხარისხიც იმასთან არის კავშირში, თუ რა კონტენტი იწერება ინტერნეტში. ქართული ვიკიპედია, სამეცნიერო ტექსტები და ლიტერატურა შეიძლება ბევრად უფრო მნიშვნელოვანი იყოს, თუმცა ხშირია ფორუმების ტექსტები, ბიზნესების ვებსაიტები და ა.შ. ამ შეზღუდვების დასაძლევად საინტერესო იქნება სხვა კვლევები, რომლებიც, მაგალითად, ქართული წიგნების ტექსტებს გააციფრულებენ და ამოიღებენ ტექსტს კორპუსისთვის, თუმცა ეს ჩვენი პროექტის საზღვრებს სცდება.

კორპუსის ხარისხიანობას ვამოწმებთ მოდელის დატრენინგებითა და სხვა მრავალენოვან მოდელებთან სხვადასხვა მეტრიკების შედარებით. მოგვიწია ტოკენაიზერის და მოდელის სულ თავიდან გაწვრთნა. ავირჩიეთ bert-small (Devlin et al., 2019)(Turc et al., 2019)(Bhargava et al., 2021) არქიტექტურის მქონე მოდელი, რომელზეც გავუშვით pretraining სწავლება ჩვენივე დატრენინგებული ტოკენაიზერით. ტოკენიზაციისა და ტრენინგის შედეგები ამ ნაშრომში უფრო დეტალურად იქნება განხილული.

კორპუსის შექმნა

მთავარი გამოწვევა იყო Common Crawl-ის დამფებიდან URL-ებისა და მათი მეტადატის მასიურად გადმოწერა. ჩვენ გადმოვწერეთ ყველა დამფი სერვერიდან და გავფილტრეთ URL-ები, რომ მხოლოდ ქართული საიტების მისამართები დარჩენილიყო. თითო დამფი, რომელიც იყო parquet ფორმატში, ზომაში დაახლოებით 200GB იყო. ჯამში იყო 100-მდე დამფი, რაც ნიშნავს, რომ გადმოვწერეთ URL-ების ~20TB მოცულობის მეტადატა, საიდანაც არაქართული ლინკები გადავყარეთ. ქართული ლინკების მეტადატა შევინახეთ csv ფორმატში. თითო დამფის შესაბამისი csv ფაილი საშუალოდ ~700MB გამოვიდა. ამ რიცხვებიდან გამომდინარე, ქართულად ჩაითვალა ლინკების 0.0001%-ზე ნაკლები. მონაცემების გადმოწერა/გაფილტვრისთვის თავდაპირველად გამოვიყენეთ pandas და requests პითონის ბიბლიოთეკები. შემდეგ ფიზიკური მეხსიერების უკმარისობის გამო pandas-ი ჩავანაცვლეთ Dask-ით და ასაჩქარებლად დავამატეთ პარალელიზება.

2018 წელზე ახალ დამფებს მეტადატაში ეწერა ვებგვერდის ენა, შესაბამისად მარტივი იყო მათი გაფილტვრა და მხოლოდ ქართული გვერდების ლინკების დატოვება, თუმცა უფრო ძველი დამფები ამ მეტადატას არ შეიცავდა. რადგანაც ყველა დანარჩენი დამფის გადმოსაწერად და გასაფილტრად საკმარისი რესურსი არ გვქონდა, ძველი დამფებისთვის ჩვენ გამოვიმუშავეთ ფილტრაციის ჩვენივე ევრისტიკები არაქართული საიტების გადასაყრელად. ამისათვის რამდენიმე კრიტერიუმი გვქონდა:

- მოვიკვლიეთ დომენები, რომელიც გამორჩეულად დიდი კონცენტრაციით შეიცავდა ქართულ გვერდებს. თუ ლინკი ამ დომენს მიეკუთვნებოდა, ვინახავდით.
- ლინკში ვამოწმებდით url-encoded ქართული სიმბოლოების არსებობას. თუ ქართულ სიმბოლოს შეიცავდა, ვინახავდით.
- თუ დომენს ჰქონდა “ka.” პრეფიქსი, “.ge” სუფიქსი, ან შეიცავდა “/ka/” ან “/ge/”-ს, ვინახავდით.

რადგანაც დატა მასიური იყო, გადმოწერისა და პროცესორის დიდ სიჩქარეს მოითხოვდა. შესაბამისად, ეს პროცესი გავყავით ნაწილებად სხვადასხვა კომპიუტერზე. თავდაპირველად გადმოწერას ვუშვებდით ჩვენს პირად კომპიუტერებზე და ერთი დამფის გადმოწერას სჭირდებოდა დაახლოებით 48 საათი. ამის ასაჩქარებლად დავიწყეთ სხვადასხვა ქლაუდ პროვაიდერების ვირტუალური მანქანების გამოყენება, რამაც ბევრად გაზარდა გადმოწერისა და გაფილტვრის სიჩქარე (~16 საათი). ამ ნაბიჯზე დაიხარჯა ყველაზე მეტი დრო.

წლიდან წლამდე დამფებში არსებობდა ბევრი ზუსტად იგივე URL, ამიტომაც საბოლოოდ ყველა ნაწილის გაერთიანებისას თუ ლინკი ზუსტად ემთხვეოდა, ვტოვებდით მხოლოდ უფრო ახალს. მეტადატის გადმოწერა-გაფილტვრის შემდეგ გვქონდა 91,214,441 URL, მათი დედუპლიკაციის შემდეგ დაგვრჩა 33,438,905 ლინკი.

ვებგვერდების გადმოწერა

ლინკების გადმოწერისა და გაფილტვრის შემდეგ უკვე თვითონ ვებგვერდების კონტენტის გადმოწერა დავიწყეთ. რადგანაც ლინკების კონტენტში ძალიან ბევრი HTML keyword-ი მოიპოვება, რაც არ გვჭირდება, გადავწყვიტეთ გამოგვეყენებინა [trafilatura](), რომელიც ასეთ keyword-ებს აგდებს ტექსტიდან და ტოვებს მხოლოდ ტექსტს თეგების გარეშე.

ჯამურად შევძელით 30,620,332 დოკუმენტის გადმოწერა. ეს პროცესიც გაყოფილი და გადანაწილებული იყო ქლაუდის სხვადასხვა ვირტუალურ მანქანაზე. csv ფაილი,

რომელიც ყველა ლინკს შეიცავდა, გავყავით 12 ტოლ ნაწილად (split_00-11.csv), თითო csv-ს ყველა ლინკის კონტენტსი გადმოწერას ვირტუალურ მანქანაზე დაახლოებით 50 საათი სჭირდებოდა. აქვე აღვნიშნავთ, რომ Common Crawl-ის სერვერები ხშირად rate-limited იყო და აბრუნებდა 403 სტატუსს, რის გამოც ზოგიერთ ლინკს ვერ ვიწერდით.

გადმოწერის შემდეგ თითო ნაწილის კონტენტი გავყავით 10 პარკეტ (.parquet) ფაილად, რომელზეც უკვე გაფილტვრა, გასუფთავება და დედუპლიკაცია უნდა გაგვეშვა. თითო პარკეტ ფაილი ზომაში დაახლოებით 300MB იყო.

გაფილტვრა ენის მიხედვით

წინა ნაწილებში ქართული URL-ების გაფილტვრა საკმარისი არ არის იმისთვის, რომ დავრწმუნდეთ, რომ მარტო ქართული ტექსტი გადმოვიწერეთ, ამიტომ ჩვენ დამატებით კიდევ ერთ ნაბიჯს ვამატებთ აქ, რაც არის ენის გაფილტვრა. ამისათვის გამოვიყენეთ fastText-ის მიერ გაწვრთნილი მოდელი. fastText-ის მოდელი ძირითადად გაწვრთნილია სხვადასხვა ენის wikipedia-ს ტექსტებზე და, ჩვენი გამოცდილებით, საკმაოდ კარგად არჩევს ქართულ ენაზე დაწერილ ტექსტებს. fastText-ის მოდელს ვაწვდით ტექსტს და გვიბრუნებს ვარაუდებს, რომელ ენაზე რა ალბათობითაა ეს ტექსტი. ჩვენ ეს მოდელი გავუშვით ყველა გადმოწერილ ტექსტზე და გავფილტრეთ დოკუმენტები, რომლებიც 95%-ზე ნაკლები ალბათობით იყვნენ ქართული. ამ კონკრეტულმა ნაბიჯმა გაფილტრა დოკუმენტების 30.02%.

გასუფთავება

ინტერნეტიდან გადმოწერილი გვერდების კონტენტს აუცილებლად სჭირდება გასუფთავება. ჩვენ გვინდოდა, რომ მაქსიმალურად მაღალი ხარისხის ტექსტი მოგვეპოვებინა და ამისთვის აუცილებელია, რომ ამოვშალოთ ხშირად გამეორებული ნაწილები როგორიცაა ნავბარები, ფუტერები და ა.შ., ან დოკუმენტი საერთოდ გადავაგდოთ. ამისათვის ვიყენებთ რამდენიმე ფილტრსა და ტრანსფორმაციას, რომლის ინსპირაცია ამ

სტატიებიდან მივიღეთ: CulturaX(Nguyen et al., 2023) და Scaling Language Models(Rae et al., 2021). კონკრეტულად ეს ფილტრები და ტრანსფორმაციები ხელით იყო არჩეული ბევრი ტესტირებითა და შემოწმებით, თითო კრიტერიუმის პარამეტრების მოდიფიკაციით და დადგენით, ტექსტის ხარისხს ზრდიდა თუ არა. გასუფთავების ჩვენ მიერ იმპლემენტირებული პროცესი ასე გამოიყურება:

- ვშლით ხაზებს, სადაც 4-ზე ნაკლები სიტყვაა.
- ვშლით ხაზებს, სადაც საერთოდ არ მოიძებნება ქართული ასო
- ვშლით მოკლე ხაზებს (<30 სიმბოლო) დოკუმენტის თავსა და ბოლოში
- ვშლით ისეთ დოკუმენტებს, სადაც ჯამში 50 სიტყვაზე ნაკლებია
- ვშლით ისეთ დოკუმენტებს, სადაც არასასურველი უწმაწური სიტყვები ხშირად მეორდება
- ვაგდებთ დოკუმენტებს, სადაც ხაზების 90% იწყება '*', '-', ან '·' სიმბოლოებით
- ვაგდებთ დოკუმენტებს, სადაც ხაზების 30% მთავრდება "..." სიმბოლოებით.
- ვაგდებთ ისეთ დოკუმენტებს, სადაც არ მოიპოვება რაღაც რაოდენობა ისეთი სიტყვების, რომელიც მიანიშნებს ტექსტის კითხვადობაზე, მაგალითად „და", „ან", „რა", „თუ", „არ" და ა.შ. (სრული სიტყვების სია გიტჰაბზე შეგიძლიათ ნახოთ)
- ვშლით ისეთ დოკუმენტებს, სადაც 80%-ს არ სცდება ისეთი სიტყვების რაოდენობა, რომელშიც ქართული სიმბოლოები მოიპოვება.

ამ პროცესის შემდეგ გადავყარეთ დოკუმენტების 53.92%.

გვინდა გავუსვათ ხაზი იმას, რომ ეს ნაწილი დიდი ალბათობით გაუმჯობესებადია. ქართული ენის ლინგვისტიკის ექსპერტი უკეთესად შეაფასებდა, თუ რა კრიტერიუმებით უნდა გაგვესუფთავებინა და გადაგვეყარა ტექსტები. თუ სამომავლოდ ვინმემ მსგავსი ნაშრომის

შექმნა გადაწყვიტა, ვურჩევდით ამ ნაწილის გაუმჯობესებასა და გასუფთავების ფილტრების მოდიფიკაციას.

დედუპლიკაცია

ძალიან ბევრ ლინკზე გამეორებული არის ტექსტის დიდი ნაწილი. შეიძლება მთლიანად არ ემთხვეოდეს ერთმანეთს, მაგრამ ასეთი დოკუმენტები მაინც გადასაყრელია. ამისათვის გამოვიყენეთ minHashLSH, რომლითაც ვყრით ისეთ დოკუმენტებს, რომლებიც უკვე შენახულ დოკუმენტს ზედმეტად ჰგავს. ამით თავს ვიზღვევთ ისეთი დატასეტის შექმნისგან, სადაც ბევრი განმეორება გვაქვს. კვლევებით ნაჩვენებია, რომ ამგვარი გამეორება მოდელების პერფორმანსს ძალიან დაბლა აგდებს, ამიტომ ეს ნაწილი ჩვენთვის მნიშვნელოვანი იყო. RefinedWeb(Penedo et al., 2023) და „Deduplicating training data makes language models better“(Lee et al., 2021) სტატიაში ნაჩვენებია რომ დუპლიკატი დოკუმენტები დიდ კორპუსებში განსაკუთრებით შემაფერხებელია მრავალპარამეტრიანი მოდელებისთვის.

MinHashLSH (Locality-Sensitive Hashing with MinHash) არის ალგორითმი, რომელიც გამოიყენება დიდი მოცულობის მონაცემებში მსგავსი ელემენტების სწრაფად მოსაძებნად. ის განსაკუთრებით სასარგებლოა, როდესაც გვჭირდება ტექსტებს/დოკუმენტებს შორის მსგავსების პოვნა, რადგან ბევრად ჩქარია, ვიდრე დოკუმენტების ყველა წყვილის ერთმანეთთან შედარება. MinHashLSH ორი მთავარი კომპონენტისგან შედგება: MinHash და Locality-Sensitive Hashing (LSH).

MinHash არის ტექნიკა, რომელიც გამოიყენება სიმრავლეების მსგავსების სწრაფად შესაფასებლად. ის ეფუძნება ჯაკარის მსგავსების ინდექსს.

1. თითოეული დოკუმენტი წარმოდგენილია სიმრავლედ, ჩვენს შემთხვევაში n-gram-ებად დაყოფის შემდეგ მიღებულ სიმრავლედ, რასაც ვიღებთ შინგლინგით (Shingling).

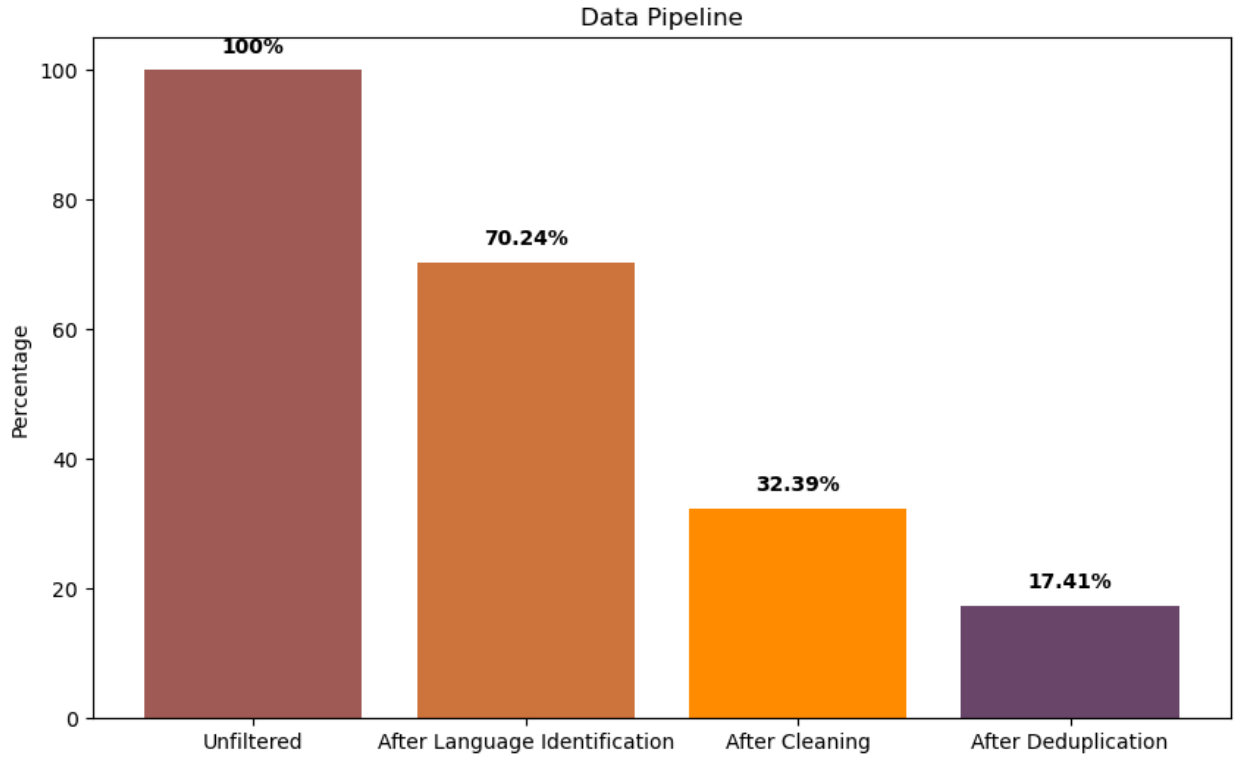
2. გამოვიყენოთ შემთხვევითად აღებული ჰეშფუნქციები, რომ დავჰეშოთ ყველა სიმრავლის ელემენტი.
3. თითოეული ჰეშ-ფუნქციისთვის ვპოულობთ სიმრავლის იმ ელემენტს, რომელსაც აქვს მინიმალური ჰეშ-მნიშვნელობა.
4. ამ მინიმალური ჰეშ-მნიშვნელობებით ვაწყობთ ვექტორს, რომელიც წარმოადგენს დოკუმენტის "ხელმოწერას" ან "სიგნატურას".
5. საბოლოოდ, თითო დოკუმენტისთვის ვიღებთ თითო ვექტორს, და შესაბამისად ყველა დოკუმენტზე ვიღებთ მატრიცას თუ ერთ მატრიცაში გავაერთიანეთ ყველა დოკუმენტის ვექტორი.

Locality-Sensitive Hashing (LSH) არის ტექნიკა, რომელიც გამოიყენება მაღალგანზომილებიან სივრცეში მსგავსი ობიექტების სწრაფად მოსაძებნად.

1. MinHash-ით მიღებული სიგნატურები იყოფა რამდენიმე ნაწილად ("ზოლებად").
2. თითოეული ზოლი გამოიყენება როგორც გასაღები ჰეშ-ცხრილში.
3. თუ ორი დოკუმენტის სიგნატურების ზოლები ემთხვევა, ისინი ხვდებიან ერთსა და იმავე bucket-ში (კოლიზია).
4. კოლიზიები ნიშნავს რომ ეს დოკუმენტები ერთმანეთის მსგავსებია

აქ დოკუმენტების 55.56% გადავაგდეთ. საბოლოოდ კორპუსში დაგვრჩა 5,333,536 დოკუმენტი.

მონაცემების დამუშავების შეჯამება



სურათი 1. შუამავალი დოკუმენტების პროცენტულობა

როგორც სურათი 1 დანართში ვხედავთ, ყველა ნაბიჯმა დიდი რაოდენობით დოკუმენტი გაფილტრა. ყველაზე მეტი დოკუმენტი გასუფთავების პროცესის დროს გადაიყარა და ჩვენ ვვარაუდობთ, რომ ეს უბრალოდ დიდი რაოდენობით უხარისხო ქართული ტექსტის გამო იყო, ან, ჩვენივე ფილტრები და ტრანსფორმაციები ბედმეტად მკაცრი შეიძლება ყოფილიყო. ქართული ენის ფილტრზე დაახლოებით 30% გაიფილტრა, რადგანაც ჩვენ არ ვითვალისწინებთ ისეთ დოკუმენტებს, რომლებიც ქართულთან ერთად სხვა ენებსაც შეიცავენ, რაც საკმაოდ ხშირია ვებსაიტებისთვის. აქ სხვა მიდგომამ შეიძლება უფრო მეტად გაამართლოს ქართული ენის შემთხვევაში, მაგალითად ჯერ გასუფთავების და შემდეგ ენის იდენტიფიცირების გაკეთებამ.

საბოლოოდ ჩვენ კორპუსში შემოგვრჩა 5,333,536 დოკუმენტი, ორიგინალური გაუფილტრავი დოკუმენტების მხოლოდ 17.41%, მაგრამ ბევრად უფრო მაღალი ხარისხის,

ვიდრე Common Crawl-იდან პირდაპირ გადმოწერილი დოკუმენტები. ჩვენ ამ კორპუსს საჯაროდ ვაქვეყნებთ HuggingFace-ზე.

მსგავსი ნაშრომები

აქამდე ჩვენთვის ცნობილი იყო ქეეკ (ქართული ენის ეროვნული კორპუსი) (Paul Meurer, et al. 2024), რომელიც მრავალი სპეციალისტის ძალისხმევით შეიქმნა. ჩვენ ამ პროექტის ერთ-ერთმა კოორდინატორმა, პაულ მოირერმა, წვდომა მოგვცა ამ კორპუსზე, რისთვისაც დიდ მადლობას ვუხდით მას. ქეეკ-ზე დაკვირვებით აღმოვაჩინეთ, რომ ის გამოირჩევა მაღალი ხარისხის ტექსტითა და ზუსტი სინტაქსური ანოტაციებით, თუმცა მისი მოცულობა შედარებით მცირეა იმაზე, რა მოცულობასაც ჩვენ ვუმიზნებდით. ჩვენ ჩვენი კორპუსისთვის ანოტაციებზე მნიშვნელოვნად ჩავთვალეთ დიდი მოცულობა, რადგან ჩვენს კორპუსს დიდი ენის მოდელის ტრენინგის მონაცემებად განვიხილავდით. ასევე აღმოვაჩინეთ, რომ ძირითადად ამ კორპუსის ტექსტები უკვე მოიპოვებოდა CommonCrawl-ში და დუბლირებული იყო, შესაბამისად აღარ გამოვიყენეთ.

ჩვენ ასევე მოვიკვლიეთ რამდენიმე ნაშრომი, რომელიც მოიცავს დიდი ზომის მრავალენოვანი, საჯაროდ ხელმისაწვდომი კორპუსის შექმნას და შინაარსით ახლოს დგას ჩვენს ნაშრომთან.

CulturaX (Nguyen et al., 2023)- ამ 2023 წლის ნაშრომის ავტორები პრობლემად ასახელებენ იმას, რომ მონაცემები, რომლებზეც მაღალი შედეგის მქონე LLM-ები იწვრთნება, ხშირ შემთხვევაში არაა საჯაროდ ხელმისაწვდომი, რაც განსაკუთრებით ეხება მრავალენოვან კორპუსებს. დიდი, ხარისხიანი კორპუსის არარსებობის გამო open-source დატასეტზე მოდელის გაწვრთნა რთულია. ამ პრობლემის გადასაჭრელად ავტორებმა შექმნეს დიდი მოცულობის კორპუსი, რომელიც მოიცავს 167 ენას. CulturaX-ის შესაქმნელად აიღეს mC4 და OSCAR დატასეტები, შეკრეს ერთად და ამ საერთო მონაცემებზე გააკეთეს გაწმენდისა და დედუპლიკაციის ნაბიჯები. ამ ნაბიჯების მათ მიერ გამოყენებული იმპლემენტაცია ეფექტური აღმოჩნდა და საბოლოო კორპუსიც დიდი და ხარისხიანი

გამოვიდა, რის გამოც ჩვენც მიზანშეწონილად ჩავთვალეთ, მსგავსი მეთოდები გამოგვეყენებინა. აღსანიშნავია, რომ ამ დატასეტში ქართული ენაც საკმაოდ დიდი მოცულობით(დაახლოებით 3 მილიონი დოკუმენტი) არის წარმოდგენილი. ჩვენ, ამ ნაშრომისგან განსხვავებით, გადავწყვიტეთ, კონცენტრაცია გვქონოდა ქართულ ენაზე და კიდევ უფრო დიდი ქართული კორპუსი შეგვექმნა, ვიდრე მსგავს მრავალენოვან ნაშრომებში.

RefinedWeb (Penedo et al., 2023) - ეს არის ასევე 2023 წელს შექმნილი ნაშრომი, რომელშიც დიდი ზომის კორპუსი შექმნეს ვებგვერდებიდან შეგროვებული ტექსტებით. CulturaX-ისგან განსხვავებით, ეს დატასეტი უმეტესად ინგლისურენოვან მონაცემებს შეიცავს. ტექსტების გაფილტვრის, გაწმენდისა და დედუპლიკაციის ნაბიჯები აქაც მსგავსია. ამ ნაშრომში წამოჭრილია იდეა, რომ ვებგვერდებიდან შეგროვებული ტექსტები, მისი დიდი მოცულობიდან გამომდინარე, კარგი დედუპლიკაციისა და გაწმენდის პირობებში უფრო მეტად გამოდგება დიდი მოდელების გასაწვრთნელად, ვიდრე ხელით შერჩეული წყაროებიდან შეგროვებული კორპუსები, მიუხედავად იმისა, რომ ინტერნეტიდან შეგროვებული მონაცემები შეიძლება არ იყოს საუკეთესო ხარისხის.

The Pile (Gao et al., 2020) - 2020 წლის ბოლოს დაწერილი ნაშრომი, სადაც აღწერილია კორპუსი, რომელიც შექმნეს ყურადღებით შერჩეული წყაროებიდან. ჯამში არის 22 წყარო, რომლებიც დაყოფილია კატეგორიებად: აკადემიური, ინტერნეტი, პროზა, დიალოგი, და სხვა. ეს კორპუსი გამოირჩევა მაღალი ხარისხის ტექსტებით და არის კარგი კორპუსი LLM-ის გასაწვრთნელად. მსგავსი მიმართულების გაყოლას ჩვენი კორპუსისთვისაც განვიხილავდით, თუმცა საბოლოოდ მთლიანად ინტერნეტიდან შეგროვებული მონაცემების გამოყენება გადავწყვიტეთ, რადგან, ერთი მხრივ, ვნახეთ, რომ ინტერნეტიდან მონაცემების შეგროვებითა და მისი გაწმენდით ძალიან კარგი კორპუსის შექმნაა შესაძლებელი, და, მეორე მხრივ, ქართული ენისთვის ვერ მოვძებნეთ მაღალი ხარისხის ტექსტების წყაროები დიდი მოცულობით, რომლებსაც ჩვენს ნაშრომში გამოვიყენებდით.

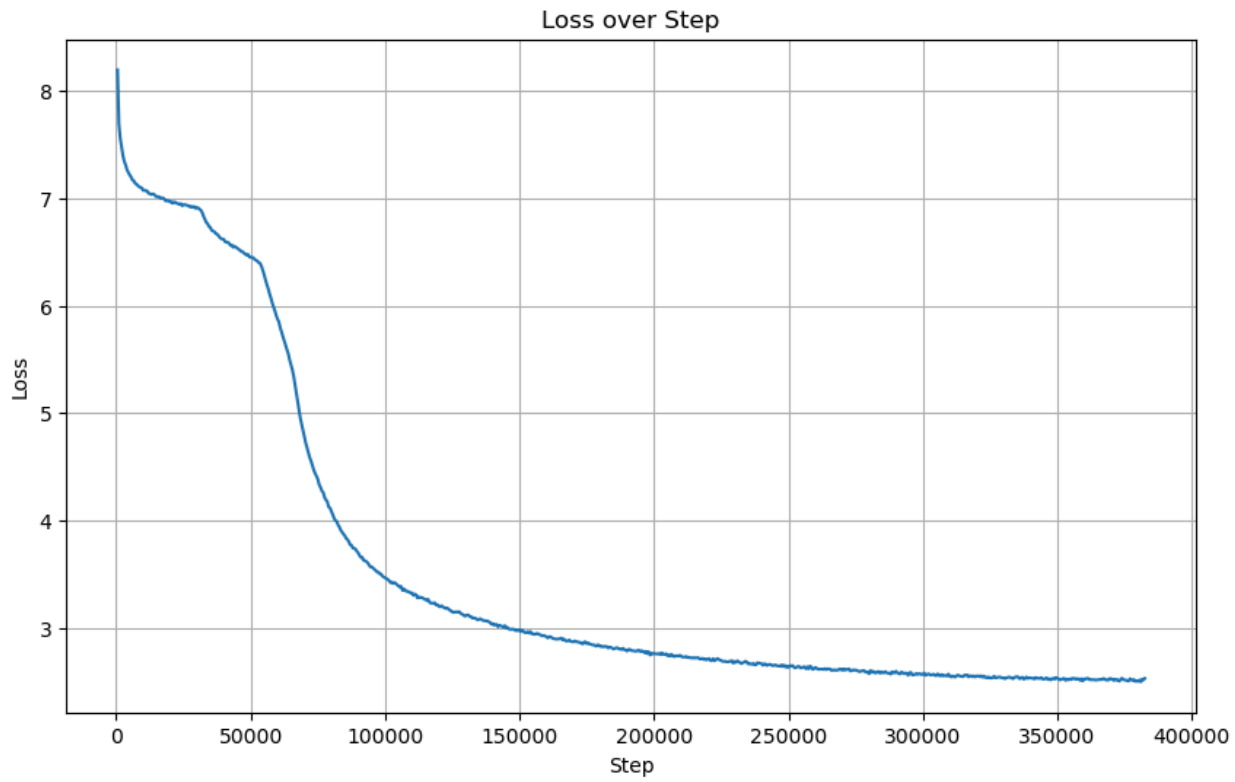
ანალიზი და ექსპერიმენტები

ჩვენი კორპუსის ანალიზისთვის გავწვრთენით bert-small (Devlin et al., 2019) (Bhargava et al., 2021) (Turc et al., 2019) არქიტექტურის მქონე მოდელი MLM (Masked Language Modeling) ამოცანაზე. ეს კონკრეტული ამოცანა გულისხმობს იმას, რომ მოდელმა უნდა ისწავლოს მასკირებული (Masked) ტოკენების ამოცნობა ნებისმიერ შეყვანილ წინადადებაში. MLM არის მოდელის გაწვრთნის მეთოდი რომელიც ხშირად გამოიყენება, რომ კარგად აღიქვას ენა, და ამის შემდგომ ამავე მოდელის კონკრეტულ ქვეამოცანებზე გაწვრთნა (Fine-Tuning) შეიძლება მოხდეს.

რადგანაც ჩვენ საერთოდ ნულიდან ვიწყებდით, საჭირო იყო ასევე ტოკენაიზერის გაწვრთნაც ამავე კორპუსზე, რადგან წინასწარ გაწვრთვინილი (Pretrained) ტოკენაიზერები უფრო მორგებულები იყვნენ ინგლისურ ენაზე და ქართული ენის ნიუანსები არ ესმოდათ. ტოკენაიზერი იწვრთნება დეტერმინისტულად, ანუ არ იქნება განსხვავება ერთსა და იმავე მონაცემებზე გაწვრთნილ ტოკენაიზერებს შორის, თუ ყველაფერი სხვაც მსგავსი აქვს. ტოკენაიზერი სტატისტიკური ანალიზის მიხედვით ითვლის ტოკენების ყველაზე ოპტიმალურ სიას ამ კონკრეტული მონაცემებისთვის. ჩვენ გამოვიყენეთ BertTokenizerFast ტოკენაიზერის კონფიგურაცია და ტრენინგი გავუშვით Kaggle-ზე მთლიან კორპუსზე, რასაც დაახლოებით 2 საათი დასჭირდა ჯამში. ტოკენაიზერი ატვირთულია საჯაროდ HuggingFace-ზე.

ტოკენაიზერის გაწვრთნის შემდეგ ჩვენ გავუშვით მთლიანი კორპუსის ტოკენიზაცია, ასევე Kaggle-ზე. ტოკენიზაციას დასჭირდა 8 საათი და ეს ტოკენიზებული კორპუსიც ავტვირთეთ HuggingFace-ზე, რომ მოდელის ტრენინგისას დრო არ დახარჯულიყო ტოკენიზაციაში.

ჩვენი მოდელის ტრენინგი ტოკენიზებულ კორპუსზე 1 ეპოქით გავუშვით, რაც ნიშნავს, რომ მხოლოდ ერთხელ გადაუყვებოდა მთლიან კორპუსს. ბატჩების ზომა იყო 8, და არ გამოგვიყენებია გრადიენტის აკუმულაცია.



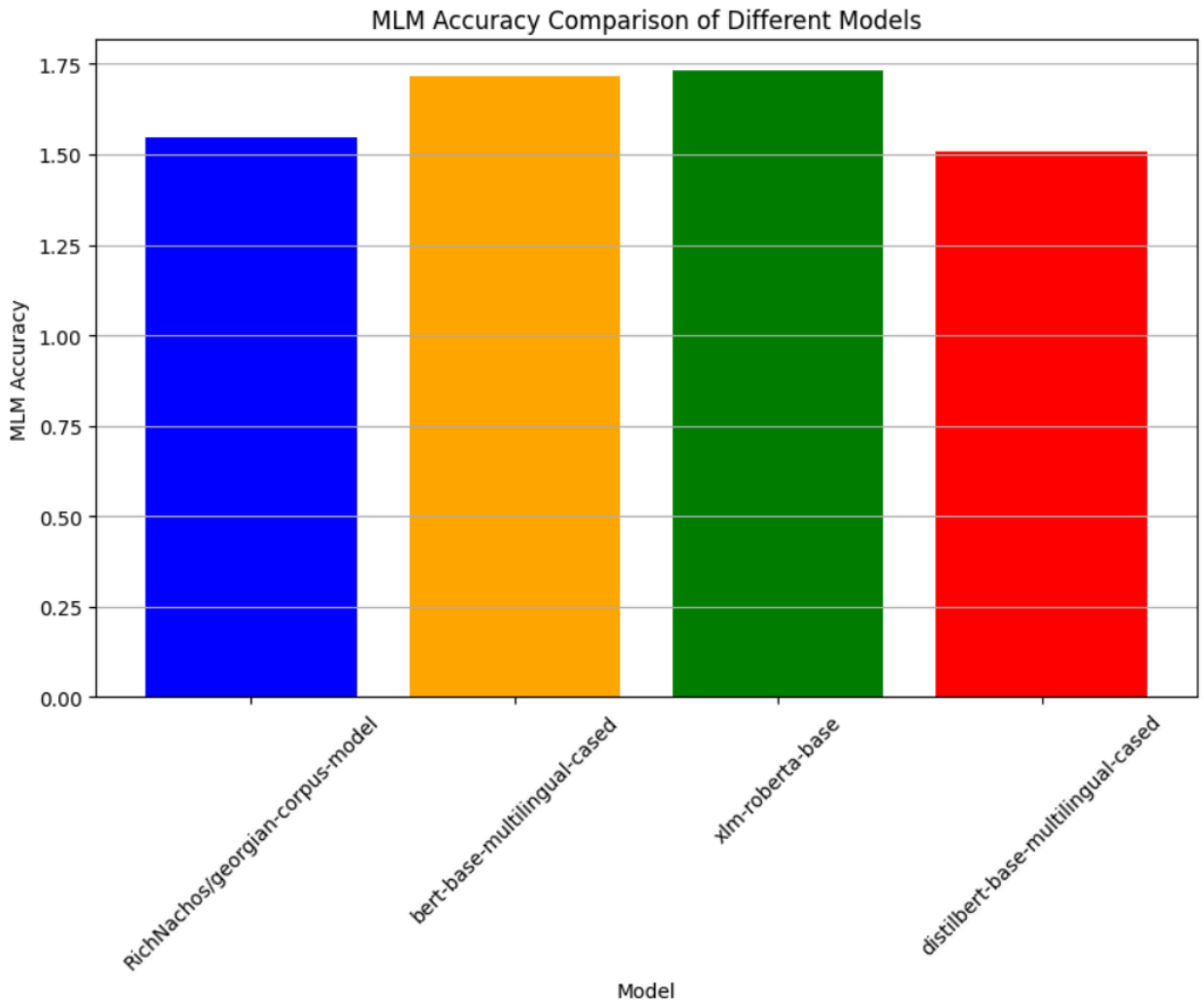
სურათი 2. მოდელის გაწვრთნისას Loss

უნდა აღვნიშნოთ, რომ შეზღუდული რესურსების გამო ვერ შევძელით უფრო დიდი მოდელის დატრენინგება. ეს მოდელი დავატრენინგეთ RTX 2060 Super ვიდეო ბარათზე, რომელსაც 8GB მეხსიერება აქვს, რაც გვზღუდავდა როგორც ტრენინგის სიჩქარეში, ასევე ბატჩების და მოდელის ზომაშიც. სურათ 2-ზეც ჩანს, რომ ტრენინგი უკვე ძალიან მცირე შედეგს გვაძლევდა 100,000 ბიჯის შემდეგ. ვვარაუდობთ, რომ უფრო მრავალპარამეტრიანი მოდელი უკეთესად აითვისებს მთლიანი კორპუსის მონაცემებს.

ჩვენი დატრენინგებული მოდელი (BERTidze) შევადარეთ სხვა მრავალენოვან მოდელებს, კონკრეტულად:

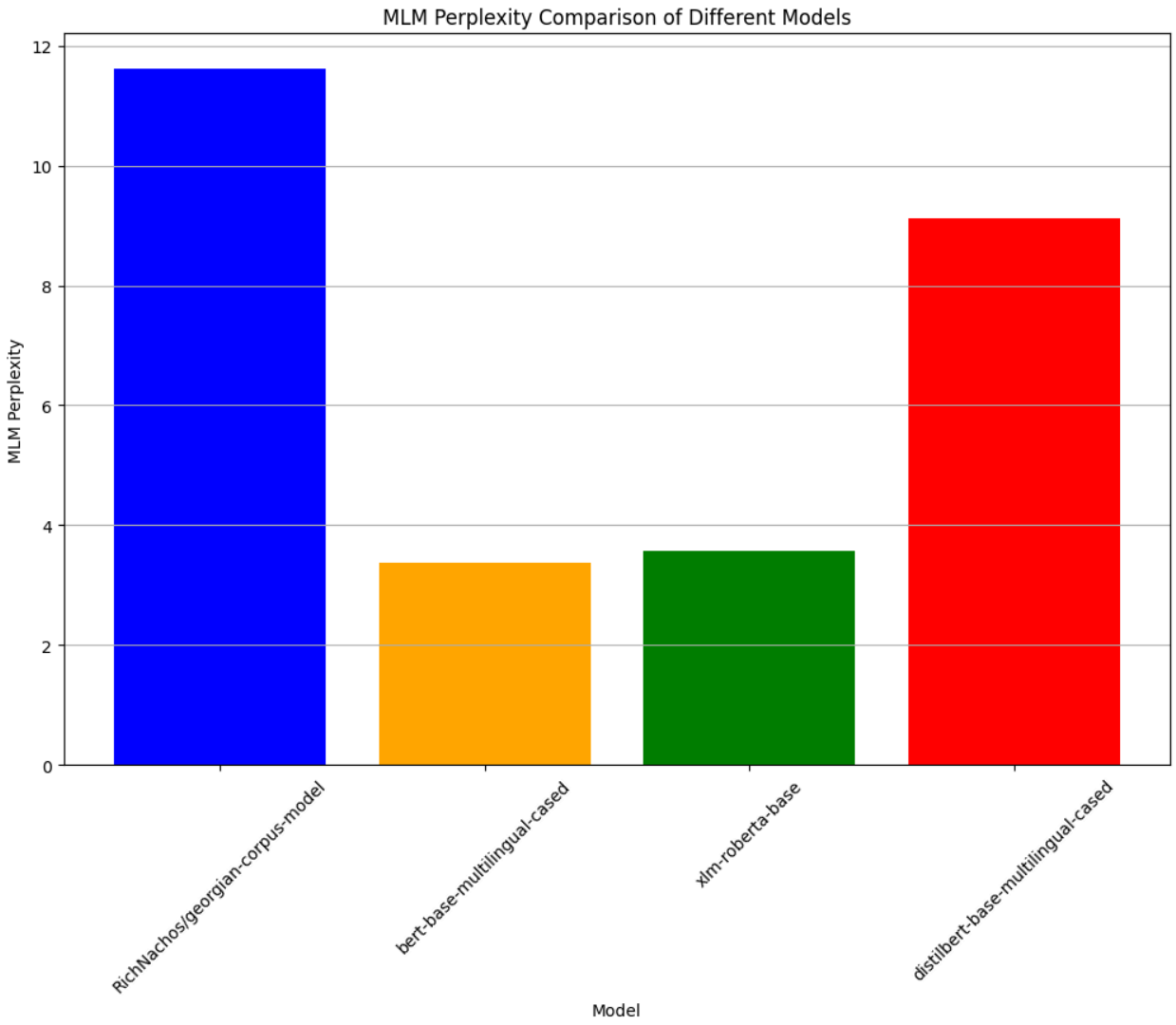
- bert-base-multilingual-cased (179 მილიონი პარამეტრი)
- xlm-roberta-base (279 მილიონი პარამეტრი)
- distilbert-base-multilingual-cased (135 მილიონი პარამეტრი)

ჩვენი მოდელი ბევრად პატარაა და მხოლოდ 29 მილიონი პარამეტრი აქვს.



სურათი 3. მოდელების MLM სიზუსტეების შედარება.

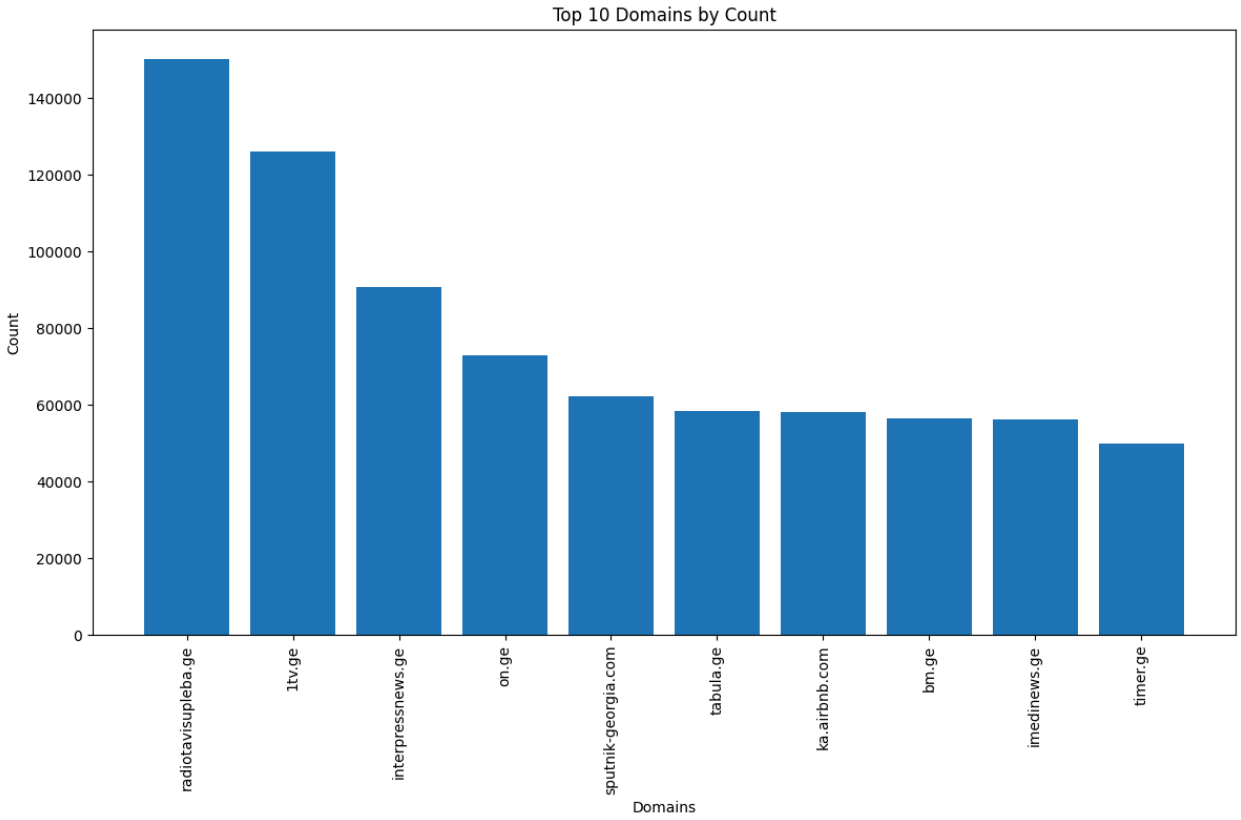
მცირე ევალუაციის დატასეტზე გატესტვისას ეს იყო MLM სიზუსტეები (Accuracy). MLM Accuracy არის მეტრიკა, რომელიც ითვლის, რამდენად კარგად იცნობს მოდელი მასკირებულ ტოკენს. ჩვენმა მოდელმა ყველაზე პატარა მოდელს მცირედით აჯობა, და ყველაზე დიდ მოდელებს ვერ დაეწია პერფორმანსში, თუმცა 30 მილიონ პარამეტრიანი მოდელის მიერ 130 მილიონ პარამეტრიან მოდელზე უკეთესი შედეგის ჩვენება უკვე საკმაოდ კარგია. ეს მიანიშნებს იმაზე, რომ ჩვენი დატასეტი უკეთესი ხარისხისაა, ვიდრე ის რაზეც ეს მოდელები გაწვრთნეს.



სურათი 4. მოდელების დაბნეულობის (Perplexity) შედარება

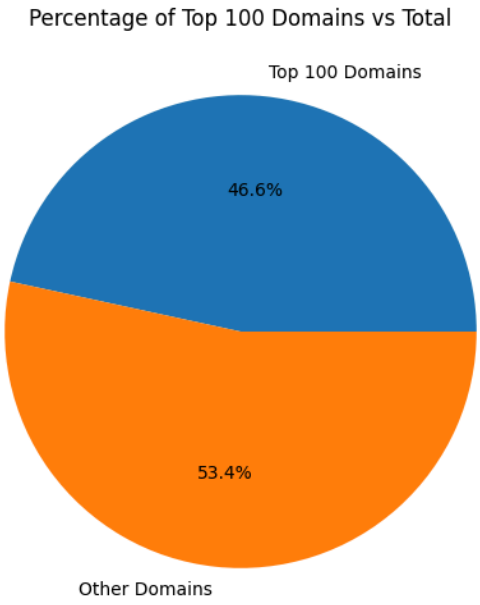
მოდელის დაბნეულობა გამოითვლება მარტივი ფორმულით e^{Loss}, სადაც *Loss* არის საშუალო *Cross-Entropy Loss*. ეს მეტრიკა ხშირად გამოიყენება იმის შესაფასებლად, თუ რამდენად კარგად ეუფლება და ესმის მოდელს კონტენტი, რაზეც ევალუაცია ხდება. ჩვენი მოდელი ამ განხრით საკმაოდ სუსტია, მაგრამ როგორც ზემოთ ვახსენეთ, *Loss* გაწვრთნის უმეტეს დროს ვერ უმჯობესდებოდა და უფრო დიდი მოდელის გაწვრთნის შემთხვევაში სავარაუდოა, რომ ბევრად უკეთესად შეითვისებდა კორპუსის მონაცემებს.

უშუალოდ კორპუსის ანალიზს რაც შეეხება, ჩვენ შეგვიძლია ვნახოთ, რომელი დომენებიდან არის ყველაზე მეტი და მაღალხარისხიანი დოკუმენტი.



სურათი 5. ტოპ 10 დომენი დოკუმენტების რაოდენობით

საკმაოდ მოსალოდნელი შედეგია, რომ ყველაზე მეტი დოკუმენტი ახალი ამბების საიტებიდანაა. სურათი 5-ში არ ჩანს ვიკიპედია, საიდანაც დაახლოებით 40,000 დოკუმენტი იყო.



სურათი 6. ტოპ 100 დომენის პროცენტულობა

დამატებით სურათი 6-ში ჩანს, რომ ტოპ 100 დომენი მთელი კონტენტის დაახლოებით 50%-ს შეადგენს, რაც იმაზე მიანიშნებს, რომ ჩვენი დატასეტი საკმაოდ მრავალფეროვანია.

შეჯამება

ამ პროექტის ფარგლებში შევქმენით ქართული ენის კორპუსი, რომელიც მოიცავს 5,333,536 დოკუმენტს. ეს კორპუსი წარმოადგენს ყველაზე დიდ საჯაროდ ხელმისაწვდომ რესურსს ქართული ენისთვის და მნიშვნელოვანი ნაბიჯია ქართული ენის ციფრული რესურსების განვითარებაში.

კორპუსის შექმნის პროცესი მოიცავდა რამდენიმე ეტაპს:

1. Common Crawl-იდან ქართული URL-ების იდენტიფიცირება და გადმოწერა
2. ვებგვერდების კონტენტის გადმოწერა
3. ენის მიხედვით გაფილტვრა
4. ტექსტის გასუფთავება
5. დედუპლიკაცია

თითოეულ ეტაპზე გამოვიყენეთ სხვადასხვა ტექნიკა და ალგორითმები, რათა მიგველო მაღალი ხარისხის ტექსტური მონაცემები. პროცესის შედეგად საწყისი მონაცემების მხოლოდ 17.41% შევინარჩუნეთ, რაც მიუთითებს გაფილტვრის მკაცრ კრიტერიუმებზე.

კორპუსის ხარისხის შესაფასებლად გავწვრთენით BERT-ის არქიტექტურაზე დაფუძნებული მოდელი (BERTidze) და იგი შევადარეთ არსებულ მრავალენოვან მოდელებს. მიუხედავად იმისა, რომ ჩვენი მოდელი გაცილებით მცირე ზომისაა (29 მილიონი პარამეტრი), მან აჩვენა კარგი შედეგები MLM სიზუსტის მხრივ, რაც მიანიშნებს კორპუსის მაღალ ხარისხზე.

პროექტის ფარგლებში შექმნილი რესურსები, მათ შორის კორპუსი, დამუშავებული მონაცემები, მოდელი და ტოკენაიზერი, საჯაროდ ხელმისაწვდომია HuggingFace-ზე, რაც ხელს შეუწყობს ქართული ენის NLP კვლევებისა და აპლიკაციების განვითარებას.

მომავალი კვლევებისთვის რეკომენდებულია:

- კორპუსის გაფართოება დამატებითი წყაროებით (მაგ., ციფრული წიგნები)
- გასუფთავების პროცესის გაუმჯობესება ლინგვისტების ჩართულობით
- უფრო დიდი და კომპლექსური მოდელების გაწვრთნა
- კორპუსის გამოყენება სხვადასხვა NLP ამოცანებისთვის

ეს პროექტი წარმოადგენს მნიშვნელოვან ნაბიჯს ქართული ენის ციფრული რესურსების განვითარებაში და ქმნის საფუძველს მომავალი კვლევებისა და აპლიკაციებისთვის. შეგვიძლია დარწმუნებით ვთქვათ, რომ ამ პროექტის მიზანს, რაც იყო ქართული ენის ყველაზე დიდი და ხარისხიანი, საჯაროდ ხელმისაწვდომი კორპუსის შექმნა, მივაღწიეთ.

გამოყენებული ლიტერატურა

Bhargava, P., Drozd, A., & Rogers, A. (2021). Generalization in NLI: Ways (not) to go beyond simple heuristics. *ArXiv Preprint ArXiv:2110.01518*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *North American Chapter of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:52967399

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., & Nabeshima, N. (2020). The pile: An 800gb dataset of diverse text for language modeling. *ArXiv Preprint ArXiv:2101.00027*.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., & Carlini, N. (2021). Deduplicating training data makes language models better. *ArXiv Preprint ArXiv:2107.06499*.

Nguyen, T., Van Nguyen, C., Lai, V. D., Man, H., Ngo, N. T., Dernoncourt, F., Rossi, R. A., & Nguyen, T. H. (2023). Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *ArXiv Preprint ArXiv:2309.09400*.

Paul Meurer. (2024). *The Georgian National Corpus*.

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., & Launay, J. (2023). The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *ArXiv Preprint ArXiv:2306.01116*.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., & Young, S. (2021). Scaling language models: Methods, analysis & insights from training gopher. *ArXiv Preprint ArXiv:2112.11446*.

Turc, I., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Well-read students learn better: On the importance of pre-training compact models. *ArXiv Preprint ArXiv:1908.08962*.