ივანე ჯავახიშვილის სახელობის თბილისის
სახელმწიფო უნივერსიტეტი

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

საბაკალავრო პროგრამის „კომპიუტერული მეცნიერება“

ჯგუფური პროექტი

კონცეპტის ფორმირების ანალიტიკურ-ევრისტიკული მეთოდის გამოყენება სემანტიკური ძებნის ამოცანებში

ხელმძღვანელი: ასისტენტ-პროფესორი
პაპუნა ქარჩავა

პროექტის შემსრულებლები:

1. აბაშიძე მალხაზი
2. გელაშვილი ნუგზარ
3. გოგალაძე სოფიკო
4. მამულაშვილი თამარ
5. მეძმარიაშვილი ანა
6. ჟვანია დეა
7. რამიშვილი ამირან
8. სარიშვილი ლევან
9. ტარიელაშვილი ლუკა
10. ფასოევი სერგო
11. ფერხული ნათია
12. ჩილაჩავა ლანა

თბილისი

2013 წელი

სარჩევი

ანოტაცია

ჯგუფური პროექტი მიზნად ისახავს კონცეპტის ფორმირების ერთერთი მეთოდის პროგრამული რეალიზაციას, რომელიც ორი მეთოდის Explicit Semantic Analysis (ESA) და ანალიტიკური ევრისტიკების მეთოდის შერწყმით შეიქმნა. ეს მეთოდი შესაძლებელია წარმატებით იქნას ინტეგრირებული და გამოყენებული ინფორმაციული ძებნის სხვადასხვა მეთოდებთან ერთად. ბუნებრივ ენებზე შედგენილი ტექსტის კომპიუტერული დამუშავება ემყარება კონცეპტების ფორმირებას და მათ შორის მიმართებების აღწერას, რომელიც არის სემანტიკური ძებნის საფუძველი.

Annotation

The sense of this topic is to give a program realization of one method of concept formulation, which is derived from method ESA and method of analytic heuristic. This method can be integrated information retrieval with other methods. In natural languages compound text computational process is based on concept formatting and describes relationship with them, which is fundamental for semantic retrieval.

შესავალი

ტერმინის „ინფორმაციული ძებნა“ (IR – Information Retrieval) არსი შეიძლება იყოს საკმაოდ ფართო. მომხმარებლის მიერ საკუთარი ელექტრონული ფოსტის დათვალიერება, ინტერნეტ სივრცეში ამინდიც ცნობის გაგება და ა.შ. შეიძლება ჩაითვალოს როგორ ინფორმაციის ძებნა. ინფორმაციული როგორც სამეცნიერო დისციპლინა შეიძლება განისაზღვროს შემდეგნაირად:

ინფორმაციული ძებნა ეს არის გარკვეულ არასტრუქტურირებულ მონაცემების (რომლებიც როგორც წესი ინახება კომპიუტერში) დიდ კოლექციაში ძებნის პროცესი, რომელიც გარკვეულ მოთხოვნას აკმაყოფილებს.

უნდა აღინიშნოს, რომ დიდი ხნის მანძილზე ინფორმაციის ძებნით დაკავებული იყვნენ მხოლოდ ამისათვის სპეციალურად მომზადებული და გარკვეული გამოცდილების მქონე სპეციალისტები, ხოლო ძებნის მეთოდოლოგია კი ეყრდნობოდა ე.წ. გასაღებ სიტყვებს, რომლებიც ენიშნებოდა დოკუმენტებს, და ბულის რთულ მოთხოვნებს. 1970-იანი წლებიდან, მას შემდეგ რაც პოპულარობა მოიპოვა ავტომატური ინდექსირების საშუალებებმა და ბუნებრივ ენოვანმა მოთხოვნებმა, IR სისტემები ხელმისაწვდომი გახდა არასპეციალისტ მომხმარებელთათვისაც. ინდექსირებულ დოკუმენტში ყოველი ტერმინი ავტომატურად განიხილება დამოუკიდებელ გასაღებ სიტყვად (რომელიც ცნობილია BOW (Bag-of-Words) წარმოდგენის სახელით) და მოთხოვნის ფორმატირება გამარტივებულია ბუნებრივ ენოვანი ფორმულირებით. ამასთან, არ შეცვლილა გასაღები სიტყვების ინდექსირების მეთოდოლოგია და არასპეციალისტი მომხმარებელი ხშირად აწყდება პრობლემას, რომელიც ცნობილია „ლექსიკონის პრობლემის“ სახელით [furnas 1987]. მომხმარებლის მიერ გამოყენებული გასაღები სიტყვები ხშირად განსხვავდება რელევანტური დოკუმენტის ავტორის მიერ გამოყენებული გასაღები სიტყვისაგან, რაც ამცირებს IR სისტემის ეფექტურობის ხარისხს. მეორეს მხრივ, კონტექსტური განსხვავება ორაზროვან გასაღებ სიტყვებს შორის, რომელიც არსებობს BOW მეთოდში, ამცირებს შედეგის სიზუსტეს. ეს ორი პრობლემა გამოწვეულია შესაბამისად გასაღები სიტყვის სინონიმებითა და მრავალმნიშვნელობით.

სინონიმების პრობლემის გადაწყვეტის მიზნით IR მკვლევარების მიერ შემოთავაზებული იყო გამოსავალი მოთხოვნის გაფართოება გასაღები სიტყვის სინონიმებით [Voorhees 1994]. თუმცა, მომხმარებლის და რელევანტური დოკუმენტის ავტორის მიერ გამოყენებული გასაღები სიტყვის გაფართოება სინონიმებით შეიძლება ცდებოდეს მარტივ სინონიმებს.

მსგავსი პრობლემის გვერდის ავლის მიზნით შემოთავაზებული იყო მოთხოვნის გაფართოების ახალი მეთოდები. მაგალითად, Xu და Croft [2000] მიერ შემოთავაზებული მეთოდი, რომელიც გულისხმობს გასაღებ სიტყვის განმმარტავი ტერმინების იდენტიფიცირებას მოთხოვნასთან მაღალი რანჟირების მქონე დოკუმენტებში, რომლებიც გამოყენებული იქნება ტერმინის გაფართოების მიზნით. ასეთმა გაფართოებამ შეიძლება მიგვიყვანოს ძებნის შედეგის მნიშვნელოვან გაუმჯობესებამდე, მაგრამ მან ასევე შეიძლება უარყოფითი ზეგავლენა იქონიოს ძებნის სისწრაფეზე: ტერმინის უმნიშვნელო გაფართოებით შეიძლება მცირე დროში მივიღოთ რელევანტური დოკუმენტი, მაგრამ დიდად გაფართოების შემთხვევაში კი მნიშვნელოვანად შემცირდეს ძებნის მეთოდის სწრაფქმედება.

მრავალმნიშვნელობის გვერდის ავლის მიზნით შეთავაზებული იქნა მოთხოვნაში და დოკუმენტში ორაზროვანი სიტყვების ავტომატური ამოცნობის ალგორითმის გამოყენება. ორაზროვანი მეთოდი იყენებს ისეთ რესურსს, როგორიცაა WordNet ლექსიკონი [Voorhees 1993] ან მონაცემები [schuetze and pedersen 1995], რომელთა გამოყენებით მოხდებოდა გასაღები სიტყვის შესაბამისი სინონიმების მოძებნა და მათი ასახვა სწორ მნიშვნელობასთან, რომელიც შემდგომ გამოყენებული იქნება ინდექსირების და მოთხოვნის ფორმირების პროცესში, ასე, რომ მხოლოდ ის დოკუმენტები, რომლებიც ემთხვევა მოთხოვნის სწორ მნიშვნელობას იქნება მოძებნილი. ამ მეთოდის გამოყენებით მნიშვნელოვანი გაუმჯობესების მთავარ დაბრკოლებას წარმოადგენს ორაზროვნების ავტომატური ამოცნობის ცდომილება, ვინაიდან ის უარყოფითად აისახება მეთოდის წარმადობაზე.

კონცეპტზე დაფუძნებული ინფორმაციის ძებნა არის ალტერნატიული IR მიდგომა, რომელიც ისწრაფვის განსხვავებულად მიუდგეს ამ პრობლემას. კონცეპტზე დაფუძნებული IR მეთოდი ნაცვლად გასაღები სიტყვებისა დოკუმენტებს და მოთხოვნებს წარმოგვიდგენს სემანტიკური თვალსაზრისით და ძებნას ანხორციელებს ამავე სივრცეში. მაღალი დონის კონცეპტების გამოყენებით მივიღებთ ძებნის მოდელს, რომელშიც დოკუმენტების და მოთხოვნის წარმოდგენა არ იქნება დამოკიდებული გასაღებ სიტყვაზე [styltsvig 2006]. ასეთი მოდელი უფრო ზუსტად მიგვიყვანს სასურველ შედეგამდე, ვიდრე მოთხოვნაში იგივე გასაღების სხვა ტერმინებით აღწერილი მოდელი, რომელშიც სინონიმების პრობლემა არ იქნებოდა გადაწყვეტილი. შესაბამისად, თუ კონცეპტი სწორად იქნება შერჩეული ორაზროვანი სიტყვები, რომელიც გამოყენებულია მოთხოვნაში და

დოკუმენტში, ამოვარდება რელევანტური დოკუმენტების ჩამონათვალიდან, მრავალმნიშვნელობის პრობლემის გვერდის ავლით და სიზუსტის გაზრდით.

სემანტიკური თვალსაზრისით ზუსტი პასუხის მისაღებად აუცილებელია თვით მომხმარებლის მოთხოვნა წარმოსდგეს ისეთი ფორმალიზებული სახით, რომ არ დაირღვეს მისი შინაარსობრივი მნიშვნელობა.

არსებობს მთელი რიგი მეთოდები, რომლებიც ნაწილობრივ აგვარებენ ამ პრობლემას, მაგრამ უმრავლესი მათგანი დიდი მოცულობის ტექსტების სახით წარმოდგენილი ცოდნის სტატისტიკურ დამუშავებაზეა დაფუძნებული. არსებული სტატისტიკური ანალიზისაგან განსხვავდება Explicit Semantic Analysis (ESA) მეთოდი [1], რომელიც სემანტიკურად ზუსტად წარმოადგენს განუსაზღვრელი მოცულობის ბუნებრივ ენოვან ტექსტებს. ეს მეთოდი ძირითადად ეფუძნება ცნებათა ფორმირებას ვიკიპედიაში განთავსებული სხვადასხვა სტატიების ანალიზით. კვლევები რომლებიც ცნებათა ფორმირების მოდელირების სფეროში გასული საუკუნის 60-იანი წლებიდან მიმდინარეობს [2,3], აქტუალობას არ კარგავს და ნებისმიერმა ახალმა ან მოდიფიცირებულმა მოდელმა შესაძლოა ახალი რეალობები წარმოაჩინოს კონცეპტების გამოყენების შესაძლებლობებში.

არანაკლებ საინტერესოა ცნებათა ფორმირების, სახეთა ამოცნობის და ობიექტთა კლასიფიკაციის ანალიტიკური ევრისტიკების მეთოდი [4]. ეს მეთოდი ასევე წარმატებით გამოიყენებოდა ცოდნის ბაზის ფორმირებისათვის სხვადასხვა დანიშნულების ექსპერტული სისტემებისათვის [5,6,7].

ტერმინების ლექსიკონის განსაზღვრა

დოკუმენტიდან სიმბოლოების მიმდევრობის იდენტიფიცირების შემდეგ ტექსტის დაყოფა ლექსემებად. გარდა ამისა, ხშირად მისგან იშლება ზოგიერთი სიმბოლოები, მაგალითად როგორიცაა სასვენი ნიშნები და ხდება ზედა რეგისტრში გამოყენებული სიმბოლოების გადაყვანა ქვედა რეგისტრში, რათა არ მოხდეს ერთიდაიმავე ტერმინის ლექსიკონში განსხვავებულ ტერმინად გათვალისწინება.

ლექსემა (token) ეს არის, გარკვეულ დოკუმენტში ეგზემპლარი სიმბოლოების მიმდევრობისა, რომლებიც დამუშავების მიზნით გაერთიანებულია სემანტიკურ ერთეულად. ტიპი (type) ეს არის ყველა ლექსემას კლასი, რომელიც შედგება ერთიდაიმავე სიმბოლოების მიმდევრობისაგან. ტერმინი (term) ეს არის ტიპი, რომელიც ჩართულია ინფორმაციული ძებნის ლექსიკონში. ტერმინების სიმრავლე შეიძლება სრულად განსხვავდებოდეს ლექსემასაგან, რომელებიც მაგალითად შეიძლება წარმოადგენდნენ სემანტიკურ იდენტიფიკატორებს იერარქიაში, მაგრამ თანამედროვე ინფორმაციული ძებნის სისტემებში პირდაპირ დაკავშირებული არიან ლექსემებთან დოკუმენტში.

Stop-word

ხანდახან ხშირად გამოყენებადი სიტყვები არ წარმოადგენენ არანაირ ღირებულებას ინფორმაციული ძებნისადმი მომხმარებლის მოთხოვნისათვის. ასეთ სიტყვებს stop-word-ს უწოდებენ. ასეთი სიტყვების ნაკრები ძირითადად შეიძლება შექმნილი იქნას წინასწარ. ძირითადად ეს არის არტიკლები, კავშირები და ა.შ., რომელთა გამოტოვება (იგნორირება) ინფორმაციული ძენისას ურყოფითად არ აისახება შედეგზე. ასეთი ტერმინებს შეიძლება მივაკუთვნოთ

a an and are as at be by for from has

he in is it its of on that the to …

ინვერტირებული ინდექსი

ინვერტირებული ინდექსის გამოყენებით შესაძლებელია ძებნის სისწრაფის ამაღლება. ინვერტირებული ინდექსის აგების მიზნით დავუშვათ დოკუმენტების კოლექციაში თითოეულ დოკუმენტს გააჩნია მიმდევრობის საკუთარი უნიკალური ნომერი, რომელსაც დოკუმენტის იდენტიფიკატორს უწოდებენ (docID). ინდექსის აგებისას შესაძლებელია ეს მიმდევრობითი ნომერი მივანიჭოთ ყოველ დოკუმენტს მასზე პირველი მიმართვისას. ინდექსირებისათვის შემავალ მონაცემებს წარმოადგენს ყოველი დოკუმენტისატვის ნორმალიზებული ლექსემების (ლექსემა - ეს არის გარკვეულ

დოკუმენტში სიმბოლოების მიმდევრობის ეგზემპლარი, რომელიც დამუშავების მიზნით გაერთიანებულია სემანტიკურ ერთეულად) ჩამონათვალი, რომელიც შეგვიძლია განვიხილოთ, როგორც სია წყვილებისა „ტერმინი-docID“. ინდექსირების ძირითად ეტაპს წარმოადგენს სიის ისეთი სორტირება, რომლის დროსაც ტერმინები დალაგებული იქნება ალფავიტური მიმდევრობით. ერთიდაიმავე დოკუმენტში ერთნაირი ტერმინები ერთიანდება. ამის შემდეგ ერთიდაიმავე ტერმინის ეგზემპლარების რაოდენობა ჯამდება, ხოლო შედეგი იყოფა ლექსიკონად (dictionary) და სიტყვათ წყობებად (postings). ვინაიდან ტერმინი შეიძლება გვხვდებოდეს მრავალ დოკუმენტში, შესაბამისად მათი ასეთი ორგანიზაცია იძლევა ინდექსის შემცირების საშუალებას. ლექსიკონი ასევე შეიძლება შეიცავდეს ისეთ სტატიკურ მაჩვენებელს, როგორიცაა მაგალითად, იმ დოკუმენტების რაოდენობა, რომლებიც შეიცავენ ტერმინს, ანუ დოკუმენტების სიხშირე (document frequency). ეს ინფორმაცია შეიძლება გამოყენებული იქნას ინფორმაციული ძებნის რანჟირების მრავალ მოდელში.

ტერმინის წონა

დოკუმენტი ან მისი ნაწილი, სადაც ყველაზე ხშირად გვხვდება მოთხოვნილი ტერმინი ითვლება მოთხოვნის რელევანტურ დოკუმენტად და გას უნდა გააჩნდეს რელევანტურობის მაღალი კოეფიციენტი. ვებ სივრცეში მოთხოვნას განიხილავენ როგორც სიტყვების ნაკრებს, რომელიც მომხმარებელს შეჰყავს საძიებო მანქანის ინტერფეისში თავისუფალი ფორმით, ყოველგვარი დამაკავშირებელი ოპერატორების (როგორიცაა ბულის ოპერატორები) გარეშე. შესაბამისად ასეთი დოკუმენტის მაჩვენებლის დასათვლელად საკმარისია დაჯამდეს მოთხოვნაში გამოყენებული ყველა სიტყვისათვის მიმდინარე დოკუმენტის შესაბამისობის მაჩვენებლები.

ამ მიზნით, დოკუმენტში გამოყენებულ ყოველ ტერმინს ენიჭება მისი წონა (weight), რომელიც დამოკიდებულია მიმდინარე დოკუმენტში ამ ტერმინის განმეორებათა რაოდენობაზე. d დოკუმენტში t ტერმინის წონის ყველაზე მარტივ მაგალითს წარმოადგენს ამ ტერმინის წონის როლში მისი განმეორებათა რაოდენობის აღება. წონის ასეთ სქემას ტერმინის სიხშირეს (term frequency) უწოდებენ და აღნიშნავენ შემდეგნაირად $\mathrm{tf}_{t,d}$, სადაც ინდექსი t აღნიშნავს დოკუმენტში გამოყენებულ კონკრეტულ ტერმინს, ხოლო d ინდექსი კი მიმდინარე დოკუმენტს.

d დოკუმენტისათვის tf წონების სიმრავლე შეიძლება ინტერპრეტირებული იქნას, როგორც დოკუმენტის დაიჯესტი, გამოსახული რიცხვითი ფორმით. სამეცნიერო

ლიტერატურაში ეს მეთოდი ცნობილია BOW (Bag-of-Words) სახელით. ამ მეთოდის მიხედვით ხდება დოკუმენტში გამოყენებული ტერმინების ზუსტი მიმდევრობის იგნორირება, მნიშვნელოვანია მხოლოდ მათი განმეორებათა რაოდენობა.

შევნიშნოთ, რომ დოკუმენტში გამოყენებული ყოველი სიტყვა არ შეიძლება იყოს ერთნაირად მნიშვნელოვანი მოთხოვნისათვის. სიტყვებს, რომლებიც არ ზემოქმედებენ ძებნის შედეგზე stop-words-ს ეძახიან. ასეთ სიტყვებს წარმოადგენს არტიკლები, კავშირები და ა.შ.

ინვერტირებული დოკუმენტური სიხშირე

ტერმინის წონის როლში მისი სიხშირის გამოყენებას გააჩნია ნაკლოვანება: დოკუმენტის რანჟირებისას მოთხოვნისათვის ყოველი ტერმინი ითვლება ერთნაირად მნიშვნელოვნად. სინამდვილეში დოკუმენტის რელევანტურობის განსასაზღვრავად ზოგიერთ ტერმინი ნაკლებად მნიშვნელოვანია ან საერთოდ უმნიშვნელო. მაგალითად, მანქანების მწარმოებელ ქარხანაში პრაქტიკულად ყოველი დოკუმენტი შეიძლება შეიცავდეს სიტყვას „auto“. ამ ნაკლოვანების აღმოფხვრის მიზნით იხილავენ წონით კოეფიციენტს, რომელიც იზრდება დოკუმენტში ტერმინის სიხშირესთან ერთად და ამით ამცირებს მის წონას.

ამ მიზნით ხშირად იყენებენ დოკუმენტის სიხშირეს (document frequency) df_t, რომელიც აღნიშნავს t ტერმინის შემცველი დოკუმენტების რაოდენობას. ეს აიხსნება იმითი, რომ მოთხოვნის შესაბამისად რანჟირების მიზნით დოკუმენტებს შორის განსხვავების ძიებისას უმჯობესია გამოყენებული იყოს ამ ტერმინის შემცველი დოკუმენტების სტატიკური მაჩვენებელი (მაგალითად, მათი რაოდენობა), ვიდრე მთლიანი კოლექციის სტატიკური მაჩვენებელია (მაგალითად, კოლექციაში დოკუმენტების საერთო რაოდენობა).

ტერმინის წონის კორექცია დოკუმენტის სიხშირის გამოყენებით შესაძლებელია შემდეგი ფორმულით

$$\mathrm{idf}_t = \log \frac{\mathrm{N}}{\mathrm{df}_t} \qquad (1)$$

სადაც N არის მთლიან კოლექციაში დოკუმენტების რაოდენობა. (1) ფორმულით გამოთვლილ idf_t სიდიდეს ინვერტირებულ დოკუმენტურ სიხშირეს უწოდებენ.

ცხადია, რომ ინვერტირებული დოკუმენტური სიხშირე იშვიათად შემხვედრი სიტყვებისათვის საკმაოდ მაღალია, მაშინ როცა ხშირად შემხვედრი სიტყვებისათვის მისი მნიშვნელობა მცირეა.

$\mathrm{tf}-\mathrm{idf}$ *წონა*

ყოველ დოკუმენტში გამოყენებული ყოველი ტერმინის წონისათვის ახდენენ ტერმინის სიხშირისა და ინვერტირებული დოკუმენტური სიხშირის კომბინირებას. ამ ფორმით ხდება მთლიან კოლექციაში გამოყენებული ყოველი ტერმინისათვის ერთიანი წონის მიღება. $\mathrm{tf}-\mathrm{idf}$ სქემით d დოკუმენტში გამოყენებულ ყოველ t ტერმინს წონა ენიჭება ფორმულით

$$\mathrm{tf}-\mathrm{idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t \ .$$

ცხადია, რომ თუ ტერმინი გვხვდება ხშირად და დიდი რაოდენობის დოკუმენტებში, მაშინ მისი წონა მიაღწევს მაქსიმალურ მნიშვნელობას. ტერმინის წონა მიაღწევს მინიმალურ მნიშვნელობას, თუ ის პრაქტიკულად გვხვდება თითქმის ყველა დოკუმენტში.

შევნიშნოთ, რომ ლექსიკონური ტერმინის წონა, თუ ის გვხვდება მიმდინარე დოკუმენტში, გამოითვლება (1) ფორმულის გამოყენებით. წინააღმდეგ შემთხვევაში მისი წონა 0-ის ტოლია.

$\mathrm{tf}-\mathrm{idf}$ სქემით d დოკუმენტის რელევანტურობა გამოითვლება ფორმულით

$$\mathrm{Score}(q, d) = \sum_{t \in d} \mathrm{tf}-\mathrm{idf}_t$$

სადაც ჯამში მონაწილეობს q მოთხოვნაში შემავალი ყოველი t ტერმინის $\mathrm{tf}-\mathrm{idf}$ წონა.

ვექტორული სივრცის მოდელი

ყოველი დოკუმენტი შეიძლება გამოვსახოთ, როგორც ვექტორი ლექსიკონში გამოყენებული ტერმინების წონითი მნიშვნელობებით. დოკუმენტების კოლექციის წარმოდგენას ვექტორების სახით ვექტორულ სივრცეში ეწოდება ვექტორული სივრცის მოდელი (vector space model) და ფუნდამენტური მნიშვნელობისაა ინფორმაციული ძებნის მრავალ ამოცანაში, მოთხოვნის შესაბამისად დოკუმენტის რანჟირებისა და კლასიფიკაციის ჩათვლით.

ასეთ სივრცეში ყოველი d დოკუმენტისათვის აიგება $\vec{v}(d)$ ვექტორი, რომელიც ლექსიკონში გამოყენებული ტერმინისათვის შეიცავს ცალკეულ კომპონენტს (მაგალითად, ამ ტერმინის $\mathrm{tf}-\mathrm{idf}$ სქემის შესაბამის წონას). ამ სიმრავლეში, მსგავსად ჩვეულებრივი

ვექტორული სივრცისა, შესაძლებელია განისაზღვროს ვექტორების სკალარული ნამრავლი, ნორმა და ა.შ.

N დოკუმენტისაგან შემდგარი კოლექციის წარმოდგენა ვექტორების საშუალებით გვაძლევს შესაძლებლობას ბუნებრივად გამოვსახოთ კოლექცია „ტერმინი-დოკუმენტი“ მატრიცის ფორმით, ანუ $\mathrm{M} \times \mathrm{N}$, სადაც M აღნიშნავს ლექსიკონში გამოყენებული ტერმინების რაოდენობას.

დოკუმენტის ვექტორის ფორმით წარმოდგენის კიდევ ერთ მიზეზს წარმოადგენს ის, რომ მოთხოვნა შეიძლება განხილული იქნას როგორც ვექტორი. ასეთი მიდგომის იდეა მდგომარეობს იმაში, რომ ყოველ d დოკუმენტს მიენიჭოს რელევანტურობის მნიშვნელობა, რომელიც ტოლია სკალარული ნამრავლის

$$(\vec{\mathrm{v}}(\mathrm{q}), \vec{\mathrm{v}}(\mathrm{d}))$$

სადაც $\vec{\mathrm{v}}(\mathrm{q})$ არის q მოთხოვნის შესაბამისი ვექტორი.

დოკუმენტის რანჟირება

მოთხოვნა ხშირად შეიძლება წარმოდგენილი იყოს ტერმინების ვექტორის სახით. ასეთი მოთხოვნის შესაბამისი დოკუმენტის რანჟირებისათვის საჭიროა კოლექციაში დოკუმენტის წონის განსაზრვრა. ამისათვის კი საკმარისია გამოითვალოს კუთხის კოსინუსი ცალეკული d დოკუმენტის $\vec{\mathrm{v}}(\mathrm{d})$ ერთეულოვან ვექტორსა და q მოთხოვნის $\vec{\mathrm{v}}(\mathrm{q})$ ვექტორს შორის

$$\cos\left(\vec{\mathrm{v}}(\mathrm{d}), \vec{\mathrm{v}}(\mathrm{q})\right) = \frac{\left(\vec{\mathrm{v}}(\mathrm{d}), \vec{\mathrm{v}}(\mathrm{q})\right)}{\left\|\vec{\mathrm{v}}(\mathrm{d})\right\| \cdot \left\|\vec{\mathrm{v}}(\mathrm{q})\right\|}.$$

კონცეპტების გამოყენება ინფორმაციის ძებნაში

კონცეპტებზე დაფუძნებული ინფორმაციული ძებნა არის ალტერნატიული მიდგომა, რომელიც ეფუძნება ადამიანის მიერ სამყაროს რეალურ აღქმას. კონცეპტებზე დაფუძნებული ძებნისას ობიექტის და მოთხოვნის წარმოდგენა ხდება კონცეპტებით, საგასაღებო სიტყვებით (ტერმინებით) წარმოდგენისაგან (BOW მეთოდი) განსხვავებით. კონცეპტუალურ სივრცეში განხორციელებული ძებნა ნაკლებადაა დამოკიდებული სპეციფიურ ტერმინებზე (საგასაღებო სიტყვებზე), რაც მეტყველებს მის ღირსებებზე. გარდა ამისა, ცნობილია, რომ ძებნის პროცესში საკმაოდ აქტუალურია ომონიმების პრობლემა. კონცეპტებით ძებნა უწყობს ხელს ამ პრობლემის გადაჭრას. კონცეპტებით ძებნა გაზრდის დაბრუნებული შედეგის რელევანტობას. შედეგი იქნება უფრო ზუსტი და სრული. ერთ-ერთი წარმატებული მცდელობა, ცოდნის კონცეპტებით წარმოდგენის, რომელიც გაკეთდა მათემატიკაში, აღწერილია ლენარტის შრომაში [12].

ადამიანის მიერ სამყაროს შემეცნების პროცესი და შესაბამისად ობიექტების აღქმა და ცოდნის დაგროვება ხდება კონცეპტებით. ამაზეა საუბარი ვიგოდსკის შრომაში [13]. ადამიანის მსოფლმხედველობის ფორმირება და მის მიერ სამყაროს შემეცნება და მის მიერ ცოდნის დაგროვების და წარმოდგენის ძირითადი ერთეული არის კონცეპტი [14].

დღეისათვის ცნობილია კონცეპტებზე დაფუძნებული ინფორმაციული ძებნის ESA (Explicit Semantic Analysis) [15], WordNet [16], LSA (Latent Semantic Analysis) [17], WikiRelate [18] მეთოდები.

მეთოდის აღწერა

ESA (Explicit Semantic Analysis) მეთოდი ტერმინის განმმარტავი მონაცემების წყაროდ იყენებს ვიკიპედიის საცავს [9]. ძირითადად გამოიყენება ამა თუ იმ ცნების (აღვნიშნოთ ეს ცნება, როგორც c_j, $j = 1,..., N_{wik}$[1], N_{wik} არის ვიკიპედიის საცავში შემავალი ცნებების რაოდენობა) განმმარტავი ტექსტი (როგორც წესი ერთ ცნებას შეესაბამება ერთი ტექსტი). აღვნიშნოთ ეს ტექსტი, როგორც T_j^{wik}. ყოველი ასეთი ტექსტი წარმოდგება წონითი ნაკრებების სახით, სახელდობრ ე.წ. TF-IDF scheme [10] საშუალებით. სემანტიკური გადამყვანი ახდენს ტექსტის სიტყვების იტერაციას, იღებს ინვერტირებული ინდექსებისაგან შემდგარ შესაბამის ჩანაწერს და აერთიანებს მას ცნების ვექტორში, რომელიც წარმოადგენს ტექსტს.

[1] ზედა ინდექსს (ამ შემთხვევაში wik-ს) ვიყენებთ იმის განსასხვავებლად, რომ ვიყენებთ ერთ კონკრეტულ საცავს მაგ.Wikipedia-ს

ნაშრომი [9]-ის შესაბამისად წარმოვიდგინოთ c_j კონცეპტის აღმწერი ტექსტი w_i სიტყვების სიმრავლის სახით $T_j^{wik} = \{w_i\}, i = 1,...,Mwik$ ($Mwik$ არის ვიკიპედიის საცავში შემავალი სიტყვების რაოდენობა), რომელსაც შეესაბამება TF-IDF $\langle v_i^{wik} \rangle$ ვექტორი, სადაც ყოველ w_i სიტყვას შეესაბამება v_i^{wik} წონა.

სიტყვის (ტერმინის) წონა დამოკიდებულია მის სიხშირეზე. ერთიდაიგივე ტერმინს სხვადასხვა დოკუმენტში შესაძლებელია სხვადასხვა წონა გააჩნდეს. ვექტორული სივრცის მოდელში დოკუმენტში ტერმინის წონები ,,ტერმინების სივრცეში" განიხილება როგორც მისი კოორდინატები. დოკუმენტების და ტერმინების ,,სივრცის" გაერთიანებით მიიღება მატრიცა დოკუმენტი-ტერმინი.

ტერმინის წონების დასადგენად წარმატებით გამოიყენება წონების ავტომატური გენერაციის სქემა „term frequency * inverse document frequency", რომელიც შემდეგნაირად გამოისახება:

$$\mathrm{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

ასევე იქმნება $\langle k_j^{wik} \rangle$ ვექტორი, რომელშიც k_j^{wik} წარმოადგენს w_i სიტყვის ინვერტირებულ ინდექსს c_j კონცეპტის შესაბამისი ტექსტისათვის ვიკიპედიის საცავიდან. შესაბამისად მიიღება საცავში არსებული c_j კონცეპტის შესაბამისი T_j^{wik} ტექსტისათვის გვექნება V_j^{wik} წონების ვექტორი, რომელიც განისაზღვრება როგორც:

$$\sum_{i \in T_j^{wik}} v_i^{wik} \cdot k_j^{wik}$$

აღვწეროთ მიღებული c_j კონცეპტი ანალიტიკური ევრისტიკების მეთოდისათვის [11] მისაღებ ფორმაში. გავამარტივოთ აღნიშვნები და ჩავთვალოთ, რომ ყოველი c_j ზოგადად აღიწერება $w_1^{wik}, w_2^{wik}, ..., w_N^{wik}$ "სიტყვებით". ყოველი w_j^{wik} სიტყვის მოხვედრა აღწერაში განისაზღვრება c_j კონცეპტის შესაბამისი T_j^{wik} ტექსტის V_j^{wik} წონების ვექტორით.

გავაფართოოთ კონცეპტების შესამუშავებელი სივრცე და გამოვიყენოთ ვიკიპედიის მსგავსი სხვა საცავებიც (AllRefer.com, bartleby.com, Britannica.com, infoplease.com, Encyclopedia.com, techweb.com/encyclopedia, libraryspot.com/encyclopedias.htm#science, education.yahoo.com/reference/encyclopedia). ყველა ამ საცავშიც არსებობს c_j კონცეპტის

შესაბამისი T_j^x ტექსტი. შესაბამისად ამ საცავისთვისაც შეგვიძლია გამოვიყენოთ ESA მეთოდი და შემდგომ აღვწერთ $W_1^x, W_2^x, \ldots, W_L^x$ "სიტყვებით" (L არის ამ საცავში შემავალი სიტყვების რაოდენობა). თუ ამ პროცედურას ჩავატარებთ ჩვენს ხელთ არსებული ყველა საცავისათვის, მივიღებთ ერთი c_j კონცეპტის შესაბამის რამდენიმე, შესაძლოა განსხვავებულ, აღწერას.

ცხრილი 1.

საცავები	c_j კონცეპტის აღწერა
Wikipedia	$w_1^{wik}, w_2^{wik}, \ldots, w_N^{wik}$
x	$w_1^x, w_2^x, \ldots, w_L^x$
…	…
y	$w_1^y, w_2^y, \ldots, w_K^y$

ცხადია, რომ ყოველი c_j კონცეპტის აღმწერ სხვადასხვა „ვექტორში" შემავალი სიტყვები მეორდება. გავაერთიანოთ ეს სიტყვები და მივიღოთ საცავების სიტყვების საერთო სიმრავლე $W = \{w_1, w_2, \ldots, w_{max}\}$, max - არის ყველა საცავში არსებული მაქსიმალური განსხვავებული სიტყვის რაოდენობა. ჩავთვალოთ, რომ ეს რიცხვია N. ამ აღნიშვნებში ჩვენ შეგვიძლია c_j კონცეპტის აღმწერი ყველა ვექტორი წარმოვადგინოთ ერთი და იგივე N სიგრძის ვექტორის სახით, რომლის ელემენტებია $\widetilde{w_i}$, *i=1, …N*

$$\widetilde{w_i} = \begin{cases} w_i \text{ მონაწილეობს } c_j \text{კონცეპტის აღწერაში;} \\ \overline{w}_i \text{არ მონაწილეობს } c_j \text{კონცეპტის აღწერაში.} \end{cases}$$

შესაბამისად ცხრილი 1, მიიღებს უნიფიცირებულ სახეს:

ცხრილი 2

საცავი	$\mathbf{w_1}$	$\mathbf{w_2}$	…	$\mathbf{w_i}$	…	$\boldsymbol{w_N}$
R^1	$\widetilde{w}_{1,1}$	$\widetilde{w}_{2,1}$	…	$\widetilde{w}_{i,1}$	…	$\widetilde{w}_{N,1}$
R^2	$\widetilde{w}_{1,2}$	$\widetilde{w}_{2,2}$	…	$\widetilde{w}_{i,2}$	…	$\widetilde{w}_{N,2}$
…	…	…	…	…	…	…
R^k	$\widetilde{w}_{1,k}$	$\widetilde{w}_{2,k}$	…	$\widetilde{w}_{i,k}$	…	$\widetilde{w}_{N,k}$
…	…	…	…	…	…	…
R^m	$\widetilde{w}_{1,m}$	$\widetilde{w}_{2,m}$	…	$\widetilde{w}_{i,m}$	…	$\widetilde{w}_{N,m}$

სადაც

$$\widetilde{w_{i,k}} = \begin{cases} w_i \text{ მონაწილეობს } c_j \text{კონცეპტის აღწერაში} R^k \text{ საცავში;} \\ \overline{w}_i \text{არ მონაწილეობს } c_j \text{კონცეპტის აღწერაში} R^k \text{ საცავში.} \end{cases}$$

რადგან ყოველი c_j კონცეპტის აღწერა სასრულია, სასრულია $W = \{w_1, w_2, \dots, w_{max}\}$ სიმრავლეც, ჩვენ შეზღუდვის გარეშე შეგვიძლია ეს სივრცე ალ-სიმრავლედ წარმოვიდგინოთ [12], და მასში განვმარტოთ ყველა ის ოპერაციები, რაც განმარტებულია ასეთი ტიპის სიმრავლეებში. აქედან გამომდინარე c_j-ს ყოველი რეალიზაცია მოხერხებულია წარმოვადგინოთ ჩვეულებრივი იმპლიკანტის სახით. მაგალითად, წარმოვადგინოთ პირობითად რაღაც c კონცეპტის სხვადასხვა აღწერები. დავუშვათ გვაქვს 5 სხვადასხვა აღწერა, რომელშიც მონაწილეობას ღებულობს 4 განსხვავებული სიტყვა:

საცავი	C კონცეპტის *იმპლიკანტი*
R^1	$w_1 \& w_2 \& \overline{w}_3 \& w_4$
R^2	$w_1 \& \overline{w}_2 \& \overline{w}_3 \& w_4$
R^3	$w_1 \& w_2 \& \overline{w}_3 \& w_4$
R^4	$w_1 \& w_2 \& w_3 \& w_4$
R^5	$\overline{w}_1 \& w_2 \& w_3 \& w_4$

ჩავწეროთ ეს რეალიზაციები დიზიუნქციური ნორმალური ფორმის სახით:

$$(w_1 \& w_2 \& \overline{w}_3 \& w_4) \vee (w_1 \& \overline{w}_2 \& \overline{w}_3 \& w_4) \vee (w_1 \& w_2 \& \overline{w}_3 \& w_4) \vee (w_1 \& \overline{w}_2 \& \overline{w}_3 \& w_4) \vee (w_1 \& w_2 \& \overline{w}_3 \& w_4)$$

მოვახდინოთ ამ ფორმის მინიმიზაცია. შედეგად მივიღებთ c კონცეპტის აღწერის განზოგადოებულ სახეს, რომელიც ყველა საცავის ტექსტებს ეფუძნება.

$$w_1 \& \overline{w}_3 \& w_4$$

რა თქმა უნდა რეალურად კონცეპტის აღწერაში ბევრად უფრო დიდი რაოდენობის სიტყვები იღებენ მონაწილეობას, მაგრამ ჩვენ შეგვიძლია შევარჩიოთ ყველაზე მაღალი წონის შესაბამისი სიტყვები ESA (Explicit Semantic Analysis) მეთოდით მიღებული ვექტორზე დაყრდნობით. რაოდენობის განსაზღვრა შეიძლება დამოკიდებული იყოს ტექსტში სიტყვების რაოდენობის და განსხვავებული სიტყვების რაოდენობის თანაფარდობაზე. ასეთი მიდგომა გააძლიერებს საბოლოოდ მიღებული კონცეპტის აღწერის სემანტიკურ მნიშვნელობას. რაც უფრო მეტი სიტყვა მიიღებს მონაწილეობას აღწერაში, მით უფრო სემანტიკიურად ადექვატური იქნება შედეგი. მეორეს მხრივ სიტყვების დიდმა რაოდენობამ შესაძლოა გაართულოს კონცეპტის გამოყენება ინფორმაციის ძებნისათვის.

მეთოდის საცდელი შემოწმება

შემოთავაზებული მეთოდის ეფექტურობის შესამოწმებლად ჩატარდება სატესტო შემოწმება. ტესტირება მოიცავდა ორ საფეხურს:

1. ცნებების ფორმირება;
2. ძებნა ფორმირებული ცნების შესაბამისად.

სხვადასხვა საგნობრივი არის 5 ცნებისათვის ზემოთ მოყვანილი საცავებიდან შეირჩა ტექსტები - სულ 70 ტექსტი. ყოველი ცნებისათვის გამოიყო 10 ყველაზე მაღალწონიანი სიტყვა, რომლის საფუძველზე ყოველი აღმწერი ტექსტისათვის შეიქმნა იმპლიკანტი. ყოველი ცალკეული ცნებისთვის დამუშავდა 10-16 განსხვავებული აღმწერი ტექსტი. მოხდა ცნების ფორმირება ზემოთ აღწერილი მეთოდის შესაბამისად. მივიღეთ 5 ცნების განსხვავებული აღწერა ნორმალური დიზიუნქციური ფორმით.

ყოველი განსხვავებული კონცეპტის გამოყენებით განხორციელდა ძებნა 300 განსხვავებული ტექსტის შემცველ საცავში. აქ არ შედიოდა ის ტექსტები, რის საფუძველზეც მოხდა ცნებების ფორმირება. ყოველ ცნებას შეესაბამებოდა 42-65 ტექსტი. მიღებული ცნებების საფუძველზე ჩატარდა ძებნის პროცედურა ყოველი ცნებისათვის ცალ-ცალკე. ძებნის სიზუსტე მერყეობს 0.81 დან 0.92 მდე.

ზემოთ აღწერილი მეთოდი სქემატურად შეიძლება წარმოვადგინოთ ნახატზე ნაჩვენები ფორმით

დასკვნა

მეთოდის სატესტო შემოწმებამ აჩვენა, რომ შემოთავაზებული მეთოდით კონცეპტების შემუშავება განზოგადოებულად აღწერს მის სემანტიკუს არსს.ეს მეთოდი საშუალებას იძლევა კონცეპტის არასტრუქტურირებული აღმწერი მეტამონაცემების (ტექსტების) საფუძველზე მოხდეს განზოგადოებული სემანტიკურიარსის მქონე სტრუქტურის ფორმირება. ეს სტრუქტურა წარმოდგება, როგორც ინფორმაციის ძებნის ერთერთი ძირითადი კომპონენტი.

ამჟამად მიმდინარეობს კონცეპტების ფორმირების საბაზო ტექსტების რაოდენობის გაზრდა და აღმწერი სიტყვების ოპტიმალური რაოდენობის შერჩევა. რის საფუძველზეც მოხდება ძებნის ალგორითმის ოპტიმიზირება.

ლიტერატურა

1. GABRILOVICH, E. AND MARKOVITCH, S. 2007. Computing semantic related nessusing wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*. Morgan Kaufmann Publishers, 1606–1611.
2. Hunt, E.B.,Sternberg, R. (1990) Metaphors of Mind: Conceptions of the Nature of Intelligence. Cambridge: Cambridge University Press.
3. Hunt, E.B. (1962) *Concept Learning: An Information Processing Problem*. New York: Wiley. (Reprinted inRussian.)
4. D. Manning, Prabhakar Raghavan&Hinrich Schütze. Introduction to Information Retrieval. Website: http://informationretrieval.org/. Cambridge University Press .© 2008 Cambridge University Press
5. R. Davis and D. Lenat. *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill Advanced Computer Science Series, 1982.
6. Vygodsky, L.S., *Thought and Language*, MIT press, Massachusetts1986
7. Bolton, N., *Concept Formation*, Pergamon Press, Durhan, 1977
8. Egozi Concept-Based Information Retrieval using ExplicitSemantic Analysis. EGOZI, SHAUL MARKOVITCH, and EVGENIY GABRILOVICH. ACM Transactions on Information Systems, Vol. 0, No. 0, 2000, Pages 1-38.
9. Alexander Budanitsky and GraemeHirst. Evaluating wordnet-based measures of lexical semanticrelatedness. Computational Linguistics, 32(1):13–47, 2006.
10. 17. S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis.JASIS, 41(6):391–407, 1990.
11. Michael Strube, Simone Paolo Ponzetto Wikirelate! Computing semantic relatedness using Wikipedia . In AAAI'06 Boston, MA, 2006