სტატისტიკური ენის მოდელი

ალექსანდრე მელაძე, კონსტანტინე კამკამიძე

საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

განხილულია სტატისტიკური ენის მოდელის შექმნა, დამუშავება და მისი აღწერა N-gram-ების საფუძველზე, მისი ძირითადი მახასიათებლები, გამოყენების არეალი, პრობლემები და მათი გადაჭრის გზები. მოყვანილია მაგალითები თუ როგორ უნდა შეიქმნას ასეთი მოდელი ქართული ენის მაგალითზე. შემოთავაზებულია ყველა საჭირო ალგორითმის როგორც თეორიული, ასევე პროგრამული გადაწყვეტილება. განხილულია ალგორითმები, რომელთა გამოყენებითაც შესაძლებელია ქართული ენის სტატისტიკური ენის მოდელის შექმნა, დამუშავება და მისი პრაქტიკაში გამოყენება. მოყვანილია შესაბამისი პროგრამული კოდის ფრაგმენტები N-gram მოდელის ასაგებად.

საკვანძო სიტყვები: სტატისტიკური ენა. მოდელი. N-gram

1. შესავალი

სტატისტიკური ენის მოდელი მნიშვნელოვანი მოდელია ისეთი სისტემებისთვის როგორებიცაა: ხმის ამომცნობი სისტემა, ტექსტის ამომცნობი სისტემა, ტექსტის გრამატიკული ანალიზატორი, ტექსტის მთარგმნელი სისტემები და სხვ. [1-4]. ასეთი მოდელი აგროვებს ცოდნას და შემდგომ ამ ცოდნაზე დაყრდნობით იღებს გადაწყვეტილებებს. ცოდნა მიიღება სხვადასხვა ინფორმაციაზე დაყრდნობით, გროვდება ინფორმაცია, რომლის გამოყენებითაც სისტემა იღებს გადაწყვეტილებას თუ რა მოსალოდნელი სიტყვა თუ სიტყვები იქნება შეყვანილი. ასეთი სახის დამთხვევა ამარტივებს ტექსტის ამოცნობის პროცესს [4]. ჩვენი ნამუშევარი მოიცავს სტატისტიკური ენის მოდელის შექმნას, მის დამუშავებას და გამოყენებას ქართული ენის მაგალითზე.

2. ძირითადი ნაწილი

2.1. N-gram მოდელის აიგება და მარკოვის ვარაუდი

ტექსტის ამოცნობა ერთერთი ყველაზე კომპლექსური და რთული პრობლემაა, რადგან ის დამოკიდებულია ენის გრამატიკაზე, მის ლინგვისტურ მახასიათებლებზე და ცოდნაზე, რომლის შექმნა კომპიუტერული ტექნიკისთვის საკმაოდ რთულია. ჩვენ დღეს-დღეობით გვაქვს რამდენიმე დახვეწილი სისტემა რომელიც გამოიყენება ჩვენ სმარტფონებში [2,4], რომლის დახმარებით ჩვენ სმარტფონზე შეხების გარეშე ხმით შეგვიძლია ტელეფონს ბრძანებები გადავცეთ, როგორიცაა: დარეკვა, ინფრმაციის მოძიება თუ სმს-ის გაგზავნა. მსგავსი სისტემები გამოიყენება ავტომობილებში, სადაც ხმის საშუალებით შესაძლებელია ნავიგაციის განსაზღვრა და ა.შ. ასეთი სისტემები საკმაოდ პოპულარულია და ხდება მათი დღითიდღე დახვეწა.

ხმის ამომცნობ სისტემებში ერთერთ ყველაზე მნიშვნელოვან როლს თამაშობს ენის სტატისტიკური მოდელი. ასეთი მოდელი აიგება N-gram ების საშუალებით. N-Gram ზე დაყრდნობით, ხმის ამომცნობი სისტემა იღებს გადაწყვეტილებას მოსალოდნელი სიტყვის

შესახებ და უმარტივდება ამოცნობის პროცესი [8]. პირველად N-gram მოდელი წარმოდგენილ იქნა Markov-ის მიერ, ამ მოდელის მიხედვით დაიწყეს ხმის ამომცნობი სისტემების აგება, ასევე ეს მეთოდი გამოიყენება სხვადასხვა მიმართულებით, მაგალითად Shannon-მა პირველად გამოიყენა ეს მეთოდი ინფორმაციის თეორიაში [1].

N-gram მოდელის მთავარი იდეა არის ის, რომ მას გააჩნია სიტყვები და ამ სიტყვების მოსვლის ალბათობები, იდეა არის რომ ნებისმიერი სიტყვა N-სიტყვა დამოკიდებულია მის წინა N-1 სიტყვაზე. როდესაც პირველად წარმოდგენილ იქნა N-gram მოდელი, მისმა ასეთმა მიდგომამ სიტყვებისადმი გამოიწვია ლინგისტების კრიტიკა. პირველად ის გააკრიტიკა ამერიკელმა ლინგვისტმა Noam Chonsky-იმ, რის გამოც გარკვეული პერიოდი ამ მოდელზე მუშაობა შეწყვეტილიც კი იყო, მანამ სანამ Jelinek-მა ხელახლა არ წარმოადგინა მოდელი 1971 წელს და პირველმა გამოიყენა ის ხმის ამომცნობ სისტემაში [1,3].

დღესდღეისობით N-gram მოდელი ყველაზე მნიშვნელოვანი მიდგომაა ხმის ამომცნობ სისტემებში. ასეთი მოდელის აგებისთვის ხდება წინასწარ მონაცემთა ბაზის განსაზღვრა. მონაცემთა ბაზა შედგება სიტყვებისაგან და სიტყვის მოსვლის ალბათო-ბებისგან. ხდება რამდენიმე გრამ- მოდელის აგება. როგორებიცაა: 1-gram(gram) მოდელი, 2-gram(Bigram) მოდელი, 3-gram(Unigram) მოდელი და ა.შ. N-gram რაოდენობამდე. ძირითადად თანამედროვე სისტემებში გამოიყნება მაქსიმუმ 5-gram მოდელი, უფრო დიდი მოდელის შექმნა მოითხოვს დიდი რაოდენობით რესურსს და მეხსიერებას.

N-gram -ი დამოკიდებულია მის ალბათურ მოდელზე, რომლის მიხედვითაც იღებს გადაწყვეტილებას, რამდენად მოსალოდნელია სიტყვის მოსვლა. განვიხილოთ მაგალითი. ვთქვათ გვაქვს წინადადება: „ბობი აგზავნის ინფორმაციას“. ეს წინადადება წარმოადგენს 3-gram მოდელს და ამ წინადადების მოსვლის ალბათობა გამოითვლება ასე:

P(ინფორმაციას|ბობი აგზავნის)= C(ბობი აგზავნის ინფორმაციას) / C(აგზავნის ინფორმაციას) (1)

როგორც (1)-დან ჩანს წინადადების მოსვლის ალბათობა დამოკიდებულია მის მიერ N-1 წინადადებაზე და ითვლება ცოდნის ბაზაში მოხვედრილი წინადადების რაოდენობა შეფარდებული მის წინ მოსული სიტყვის ალბათობაზე. ფორმალურად რომ ავღწეროთ, ვთქვათ გვაქვს W მოცემული წინადადება და ისტორიული სიტყვების რაოდენობა h, და ყველა ისტორიული სიტყვის მოსალადნელი ალბათობები W.წინადადება იყოფა სიტყვებად $w_1 w_2 w_3 \ldots . w_n$ ან w_1^n ხოლო წინადადების მოსალოდნელი ალბათობა გამოითვლება ფორმულით:

$$P(w_1 w_2 w_3 \ldots . w_n)$$

საბოლოოდ მივიღებთ:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2) \ldots P(w_n|w_1^{n-1})$$

$$=\prod_{k=1}^{n} P\left(w_k|w_1^{k-1}\right) \quad (2)$$

(2) ფორმულიდან ჩანს, რომ მოსალოდნელი სიტყვის ალბათობის გამოსათვლელად ჩვენ ვიყენებთ ისტორიულ სიტყვებს, ვაჯამებთ მათ მოსალოდნელ ალბათობებს და ახალ წინადადებას ვანიჭებთ შესაბამის ალბათობას [8,9]. ასეთი მიდგომა გამოიყენებოდა დიდი ხნის განმავლობაში, ეს გარკვეულწილად ქმნის პრობლემას, რადგან შესაძლებელია რამდენიმე წინადადება საერთოდ არ გვქონდეს ბაზაში, ამის გამო ყოველი შემდგომი

წინადადების მოსალოდნელი ალბათობა იქნება 0-ის ტოლი. მაგალითად ჩვენ შემთხვევაში თუ წინადადება „ბობი აგზავნის“ არ გვექნება ბაზაში, მაშინ წინადადება „ბობი აგზავნის ინფორმაციას“ ალბათობა ყოველთვის 0-ის ტოლი იქნება. ამ პრობლემის აღმოსაფხვრელად მოიფიქრეს, რომ მთლიანი ისტორიული წინადადებების ალბათობების ნაცვლად აეღოთ წინადადებაში ბოლო სიტყვების მოსალოდნელი ალბათობები [8] ანუ:

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1}) \quad (3)$$

ამ მიდგომას ეწოდა მარკოვის ვარაუდი (Markov assumption).

განვიხილოთ მაგალითი :

<s> მე ვარ ლექსო </s>

<s> ლექსო მე ვარ </s>

<s> მე ვცდილობ საინტერესო თემა დავწერო </s>

P(მე | <s>) = 2/3 =0.67 (ლექსო | < s>) = 1/3 =0.33 P(ვარ | მე) = 2/3 =0.67

P(<s> | ლექსო) = 1/2 =0.5 P(ლექსო | ვარ) = 1/2 =0.5 P(ვცდილობ | მე) = 1/3 =0.33

მოდელის დასწავლა

მსგავსი მოდელის შექმნა როგორიცაა 1-gam ,2-gram, 3-gram ….. N-gram, წარმოადგენს საკმაოდ დიდ სირთულეს. ასეთი ამოცანის გადასაჭრელად საჭიროა შემდეგი პრობლემების გადაჭრა [5,6,8]:

- სიტყვების ლექსიკონში არსებობა;
- არსებული ლექსიკონის N-gram -ებად ფორმირება;
- არსებული სიტყვების ალბათობების განსაზღვრა;
- დიდი მონაცემების დროული დამუშავება.

2.2. სიტყვების ლექსიკონში არსებობა

ქართული ენა საკმაოდ დიდი მონაცემებისგან შედგება. სრული ლექსიკონის ელექტრონული ფორმით ქონა ალბათ შეუძლებელია. არსებობს რამდენიმე ელექტრონული ლექსიკონი, რომელშიც საკმაოდ ბევრი ქართული სიტყვაა. ჩვენ შემთხვევაში საჭიროა მოვახდინოთ ლექსიკონის ფორმირება. სათითაოდ ყველა სიტყვის შეყვანა საკმაოდ დიდ რესურსს და დროს მოითხოვს, ამის თავიდან ასაცილებლად გაკეთდა პროგრამა, რომელიც ქართულ საიტებს ამუშავებს, იღებს ყველა განსხვავებულ ქართულ სიტყვას და ინახავს ლექსიკონში. ასეთი სისტემის ასაგებად საჭიროა მონაცემთა ბაზა.

N-gram მოდელის დასამუშავებლად დაყენებულია რელაციური მონაცემთა ბაზა MS SQL Serever. რადგან მონაცემები საკმაოდ დიდ რესურსს მოითხოვს, ასევე სიტყვის ამორჩევა უნდა ხდებოდეს ძალია სწრაფად. ამისთვის თითოეული N-გრამისთვის გამოყენებულ იქნა დამოუკიდებელი ცხრილი, რომლის გასაღებ ველს (primary key) წარმოადგენს სიტყვა.

2.3. არსებული ლექსიკონის N-gram ებად ფორმირება

ჩვენი მიზანია მონაცემთა ბაზის ფორმირების შემდეგ მისი შევსება ქართული საიტებიდან ამოღებული და დამუშავებული ინფორმაციით. ყველაზე დიდი ინფორმაცია, რომელიც რაღაც კუთხით ნორმირებულია, ინახება https://ka.wikipedia.org -ზე, რომელიც წარმოადგენს ქართულ ვიკიპედიას (ენციკლოპედიას). რადგან ვიკიპედია ენციკლოპედიაა, მისგან აღებული ინფორმაცია იქნება რეალურთან ახლოს. პროგრამას უნდა ჰქონდეს შემდეგი თვისებები, რათა კორექტულად დაამუშაოს ინფორმაცია:

- შეძლოს სრული ინფორმაციის წამოღება;
- მოახდინოს ყველა გვერდის დამუშავება;
- მოახდინოს საჭირო ინფრომაციის გაფილტვრა;
- მოახდინოს მხოლოდ ქართული ტექსტების ამორჩევა;
- წინადადებებიდან შექმნას N-gram მოდელი;
- შეინახოს დამუშავებული ინფორმაცია ცხრილებში;
- პროგრამის დასაწერად მაგალითები მოყვანილია c# ენაზე.

საიტის დასამუშავებლად და მისგან ინფორმაციის წამოსაღებად გამოყენებულია ბიბლიოთეკა HtmlAgilityPack, რომელიც არის საკმაოდ კარგი ხელსაწყო, რადგან ის გარდაქმნის მთლიან html ფაილს C# სთვის არსებულ XNode (xml) ტეგებში, ხოლო Xnode - ტეგების დამუშავება და მასში ძიებების განხორციელება შედარებით იოლია, html- თან შედარებით.

```
HtmlWeb web = new HtmlWeb();
HtmlDocument doc = web.Load(url.Url);
doc.DocumentNode.SelectNodes("//div[@id='mw-content-text']")
```

ლისტინგი 1: HtmlAgilityPack-ის საშუალებით საიტიდან ინფორმაციის წამოღება

როგორც აქედან ჩანს, გვიბრუნდება სრული სტრუქტურა და შემდგომ ვახდენთ ინფრომაციის გაფილტრვრას. ყველა გვერდის დასამუშავებლად ვიყენებთ გვერდის ავტომატური დამუშავების საშუალებას. ხდება ერთჯერადად საწყისი გვერდის განსაზღვრა. მიღებული საწყისი გვერდის დამუშავების მომენტში ხდება გვერდიდან ამოღებული ყველა მისამართის გაფილტვრა და მიღებული მისამართების დამუშავება. რადგან ვიკიპედიას თითოეულ სტატიაში შესული ინფორმაცია დამისამართებულია მთავარ წყაროზე, ჩვენ ვახდენთ ამ წყაროების დამუშავებას. ეს პროცესი რეკურსიულია და გვაძლევს საშუალებას მთლიანად დავამუშაოთ ვიკიპედიის ყველა გვერდი.

```
var friendlyUrls = node.SelectNodes("//a[@href]");
 foreach (var url in friendlyUrls)
  { string href = url.Attributes["href"].Value;
    if (href.Contains("/wiki/"))
       urls.Add(new QueueUrl($"{mainUrl}{url.Attributes["href"].Value}"));
 }
```

ლისტინგი 2. გვერდების დამუშავება

2.3. მხოლოდ ქართული ტექსტების ამორჩევა

რადგან ჩვენი მიზანია ქართული სტატისტიკური მოდელის შექმნა, ამიტომ ლექსიკონში უნდა შევინახოთ მხოლოდ ქართული სიტყვები. ვიკიპედიის გვერდები საკმაოდ მრავალრიცხოვან ინფორმაციას შეიცავს. ეს შეიძლება იყოს უცხოური სიტყვები, დასახელებები, სურათები და ა.შ. ჩვენმა სისტემამ რაღაც კუთხით უნდა გაფილტროს და გამოარჩიოს მხოლოდ ქართული სიტყვები. წინადადებიდან სიტყვების მიხედვით რომ განისაზღვროს თუ რომელ ენაზეა დაწერილი, ეს საკმაოდ რთული პროცესია. ჩვენ შემთხვევაში უფრო მარტივადაა საქმე, რადგან ჩვენ ვამუშავებთ მხოლოდ ისეთ სიტყვებს და წინადადებებს, რომელთაც თითოეული სიმბოლო (უნიკოდი) არის ქართული.

2.4. წინადადებებიდან N-Gram მოდელის შექმნა

ჩვენი მიზანია, რომ სისტემას შეეძლოს ნებისმიერი (1,….n) გრამ მოდელის შექმნა და დამუშავება. რადგან ვიკიპედია არის ქართული ენციკლოპედია და მისი მონაცემები საკმაოდ სანდოა, ჩვენ ვახდენთ მოდელის და ალბათობების ვიკიპედიაზე დაყრდნობით აგებას. ყოველი წინადადება გადის ვალიდაციას სისტემაში. პირველად ენიჭება მოსალოდნელი ალბათობა 0.1, შემდგომ თუ კვლავ შეგვხვდა ეს სიტყვა შესაბამისად იზრდება მისი ალბათობა. აიგება 1-gram და შემდგომ 2-gram, 3-gram მოდელები. ისინი იგება 1-gram-ის მონაცემთა ბაზაზე და ვიკიპედიაში არსებულ სიტყვათა ალბათობაზე დაყრდნობით. განვიხილოთ მაგალითი:

ვთქვათ ვიკიპედიიდან ამოვიცანით წინადადებები:

„დღეს კარგი დღეა“

„დღეს შაბათია“

„კარგი საქმის კეთების დღეა“

// n-gram -ის შექმნის მაგალითი

მაგალითზე პირველ რიგში მოხდება 1- gram სისტემის დამუშავება. შეიქმნება ცოდნის ბაზა. ამ შემთხვევაში სიტყვისთვის „დღეს“ ექნება მოსალოდნელი ალბათობა 0.2, რადგან 2-ჯერ შეგხვდა, ხოლო 2-gram-ის დამუშავების პროცესში გაითვალისწინება სიტყვა „დღეს“ მოსალოდნელი ალბათობა და მასზე დაყრდნობით შეიქმენა მისი ალბათობა.

3. დასკვნა

განხილულია სტატისტიკური ენის მოდელის აგება n-gram-ების გამოყენებით, აღწერილია ძირითადი კონცეფციები, მისი აგების მეთოდები და პროგრამული გადაწყვეტები, რომლის საშუალებითაც შესაძლებელია ქართული ენისთვის აიგოს სტატისტიკური მოდელი.

ლიტერატურა:

1. Algoet, P. H. and Cover, T. M. (1988). A sandwich proof of the Shannon-McMillan-Breiman theorem. The Annals of Probability, 16(2), 899–909.

2. Bacchiani, M., Riley, M., Roark, B., and Sproat, R. (2006). Map adaptation of stochastic grammars. Computer Speech & Language, 20(1), 41–68.

3. Bacchiani, M., Roark, B., and Saraclar, M. (2004). Language model adaptation with MAP estimation and the perceptron algorithm. In HLT-NAACL-04, pp. 21–24.

4. Bahl, L. R., Jelinek, F., and Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. IEEE Trans. on Pattern Analysis and Machine Intellig.5(2),179–190.

5. Baker, J. K. (1975). The DRAGON system – An overview. IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-23(1), 24–29. .

6. Nadas, A. (1984). Estimation of probabilities in the language ´ model of the IBM speech recognition system. IEEE Transactions on Acoustics, Speech, Signal Processing, 32(4), 859–861.

7. Newell, A., Langer, S., and Hickey, M. (1998). The role of ˆ natural language processing in alternative and augmentative communication. Natural Language Engineering, 4(1), 1–16.

8. https://lagunita.stanford.edu/c4x/Engineering/CS-224N/asset/slp4.pdf

9. https://research.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html

STATISTICAL LANGUAGE MODEL

Meladze Aleksandre, Kamkamidze Konstantin

Georgian Technical University

Summary

The topics discussed how to create, monitoring, process and describe statistical language model using with N-Gram. Discussed the key characteristics of their area of use, model problems and their solutions. There are examples of how to create such a model for Georgian language, the algorithm provides all the necessary theoretical, practical and as well as a software solutions. There are algorithms which can use to create Georgian statistical language model, process this model and use it in real examples. There are programming codes which helps to create real software solutions with N-Gram model.

СТАТИСТИЧЕСКИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Меладзе А., Камкамидзе К.

Грузинский Технический Университет

Резюме

Темы, обсуждаемые в статистической модели языка для создания, обработки и описания N-грамм, касаются ключевых характеристик и их областей применения, проблем и путей их решения. Есть примеры того, как создать такую модель, пример грузинского языка, алгоритм котояово предоставляет всю необходимую теоретическую часть а также программное решение. Использование статистических алгоритмов обсуждаются на грузинском языке, языковой модели, обработки и ее использование на практике приведени отрывки из некоторых программ, с помощью которых можно построить модель N-грамм.