სსიპ “ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი”

დავით აბულაშვილი

ქართული ტექსტის ხმად გარდაქმნა

ნაშრომი შესრულებულია კომპიუტერული მეცნიერებების მაგისტრის აკადემიური ხარისხის მოსაპოვებლად.

ხელმძღვანელი: პროფესორი პაატა გოგიშვილი

თბილისი

2020

ანოტაცია

In this research we provide one of the most interesting problems: text to speech. Relevance of this task and importance for speakers of Georgian language is shown. Various potential uses and implications of the product are provided.

Different ways to solve this task are discussed, including classical programing approach, as well as artificial intelligence technologies.
The processes of data collection, processing, and analysis as well as the results of the experiment conducted are presented in this paper. The steps taken for training the neural networks and challenges faced during the process are also discussed.

The results are summarized, future goals are set and plan for achieving goals is provided.

ნაშრომში წარმოდგენილია კვლევა დღესდღეობით ერთ-ერთ ყველაზე საინტერესო ამოცანის გარშემო: ტექსტის ხმად გარდაქმნა. ნაჩვენებია ამ საკითხის აქტუალურობა და მნიშვნელობა ქართული ენისთვის.

წარმოდგენილია ამ ტექნოლოგიის დანერგვის სხვადასხვა გზები და გამოყენების სფეროები. განვიხილავთ სხვადასხვა მიდგომებს ამ ამოცანის გადაჭრისთვის, როგორც კლასიკური დაპროგრამებით, ასევე ხელოვნური ინტელექტის ტექნოლოგიების გამოყენებით. ავარჩევთ ჩვენი აზრით ყველაზე ეფექტურ გზას.

წარმოვადგენთ ჩატარებული ექსპერიმენტების შედეგებს და განვიხილავთ მონაცემების შეგროვების, დამუშავების და მომზადების საკითხებს. ასევე ნეირონული ქსელების წვრთნისთვის საჭირო ნაბიჯებს და "წყალქვეშა ქვებს", რომელიც გზადაგზა შეიძლება შეგვხვდეს.

შევაჯამებთ შედეგებს, დავისახავთ სამომავლო მიზნებს და დავგეგმავთ კონკრეტულ ნაბიჯებს ამ მიზნების მისაღწევად.

შესავალი

ხელოვნური ინტელექტის ტექნოლოგიები დღესდღეობით ყველაზე სწრაფად განვითარებადი და პერსპექტიულია. მათი მეშვეობით ისეთი ამოცანების გადაჭრაა შესაძლებელი, რომლებიც კლასიკური პროგრამირებით შეუძლებელია.

ხელოვნურმა ინტელექტმა განსაკუთრებით კარგი შედეგები აჩვენა სტატისტიკურ პროგნოზირებაში, კომპიუტერულ ხედვაში, ობიექტების ამოცნობაში, კლასიფიკაციაში და სხვა. ამ სფეროებში ის ადამიანზე უკეთეს შედეგებს აჩვენებს. ბოლო წლებში ეს ტექნოლოგია სულ უფრო მეტ სფეროს ეუფლება. ყველასთვის მოულოდნელი შედეგები კი ხელოვნურმა ინტელექტმა ხელოვნებაში აჩვენა, როდესაც მან შეძლო სურათების და ხმის გენერირება ცნობილი ხელოვანების სტილში.

ერთ-ერთი მნიშვნელოვანი მიმართულება ხელოვნურ ინტელექტში არის ბუნებრივი ენების დამუშავება. მისი საშუალებით შესაძლებელია თარჯიმნის გაკეთება, ტექსტიდან შინაარსის ამოღება და თემატურად განაწილება, ხმოვანი ბრძანებების აღქმა, ხმის ტექსტში და პირიქით - ტექსტის ხმაში გარდაქმენა.

თანამედროვე ტექნოლოგიების განვითარება, ძირითდად ისეთი ენებისთვის ხდება, რომლებზეც მოსაუბრე ადამიანების რაოდენობა ითვლის ათეულ მილიონებს.

აქედან გამომდინარე ქართულ ენაზე მსგავსი მონაცემთა სიმრავლე არ არსებობს (თუ რომელიმე კომპანიას აქვს შექმნილი, არ არის მის შესახებ ინფორმაცია ხელმისაწვდომი) და არც წინასწარ გაწვრთნილი მოდელებია ღიად ხელმისაწვდომი.

ვფიქრობთ ერთ-ერთი ყველაზე მნიშვნელოვანი და საინტერესო ამოცანაა ტექსტიდან ხმის დაგენერირებაა. ქართულ ენაზე არც ეს პრობლემაა გადაჭრილი და ამ კვლევის მიზანიც სწორედ ეს იქნება.

ქართული ტექსტის ხმაში გადაყვანა

1. ამოცანის არსი

ტესტის ხმაში გადაყვანა დიდი ხანია აქტუალური ამოცანაა ჩვენს ტექნოლოგიურ ხანაში. მეოცე საუკუნის მეორე ნახევარში უკვე დაიწყო მუშაობა იმაზე, რომ ინტერაქტიულ ავტომოპასუხეებში ან აუდიო განცხადებებში (მაგ. აეროპორტი, სატრანსპორტო სადგურები) გამოეყენებინათ არა ადამიანური რესურსები, არამედ კომპიუტერი. თავდაპირველად ყველაფერი ხდებოდა პროგრამული უზრუნველყოფით და წინასწარ ჩაწერილი წინადადებებისა თუ სიტყვების კომბინაციებით, თუმცა ტექნოლოგიების განვითარებასთან ერთად ამოცანის გადაჭრის მეთოდებიც უფრო დაიხვეწა.

ხელოვნური ინტელექტის ტექნოლოგიების განვითარებამდე აუდიო ფაილების ხელოვნურად გენერირების მეთოდები არ არსებობდა. ბოლო დროს დიდმა კომპანიებმა თავიანთი რესურსები უფრო აქტიურად მიმართეს ამ ამოცანის გადასაჭრელად. ერთ-ერთი პირველები, რომლებმაც ეს პრობლემა ასე თუ ისე გადაჭრეს, იყვნენ Google და Amazon. ინგლისური ენის ტექსტიდან ხმაში გადაყვანის სერვისები უკვე საკმაოდ დახვეწილია და ადამიანის ბუნებრივ საუბართან მიახლოებულ შედეგამდეც ძალიან ახლოსაა.

აღსანიშნავია, ასევე ამ ამოცანის შებრუნებული ამოცანაც, ხმის ტექსტად გარდაქმნა. ეს საკითხი უფრო მარტივი აღმოჩნდა ხელოვნური ინტელექტისთვის და უკვე ბევრმა კომპანიამ მიაღწია წარმატებას ამ მიმართულებით. ეს ტექნოლოგია გამოიყენება ყველგან, სადაც არის ელექტრონული მოწყობილობების ხმოვანი მართვა. ეს მიდგომა ძალიან პოპულარულია, ტელეფონების, ტელევიზორების, ჭკვიანი სახლის, უკვე მანქანის მართვისთვისაც კი, თუმცა ამ ეტაპზე ძალიან კარგი შედეგები მიღწეულია მხოლოდ რამდენიმე ათეულ ძალიან ხშირად გამოყენებად ენაზე - ინგლისური, ესპანური, გერმანული, ჩინური, რუსული, კორეული და სხვა (იმ ენებზე, რომლებზეც მრავალი ათეული მილიონი ადამიანი საუბრობს). ამავე დროს,

მისი ხარისხი, განსაკუთრებით ნაკლებად გავრცელებულ ენებზე, არ არის დამაკმაყოფილებელი, მონაცემების ან სამომხმარებლო ბაზრის სიმცირის გამო.

მიუხედავად იმისა, რომ ქართულ ენაზე ხმიდან ტექსტში გადაყვანის ტექნოლოგიები არ არის მაღალ დონეზე განვითარებული, მაინც შესაძლებელია პირობითად დამაკმაყოფილებელი შედეგის მიღება თუნდაც Google-ის API-ის მეშვეობით. ხოლო ტექსტის ხმაში გადაყვანის მეთოდები ქართულ ენაზე ძალიან მწირია, იმის გათვალიწინებით, რომ არის რამდენიმე ნაშრომი, სადაც ქართულად "აალაპარაკეს" კომპიუტერი. თუმცა არც ეს შედეგი და არც მონაცემები არაა ღიად ხელმისაწვდომი, ანუ არაა შესაძლებელი თქვენთვის სასურველი ტექსტის გარდაქმნა ხმოვან სიგნალად.

ამ ნაშრომის მიზნებია:

1. შეიქმნას და საჯაროდ ხელმისაწვდომი გახდეს მონაცემები, რომელთა გამოყენებას შეძლებს ნებისმიერი მსურველი ხმის გენერაციის ქსელების წვრთნისთვის;
2. შეიქმნას ნეირონული ქსელი, რომელიც შეძლებს ქართული ტექსტის ხმაში გადაყვანას ადამიანის ბუნებრივ საუბართან მიახლოებული ხარისხით;

უკანასკნელი რამდენიმე წლის განმავლობაში, განსაკუთრებით, 2020 წელს დიდი რაოდენობის ახალი კვლევა მიმდინარეობს ამ მიმართულებით და სულ უფრო და უფრო მეტი ახალი ნეირონული ქსელის მოდელი იქმნება. ეს, რა თქმა უნდა, ხაზს უსვამს ამოცანის აქტუალურობასა და მნიშვნელოვნობას. წამყვანი კომპანიები ფლობენ მონაცემების დიდ ბაზას, ეს საშუალებას აძლევს მათ სხვადასხვა მიდგომა გამოიყენონ და მიაღწიონ ისეთ შედეგებს, რომლთა მიღწევა სახლის პირობებში შეუძლებელია. ამგვარად, ამ ამოცანის ერთ-ერთი მნიშვნელოვანი ნაწილი არის სწორედ ისეთი მოდელის შერჩევა და იმ რაოდენობის მონაცემის შეგროვება, რომელთა საშუალებითაც მივიღებთ დამაკმაყოფილებელ შედეგს.

2. გადაჭრის მეთოდები

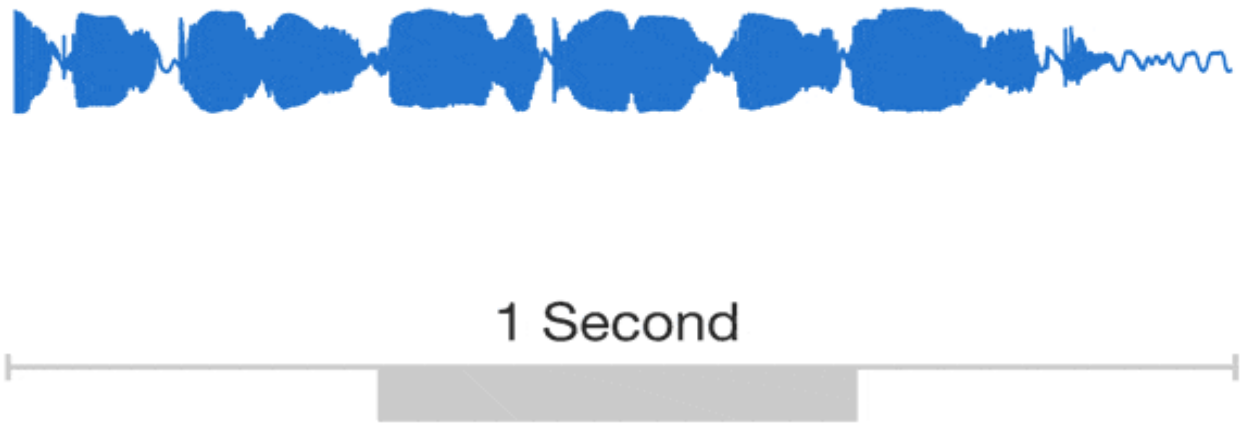
დღესდღეობით ინტერნეტში მოიპოვება მრავალი სახის ნეირონული ქსელის მოდელი, რომელიც ტექსტიდან ხმას აგენერირებს. მათი არქიტექტურაც და მიღწეული შედეგებიც ერთმანეთისგან განსხვავებული და მრავალფეროვანია.

თავდაპირველად ეს ტექნოლოგია გამოიყენეს დიდმა კომპანიებმა თავიანთ ხმოვან ასისტენტებში, როგორებიცაა: Apple Siri, Google Assistant, Amazon Alexa, Microsoft Cortana და სხვა. მათი ხმა ძალიან არაბუნებრივი იყო, რადგან საუბრის დაგენერირებისთვის იყენებდნენ გადაბმის მეთოდს (concatenative methods). რაც გულისხმობს შემდეგს: დანაწევრებული ფრაზები და სიტყვები იყო ჩაწერილი ადამიანის მიერ, ხოლო ამ ხმების ყველაზე ლოგიკური გზით ერთმანეთთან გადაბმა ხდებოდა ხელოვნური ინტელექტის ან რთული ალგორითმის საშუალებით. აღსანიშნავია, რომ ასეთ გენერირებულ ხმას ჰქონდა ხარვეზები, მაგ. არაბუნებრივი ინტონაციები და პაუზები არასათანადო ადგილებში.

ტექსტის ხმად გარდაქმნის ამოცანის განვითარების შემდეგი ეტაპი იყო ხმის სინთეზი მარკოვის ფარულ მოდელზე დაყრდნობით (Hidden Markov model) [6] [18]. ეს მოდელი მუშაობს როგორც სასრული ავტომატი (Finite State Machine) და მისი მეშვეობით აგებს წინადადების შესაბამის ხმოვან მოდელს, იყენებს ისეთ ხმებს, რომელიც უფრო ბუნებრივი იქნება წინადადების ამა თუ იმ ადგილას. ამ ხერხით მიღებული შედეგი ბევრად უკეთესი გამოდგა, ვიდრე უბრალო გადაბმით მიღებული ხმა, თუმცა არსებული პრობლემები არ მოუგვარებია და ბუნებრივისგან ისევ საკმაოდ შორს იყო.

პირველი ფართოდ გამოყენებადი ასეთი მიღწევა იყო Google-ის თარჯიმანი, რომელიც გარდა იმისა, რომ ტექსტური ფორმით წარმოგვიდგენდა სხვადასხვა ენებზე თარგმანს, კითხულობდა კიდეც ამ ტექსტებს. Google-მა ასევე ხელმისაწვდომი გახადა ეს სერვისი API-ს საშუალებით. გასაკვირი არ იქნება თუ ვიტყვით, რომ ამ სფეროში პირველობას კვლავ Google ინარჩუნებს.

რამდენიმე წელიწადში გამოქვეყნდა რევოლუციური ხმის გენერირების მოდელი WaveNet [2], რომელიც ძალიან ჰგავდა ადამიანის ბუნებრივი საუბარს, გაქრა არაბუნებრივი წყვეტები სიტყვებს შორის და ინტონაციების უზუსტობები, რომელიც მანამდე ძალიან ხშირი იყო. ეს მოდელი იყენებს გენერაციული ტიპის ღრმა ნეირონულ ქსელს (Generative Deep Neural Network) და ცდილობს ტექსტიდან შექმნას ისეთი ხმა, რომელიც იმეორებს ბუნებრივი ხმის სინაფსებს, მაქსიმალურად შესაძლებელი სიზუსტით. ასევე ისეთ ხმებსაც კი, რომელიც წინა მეთოდებში არ იყო გათვალისწინებლი, როგორიცაა ენის მოძრაობის ხმა, სუნთქვა, ნერწყვის გადაყლაპვა. სწორად ესაა საიდუმლო იმისა, რომ ხმა ძალიან ახლოსაა ბუნებრივთან.



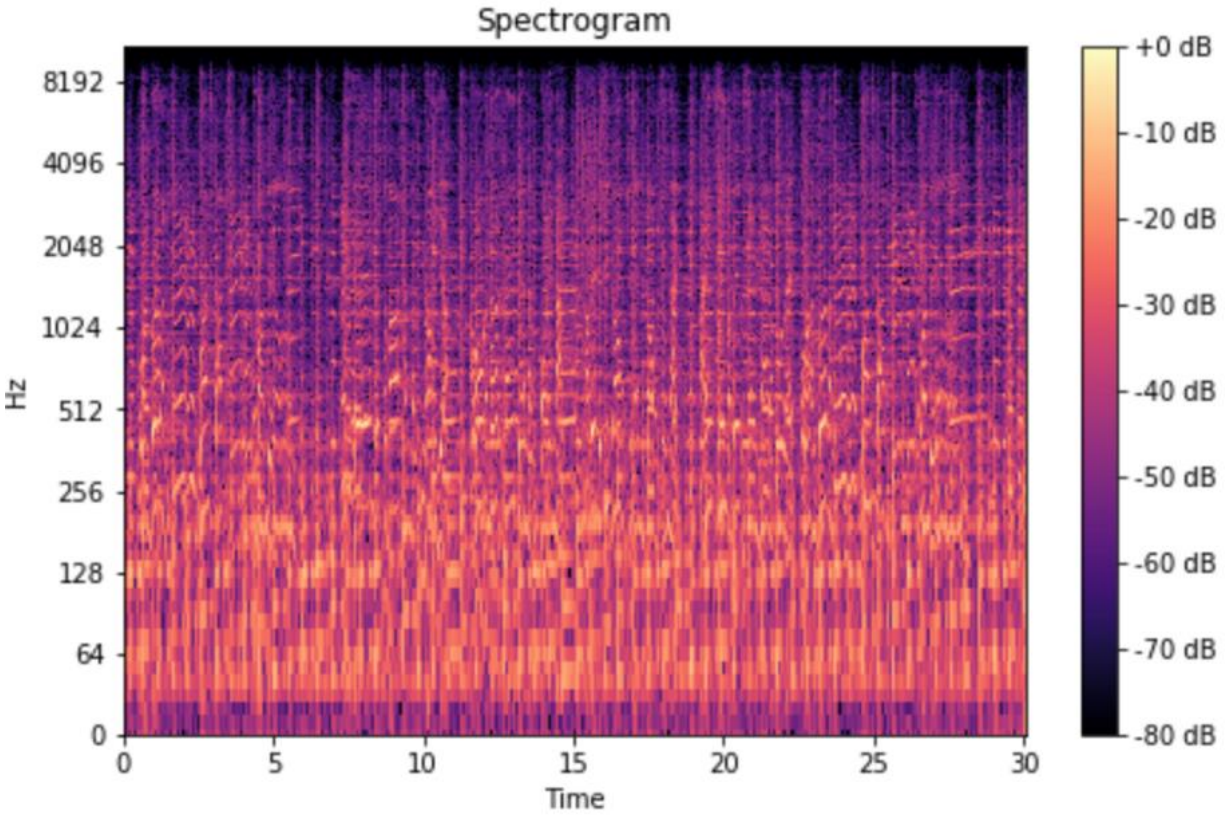
სურათი 1. წყარო [5]

ქსელის არქიტექტურა ორი ნაწილისგან შედგება. პირველი ნაწილი იღებს წინადადებებიდან ენისთვის დამახასიათებელი თვისებებს (Language Features) და მეორე ნაწილი ამ თვისებებს წარმოადგენს ხმოვანი სიგნალების სახით. ქსელის წვრთნა მიმდინარეობს დაახლოებით 100 საათი (10GiB) ზომის მონაცემების "მოსმენით". ერთი კვირის წვრთნის შემდეგ Nvidia Tesla V100 ვიდეოდაფის გამოყენებით მას უკვე შეუძლია "საუბარი" და დაგენერირებულ წინადადებებში

სიტყვების გარჩევაა შესაძლებელი. ადამიანთან მიახლოებული ხმის მისაღებად სავარაუდო წვრთნის დროა ერთი თვე.

ცხადია ამ მოცულობის მონაცემების შეგროვება და ამდენი ხნით ქსელის წვრთნა არაა მარტივი და ხელმისაწვდომი. ამიტომ ამ მიმართულებით კვლავ გაგრძელდა კვლევა პრატიკულად მეტად გამოსადეგი და ადვილად საწვრთნელი მოდელის მისაღებად. ამ მცდელობების შედეგად მრავალი სხვა მოდელი შეიქმენა, თუმცა აღსანიშნავია, რომ WaveNet-ის იდეა არის ყველა შემდგომ ცნობილ ქსელში წამყვანი და გარკვეულწილად ყველა ამ ქსელის "შთამომავალია".

Google-ის უახლესი მიღწევა არის ხმის სინთეზი Tacotron ქსელის [4] საშუალებით. ეს ქსელი გვაძლევს საშუალებას ტექსტი გარდავქმნათ მელის სპექტროგრამად (რაც სპექტროგრამის გამოსახვის სპეციფიკურ ფორმატს წარმოადგენს). იხილეთ სურათი 2.



სურათი 2. წყარო [14]

როგორც უკვე აღვნიშნეთ, ტექსტის ხმაში გადაყვანის თითქმის ყოველ მომდევნო მოდელში გამოიყენეს WaveNet ქსელი ან მისგან აიღეს იდეები. Tacotron ქსელიც იყენებს მას, თუმცა მიდგომა ცოტათი შეიცვალა და იმის მაგივრად, რომ WaveNet-ს პირდაპირ ტექსტიდან დაეგენერირებინა ხმა, ახლა ეს პროცესი ორ ეტაპად დაიყო: ტექსტი ჯერ სპექტროგრამის სახით წარმოგვიდგება, ხოლო შემდეგ სპექტროგრამის გარდაქმნა ხდება ხმაში. ასეთმა არქიტექტურამ კიდევ უფრო კარგი შედეგი გამოიღო.

მოგვიანებით NVIDIA-მ გამოაქვეყნა კვლევა Tacotron 2, რომელში ტექსტის სპექტროგრამაში გადაყვანის მოდელი იყო დატოვებული Google-ის, ხოლო WaveNet-ის მაგივრად გამოიყენეს WaveGlow [3], რომელიც მათ თვითონ შექმეს WaveNet და Glow [17] ქსელების იდეების გაერთიანებით. Glow წარმოადგენს კომპანია OpenAI-ს შექმნილ გენერაციულ ნეირონულ ქსელს, რომელიც გენერაციის ძალიან მაღალი ხარისხით გამოირჩევა. WaveGlow [16] გამოირჩევა ასევე იმითაც, რომ მონაცემების ფორმატი, რითიც ის იწვრთნება ისეთივეა, როგორც Tacotron-ზე და მის წვრთნას სჭირდება შედარებით ნაკლები დრო. მონაცემების რაოდენობა 24 საათი (2 GiB) და 2 დღე-ღამის წვრთნა უკვე დამაკმაყოფილებელ შედეგს იძლევა.

რაც შეეხება ტექსტის ხმაში გადაყვანის უახლეს მოდელებს, ზოგიერთი მათგანი არ არის ჯერ იმპლემენტირებული, ზოგიერთი არაა ღიად განთავსებული, ზოგიც კი ჯერ არაა გამოცდილი კომპანიებისა თუ ადამიანების მიერ და შესაბამისად ნაკლებად სანდოა.

3. საკითხის მოგვარების შერჩეული მეთოდი

ქართულ ენაზე ამ ამოცანის გადაჭრისას არსებითად გავითვალისწინე ის ფაქტი, რომ ხელმისაწვდომი სასწავლო მონაცემები იქნებოდა მცირე ოდენობით. არსებული ნეირონული ქსელებიდან ავარჩიე ისეთები, რომლის წვრთნისთვისაც მინიმალური რაოდენობის მონაცემი იქნებოდა აუცილებელი. აღმოჩნდა, რომ საჭიროა მინიმუმ 24 საათი ჩანაწერების ჯამური ხანგრძლივობა საწვრთნელი მონაცემებისთვის, რომ შედეგი დამაკმაყოფილებელი მივიღოთ. ყველაზე გავრცელებულ, კლასიკურ მონაცემთა სიმრავლეს ინგლისური ენისთვის წარმოადგენს LJ Speech [13]. ეს მონაცემთა სიმრავლე თითქმის სტანდარტად არის ქცეული ამ ტიპის ქსელების წვრთნის შედეგის შესამოწმებლად. მოცემულ ნაშრომში დასახული მიზნების მისაღწევად გადაწყდა ქართული ენისთვის ამ მონაცემთა სიმრავლის ანალოგის შექმნა. ეს მონაცემთა სიმრავლე ღიად განთავსდება ინტერნეტში.

ასევე მნიშვნელოვანი იყო ამერჩია ქსელი, რომელზეც შევამოწმებდი ჩემს შეგროვილ მონაცემებს. თავდაპირველად შევარჩიე DC TTS მოდელი [7], რომელიც ეყრდნობა ამ კვლევას [1] და ძალიან კარგ შედეგს იძლევა მცირე დროში. აღსანიშნავია, რომ ამ ქსელს შეუძლია ერთი ადამიანის ხმაზე უკვე გაწვრთნილი მოდელი გადაწვრთნას ისე, რომ ქსელი “საუბრობდეს” მეორე ადამიანის ხმაზე, ამისთვის კი მეორე ადამიანის ხმის 2-3 საათიანი მონაცემთა სიმრავლის დამუშავებაც კი საკმარისია მოდელის მიერ. ამ მიმართულებით გადავდგით პრაქტიკული ნაბიჯები, თუმცა რამდენიმე მიზეზის გამო გადაწყდა ამ ტექნოლოგიების შეცვლა, კონკრეტულად, - სირთულეებს ვაწყდებოდით შუალედური შედეგების შემოწმების პროცესში, ასევე, ეს ქსელი იმპლემენტირებულია მოძველებული ტექნოლოგიებით, რაც ეფექტურობის თვალსაზრისითაც ქმნის სირთულეებს. შემდეგ ეტაპზე გადაწყდა Tacotron 2-ზე გადასვლა. ეს ქსელი არის დღესდღეობით ყველაზე ხშირად გამოყენებადი, წვრთნის დრო უფრო ნაკლები აქვს და მიღებული შედეგებიც უკეთესი აღმოჩნდა.

ჩვენს კვლევაში ერთ-ერთი ყველაზე შრომატევადი აღმოჩნდა მონაცემთა სიმრავლის შექმნა. საბოლოოდ (ნორმალიზაციის პროცესის დასრულების შემდეგ) მივიღეთ ქართული

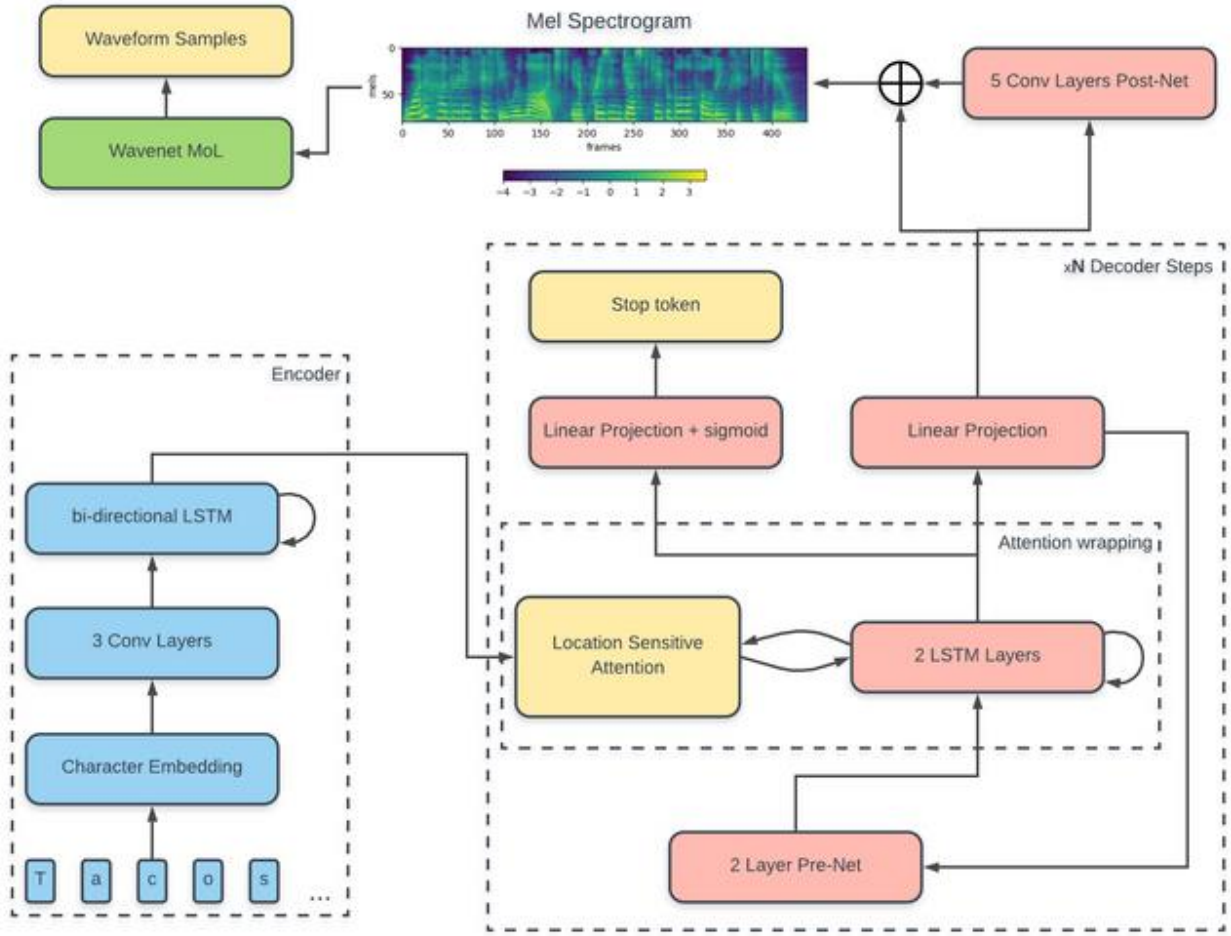
ტექსტების და შესაბამისი ხმოვანი ფაილების 22 საათის ხანგრძლივობის მონაცემთა სიმრავლე.

არსებობს მონაცემების შეგროვების რამდენიმე მეთოდი. ერთ-ერთი გულისხმობს აუდიო წიგნების დაყოფას წინადადებებად და შესაბამის ხმოვან ფაილებად. აქ მნიშვნელოვანია, რომ ხმოვანი ფაილები ეკუთვნოდეს ერთ ადამიანს და არ ჰქონდეს ფონად სხვა ხმები. სამწუხაროდ, ღიად დადებული ასეთი ქართული აუდიო წიგნების სათანადო ოდენობით მოძიება ვერ ხერხდება. შემდეგი მეთოდი არის ტელევიზიის ან რადიოს ჩანაწერები ტრანსკრიპთან ერთად. კვლევის პროცესში დავუკავშირდით ტელევიზიებს და რადიო მაუწყებლებს. სამწუხაროდ, გამოხმაურება მხოლოდ ერთი ტელევიზიიდან მივიღეთ, თან მათი მონაცემების გამოყენების ფორმა არ გულისხმობდა ღია სახით განთავსებას. აქედან გამომდინარე, საბოლოოდ, გადაწყდა, რომ საკუთარი ძალებით მოგვეპოვებინა საჭირო რაოდენობის მასალა.

შევქმენით ვებ-გვერდი, რომლის საშუალებითაც შესაძლებელია წინასწარ დაყოფილი ტექსტური ფორმით მოცემული წინადადების გახმოვანება და შესაბამისი ხმის ჩაწერა. ეს წინადადებები მზადდებოდა წიგნების და სხვადასხვა თემატიკის სტატიების მიხედვით. საიტი ამ წინადადებებს ავტომატურად აწვდის გამხმოვანებელ პიროვნებას. წინადადების ჩაწერის შემდეგ, მონაცემთა ბაზაში ინახება ხმოვანი ფაილი, წინადადების ტექსტი და გამხმოვანებლის იდენტიფიკატორი. შესაბამისად, მონაცემებს აღარ სჭირდება დამატებითი მითითება რომელი აუდიო ფაილი შეესაბამება რომელ ტექსტურ ფაილს. მონაცემთა სიმრავლე ავტომატურად იქმნება.

4. Tacotron2 და Waveglow

Tacotron 2 არის კომპანია DeepMind-ის შექმნილი ქსელი [4], რომელიც გარდაქმნის წინადადებას სპექტროგრამად და იყენებს დამატებით ქსელ WaveNet-ს იმისთვის, რომ სპექტროგრამა გარდაქმნას აუდიო სიგნალად. ქვემოთ მოცემულია თვითონ Tacotron-ის ქსელის არქიტექტურა. იხილეთ სურათი 3.



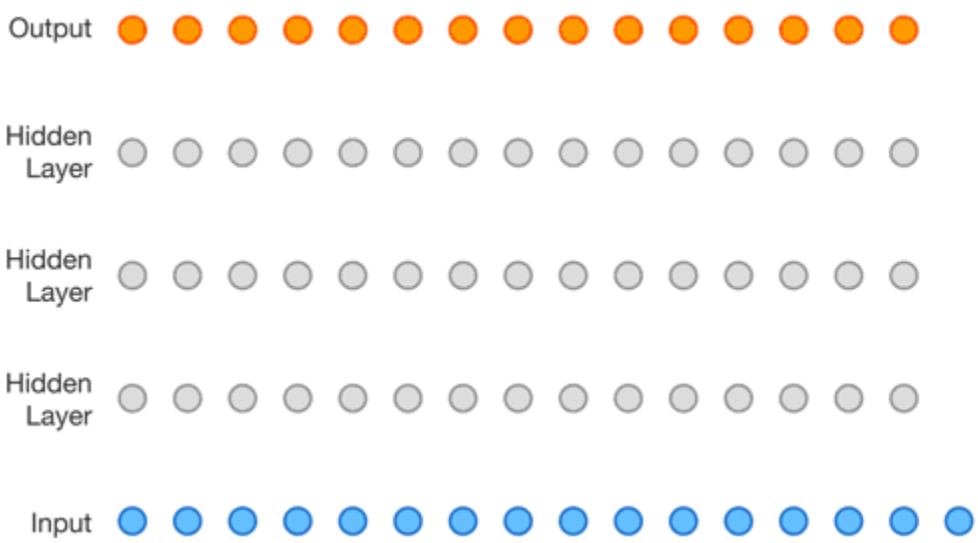
სურათი 3. წყარო [4]

Tacotron 2 შედგება სამი ნაწილისგან:

1) ენკოდერი, რომელიც კითხულობს სიმბოლოებს და ენის თვისებებში გარდაქმნის;
2) დეკოდერი, რომელიც ენის ამ თვისებებს გარდაქმნის სპექტროგრამის თვისებებში;
3) გენერატორი, რომელიც აგენერირებს სპექტროგრამის მატრიცას.

დამატებით, ამ მატრიცის სურათად წარმოდგენაც არის შესაძლებელი და შეგვიძლია ხმის დაგენერირებამდეც შევაფასოთ მიღებული შედეგი.

WaveGlow მოდელის ქსელს აქვს საინტერესო არქიტექტურა. ეს არის ავტორეგრესიული ნეირონული ქსელი, რომელიც იყენებს კონვოლუციურ შრეებს და ყოველი კონვოლუციის შედეგს იმატებს ქსელს პარამეტრად. ეს გვაძლევს იმას, რომ ხმოვანი სიგნალების მიმდევრობის ნაწილი დამოკიდებული იყოს წინა ნაწილის შედეგებზე. აქვე მოცემულია თვალსაჩინო ილუსტრაცია, თუ როგორ მუშაობს ეს დამოკიდებულება. იხილეთ სურათი 4



სურათი 4. წყარო [13]

5. მონაცემების მომზადება

მონაცემების მომზადება განხორციელდა პროგრამის [11] მეშვეობით, რომლიც შეიქმნა ამ კვლების ფარგლებში, თუმცა ეს პროგრამა ცდება ამ კონკრეტული კვლების ფარგლებს. ნებისმიერი მსურველი, ვინც გადაწყვეტს იმავე გზით შეაგროვოს მონაცემები, შეძლებს ამ პროგრამის გამოყენებას. მისი საშუალებით შესაძლებელია ხმის ჩაწერა, წინადადებების სრული სიის, ჩაწერილი წინადადებების და ჩაწერილი ხმების საერთო ხანგრძლივობის ნახვა. ასევე, მოცემული პროგრამა შენახვისას ხმას გარდაქმნის საჭირო ფორმატში.

აუდიო ფაილები შესაძლებელია ჩაიწეროს სახლის პირობებში, თუმცა მნიშვნელოვანია, რომ ჩაწერის პროცესის განმავლობაში მაქსიმალურად გამოირიცხოს ფონური ხმაური. ხმის ჩაწერა ვცადეთ რამდენიმე მოწყობილობაზე, მათ შორის ლეპტოპზე და სხვადასხვა ოპერაციული სისტემების მქონე რამდენიმე მობილურ ტელეფონზე. არ იყო გამოყენებული სპეციალური პროგრამული უზრუნველყოფა, ჩაწერისთვის გამოვიყენეთ ინტერნეტ ბრაუზერში ჩაშენებული საშუალებები. შესაბამისად ხმის ხარისხი მნიშვნელოვნად დამოკიდებული იყო მოწყობილობის მიკროფონის ხარისხზე.

კლასიკური ინგლისური მონაცემების სიმრავლის [12] აღწერაში იყო მითითებული, რომ ხმა იყო ჩაწერილი MacBook Pro-ს მეშვეობით, თუმცა ჩვენს ხელთ არსებული არცერთი ლეპტოპის ხმა არ აღმოჩნდა დამაკმაყოფილებელი ხარისხის. სამწუხაროა, რომ ერთ საათამდე ხანგრძლივობის ხმის ჩანაწერების გაუქმება მოგვიხდა ამ მიზეზით. შემდეგ ეტაპზე ხმების ჩაწერა გაგრძელდა მობილური ტელეფონის საშუალებით და ხარისხთან დაკავშირებით პრობლემები აღარ შექმნილა.

აუდიო ფაილებისთვის მოთხოვნილი თვისებები უნდა იყოს შემდეგნაირი:

1) აუდიო ფაილის ფორმატი- wav;
2) აუდიოს არხი - mono;
3) სიხშირე - 22050 Hz;

ნეირონული ქსელი ამ ფორმატის ფაილებზეა მორგებული.

ჩანაწერების თავდაპირველი ხარისხი და მონაცემები განსხვავებოდა საბოლოო მონაცემების სიმრავლისგან, რადგან თითქმის შეუძლებელია ბრაუზერით ხმის ჩაწერისას ამ პარამეტრების მითითება. მონაცემების სწორ ფორმატში გადასაყვანად გამოვიყენეთ პროგრამა ffmpeg, რომელიც არის უალტერნატივო და სტანდარტული ხელსაწყო მულტიმედია (აუდიო, ფოტო, ვიდეო) ფაილებს დასამუშავებლად და მანიპულაციისთვის.

აქვე უნდა აღინიშნოს, რომ რამდენიმე ექსპერიმენტის შემდეგ აღმოჩნდა, რომ ჩვენი მონაცემები ბევრად დაბალ ხმაზე იყო ჩაწერილი, ვიდრე ინგლისური ანალოგი. ჩვენს შემთხვევაში იმავე ხელსაწყოთი ხმის 3-ჯერ გაზრდამ დამაკმაყოფილებელი შედეგი გამოიღო.

6. მონაცემების განაწილება და ნორმალიზება

მონაცემების მომზადების შემდეგ ჩავატარეთ რამდენიმე ექსპერიმენტი, რომლებმაც სამწუხაროდ, სასურველი შედეგი ვერ მოგვცა. წარუმატებლობის მიზეზების გარკვევის მიზნით, გადაწყდა ჩატარებულიყო მონაცემთა ანალიზი.

ვივარაუდეთ, რომ სპექტროგრამის მატრიცების სხვადასხვა მნიშვნელობების ანალიზი საშუალებას მოგვცემდა აღმოგვეჩინა მონაცემთა სიმრავლეში არსებული პრობლემები. გადავწყვიტეთ შესწავლილი და შერჩეული მნიშვნელობები შეგვედარებინა მუშა ინგლისურ მონაცემთა სიმრავლეს, რომელიც მოცულობით თითქმის უტოლდებოდა ქართულს. მატრიცები ერთმანეთს შევადარეთ შემდეგი მონაცემების მიხედვით: მატრიცის მინიმუმი, მაქსიმუმი, საშუალო მნიშვნელობა, აბსოლუტური მნიშვნელობის მაქსიმუმი და სხვაობა მაქსიმუმსა და მინიმუმს შორის, ანუ ამპლიტუდა.

აღმოჩნდა, რომ საშუალო მნიშვნელობა არანაირ ინფორმაციას არ იძლეოდა, რადგან ხმოვანი ტალღების მოძრაობა ერთმანეთს "ახშობდა" და ძალიან ახლოს იყოს ნულთან. მაქსიმუმი და მინიმუმის მნიშვნელობა კი აღარ იყო საჭირო ამსოლუტური მნიშვნელობის მაქსიმუმის შემდეგ, რადგან ეს სამი პარამეტრი თითქმის ერთ სურათს აჩვენებდა, ხოლო აბსოლუტურში იყო გათვალისწინებული მაქსიმუმის და მინიმუმის გადახრებიც.

სპექტროგრამის შესაბამისი მატრიცების აბსოლუტური მნიშვნელობების მაქსიმუმების განაწილება მონაცემებში გამოიყურება შემდეგნაირად: პირველ სურათში არის ინგლისური მონაცემთა სიმრავლის განაწილება, რომელიც ახლოსაა ნორმალურ განაწილებასთან. მეორე სურათში არის დაუმუშავებელი ქართული მონაცემების განაწილება, რომელიც ძალიან არის აცდენილი ნორმალურ განაწილებას. ამ გრაფიკებმა აჩვენა, რომ მონაცემების დიდი ნაწილის მნიშვნელობები მერყეობს 15000-დან 25000-მდე ინგლისურში და 2500-დან 10000-მდე ქართულში. ვივარაუდეთ, რომ ხმის ხელოვნური აწევა ჩანაწერებისთვის ამ მნიშვნელობებსაც აწევდა. ვარაუდი გამართლდა და გარდაქმნილი მონაცემების მნიშვნელობები მიუახლოვდა ინგლისურისას.

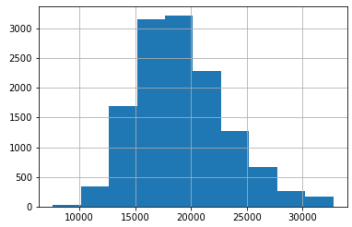
ამის შემდეგ ექსპერიმენტულად დადასტურდა, რომ ეს გარდაქმნა არ აღმოჩნდა საკმარისი შედეგის მისაღწევად, ქსელი რამდენიმე ათასი იტერაციის შემდეგ ისევ მხოლოდ შემთხვევით ხმაურს აგენერირებდა. წვრთნის პროცესზე დაკვირვებისას შეიმჩნეოდა, რომ გარკვეული მონაცემების გათვალისწინების შემდეგ, ცდომილების ფუნქცია იღებდა მისთვის მეტისმეტად შეუსაბამო მნიშვნელობას, რაც მიუნიშნებდა იმაზე, რომ მონაცემთა სიმრავლეში იყო გამონაკლისები (outliers).

გადაწყდა ჩაგვეტარებინა ინგლისურენოვანი მონაცემების სრული ანალიზი და შეგვესწავლა რა ჩარჩოებში არის მოქცეული თითოეული ხმოვანი ფაილი. აღმოჩნდა, რომ ხმოვანი ფაილების სიგრძე არ უნდა ყოფილიყო 1 წამზე ნაკლები და 10 წამზე მეტი ხანგრძლივობის.
ასევე დავაგენერირეთ თითოეული აუდიო ფაილის სპექტროგრამა, როგორც ინგლისურის, ასევე ქართულის და შერჩევითად გადამოწმების შედეგად კვლავ აღმოჩნდა რამდენიმე გამონაკლისი ფაილი.

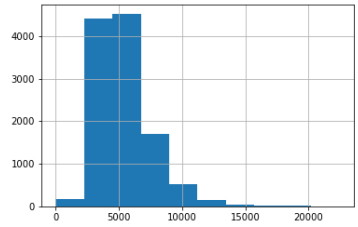
სურათი 5: ინგლისური მონაცემთა სიმრავლის მატრიცების აბსოლუტური მნიშვნელობების მაქსიმუმი.

სურათი 6: ქართული მონაცემთა სიმრავლის მატრიცების აბსოლუტური მნიშვნელობების მაქსიმუმი.

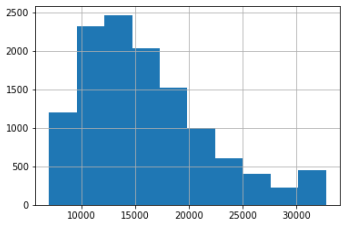
სურათი 7: ქართული მონაცემთა სიმრავლის მატრიცების აბსოლუტური მნიშვნელობების მაქსიმუმი ნორმლიზების და გაფილტვრის შემდეგ.



სურათი 5

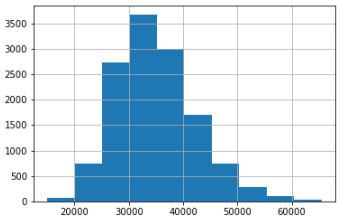


სურათი 6

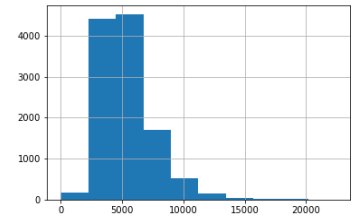


სურათი 7

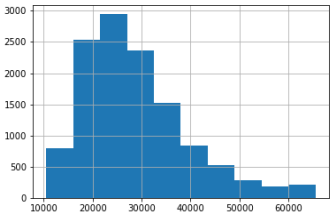
ქვემოთ გრაფიკებში მოცემულია მონაცემების მაქსიმუმისა და მინიმუმის სხვაობების (ამპლიტუდის) განაწილება. ეს განაწილებები ძალიან ჰგავს აბსოლუტური მნიშვნელობების განაწილებას, თუმცა უფრო თვალსაჩინოს ხდის მნიშვნელობების ტენდენციას.



სურათი 8



სურათი 9



სურათი 10

შესამჩნევია, რომ მიღებული მონაცემთა სიმრავლის განაწილება ცოტათი გადახრილია მარცხნივ, რაც ნიშნავს, რომ ხმები ისევ ხმადაბლაა და კიდე უფრო მაღალ ხმაზე შეიძლებოდა გარდაქმნა. ასევე რამდენიმე გამონაკლისი (outliers) ისევ გვხვდებოდა მონაცემებში. მიუხედავად ამისა ვივარაუდეთ, რომ ასეთი მცირე განსხვავება გადახრაში და გამონაკლირების ცოტა რაოდენობა არ გამოიწვევდა ქსელის შედეგის გაფუჭებას. ეს ვარაუდი ექსპერიმენტულადაც დამტკიცდა და დაახლოებით 20 ათასი იტერაციის შემდეგ, ქსელის მიერ დაგენერირებულ ხმებში გამოიკვეთა სიტყვები.

აღსანიშნავია რომ, წვრთნის დაწყებამდე გასათვალისწინებელია აუდიო ფაილების შესაბამისი მატრიცების მაქსიმალური მნიშვნელობის გამოთვლა და შესაბამის ველში მითითება. წვრთნის პროცესში ამ მატრიცის ელემენტების მნიშვნელობები აისახება -1 დან 1-მდე მნიშველობებში, რათა სხვადასხვა ტიპის მონაცემებისთის ერთნაირად იმუშავოს ქსელის წვრთნამ. გასაკვირია, რომ ამის მიუხედავად მონაცემებს მაინც დასჭირდა გარდაქმნა და მნიშვნელობების ხელოვნურად გაზრდა, რათა ქსელის წვრთნას შედეგი გამოეღო.

მომავალში ვგეგმავთ ამ მონაცემების უკეთეს გაფილტვრასა და ნორმალიზებას. ექსპერიმენტულად დადასტურდა, რომ მონაცემების სიმრავლის ნორმალიზება და მისი მნიშვნელობების სიმრავლის გადახრის სიახლოვე სტანდარტულ გადახრასთან პირდაპირ კავშირშია წვრთნის სიჩქარესთან.

მონაცემების მომზადების მეტად მნიშვნელოვანი ნაწილია ასევე ქართული წინადადებების სწორად შერჩევა. Tacotron 2 ქსელის გამოყენებისთვის საჭიროა, რომ მონაცემთა სიმრავლეს ჰქონდეს თავისი ანბანი და ყველა სიმბოლო იყოს წარმოდგენილი ამ ამბანიდან. შესაბამისად, ქართულ წინადადებებში დაუშვებელია სხვა ენის ანბანის სიმბოლოები. ასეთი წინადადებები შესაბამისად გავფილტრეთ და ამოვიღეთ მონაცემთა სიმრავლიდან.

ზემოთ აღვნიშნეთ, რომ ხმოვანი ფაილების სიგრძე უნდა იყოს 1-დან 10 წამამდე ხანგრძლივობის. ინგლისური მონაცემთა სიმრავლის ანალიზის შემდეგ დავადგინეთ, რომ წინადადებები 30-დან 170-მდე სიმბოლოს მოიცავდა. ჩვენი მონაცემების შეგროვების პროცესში დავაკვირდით, რომ 130-ზე მეტი სიგრძის წინადადებების შესაბამისი აუდიო ჩანაწერი ზედმეტად გრძელი გამოდიოდა. ამის გამო შევამცირეთ ასეთი წინადადებების სიგრძე.

ქართული სიმბოლოების გარდა ანაბში დასაშვებია არაბული ციფრების გამოყენებაც, თუმცა, ასევე საჭიროა ასეთი წინადადების ორ ნიმუშად დაწერა, ერთის არაბული ციფრებით, ხოლო მეორის შესაბამისი ქართული ასოებით. ეს საჭიროა იმისთვის, რომ შემდგომში ქსელმა ტექსტში არსებული სიმბოლოები გაარჩიოს და წაიკითხოს.

ასევე მნიშვნელოვანია ტექსტში არსებული აბრევიატურების სრული ფორმის დაწერაც წინადადებებში, რათა ქსელმა იცოდეს შემოკლებულის და სრული სიტყვის წაკითხვაც.

ქვემოთ არის მოცემული რამდენიმე მაგალითი ტექსტური მონაცემთა სიმრავლიდან:

01323-5eedffd20f29fa30c8e8c84f|ძირითადად ესენი არიან რეგიონებიდან დროებით ჩამოსული, სტუდენტები, მუშები, გლეხები და ა.შ.|ძირითადად ესენი არიან რეგიონებიდან დროებით ჩამოსული, სტუდენტები, მუშები, გლეხები და ასე შემდეგ.

01325-5eee00000f29fa30c8e8c854|წყალდიდობა იცის გაზაფხულზე და ზაფხულის დასაწყისში, წყალმცირობა — ზამთარში.|წყალდიდობა იცის გაზაფხულზე და ზაფხულის დასაწყისში, წყალმცირობა — ზამთარში.
01328-5eee002a0f29fa30c8e8c85a|მდინარე მტკვრის ხეობის ფსკერი, ქალაქის ფარგლებში, მერყეობს ზღ. დ 425 მ-დან .|მდინარე მტკვრის ხეობის ფსკერი, ქალაქის ფარგლებში, მერყეობს ზღვის დონიდან ოთხას ოცდახუთი მეტრიდან.

ველები | ამ სიმბოლოთია გამოყოფილი, პირველი ველი არის ფაილის სახელი, მეორე - ციფრების შემცველი ტექსტი და აბრევიატურა, ხოლო მესამე ველი სრულად დაწერილი ქართული ტექსტია.

7. Waveglow ქსელის წვრთნა და შედეგების შეფასება

Waveglow ქსელის წვრთნისთვის ავიღეთ Nvidia-ს ღია პროექტი [10]. ეს პროექტი დაწერილია pytorch ფრეიმვორკზე. წვრთნის გასაშვებად საკმარისია მონაცემების სიმრავლის გამზადება და კონფიგურაციის მორგება საწვრთნელ მანქანაზე.

მონაცემთა მომზადება მარტივია, საკმარისია მხოლოდ აუდიო ფაილების სიის შექმნა და რამდენიმე, კერძოდ 10, ფაილის გამოყოფა შუალედური შედეგების შეფასებისთვის. რადგან Waveglow არის გენერაციული ტიპის ნეირონული ქსელი, მას არ სჭირდება ვალიდაციისა და ტესტის მონაცემთა სიმრავლეები.
წვრთნის პროცესი შემდეგნაირად მუშაობს: ყოველი აუდიო ფაილიდან პროგრამულად ხდება მელის სპექტროგრამის ამოღება, შემდეგ, ქსელი ცდილობს ამ სპექტროგრამიდან აუდიო ფაილის დაგენერირებას. ცდომილების ფუნქცია (loss function) აფასებს რამდენად ახლოსაა ორიგინალი ხმა დაგენერირებულთან. ამის შემდეგ, ქსელი ცდილობს შედეგის ოპტიმიზაციას.

ასევე მნიშვნელოვანია ქსელის წვრთნის პარამეტრების სწორად შერჩევა. ამაზეა დამოკიდებული რამდენად მცირე დროში გაიწვრთნება ქსელი და რამდენად კარგი იქნება მის მიერ დაგენერირებული ხმის ხარისხი.

batch_size არის პარამეტრი, რომელიც მიუთითებს ერთდროულად რამდენ აუდიო ფაილს გაატარებს ქსელი წვრთნისას. რაც უფრო დიდია ეს რიცხვი, მით მეტ ვიდეო ბარათის მეხსიერებას მოიხმარს ქსელი, მაგრამ უფრო ნაკლებადაა შესაძლებელი გამონაკლისმა (outlier) იმოქმედოს ცდომილების ფუნქციაზე და შესაბამისად, უფრო კარგი ხარისხით მიიღება დაგენერირებული ხმა. ეს პარამეტრი უნდა მივუთითოთ ჩვენი რესურსის შესაბამისად. მაგალითად Nvidia Tesla V100 ვიდეოდაფის შემთხვევაში ოპტიმალური მნიშვნელობა იყო 24.

fp16_run - ეს არის ვიდეოდაფის მიერ ოპერატიული მეხსიერების უფრო ეფექტური გამოყენების ტექნოლოგია. ეს დამოკიდებულია თავად ვიდეოდაფაზე და თუ ამის მხარდაჭერა აქვს, მაშინ ორჯერ იმატებს მისი წარმადობა.

learning_rate - ეს პარამეტრი არის სწავლის ფუნქციის გრადიენტის ბიჯის გამსაზღვრელი. რაც უფრო პატარაა, მით უფრო ნელა მიდის წვრთნა, თუმცა, შესაძლებელია უკეთესი შედეგის მიღება.

max_wav_value - ეს პარამეტრი აღნიშნავს აუდიო ფაილების შესაბამისი მატრიცების მაქსიმალურ მნიშვნელობას. ის გამოიყენება წვრთნის პროცესში ამ მატრიცების წევრების მნიშვნელობის ასახვისთვის, რიცხვები კი შეიძლება ვარირებდეს -1 დან 1 მდე.

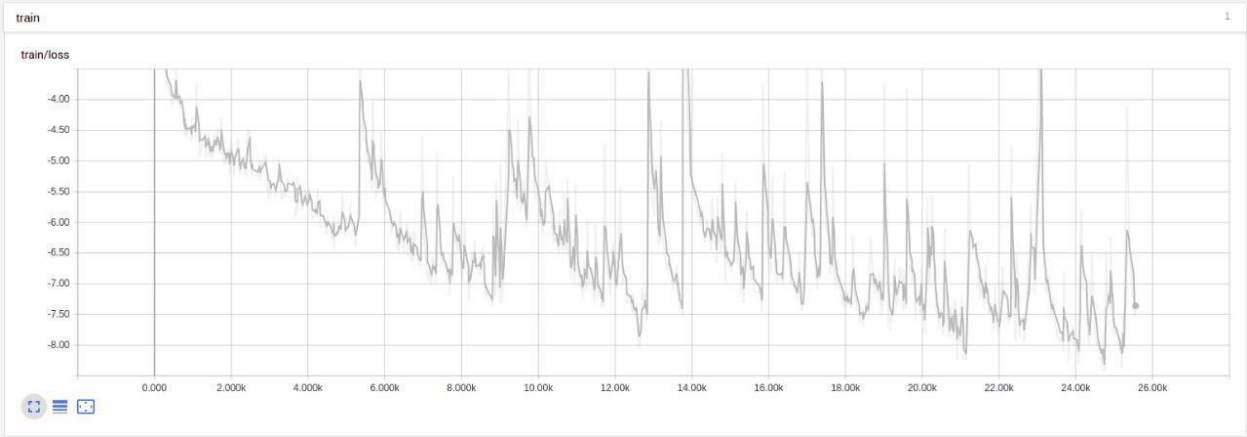
ქვემოთ მოცემულია (სურათი 11) მოდელის წვრთნის კოდის ნიმუში. თვითოეული ეპოქისთვის აღებულია ერთდროულად ქსელის წვრთნისთვის გათვალისწინებული რაოდენობის აუდიო ფაილები და მათი შესაბამისი მელის სპექტროგამები. ამის შემდეგ ორივე გადაეცემა მოდელს როგორც პარამეტრი და მოდელი აგენერირებს სპექტროგრამას. ხოლო დანაკარგის ფუნქციის მეშვეობით ხდება იმის შეფასება რამდენად სწორი იყო მიღებული შედეგი.

```
for epoch in range(epoch_offset, epochs):
    print("Epoch: {}".format(epoch))
    for i, batch in enumerate(train_loader):
        model.zero_grad()

        mel, audio, filename = batch
        mel = torch.autograd.Variable(mel.cuda())
        audio = torch.autograd.Variable(audio.cuda())
        outputs = model((mel, audio))

        loss = criterion(outputs)
        if num_gpus > 1:
            reduced_loss = reduce_tensor(loss.data, num_gpus).item()
        else:
            reduced_loss = loss.item()

        if fp16_run:
            with amp.scale_loss(loss, optimizer) as scaled_loss:
                scaled_loss.backward()
        else:
            loss.backward()

        optimizer.step()
```

სურათი 11

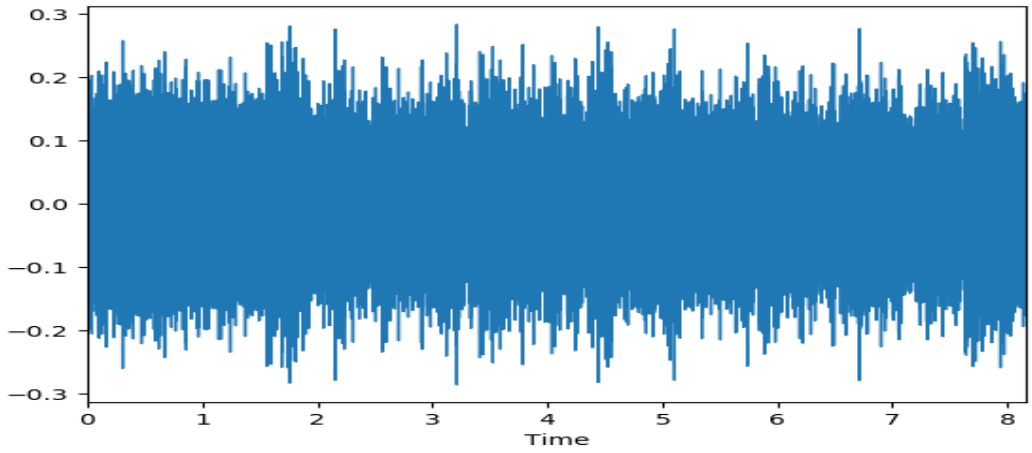
პარამეტრების სწორად შერჩევის შემდეგ ჩავატარეთ მრავალი ექსპერიმენტი.

ზემოთ აღწერილი მიზეზების გამო, თავდაპირველად წვრთნა არაეფექტურად განხორციელდა და დანაკარგის (ცდომილების) ფუნქციის მნიშვნელობა იძლეოდა დიდი მაჩვენებელს. როგორც გაირკვა ამ პროექტის ავტორების კომენტარებიდან, დანაკარგის ფუნქციის მაჩვენებელი უნდა დაწყებულიყო 0-1 დან და მისი მნიშვნელობა უნდა შემცირებულიყო -8 -9 მდე. ხოლო არასწორი მონაცემების პირობებში ეს მნიშვნელობა ხდებოდა 50-დან 100-მდე, რაც პირდაპირ მეტყველებდა მონაცემთა სიმრავლის შეცდომაზე. შედარებით უკეთესი მონაცემების პირობებში მხოლოდ პერიოდულად ხდებოდა ამ მნიშვნელობის "ახტომები", როგორც ეს ქვემოთ ნახაზშია (სურათი 12) მოცემული.



სურათი 12.

არასწორად გაწვრთნილი ქსელის მიერ მიღებული სპექტროგრამა (სურათი 13):



სურათიო 12

იმ შემთხვევაში, როცა მონაცემები ნორმალიზებულია, დანაკარგის ფუნციის გრაფიკი თანაბრად ეშვება ნიშნულისკენ.

ჩვენს შემთხვევაში ნორმალიზებულ მონაცემებზე 100 000 იტერაციის შემდეგ ფუნქციის ნიშნული ჩამოვიდა -7 -8-ის ფარლებში. ამ წვრთნას Nvidia GTX 1660 Ti ვიდეოდაფაზე დასჭირდა 2,5 დღე.

თავდაპირველად, რამდენიმე წარუმატებელი ექსპერიმენტის შემდეგ გადაწყდა, რომ ქსელის კოდი შეგვეცვალა ისე, რომ ჩვენ მიერ მითითებული ნიშნულის შემდეგ ქსელის შედეგებზე გადატარებულიყო საცდელი მონაცემები და შემოწმებულიყო მიმდინარე მდგომარეობის ეფექტურობა. ეს ნიშნულიც ქსელის ერთ-ერთი პარამეტრია და ჩვენ გამოვიყენეთ 2000. რაც ნიშნავს იმას, რომ ყოველი 2000 იტერაციის შემდეგ სრულდებოდა გარკვეული პროგრამა, რომელიც საცდელი მელის სპექტროგრამებისთვის აგენერირებდა აუდიო ფაილებს. ამ აუდიო ფაილებისთვის ასევე გენერირდებოდა ჩვეულებრივი სპექტროგრამა მეტი თვალსაჩინოებისთვის და ინახებოდა. წვრთნის პროცესში ნიშნულის მიღწევისას საკმარისი იყო გარკვეულ ფოლდერში ჩახედვა, ვიზუალურად სპექტროგრამით იყო შესაძლებელი შედეგის დანახვა და შემდეგ ხმის მოსმენითაც გადამოწმება. თუ კვლევის დასაწყისისთვის საჭირო იყო წვრთნის პროცესის შეწყვეტა და შეფასება ცალკე პროგრამის საშუალებით, საბოლოოდ უშედეგო ექსპერიმენტის აღმოჩენა დროულად ხდებოდა, რამაც კვლევის პროცესი საგრძნობლად აჩქარა.

8. Tacotron2 ქსელის წვრთნა და შედეგების შეფასება

Waveglow ქსელის წვრთვნისთვის ავიღეთ Nvidia-ს ღია პროექტი [9]. ეს პროექტი დაწერილია pytorch ფრეიმვორკზე. წვრთნის გასაშვებად საკმარისია მონაცემების სიმრავლის გამზადება და კონფიგურაციის მორგება საწვრთნელ მანქანაზე.

მონაცემების მომზადებისას, მისათითებელია შესაბამისი აუდიო ფაილები და ტექსტური ფაილი, სადაც მომზადებულია ყველა წინადადება. ამ ქსელის მიზანია ტექტის გარდაქმნა მელის სპექტროგრამაში. მისი წვრთნის პროცესი ძალიან ჰგავს Waveglow ქსელის წვრთნის პროცესს. აუდიო ფაილებიდან პროგრამულად იღებს სპექტროგამებს და ტექსტიდან დაგენერირებულ სპექტროგრამებს ადარებს ერთმანეთს. შემდეგ კი ცდილობს გაასუმჯობესოს შედეგი.

უნდა აღინიშნოს, რომ Tacotron 2-ში მოიაზრება სრული პროცესი, თუმცა ტექსტიდან სპექტროგრამის დაგენერირების შემდეგ, აუდიო ფაილის დასაგენერირებლად გამოიყენება წინასწარ გაწვრთნილი WaveGlow-ს ან მისი რომელიმე ანალოგის მოდელი.

წვრთნის პარამეტრები ისეთივეა, როგორც WaveGlow-ზე. ესენია:

1) batch_size
2) fp16_run
3) learning_rate
4) max_wav_value

მათი დანიშნულება ისეთივეა და შედეგზეც შესაბამისად აისახება.

Tacotron 2-ისთვის საჭიროა თითოეული ენისთვის დაიწეროს რამდენიმე პროგრამა, რომელიც წვრთნის დროს წინადადებებს სწორ ფორმატში გარდაქმნის. ეს პროგრამა შედგება კონვერტორებისა და ფილტრებისგან, რომლებიც თითოეული წინადადებისთვის სრულდება. ეს ფილტრებია:

1) სტანდარტული აბრევიატურების გაშლა - დგება აბრევიატურების და მისი გაშლილი ფორმების ასოციაციური რუქა და ეს სიტყვები იცვლება შესაბამისად;
2) რიცხვების გარდაქმნა სიტყვებად;
3) არასაჭირო სიმბოლოების მოშორება;
4) მიყოლებით რამდენიმე ჰარის სიმბოლოებიდან მხოლოდ ერთის დატოვება.

პირველი და მეორე პუნქტის ქართული ენისთვის რეალიზება რთულია, რადგან ქართულში ბრუნვისა და პირის ნიშნები სიტყვებშივე ემატება, შესაბამისად, მონაცემთა მომზადების დროს რიცხვები და აბრევიატურები სრულად, გაშლილად ქართულ ტექსტად არის დასაწერი.

მესამე პუნქტში იგულისხმება ისეთი სიმბოლოების მოშორება, რომლებიც არ შედის სტანდარტულ სასვენ ნიშნებში და არანაირ გავლენას არ იქონიებს ტექსტის წაკითხვაზე. ასეთებია სხვადასხვა სახეობის ორმაგი ბრჭყალები, დაბლატირე და სხვა.

მონაცემების მომზადების შემდეგ კვლავ ჩატარდა რამდენიმე ექსპერიმენტი და საწყის შედეგებში რაიმე ძირეული პრობლემა არ გამოვლინდა. თუმცა წვრთნის დრო არ არის საკმარისი იმისთვის, რომ შეფასდეს ქსელის ხარისხი.

ამ მოდელზე იგეგმება მუშაობის გაგრძელება ნაშრომის შემდეგაც.

9. არსებული შედეგის გაუმჯობესების მეთოდები

9.1. მონაცემების შეგროვება და დამუშავება

მონაცემთა სიმრავლე დაახლოებით 22 საათის ხანგრძლივობისაა, იმ ხმოვანი ფაილების ჩათვლით, რომელიც ნორმალიზების შედეგად არ ხვდება საბოლოო სიმრავლეში. სასურველი იქნებოდა, რომ მინიმუმ 24 საათის ხანგრძლივობის შედეგი მიგვეღო ამ ნაშრომის ფარგლებში, ასე რომ ხმების ჩაწერა მომავალშიც გაგრძელდება და შეგროვდება უფრო მეტი. ასევე სასურველია, რომ გვქონდეს არა მარტო ქალის, არამედ მამაკაცის ხმაც.

შეგროვილი მონაცემების დამუშავებაც უფრო ხარისხიანად უნდა დამუშავდეს. კერძოდ, უნდა გადამოწმდეს ყველა აუდიო ფაილი ცალ-ცალკე, რომ არ აღმოჩნდეს გამონაკლისი (outlier) და არ შეუშალოს წვრთნის პროცესს ხელი.

ვვარაუდობთ, რომ შეგროვილ წინადადებებში ბიოგრაფიული და გეოგრაფიული შეინაარსის ტექსტი შეიცვალოს უფრო ზოგადი ტექსტით, რადგან ამ ტიპის წინადადებებში არის ბევრი აბრევიატურა, რომლიც სხვაგან არსად არ გვხვდება და ასევე შეიცავს დიდი რაოდენობით თარიღებსა თუ ციფრებს.

აუდიო ფაილების ჩაწერა საკმაოდ შრომატევადი და რთული საქმეა. ასევე, გასათვალისწინებელია ის ფაქტორიც, რომ თუ გვსურს მაღალი ხარისხის ჩანაწერების გაკეთება, სჯობს ამ მიმართულებით უკვე გამოცდილმა ადამიანმა ჩაწეროს ხმა. თუმცა აღსანიშნავია რომ, გამოცდილების მქონე ადამიანის ჩანაწერების შეგროვება დიდ ხარჯებთანაა დაკავშირებული. ტელევიზიებსა და რადიო მაუწყებლებში კი არსებობს უკვე მაღალი ხარისხის ჩანაწერები, რომლის ხანგრძლივობაც შეიძლება რამდენიმე ასეულ საათს აღწევს. მაგალითისთვის github-ზე შეხვდებით რუსული ენისთვის იმპლემენტირებულ Tacotron-ს, რომელიც სწორედ რედიო მაუწყებელის მიერ გამოყოფილ 700 საათიანი მონაცემთა სიმრავლით არის გაწვრთნილი და აქვს ძალიან კარგი შედეგი.

ჩვენ ვგეგმავთ თანამშრომლობის გაგრძელებას მონაცემების მოძიებასთან დაკავშირებით რადიო მაუწყებლებსა და ტელევიზიებთან.

9.2. აუდიოს გენერაციის ქსელის შედეგის გაუმჯობესება

ქსელის დღევანდელი სახით შედეგის მიღებას 100 000 იტერაცია და 15 ეპოქა დასჭირდა. წვრთნა მიმდინარეობდა საკმაოდ სუსტ ვიდეოდაფაზე და შესაბამისად 2,5 დღის შედეგად მივიდა ამ შედეგამდე. ვვარაუდობთ 100-150 ეპოქის შემდეგ შედეგი ისეთი იქნება, რომ შესაძლებელი იქნება მისი გამოყენება ღია ან კომერციულ პროგრამულ უზრუნველყოფაში.

ვფიქრობთ, რომ კარგი იქნება, თუ ექსპერიმენტები ჩატარდება სხვა აუდიოს გენერაციის მოდელებზე და შეფასდება მათი ეფექტურობა. იგივე WaveNet-ი ან რომელიმე მისი მოდიფიკაცია. ამ ტიპის ქსელები საკმაოდ დიდი მრავალფეროვნებით გამოირჩევა და ყველა ჭრის გარკვეული ტიპის პრობლემებს. ზოგიერთი არის მარტივი არქიტექტურის და ნაკლები მონაცემთა სიმრავლითაც ეფექტურად იწვრთნება, როგორც WaveGlow, ზოგიერთი მათგანი შედარებით ზოგადი მოხმარებისაა, თუმცა საწვრთნელი მონაცემიც მეტი სჭირდება, როგორიცაა WaveNet, ზოგი კი აუდეოს აგენერირებს რეალურ დროზე უფრო ჩქარა, როგორიცაა nv-wavenet [8].

9.3 Tacoton 2 ქსელის გაწვრთნა

ამ ქსელის შედეგები ქართული ენისთვის ჯერ საკმაოდ ნედლია და სულ რამდენიმე ექსპერიმენტია ჩატარებული. თუმცა ეს ექსპერიმენტები საკმარისია იმისთვის, რომ იმედიანად ვიყოთ. მომავალში დაგეგმილია ამ წვრთნის გაგრძელება და მიღებული შედეგის განთავსება ვებ გვერდზე.

9.4. უახლესი კვლევების შესწავლა

მოცემულ კვლევაში არჩეული ნეირონული ქსელი დღესდღეობით ყველაზე პოპულარული და ფართოდ გამოყენებადია, უნდა აღინიშნოს რომ მუშაობა ამ კონკრეტული მიმართულებით უფროდაუფრო აქტიურ ფაზაში გადავიდა და ყოველკვირეულად ახალი და უკეთესი

კვლევები იდება ინტერნეტში. აღსანიშნავია, რომ ამ პერიოდში (კვლევის ჩატარების) გამოქვეყნდა 10-მდე ახალი სამეცნიერო ნაშრომი ხმის ჩანაწერებთან დაკავშირებით. .

ამ 10 ნაშრომიდან ყველაზე მეტად ყურადღება მისაქცევი არის კომპანია facebook-ის კვლევა [15]. ეს ნაშრომი უფრო მეტად შედეგების ჩვენების ხასიათს ატარებს, ვიდრე პრაქტიკული გამოყენების, თუმცა საინტერესო და ყურადღება გასამახვილებელია ამ სფეროში მნიშვნელოვანი შედეგების მიღწევის თვალსაზრისით.

9.5. საიტი tts.ge

დავგეგმეთ, რომ მონაცემთა სიმრავლე, რომელიც მოგროვდა ამ კვლევის ფარგლებში, განთავსდეს ვებ-გვერდზე და ყველასთვის ხელმისაწვდომი იყოს. მონაცემებში იქნება როგორც აუდიო ფაილები, ასევე წინადადებების სიმრავლე, რომელიც წინასწარ არის დამუშავებული შესაბამის ფორმატში.

საიტზე ასევე განთავსდება წინასწარ გაწვრთნილი ქსელის წონები ორივე მოდელისთვის, რომელთა გამოყენებაც ასევე ყველას შეეძლება წვრთნის გარეშე.

ასევე გვინდა საიტზე იყოს შესაძლებელი ინტერაქტიულ რეჟიმში ტექსტის შეყვანა და მისი შესაბამისი აუდიო ფაილის მიღება ონლაინში. ეს მოითხოვს დიდ გამოთვლით რესურსს და დაკავშირებულია საკმაოდ დიდი რაოდენობის თანხებთან, ამიტომ შესაძლებელია, პერიოდულად დროებით ჩავრთოთ ეს ფუნქცია.

9.6 გამოყენების ასპექტები

მოცემული კვლევის როგორც თეორიული ისე პრაქტიკული ღირებულება მეტად მნიშვნელოვანია. მისი პრაქტიკული გამოყენება შეიძლება მრავალ სფეროში საკმაოდ ეფექტურად. ყველაზე მარტივი გამოყენება იქნება ვებ-გვერდი, სადაც ნებისმიერი მსურველი შეძლებს ტექსტის შეყვანას და მიიღებს შესაბამის აუდიო ფაილს. ეს გამოყენება უფრო ზოგადი ხასიათისაა და შეიძლება გარკვეული ტიპის ავტომოპასუხისთვის რაიმე წინადადებების ჩაწერა მოხდეს მარტივად ამ გზით.

კიდევ ერთი გამოყენება შეიძლება იყოს აპლიკაცია, რომელიც მობილურ ტელეფონზე მოსულ ქართულ შეტყობინებებს წაიკითხავს. ეს განსაკუთრებით გამოსადეგი იქნება მანქანის მართვისას, ვარჯიშისას და მრავალ სხვა სიტუაციაში, როდესაც ადამიანს არ შეუძლია ტექსტის წაკითხვა.

რა თქმა უნდა, მსგავსი აპლიკაცია გამოდგება უსინათლოებისთვისა და მცირედმხედველი პირებისთვის ქართულენოვანი ტექსტის წასაკითხად. დღესდღეობით არსებული პროგრამები (ეკრანის წამკითხველები NVDA, JOWS) მხედველობის პრობლემების მქონე და უსინათლო ქართველებისთვის საჭიროებებს დამაკმაყოფილებლად ვერ პასუხობს.

პროგრამის გამოყენება შეეძლებათ საინფორმაციო წყაროებსაც, რომლებიც შეძლებენ ტექსტურ სიახლეს დაურთონ აუდიო მასალაც. ეს მასალა შესაძლებელია ავტომატურადაც კი დაგენერირდეს.

დასკვნა

მოცემული კვლევის შედეგად შესრულდა დასახული მიზნებიდან პირველი სრულად, ხოლო მეორე მიზანი ნაწილობრივ. ქსელის ნაწილი, რომლიც აგენერირებს სპექტროგრამიდან აუდიო ფაილს გაწვრთნილია და მისი წონები ხელსმისაწვდომი გახდება ყველასთვის ვებ გვერდის გაშვებისას. რაც შეეხება მეორე ქსელს, რომელიც ტექსტს გარდაქმნის სპექტროგრამაში, მასზე ჩატარებულია ექსპერიმენტები და მონაცემები მომზადებულია.

მონაცემთა სიმრავლე, რომელიც უკვე დამუშავებული და შეკრებილია, განთავსდება ვებ გვერდზე ნებისმიერი მსურველისთვის. ვფიქრობთ, რომ ეს მონაცემები არის ძალიან დიდი ნაბიჯი ქართული ენის ტექნოლოგიურ სფეროში გამოჩენისთვის. ქართულ ენას სამწუხაროდ ძალიან იშვიათად ამატებენ უახლეს ტექნოლოგიებში, როგორიცაა ხმოვანი ასისტენტი, ჭკვიანი თარჯიმანი და სხვა. შეგროვილი მონაცემებით ყველა მსგავსი საჭიროების დაკმაყოფილება იქნება შესაძლებელი, რაც ხელს შეუწყობს ხელოვნური ინტელექტის გამოყენებასა და განვითარებას საქართველოში. რაც მთავარია, ქართველი მომხმარებლებიც უმოკლეს დროში მიიღებენ ტექნოლოგიურად მოწინავე პროდუქტებს.

ჩვენ ვაპირებთ ჩვენ მიერ მიღებული შედეგის გაუმჯობესებას, ეს კვლევა დღეს ისევ გრძელდება და პერიოდულად ვატარებთ მონაცემთა ანალიზს. ვფიქრობთ, რომ ამ ტექნოლოგიას აქვს უამრავი გამოყენება და გვინდა რამდენიმე მათგანში მაინც დავნერგოთ. კვლევის პროცესში შექმნილი ხელსაწყობი და გამოცდილება, რომელიც აისახება რჩევების და შეზღუდვების სახით კოდში, ყველასთვის იქნება გამოსადეგი ვინც ამ მიმართულებით მოისურვებს მუშაობას.

სასურველია მომავალში უფრო მეტი კვლევა ჩატარდეს ამ მიმართულებით და ქართული ენა სრულიად იყოს წარმოდგენილი თანამედროვე ტექნოლოგიებში.

გამოყენებული ლიტერატურა

[1] https://arxiv.org/pdf/1710.08969.pdf

[2] https://arxiv.org/pdf/1712.05884.pdf

[3] https://arxiv.org/pdf/1811.00002.pdf

[4] https://awesomeopensource.com/project/Rayhane-mamah/Tacotron-2

[5] https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

[6] https://en.wikipedia.org/wiki/Hidden_Markov_model

[7] https://github.com/Kyubyong/dc_tts

[8] https://github.com/NVIDIA/nv-wavenet/

[9] https://github.com/NVIDIA/tacotron2

[10] https://github.com/NVIDIA/waveglow

[11] https://github.com/dabulashvili/dataset-collector

[12] https://keithito.com/LJ-Speech-Dataset/

[13] https://medium.com/@evinpinar/wavenet-implementation-and-experiments-2d2ee57105d5

[14] https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

[15]https://medium.com/syncedreview/facebooks-highly-efficient-new-real-time-text-to-speech-system-runs-on-cpus-4471e4c78268

[16] https://nv-adlr.github.io/WaveGlow

[17] https://openai.com/blog/glow/

[18] https://www.ijcaonline.org/research/volume130/number3/kayte-2015-ijca-906965.pdf