

საქართველოს ტექნიკური უნივერსიტეტი

ლაშა გავაშელაშვილი

საძიებო სისტემაში ინფორმაციის ოპტიმალურად ძებნის
მეთოდის შემუშავება

წარმოდგენილია დოქტორის აკადემიური ხარისხის მოსაპოვებლად

სადოქტორო პროგრამა „ინფორმატიკა“ შიფრი 0401

საქართველოს ტექნიკური უნივერსიტეტი

თბილისი, 0175, საქართველო

ივლისი, 2015 წელი

საავტორო უფლება © 2015 წელი, ლაშა გავაშელაშვილი

თბილისი

2015 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკური უნივერსიტეტში
ინფორმატიკისა და მართვის სისტემების ფაკულტეტი
კომპიუტერული ინჟინერიის დეპარტამენტი

ხელმძღვანელები: პროფ. გურამ ჩაჩანიძე
პროფ. რომან სამხარაძე

რეგენზენტები: -----

დაცვა შედგება ----- წლის "-----" -----, ----- საათზე
საქართველოს ტექნიკური უნივერსიტეტის -----
----- ფაკულტეტის სადისერტაციო საბჭოს
კოლეგიის
სხდომაზე, კორპუსი -----, აუდიტორია -----
მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ს ბიბლიოთეკაში,
ხოლო ავტორეფერატისა - ფაკულტეტის ვებგვერდზე

სადისერტაციო საბჭოს მდივანი პროფ. თინათინ კაიშაური

საქართველოს ტექნიკური უნივერსიტეტი

ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ჩვენ ქვემოთ ხელისმომწერი ვადასტურებთ, რომ გავეცანით "ლაშა გავაშელაშვილის" მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: „სამიეზო სისტემაში ინფორმაციის ოპტიმალურად ძებნის მეთოდის შემუშავება“ და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის "ინფორმატიკისა და მართვის სისტემების ფაკულტეტის" სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

ხელმძღვანელები: პროფ. გურამ ჩაჩანიძე

პროფ. რომან სამხარაძე

რეცენზენტი:

რეცენზენტი:

საქართველოს ტექნიკური უნივერსიტეტი

2015

ავტორი: ლაშა გავაშელაშვილი
დასახელება: „სადიებო სისტემაში ინფორმაციის ოპტიმალურად
მეზნის მეთოდის შემუშავება“
ფაკულტეტი: "ინფორმატიკისა და მართვის სისტემების ფაკულტეტი"
აკადემიური ხარისხი: დოქტორი
სხდომა ჩატარდა:

ინდივიდუალური პიროვნების ან ინსტიტუტების მიერ შემოყვანილი დასახელების დისერტაციის გაცნობის მიზნით მოთხოვნის შემთხვევაში მისი არაკომერციული მიზნების კოპირებისა და გავრცელების უფლება მინიჭებული აქვს საქართველოს ტექნიკურ უნივერსიტეტს.

ავტორის ხელმოწერა

ავტორი ინარჩუნებს დანარჩენ საგამომცემლო უფლებებს და არც მთლიანი ნაშრომის და არც მისი ცალკეული კომპონენტების გადაბეჭდვა ან სხვა რაიმე მეთოდით რეპროდუქცია დაუშვებელია ავტორის წერილობითი ნებართვის გარეშე.

ავტორი ირწმუნება, რომ ნაშრომში გამოყენებული საავტორო უფლებებით დაცულ მასალებზე მიღებულია შესაბამისი ნებართვა (გარდა იმ მცირე ზომის ციტატებისა, რომლებიც მოითხოვენ მხოლოდ სპეციფიურ მიმართებას ლიტერატურის ციტირებაში, როგორც ეს მიღებულია სამეცნიერო ნაშრომების შესრულებისას) და ყველა მათგანზე იღებს პასუხისმგებლობას.

დისერტაციას ვუძღვნი

ნაშრომი ეძღვნება ჩემს საუკეთესო მასწავლებელსა და მეგობარს, ქალბატონ მანანა ქარქაშაძეს. მისი მუხლჩაუხრელი თანადგომისათვის, მეგობრობისა და სიყვარულისათვის.

რეზიუმე

სადისერტაციო ნაშრომის „ინტერნეტ ძიების ოპტიმალური მეთოდების შემუშავება“ მიზანს წარმოადგენს, რომ ინტერნეტ ძიების პროცესში დროის მოკლე მონაკვეთში შესაძლებელი იყოს მაქსიმალურად რელევანტური რესურსების მოძიება და ძიების ახალი მეთოდის შემუშავება.

მოცემული ნაშრომის მეცნიერულ სიახლეს წარმოადგენს, ინტერნეტ ძიების რთულ პროცესში, მაქსიმალურად სასარგებლო ინფორმაციის მოწოდება. შემუშავებულია მეთოდი, რომელიც ახდენს კონკრეტული ინტერნეტ-მომხმარებლის მიერ გამოყენებული ვებ-გვერდის სიხშირის დაანგარიშებას. ალგორითმი ითვლის კონკრეტული ძიებისას პიროვნების მიერ გამოყენებული ვებ-რესურსების სიხშირეს და ქმნის მის შესაბამის ინდექსს. აღნიშნული მეთოდი იძლევა ძიების უფრო მაღალ ხარისხს, მკვეთრად უმჯობესდება ძიების დრო და ძიების პროცესი. პირველ თავში, ჩატარებულია ინტერნეტ ძიების პროცესების მიმოხილვა და ანალიზი. განზოგადებულია ინფორმაციულ საზოგადოებაზე გადასვლის აუცილებლობა. ნაჩვენებია ინტერნეტ-ძიების მუშაობის პრინციპი. დასაბუთებულია, რომ დღეისათვის არსებული მეთოდები აღარ არის ეფექტური, ძიების პროცესში გამოვლენილი პრობლემების გადასაწყვეტად. დასაბუთებულია ინტერნეტ-ძიების ახალი მეთოდების გამოყენების აუცილებლობა. ჩამოყალიბებულია მათი გამოყენების აუცილებლობა.

განხილულია კლასიკური ძიების არსებული მეთოდი, კერძოდ Seo-ს ოპტიმიზაციის მაგალითზე, გამოვლენილია ნაკლოვანებები და დადებითი მხარეები. ჩატარებულია წესებზე დაფუძნებული ძიების პრინციპების მიმოხილვა და კრიტიკული ანალიზი. დასაბუთებულია, რომ არსებული მეთოდები არ პასუხობენ ძიების მაღალ ხარისხს და შესაბამისად მეტად მნიშვნელოვანია ამ პრობლემის გადასაწყვეტი გზების ძიება და შესაბამისი მოდელებისა და ალგორითმების შემუშავება.

მეორე თავში, აგებულია ინტერნეტ ძიების ეფექტური მოდელი. განხილულია მომხასურების ხარისხის მახასიათებლების კონტროლის მეთოდები. დადგენილია, რომ აღნიშნული მეთოდები ოპტიმალური ძიების პროცესში არიან შეზღუდულნი. გამოვლენილია დღევანდელი გლობალური ქსელების პრობლემები და შემუშავებულია ამ პრობლემების გადაწყვეტის მეთოდები. გაანალიზებულია თითოეული მათგანი.

მესამე თავში, შემუშავებულია ალგორითმის კონცეფცია, რომელსაც საფუძვლად უდევს ძირითადი მოთხოვნა: ალგორითმებს უნდა შეეძლოთ ძიების ხარისხის ამაღლება და ხასიათდებოდეს ძიების მაღალეფექტურობით! შემუშავებული ალგორითმი იყენებს ძიების ინტელექტუალურ მეთოდს, რომელიც ახდენს საკუთარი თავის „სწავლებას“ და ეს დინამური პროცესია. მას შეუძლია კონკრეტული მომხმარებლისათვის ძიების შედეგის შეთავაზება. ის აუმჯობესებს უკვე არსებულ ძიების საშუალებებს, ვებ-რესურსის გამოყენების სიხშირის კონტროლით. განხილულია შემუშავებული ალგორითმის ბრაუზერებსა და ოპერაციულ სისტემებზე ინტეგრირება და მისი მუშაობის პრინციპის ანალიზი. დასაბუთებულია ძიების ალგორითმის შესაძლებლობები და მათი გამოყენების ასპექტები ძიების ფაზაში.

გაანალიზებულია ძიების ხარისხის მახასიათებლების კონტროლის ალგორითმების მუშაობის შედეგები.

დასკვნების ნაწილში, შეჯამებულია სადისეტაციო ნაშრომის ძირითადი მიზნები და მიღებული შედეგები; თემის აქტუალურობა და შემუშავებული მეთოდების მეცნიერული სიახლე.

Abstract

The aim of the dissertation “Working out optimal methods of Internet search” is to make it possible to find the most relevant resources in the Internet in a very short period of time and to work out a new method of search.

The scientific novelty of the dissertation is to provide highly useful information in the difficult process of Internet search. We have worked out a method which counts the frequency of usage of a web-page by a specific Internet user. The algorithm counts the frequency of usage of web-resources in specific search and creates its relevant index. The given method gives a higher quality search, which improves the search time and process.

The first chapter overviews and analyses the process of Internet search. It generalises the necessity of moving to informational society; shows the working process of Internet search; proves that the existing methods are no longer effective to solve the problems revealed in the process of Internet search; justifies the necessity of using new methods of Internet search.

We have discussed the existing method of classic search, namely, on the example of Seo optimisation, and revealed the negative and positive sides. We have carried out an overview and critical analysis of rule-based search principles. It is proved that the existing methods don't correspond to the high quality of search and, therefore, it is utterly important to search for new ways of solving this problem and to work out relevant models and algorithms.

The second chapter presents an effective model of Internet search. It discusses the methods of controlling the characteristic features of service quality. It is stated that the given methods are limited in the process of optimal search. The chapter reveals the problems of present global nets and works out the methods of solving these problems. Each of them is analysed.

The third chapter works out the conception which is based on the main demand: algorithms must be able to improve the quality of search and must be characterised by high efficiency of search!

The studied algorithm uses intellectual method of search, which “studies” itself, and this is a dynamic process. It can offer search results to individual users. It improves the means of existing search by controlling the usage frequency of web-resources. The chapter discusses the integration of algorithm in browsers and operational systems and the analysis of its working principles. It justifies the possibilities of algorithms and the aspects of their usage in the search phase.

The results of the search quality of controlling characteristics of algorithm functioning are analysed.

The conclusions summarise the main aims and the acquired results of the dissertation, present the actuality and scientific novelty of the paper.

შინაარსი

შესავალი	16
ლიტერატურული მიმოხილვა	
თავი I. ინტერნეტ ძიების პრინციპები და საშუალებები	21
1.1 ინტერნეტის შექმნის ასპექტები.....	21
1.2. საძიებო სისტემები და მისი განმარტება.....	23
1.2.1 საძიებო სისტემის სახეები და ძიების ფორმები	25
1.3 საძიებო სისტემების ოპტიმიზაცია(SEO)	26
1.3.1 SEO-ს ოპტიმიზაციის მეთოდები	27
თეთრი ოპტიმიზაცია	28
ნაცრისფერი ოპტიმიზაცია.....	28
შავი ოპტიმიზაცია.....	28
1.3.2 Web Crawler სისტემა და ყველაზე გავრცელებული ტიპები	29
1.3.3 საძიებო სისტემის მუშაობის ასპექტები.....	34
1.3.4 არალეგალური მეთოდები: spamindexing.....	35
1.4. ძიების სტრუქტურა Google-ს მაგალითზე	36
პირველი თავის დასკვნები.....	39
შედეგები და მათი განსჯა	
თავი II ინტერნეტ ძიების მოდელი.....	40
2.1. ინტერნეტ ძიების არქიტექტურა.....	40
2.2 შეზღუდვები	41
2.3 ძიების ალგორითმები	43
2.3.1 Breath-First.....	43
2.3.2 Best-First.....	45

2.3.3 Hints.....	45
2.3.4 Shark Search	47
2.3.5 Info Spider	48
2.3.6 Google Panda.....	49
2.3.7 Google Pinguin.....	50
2.3.8 Pagerank	50
2.3.9 Hummingbird.....	53
2.3.10 Wordnet.....	53
2.4. რანჟირების ალგორითმების კვლევა	55
2.5. კლასიფიკაცია	64
2.6. ძიების სტრუქტურული მიმოხილვა	71
II თავის დასკვნა.....	74
ექსპერიმენტული ნაწილი	
თავი III. ინტერნეტ ძიების ოპტიმალური მეთოდები	75
3.1 ინტერნეტ ძიების ხერხები.....	75
3.2 პრობლემის გადაწყვეტის გზები	77
3.3 ალგორითმის მუშაობის არსი	80
3.4 დანერგვა/ინტეგრაცია	84
3.5 კონცეფციის პრაქტიკული რეალიზაცია.....	84
3.6 დეტალური აღწერა	85
III თავის დასკვნა.....	88
დისერტაციის საბოლოო დასკვნები და რეკომენდაციები.....	89
ლიტერატურა.....	92
დანართი I	97
დანართი II.....	10

დანართი III..... 101

ნახაზების ნუსხა

ნახ. 1.....	31
ნახ. 2.....	35
ნახ. 3.....	36
ნახ. 4.....	37
ნახ. 5.....	43
ნახ. 6.....	46
ნახ.7.....	47
ნახ. 8.....	48
ნახ. 9.....	49
ნახ. 10.....	51
სურ. 11.....	54
ნახ. 12.....	55
ნახ. 13.....	58
ნახ. 14.....	60
ნახ. 15.....	63
ნახ. 16.....	69
ნახ. 17.....	70
ნახ. 18.....	72
ნახ. 19.....	73
ნახ. 20.....	74
ნახ. 21.....	75
ნახ. 22.....	82
ნახ. 23.....	85
ნახ. 24.....	87
ნახ. 25.....	92

გამოყენებული აბრევიატურების ნუსხა

1. Seo-Search engine optimization.
2. HTML-HyperText Markup Language.
3. IPTO- Information Processing Technology Office.
4. BBN- Beranek and Newman.
5. NFS- National Science Foundation.
6. ARPA- Advanced Research Projects Agency.
7. SAGE-Semi Automatic Ground Environment.
8. DARPA- Defense Advanced Research Projects Agency.
9. URL- uniform resource locator.
10. IPSS- International Packet Stream Service.
11. MIT- Massachusetts Institute of Technology.
12. FTP-file transfer protocol.
13. DOM- document object model.
14. APLs- application programming interface.

მადლიერება

მადლობას ვუხდით ჩემს ხელმძღვანელებს, პროფესორებს: გურამ ჩაჩანიძეს და რომან სამხარაძეს სამეცნიერო მუშაობის პერიოდში გაწეული კონსულტაციებისა და ფასდაუდებელი დახმარებისათვის.

შესავალი

საინფორმაციო საზოგადოება, ესაა სოციალური სისტემა, რომელშიც საქონლისა და მოხმარების წარმოება არსებითად დამოკიდებულია ინფორმაციის შეგროვებაზე, დამუშავებასა და გადაცემაზე. თუ ინდუსტრიულმა საზოგადოებამ შესაძლებელი გახადა უდიდესი რაოდენობის სხვადასხვა საქონლის წარმოება, ინფორმაციული ტექნოლოგიების ერის ჩამოყალიბებამ უზრუნველყო ინფორმაციის უზარმაზარი რაოდენობის წარმოების, დამუშავებისა და გადაცემის შესაძლებლობა. თუ ინდუსტრიული საზოგადოების ძირითადი რესურსი კაპიტალი იყო, პოსტინდუსტრიული საზოგადოების სტრატეგიული რესურსებია ინფორმაცია და თეორიული ცოდნა. მეცნიერება კი გადაიქცა ძირითად საწარმოო ძალად. ახალი ტექნიკის, ახალი ტექნოლოგიებისა და კომუნიკაციის შესაძლებლობანი გრანდიოზული გახადა.

საინფორმაციო საზოგადოება ზოგადად სერვისული ტექნოლოგიების საზოგადოების ერთ-ერთ ვარიანტს წარმოადგენს. ამ საზოგადოებაზე თანდათანობით გადასვლა თანამედროვე ისტორიული ეტაპის ერთ-ერთი თავისებურებაა, რაშიც დიდი როლს მეცნიერული და საინფორმაციო უზრუნველყოფა ასრულებს. მომსახურებათა მრავალფეროვნება მომსახურებათა ფართო დიაპაზონს გულისხმობს, დაწყებული უმარტივესიდან ინტერნეტით შემოთავაზებული მომსახურებით დამთავრებული.

უკანასკნელ რამდენიმე წელიწადში შემჩნეული საინფორმაციო და საკომუნიკაციო ტექნოლოგიების განვითარება და გამოყენების მკვეთრი დაჩქარება “ინდუსტრიულიდან” “ინფორმაციული” საზოგადოებისკენ გადასვლის მსოფლიო პროცესის დასაწყისი გახდა.

ინფორმაციული საზოგადოების ცხოვრების განმსაზღვრელი ფაქტორი, მთლიანობაში, თეორიული ცოდნაა. სოციალური

დიფერენციაციის განმსაზღვრელი ფაქტორი ხდება ცოდნის დონე და არა საკუთრება. “მდიდრებად” და “უპოვრებად” დაყოფა პრინციპულად ახალ ხასიათს იძენს: პრივილეგირებულ ფენას ქმნიან ინფორმირებულნი, ხოლო არაინფორმირებულნი “ახალი დატაკებია”. ანუ უპირველესია ინფორმაციული უზრუნველყოფა.

ინფორმაციული საზოგადოების ინფრასტრუქტურას ახალი “ინტელექტუალური” ტექნიკა წარმოადგენს და არა “მექანიკური”. სოციალური ორგანიზაცია და საინფორმაციო ტექნოლოგიები სიმბიოზს ქმნიან. საზოგადოება ისეთ ეპოქაში შედის, როდესაც სოციალური პროცესები დაპროგრამებადი ხდება. თანამედროვე ეპოქაში ინფორმაცია დაკავშირებულია არა ცოდნასთან, არამედ კომუნიკაციებთან. უფრო ინფორმირებული ის ადამიანია რომელსაც ბევრი კომუნიკაციის საშუალება გააჩნია და არა ის ვინც მეტი იცის.

ინფორმაციის კომუნიკაციური ასპექტის უდიდესი მნიშვნელობა მის ოპერაციონალურობაშია, ე. ი. მოქმედებათა გამართლება მის დასაბუთებაში მდგომარეობს. სწორედ ამით აიხსნება ინფორმაციის საჭიროება, თანამედროვე ადამიანისათვის, სწორედ ამიტომ თამაშობს ინფორმაცია უდიდეს როლს თანამედროვე საზოგადოებაში.

ინფორმაციული საზოგადოების ძირითადი მახასიათებლების შემდეგი კრიტერიუმები:

- ტექნოლოგიური- მთავარი ფაქტორია საინფორმაციო ტექნოლოგიები, რომლების ფართოდ გამოიყენება, წარმოებაში, დაწესებულებებში, განათლებაში და ა.შ.
- სოციალურ ინფორმაცია, როგორც ცხოვრების ხარისხის შეცვლის სტიმულატორი, ვითარდება და მტკიცდება ინფორმაციული შეგნება.
- ეკონომიური ინფორმაცია, არის ძირითადი ფაქტორი ეკონომიკაში, როგორც რესურსი. მომსახურება, საქონელი, დამატებითი შემოსავლის წყარო.

- პოლიტიკური ინფორმაციის თავისუფლება, რომელსაც მივყავართ პოლიტიკური პროცესისაკენ და რომელიც ხასიათდება მზარდი მონაწილეობით და კონსესუსით სხვადასხვა კლასებსა და სოციალურ ფენებს შორის.
- კულტურული ინფორმაციის კულტურული ღირებულების აღიარება.

ამ ცვლილებებიდან გამომდინარე, ინფორმაციული საზოგადოება შეიძლება განისაზღვროს, როგორც საზოგადოება, რომელშიც ცხოვრების ხარისხი ისევე როგორც სოციალური ცვლილებების და ეკონომიური განვითარების პერსპექტივები დამოკიდებულია ინფორმაციაზე და მის ექსპლუატაციაზე.

საინფორმაციო საზოგადოება არა მხოლოდ დღევანდელი მოწინავე ქვეყნების, არამედ მთლიანად კაცობრიობის მომავალია და ამჟამად მსოფლიო განვითარების უმთავრესი ტენდენცია სწორედ გლობალური საინფორმაციო საზოგადოების ჩამოყალიბებაა. გლობალური საინფორმაციო საზოგადოება თანამედროვე ცივილიზაციის განვითარების ახალი ეტაპია, რომელსაც ახასიათებს:

1. ინფორმაციისა და ცოდნის განმსაზღვრელი მნიშვნელობა ცხოვრების ყველა სფეროსათვის, ინფორმაციისა და ცოდნის ბაზრის შექმნა და მისი პრიორიტეტული განვითარება.
2. გლობალური საინფორმაციო სივრცის შექმნა.
3. ახალ ტექნოლოგიურ წყობათა შექმნა და მათი შემდგომი დომინირება.
4. შრომის ბაზარზე მოთხოვნილების წამყვან მახასიათებლებად მაღალი კვალიფიკაციის, პროფესიონალიზმისა და შემოქმედებითი ნიჭის გადაქცევა.
5. ადამიანისა და მოქალაქის პოლიტიკურ, სოციალურეკონომიკურ უფლებათა უზრუნველყოფის, მოსახლეობისა და ხელისუფლების ურთიერთქმედებების ეფექტიანი სისტემების მზარდი განვითარება.

როგორც ზემოთ აღინიშნა, ცოდნა წარმოადგენს ძირითად მამოძრავებელ ძალას. ცხადია დღეს, როდესაც ინტერნეტის მოცულობა ძალიან დიდია, მნიშვნელოვანია მისი გამოყენების აუცილებლობა ცოდნის დაგროვების პროცესში. შესაბამისად აუცილებელია ძიების საშუალებების არსებობა, რომ მივიღოთ სასურველი ინფორმაცია, უდიდესი მოცულობის ინფორმაციული საცავიდან[36].

გლობალური ქსელის განვითარების შესაბამისად და მასში არსებული დოკუმენტების რაოდენობის ზრდის გამო წარმოიშვება ინტერნეტ ძიების პრობლემა. დღევანდელი მდგომარეობით ეს პრობლემა წყდება ინტერნეტ სამიებო სისტემების დახმარებით, რომლებიც განსაზღვრავენ იმ დოკუმენტების(ვებ-გვერდების) სიას, რომლებიც შეესაბამებიან მომხმარებლების მოთხოვნებს. ასეთი დოკუმენტები შეიძლება იყოს მილიონობით და სამიებო სისტემის ამოცანას წარმოადგენს მიაწოდოს მომხმარებელს მოპოვებული გვერდების სია, სადაც ისინი დალაგებულნი იქნებიან რელევანტურების კლების მიხედვით.

რელევანტურობა განსაზღვრავს როგორც დოკუმენტის ძირეულ თვისებებს ასევე მომხმარებლის მოთხოვნას, რასაც ეწოდება რანჟირების ფაქტორები. მათმა რიცხვმა შეიძლება მიაღწიოს რამდენიმე ასეულს. თითოეული ფაქტორი შედის რანჟირების ფორმულაში, რომელიც განსაზღვრავს წარსადგენი ვებ-გვერდების სიას მომხმარებლის მოთხოვნის შესაბამისად.

მაგალითად, ეს ფაქტორები შეიძლება იყოს, მოთხოვნის ერთეული სიტყვები, რომლებიც გვხვდება დოკუმენტის სათაურში ან ტექსტში, მომხმარებლის გეოგრაფიული მდებარეობა(მაგალითად მომხმარებლის მოთხოვნაზე პასუხად შეიძლება მივიღოთ განსხვავებული სიები თუ ძიება ხდება ვთქვათ გერმანიიდან და სხვა სიები-საქართველოდან.) და სხვა.

ინფორმაციის ზრდის კვალდაკვალ ხდება სამიებო სისტემების ალგორითმების უწყვეტი განვითარება. უკანასკნელი წლის განმავლობაში შეიმჩნევა მრავალი სიახლე, მაგალითად, სამიებო სისტემებმა ისწავლეს

განსაზღვრონ დოკუმენტის პირველწყარო(რომ გამოირიცხოს დუბლიკატების არსებობა), რეალიზებულია ძიება მომხმარებლის რეგიონის შესაბამისად და ა.შ. ყოველივე ეს ფაქტორები ამაღლებენ საძიებო სისტემის ხარისხს და საიმედოობას მათ შორის სანდოობას მათი საგანმანათლებლო დანიშნულების მხრივ.

ზემოთ თქმულიდან გამომდინარე, მეტად აქტუალურია აღნიშნული პრობლემის გადაწყვეტის გზებისა და საშუალებების შემუშავება.

ნაშრომის მეცნიერული სიახლე მდგომარეობს საძიებო სისტემის ოპტიმიზების მიზნით ძებნის ახალი ალგორითმის შემუშავებაში. შემოთავაზებული ალგორითმი ინტეგრირდება მომხმარებლის ინტერნეტ ბრაუზერში, შეარულებს მის მიერ გამოყენებული ვებ-რესურსების გამოყენების სიხშირის გამოთვლას და ყოველივე ამის საფუძველზე მოახდენს ძებნის პროცესის ოპტიმიზებას ისე, რომ კონკრეტულ შემთხვევაში მომხმარებელს წარედგინება მის მიერ გამოყენებული ვებ-გვერდების რეიტინგული სია. სიის სათავეში განლაგებული იქნება ის ვებ-გვერდები რომელიც მან ყველაზე მეტი სიხშირით გამოიყენა. ასეთი მოდგომა მკვეთრად ამაღლებს მოძებნილი ინფორმაციის სარგებლიანობას

ლიტერატურული მომოხილვა

თავი I

ინტერნეტ ძიების პრინციპები და საშუალებები

ამ თავში არის მიმოხილული ინტერნეტისა შექმნისა და ინტერნეტ ძიების ასპექტები, რომლებიც პირდაპირ კავშირში არიან ეფექტური ინტერნეტ ძიების ალგორითმის შექმნასთან, განხილულია მათი ნაკლოვანებები და უპირატესობები. ასევე მათი მუშაობის პრინციპები, რომლებიც პირდაპირ ზეგავლენას ახდენენ მოცემულ ნაშრომში დამუშავებულ მეთოდებზე. განხილულია SEO-ს დანიშნულება და მისი ოპტიმიზაციის მეთოდები, რომელიც წარმოადგენს საფუძველს მთელი ინტერნეტ საძიებო სისტემებისათვის.

1.1 ინტერნეტის შექმნის ასპექტები

ინფორმაციულ საზოგადოებაზე გადასვლამ, წარმოშვა სხვადასხვა სახის ინფორმაციის ძიების მოთხოვნილება. ცხადია ყველამ ყველაფერი არასოდეს არ იცის. სწორედ ამ მიზნით, შეიქმნა ინტერნეტი. საბჭოთა კავშირის მიერ გაშვებულმა თანამგზავრმა, 1958 წელს, ტექნოლოგიური ლიდერობის დასაბრუნებლად, ამერიკის შეერთებულ შტატებს შთააგონა შეექმნა მოწინავე პროექტების კვლევის სააგენტო(Advanced Research Projects Agency), ცნობილი, როგორც ARPA. ARPA-მ, ნახევრად ავტომატური მიწის გარემოს (Semi Automatic Ground Environment (SAGE)) პროგრამის წინსვლისათვის, შექმნა ინფორმაციის გადამუშავების ტექნოლოგიის ოფისი(Information Processing Technology Office), ცნობილი, როგორც IPTO, რომელიც მიერთებული იყო ქვეყნის სარადარო სისტემის ქსელში J.C.R.Licklider არჩეულ იქნა ITPO-ს მეთაურად. Licklider 1950 წელს, ჰარვარდის უნივერსიტეტში არსებული ფსიქო-აკუსტიკური ლაბორატორიიდან გადავიდა, მასაჩუსეტის ტექნოლოგიის

ინსტიტუტში (MIT), რის შემდეგაც ის დაინტერესდა ინფორმაციული ტექნოლოგიებით. 1957 წელს ის გახდა Bolt Beranek and Newman(BBN)-ის ვიცე-პრეზიდენტი. სადაც მან შეიძინა PDP-1 კომპიუტერის პირველი პროდუქცია და გაუძღვა, მრავალ მომხმარებელს შორის რესურსების გამოთვლის განაწილების(Time-Sharing) დემონსტრაციას.

IPTO-ში, Licklider -მა ქსელის განვითარების პროექტის მეთაურობა გადააბარა Lawrence Roberts-ს. ამის შემდეგ რობერტსი დაეყრდნო Paul Baran-ის ტექნოლოგიას, რომელმაც ამერიკის საჰაერო ძალებისთვის(U.S. Air Force) დაწერა ამომწურავი ინფორმაციის მქონე წიგნი, რომელსაც რეკომენდაციას უწევდა Packet Switching. ხანგრძლივი მუშაობის შემდეგ, Menlo Park-ში, კალიფორნიაში, 29 ოქტომბერს, 1969 წელს, შეერთდა პირველი კვანძი, რომელიც ცნობილია, როგორც ARPANET, კალიფორნიის უნივერსიტეტსა (University of California, Los Angeles(UCLA)) და სტენფორდის კვლევით ინსტიტუტს(SRI International) შორის. ARPANET გახდა დღევანდელი ინტერნეტის ერთ-ერთი "მშობელი". დემონსტრაციის მიხედვით, რომელსაც აწარმოებდა Packet Switching, British Post Office, Telenet, DATAPAC და TRANSPAC ითანამშრომლეს, რათა შეექმნათ პირველი საერთაშორისო Packet-Switching ქსელის სერვისი. გაერთიანებულ სამეფოში 1978 წელს, ეს პროექტი ცნობილი იყო, როგორც საერთაშორისო პაკეტის დინების სერვისი(International Packet Stream Service (IPSS)). X.25 კოლექციის ბაზირებული ქსელი გაიზარდა ევროპიდან და ამერიკიდან, კანადაში, ჰონგ-კონგში და ავსტრალიაში X.25 პაკეტის სტანდარტი შეიქმნა CCITT-ში 1976 წელს(ახლა მას ქვია საერთაშორისო ტელეკომუნიკაციების კავშირი(ITU-T)). X.25 იყო დამოუკიდებელი TCP/IP პროტოკოლი, რომელიც შეიქმნა 1974 წლის, დეკემბრის დაცვის მოწინავე პროექტების კვლევის სააგენტოს(Defense Advanced Research Projects Agency(DARPA)) მიერ წარმოებულ ARPANET, Packet Radio Net და Packet Satellite Net-ზე ექსპერიმენტის შედეგად. Vinton Cerf და Robert Kahn-მა 1973 წელს შექმნეს

TCP პროტოკოლის პირველი განსაზღვრება, რომელიც გამოქვეყნდა 1974 წლის მაისში TCP-ს პირველი სრული სპეციფიკაცია დაიწერა Vinton Cerf, Yogen Dalal და Carl Sunshin-ის მიერ, შემდეგ კი სტენფორდის უნივერსიტეტში. შემდეგი ცხრა წლის განმავლობაში, მუშაობა მიმდინარეობდა პროტოკოლების გაუმჯობესებასა და მათ გამოყენებაზე ოპერაციული სისტემების ქსელის რანგში.

პირველი TCP/IP ფართო მასშტაბის ქსელი შეიქმნა 1983 წლის 1 იანვარს, როდესაც ARPANET-ში არსებული ყველა ჰოსტი ძველი NCP პროტოკოლიდან შეერთდა ახალ TCP/IP პროტოკოლში.

David L. Mills-ის ინიციატივით, 1985 წელს, ამერიკის შეერთებული შტატების ნაციონალური მეცნიერების ორგანიზაციამ (National Science Foundation (NSF)) შექმნა 56 კილობიტი/წამში სიჩქარის მქონე ქსელი, რომელსაც იყენებდნენ კომპიუტერები სახელად "fuzzballs". მიმდინარე წელში NSF-მა დაასპონსორა მაღალ სიჩქარიანი 1.5მეგაბიტი/წამში სიჩქარის ქსელი, რომელიც შემდგომ გახდა NSFNet[7][37].

1.2 საძიებო სისტემები და მისი განმარტება

პირველი საძიებო სისტემა იყო «Wandex». დღეისათვის იგი აღარ არსებობს. იგი შექმნა მეთიუ გრეიმ (ინგლ. *Matthew Gray*), მასაჩუსეტსის ტექნოლოგიური უნივერსიტეტიდან, 1993 წელს. ამ წელსვე შეიქმნა საძიებო სისტემა «Aliweb», რომელიც დღემდე მუშაობს. თუმცა პირველი საძიებო სისტემა, რომელიც მთლიანად "კითხულობდა" ვებ გვერდზე განთავსებულ ტექსტს, იყო «WebCrawler», რომელიც 1994 წელს გაეშვა. მისი წინამორბედებისგან განსხვავებით, იგი საშუალებას იძლეოდა ნებისმიერი საკვანძო სიტყვებით ძებნისა, ნებისმიერ საიტზე. რაც დღეისათვის ერთგვარ სტანდარტადაა ქცეული თითქმის ყველა საძიებო სისტემისთვის. 1994 წელს გაეშვა საძიებო სისტემა Lycos-იც, რომელიც კარნეგი მელონის

უნივერსიტეტში შეიქმნა.

მალე სხვა საძიებო სისტემებიც გამოჩნდნენ და დაიწყეს პოპულარობისთვის ბრძოლა. ესენი იყვნენ Magellan, Excite, Infoseek, Inktomi, Northern Light, და AltaVista. ასევე საკმაოდ პოპულარული იყო Yahoo!-ც, თუმცა იმ დროისთვის მისი საძიებო ფუნქცია მხოლოდ ვებ დირექტორიით შემოიფარგლებოდა, მას არ შეეძლო სხვა საიტების ინდექსირება. 1998 წელს შეიქმნა გუგლი, რომელიც დღეისათვის ყველაზე პოპულარული საძიებო სისტემაა. იგი იყენებდა და იყენებს საიტების შეფასების ინოვაციურ ფუნქციას, რომელსაც PageRank ეწოდება. გუგლი იღებს ინფორმაციას თუ რამდენ სხვა საიტზეა განთავსებული მოცემული საიტის ბმული და ამას ითვალისწინებს საიტების შეფასებისას. გუგლი ასევე გამოირჩევა თავისი სადა და მარტივი დიზაინით, განსხვავებით მრავალი მისი კონკურენტისგან, რომელთა დიზაინიც პორტალის სტილშია გადაწყვეტილი.

2000 წელს Yahoo!-მ წარმოადგინა ძიების სერვისი, რომელიც Inktomi საძიებო სისტემის ბაზას იყენებდა. Yahoo!-მ შეიძინა Inktomi და Overture (რომელიც ფლობდა AlltheWeb-ს და AltaVista-ს) 2003 წელს. 2004 წლამდე Yahoo! გუგლის ბაზას იყენებდა, 2004 წელს კი მან თავისი საძიებო სისტემა შექმნა [14, 15]. დღეისათვის საძიებო სისტემები მრავლად არის წარმოდგენილი. მათ შორის ყველაზე ხშირად მოხმარებადია:

- ✓ Google
- ✓ Yahoo
- ✓ Yandex
- ✓ Bing

ამ კომპანიებს შორის დღემდე არსებობს დიდი კონკურენცია და თითოეული მათგანი ცდილობს დახვეწოს საკუთარი ძიების ალგორითმი

საძიებო სისტემა (ინგლ. Search engine) არის ვებსაიტი, რომლის დანიშნულებაცაა ინტერნეტში ინფორმაციის მოძიება. მოძიებული

ინფორმაცია, ტიპების მიხედვით შეიძლება დაიყოს ვებ-გვერდებთან, სურათებთან, ვიდეოებთან და ა.შ. არსებობს ისეთი საძიებო სისტემები, რომლებიც FTP სერვერზე ეძებენ ფაილებს. საძიებო სისტემა მუშაობს ადამიანის მიერ შექმნილი სპეციალური ალგორითმებით. საძიებო სისტემა, მომხმარებლის მიერ მითითებული ძებნის პირობებისა და ტერმინების მიხედვით ეძებს ვებ-გვერდს, რომელიც შეესაბამება ამ პირობებსა და ტერმინებს, ძიებისას ბრაუზერი ჩვეულებრივ უკავშირდება ერთ ან რამოდენიმე საძიებო სისტემას. საძიებო სისტემა მოთხოვნილ ინფორმაციას ეძებს მის მიერ შექმნილ მონაცემთა ბაზაში. მონაცემთა ბაზა კი, დროთა განმავლობაში იქნება, სხვადასხვა ტერმინების მიხედვით, სხვადასხვა ვებ-გვერდების გამოკვლევის შედეგად.

1.2.1 საძიებო სისტემების სახეები და ძიების ფორმები

არსებობს ორი ძირითადი სახის საძიებო სისტემა ინდექსი და დირექტორია.

ინდექსი-არის სია, რომელშიც მითითებულია ძირითადი სიტყვები და იმ ვებ-გვერდების მისამართები. სადაც ეს სიტყვები მოიხსენება. მონაცემთა ბაზის შექმნისათვის გამოიყენება პროგრამა რომელსაც ობობას (spider) უწოდებენ. ობობა უზრუნველყოფს მოძებნილი ვებ-გვერდის შენახვას. ინდექსი შეიძლება შევადაროთ ანბანურ საძიებელს.

დირექტორია ანუ ცნობარი- არის ადამიანი მკვლევარების მუშაობის შედეგი, რომლებიც გადადიან საიტიდან საიტზე და ახდენენ მათ ორგანიზაციას შინაარსის, კატეგორიების მიხედვით. დირექტორია შეიძლება შევადაროთ წიგნის შინაარსს.

ჰიბრიდული ძიება- ძიებისას შეიძლება გამოყენებული იქნას ერთდროულად ორივე ზემოთ აღწერილი მეთოდი, ვინაიდან ორივეს აქვს თავისი ნაკლოვანებები და უპერატესობები და ინფორმაციის მოძიებისას შეიძლება გამოვიყენოთ ინფორმაციის ძიების ჰიბრიდული საშუალებები.

მეტაძიება- სხვადასხვა საძიებო სისტემა, ძებნის მოთხოვნის საპასუხოდ თითქმის ყოველთვის სხვადასხვა შედეგების სიას აწარმოებს. საძიებო

სისტემები განსხვავდებიან ზომით, შინაარსითა და ძიების მეთოდოლოგიით. ამის გამო, მაძიებელი ერთიდაიგივე ინფორმაციას სხვადასხვა საძიებო სისტემაში წერს და შემდეგ ახდენს სასურველი საიტის ამორჩევას. არსებობს ისეთი საძიებო სისტემები, რომლებით თვითონ მოძებნიან სხვადასხვა საძიებო სისტემებს და თვითონ განათავსებენ მოთხოვნებს ამ საძიებო სისტემებში. ანუ ერთიდაიგივე მოთხოვნას დაამუშავებს რამოდენიმე საძიებო სისტემაში, შვიდი, რვა და ა.შ. შემდეგ კი მიღებულ შედეგს გააერთიანებს ერთ სიაში . ასეთ ძიებას მეტაძიება ეწოდება, ხოლო სისტემას მეტასაძიებო სისტემები[5].

1.3 საძიებო სისტემების ოპტიმიზაცია

SEO (search engine optimization)საძიებო სისტემების ოპტიმიზაცია, ონლაინ მარკეტინგის ერთგვარი ფორმაა, რომელიც საიტის შიგთავსს როგორც საძიებო სისტემების, ასევე მომხმარებლებისთვის მეტად ფასეულს ხდის. SEO ვებ-მარკეტინგის ერთ-ერთი საკვანძო საქმიანობაა, რომელსაც შეუძლია მიიზიდოს სხვადასხვა ტიპის საძიებო სისტემები, როგორცაა:

- სურათებით ძიება;
- ლოკალური ძიება;
- ვერტიკალური ძიება;

SEO-ს დროს გასათვალისწინებელია, თუ როგორ მუშაობს საძიებო სისტემა და როგორ ეძებენ მომხმარებლები.

ოპტიმიზაცია, პირველ რიგში მოიცავს ვებ-გვერდის კონტენტისა და HTML კოდის რედაქტირებას, რაც აუცილებელია გარკვეული საკვანძო სიტყვების რანგის გასაზრდელად და ყველა იმ ბარიერის გასაქრობად, რომელიც საძიებო სისტემებში ინდექსირებას უშლის ხელს. ეს არის პროცესი, როდესაც ვებ-გვერდისთვის გამოიყენება ისეთი სტრუქტურა, ტექნოლოგია და კონტენტი, რომ საძიებო მანქანები ხელს უწყობენ და ვებ-გვერდს აძლევენ მაღალ რეიტინგს გარკვეულ საკვანძო სიტყვებზე.

როგორც ინტერნეტ მარკეტინგის სტრატეგია, SEO განსაზღვრავს, თუ როგორ მუშაობს საძიებო ძრავი, რას ეძებს ხალხი. ფაქტობრივი საძიებო ტერმინები და საკვანძო სიტყვები იბეჭდება საძიებო ძრავებში და უპირატესობა აქვთ მინიჭებული თავიანთი მიზნობრივი აუდიენციით.

ვებ-გვერდის ოპტიმიზაცია, შეიძლება გულისხმობდეს გვერდის კორექტირებას, HTML – სა და ასოცირებული კოდების გაზრდას სპეციფიურ საკვანძო სიტყვებთან შესაბამისობაში და აუცილებლად ყველა ბარიერის წაშლას, რაც დაკავშირებულია საძიებო სისტემის საინდექსო საქმიანობასთან. SEO – ს ("search engine optimizers") აბრევიატურის მრავლობითი რიცხვი, აღნიშნავს ადამიანებს ან მათ ჯგუფს, ე.წ. „საძიებო სისტემის ოპტიმიზატორებს“, რომლებიც გვთავაზობენ SEO – ს მომსახურებებს[31][32].

1.3.1 Seo-ს ოპტიმიზაციის მეთოდები

ოპტიმიზაციის მეთოდების დაყოფა სამ კლასად შეიძლება, თუმცა მსოფლიოს საძიებო სისტემების უკანასკნელმა მოქმედებებმა მოგვცა იმის გაგების საშუალება, რომ ეს დაყოფა ძალზედ პირობითია. ნებისმიერი მანიპულაციები საიტის განსაზღვრული პარამეტრებით, შეიძლება საძიებო სისტემების მიერ აღქმული იქნეს, როგორც არასასურველი ზემოქმედება მის შედეგზე. ნებისმიერი მანიპულირება, საძიებო შედეგებზე პირდაპირ აკრძალულია ლიცენზიით. Yandex სისტემის გამოყენებით. „თეთრი“ ოპტიმიზატორები და მარკეტოლოგები იყენებენ Yandex – ის რეკომენდაციებს „კარგი“ საიტების შესაქმნელად[32].

თეთრი ოპტიმიზაცია

თეთრი ოპტიმიზაცია – ოპტიმიზირებული სამუშაო რესურსებთან, რომლებიც ოფიციალურად აკრძალულია ყოველი საძიებო სისტემის მიერ – საიტის ალგორითმებზე ზეგავლენის მოხდენის გარეშე. ეს მოიცავს საიტთან მუშაობას, მის შიგა ნავიგაციასთან და შინაარსთან, ასევე გარე სამუშაოებს, საიტის ოპტიმიზირება გამოკითხვების მეშვეობით, პრეს – რელიზების, სოციალურ ქსელებში რეგისტრირების, პარტნიორულ პროგრამებსა და ა.შ.

ნაცრისფერი ოპტიმიზაცია

ნაცრისფერი საძიებო სისტემის ოპტიმიზაცია, მოიცავს საკვანძო სიტყვების დიდი რაოდენობით ჩასმას საიტის ტექსტში, ხშირად ადამიანებისთვის ყველაზე გამოსაჩენ ადგილებში, მაგალითად: „რძე“, „რძის ნაწარმის საუკეთესო საჩუქარია, რადგანაც მას რძის ცხიმი გააჩნია“. ამასთან, ხშირად გამოიყენება HTML – ის სათაურის ტეგში, h1, საკვანძო სიტყვების ატრიბუტი.

ნაცრისფერი ოპტიმიზაცია, იმით განსხვავდება შავი ოპტიმიზაციისაგან, რომ ის ოფიციალურად არ არის აკრძალული, მაგრამ მისი გამოყენება შესაძლოა შეფასებულ იქნას, როგორც საიტის პოპულარობის არაზუნებრივ მცდელობად. ზოგიერთი საძიებო სისტემებმა, მაგალითად Google – მა შესაძლოა დროებით ან სამუდამოდ დაბლოკოს ასეთი საიტი. საბოლოო გადაწყვეტა იმაში მდგომარეობს, რომ საიტის რეიტინგის წამოწევა კანონიერია თუ არა, ამას არა პროგრამა, არამედ საძიებო სისტემის მოდერატორი წყვეტს[32].

შავი ოპტიმიზაცია

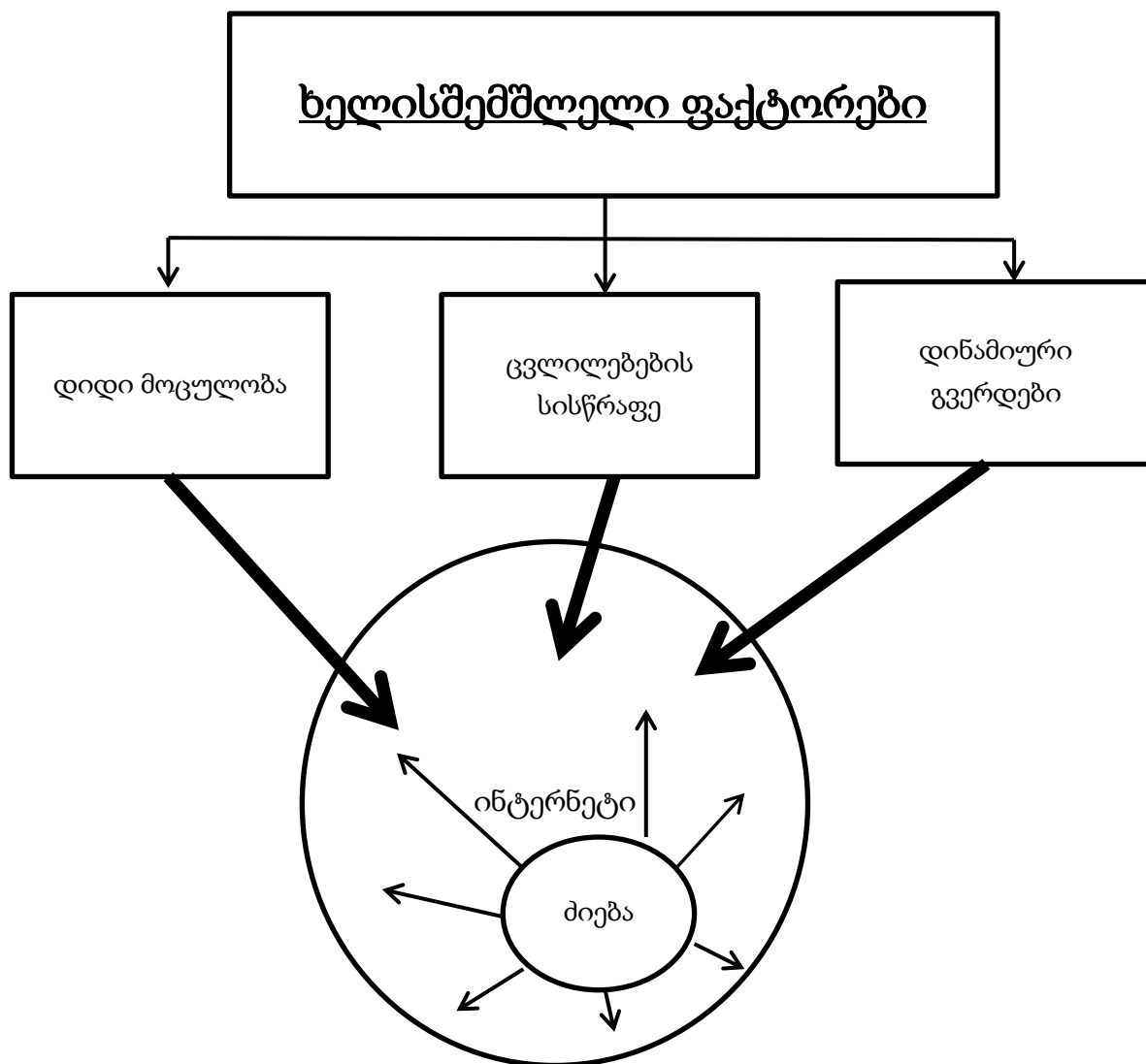
შავ ოპტიმიზაციაში შედის მეთოდები, რომლებიც საძიებო სისტემას ეწინააღმდეგებიან. მათ შორის შემდეგის გამოყოფა შეიძლება:

- “Doorway” - ს (გვერდები და რესურსები, სპეციალურად შექმნილი საძიებო სისტემების სამუშაოდ, ხშირად დიდი რაოდენობის საკვანძო სიტყვები საიტზე) გამოყენება.
- “Cloaking” (მომხმარებელს მიეცემა ერთი გვერდი, საძიებო სისტემას – მეორე, ოპტიმიზირებული გარკვეული მოთხოვნით) გამოყენება.
- საიტის დაფარული ტექსტის გამოყენება.
- „ერთ პიქსელიანი“ ლინკის გამოყენება[32].

1.3.2. Web Crawler სისტემა და ყველაზე გავრცელებული ტიპები

არსებობს ნორმალური ინტერნეტ ძიებისათვის რამდენიმე ხელისშემშლელი ფაქტორი(ნახაზი 1):

- დიდი მოცულობა.
- ძალიან სწრაფი ცვლილებები.
- დინამიური გვერდის გენერირება.



ნახაზი 1 ხელისშემშლელი ფაქტორები

Crawler-ს საიტზე სტუმრობისას განკუთვნილი აქვს მოცემული დრო და ამ მოცემულ დროში, მან უნდა მოასწროს მთელი ინფორმაციის მონაცემთა ბაზაში წაღება. თუ ჩვენი რესურსი ზედმეტად დიდი ზომისაა, ანუ ჩვენი ვებ-გვერდი შეიცავს ძალიან დიდ ინფორმაციას, მაშინ Crawler-ს შეიძლება პრიორიტეტები აეროს, ამიტომ საჭიროა ან მაქსიმალურად ცოტა რესურსის გამოყენება ან კიდევ პრიორიტეტების სწორად მითითება.

ძალიან სწრაფი ცვლილებები კი იმ ზიანს გვაყენებს, რომ არავინ იცის რამხელაა შუალედი Crawler-ის სტუმრობის გვერდზე, როდის შემოდის, რამდენი ხანი ჩერდება და ა.შ. სწრაფი ცვლილებებით კი შესაძლებელია ახლადგამწვეულ Crawler-ს მთავარი ინფორმაცია წაუღებელი დარჩეს.

Web Crawler-ისთვის ინფორმაციის სპეციალურად გენერირება

რომელიმე სერვერული პროგრამით შეიძლება ასევე წამგებიანი აღმოჩნდეს. იმიტომ, რომ შეიძლება მან ორი ერთნაირი ინფორმაცია წაიღოს. იგი ვერ გაარჩევს რომელია დუბლიკატი, რომელი ორიგინალი და რომელი რომლისგან განსხვავდება. ერთ-ერთი მთავარი გასათვალისწინებელი ასპექტი ისაა, რომ ვებ-გვერდი უნდა იყოს მაქსიმალურად მარტივი. მაგალითად ჩვენ გვაქვს სურათების პორტალი და მომხმარებელს ვთავაზობთ შემდეგ სერვისს. მათ შეუძლიათ გადმოწერონ სურათი JPG ან GIF ფორმატში, ასევე შეუძლიათ გადმოწერონ 3 სხვადასხვა ზომაში. ამ ყველაფერს გარკვეული ბმულების კომბინაცია დასჭირდება. ადვილი წარმოსადგენია როგორ “დაიბნევა” Crawler. თუ მომხმარებელს შევსთავაზებთ მხოლოდ ერთი ფორმატს და ერთი ზომას, მაშინ Crawler უფრო კარგად “იცოცებს” საიტზე[40].

Crawler -ის ძირითადი თვისებებია:

- **სელექციის პოლიტიკა** – Crawler-ისთვის პრიორიტეტულად შეიძლება ჩაითვალოს თემატურობა. მაგალითად თუ ვებ-რესურსი მთლიანად ეძღვნება მუსიკას, ხოლო, მეორე გვერდი ნაკლებ ყურადღებას აქცევს ამ თემას, მაგრამ მაინც აქვს ამის შესახებ ინფორმაცია, Crawler პირველს ჩათვლის პრიორიტეტულად.

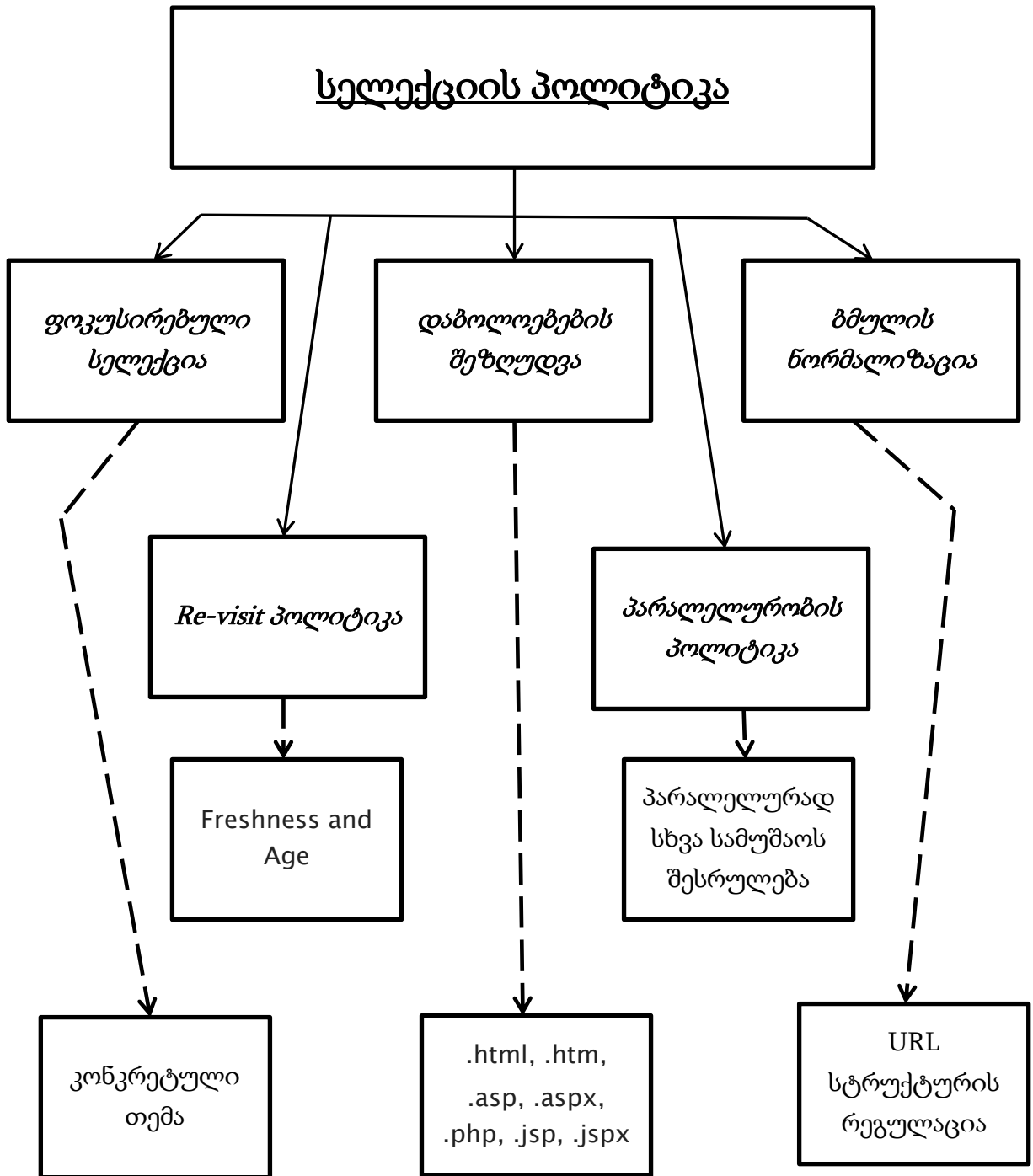
უფრო დაწვრილებით, რომ ვთქვათ სელექციის პოლიტიკას გააჩნია მნიშვნელოვანი ნაწილები:

- **ფოკუსირებული სელექცია** – როდესაც Crawler-ს ნაბრძანები აქვს კონკრეტული თემა და იგი მხოლოდ ამ თემასთან დაკავშირებით ეძებს რესურსს.
- **დაბოლოებების შეზღუდვა** – სელექციის ტიპი, როდესაც Crawler ეძებს მხოლოდ გარკვეული დაბოლოებების რესურსს. მაგალითად “მცოცავს” შეიძლება ნაბრძანები ჰქონდეს მხოლოდ: „.html“, „.htm“,

„.asp“, „.aspx“, „.php“, „.jsp“, „.jspx“ დაბოლოებული რესურსის მიება. ანუ ასეთი ძებნის შემთხვევაში Crawler არ მოძებნის არცერთ სურათს და მულტიმედიურ ფაილს.

- **ლინკის ნორმალიზაცია** - ასეთი სელექციის ტიპი არეგულირებს URL-ს სტრუქტურას, სანამ მას თავის მონაცემთა ბაზაში წაიღებს. მაგალითად: `http://www.example.com`. ამ შემთხვევაში Crawler წაიღებს: `http://www.example.com/`
ან `http://www.example.com:80/test.html` -ს.
ასევე: `http://www.example.com/test.html`
- **Re-Visit პოლიტიკა** - ვებ-გვერდებში იმდენად სწრაფია ცვლილებები, რომ Crawler-ს შეიძლება თვეები დაჭირდეს ყველა რესურსიდან სიახლის წასაღებად. შეიძლება ბევრი განახლება მერე მოხდეს, როდესაც Crawler დაასრულებს ძიებას. სწორედ ასეთი გაუგებრობების თავიდან ასაცილებლად დაიწერა სპეციალური ალგორითმი, რომელიც დღესაც ყველაზე ფართოდ გამოიყენება, სახელად: "Freshness and Age".
 - **Freshness** ატყობინებს "მცოცავს" რამდენად კარგადაა შესრულებული საქმე.
 - ხოლო **Age** ატყობინებს, რამდენად ძველია შენახული ინფორმაცია.
- **პარალელურობის პოლიტიკა** -Crawler-ებს შეუძლიათ პარალელურ რეჟიმში "მუშაობა". ასე, რომ შესაძლებელია რამდენიმე პროცესის პარალელურ რეჟიმში შესრულება.

ყოველივე ზემოთთქმული, ნათლად არის წარმოდგენილი ნახაზი 2-ზე

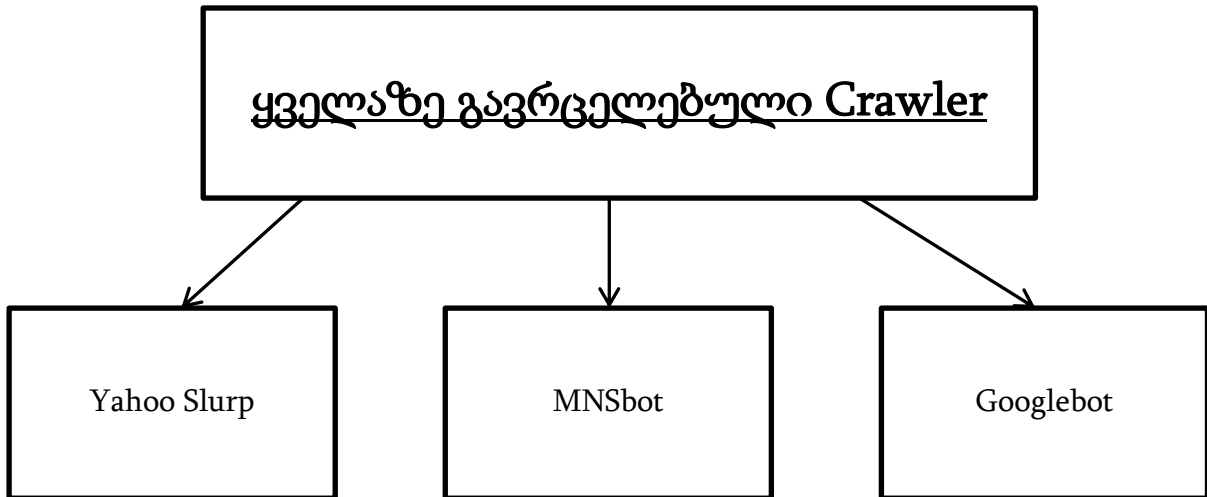


ნახაზი 2 სელექციის პოლიტიკა

ყველაზე გავრცელებულ Crawler-ები წარმოდგენილია ნახაზი 3-ზე, ესენია:

- **Yahoo Slurp** – yahoo search-ის Crawler

- **MSNBOT** – Crawler ბოტი, რომელიც შექმნა Microsoft-მა სპეციალურად Bing-ისთვის. მას აქვს ზუსტად იგივე ფუნქციები, რაც GoogleBot-ს და Yahoo Slurp-ს. 2009 წლის ნოემბერეს შექმნეს MSNBOT-ის ახალი ვერსია, ანუ MSNBOT 2.0
- **GoogleBot** ყველაზე ცნობილი Crawler ბოტი, რომელსაც გუგლი ხმარობს[40].



ნახაზი 3 Crawler-ები

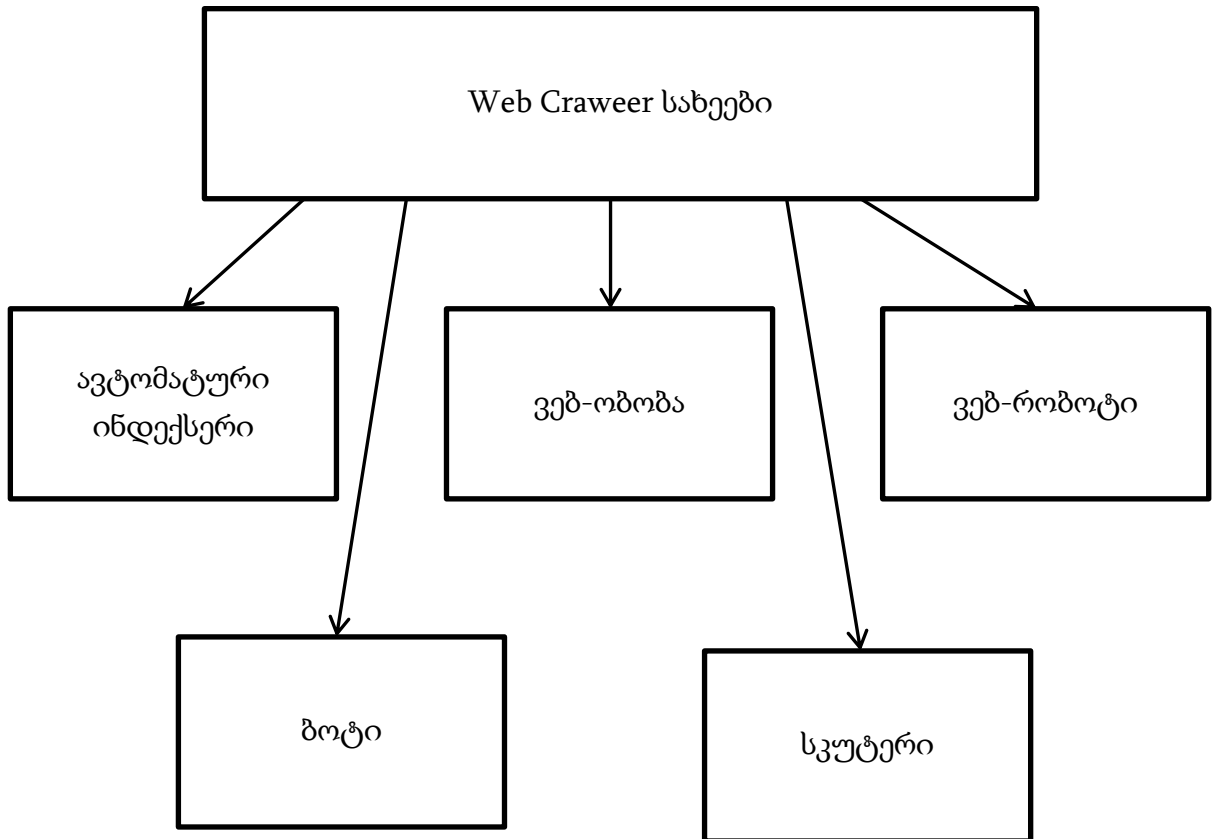
1.3.3 საძიებო სისტემის მუშაობის ასპექტები

Web Crawler შეიძლება მრავალი სახეობის იყოს(იხილეთ ნახაზი 1.4):

- ავტომატური ინდექსირება
- ვებ-ობობები
- ვებ-რობოტები
- ბოტები
- სკუტერები და ა.შ.

“მცოცავები” კოპირებას უკეთებენ ყველა მის მიერ მოძებნილ ვებ-გვერდს, რადგან იმ გვერდზე შეიძლება ყველანაირი ინფორმაცია იყოს, რაც მომხმარებელს შეიძლება დასჭირდეს. არავინ იცის კონკრეტულად რისი მოძებნა სჭირდება ამა თუ იმ მომხმარებელს, ასე რომ Crawler „თავს

იზღვევს“. ის საფუძვლიანად იკვლევს საიტის ყველა გვერდის შიგთავსს და იმახსოვრებს ყველა ბმულს/URL[40].



ნახაზი 4 Crawler-ები

1.3.4 არალეგალური მეთოდები: Spamdexing

Spamdexing (ასევე ცნობილი, როგორც სამიუბო სისტემის სპამერი, ძებნის სპამი, ვებ სპამი და ა.შ) არის იმ მეთოდების ნაკრები, რომელთა მიზანი ერთია, არალეგალური გზით, წარმატებული ოპტიმიზაციის მიღწევა. არალეგალურ გზაში იგულისხმება: არასაჭირო/არასწორი ფრაზების

გამეორება. მაგალითად თუ ვებ-გვერდის ავტორს აქვს ვებ-გვერდი, სადაც როკ მუსიკაზეა ინფორმაცია და ის სპამურად ამეორებს ერთიდაიგივე ტეგებს, საიტის შიგთავსში ფარულად ათავსებს ამ სიტყვებს, მაგალითად "Rock Music". რა თქმა უნდა ერთხელ ამ სიტყვების გამოყენება შეიძლება, მაგრამ თუ ეს გავამრავლეთ და ბევრჯერ დავწერეთ ერთსა და იმავე ადგილზე, მაშინ დიდი შანსია საძიებო სისტემამ ეს Spamdexing-ის მცდელობად აღიქვას და თქვენი ვებ-გვერდი დაბლოკოს[40].

1.4. ძიების სტრუქტურა Google-ს მაგალითზე

დღეისათვის, ყველაზე მძლავრ საძიებო სისტემას ფლობს კომპანია გუგლი. (google) გუგლის საძიებო სისტემას ინტერნეტში ძიებისათვის გააჩნია სამი სახის საშუალება:

- გუგლბოტი (googlebot).
- ინდექსატორი (indexer).
- შეკითხვის, მოთხოვნის პროცესორი (query processor).

გუგლბოტი წარმოადგენს რობოტს, რომელიც მოგზაურობს ინტერნეტ სივრცეში, პოულობს ვებ-გვერდებს და გადასცემს მათ ინდექსატორს შემდგომი დამუშავებისათვის. ეს შეიძლება შევადაროთ კიბერსივრცეში „გახლართულ“ ობობას. გუგლბოტი შედგება უამრავი კომპიუტერისაგან. მას უმოკლეს დროში შეუძლია ათასობით განსხვავებული ვებ-გვერდის ერთდროული ჩატვირთვა. გუგლბოტი ვებ-გვერდებს ორი განსხვავებული მეთოდით პოულობს:

- url დამატების ფორმა ([www.google.com /addurl.html](http://www.google.com/addurl.html)).
- გვერდების ჩამოცოცებით.

პირველი მეთოდის ნაკლი იმაში მდგომარეობს, რომ სპამერები აგზავნიან ავტომატურ ბოტებს, რომლის მიზანია დამატების ფორმის „დაბომბვა“. გუგლის კომპანია ამ პრობლემას მხოლოდ იმით ებრძვის, რომ

შლის ფარული ტექსტების შემცველ მისამართებს, აქვს გადამისამართების ბმული. როდესაც ბოტი წამოიღებს გვერდს, ის შეარჩევს ყველა ბმულს, რომელიც უკავშირდება ამ გვერდს და წარმოგვიდგენს მათ შემდეგი მოხმარებისათვის, ხშირად აწყდება სპამს. ბოტს შესწევს უნარი სწრაფად ააგოს კავშირების სქემა და ფართოდ დაიპყროს ქსელი. გუგლბოტის დანიშნულებაში შედის აგრეთვე გააკეთოს ანალიზი, თუ რამდენად ხშირად ვუბრუნდებით ერთსა და იმავე ვებ-გვერდს და წარმოადგინოს ვერ-რესურსები განსაზღვრული რიგითობით. იგი გამორიცხავს ინდექსში დუბლიკატების არსებობას და მუდმივად ანახლებს მას.

ინდექსი არის მონაცემთა ბაზა, რომელშიც ინფორმაცია ინახება ალფავიტის მიხედვით. იგი შეიცავს ინფორმაციას სიტყვებისა და მათი ადგილმდებარეობის შესახებ. მონაცემთა სტრუქტურა იძლევა საშუალებას, მივიღოთ სწრაფი დაშვება დოკუმენტებზე, რომელიც შეესაბამება ჩვენს მოთხოვნას. ძებნის გაუმჯობესების მიზნით, ინდექსატორი იგნორირებას უკეთებს მაკავშირებელ სიტყვებს როგორცაა („ან“, „და“) ერთ ასობგერას, ერთ ციფრს.

შეკითხვის მოთხოვნის პროცესორი შედგება რამდენიმე ნაწილისაგან. მის ინტერფეისს წარმოადგენს „ძებნის ყუთი“, „ძრავი“, რომელიც მოთხოვნებს შეუსაბამებს სათანადო დოკუმენტაციას და შედეგების ფორმატირების საშუალება.

გვერდის ხარისხი (Pagerank) არის გუგლის სიტემა, რომლის დანიშნულებაც ვებ-გვერდების შეფასება. ფასდება ვებ-გვერდების ხარისხიანობა. გვერდი, რომელსაც მაღალი PageRank გააჩნია, ითვლება, რომ უფრო მოთხოვნადია. გუგლს გააჩნია 100-ზე მეტი ფაქტორი PageRank-ის გამოსათვლელად. ასეთ დროს ფასდება:

- საიტის პოპულარობა.
- მდებარეობა და ზომა.
- ძიების კრიტერიუმები.
- შეთავაზებული გვერდები.

- საიტიდან სხვა საიტის პოვნის კრიტერიუმების სიახლოვე.

გუგლი აგრეთვე იყენებს მანქანას თვითშესწავლის „უნარი“, რათა ავტომატურად წარმართოს სისტემის მუშაობა; ისწავლოს კავშირები და ასოციაციური მსგავსებები მონაცემებს შორის. მაგალითად, მას შეუძლია შემოგთავაზოთ მოთხოვნილ ძიებაზე ალტერნატივა: „ხომ არ გულისხმობდით?“[9].

I თავის დასკვნები

1. თანამედროვე გლობალურ ქსელებში ძიების ამოცანებისა და პრობლემების გადასაჭრელად გამოყენებული მეთოდებისადმი მიძღვნილი ლიტერატურის მიმოხილვა და ანალიზი გვიჩვენებს, რომ ძიების თანამედროვე საშუალებები იყენებენ ოპტიმიზაციის მეთოდებს და ამის საშუალებით ახდენენ ძიების ოპტიმიზაციას.
2. შედეგად ვღებულობთ იმას, რომ მიიღწევა ძიების ხარისხის ნაკლებ ეფექტურობა.
3. ინტერნეტ ძიების პროცესში შედეგი დამოკიდებულია ვებ-რესურსის ციფრულ მაჩვენებელზე.
4. ძიების მეთოდები არ არის ერთნაირად საუკეთესო ნებისმიერ შემთხვევაში.
5. განხილულია Seo-ს ოპტიმიზაციის მეთოდები.
6. განხილულია Crawler მუშაობის სტრუქტურა.
7. ნაჩვენებია, რომ ინფორმაციის დაკარგვის გამორიცხვის მიზნით, Crawler ახდენს ყველა ბმულის დამახსოვრებას.
8. ნაჩვენებია რომ ოპტიმიზაციის პრობლემა არის ერთ-ერთი მთავარი პრობლემა. ინტერნეტ ძიების პროცესში.

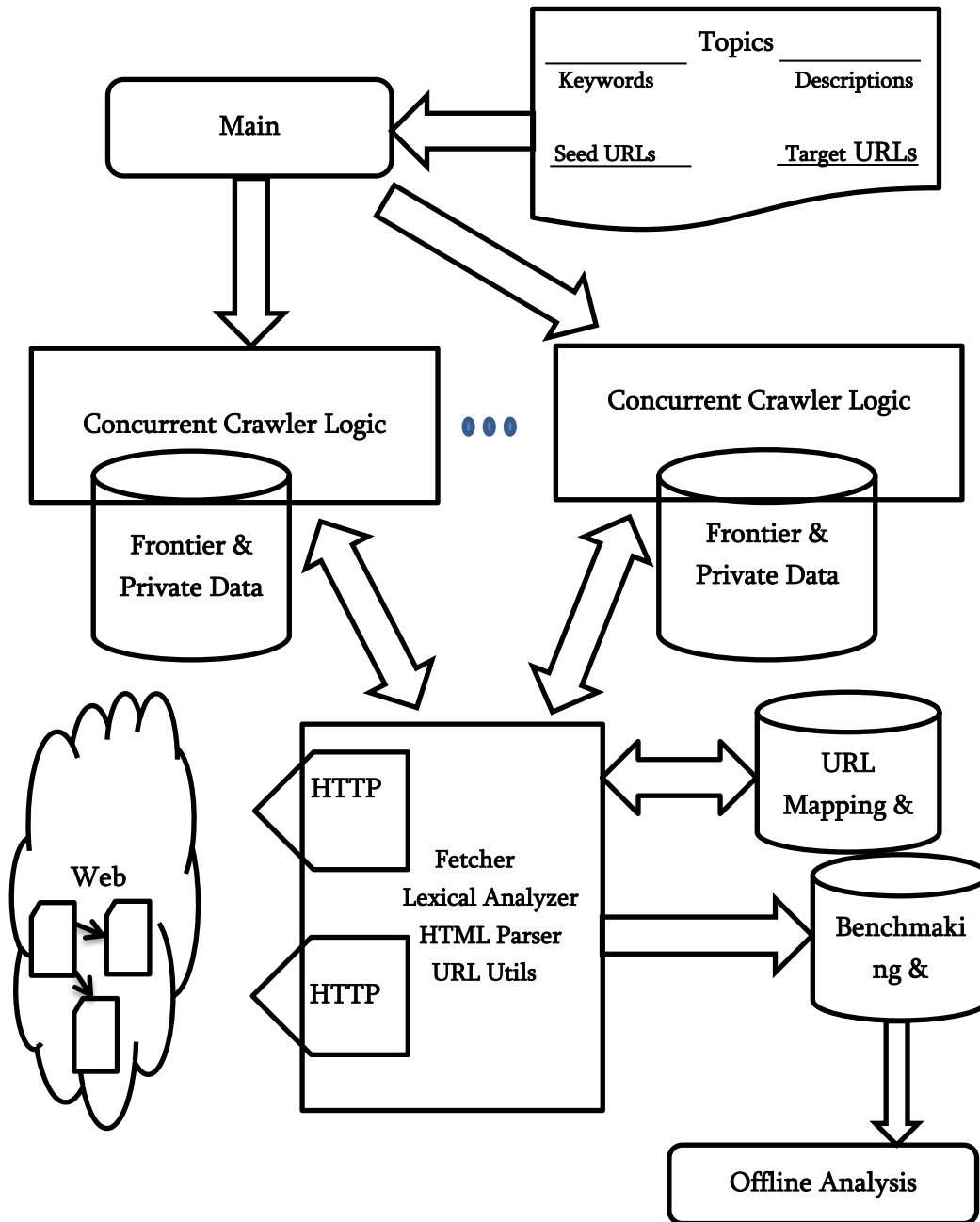
შედეგები და მათი განსჯა

თავი II

ინტერნეტ ძიების მოდელი

2.1. ინტერნეტ ძიების არქიტექტურა

ნახაზი 5-ზე წარმოდგენილია ძიების არქიტექტურა განვითარების ფარგლებში. სისტემა დაპროექტებულია ისე, რომ ძიების ალგორითმები კაფსულირებულია ძიების მოდულში, რომელიც ადვილად მიერთებადია სისტემაზე მისი ინტერფეისით. ყველა მოდულის მონაცემთა სტრუქტურა გაზიარებადია საზოგადოდ ყველა სხვა დანარჩენი მოდულისათვის. ყველა მცოცავს შეუძლია ეწვიოს MAX_PAGES. საიტის ჩამოტვირთვას სჭირდება დაახლოებით 10 წამი, ხოლო დიდი საიტების შემთხვევაში, ის იმდაგვარად არის დახარისხებული რომ, ჩვენ ვიყენებთ მხოლოდ პირველ 10kb-ს. პროტოკოლი HTTP GET იძლევა უფლებას მოხდეს ყველა საიტის კონტენტის text/html ფილტრაცია [1][10][14].



ნახაზი 5 ძიების არქიტექტურა

2.2. შეზღუდვები

Crawler-ები მოიხმარენ შემდეგ რესურსებს:

- ✓ ქსელის სიჩქარე გვერდის ჩამოსატვირთად.
- ✓ მეხსიერება, რომ შეინარჩუნოს მონაცემთა სტრუქტურები.
- ✓ საკუთარი ალგორითმი.
- ✓ CPU.

✓ დისკის მოცულობა.

ცხადია, რაც უფრო რთულია ალგორითმი ის მით უფრო მეტ რესურსს მოითხოვს. იმისათვის რომ მოხდეს სამართლიანი შედარება ძიების ალგორითმებისა, არსებობს შემდეგი ორი გზა:

1. თვალყური ვადევნოთ CPU-ს, დრო რომელიც დაჭირდება თითოეულ crawler თითოეული გვერდისათვის და მოვახდინოთ იგნორირება მოდულის ჩამოტვირთვის დროზე, რომელიც საერთოა ყველა crawler-ისათვის. დროის ნიშნული საჭიროა მხოლოდ crawler სპეციფიკურ ოპერაციებში. მონიტორინგი CPU-ზე და მის მიერ გამოყენებული დრო საჭიროა სხვადასხვა სირთულის ძიების ალგორითმის შედარებისათვის.
2. შევუზღუდოთ მეხსიერება, თითოეულ crawler და მივცეთ ბუფერის განსაზღვრული ზომა. ბუფერი გამოყენებულ იქნება crawler-ის მიერ დროებითი ბმულებისათვის. თითოეულ crawler-საქვს უფლება აკონტროლოს მისი MAX_BUFFER ბმულები. თუ ბუფერი შეივსება მაშინ crawler-ს შეუძლია გადაწყვიტოს რომელი ბმული შეცვალოს და მის ნაცვლად ჩასვას ახალი. ბუფერის ზომა შეიძლება სხვადასხვანაირი 2^8 -დან 2^{11} -მდე, რომ შეფასდეს, რამდენად იყენებს მეხსიერების რესურსს ძიების ალგორითმი

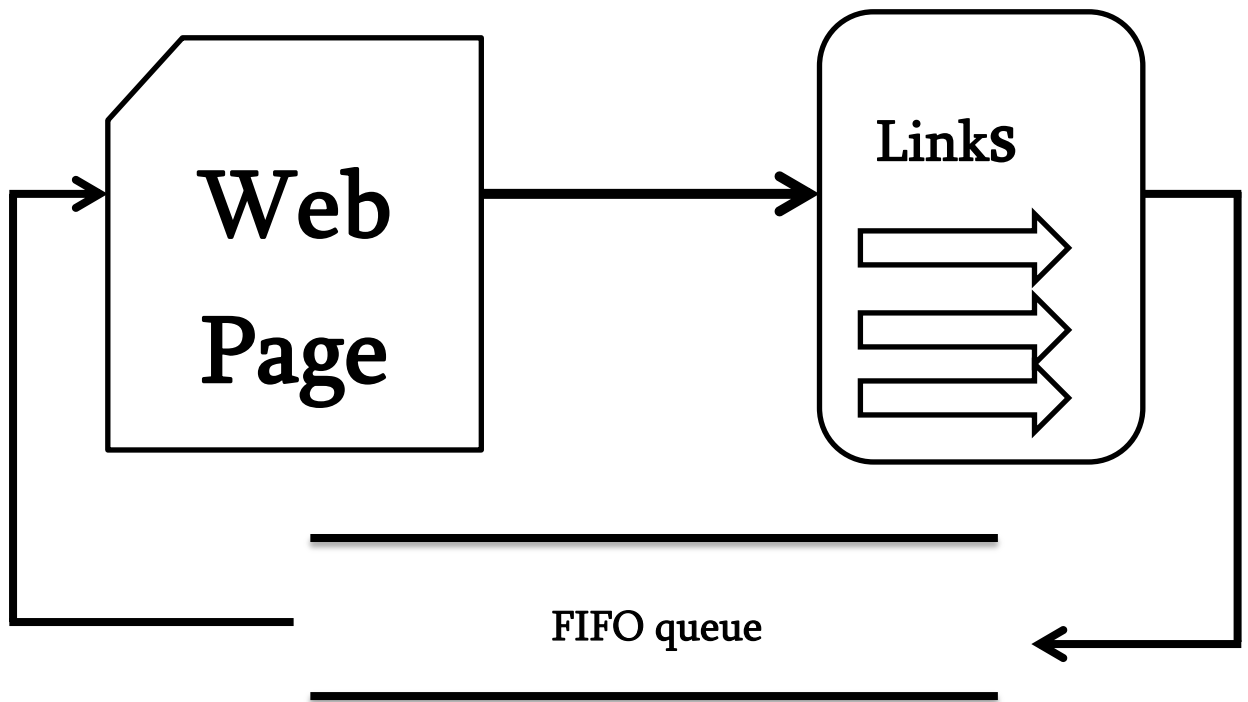
ალგორითმების შეფასებისათვის საჭიროა, მოხდეს თემების და მიზნების თეორიული გენერირება, რომელსაც გამოიყენებს საძიებო სისტემა. თუმცა ასეთი მიდგომა არის საკმაოდ ძვირი, გამომდინარე იქედან, რომ ძნელია შევინარჩუნოთ მონაცემთა ასეთი სტრუქტურები, ვებ-სამყაროს დინამიური თვისებებიდან გამომდინარე[1].

2.3. ძიების ალგორითმები

Crawler-ებს გააჩნიათ ჰიპერკავშირების სტრუქტურა, რომ მოახდინონ ახალი გვედების მოძიება და მისი ინდექსში განთავსება. ყოველი ახალი ვებ-გვერდი ემატება ვიზიტორებული ვებ-გვერდების სიაში და ის მოხსენებულია როგორც ძიების ბოლო საზღვარი. ყველაზე მთავარი ძიების დროს არის, რომ მოხდეს მომდევნო ბმულის განსაზღვრა და დაცული იყოს ე.წ. საზღვარი. მისი სტრუქტურა გრაფიკულად წარმოდგენილია ნახაზი 6-ზე[1].

2.3.1. Breadth-First

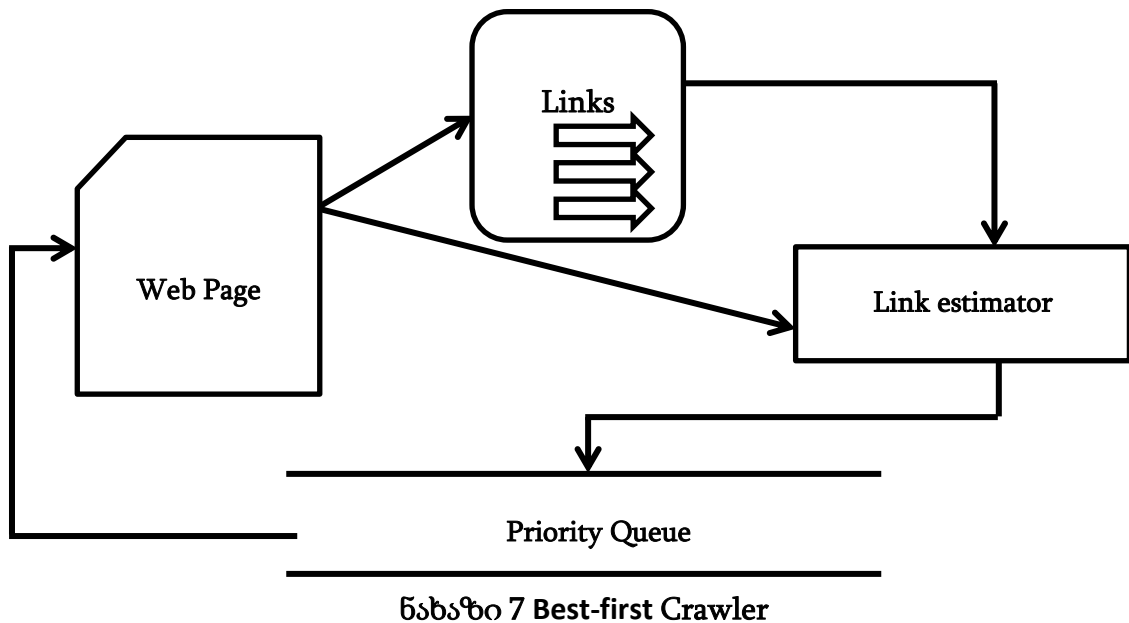
Breadth-First-იყენებს ძიების მარტივ სტრატეგიას, ის შემუშავებული იქნა 1994 წელს. მისი მუშაობის სქემა მარტივია, ის ახდენს ახალი ბმულების მოძიებას და ქმნის დალაგებულ ინდექსს. მის სასაზღვრო ზოლს წარმოადგენს ე.წ. FIFO რიგები. ბოლო ლინკი წარმოადგენს საზღვარს და მიმდინარე ძიებისას შესაძლებელია მხოლოდ ერთი ლინკის დამატება, რომელიც ხდება ბმულთა მონაცემთა ბაზის ახალი საზღვარი[1].



ნახაზი 6 Breadth-First Crawler

2.3.2 Best-First

Best-First- ალგორითმი შექმნილი იქნა CHO-სა და Hersovici-ს მიერ, 1999 წელს. მისი იდეა მდგომარეობს შემდეგში, რომ მოახდინოს კონკრეტული კრიტერიუმით ბმულების შეფასება და დააღაგოს პრიორიტეტული სიით.(იხ. ნახაზი 7) [1]



2.3.3 Hits

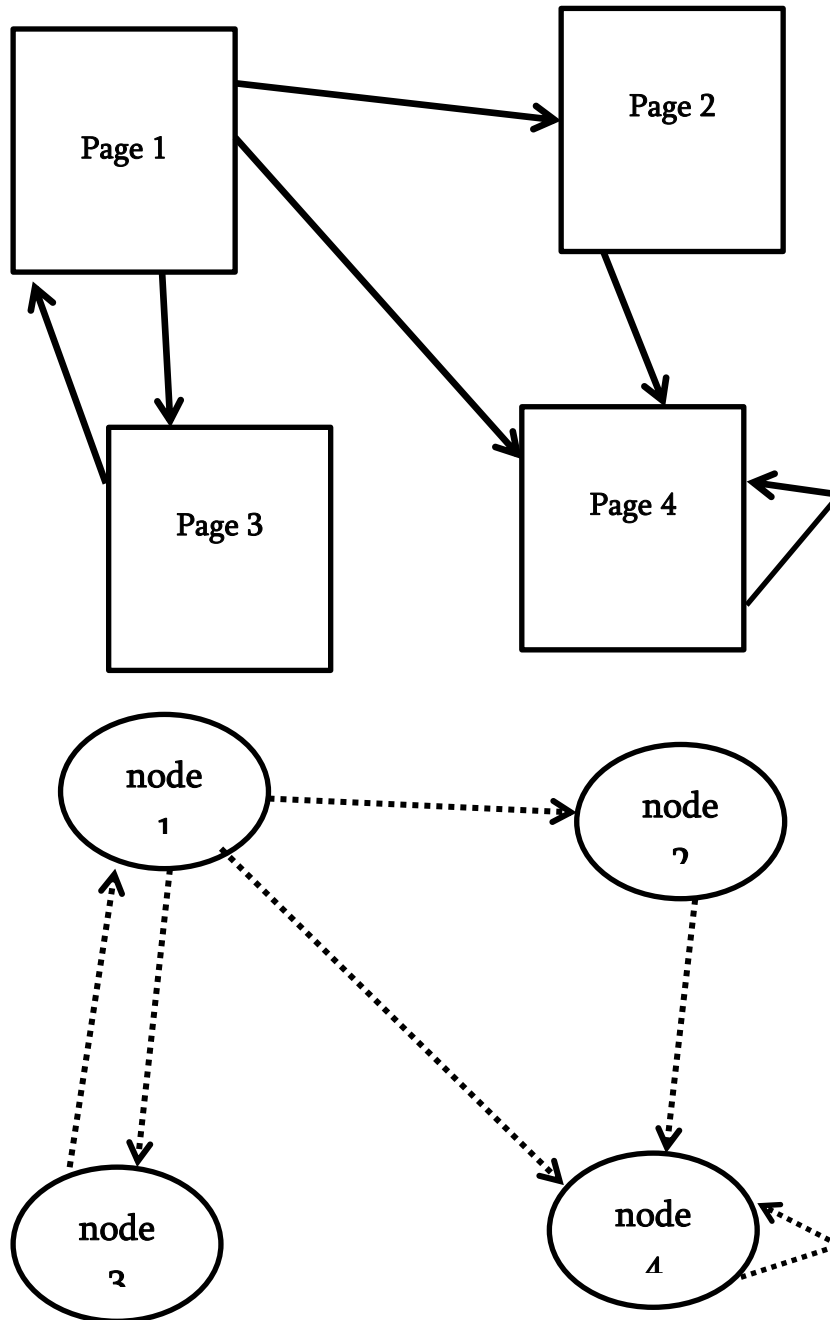
Hits(Hyperlink-Induced Topic Search)- წარმოადგენს ბმულის ანალიზის ალგორითმს. ის ითვლება PageRank-ის წინამორბედად. შეიძლება მოხსენებულ იქნეს როგორც ჰაბი. ის წარმოადგენს ავტორიტეტული საიტების კატალოგს, რომელიც მიუთითებს სხვა, ავტორიტეტულ ვებ-გვერდებზე. სხვა სიტყვებით რომ ვთქვათ: ჰაბი წარმოგვიდგენს იმ ვებ-გვერდს, რომელიც მოუთითებს ბევრ სხვა ვებ-გვერდზე. აგრეთვე კარგი ავტორიტეტით სარგებლობს ის ვებ-გვერდი, რომელიც აკავშირებს ჰაბებს ერთმანეთთან.

ვებ-გვერდს ენიჭება ორი ქულა:

- მისი კონტენტის მიხედვით.
- ჰაბის მნიშვნელობის მიხედვით (რომელიც აფასებს მისი ლინკებს სხვა გვერდებზე).

ჰაბის ალგორითმი ეფუძნება მიმართულ გრაფს $G(V,E)$, სადაც V არის წარმოდგენილი გვერდების მწვერვალი და E არის კიდეები, რომელიც შეესაბამება ბმულს. მცდელობა იმისა, რომ კომპიუტერულად

განისაზღვროს ავტორიტეტულობა და ჰაბი, საჭიროებს კონკრეტული გრაფის ანალიზს ქსელში(იხ. ნახაზი 9)[3].

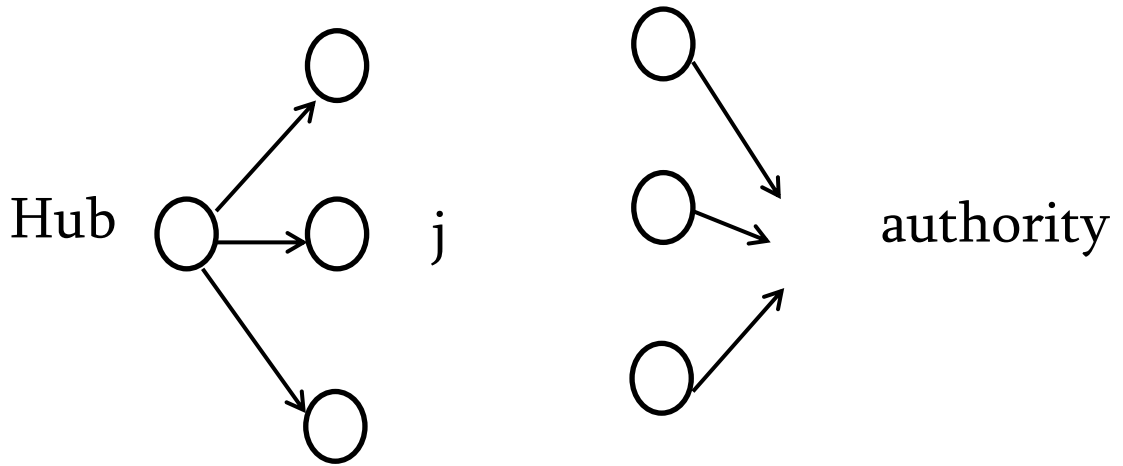


ნახაზი 8 მიმართული გრაფი

წარმოიქმნება რეკურსიული ურთიერთდამოკიდებულება. ჰაბი მიუთითებს ბევრ ავტორიტეტზე და ავტორიტეტი მიუთითებს ბევრ ჰაბზე(იხ.ნახაზი 2.5).

- ავტორიტეტი: მიუთითებს დიდი რაოდენობის ჰიპერკავშირს.

- **ჰაბი:** ვებ-გვერდი, რომელიც მიუთითებს მრავალ ავტორიტეტულ ვებ-გვერდზე და თვითონ წარმოადგენს რესურსს.



ნახაზი 9 ჰაბი და ავტორიტეტი

2.3.4 Shark-Search

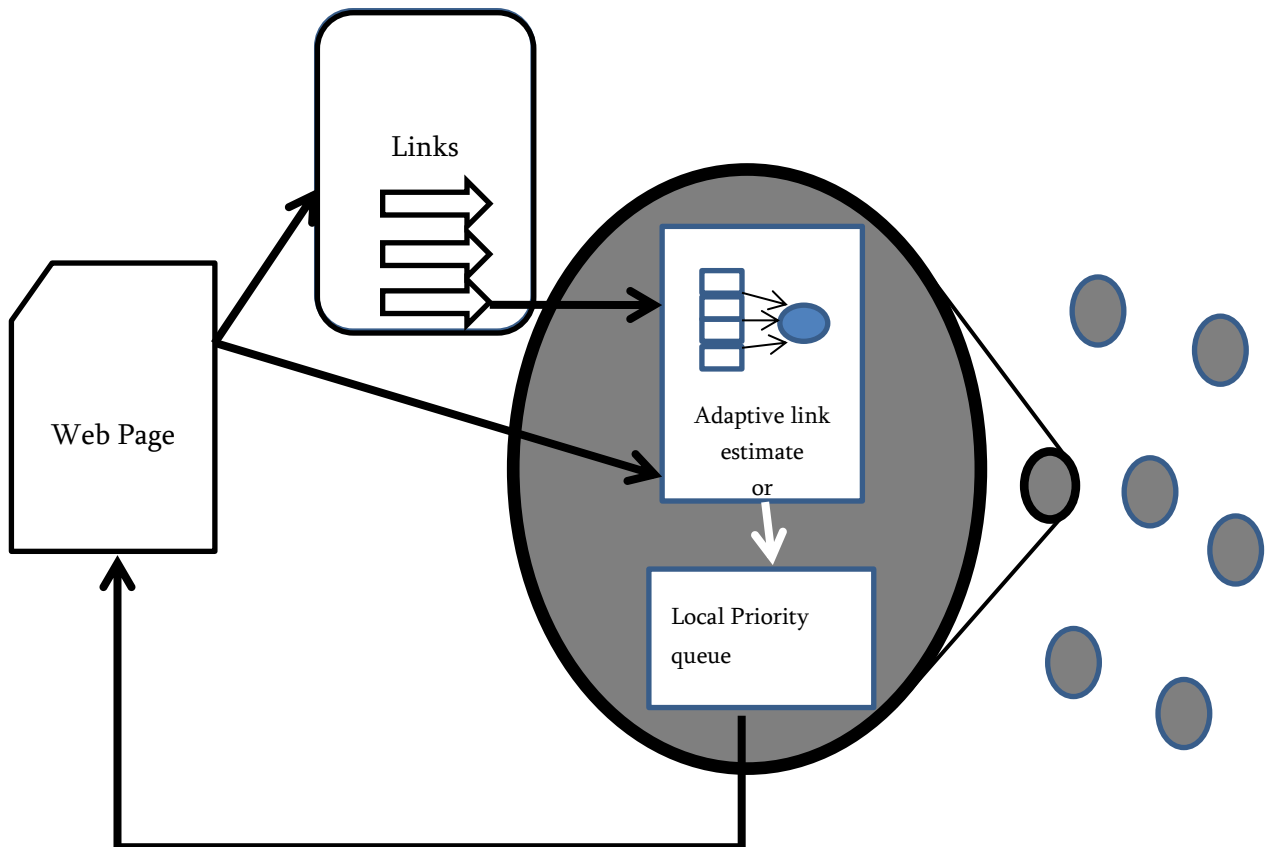
Shark-Search- „ზვიგენი-ძიება“ წარმოადგენს „თევზი-ძიების“ აგრესიულ ფორმას, ის შემუშავებული იქნა 1998 წელს. Crawler უფრო მტკიცე ძიებას ანხორციელებს web სივრცის იმ არეალში, სადაც ნაპოვნი იქნება რელევანტური გვერდები. ამავე დროს ალგორითმი დაუსრულებელ ძიებას ანხორციელებს იმ web რაიონებში, რომელიც არ იძლევა რელევანტურ გვერდებს. განსხვავებით „თევზის-ძიებისა“ , მას აქვს ორი გაუმჯობესებული ფუნქცია:

1. ის იყენებს ძიების ორობით უწყვეტ ფუნქციას.
2. აქვს უფრო მეტად დახვეწილი შეფასების სისტემა სასაზღვრო ბმულის შეფასებისათვის.

„ზვიგენის-ძიება“ წარმოადგენს „პირველი-საუკეთესო“ ძიების გაუმჯობესებულ ვარიანტს[1].

2.3.5 Info Spider

InfoSpider- ძიების აგენტების ადაპტირებული პოპულაცია, იყენებს ცვალებადი მოთხოვნის ვექტორს და ნეირონულ ქსელს, იმისათვის, რომ განსაზღვროს, რომელი ბმულია სასარგებლო, ნახაზი 10-ზე წარმოდგენილია Infospider-ის პოპულაცია[1].



ნახაზი 10 Infospider პოპულაცია

მისი ალგორითმი არის უკეთესად შესრულებული ვიდრე BFS, გამომდინარე მისი ლოკალური ქცევით და მეხსიერების არარსებობით.

აგენტი მხოლოდ ეწვევა ვებ-გვერდს შესაბამისი ბმულით. Infospider იყენებს ნოველის ალგორითმს, რომელიც გულისხმობს, რომ საძიებო მანქანამ ისწავლოს ბმულის შეფასება ნეირონული ქსელის დახმარებით.

ნეირონული ქსელის გამოსავალი იყენებს ხარისხის შეფასების ციფრულ მეთოდს. თითოეული ბმული წარმოადგენს შემავალ მონაცემს. შეფასება ითვლება ისეთ შეფასებად, რომლის შედეგი დაფუძნებულია

კოსინუსურ მსგავსებაზე, აგენტის სიტყვებსა და გვერდის ბმულებს შორის. აგენტი იყენებს სტოქასტურ შემრჩევს, რომ აირჩიოს ბმულების ერთი ალბათობა.

$$\Pr(\lambda) = \frac{e^{\beta\mu(\lambda)}}{\sum_{\lambda' \in p} e^{\beta\mu(\lambda')}} \quad (1)$$

სადაც: λ არის ბმული p გვერდიდან. $\mu(\lambda)$ არის კომბინირებული ქულა და β პარამეტრი β , არეგულირებს სელექტორის სიხარბეს. აგენტი იღებს ენერჯიას სიტყვებისა და გვერდების მსგავსების პროპორციულად.

აგენტი გადადის ახალ არჩეულ გვერდზე მხოლოდ იმ შემთხვევაში თუ ფუნქცია დააბრუნებს ჭეშმარიტ მნიშვნელობას.

$$\Pr(\delta) = \frac{1}{1 + e^{-\delta/T}} \quad (2)$$

სადაც δ არის განსხვავება ახალსა და მიმდინარე ვებ-გვერდს შორის. $T=0.1$ არის პარამეტრის „ტემპერატურა“ [1].

2.3.6 Google panda

Panda არის შედეგების რანჟირების ალგორითმი. მისი დანიშნულებაა შეამციროს „დაბალი ხარისხის ვებ-გვერდები“ და მოგვაწოდოს მაღალი ხარისხის დალაგებული ვებ-რესურსების სია. ის გამოირიცხავს ვებ-

გვერდებს, რომელთაც თავის მხრივ მოახდინეს ტექსტისა და წყაროების დუბლირება სხვა საიტებიდან. ისტორიულად, საძიებო სისტემებში ისეთი ვებ-გვერდები წინაურდებოდნენ, რომლებიც ტექსტებს სხვა გვერდებიდან იღებდნენ, თუმცა Google – მა ამ სისტემის მეშვეობით მოახდინა ამ ვებ-გვერდების გარიცხვა საძიებო სისტემიდან[24].

2.3.7 Google pinguin

იგი წარმოადგენს „გუგლ პანდას“ გაუმჯობესებულ ვერსიას რომლის მიზანია ისეთი ვებ-გვერდების დასჯა, რომლებიც სხვადასხვა ტექნოლოგიების გამოყენებით ცდილობდნენ საძიებო სისტემაზე მანიპულირებას[25].

2.3.8 Pagerank

Pagerank- შეიმუშავა კომპანია Google-მ 1998 წელს. ეს არის ვებ გვერდის “მნიშვნელობის” აღმნიშვნელი ციფრული მაჩვენებელი. რაც უფრო მეტია შემომავალი (მომმართავი) ბმული, მით უფრო მაღალი ხდება მისი “მნიშვნელობა”. გარდა ამისა, A გვერდის “წონა” განისაზღვრება მასზე მიმმართავი B გვერდის “წონით”. შესაბამისად, PageRank არის გვერდის “წონის” კალკულაციის მეთოდი, მასზე მიმმართავი ბმულების “მნიშვნელობის” მიხედვით. იგი გამოითვლება ფორმულით:

$$PR(p) = (1 - \gamma) + \gamma \sum_{\{d \in in(p)\}} \frac{PR(d)}{|out(d)|} \quad (3)$$

სადაც p არის გვერდი, საიდანაც იწყება დათვლა $in(p)$ არის გვერდების კომპლექტი, რომელიც მიუთითებს $out(d)$ -ზე, გამომავალ ბმულებზე. $\gamma < 1$ კი არის ბიძგის ფაქტორი, ანუ ის, რომ შემთხვევითი სერფერი ახდენდეს ვებ-

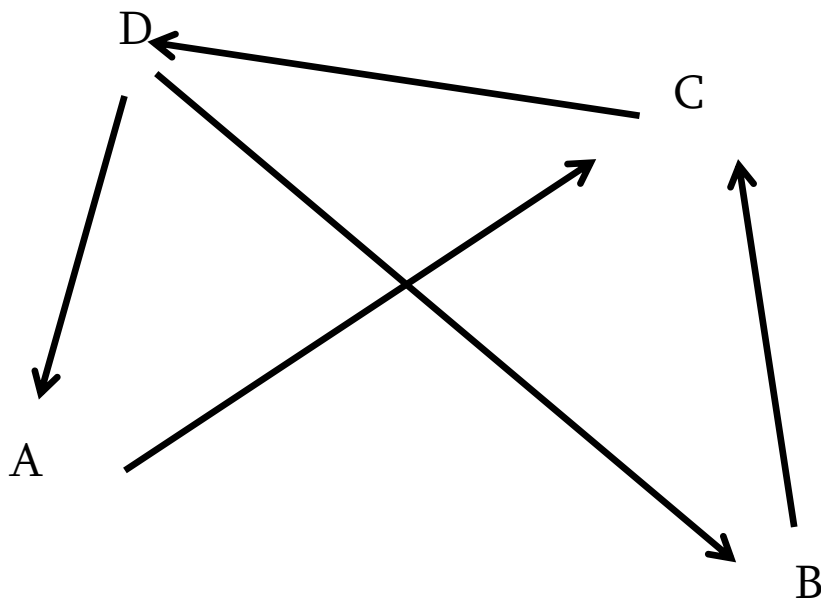
განვიხილოთ კონკრეტული ამოცანა ნახაზი 12-ზე მოცემულია კონკრეტული დამოკიდებულება ვებ-გვერდებს შორის. გვერდი A არის გვერდი B-ს უკუბმული, სანამ C და B არიან D გვერდის უკუბმულები, ამ შემთხვევაში PageRank- ის ალგორითმს ექნება სახე

$$PR(P)=(1-d)+d(PR(T1)/C(T1)+.....PR(Tn)/C(Tn)) \quad (4)$$

სადაც PR(P)-არის P გვერდის PageRank

PR (Ti)-არის Ti-ს PageRank რომელიც უკავშირდება გვერდს

D-არის ბიძგის ფაქტორი, რომელიც მერყეობს 0-სა და 1-ს შორის.



ნახაზი 12 Pagerank-ის ამოცანა

კავშირი A გვერდიდან B გვერდისკენ ნიშნავს, რომ A გვერდი „ხმას აძლევს“ B გვერდის წონას და ხდის მას უფრო პრიორიტეტულს.[1][3][38]

2.3.9 Hummingbird

2013 წლის სექტემბერში Google – მა გამოუშვა Google – ის Hummingbird Update, შეცვლილი ალგორითმი, რომლის მიზანია გააუმჯობესოს Google – ის ბუნებრივი ენა და უკეთ გაიგოს საიტის აზრობრივი მნიშვნელობა. იგი ფოკუსირდება კონკრეტულ სიტყვებზე და მის მნიშვნელობებზე წინადადებაში. მისი მიზანს წარმოადგენს იპოვნოს ისეთი გვერდები, რომელსაც შეესაბამება შეკითხვის შემადგენელი სიტყვების უმრავლესობა. იგი წარმოადგენს „ცოდნის გრაფს“ რომლის მიზანია განსაზღვროს სიტყვებს შორის კავშირები, ცნებები და ა.შ[26].

2.3.10 WordNet

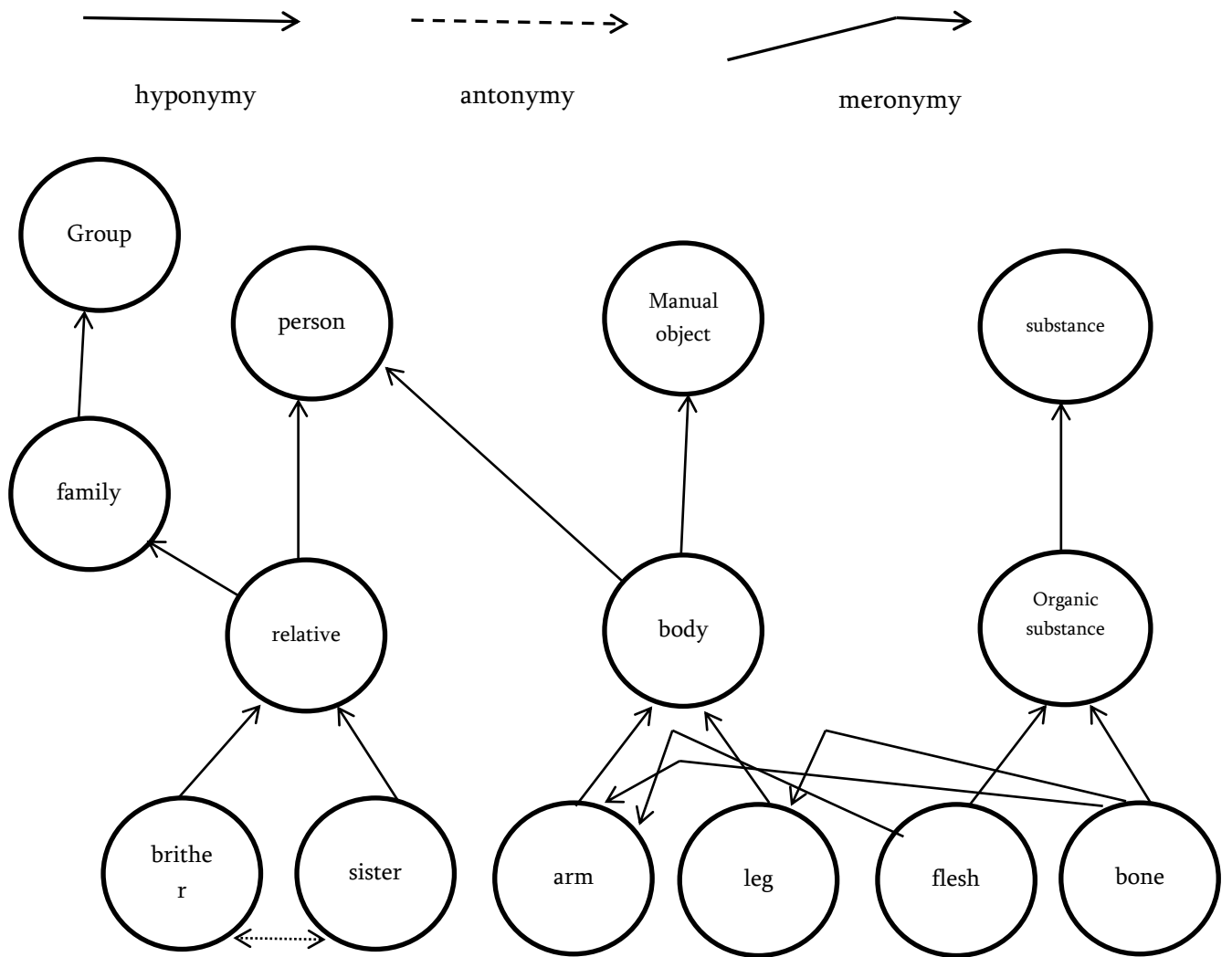
Wordnet არის ელექტრონული ლექსიკონი, რომელიც შეიცავს სიტყვებს შორის კავშირებს და აქვს ხის მსგავსი სტრუქტურა. ეს არის ცდის რუკა, გაიგოს კავშირი ადამიანის მიერ გაგებად სიტყვებსა და ელექტრონულ მონაცემთა ბაზას შორის. იგი შეიძლება თამამად ჩაითვალოს მონაცემთა მოპოვების კარგ საშუალებათ, რადგან, პირველ რიგში მისი მუშაობა ეფუძნება მორფოლოგიურ პროცესს, მაგალითად სიტყვა: „გააკეთა“, „აკეთებს“, „გააკეთებს“ არის ერთიდაიგივე მნიშვნელობის და მათ შორის სხვაობა მხოლოდ შესრულების დროშია. შესაბამისად ჩვენ გვინდა, რომ ბაზაში მოხდეს სიტყვა „კეთება“. სწორედ ამ პროცესს ემსახურება Wordnet. ის ამცირებს ატრიბუტების რაოდენობას.

მეორე მხრივ, ატრიბუტები შეიძლება არ იყოს სასარგებლო და Wordnet-ს აქვს უფლება მოიპოვოს სასარგებლო სიტყვები და წაშალოს ე.წ. stop სიტყვა. Stop სიტყვა ეს არის ისეთი სიტყვა, რომელიც გადაიცემა ნელა და არ შეიცავს გამოყენებად ინფორმაციას.

მესამე- მას შეუძლია დაკავდეს ინტელექტუალური სიტყვების ანალიზით, რომ შეამციროს ატრიბუტების რაოდენობა. ეს პროცესი საკმაოდ კარგი იქნება თუ მოხდება რამოდენიმე სიტყვის ერთ სიტყვაში

გაერთიანება. მაგალითად : თუ ერთი გვერდი შეიცავს სიტყვებს: „ფანჯარა“, „კარები“, „ბუხარი“, „ოთახი“, „სახურავი“ და ა.შ. WordNet მოახდენს ამ სიტყვების ხის სტრუქტურაზე განაწილებას, იპოვის მათ შორის კავშირებს და მოგვცემს ერთ სიტყვას: „სახლი“. ცხადია კარგი შედეგია, მან მოახდინა 5 სიტყვის ერთ სიტყვაში გაერთიანება.

შევხედოთ ამას მეორე მხრიდან. აღნიშნული მოქმედება შეიძლება სარისკოც აღმჩნდეს, რადგან შესაძლოა WordNet-მა იპოვოს მათ შორის სხვა კავშირები და მოგვცეს სხვა არასწორი განმარტება. განვიხილოთ WordNet-ის მუშაობა. ნახაზი 13-ზე მოცემულია მისი ხის მსგავსი სტრუქტურა.



ნახაზი 13 WordNet-ის ხე

ნახაზზე ჩანს, რომ „ძმებს“ და „დეებს“ ჰყავთ ერთი საერთო კვანძი „ნათესავი“ ანუ „და“ და „ძმა“ შესაძლებელია შეიცვალოს ერთი სიტყვით „ნათესავი“. შეგვიძლია ვნახოთ მისი ალგორითმიც, რომელიც ქვემოთაა წარმოდგენილი[4][34][35].

2.4 რანჟირების ალგორითმების კვლევა

ინტერნეტში განთავსებული ინფორმაცია სწრაფად იცვლება: ყოველწამიერად ჩნდება და ქრება ინტერნეტრესურსები.

მიუხედავად ამისა, არსებობს რამდენიმე მნიშვნელოვანი რამ, რაც ინტერნეტში განთავსებულ ვებგვერდებს აერთიანებს: ისინი დაწერილია ერთსა და იმავე ენაზე ან მასთან დაკავშირებულ „დიალექტზე“ და

ერთმანეთთან ჰიპერბმულების საშუალებით არის დაკავშირებული. ჰიპერბმულების ერთ-ერთი მნიშვნელოვანი ნაწილია მისამართი, რომლითაც იგი სხვა რესურსზე მიუთითებს. ეს მისამართები მომხმარებლისთვის უმეტესწილად უხილავია. მომხმარებელი ერთი რესურსიდან მეორეზე მხოლოდ ჰიპერბმულის შესაბამის ვიზუალურ კომპონენტზე (მაგ., ტექსტი, ღილაკი, ნახატი) დაჭერით გადადის.

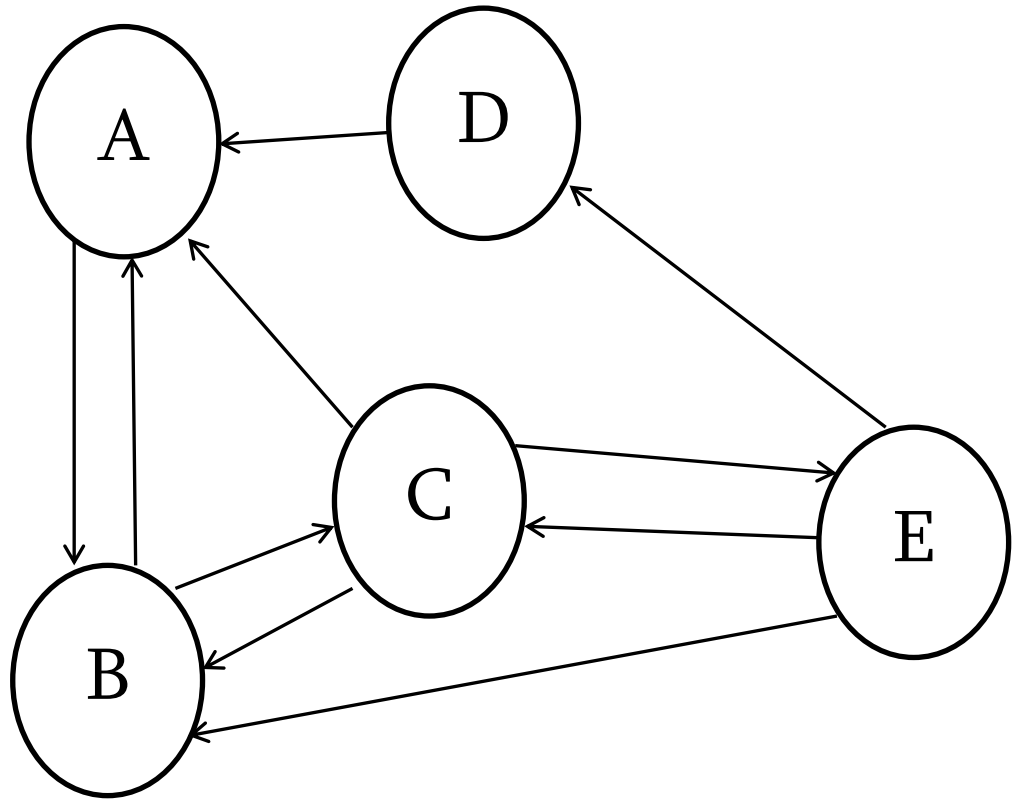
ყოველი რანჟირების ალგორითმი ეფუძნება ბმულების ანალიზს. ისინი შეიძლება დაიყოს ორ ქვეკატეგორიად

1. კითხვაზე დამოკიდებული.
2. კითხვაზე დამოუკიდებელი.

პირველ კატეგორიას მიეკუთვნება ალგორითმი, რომელიც ახდენს საიტის შეფასებას მასზე მიმართული ბმულების მიხედვით და ალაგებს პრიორიტეტულად. ხოლო მეორე კატეგორიას მიეკუთვნება ის ალგორითმი, რომელიც იყენებს შეკითხვაზე პასუხის გაცემის სისტემას, რომ გააძლიეროს ძიება. იქმნება ჰიპერკავშირების გრაფი ყოველი გვერდისათვის. თითოეულ კვანძში წარმოადგენს ვებ-გვერდების კავშირის ამლნიშვნელს, რომელთაც აქვთ ერთმანეთთან ჰიპერკავშირი. [2]

ინტერნეტის მოდელირებისთვის შეიძლება გამოვიყენოთ მიმართული გრაფი(იხ.ნახაზი 2.9). ამ გრაფში წვეროები ვებ-გვერდებს შეესაბამება, ამ წერტილების შემაერთებელი ისრიანი წირები ნიშნავს, რომ: გვერდზე არსებობს ერთადერთი ჰიპერბმული სხვა გვერდისადმი;

იმისათვის რომ მოვახდინოთ რანჟირების ალგორითმის დემონსტრირება ინტერნეტის ამ წარმოსახვით, „მინი მოდელზე“, დავუშვათ რომ მომხმარებელი იწყებს ნავიგაციას შემთხვევით შერჩეული ჰიპერბმულების გასწვრივ. იმ შემთხვევაში, როდესაც მას აქვს მხოლოდ ერთი არჩევანი (მაგალითად, თუ იგი იწყებს D გვერდიდან), მაშინ მას შეუძლია გადაადგილდეს მხოლოდ A გვერდისკენ. თუ მომხმარებელი იწყებს C გვერდიდან, მაშინ მას შეუძლია მიყვეს სამი ჰიპერბმულიდან ერთ-ერთს.(ე.ი. რომელ მათგანზეც გადაადგილების ალბათობა არის 1/3)



ნახაზი 14 მიმართული გრაფი

სხვა სიტყვებით რომ ვთქვათ, იმ გვერდიდან, რომელზეც მოცემულ მომენტში იმყოფება მომხმარებელი, იგი შემთხვევით ირჩევს ამ გვერდზე არსებული ჰიპერბმულებიდან ერთ-ერთს, თანაბარი ალბათობით. დავუშვათ რომ ეს მომხმარებელი გადაადგილდება ერთი ჰიპერბმულის გასწვრივ ერთი წუთის განმავლობაში. ისმება შეკითხვა: რომელ გვერდზე აღმოჩნდება ეს მომხმარებელი

- ერთი საათის შემდეგ?
- ერთი დღის შემდეგ?
- ჰიპერბმულების გასწვრივ გადაადგილებების ძალიან დიდი რაოდენობის შემდეგ?

უფრო ზუსტად, იმის დაშვებით რომ მისი გადაადგილების გზა ალბათურია, რისი ტოლი იქნება იმის ალბათობა იმისა, რომ დროის მოცემული მონაკვეთის შემდეგ იგი აღმოჩნდება მოცემულ გვერდზე?

აქ მოცემული ნახაზი დაგვეხმარება იმაში, რომ ამ კითხვას პასუხი

გავცეთ იმ შედარებით მარტივ შემთხვევაში, როდესაც მომხმარებელი იწყებს ინტერნეტში გადაადგილებას C გვერდიდან და გადაადგილდება ორი ბმულით. როგორც ნახაზიდან ჩანს, ამ გვერდზე განთავსებულია სამი გარეთ გამავალი ბმული, რომლებიც დაკავშირებულია A, B და E გვერდებთან. ამრიგად პირველი ნაბიჯის შემდეგ, მომხმარებელი შეიძლება აღმოჩნდეს A გვერდზე ალბათობით $1/3$, B გვერდზე ალბათობით $1/3$, C გვერდზე ალბათობით $1/3$. ეს ნახაზზე მითითებულია ასეთი ჩანაწერით:

$$P(A)=1/3; P(B)=1/3; P(E)=1/3 \quad (5)$$

ანალოგიური შინაარსისაა ჩანაწერები $P(C)=0$ და $P(D)=0$ ჩანაწერი მიუთითებს, რომ მომხმარებელი პირველი ნაბიჯის შემდეგ ვერ მოხვდება C ან D გვერდზე. რადგან გვერდიდან არ არსებობს ჰიპერბმულები ამ გვერდებზე. მივაქციოთ ყურადღება იმას,

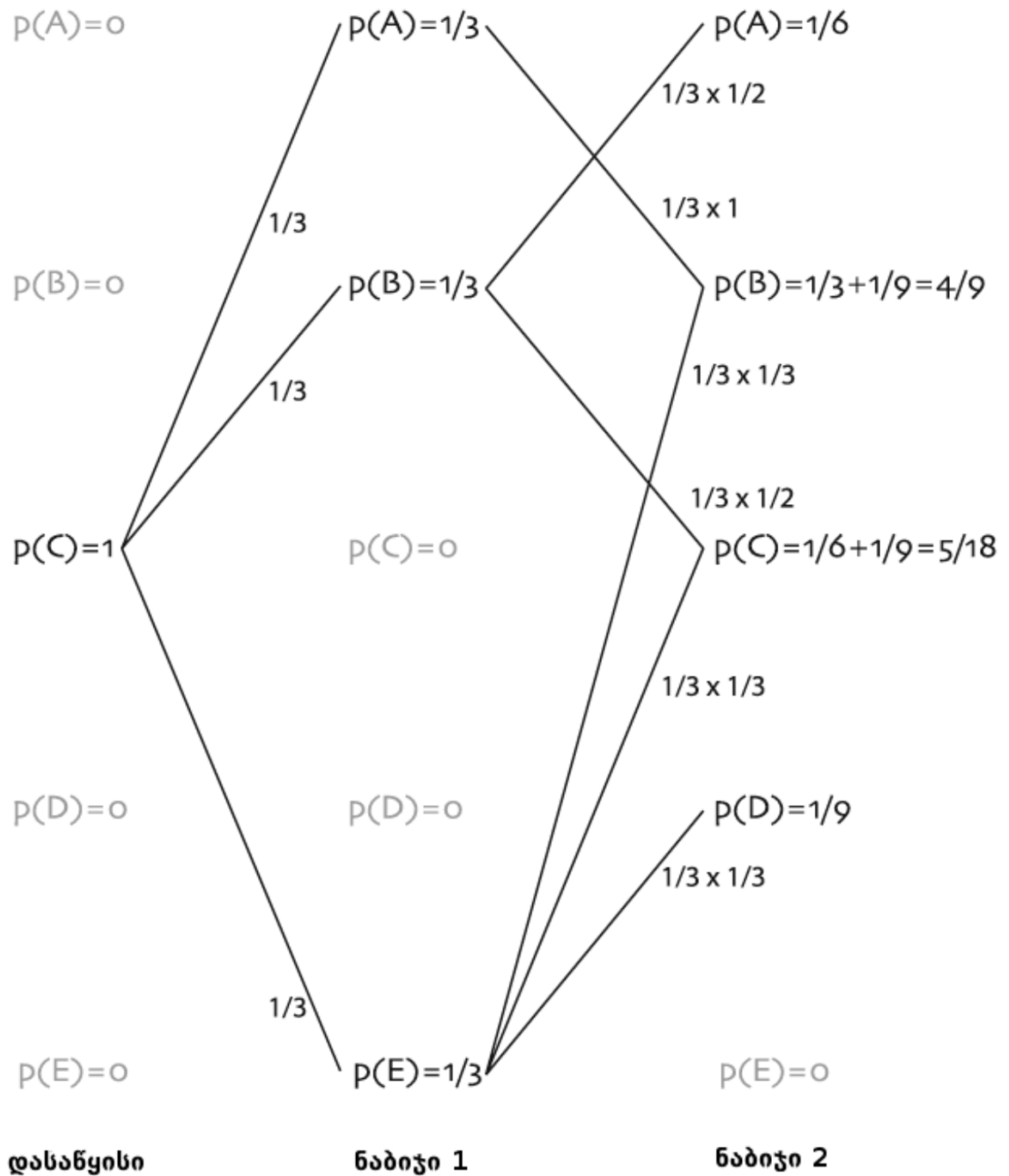
რომ ყოველი ნაბიჯის შემდეგ, შესაბამისი ალბათობების ჯამი 1-ის ტოლია:

$$P(A)+P(B)+P(C)+P(D)+P(E)=1 \quad (6)$$

მეორე ნაბიჯის შემდეგ, ალბათობა $P(B)$ უკვე აღარ არის $1/3$ ის ტოლი, რადგან გარდა $C \rightarrow B$ -სა არსებობს მეორე გზა, რომელსაც B წერტილში მივყავართ. $C \rightarrow E \rightarrow B$, თუ პირველი საფეხურის შემდეგ მომხმარებელი აღმოჩნდა E გვერდზე, მაშინ მას შეუძლია მეორე საფეხურზე თანაბარი ალბათობით აირჩიოს სამიდან ერთი $E \rightarrow B$, $E \rightarrow C$, $E \rightarrow D$. წარმოდგენილ ნახაზზე კარგად ჩანს, თუ რისი ტოლი იქნება ალბათობები $P(A)$, $P(B)$, $P(C)$, $P(D)$. ხდომილებების ნამრავლისა და ჯამის ალბათობების გამოთვლის წესის მიხედვით, მეორე საფეხურის შემდეგ:

$$P(A)=1/6; P(B)=4/9; P(C)=5/18; P(D)=1/9; P(E)=0 \quad (7)$$

მივაქციოთ ყურადღება, რომ ამ ალბათობების ჯამი კვლავ 1-ის ტოლია, რაც შესაბამისობაშია ალბათობის თვისებასთან (ხდომილებების სრული სისტემის ალბათობების ჯამი 1-ის ტოლია)(იხ.ნახაზი 15).



ნახაზი 15 ბმულებზე გადასვლა

ამის შემდეგ, უკვე შეგვიძლია ჩამოვყალიბოთ მეთოდის ზოგადი აღწერა: გავაგრძელოთ სხვადასხვა გვერდებზე მოხვედრის ალბათობების გამოთვლა ნაბიჯების რაოდენობის გაზრდასთან ერთად და შევადაროთ ერთმანეთს ეს

ალბათობები. კერძოდ, პირველი ორი ნაბიჯის შემდეგ, ყველაზე მაღალი ალბათობა აქვს იმას, რომ მოხვდებით B გვერდზე, თუმცა ნაბიჯების რაოდენობის გაზრდის შემდეგ შესაძლოა სურათი შეიცვალოს და სხვა რომელიმე გვერდზე მოხვედრის ალბათობა გახდეს უფრო მეტი. საუკეთესო მათემატიკური აპარატი, რომლის გამოყენებითაც შესაძლებელია ამ პროცესის შესწავლა, არის მარკოვის ჯაჭვების თეორია, რომლის შესწავლის ძირითადი ობიექტი არის ე.წ. გადასვლის (ან ჰიპერბმულების) მატრიცა (ცხრილი):

$$P = \begin{bmatrix} P_{11} \cdots P_{1n} \\ \dots\dots\dots \\ P_{n1} \cdots P_{nn} \end{bmatrix}$$

(8)

ამ მატრიცის ელემენტები განისაზღვრება ასე:

$$P_{ij} = \begin{cases} \frac{1}{k}, \\ 0 \end{cases} \quad (9)$$

თუ i გვერდიდან არსებობს გამომავალი ჰიპერბმულო j გვერდისაკენ
 თუ i გვერდიდან არ არსებობს გამომავალი ჰიპერბმულო j გვერდისაკენ
 სადაც რიცხვი k არის i გვერდიდან გამომავალი ჰიპერბმულების
 მთლიანი რაოდენობა. მაგალითად, აქ განხილული მინი ინტერნეტის
 შემთხვევაში (5 გვერდი), ჰიპერ-ბმულების მატრიცას აქვს ასეთი სახე

$$P = \begin{matrix} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} & \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \end{matrix}$$

(10)

ამ მატრიცის სვეტებში წერია შესაძლო მდგომარეობებში გადასვლის ალბათობები. მაგალითად, A გვერდიდან შესაძლებელია მხოლოდ B გვერდზე მოხვედრა, ამიტომ იმ სვეტში, რომელსაც თავზე აწერია A, მხოლოდ ერთი რიცხვი განსხვავდება 0-საგან და იგი 1-ის ტოლია.

B გვერდიდან შესაძლებელია მხოლოდ A და C გვერდებზე მოხვედრა (ერთნაირი ალბათობით), ამიტომ იმ სვეტში, რომელსაც აწერია B, ამ ორი გვერდის შესაბამის ადგილებზე წერია 1/2. იგივე შეიძლება ითქვას სხვა სვეტებისთვისაც. ინტერნეტში ნავიგაციის დროს, რომელიმე ნაბიჯის შესაბამისი ალბათობის ვექტორი არის ვექტორი, რომლის კომპონენტები მიუთითებს იმ ალბათობას, რომლითაც მომხმარებელი ამ ნაბიჯის შემდეგ მოხვდება ამ ვებ-გვერდზე. მაგალითად, თუ მომხმარებელი ნავიგაციას იწყებს C გვერდიდან, მაშინ დასაწყისში ალბათობის ვექტორი იქნება

$$p^0 = \begin{pmatrix} p(X_0 = A) \\ p(X_0 = B) \\ p(X_0 = C) \\ p(X_0 = D) \\ p(X_0 = E) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

(11)

ალბათობის თვისებების გამოყენებით ადვილი სანახავია, რომ პირველი ნაბიჯის შემდეგ ალბათობის ვექტორი მიიღება P მატრიცის გამრავლებით საწყის ვექტორზე: $p=P \cdot p^0$ და ასე შემდეგ:

$$p^1 = \begin{pmatrix} p(X_1 = A) \\ p(X_1 = B) \\ p(X_1 = C) \\ p(X_1 = D) \\ p(X_1 = E) \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ \frac{1}{3} \end{pmatrix},$$

$$p^2 = P \cdot p_1:$$

$$p^2 = \begin{pmatrix} p(X_2 = A) \\ p(X_2 = B) \\ p(X_2 = C) \\ p(X_2 = D) \\ p(X_2 = E) \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{6} \\ \frac{4}{9} \\ \frac{5}{18} \\ \frac{1}{9} \\ 0 \end{pmatrix}.$$

$$\text{საზოგადოდ: } p^n = P \cdot p^{n-1}.$$

(12)

ამ პროცედურის შინაარსიდან ადვილად ჩანს, რომ ვებ-გვერდების რანჟირება ნიშნავს მათ დალაგებას იმის მიხედვით, თუ რომელ მათგანში მოხვედრის ალბათობა არის უფრო მეტი ინტერნეტში ნავიგაციის პროცესის უსასრულოდ გაგრძელების შედეგად. ე.ი. თუ ვექტორების მიმდევრობისათვის $p^n = P \cdot p^{n-1}$ $n=1,2,\dots$, ვიპოვით „ზღვრულ“ ვექტორს და შემდეგ გვერდებს დავალაგებთ ამ ვექტორის შესაბამისი კომპონენტების ზრდის მიხედვით, ეს მოგვცემს გვერდების დალაგებას მათი რანგების ზრდის მიხედვით. როგორც მარკოვის ჯაჭვების თეორია ამბობს, ეს „ზღვრული“ ვექტორი კი, უმეტეს შემთხვევებში, არის ისეთი ვექტორი, რომელიც არის გადასვლის მატრიცის საკუთარი ვექტორი, რომლის შესაბამისი საკუთარი მნიშვნელობა 1-ის ტოლია („უძრავი წერტილი“) და რომლის კომპონენტების ჯამი 1-ის ტოლია (ალბათობების ჯამი). ჩვენი „მინი ინტერნეტის“ შემთხვევაში ეს ვექტორი ადვილი მოსაძებნია წრფივი ალგებრის ელემენტარული აპარატის გამოყენებით და იგი არის:

$$\pi = \frac{1}{41} \begin{pmatrix} 12 \\ 16 \\ 9 \\ 1 \\ 3 \end{pmatrix}$$

(13)

მიღებული შედეგის მიხედვით შეიძლება ითქვას, რომ საკმაოდ ხანგრძლივი ნავიგაციის შემდეგ მომხმარებელი ყველაზე ხშირად მოხვდება B გვერდზე (ალბათობა ტოლია 16/41-ის). ხოლო გვერდების რანჟირება კი ასეთია: B გვერდის რანგი იქნება 1-ის ტოლი, რადგან მასში მოხვედრის ალბათობა ყველაზე მაღალია; A გვერდის რანგი იქნება 2-ის ტოლი, რადგან მასში მოხვედრის ალბათობა B-ს მერე ყველაზე მაღალია; მას მოსდევს C გვერდი, ხოლო ბოლო ადგილზეა D გვერდი, რადგან მასში მოხვედრის ალბათობა ყველაზე მცირეა[31].

შევნიშნავთ, რომ მართალია აქ აღწერილი პროცედურა წარმოადგენს რეალობაში გამოყენებული პროცედურის საფუძველს, Google იყენებს ამ პროცედურის გაუმჯობესების სხვადასხვა ხერხებს და ინოვაციებს, რომლებიც კომერციულ საიდუმლოებას წარმოადგენს. გარდა ამისა, ჩვენ განვიხილეთ წარმოსახვითი შემთხვევა, როდესაც „მინი ინტერ-ნეტი“ შედგებოდა მხოლოდ 5 გვერდისაგან. ამ შემთხვევაში, ალბათობების ვექტორის მოძებნა საკმაოდ მარტივია და შესაძლებელია მხოლოდ ფურცლისა და კალმის გამოყენებითაც. რეალური ინტერნეტი კი მოიცავს მილიარდობით გვერდს. ე.ი. რანჟირებისათვის საჭიროა ისეთი მატრიცის საკუთარი ვექტორის მოძებნა, რომელიც შეიცავს რამოდენიმე ასეულ მილიარდ ელემენტს. როგორც მკვლევარები აღნიშნავენ, ეს არის მატრიცებთან დაკავშირებული გამოთვლების ამოცანებს შორის ერთ-ერთი ყველაზე რთული ამოცანა. თუ რომელ ხერხს იყენებს Google ამ ამოცანის

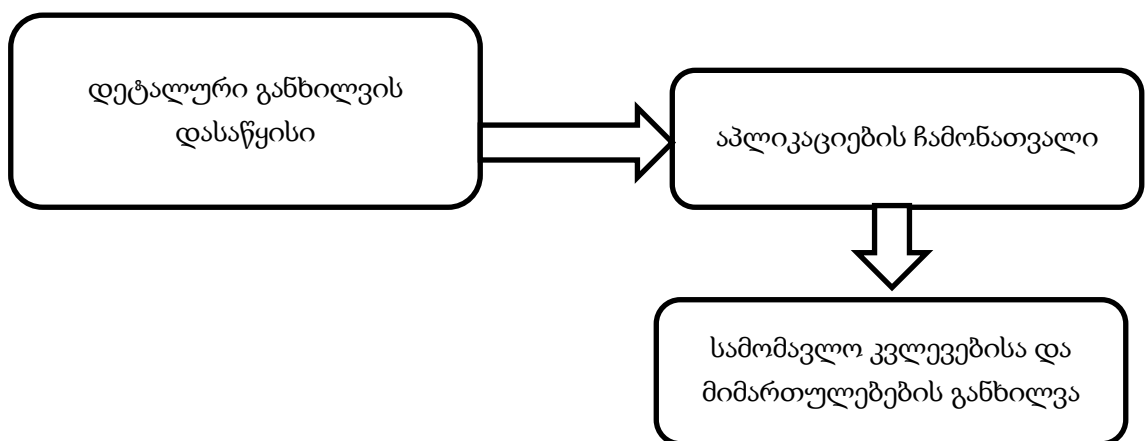
ამოსახსნელად, ასევე კომერციული საიდუმლოა[27].

2.5 კლასიფიკაცია

კლასიფიკაცია მნიშვნელოვან როლს ასრულებს ბევრი ინფორმაციის მოძიებისა და მართვის საქმეში. ეს მნიშვნელოვანია, რადგან მოხდეს ორიენტირებული ძიება და დაეხმაროს ვებ-დირექტორიებს განვითარებაში. მოხდეს თემატური ბმულის ანალიზი და ანალიზის აქტუალური სტრუქტურის ჩამოყალიბება ქსელში. ვებ-გვერდების კლასიფიკაცია ასევე აუმჯობესებს ინტერნეტ ძიების ხარისხს.

ვებ-გვერდების კლასიფიკაციის დროს, როგორც წესი ხდება ვებ-გვერდის გამოყენების სპეციფიკის დეტალური განხილვა. ყურადღება ეთმობა შემდეგ დეტალებს (იხ. ნახაზი 16)

- მოხდეს დეტალური განხილვა თუ როგორ უნდა გამოისახოს სასარგებლო ვებ-გვერდი კლასიფიკაციისას.
- გაკეთდეს სია იმ აპლიკაციების რომლების მოახდენენ კლასიფიკაციას.
- განხილულ იქნას სამომავლო კვლევები და მიმართულებები.



ნახაზი 16 კლასიფიკაციის დეტალები

განვიხილოთ თითოეული მათგანი:

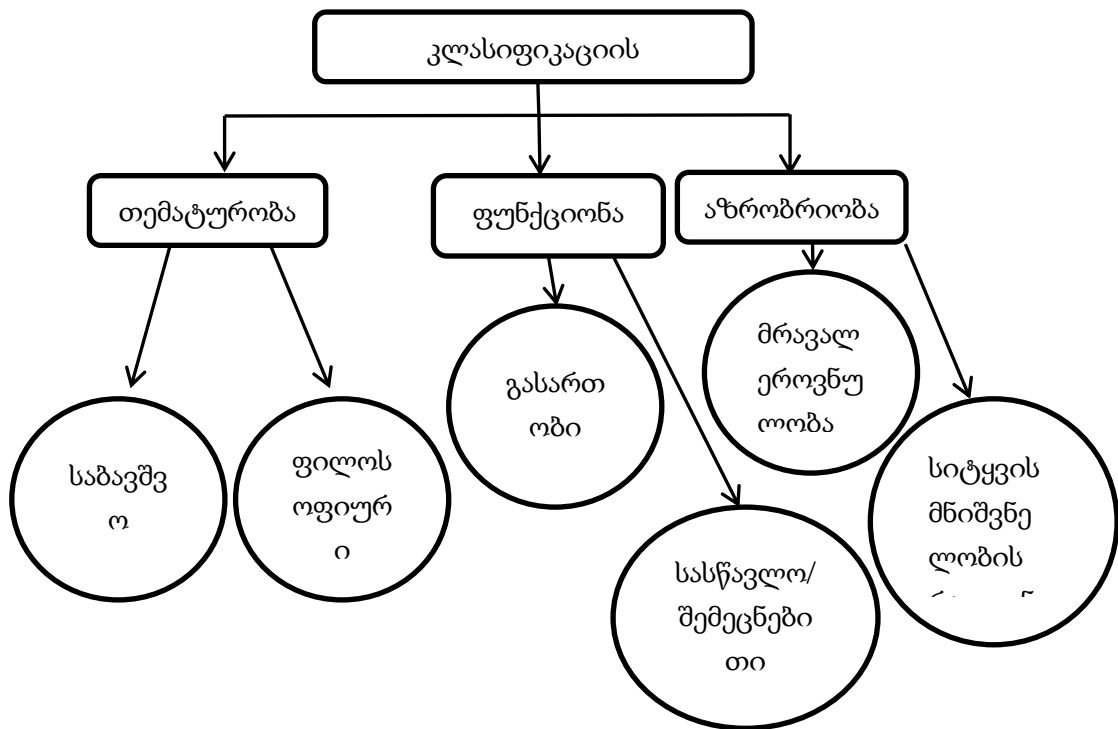
დეტალური განხილვისას ყურადღება ეთმობა იმას, თუ როგორ უნდა გამოისახოს სასარგებლო ვებ-რესურსი. საქმე ისაა რომ, კონკრეტული ძიებისას საძიებო სისტემამ არ იცის, თუ რას ეძებს მომხმარებელი: ფილმს, სურათს თუ ინფორმაციას. ამიტომ ასეთ დროს ხდება ძიების მიღებული შედეგის წარმოდგენა, რაც შეიძლება ოპტიმალური ფორმით.

აპლიკაციები არის ის საშუალებები რომელთაც უნდა მოახდინონ კლასიფიკაცია. ასეთ დროს განისაზღვრება იმ აპლიკაციების ჩამონათვალი, რომლებიც რომლებიც მონაწილეობას მიიღებენ ამ პროცესში ოპტიმალური შედეგის მისაღწევად.

სამომავლო კვლევები და მიმართულებები კი ეფუძნება, ახალი მეთოდოლოგიის შემუშავებას, რომ მოხდეს ძიების პროცესის დახვეწა და საძიებო სისტემები გახდნენ უფრო სწრაფები და შედეგზე ორიენტირებულნი. მთავარ პრობლემას წარმოადგენს ის, რომ ვებ-გვერდის კლასიფიკაციის წარმოიჭრება პრობლემები, როგორიცაა:

- პრობლემა თემატური კლასიფიკაციისას.
- პრობლემა ფუნქციონალური კლასიფიკაციისას.
- პრობლემა აზრობრივი კლასიფიკაციისას .

პრობლემები უფრო დეტალურად წარმოდგენილია ნახაზი 17-ზე



ნახაზი 17 კლასიფიკაციის პრობლემები

მაგალითად: თუ გვინა მოვიძიოთ ინფორმაცია ბინაზე. ცხადია ინტერნეტში იქნება უამრავი სახის ვებ-გვერდი, რომელიც შემოგვთავაზებს ამასთან დაკავშირებით ინფორმაციას. მაგრამ ის ვერ მიხვდება მე რა მჭირდება კონკრეტულად. მაგალითად: შეიძლება მე მჭირდებოდეს ინფორმაცია, პირობითად ბინის მშენებლობის შესახებ, ან ბინის ყიდვა მინდოდეს. ანუ ჩემმა საძიებო სისტემამ მე უნდა გადამიყვანოს კატალოგში ბინების შესახებ, სადაც უკვე წარმოიშობა ახალი პრობლემა ფუნქციონალურობის და აზრობრივი წარმოდგენის. სწორედ ამ პრობლემის გადასაჭრელად შემოთავაზებული იქნა ძიება, კატალოგების დახმარებით [2].

კატალოგები წარმოადგენენ იერარქიულ სტრუქტურას, რაც ხელს უწყობს მომხმარებელს მისთვის სასურველი ინფორმაცია მოიძიოს თემატურად და დალაგებულად. თუმცა, ასეთი სახით შემოთავაზებული ძიება არის გარკვეულწილად არაეფექტური. ვინაიდან მომხმარებელთა

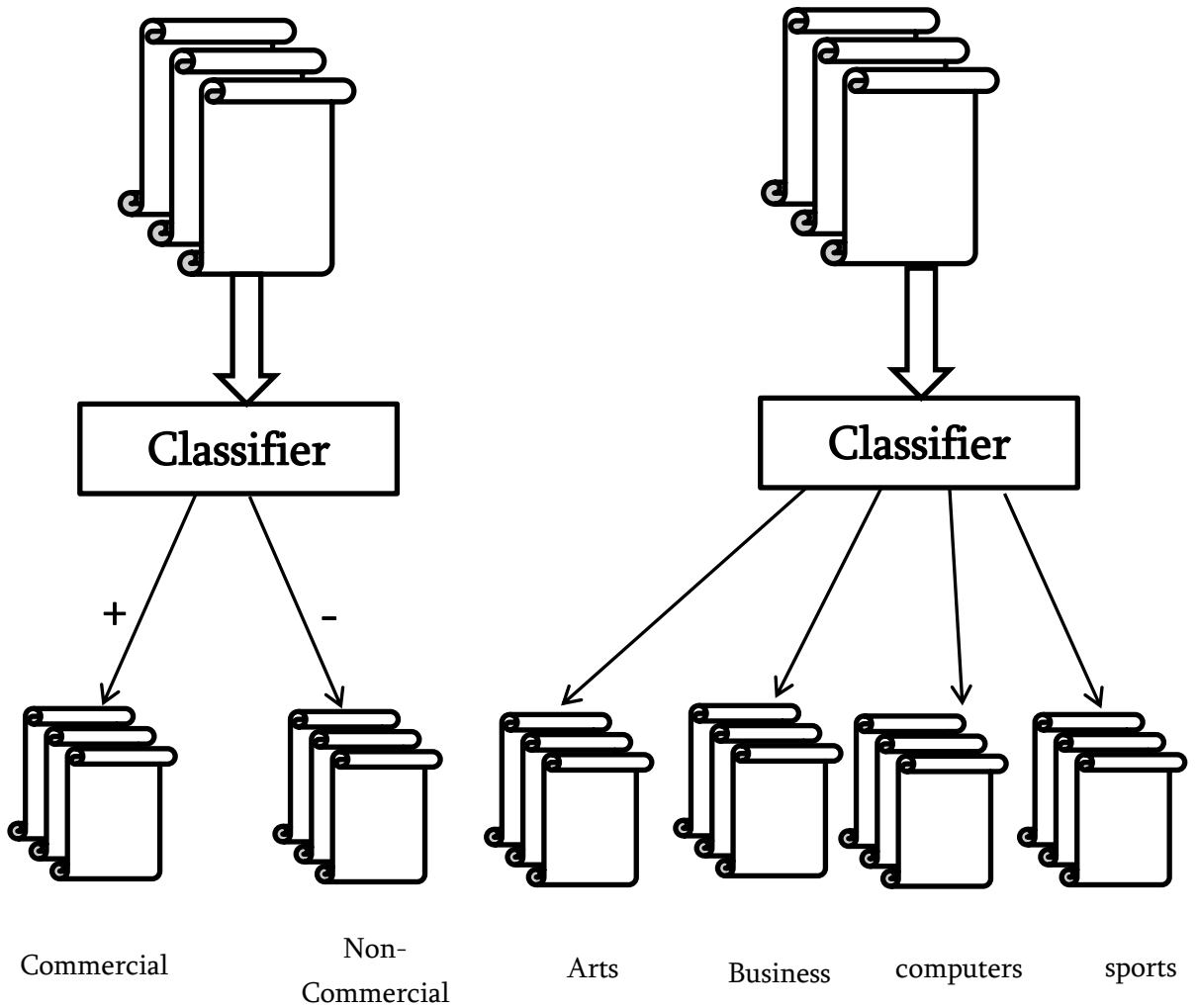
გარკვეულ ნაწილს შეიძლება არც კი ჰქონდეს საწყის ეტაპზე ჩამოყალიბებული თუ რომელი თემატური კატალოგი სჭირდება.

„არ ვიცი შეკითხვის თვალსაზრისი“- აღნიშნული მიდგომა, შემოთავაზებული იქნა 2005 წელს და ის იძლევა დამატებით სარგებელს, მაშინ როდესაც წარმოდგენილი რანჟირებული სიები არ არის დამაკმაყოფილებელი[3]. მიებას უფრო ეფექტურს ხდის „შეკითხვაზე პასუხის გაცემის სისტემა“. ამ დროს ჩამოყალიბებულია მთელი რიგი შეკითხვებისა და გაგზავნილია საძიებო სისტემაში. სისტემა აბრუნებს შედეგს უკან და ახდენს მის კლასიფიცირებას ხე კლასიფიკატორის დახმარებით შემდეგ ოთხ კატეგორიაში:

- გვერდების კრებული:
 - შემადგენელი ელემენტების სია.
- თემატური გვერდები:
 - მოთხოვნის პასუხები.
- რელევანტური გვერდები:
 - დამხმარე პასუხები.
- არარელევანტური გვერდები.

კლასიფიკაციის შემდეგ ხდება იმ თემატური გვერდების დაჯგუფება, რომლებიც პასუხობენ დასმულ კითხვას.

კლასიფიკაცია შეიძლება დაიყოს ორ კლასად. ორობითი კლასიფიკაცია და მულტიფუნქციური კლასიფიკაცია(იხ. ნახაზი 18).



ნახაზი 18 ბინარული და მულტიფუნქციური კლასიფიკაცია

ორობითი კლასიფიკაცია, თავის მხრივ შეიცავს მხოლოდ ორ კატეგორიას:

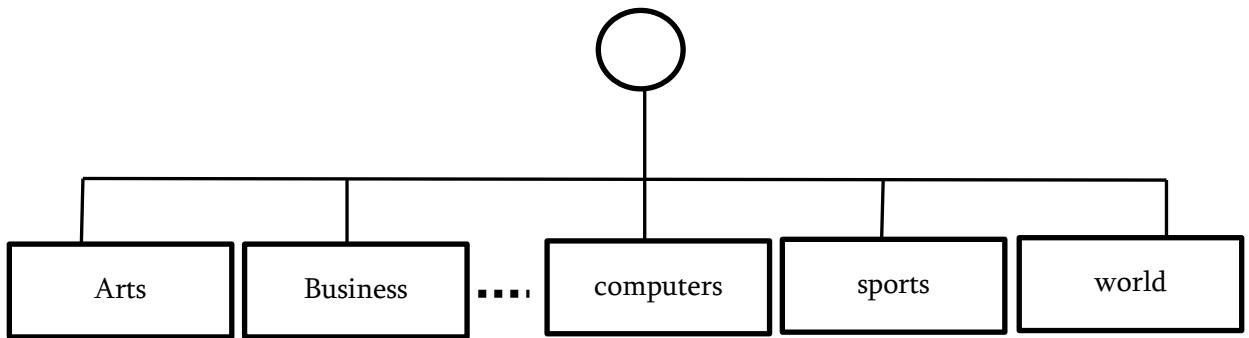
- კომერციული.
- არაკომერციული.

მულტიფუნქციური კლასიფიკაცია, კი შეიცავს მრავალ კატეგორიას მაგალითად:

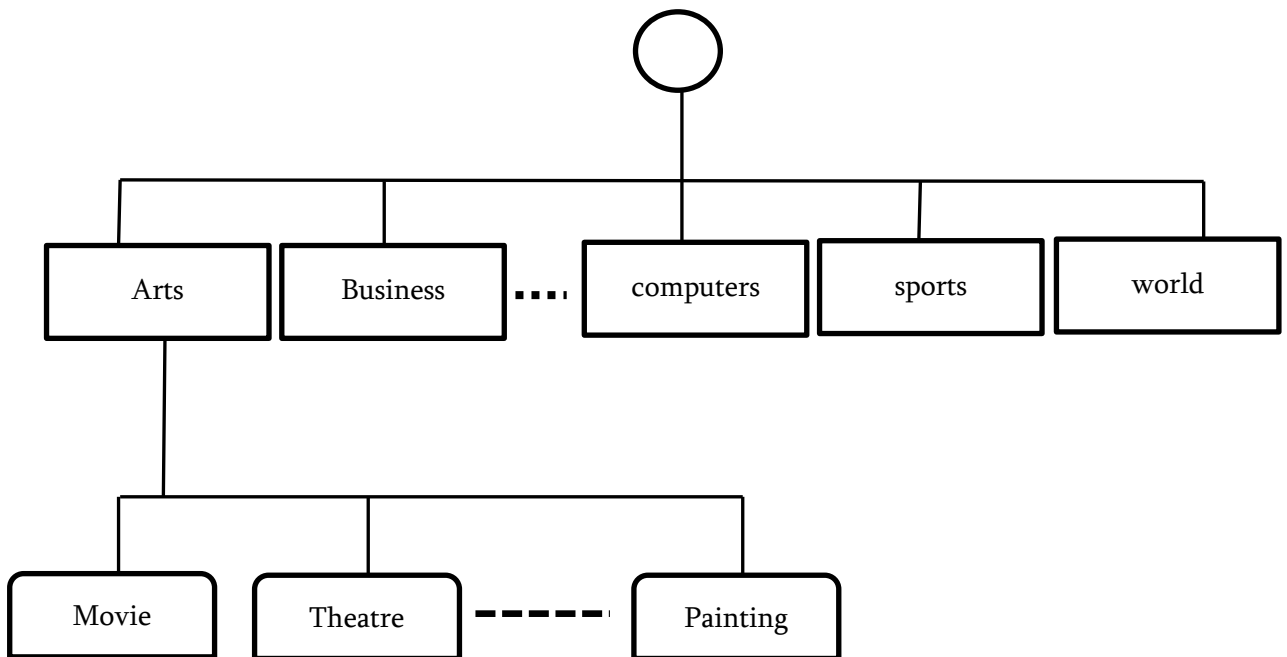
- ხელოვნება.
- სპორტი.
- კომპიუტერები.
- მეცნიერება და ა.შ.

თვით კლასების კლასიფიკაცია. შეიძლება ასევე იყოს. ე.წ. ბრტყელი(იხ ნახ 19) და იერარქიული(იხ. ნახაზი 20).

ბრტყელი კლასიფიკაციისას, კატალოგები დალაგებული არიან პარალელურად, ანუ მათ შორის დამოკიდებულება არ არსებობს. ხოლო იერარქიული კლასიფიკაციისას, ხდება იერარქიული ხის სტრუქტურის ჩამოყალიბება, სადაც თითოეულ კატეგორიას აქვს კავშირი სხვა კატეგორიასთან.



ნახაზი 19 ბრტყელი კლასიფიკაცია



ნახაზი 20. იერარქიული კლასიფიკაცია

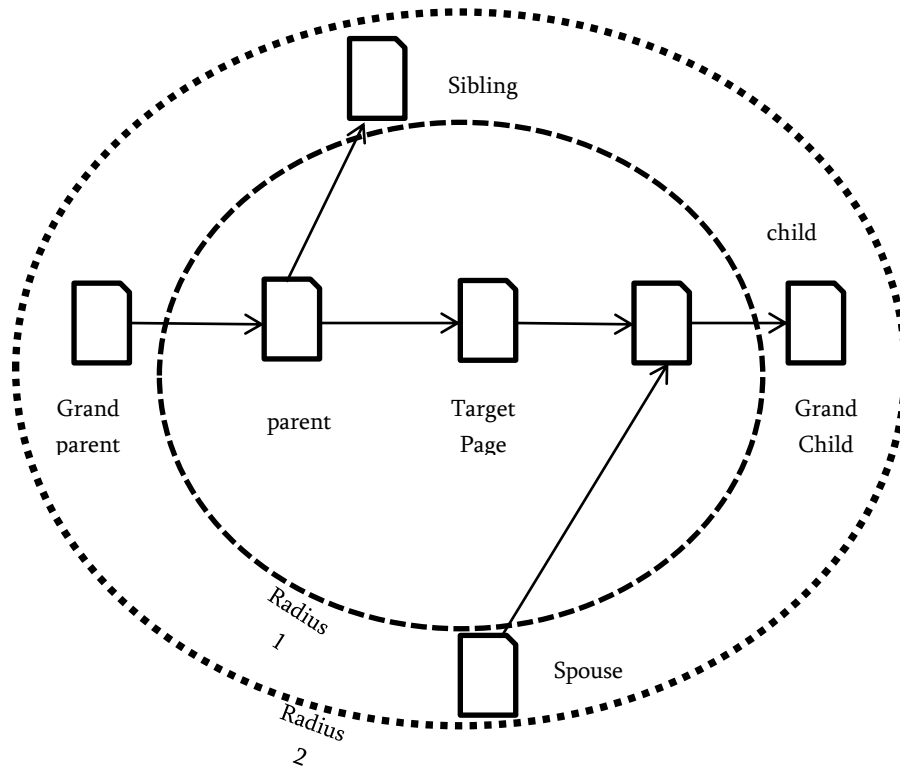
არსებობს უამრავი სახის მიდგომა, რომელიც მხარს უჭერს კლასიფიცირების ეფექტურ შესრულებას.

ერთ-ერთი ასეთი არის მიდგომა:

“დახმარება კითხვაზე პასუხის გაცემისას” ის ეხმარება საძიებო სისტემას ინტერნეტში არსებული საიტების ქაოსური განლაგების კატალოგებად დახარისხებას. ეს არის ერთგვარი ქვესისტემა, რომელსაც აქვს მთელი რიგი კითხვების ჩამონათვალისა და საძიებო სისტემაა ამ კითხვებზე პასუხის გაცემით ახდენს მოცემული საიტის თემატურ კატალოგზე მიკუთვნებას. ხოლო **ვერტიკალური ძიება** გამოიყენება მაშინ, როდესაც კითხვებზე პასუხის გაცემა არ არის ეფექტური და არ იძლევა შედეგს. ის შეიცავს წინასწარ განსაზღვრულ თემებს, ინტერესებს და მიდგომებს და ახდენს ვებ-გვერდის კატალოგზე მიკუთვნებას [2][11][13].

ვიზუალური ანალიზი- ყველა საიტი იწერება HTML ენაზე და იძლევა სხვადასხვა ვიზუალურ წარმოდგენებს. შემოთავაზებულია **“ვიზუალური სიახლოვის გრაფები”**. თითოეული კვანძი წარმოადგენს HTML ობიექტს და ყველა წვერო განსაზღვრავს მის დამოკიდებულებას ვიზუალთან. უმთავრესი მიდგომა კლასიფიკაციისას, უმეტეს შემთხვევაში ითვალისწინებს მხოლოდ ვიზუალს და ყურადღებას არ აქცევს მის კოდს. უნდა აღინიშნოს, რომ ეს მეთოდი ნაკლებ ეფექტურია და აქ მხოლოდ შემოქმედებითობაზეა ყურადღება გადატანილი[2,19,21]

კლასიფიკაცია შეიძლება მოხდეს ვებ-გვერდზე არსებული ბმულების მეშვეობითაც. ამ შემთხვევაში განისაზღვრება ექვსი სახის მეზობელი, რომელთაგანაც შეიძლება კავშირი ჰქონდეს ვებ-გვერდს. პირობითად ესენია: „მშობელი“, „შვილი“, „და“, „ძმა“, „ბებია“, „ზაბუა“(იხ.ნახაზი 21)



ნახაზი 21. „მეზობლები“

ამ შემთხვევაში სამიზნე გვერდს წარმოადგეს მასთან დაკავშირებული გვერდების პრიორიტეტულობა. ანუ ვის აქვს მასთან კავშირი და საიდან მომდინარეობს ის[2].

2.6 ძიების სტრუქტურული მიმოხილვა

ინტერნეტში ძიება ეფუძნება შემდეგი ამოცანების გადაჭრას:

- რესურსების მოძიება.
- ინფორმაციის დახარისხება.
- გენერალიზაცია.
- ანალიზი.

ძიების პროცესი იყოფა სამ ნაწილად ესენია:

- შინაარსის მოპოვება.
- გამოყენების სფერო.
- სტრუქტურის მოპოვება.

- ✓ **შინაარსის მოპოვება-** ეს არის ტექნიკა, რომელიც იყენებს ფაქტიურ ინფორმაციას, რომელიც ინახება ინტერნეტში და ახდენს მისი შაბლონის ძიებას. იგი თავის თავში ახდენს ტექსტის ანალიზს, რომელიც ინახება ფაილებში.
- ✓ **გამოყენების სფერო-აქცენტი** გაკეთებულია იმაზე, თუ როგორ იყენებენ მომხმარებლები ინტერნეტს. ის იყენებს დაშვების ჟურნალს ვებ-სერვერზე, ცდილობს აღმოაჩინოს ნიმუშები, რომლებიც იქმნება მომხმარებელთა მიერ.
- ✓ **სტრუქტურის მოპოვება-** გულისხმობს იპოვოს გზა, რომლის დახმარებითაც ვებ-გვერდები უკავშირდება ერთმანეთს .

მას შემდეგ რაც ყველა შემადგენელი ობიექტი მოიპოვება, შეიძლება გვერდი თავისუფლად მივაკუთნოთ იმ გვერდების სიას რომელსაც გააჩნია ატრიბუტები. უნდა აღინიშნოს, რომ ვებ-კლასიფიკაცია არ არის მხოლოდ კონკრეტული ატრიბუტების ძიება, მას გააჩნია სირთულები და მნიშვნელოვანი განსხვავებები. **პირველ რიგში**, უჩვეულოდ დიდი რაოდენობის სიტყვის მოპოვება შეიძლება. მაგრამ არსებობს ათიათასობით სხვადასხვა სიტყვა, სხვადასხვა ვებ-გვერდზე.

მეორე პრობლემა ის არის, რომ სიტყვებს აქვთ ორაზროვანი გაგება და მანქანისათვის უბრალოდ შეუძლებელია თუ რა მნიშვნელობითაა სიტყვა გამოყენებული, კონკრეტულ სიტუაციაში [4].

თუმცა უნდა აღინიშნოს, რომ HTML კოდი გვაძლევს საშუალებას თავად გვერდის ადმინისტრატორმა განსაზღვროს, თუ რა სიტყვები იქნება მისი ვებ-გვერდისათვის ე.წ. საგასაღებო მნიშვნელობის, რომლის დახმარებითაც მისი გვერდი აღქმული იქნება სამიუბო სისტემის მიერ. HTML ენაზე,ის შემდეგნაირად ჩამოყალიბდება:

`<meta> tags: in <meta NAME="keywords">`

შესაძლებელია საიტის მოკლე აღწერილობის გაკეთებაც:

`<meta NAME="description">`

შეიძლება მოხდეს გვერდის აღწერა მის ბმულშიც. თუმცა უნდა აღინიშნოს, რომ ამ დროს ხდება მხოლოდ მისი მოკლე აღწერილობის გაკეთება

`Description Text`

სხვა ტექტიკაა, როდესაც ვებ-გვერდებს გააჩნიათ უკუკავშირი. ეს ნიშნავს რომ მათ შორის არის შინაარსობრივი განსხვავება[4].

II თავის დასკვნები

1. განხილილია ინტერნეტ-ძიების სტრუქტურა გუგლის საძიებო სისტემის მაგალითზე.
2. საძიებო სისტემის სტრუქტურა წარმოდგენილია გრაფიკული სახით და ახსნილია მისი მოდულები, მათ შორის დამოკიდებულება.
3. წარმოდგენილია ძიების შეზღუდვები და მისი გადაჭრის გზები.
4. განხილილია ინტერნეტ-ძიების ალგორითმები და მათი მუშაობის პრინციპები.
5. წარმოდგენილია ალგორითმები და მათი მუშაობის სტრუქტურა გრაფიკული ფორმით.
6. ჩატარებულია რანჟირების ალგორითმების ამლწერი კვლევა, კერძოდ მარკოვის ჯაჭვის დახმარებით წარმოდგენილია ძიების ოპტიმალური მეთოდების შემუშავების საკითხები.
7. წარმოდგენილია კონკრეტული მაგალითი მინი ინტერნეტის ბაზაზე.
8. ახსნილია მომხმარებლის მიერ კონკრეტულ ვებ-გვერდზე მოხვედრის ალბათობები და მოცემულია მისი გრაფიკული წარმოდგენა.
9. წარმოჩენილია არსებული პრობლემები, ვებ-გვერდების კლასიფიკაციისას.
10. ნაჩვენებია, რომ ძიების პროცესი ემყარება, ვებ-გვერდების ამლწერი სტრუქტურას.
11. ნაჩვენებია, რომ ინტერნეტ-ძიების პროცესი შესაძლოა გადაწყდეს კატალოგების გამოყენებით.
12. ინტერნეტ ძიების მოცემული ალგორითმები და მიდგომები არ იძლევა სასურველ შედეგს.
13. ნაჩვენებია რომ საძიებო კომპანიები აქტიურად მუშაობენ მოთხოვნის აზრის გამგები ალგორითმების შექმნაზე.

ექსპერიმენტული ნაწილი

თავი III

ინტერნეტ ძიების ოპტიმალური მეთოდები

ამ თავში, განხილულია ინტერნეტ ძიების ოპტიმალური მეთოდის რეალიზების საკითხი.

ჩამოყალიბებულია ინტერნეტ-ძიების ეფექტური მეთოდის კონცეფცია და ჩატარებულია მისი მუშაობის პრინციპების ანალიზი. შემუშავებულია მათი პრაქტიკული გამოყენების მეთოდები ინტერნეტ სივრცეში. ნაჩვენებია მისი პრაქტიკული რეალიზების მაგალითი და მისი მუშაობის სტრუქტურა გრაფიკული სახით.

ჩამოყალიბებულია მეთოდების დადებითი და უარყოფითი მხარეები. განხილულია მეთოდის ინტეგრაციის საკითხები ვებ-ბრაუზერებსა და ოპერაციულ სისტემებში. გაანალიზებულია მისაღები შედეგები.

3.1 ინტერნეტ ძიების ხერხები

როგორც ჩვენთვის ცნობილია, არსებობს ინტერნეტში ძიების გარკვეული წესები, რომლებსაც უნდა დავემორჩილოთ, რომ მივიღოთ შესაბამისი შედეგი, მოთხოვნილ რესურსზე. ესენია:

- “+/-“ - მაგალითად: თუ ჩავწერთ ასეთ რამეს „ხელოვნური ინტელექტი+ რობოტები-ფილმი“. ეს ნიშნავს იმას, რომ მოიძებნება რესურსები სადაც აუცილებელად უნდა იყოს „ხელოვნური ინტელექტი“ და „რობოტები“ და არ უნდა იყოს „ფილმი“.
- “|” - მაგალითად: თუ ვეძებთ ინფორმაციას ვებ-გვერდებზე, სადაც გვინდა შეგვხვდეს ერთ-ერთი სიტვა მაინც. მაგალითად: „ხელოვნური ინტელექტი|რობოტი“

- „-“ ზუსტი ფრაზის ძიება. მაგალითად: „ხელოვნური ინტელექტი“. მივიღებთ იმ რესურსებს, სადაც ზუსტად ეს ფრაზა გვხვდება.
- „!“-სიტყვის სხვადასხვა ფორმით ძიება. ეს ნიშნავს იმას, რომ მაგალითად: შეიძლება ერთ ვებ-გვერდზე იყოს სიტყვა „ინტელექტი“, მეორეზე „ინტელექტის“, „ინტელექტმა“ და ა.შ. შესაბამისად, რომ მოხდეს მისი გარჩევა ძიებისას, გამოვიყენებთ შემდეგ ფორმას „!ინტელექტი“ და მივიღებთ ყველა იმ ვებ-რესურსს, სადაც წარმოდგენილი არის სიტყვა „ინტელექტი“ მისი სხვადასხვა ბრუნვით.
- „/n“- როდესაც გვსურს ვიპოვოთ ფრაზის ყველა განმარტება, გამოვიყენებთ სიმბოლო „/“ და ციფრი რომელიც გამოხატავს ფრაზის მნიშვნელობათა რაოდენობას.
- „#url“-მისამართის ძიება, მაგალითად: „#url=www.“
- „#link“- ბმულის ძიება, მაგალითად: „#link= www.file.ge“ აჩვენებს ყველა ვებ-გვერდს, სადაც არის ლინკი მოცემულ ვებ-გვერდზე.

აღნიშნულის გათვალისწინებით ჩავატარეთ სამი ექსპერიმენტი, რომლის შედეგები წარმოდგენილია დანართში(დანართი №1 სამი გვერდი). პირველი ექსპერიმენტისას გუგლის საძებო სისტემაში ვეცადეთ მოგვეპოვებინა ინფორმაცია ხელოვნურ ინტელექტზე. საძებო სისტემაში ჩავწერეთ „ხელოვნური ინტელექტი“. მიღებულ შედეგში, წარმოდგენილ რესურსების სიაში პირველ ადგილს იკავებს ინფორმაცია ვიკიპედიიდან, მეორე და მესამე ადგილზე კი, არის ფილმები მსგავსი დასახელებით. შედარებით მოცულობით ინფორმაციას ჩვენ ვაწყდებით სიაში რიგით მე-5 რესურსზე.

ჩავატარეთ მეორე ექსპერიმენტი, კერძოდ გამოვიყენეთ ე.წ. „ბრჭყალები“.ამ შემთხვევაში მოხდა ცვლილება. სიის სათავეში გადაინაცვლეს იმ ვებ-გვერდებმა, რომლებიც შეიცავენ ფილმებს ხელოვნური ინტელექტის შესახებ და აქ ზუსტად არის ეს ფრაზა ნახსენები. ხოლო, მესამე ადგილზე გადაინაცვლა ვებ-გვერდმა, რომელიც შეიცავს ინფორმაციას და რომელიც პირველი ექსპერიმენტისას სიაში მეხუთე

ადგილს იკავებდა.

მესამე ექსპერიმენტისას გამოვიყენეთ „+/-“ მეთოდი, კერძოდ, რადგან ზემოთ აღწერილ ორ ექსპერიმენტში ჩვენ ვერ მივაღწიეთ იმას, რომ გამოგვერიცხა ძეხვისას ფილმების არსებობა, მოვიქცეთ შემდეგნაირად. გუგლის საძიებო სისტემისათვის ჩამოვაცალიბოთ შემდეგი სახის მოთხოვნა „ხელოვნური ინტელექტი+რობოტი“. მოხდა ნაწილობრივ დალაგება და ჩვენ სიის პირველ პოზიციებზე მივიღეთ ინფორმაციის შემცველი ვებ-რესურსები.

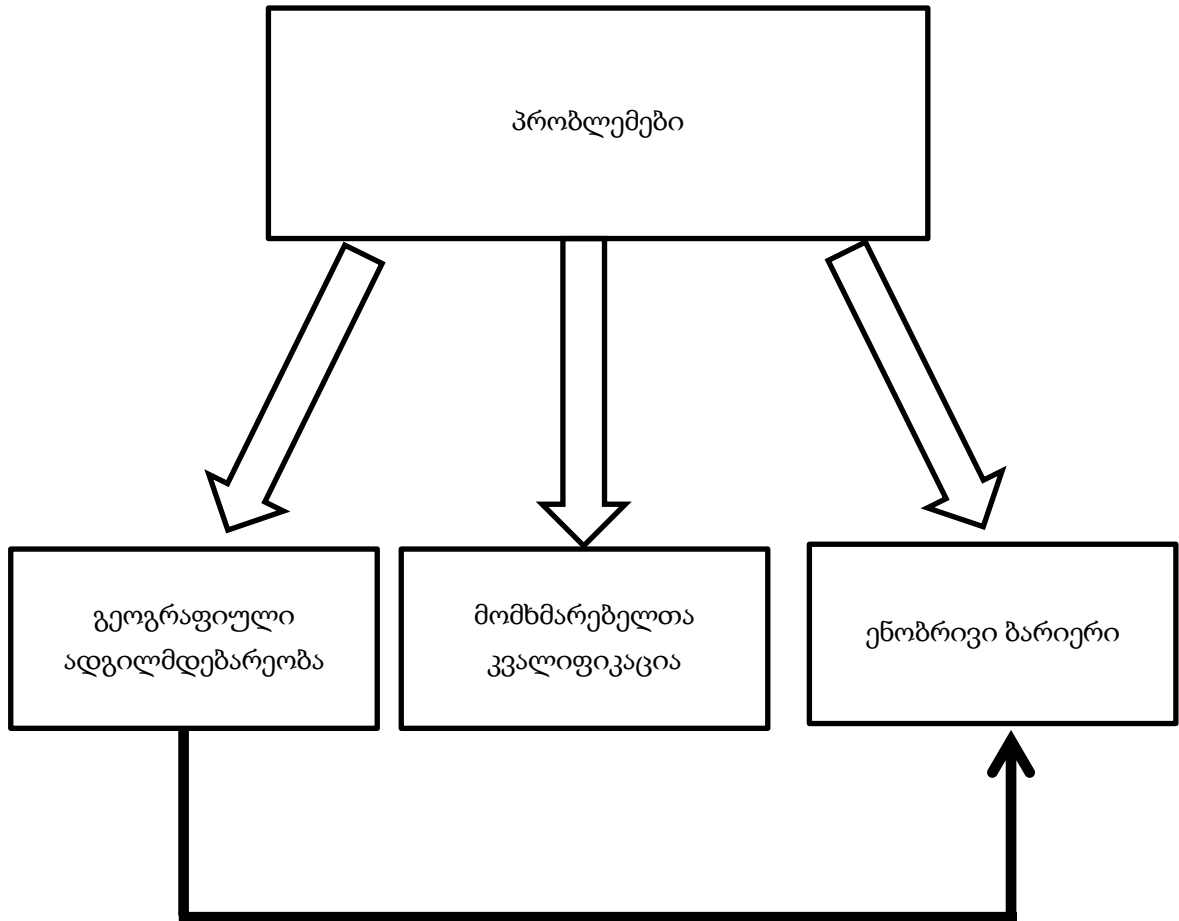
3.2 პრობლემის გადაწყვეტის გზები

რადგან მსოფლიოში არსებობს სხვადასხვა ენა და დამწერლობა, არსებობს ინფორმაციის აღქმის სხვადასხვაობა. გერმანელი როგორც ეძებს ინფორმაციას, ისე არ მოძებნის ქართველი და ქართველი როგორც ეძებს, ისე არ მოძებნის ინგლისელი და ა.შ. არსებობს სხვადასხვა სახის სირთულეები, რომელიც დღემდე ვერ წყდება და შეუძლებელი ხდება ძიების პროცესის დახვეწა. საძიებო სისტემები ამ პრობლემებს გარკვეულწილად წყვეტენ ესენია:

- ✓ გეოგრაფიული ადგილდებარეობა
- ✓ მომხმარებლის კვალიფიკაცია

ხოლო პრობლემა, რომელიც ჯერ-ჯერობით ყველაზე დიდ თავსატეხს უჩენს საძიებო სისტემებს არის:

- ენობრივი ბარიერი



ნახაზი 22 პრობლემები

განვიხილოთ დეტალურად თუ რას ნიშნავს ყოველივე ზემოთთქული:

გეოგრაფიული ადგილმდებარეობა- სამიეზო სისტემა არჩევს, თუ რომელ რეგიონში იმყოფება კონკტერულად მომხარებელი და ახდეს შესაბამის სერვერებზე ძიებას. კერძოდ მე, რომ ვარ ქართველი, პირველ რიგში, ჩემს მოთხოვნაზე ძიება განხორციელდება ქართულ სერვერებზე.

მომხმარებლის კვალიფიკაცია- როგორც ზემოთ ავღნიშნეთ, მომხმარებლთა უმრავლესობამ არ იცის ინფორმაციის მოძიების ხერხების შესახებ. შესაბამისად, სამიეზო სისტემები წყვეტენ ამ პრობლემას შეკითხვის დასმის სისტემის არსებობით : „ხომ არ გულისხმობდით“.

ენობრივი ბარიერი- ის წარმოადგენს ყველაზე რთულ და გადაუჭრელ პრობლემას. *ძნელია სამიეზო სისტემამ გაარჩიოს ყველა დამწერლობა რაც კი არსებობს და მიხვდეს მის აზრს, თუ რა სურს მომხარებელს, რადგან,*

საძიებელ სიტყვას შეიძლება ჰქონდეს სხვადასხვა აზრობრივი მნიშვნელობა. მივდივართ ხელოვნური ინტელექტის შექმნის აუცილებლობამდე, რაც დღეისათვის საკმაოდ პრიმიტიულ დონეზეა. უნდა აღინიშნოს ის, რომ საძიებო კომპანიები უკვე მუშაობენ ამ მიმართულებით. აღსანიშნავია ის ფაქტიც, რომ ენობრივი ბარიერი წამოიჭრება გეოგრაფიული ადგილმდებარეობის შესაბამისად. ცხადია არსებობს მრავალეროვანი ქვეყნები, სადაც ვერ ვიტყვით, რომ ერთი ენაა ძირითადი. მაგრამ, ეს გავლენას ახდენს ძიების პროცესზე, როდესაც, მაგალითად: ქართველი იმყოფება ჰონგკონგში და სურც მოიძიოს ინფორმაცია: „ქართულ პური“. აქ უკვე იჩენს ორი ხელისშემშლელი ფაქტორი თავს,

- პირველი გეოგრაფიულად ის იმყოფება სხვა ქვეყანაში და ეძებს ინფორმაციას სხვა ქვეყნის კულტურის შესახებ.
- მეორე საძიებო სისტემასთან დიალოგის ენა. ძნელი წარმოსადგენია რა ენაზე მოითხოვს მომხმარებელი რესურსს. თუ მოითხოვს ქართულად ის მიიღებს ნაკლებ რესურსს, ისევე როგორც ადგილობრივ ენაზე მოთხოვნის შემთხვევაში.

ამრიგად ყველაზე გადაუჭრელ პობლემას საძიებო სისტემისათვის მოთხოვნის აზრის გაუგებრობა წარმოადგენს და მისი გადაჭრის გზები ხელოვნური ინტელექტის დონეზე დადის.

საძიებო სისტემები საკუთარი ძიების ალგორითმის ეფექტურობის ამაღლებაზე უწყვეტად მუშაობენ. უდიდესი საძიებო კომპანია გუგლი ამუშავებს Hummingbird ალგორითმს საიტის რომელიც წარმოადგენს pagerank-ის განახლებას და მისი უმთავრესი დანიშნულებაა მოახდინოს ადამიანის აზრის გაგება, მაგალითად: კომპანია Yandex იყენებს ხელოვნური ინტელექტის მქონე საძიებო მანქანას: „MatrixNet“, რომელიც არის სწავლებადი და ანალიზს უკეთებს სიტყვებს.

კომპანია Microsoft-ის საძიებო სისტემა Bing იყენებს ადამიან მფილტრავეს, რომლებიც მუშაობენ საათობრივი ანაზღაურებით და ახდენენ დღის, კვირის, თვის განმავლობაში მომხმარებელთა მიერ

ჩაწერილი სიტყვების გაანალიზებას

ის რომ ვებ-გვერდს გააჩნია ბევრი მომმართველი ბმული სულაც არ ნიშნავს იმას, რომ ეს საიტი არის სასარგებლო, ვინაიდან, ის ვერ ერკვევა ძიების შინაარსში და მის მიერ მოთხოვნის შესაბამისად ინდექსირებული გვერდები შეიძლება სულაც არ იყვნენ სასარგებლო.

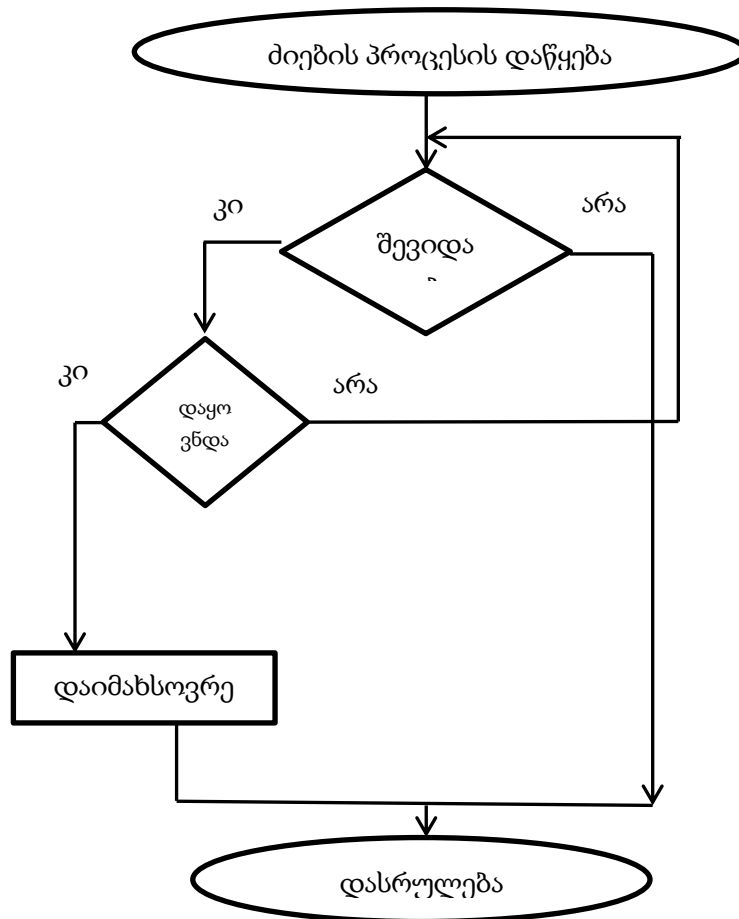
ჩვენი მეთოდი მდგომარეობს იმაში, რომ მოხდეს საძიებო სისტემის იმდაგვარად ოპტიმიზაცია, რომ მიიღწეს ძიების მაღალ ეფექტურობა, კერძოდ შემუშავებული იქნას ალგორითმი, რომელიც დათვლის მომხმარებლის მიერ გამოყენებული ვებ-გვერდების სიხშირებს და განათავსებს მათ მისამართებს საკუთარ მონაცემთა ბაზაში.

მეთოდის არსი მდგომარეობს შემდეგში: ალგორითმი მოახდენს მომხმარებლის მიერ გამოყენებული ვებ-გვერდების ფილტრაციას, იმდაგვარად, რომ კონკრეტულ შემთხვევაში მომხმარებელს წარედგინება მის მიერ გამოყენებული ვებ-გვერდების რეიტინგული სია. სიის სათავეში კი წარმოდგენილი იქნება ის ვებ-გვერდები რომელიც მან ყველაზე მეტი სიხშირით გამოიყენა.

3.3 ალგორითმის მუშაობის არსი

კონკრეტული ძიების პროცესში ალგორითმი მოახდენს იმ ვებ-გვერდის მისამართის ჩანიშვნას, რომელზე გადასვლის შემდეგაც დაფიქსირდება შეყოვნება გარკვეული დროით. ანუ, შეიქმნება ვებ-გვერდების მონაცემთა ბაზა, სადაც კონკრეტული მოთხოვნის შესაბამისად, მოხდება მასზე პასუხის გამცემი ვებ-გვერდების მისამართების ჩანიშვნა.

ალგორითმის მუშაობის სქემა მოცემულია ნახაზზე, რომელიც აღწერს მისი მუშაობის არსს (იხ. ნახაზი 23).



ნახაზი 23 ალგორითმის მუშაობის არსი

უკვე შესაძლებელია მისი ალგორითმის ჩამოყალიბება ადამიანისათვის გასაგებ ენაზე:

1. ძიების დაწყება.
2. აირჩევა სასურველ ვებ-რესურსი. მის შესაბამის ბმულზე გადასვლით.
3. თუ გადასულ ბმულზე ხდება დაყოვნება 2 წუთზე მეტი დროით.
4. მაშინ მოხდება ვებ-რესურსის ჩანიშვნა.
5. თუ არა. ვუბრუნდებით მე-2 პუნქტს.
6. დასასრული.

მაგალითად: მომხმარებელი ეძებს ინფორმაციას მხატვრობის შესახებ. ალგორითმი მოახდენს „მხატვრობა“ ბაზის შექმნას და მას დაექვემდებარება ის მონაცემთა ბაზა, სადაც იქნება იმ ვებ-გვერდების მისამართები, რომელზეც მომხმარებელი შეყოვნდა ძიების პროცესში.

თუ, გარკვეული პერიოდის შემდეგ, მომხმარებელი ისევ ეცდება

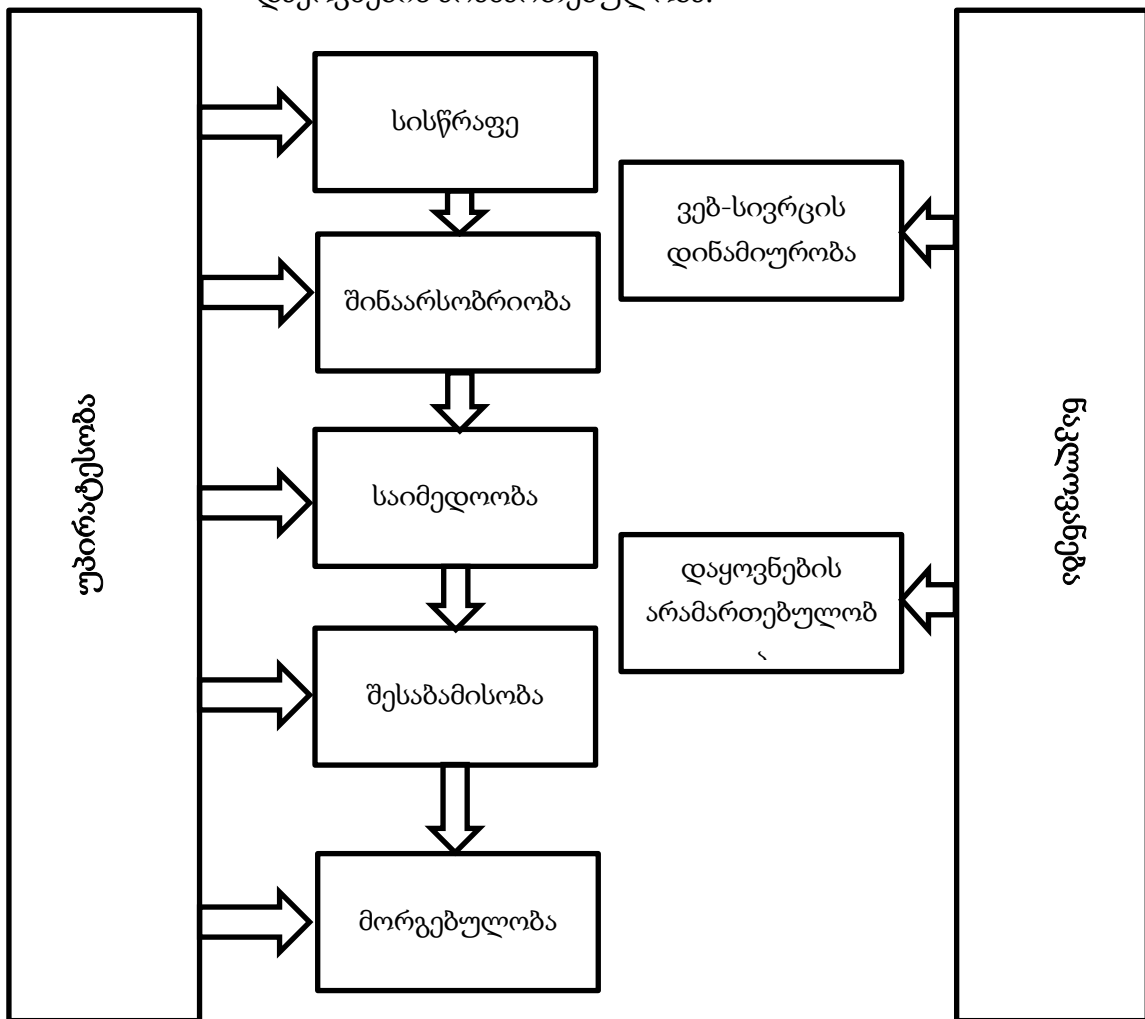
მოიძიოს აღნიშნულზე ინფორმაცია. მას ალგორითმი სიის სათავეში პირველ რიგში წარუდგენს მის ბაზაში არსებულ ვებ-გვერდებს.

შემუშავებული მეთოდის უპირატესობაა:

- ძიების სისწრაფე.
- შინაარსობრივი ძიება.
- საიმედოობა.
- შესაბამისობა.
- მომხმარებელზე მორგებულობა.

ნაკლოვანება:

- ვებ-სივრცის დინამიურობა.
- დაყოვნების არამართებულობა.



ნახაზი 24 მეთოდის უპირატესობა

ძიების სისწრაფეში იგულისხმება, ის რომ მომხმარებელი მიიღებს დალაგებულ ინდექსს მაქსიმალურად სწრაფად და რაც მთავარია, იქ წარმოდგენილი გვერდები შინაარსობრივად შესაბამისი იქნება მისი მოთხოვნისა. ყოველივე ეს ზრდის ძიების საიმედოობას, როდესაც სიის სათავეში წარმოდგენილია რელევანტური გვერდების გარკვეული უმრავლესობა.

ვებ-სივრცის დინამიურობა შეიძლება დიდ პრობლემას წარმოადგენდეს ალგორითმისათვის, თუ არ მოხდება მისი მუდმივი განახლება. ანუ გვექნება მოძველებული მონაცემთა ბაზა, თუ მასში არ შევიტან იმ ახალ მისამართებს, რომლებიც, დროში უწყვეტად იქმნებიან.

რაც შეეხება დაყოვნების არამართებულობას, აქიგულისხმება ის, რომ მომხმარებელმა შეიძლება დააჭიროს ბმულს და სულაც არ შეხედოს ასახულ ინფორმაციას, ისე დატოვოს კომპიუტერი(მაგალითად: სასწრაფოდ მოუწია წასვლამ), ამ შემთხვევაში, ალგორითმი ასახული ვებ-გვერდის მისამართს დაიმახსოვრებს შესაბამის ბაზაში, რაც დაარღვევს ალგორითმის მუშაობის წესს. პრობლემის გადასაჭრელად შეიძლება შემოღებული იქნეს კოეფიციენტი, რომელიც იქნება გარკვეული მუდმივა. ისევე როგორც pagerank-ის შემთხვევაში გვაქვს ე.წ. „ბიძგის ფაქტორი“, ანუ სერფერი შემთხვევით აჭერს ბმულზე. მესამე, ნაკლებ მნიშვნელოვანი პრობლემა, შეიძლება წარმოშვას მომხმარებლის კომპიუტერის არაკომპეტენტურმა გამოყენებამ. მაგალითად: მამის კომპიუტერით ისარგებლა მისი 6 წლის შვილმა. აღნიშნული, მარტივად გადაიჭრება ავტორიზაციის გამოყენებით კომპიუტერზე. აღნიშნული მეთოდი, ეფუძნება, დინამური „სწავლის“ პროცესს და ის მნიშვნელოვნად არის დამოკიდებული კომპეტენტურ ძიების პროცესზე. უნდა აღინიშნოს ის, რომ ამ მხრივ მიმდინარეობს მუშაობა და აღნიშნულის პრაქტიკული რეალიზება მალე იხილავს დღის სინათლეს.

3.4 დანერგვა/ინტეგრაცია

ალგორითმის რეალიზაცია შესაძლებელია სხვადასხვანაირად, კერძოდ შესაძლებელია მოხდეს მისი ინტეგრაცია ინტერნეტ ბრაუზერებში, ან გაკეთდეს სულაც ახალი, რომელშიც ჩაშენებული იქნება ძიების ეს ალგორითმი და ის იქნება ხელმისაწვდომი ნებისმიერ ოპერაციულ სისტემაზე. ისეთი ინტერნეტ ბრაუზერები, როგორცაა Mozilla firefox, Google chrome, იძლევიან საშუალებას, მოხდეს მათ აპლიკაციაში ე.წ. აპლეტის ჩაშენება, რაც აიოლებს საქმეს თუ კი გადავწყვეტთ, რომ ალგორითმი ჩაშენებულ იქნას ინტერნეტ ბრაუზერში.

Web APIs(application programming interface) არის ის საშუალება რომელიც უზრუნველყოფს, ბრაუზერის გაფართოებადობას[.].

- სტანდარტული javascript და DOM(document object model)გამოიყენება ჩვეულებრივი ვებ აპლიკაციებისათვის.
- XMLHttpRequest გამოიყენება მონაცემთა ტრანსფერისათვის სხვადასხვა სერვერებიდან.
- HTML5 არის ენა რომელსაც მხარს უჭერ Google Chrome.

3.5 კონცეფციის პრაქტიკული რეალიზაცია.

Windows გარემოში სამუშაოდ, windows აპლიკაციების შესაქმნელად გამოიყენება პროგრამული გარემო Microsoft visual studio და ობიექტ ორიენტირებული ენა მაგალითად: ეს შეიძლება იყოს C++ ან C#, რომლის დახმარებითაც შესაძლებელია ზემოაღნიშნული ვებ-ბრაუზერის გაკეთება. ჩვენს შემთხვევაში, აღნიშნული აპლიკაცია შექმნილია C# დაპროგრამების ენაზე და მის გრაფიკული სახე წარმოდგენილია დანართი #2-ში(1 გვერდი).

აპლიკაციაზე მიერთებულია შესაბამისი მონაცემთა ბაზა, რომელიც აწყობილია Microsoft access-ში და ახდენს იმ ვებ-რესურსების დამახსოვრებას, რომელიც გამოყენებულ იქნება ძიებისას.

მონაცემთა ბაზა შედგება შემდეგი ველებისაგან:

1. ID
2. name
3. intensity

აპლიკაცია შეიცავს ფარულ „აგენტს“, რომელიც ახდენს თითოეული გვერდის, დამახსოვრებას, რომელიც გამოძახებული იქნება ძიების პროცესში. უნდა აღინიშნოს ის ფაქტი, რომ ბმულზე გადასვლისთანავე ირთება დროის აღმწერელი, რომელიც აითვლის დროს, თუ რა ხნის მანძილზე იმყოფებოდა მომხმარებელი შესაბამის ვებ-გვერდზე და თუ ეს დრო იქნება მეტი პროგრამულად განსაზღვრულ კოეფიციენტზე, მოხდება მისი დამახსოვრება შესაბამის მონაცემთა ბაზაში.

შედეგად მივიღებთ იმას, რომ ძიებისას მომხმარებელი მიიღებს იმ რესურსთა ჩამონთვალს, რომელიც გამოყენებულ იქნა მის მიერ ამათუ იმ ინფორმაციის ძიების პროცესში.

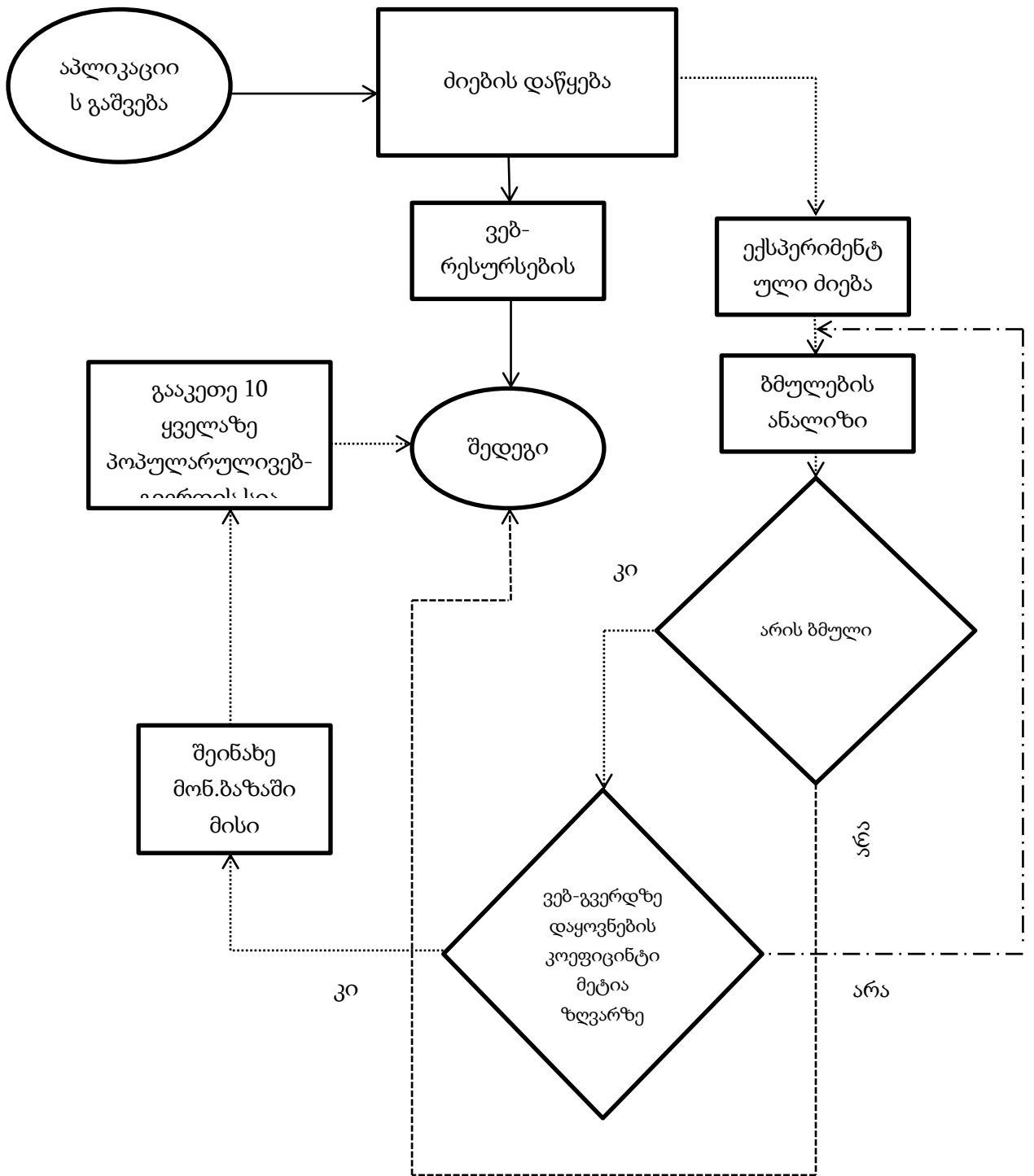
დამეთანხმებით ეს დინამიური პროცესია და დამოკიდებულია აღნიშნული მეთოდის გამოყენების სიხშირეზე. ცხადია მომხმარებელი რაც უფრო ხშირად ისარგებლებს აღნიშნული მეთოდით მით უკეთეს შედეგებს მიიღებს ძიების პროცესში.

3.6 დეტალური აღწერა

1. Windows ფორმის ჩატვირთვაზე იქმენა web ობიექტი, ამის შემდეგ ხდება მისი შევსება იმ მისამართის შესაბამისი ვებ-გვერდით, რომელიც ჩაიწერება სამისამართო გამყოფილებაში.
2. ვებ-გვერდების მისამართების დამახსოვრება იწყება მაშინ, როდესაც სამისამართო ველში მოხდება ერთ-ერთი შემდეგი საძებო სისტემის გამოძახება. იქნება ეს Google, Yandex, Bind ან Yahoo.

3. იმ შემთხვევაში, თუ მოხდა ვებ-ბმულზე განმეორებით გადასვლა, მისი მნიშვნელობა შესაბამის მონაცემთა ბაზაში გაიზრდება ერთით.
4. იმისათვის, რომ მონაცემთა ბაზა დავაკავშიროთ აპლიკაციასთან აუცილებელია მაკავშირებელი ობიექტის შექმნა და შესაბამისი დრაივერის მითითება.
5. ძიებისას ეშვება მოთხოვნა, რომელიც უზრუნველყოფს ბაზაში არსებული 10 ყველაზე მეტი მნიშვნელობის(intensivity) ვებ-ბმულის ამოტანას. და ეს სია ამოიტანება ექსპერტული ძიების კლავიშის გააქტიურებით.
6. უნდა აღინიშნოს, რომ ეს პროცესი დინამიური და ის დამოკიდებულია გამოყენების ინტენსივობაზე. აღნიშნული აპლიკაცია მუშავდება და ვფიქრობთ ძალიან მალე იხილავს დღის სინათლეს. მანამდე კი ჩამოვაცალიბოთ მისი სტრუქტურა გრაფიკული სახით. იხილეთ ნახაზი 3.4.

აპლიკაციის ფსევდოკოდი მოცემულია დანართი #3-ში(6 გვერდი).



ნახაზი 25 კონცეფციის გრაფიკული აღწერილობა

III თავის დასკვნები

1. წარმოდგენილია ინტერნეტ-ძიებისას არსებული გარკვეული წესები.
2. არსებულ წესებზე დაყრდნობით ჩატარებულია ექსპერიმენტი, რომელიც ასახავს საძებო სისტემის ალგორითმის მუშაობის შედეგს.
3. ექსპერიმენტმა აჩვენა, რომ არ მიიღწევა ძიების მაღალ ეფექტურობა.
4. გაანალიზებულია პრობლემები, რაც იწვევს ძიების შედეგის არაეფექტურობას.
5. წარმოდგენილია ინტერნეტ ძიების ახალი მეთოდის კონცეფცია.
6. განხილულია, შემუშავებული კონცეფციის ალგორითმის ინტერნეტ ბრაუზერებსა და ოპერაციულ სისტემებში ინტეგრირების საკითხები. ასევე გაანალიზებულია მათი მუშაობის პრინციპები. დასაბუთებულია მისი უპირატესობები.
7. წარმოდგენილია მისი ალგორითმის მუშაობის არსი გრაფიკული სახით.
8. წარმოდგენილია შემუშავებული მეთოდის პრაქტიკული რეალიზების მაგალითი გრაფიკულად და ახსნილია მისი ალგორითმის მუშაობის პრინციპები.
9. შემუშავებული ალგორითმი არის დაფუძნებული სტანდარტულ კომპონენტებზე და ღია კოდის უტილიტებზე.
10. აღნიშნული მეთოდი უზრუნველყოფს ძიების ეფექტურობის ამაღლებას.

დისერტაციის საბოლოო დასკვნები და რეკომენდაციები

სადისერტაციო ნაშრომში წარმოდგენილი შედეგებიდან გამომდინარე შეიძლება გაკეთდეს შემდეგი დასკვნები

1. თანამედროვე ინტერნეტ-ძიების დროს წამოჭრილი პრობლემების გადასაჭრელად მიძღვნილი ლიტერატურის მიმოხილვა და ანალიზი გვიჩვენებს, რომ ძიების ალგორითმები ირჩევენ ე.წ. გვერდის რანჟირების მეთოდს მისი ინდექსირებისათვის, იყენებენ კატალოგებისა და შეკითხვის დასმის სისტემებს.
2. განხილულია სამიეზო სისტემის ოპტიმიზაციის საკითხები. სადაც აღწერილია Seo-ს ოპტიმიზაციის მეთოდები, არსებობს ოპტიმიზაციის სამი სხვადასხვა მეთოდი: თეთრი ოპტიმიზაცია, შავი ოპტიმიზაცია და ნაცრისფერი ოპტიმიზაცია.
3. ჩატარებულია კლასიკური ძიების ალგორითმების დეტალური ანალიზი, რომელიც გვიჩვენებს, რომ სამიეზო სისტემები ვებ-რესურსების საიმედოობას განსაზღვრავენ ციფრული კალკულაციის მეთოდით.
4. ნაჩვენებია ის პრობლემები რაც წამოიჭრება ძიებისას. კერძოდ ის, რომ ვერ ხერხდება ინფორმაციის აზრობრივი მოძიება, სამიეზო სისტემა ვერ გებულობს თუ რა სურს მომხმარებელს. ძიება მიმდინარეობს მხოლოდ სინტაქსურ დონეზე, ეს გულისხმობს სამიეზო სისტემისათვის მიწოდებული სამიეზო სიტყვის ზუსტი ასლის ძიებას ინტერნეტში.
5. გაანალიზებულია Pagerank-ძიების ალგორითმის მუშაობის პრინციპები, ის ითვალისწინებს გვერდის მნიშვნელობის შეფასებას მასზე მომმართველი ბმულების რაოდენობისა და „წონის“ მიხედვით.
6. განხილულია ინტერნეტ ძიების კონკრეტული ამოცანა რანჟირების ალგორითმების შეფასებისათვის ე.წ. მარკოვის ჯაჭვის გამოყენებით, რომელიც გვიჩვენებს, რომ ვებ-გვერდების რანჟირება ნიშნავს მათ

დალაგებას იმის მიხედვით, თუ რომელ მათგანში მოხვედრის ალბათობა არის უფრო მეტი ინტერნეტში ნავიგაციის ში

7. ნაჩვენებია კლასიფიკაციის მეთოდები, რომლებიც ხელს უწყობენ ძიების პროცესის ამაღლებას. კლასიფიკაციისას ხდება დეტალური განხილვა თუ როგორ უნდა გამოისახოს სასარგებლო ვებ-გვერდი და კეთდება იმ აპლიკაციების სია რომლებიც მონაწილეობას მიიღებენ ამ პროცესში.
8. ჩატარებულია ექსპერიმენტი, რომელიც ძიების შედეგს წარადგენს საძიებო სისტემის მიერ ამ დროისათვის არსებული ალგორითმით.
9. ნაჩვენებია, რომ აღნიშნული ძიების შედეგი არ არის მაქსიმალურად სასარგებლო. საძიებო სიტყვის სხვადასხვაგვარი ფორმით ჩაწერისას მიიღება შედეგების სხვადასხვანაირად დალაგებული სია
10. შემუშავებულია კონცეფცია, რომელსაც საფუძვლად უდევს, მომხმარებლის მიერ გამოყენებული ვებ-გვერდების სიხშირეთა დაანაგარიშება, ძიების პროცესში ალგორითმი მომხმარებელს სთავაზობს მის მიერ ყველაზე ხშირად გამოყენებადი ვებ-გვერდების თემატურად დალაგებულ ინდექსირებულ სიას.
11. შემუშავებული ალგორითმი იყენებს მომხმარებლის ინდივიდუალურ მიდგომას საძიებო სისტემის მიმართ. მის მიერ მოძიებული ინფორმაციის საფუძველზე ქმნის ძიების ინდექსს და ეფუძნება დინამიურ „სწავლებას“
12. წარმოდგენილი ალგორითმი აუმჯობესებს ინტერნეტ-ძიების პროცესს, მომხმარებლის მიერ გამოყენებული ვებ-გვერდების სიხშირული კონტროლით.
13. განხილულია ალგორითმის შესაბამისი პროგრამული უზრუნველყოფის შექმნის საკითხები. შესაძლებელია აპლიკაციის შექმნა, რომელიც ინტეგრირდება ინტერნეტ ბრაუზერებში, ან ცალკეული ვებ-ბრაუზერის შექმნა, რომელიც დაფუძნებული იქნება

ე.წ. ღია პლათმორმულობის პრინციპზე და ის მორგებადი იქნება ნებისმიერ ოპერაციულ სისტემაზე.

14. მოცემულია ალგორითმის პრაქტიკული რეალიზების მაგალითი. რომელიც შექმნის პროცესშია და წარმოადგენს ჩამოყალიბებული კონცეფციის ხორცშესხმას.
15. მოცემულია კონცეფციის მუშაობის სქემა, რომელიც გრაფიკულად ზუსტად გამოსახავს ალგორითმის მუშაობის პრინციპებს.
16. წარმოდგენილია კონცეფციის დადებითი და უარყოფითი მხარეები.
17. კონცეფციის უარყოფითი მხარეები დღეისათვის წარმოადგენს გადაუჭრელ პრობლემას და მის მოგვარებას მივყავართ ხელოვნური ინტელექტის შექმნის საკითხებზე.
18. კონცეფციის დადებითი მხარეები სრულად პასუხობს, ძიების ეფექტურობის ამაღლების საკითხებს და ის უზრუნველყოფს ძიების მაღალ ეფექტურობას.

ლიტერატურა

1. Fillipo Menczer, Gautam Pant and Padmini Srinivasan The University of Iowa Topical Web Crawlers: Evaluating Adaptive Algorithms.
2. Nicholas Holden Web Page Classification with an Ant Colony Algorithm.
3. Web Page Classification: Features and Algorithms* Xiaoguang Qi and Brian D. Davison.
4. Mr.Ramesh Prajapati A Survey Paper on Hyperlink-Induced Topic Search (HITS) Algorithms for Web Mining.
5. დავით ზაუტაშვილი ინტერნეტი 2008წ.
6. A Survey of Web Crawler Algorithms Pavalam S M1, S V Kashmir Raja2, Felix K Akorli3 and Jawahar M4.
7. საერთაშორისო სამეცნიერო ჟურნალი ინტელექტი 1(48). 2014წ. „ეფექტური ინტერნეტ ძიების ალგორითმები(PageRank,MartxNET) მიმოხილვა“ გვ 63. ლაშა გავაშელაშვილი.
8. აკაკი წერეთლის სახელმწიფო უნივერსიტეტი IV საერთაშორისო სამეცნიერო მეთოდური კონფერენცია. სწავლებისა და აღზრდის აქტუალური საკითხები. 2013 შრომები. „ინფორმაციის ძიების ოპტიმალური მეთოდების შემუშავების საკითხები“ გვ.191. ლაშა გავაშელაშვილი.
9. პედაგოგთა კვალიფიკაციის ამაღლების ქუთაისის ზონალური ინსტიტუტი.
I საერთაშორისო სამეცნიერო-მეთოდური კონფერენცია. განათლებისა და მეცნიერების მაღალი ინტელექტუალური ტექნოლოგიები 2013.შრომები „ინტერნეტში ძიების თანამედროვე საშუალებების ანალიზი“ გვ. 16. ლაშა გავაშელაშვილი.
10. მეცნიერებისა და საზოგადოების განვითარების ფონდი "ინტელექტი"-საქართველო III საერთაშორისო სამეცნიერო კონფერენცია.

- თეზისები. 2014 „ინტერნეტ ძიების ალგორითმი და GOOGLE“-ს არქიტექტურა“ გვ.37. ლაშა გავაშელაშვილი.
11. მეცნიერებისა და საზოგადოების განვითარების ფონდი "ინტელექტი"-საქართველო III საერთაშორისო სამეცნიერო კონფერენცია. თეზისები.2014. „ვებ-გვერდების კლასიფიკაცია“ გვ.38. ლაშა გავაშელაშვილი.
 12. პედაგოგთა კვალიფიკაციის ამაღლების ქუთაისის ზონალური ინსტიტუტი.
რესპუბლიკური სამეცნიერო-მეთოდური კონფერენცია II შრომები „ინტერნეტ ძიების ალგორითმები“ გვ.27. ლაშა გავაშელაშვილი.
 13. მეცნიერებისა და საზოგადოების განვითარების ფონდი "ინტელექტი"-საქართველო III საერთაშორისო სამეცნიერო კონფერენცია. შრომები „ვებ-გვერდების კლასიფიკაცია“ ლაშა გავაშელაშვილი გურამ ჩაჩანიძე რომან სამხარაძე.
 14. მეცნიერებისა და საზოგადოების განვითარების ფონდი "ინტელექტი"-საქართველო III საერთაშორისო სამეცნიერო კონფერენცია. შრომები „ინტერნეტ ძიების ალგორითმი და GOOGLE“-ს არქიტექტურა“ ლაშა გავაშელაშვილი.
 15. https://developer.chrome.com/extensions/api_other. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
 16. http://en.wikipedia.org/wiki/Google_Panda. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
 17. http://en.wikipedia.org/wiki/Google_Penguin. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
 18. <http://en.wikipedia.org/wiki/Hummingbird>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
 19. [http://mastsavlebeli.ge/uploads/matematika/google-MATH%20\(1\).pdf](http://mastsavlebeli.ge/uploads/matematika/google-MATH%20(1).pdf) უკანასკნელად იქნა გადამოწმებული - 13.05.2015.

20. <http://blog.hubspot.com/marketing/history-of-seo-glimpse-into-future-tl>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
21. <http://www.zrabo.com/seo-services/>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
22. <http://mor.ge/wm/5150-bb-bbbb-google-page-rank.html>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
23. <http://wordnet.princeton.edu>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
24. <http://www.cs.princeton.edu/chazelle/courses/BIB/pagerank.htm>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
25. <http://www.ams.org/samplings/feature-column/fcarc-pagerank>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
26. <http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
27. <http://pr.efactory.de/e-pagerank-algorithm.shtml>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
28. <http://infolab.stanford.edu/backrub/google.html>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
29. <http://www.exler.ru/exprompt/06-07-2006.htm>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
30. Ranking Based Web Search Algorithms Akshata D. Deore*, Prof. R. L. Paikrao** უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
31. http://en.wikipedia.org/wiki/Search_engine_optimization. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
32. http://ka.wikipedia.org/wiki/საძიებო_სისტემის_ოპტიმიზაცია. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
33. <http://mor.ge/wm/5150-bb-bbbb-google-page-rank.html>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.

34. <http://wordnet.princeton.edu/>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
35. <http://en.wikipedia.org/wiki/WordNet>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
36. <http://filosof.historic.ru/books/item/f00/s00/z0000690/st000.shtml>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
37. <http://ka.wikipedia.org/wik/ინტერნეტი>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
38. <http://levanasatiani.wordpress.com>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015.
39. [http://www.gen.ge/ინფორმაციული საზოგადოება](http://www.gen.ge/ინფორმაციული_საზოგადოება). უკანასკნელად იქნა გადამოწმებული - 13.05.2015
40. <https://seingeorgia.wordpress.com/>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
41. <https://seingeorgia.wordpress.com/სადიებო-სისტემების-ოპტიმ/>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
42. <http://scripts.ge/რა-არის-seo/>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
43. <http://infolab.stanford.edu/~ullman/mining/pdf/pagerank.pdf>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
44. <http://www.smashingmagazine.com/pdfs/pagerank.pdf>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
45. <http://searchengineland.com/library/google/hummingbird-google>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
46. <https://blog.kissmetrics.com/google-hummingbird/>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015
47. <http://static.googleusercontent.com/media/research.google.com/en/us/archive/googlecluster-ieee.pdf>. უკანასკნელად იქნა გადამოწმებული - 13.05.2015

48. <http://www.cs.cmu.edu/~dpelleg/bin/360.html>. უკანასკნელად იქნა
გადამოწმებული - 13.05.2015
49. <http://searchengineland.com/guide/what-is-seo>. უკანასკნელად იქნა
გადამოწმებული - 13.05.2015
50. <http://wordnet.princeton.edu/>. უკანასკნელად იქნა გადამოწმებული -
13.05.2015

დანართი №1 ექსპერიმენტის შედეგები(3 გვერდი)

დანართი №1 (3გვერდი)

Google ხელოვნური ინტელექტი

Web Images Videos Maps News More Search tools

About 41,100 results (0.39 seconds)

ხელოვნური ინტელექტი - ვიკიპედია - Wikipedia.Org
ka.wikipedia.org/wiki/ხელოვნური_ინტელექტი Translate this page
ხელოვნური ინტელექტი (ინგლ. Artificial Intelligence, შემოკლებით AI) არის კომპიუტერული მეცნიერების დარგი, რომელიც მიზნად ისახავს ...

ხელოვნური ინტელექტი / A.I. Artificial Intelligence ...
srulad.com/.../290-a-i-artificial-intelligence-xelovhuri-... Translate this page
ხელოვნური ინტელექტი A.I. Artificial Intelligence · A.I. Artificial Intelligence · ხელოვნური ინტელექტი. წელი: 2001. ენები: geo , eng , rus. კანრი: დრამა ...

Artificial Intelligence: AI / ხელოვნური ინტელექტი ...
geo.saitebi.ge/.../1962-artificial-intelligence-ai-bbbbbb... Translate this page
დასახელება: ხელოვნური ინტელექტი ორიგინალი დასახელება: Artificial Intelligence- AI გამოსვლის წელი: 2001 კანრი: ფანტასტიკა, დრამა, სათავგადასავლო, ...

Images for ხელოვნური ინტელექტი Report images

More images for ხელოვნური ინტელექტი

ხელოვნური ინტელექტი - - ცოდნის ბლოგი
icode.ge/2013/02/07/ხელოვნური-ინტელექტი/ Translate this page
Feb 7, 2013 - Artificial intelligence (AI) ანუ ჩვენებურად ხელოვნური ინტელექტი, სამეცნიერო ფანტასტიკიდან თანამედროვე კომპიუტერული მეცნიერების ...

ხელოვნური ინტელექტი / Artificial Intelligence: AI (2001)
www.imovies.ge/movies/3595 Translate this page
ხელოვნური ინტელექტი. Artificial Intelligence: AI. 2001. დრამა, ფანტასტიკა, სათავგადასავლო. USA. 146 Min. Currently 4.3/5 Stars. 1; 2; 3; 4; 5. IMDb7.0 ...

სურათი 1. №1 ექსპერიმენტის შედეგი.

About 25,600 results (0.47 seconds)

ხელოვნური ინტელექტი - ვიკიპედია - Wikipedia.Org
ka.wikipedia.org/wiki/ხელოვნური_ინტელექტი Translate this page
ხელოვნური ინტელექტი (ინგლ. Artificial Intelligence, შემოკლებით AI) არის კომპიუტერული მეცნიერების დარგი, რომელიც მიზნად ისახავს ...

ხელოვნური ინტელექტი / A.I. Artificial Intelligence ...
srulad.com/.../290-a-i-artificial-intelligence-xelovnuri-... Translate this page
ხელოვნური ინტელექტი A.I. Artificial Intelligence - A.I. Artificial Intelligence -
ხელოვნური ინტელექტი. წელი: 2001. ენები: geo , eng, rus. ჟანრი: დრამა ...

ხელოვნური ინტელექტი - - ცოდნის ბლოგი
icode.ge/2013/02/07/ხელოვნური-ინტელექტი/ Translate this page
Feb 7, 2013 - Artificial intelligence (AI) ანუ ჩვენებურად ხელოვნური ინტელექტი,
სამეცნიერო ფანტასტიკიდან თანამედროვე კომპიუტერული მეცნიერების ...
You've visited this page 2 times. Last visit: 4/2/15

Images for "ხელოვნური ინტელექტი"

Report images



More images for "ხელოვნური ინტელექტი"

Artificial Intelligence: AI / ხელოვნური ინტელექტი ...
geo.saitebi.ge/.../1962-artificial-intelligence-ai-bbbbbb... Translate this page
დასახელება: ხელოვნური ინტელექტი ორიგინალი დასახელება: Artificial
Intelligence- AI გამოსვლის წელი: 2001 ჟანრი: ფანტასტიკა, დრამა,
სათავადასავლო, ...

ხელოვნური ინტელექტი / Artificial Intelligence: AI (2001)
www.imovies.ge/movies/3595 Translate this page
ხელოვნური ინტელექტი. Artificial Intelligence: AI. 2001. დრამა, ფანტასტიკა,
სათავადასავლო. USA. 146 Min. Currently 4.3/5 Stars. 1; 2; 3; 4; 5. IMDb7.0 ...

სურათი 2. №2 ექსპერიმენტის შედეგი.

8 results (0.59 seconds)

No results found for "ხელოვნური ინტელექტი+რობოტი".

Results for **ხელოვნური ინტელექტი+რობოტი** (without quotes):**ხელოვნური ინტელექტი - ვიკიპედია - Wikipedia.Org**ka.wikipedia.org/wiki/ხელოვნური_ინტელექტი ▾ [Translate this page](#)**რობოტები. ხელოვნური ინტელექტი** (ინგლ. Artificial Intelligence, შემოკლებით AI) არის კომპიუტერული მეცნიერების დარგი, რომელიც მიზნად ისახავს ...**ხელოვნური ინტელექტი - - ცოდნის ბლოგი**icode.ge/2013/02/07/ხელოვნური-ინტელექტი/ ▾ [Translate this page](#)

Feb 7, 2013 - ეს დაყოფა გამოდინარეობს პრინციპიდან, თუ რამდენად შეუძლია

ხელოვნური ინტელექტის მქონე ფიზიკურ აგენტს (მარტივად: **რობოტს**) ...

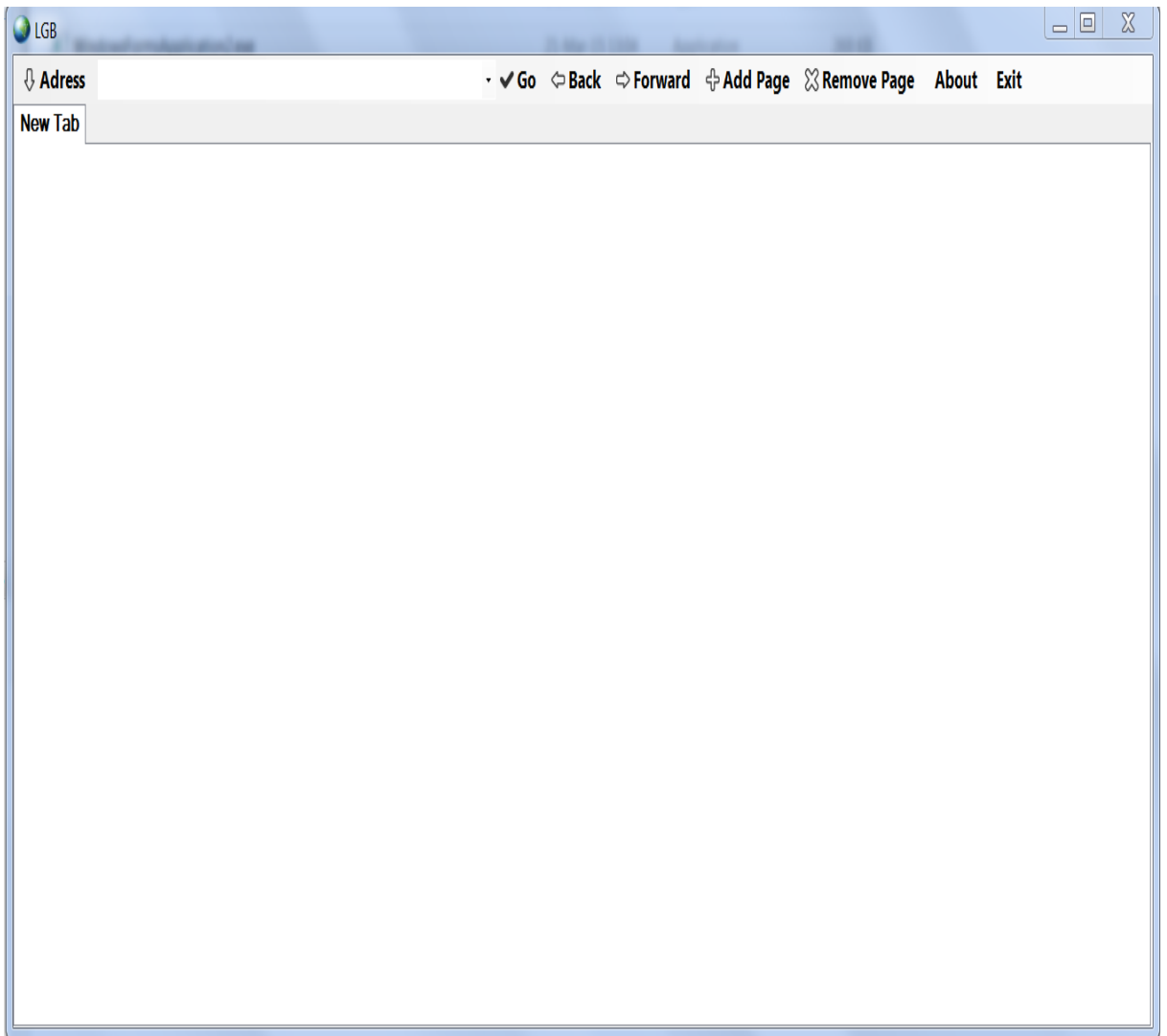
You've visited this page 2 times. Last visit: 4/2/15

ხელოვნური ინტელექტი / A.I. Artificial Intelligence ...srulad.com/.../290-a-i-artificial-intelligence-xelovnuri-... ▾ [Translate this page](#)პატარა ბავშვი **რობოტი** დევიდის გაჩენის შემდეგ მისი მშობლები მზად არ

აღმოჩნდებიან ამ ცვლილებისთვის.ამიტომ იგი მარტო აღმოჩნდება მისთვის ...

კიბერნეტიკა და ხელოვნური ინტელექტი - მაღალი მატერია<https://fizikosi.wordpress.com/.../კიბერნეტიკა-და-ხე...> ▾ [Translate this page](#)Feb 11, 2010 - თვალსაჩინოებისათვის ავიღოთ **ხელოვნური ინტელექტის**თანამედროვე ნიმუში Asimo. ეს ჰუმანოიდი **რობოტი** გახლავთ იაპონური ...**ზოგადად რობოტები - უცნაური ბლოგი**ucnauri.com/229501/ზოგადად-რობოტები/ ▾ [Translate this page](#)მართალია,მრავალი სახეობისა და დანიშნულების **რობოტი** არსებობს , მაგრამ მათი... **ხელოვნური ინტელექტი** კომპიუტერული მეცნიერებისა და ინჟინერიის ...**ხელოვნური ინტელექტი შეიქმნა - უცნაური ბლოგი**ucnauri.com/195263/ხელოვნური-ინტელექტი-შეიქ/ ▾ [Translate this page](#)უკვე არაერთი ათწლეულია რაც ადამიანი **ხელოვნური ინტელექტის** ... აქამდე**რობოტებს** მხოლოდ იმ პროგრამების შესრულება შეეძლოთ რაც მათ სისტემაში ...**სურათი 3. №3 ექსპერიმენტის შედეგი.**

დანართი №2 კონცეფციის პრაქტიკული რეალიზაცია(1 გვერდი)



სურათი 1. აპლიკაციის ფანჯარა

№3 აპლიკაციის ფსევდოკოდი(4 გვერდი)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
using System.Data.SqlClient;
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        //string constring ="Provider=Microsoft.ACE.OLEDB.12.0;Data
        Source=webnavigation.accdb; Jet OLEDB:Database;";
        OleDbConnection con = new
        System.Data.OleDb.OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data
        Source=webnavigation.accdb");
        int a = 1;
        public Form1()
        {
            InitializeComponent();

        }
        WebBrowser web = new WebBrowser();
        int i = 0;
        private void Form1_Load(object sender, EventArgs e)
        {
            web = new WebBrowser();
            web.ScriptErrorsSuppressed = true;
            web.Dock = DockStyle.Fill;
            web.DocumentCompleted += web_DocumentCompleted;
            web.Visible = true;
            tabControl1.TabPages.Add("New Tab");
            tabControl1.SelectTab(i);
            tabControl1.SelectedTab.Controls.Add(web);
            i += 1;
        }
        void web_DocumentCompleted(object sender,
        WebBrowserDocumentCompletedEventArgs e)
        {
```

```

        tabControl1.SelectedTab.Text =
((WebBrowser)tabControl1.SelectedTab.Controls[0]).DocumentTitle;
    }
    private void goToolStripMenuItem_Click(object sender, EventArgs e)
    {
        timer1.Enabled = true;

((WebBrowser)tabControl1.SelectedTab.Controls[0]).Navigate(toolStripComboBox1.Text);

        if (!toolStripComboBox1.Items.Contains(toolStripComboBox1.Text))
        {
            toolStripComboBox1.Items.Add(toolStripComboBox1.Text);
            con.Open();
            string shedareba = "SELECT Name, intensivity from webrange";
            OleDbDataReader reader = null;
            OleDbCommand shesruleba = new OleDbCommand(shedareba, con);
            reader = shesruleba.ExecuteReader();
            while (reader.Read())
            {
                string shesadarebeli = reader["Name"].ToString();
                string intensivoba = reader["intensivity"].ToString();
                int b=0;
                b=Convert.ToInt32(intensivoba);
                if (shesadarebeli == toolStripComboBox1.Text)
                {

                    string ganaxleba = string.Format("UPDATE webrange SET intensivity={0}
WHERE Name='{1}'", ++b, shesadarebeli);
                    OleDbCommand cmdi = new OleDbCommand(ganaxleba, con);
                    cmdi.ExecuteNonQuery();
                    con.Close();

                }
                else
                {
                    string sqlstring = string.Format("INSERT INTO webrange (name,
intensivity) VALUES ('{0}','{1}'",
toolStripComboBox1.Text, a);
                    OleDbCommand cmd = new OleDbCommand(sqlstring, con);
                    cmd.ExecuteNonQuery();
                    con.Close();
                    reader = null;
                }
                break;
                reader = null;
                shesadarebeli =null;
            }
        }
    }
}

```

```

}
private void backToolStripMenuItem_Click(object sender, EventArgs e)
{
    ((WebBrowser)tabControl1.SelectedTab.Controls[0]).GoBack();
}
private void forwardToolStripMenuItem_Click(object sender, EventArgs e)
{
    ((WebBrowser)tabControl1.SelectedTab.Controls[0]).GoForward();
}
private void addPageToolStripMenuItem_Click(object sender, EventArgs e)
{
    web = new WebBrowser();
    web.ScriptErrorsSuppressed = true;
    web.Dock = DockStyle.Fill;
    web.DocumentCompleted += web_DocumentCompleted;
    web.Visible = true;
    tabControl1.TabPages.Add("New Tab");
    tabControl1.SelectTab(i);
    tabControl1.SelectedTab.Controls.Add(web);
    i += 1;
}
private void removePageToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (tabControl1.TabPages.Count - 1 > 0)
    {
        tabControl1.TabPages.RemoveAt(tabControl1.SelectedIndex);
        tabControl1.SelectTab(tabControl1.TabPages.Count - 1);
        i += 1;
    }
}
private void toolStripComboBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;

        ((WebBrowser)tabControl1.SelectedTab.Controls[0]).Navigate(toolStripComboBox1.Text);
        if (!toolStripComboBox1.Items.Contains(toolStripComboBox1.Text))
        {
            toolStripComboBox1.Items.Add(toolStripComboBox1.Text);
        }
    }
}
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show(" LGB Version 1.0\n All Right Reserved\n Lasha
Gavashelashvili", "about", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

```
private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Are you sure that you would like to close
the form?", "Close", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        Application.Exit();
    }
}
/* private void timer1_Tick(object sender, EventArgs e)
{
    timer1.Enabled = true;
    trackBar1.Value = trackBar1.Value + 10;
}
*/
}
```