ავტორის სტილი დაცულია

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი

მაია არჩუაძე

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი
კომპიუტერულ მეცნიერებათა მიმართულება

ქართულენოვანი ტექსტების კლასიფიკაციის ამოცანა ინფორმაციულ ძებნაში

ს ა დ ო ქ ტ ო რ ო დ ი ს ე რ ტ ა ც ი ა

ხელმძღვანელები:

სადოქტორო პროგრამის ხელმძღვანელი:
თსუ პროფესორი
ფიზ. მათ. მეცნ. დოქტორი გია სირბილაძე

სამეცნიერო ხელმძღვანელი:
თსუ პროფესორი
ტექ.მეცნ. კანდიდატი მანანა ხაჩიძე

თბილისი, 2017 წელი

ანოტაცია

ნაშრომში წარმოდგენილია არასტრუქტურირებული დოკუმენტების დაჭდევების ახალი მეთოდი, რომელიც გამოყენებულია ქართულენოვანი ტექსტების კლასიფიცირების პროცესის განსახორციელებლად. მეთოდი ეფუძნება ანალიტიკური ევრისტიკების მეთოდით, ცნების „პატერნების“ ცოდნის ბაზის ფორმირებას ტექსტების კლასიფიკაციისათვის და პირველად იქნა გამოყენებული ასეთი ტიპის ამოცანაში.

ინფორმაციის ძებნის პროცესი არ წარმოადგენს ერთგვაროვან ოპერაციას. მისი წარმატებულობა და რელევანტუროობა დამოკიდებულია ძებნის ციკლის ადეკვატურობაზე და სისრულეზე. ამ ციკლში ერთ-ერთი მნიშვნელოვანი ადგილი უკავია კლასიფიკაციის ეტაპს, რომლითაც, როგორც წესი, იწყება ძებნის პროცესი.

ძებნის სტრატეგიები, იდენტიფიცირებულია, როგორც ძებნის მოდელები. ინფორმაციული ძებნისთვის განხილულია ბულის მოდელი, ვექტორული სივრცის მოდელი და ალბათური ძებნის მოდელი, მათი მუშაობის თავისებურებანი და ის დადებითი და უარყოფითი თვისებები, რომლებიც ახასიათებს თითოეულ მათგანს. ძებნის მოდელები ეფუძნება ტერმინების წონის დათვლის პრინციპს. ტერმინს წონა არის სტატისტიკური სიდიდე, რომელიც განისაზღვრება დოკუმენტში ტერმინის შეხვედრის სიხშირით და განსაზღვრავს ტერმინის მნიშვნელოვნებას. შესაბამისად, ძებნის მოდელების ფუნქციონირების ჩარჩოებში, განხილულია ტერმინის წონის განსაზღვრის მეთოდები, რომელთა ნაწილი მუშავდება ალბათური ძებნის მოდელების საზღვრებში, ხოლო ნაწილი რეალიზდება ვექტორული სივრცის მოდელის ფარგლებში.

ამასთანავე, აღწერილია ინფორმაციული ძებნის ამოცანებში ბუნებრივი ენის ანალიზის მეთოდები, მათი ძირითადი ეტაპები და ის მნიშვნელოვანი თვისებები, რომელიც ახასიათებს თითოეულ მათგანს. ბუნებრივი ენის ანალიზი მოიცავს სინტაქსსა და სემანტიკაზე დაყრდნობით ცოდნის ამოღებას ბუნებრივ ენაზე დაწერილი დოკუმენტებიდან. ასეთი მიდგომა შეიძლება განვიხილოთ, როგორც „სემანტიკური“ მიდგომა იმ ლოგიკით, რომ დოკუმენტის შინაარსი და სტრუქტურა განისაზღვრება არასტატისტიკური მეთოდებით. დღეისათვის სტატისტიკური/ალბათური მეთოდებისა და სინტაქსური/სემანტიკური მეთოდების ინტეგრაცია იდეალური გამოსავალია ძებნის პროცესის ეფექტურობის გაზრდისათვის.

ნაშრომში განხილულია ტექსტის საწყისი დამუშავების პროცესები, რომელთა განხორციელება აუცილებელია კლასიფიკაციის საწყის ეტაპზე. განვიხილეთ სტემინგისა და ლემატიზაციის პროცესი, რომელიც წარმოადგენს დოკუმენტების დამუშავების უმნიშვნელოვანეს ეტაპს. საუბარია სტემინგის პოპულარულ ალგორითმებზე - ლოვინსის(Lowins), პორტერის (Porter) და პაის/ჰასკის (Pice/Hask) ალგორითმებზე.

განხილულია სტემინგის ალგორითმების გამოყენების თავისებურებანი ანალიზურ და სინთეზურ ენებში და, ამ თავისებურებათა გათვალისწინებით, მათი მოდიფიკაციის აუცილებლობა სხვადასხვა ენის კორპუსისათვის, თუმცა არსებობს ისეთი ენებიც, რომელთა დამუშავება მოითხოვს საერთოდ ახალი სტემერის შექმნას.

არსებული სტემინგის ალგორითმების გამოყენება ქართული ენის თავისებურებებიდან გამომდინარე შეუძლებელი გახდა, ამიტომ ქართულენოვანი ტექსტების კლასიფიკაციის ამოცანაში (მსგავსად სხვა ენებისა), ტექსტის დამუშავებისათვის შემუშავებულ იქნა სტემინგის ახალი ალგორითმი. იგი ეფუძნება სიტყვების და სუფიქსების ბაზას და ეფექტურად მუშაობს სიტყვის კვეცის პრობლემებზე.

განხილულია ბუნებრივი ენის დამუშავების მეთოდები კლასიფიკაციის ამოცანებში, კერძოდ, კონცეპტებზე დაფუძნებული ინფორმაციული ძებნა.

აღწერილია ინფორმაციული ძებნის ორი მნიშვნელოვანი პრობლემა: სინონიმია და პოლისემია, გადაჭრის სხვადასხვა მეთოდი და მათი ქმედითი ასპექტები. აღნიშნული მეთოდებიდან გამოვყავით ლატენტური სემანტიკური და ზუსტი სემანტიკური ანალიზის მეთოდები, როგორც საუკეთესო აღნიშნული პრობლემების გადასაჭრელად.

დღეისათვის კლასიფიკაციის ამოცანა შეიძლება განიხილოს, როგორც მანქანური სწავლებისა და ინფორმაციული ძებნის მეთოდების ერთობლიობა. ნაშრომში დავახასიათეთ კლასიფიკაციის ამოცანებში ყველაზე ხშირად გამოყენებადი სამი ალგორითმი, რომლებიც გამოყენებული იქნა ჩვენს მიერ განხორციელებულ კვლევებში. ესენია: უახლოესი მეზობლის ალგორითმი, მხარდამჭერი ვექტორების ალგორითმი და ბაიესის ალგორითმი. განვიხილეთ ყველა თავისებურება, რომელიც ახასიათებთ თითოეულს მუშაობის სხვადასხვა საფეხურზე.

სემანტიკური მიდგომის თავისებურებას უმთავრესად წარმოადგენს ის ფაქტი, რომ გამოიყენება დოკუმენტების კონცეპტუალური წარმოდგენა, რომელიც იქმნება საგნობრივი არის ცოდნის სემანტიკურ მოდელებზე დაყრდნობით, ხოლო ცოდნის წარმოდგენის არსებულ ინსტრუმენტებს შორის, ონტოლოგია წარმოადგენს ყველაზე გამოსახვით ხერხს. ჩვეულებრივ ონტოლოგიებში საგნობრივი არეების ცოდნა აღიწერება ცნებებისა და თვისებების იერარქიით, ასევე შეერთებული ცნებების ეგზემპლარების სემანტიკური ქსელებით. არსებულ მიდგომებთან შედარებით, ონტოლოგიის გამოყენებამ შესაძლოა მოგვცეს საშუალება გავაუმჯობესოთ ძებნის ხარისხი. ამიტომ მნიშვნელოვანია საგნობრივი არის აღმწერი ცოდნის ბაზის შემუშავების მოქნილი ალგორითმის შექმნა. ჩვენს მიდგომაში ეს ალგორითმი ეფუძნება ანალიტიკური ევრისტიკების მეთოდს.

ეს მეთოდი ეფუძნება აკად. ვლადიმერ ჭავჭანიძის მეთოდს, რომელიც ცნობილია კონცეპტების ფორმირების ანალიტიკური ევრისტიკების მეთოდის სახელით.

მეთოდი მოიცავს სხვადასხვა ეტაპებს: ნიშანთვისებათა ბინარიზაცია; ნიშანთვისებათა გადაკოდირება; ორთონორმირებული ბინარული მდგომარეობის ვექტორების აგება; ფილტრაციის ოპერაცია; დიზიუნქციური სუპერპოზიციის ოპერაცია; ბულის ცვლადებზე პირობითი გადასვლის ოპერაცია.

ანალიტიკური ევრისტიკების მეთოდის გამოყენებით, დოკუმენტების კოლექციიდან მოხდა კონკრეტული ცნების აღმწერი „კონცეპტ-პატერნების“ შემუშავება. პორტერის ალგორითმის მოდიფიცირებული ვარიანტით განხორციელდა კლასის აღმწერი ნიშან-თვისებების სიმრავლის გამოყოფა და შესაბამისად წონების დათვლა tf-idf სტანდარტული სქემით.

ამ ამოცანის განხორციელების შემდეგ, მოხდა მისი პრაქტიკული რეალიზაცია ქართულენოვანი სამედიცინო ჩანაწერების კლასიფიკაციისათვის. წარმოდგენილი იქნა სამდიცინო ჩანაწერების კლასიფიცირების მეთოდი ქართულენოვანი ტექსტებისათვის.

კვლევებისათვის გამოყენებული იქნა 24.855 ჩანაწერი. დოკუმენტების კლასიფიკაცია განხორციელდა სამ ძირითად ჯგუფად (ულტრასონოგრაფია, ენდოსკოპია, რენტგენი) და 13 ქვეჯგუფად. ამოცანის გადაწყვეტისათვის გამოყენებული იქნა ორი კარგად ცნობილი მანქანური სწავლების ალგორითმი: მხარდამჭერი ვექტორების (SVM) და უახლოესი მეზობლის (KNN). შედეგებმა აჩვენა რომ მანქანური სწავლების ორივე მეთოდი საკმაოდ შედეგინია, მაგრამ უკეთესი შედეგი გამოვლინდა SVM-ის გამოყენებისას. კლასიფიკაციის პროცესში ჩვენს მიერ შემუშავებული იქნა თვისებათა ამოკრების მეთოდის ჩვენი ვარიანტი, ე. წ. „ „შეკუმშვის“ მეთოდი, რომელმაც კლასიფიკაციის პირველ დონეზე საკმაოდ კარგი შედეგი მოგვცა. თუმცა მეორე დონეზე, რომელიც ქვეკლასებად კატეგორიას მოიცავდა შედეგები ცოტა გაუარესდა. დოკუმენტების 23%-ის მიკუთვნება ინდივიდუალური კლასებისათვის არ განხორციელდა. შედეგის გაუარესება გამოიწვია დაავადებათა აღწერაში გამოყენებული ტერმინების ერთგვაროვნებამ. ქვეკლასების მიხედვით კვლევის შედეგების დაფიქსირებისას მოხდა ერთიდაიგივე ტერმინების გამოყენება, რაც შედეგის შეცვლის მიზეზი გახდა.

აღსანიშნავია, რომ ეს არის ასეთი ტიპის ტექსტების კლასიფიკაციის პირველი მცდელობა. ზოგადად, ქართულენოვანი ტექსტებისათვის მსგავსი ამოცანა აქამდე არ განხორციელებულა.

Annotation

In the presented work the document "labeling" method for classification process is provided. The method is based on knowledge base formation using concept "patterns" for text classification.

Information retrieval process does not represent the outcome of only one type of operation. Its success and relevance is based on retrieval cycle recall and adequacy. One of the important parts of this cycle is the stage of classification- which represents the initial stage of text retrieval.

The strategies of retrieval are identified as the models of retrieval. The following basic retrieval models along with their peculiarities, pros and cons are considered: the Boolean model, Space Vector Model (SVM) and the probabilistic model. The models of retrieval are based on the principle of term weight calculation. The Weight of term represents the statistical value defined according to the frequency of its appearance in text and is defining the term value. Thus, in frames of retrieval model functionality, the methods of term frequency determination, partially in statistical retrieval models and partially in Vector Space models, is considered.

The methods on natural language processing, along their main stages and significant properties, for information retrieval task is described as well. The Natural Language Processing (NLP) contains the knowledge acquisition based on syntax and semantics of the provided natural language text. Such an approach can be considered as "semantic" based on logic that the content and structure of document is defined using non-statistical methods. Nowadays the integration of statistic/probabilistic models with syntactic/semantical models are considered to be the best solution for increasing the effectiveness of the retrieval process.

Work address the text initial Processing as a necessary part of text classification initial stage, particularly the process of Stemming and Lemmatization. The well-known and popular algorithms such as Lovins, Porter and Paice/Husk stemming algorithms are considered.

The peculiarities of stemming algorithm applications in analytical and Synthetic languages are provided and the necessity of their modification for different language Corps are underlined. However, there are languages requiring the development of a new stemmers for them.

In the work provided we considered natural language processing methods for classification task – namely the task of concept pattern based information retrieval.

We have described the two main problems of Information Retrieval: Polysemy and Synonyms along with their solution methods. From the methods considered we have selected the methods of Latent Sematic Analysis and Exact sematic analysis as the best suitable methods for the posed problem solution.

The problem of classification can be considered as a union of machine learning and IR methods. The following most common three algorithms in problems of classification: the K Nearest Neighbor (KNN), Support Vector Machine (SVM) and Bayes algorithms have been described, used later on in our research.

The semantic approach peculiarity is that the conceptual representation of document is applied, based on semantic models of subject area knowledge. From tools of knowledge representation the ontology represents the most suitable one. Generally in ontologies the subject area knowledge is described using hierarchy of notions and characteristics as well as joints notions semantic network entities. Application of Ontologies might lead to improvement of retrieval quality. This is why it is so important to develop an algorithm of subject area describing knowledgebase construction. In our case such algorithm is based on method of Heuristic Analytics.

The method considered is based on the method called "Pattern Formation Heuristic Analysis" developed by Academician Vladimer Chavchanidze.

The method consists of several stages: binarization of characteristics, re-coding of characteristics, orthonormal binary vector construction, operation of filtration, disjunctive superposition operation and Boolean variable transfer conditional operation.

Using the method of Heuristic Analysis, the certain notion defining concept-patterns were developed. Using the modified Porter algorithm, the class defining characteristics have been selected and appropriate weights have been calculated using the standard tf-idf scheme.

The practical realization of the task considered was fulfilled for medical data based document collection. The Georgian language based medical data classification method was presented.

The presented work introduces the instrument for Georgian-language-based medical records classification. It is the first attempt of classification of the Georgian-language-based medical records. Totally, 24.855 examination records were studied. The documents were classified into three main groups (Ultrasonography, Endoscopy, X-Ray) and 13 subgroups using two well-known methods: Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). The results obtained demonstrated that both machine learning methods performed successfully, with a little supremacy of SVM. In the process of classification a "shrink" method - based on features selection - was introduced and applied. At the first stage of classification the results of the "shrink" case were better, however on the second stage of classification into subclasses 23 % of all documents could not be linked to only one definite individual subclass (liver or binary system) due to common features characterizing these subclasses. The overall results of the study were successful.

We have to note that it was the first attempt of classification for such type of texts.

მადლიერება

უპირველეს ყოვლისა, მადლობას ვუხდი თსუ ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტის კომპიუტერულ მეცნიერებათა დეპარტამენტის პროფესორებს მხარდაჭერისათვის.

ჩემს სამეცნიერო ხელმძღვანელს, პროფესორ მანანა ხაჩიძეს იმ დიდი წვლილისთვის, რომელიც მას მიუძღვის ამ ნაშრომის წარმატებით დასრულებითვის და ასევე, გაწეული დახმარებისა და მხარდაჭერისათვის - ტექნიკური ინფორმატიკის დეპარტამენტის ასოცირებულ პროფესორს მაგდა ცინცაძეს, რომელიც მთელ სამუშაო პროცესში აქტიურად იყო ჩართული, დისერტაციაზე მუშაობის პერიოდში გვერდით მედგა და მიწევდა კონსულტაციებს. მადლიერი ვარ ჯგუფურ პროექტებში მონაწილე სტუდენტების, რომელთა აქტიური მუშაობის შედეგადაც განხორციელდა კვლევასთან დაკავშირებული ამოცანების პროგრამული რეალიზაცია. უღრმესი მადლობა მინდა გადავუხადო კომპიუტერულ მეცნიერებათა მიმართულების დეპარტამენტის ყველა თანამშრომელსა და კოლეგას, რომელთა მხრიდანაც ყოველთვის, განსაკუთრებით კი ახლა, ჩემთვის ამ უმნიშვნელოვანეს ეტაპზე, ვგრძნობდი მორალურ მხარდაჭერას.

დიდი მადლობა ამავე ფაკულტეტის სამეცნიერო კვლევებისა და განვითარების სამსახურის უფროსს, ქალბატონ რუსუდან ინწკირველს. გამხნევებისა და თანადგომისათვის უღრმესი მადლობა ჩემი ოჯახის წევრებსა და მეგობრებს.

შინაარსი

შესავალი

მონაცემთა ბაზის ცნება ამჟამად რამდენადმე გაფართოვდა, გამომდინარე თანამედროვე ტექნოლოგიების განვითარების ტენდენციებიდან. გაჩნდა ინტერნეტი, რომელიც წარმოადგენს უზარმაზარ ციფრულ ბიბლიოთეკას, რომელშიც ინფორმაციის განთავსება ხდება სხვადასხვა ფორმით (ტექსტური, ვიდეო, აუდიო და ა.შ) რაც ყველაზე მნიშვნელოვანია, ინფორმაცია განთავსებულია არასტრუქტურირებული სახით. ინფორმაციის ყველაზე დიდი წყარო სოციალური ქსელებია, რომლის საშუალებითაც ტერაბაიტების მოცულობის ინფორმაცია ყოველდღიურად განთავსდება ინტერნეტ სივრცეში. ამ პროცესმა წარმოშვა ინფორმაციის სიჭარბის პრობლემა, რომელიც ახლებურად წარმოაჩენს ინფორმაციის ძებნის პროცესს, მეტ მოთხოვნას უყენებს საძიებო სისტემებს და მოითხოვს ძებნის მეთოდების გაუმჯობესებას.

ინფორმაციის არასტრუქტურირებული ფორმით განთავსებამ გამოიწვია განსხვავებული მიდგომები ძებნის ამოცანების განხორციელების მხრივ. თუ მონაცემთა სტრუქტურირებული ბაზების დამუშავებისათვისაც SQL საკმარისი იყო, დღეს უკვე გვაქვს არასტრუქტურირებული მონაცემები Big Data და მათ დასამუშავებლად NoSQL. სტურქტურირებულ ინფორმაციაში ძებნის შედეგი ზუსტია, არასტრუქტურირებულ ბაზებში უკვე ჩნდება მოთხოვნის შედეგთან შესაბამისობის ხარისხი, რელევანტურობის სიდიდე. ბუნებრივია, მონაცემების სტრუქტურიზაცია გამოიწვევს ძებნის პროცესის გაუმჯობესებას, რაც, პირველ რიგში, აისახება შედეგის რელევანტურობის გაზრდაზე. ძებნის ოპერაციების შესრულებისას მნიშვნელოვანია კიდევ ერთი ფაქტორი: ძირითადად ინფორმაცია განთავსებულია ბუნებრივ სალაპარაკო ენაზე წარმოდგენილი ტექსტების „დოკუმენტების“ სახით და, შესაბამისად, ძებნის სისტემის მუშაობაში გათვალისწინებული უნდა იყოს ბუნებრივი ენის თავისებურებათა მინიმალურად აუცილებელი რაოდენობა.

კლასიფიკაციის ამოცანა ინფორმაციული ძებნის ქვეამოცანაა, რომლის შესრულებასაც სისტემა უზრუნველყოფს ძებნის საწყის ეტაპზე. მონაცემების კლასიფიკაცია ამარტივებს და, შესაბამისად, აუმჯობესებს ძებნის პროცესს, რადგან იგი მონაცემების (დოკუმენტების) სტრუქტურიზაციაზეა ორიენტირებული და ძებნის ოპერაციაც სტრუქტურირებული ბაზებისათვის უფრო მარტივია. კლასიფიკაციის პროცესის შესრულების სპეციფიკიდან გამომდინარე იგი იყენებს ყველა იმ მეთოდსა და მიდგომას, რასაც ინფორმაციული ძებნა.

პრობლემები

ისტორიულად არსებობს ინფორმაციული ძებნის ორი მიდგომა: სინტაქსური და სემანტიკური. სინტაქსური ძებნის დროს ძებნის სისტემები უზრუნველყოფენ ძებნას დოკუმენტიდან და მოთხოვნიდან ამოღებული სიტყვებით ან ფრაზებით, შესაბამისად, ასეთი ტიპის ძებნის სისტემების ფუნქციონირება ეფუძნება დოკუმენტისა და მოთხოვნის სინტაქსურ დამთხვევას. ამ პროცესზე უარყოფით გავლენას ახდენს პოლისემია და სინონიმია.

სემანტიკური ძებნა დაფუძნებულია დოკუმენტისა და მოთხოვნის სემანტიკურ შესაბამისობაზე, რომელიც ხორციელდება ბუნებრივი ენის ანალიზის მეთოდებით და განსაზღვრავს დაბრუნებული დოკუმენტების სემანტიკურ მსგავსებას მოთხოვნასთან.

სემანტიკური ძებნის მიდგომები ძირითადად უზრუნველყოფენ სინტაქსური ძებნის მიდგომების გაუმჯობესებას, თუმცა, რიგ შემთხვევებში, სინტაქსური ძებნის გამოყენება ცალსახადაა განსაზღვრული.

სემანტიკური შესაბამისობისათვის აუცილებელია მოთხოვნისა და დოკუმენტის სემანტიკური მნიშვნელობა იყოს ცნობილი. თუ მოთხოვნა განსაზღვრულია ფორმალურად, ყოველი ტერმინის სემანტიკა შეიძლება ცხადად განიმარტოს. თუ მოთხოვნა განსაზღვრულია არაფორმალურად, მაგალითად, ბუნებრივ ენაზე, მაშინ მოთხოვნის ყოველი ტერმინის სემანტიკა უნდა იყოს რაღაცნაირად გახსნილი. პრობლემა იმაშია, რამდენად შეძლებს მანქანა რომ გაიგოს, რომელი შინაარსობრივი მნიშვნელობა იგულისხმებოდა მოთხოვნაში, რათა მივიღოთ მომხმარებლისათვის აზრობრივად უფრო ახლოს მდგომი და საინტერესო დოკუმენტი.

როგორც ყველა ავტომატიზირებულ სისტემას, საძიებო სისტემასაც გააჩნია გარკვეული პრობლემები, რადგანაც მას უხდება ბუნებრივი ენის თავისებურებათა გათვალიწინება მუშაობის პროცესში. ბუნებრივი ენა აყენებს უამრავი სახის სირთულეს, რომელთა გადაჭრა ძალიან ძნელია თუ სიტყვის კონკრეტული მნიშვნელობა არაა ამოცნობილი. განუსაზღვრელობა მით უფრო ძნელად გადასაჭრელია, თუ სისტემა ტექსტის შინაარსის შემეცნებასთან ერთად, ვერ უზრუნველყოფს რეალური სამყაროს „ადამიანურ“ აღქმას. სემანტიკური ძებნის სისტემას შეუძლია რამოდენიმე ისეთი პრობლემის გადაჭრა, რომელიც ინფორმაციის ძებნისას საკმაოდ ხშირად იჩენს თავს:

- სინონიმების პრობლემა.

ხშირად ის, რაც ბუნებრივი ენისათვის „სიმდიდრედ“ ითვლება, ინფორმაციის ძებნის სისტემებს უამრავ პრობლემას უქმნის, მაგალითად, უამრავი სინონიმი. რაც უფრო კარგია „მწერალი“, იმდენი ერთი და იმავე შინაარსის მქონე ტექსტის „კარგად დაწერილი“ ვარიანტები არსებობს. შესაბამისად, საძიებო სისტემას უწევს „გარკვევა“, თუ როგორ გადმოგვცა ავტორმა თავისი აზრი. ამას ემატება ის ფაქტიც, რომ საქართველოს სხვადასხვა კუთხეში ერთი და იმავე ცნებისათვის ზოგჯერ სხვადასხვა ტერმინს იყენებენ. მაგალითად: ჯორკო და სკამი, ბაბუა და პაპა, კომში და ბია და ა.შ. ჩვენ გვჭირდება საძიებო სისტემა, რომელიც დაიჭერს “აზრს“ და არა “სიტყვას“.

- პოლისემია.

ქართულ ენაში, ისევე როგორც სხვა ბუნებრივ ენებში, უამრავ სიტყვას გააჩნია ერთზე მეტი მნიშვნელობა, ამასთან კონტექსტში მათ სხვადასხვა აზრი აქვთ, სემანტიკური ძებნის სისტემა შეძლებს მოთხოვნის დამუშავებას შესაბამისი კონტექსტიდან გამომდინარე, რაც პოლისემიის პრობლემატიკის გადაჭრის კარგ საშუალებად გვესახება.

გარდა ზოგადი პრობლემებისა, ცალკე აღსანიშნავია ქართულენოვანი დოკუმენტების ძებნის პრობლემები. მარტივი ანალიზი აჩვენებს, რომ ერთსა და იმავე საძიებო სისტემაში ქართულ და ინგლისურ ენაზე გაკეთებულ მოთხოვნაზე მიღებული შედეგების ხარისხი მკვეთრად განსხვავდება ერთმანეთისაგან. მიზეზად შეიძლება დასახელდეს ქართული ენის მორფოლოგიურ-სინტაქსური სირთულე, რაც არანაკლებ მნიშვნელოვანია; ასევე პრობლემაა ქართული ენის კორპუსების სიმცირე და ხელმისაწვდომობა, რომლებიც, რიგი საძიებო სისტემების მიერ, გამოიყენება ლექსიკონებისა და ონტოლოგიების შესაქმნელად არა მარტო ძებნის, არამედ მანქანური თარგმნის უზრუნველსაყოფად.

სემანტიკური ძებნის სისტემის გარეშე მიმდინარე პროცესებმა მეტ-ნაკლებად ამოწურეს თავინთი რესურსი, ტექსტების ბაზები სულ უფრო და უფრო დიდი ხდება - საძიებო სისტემებიც მეტ და უფრო დიდი მოცულობის დოკუმენტებს აბრუნებენ, ამასთან, ბულის ძებნის ტექნიკა საშუალებას იძლევა შევკვეცოთ ძიების შედეგად მიღებული შედეგები რეალურ დროში დამუშავებადი ზომის ინფორმაციამდე (1), თუმცა ამ დროს იკარგება საკმაოდ დიდი რაოდენობა პოტენციურად მნიშვნელოვანი დოკუმენტებისა, მეორე მხრივ, სტატისტიკური ძებნის მეთოდებით მიღებული შედეგები, რელევანტური რანჟირების მიუხედავად, უზომოდ დიდია. სემანტიკური ძებნა წარმოადგენს ახალ მიდგომას, რომელიც მოახდენს მოთხოვნის იმდაგვარ ინტერპრეტაციას, რომ წარმოაჩენს მოთხოვნილი ინფორმაციის სრულ შინაარსს და შეუსაბამებს მას სემანტიკური ძებნის ტექნოლოგიით აგებულ ცოდნის ბაზას.

სემანტიკური ძებნის სისტემებს შეუძლიათ უფრო სრულყონ სტატისტიკური ძიება და შედეგები მეტად ზუსტი გახადონ, ამასთან, ბულის ძიების განზოგადებასაც მოახდენენ, რაც ძებნის უკეთესი შედეგიანობით აისახება.

გაუმჯობესებას უზრუნველყოფს ბუნებრივი ენის ტექსტებზე დამყარებული ძებნის ჩვენეული ალგორითმი, რომელის მეშვეობით ხდება როგორც დოკუმენტის, ისე მოთხოვნის სემანტიკური ინტერპრეტაცია.

ნებისმიერი ძებნის სისტემისათვის ფუნქციონირების საწყისი ეტაპი არის კლასიფიკაციის პროცესი, რომლის გარეშეც ძებნა ფაქტიურად შეუძლებელია.

ნაშრომში წარმოდგენილია არასტრუქტურირებული დოკუმენტების დაჭდევების ახალი მეთოდი, რომელიც გამოყენებულია კლასიფიცირების პროცესის განსახორციელებლად. მეთოდი ეფუძნება ანალიტიკური ევრისტიკების მეთოდით, ცნების „პატერნების“ ცოდნის ბაზის ფორმირებას ტექსტების კლასიფიკაციისათვის.

სადისერტაციო ნაშრომი შედგება ექვსი თავისაგან:

პირველ თავში განხილულია ინფორმაციული ძებნის ამოცანები და მოდელები; მათი რეალიზაციის თავისებურებები და ის ძირითადი თვისებები, რომლებიც ახასიათებთ მათ ფუნქციონირების სხვადასხვა ეტაპზე.

მეორე თავში განხილულია კლასიფიკაცია, როგორც ინფორმაციული ძებნის ამოცანა. აღწერილია ის ძირითადი ეტაპები, რომლებიც აუცილებელია კლასიფიკაციის ამოცანის განხორციელებისათვის.

მესამე თავში განხილულია ბუნებრივი ენის დამუშავების მეთოდები კლასიფიკაციის ამოცანებში. აღწერილია ბუნებრივი ენის ანალიზზე დაფუძნებული სემანტიკური ძებნის პოპულარული ალგორითმები.

მეოთხე თავში განხილულია მანქანური სწავლების როლი ინფორმაციულ ძებნაში, აღწერილია მანქანური სწავლების ტიპები და ალგორითმები, რომლებიც ფართოდ გამოიყენება კლასიფიკაციის ამოცანებში.

მეხუთე თავში აღწერილია ანალიტიკური ევრისტიკების მეთოდი და კონცეპტის პატერნის ფორმირება კლასიფიკაციის ამოცანებისათვის.

მეექვსე თავში აღწერილია ტექსტების კლასიფიკაცია ცნების პატერნზე დაფუძნებული სისტემის გამოყენებით და შესაბამისად მეთოდის პრაქტიკული რეალიზაცია.

დისერტაციას ახლავს დანართი, სადაც მოცემულია ტექსტების კლასიფიკაციის აღწერილი მეთოდის პრაქტიკული რეალიზაციის შედეგები და შედარებითი ანალიზი.

დისერტაციის შედეგები გამოქვეყნებულია საერთაშორისო სამეცნიერო ჟურნალებში, გაკეთებულია პრეზენტაციები რესპუბლიკურ და საერთაშორისო სამეცნიერო კონფერენციებზე.

- **Manana khachidze, Magda Tsintsadze, Maia Mrchuadze**; *Natural Language Processing based Instrument For Classification of Free Text medical Record*- BioMed Research International ,Volume 2016 (2016), Article ID 8313454, 10 pages, http://dx.doi.org/10.1155/2016/8313454
- **Manana khachidze, Magda Tsintsadze, Maia Mrchuadze, Gela Besiashvili***: Concept Pattern Based Text Classification System Development for Georgian Text Based Information Retrieval*. Baltic J. Modern Computing, Vol. 3 (2015), No. 4, pp. 307–317
- **Manana khachidze, Magda Tsintsadze, Maia Mrchuadze, Gela Besiashvili** ;*Concept pattern formation in semantic search problem*. GESJ. Georgian Electronic Scientific Journal, Computer Science and Telecommunications. 2014, No 2(42) pp. 13-20 http://www.gesj.internetacademy.org.ge/en/list_artic_en.php?b_sec=comp&issue=2014-09
- **M.Khachidze, M.Tsintsadze, M.Archuadze, G.besiashvili** *Complex System State Generalized Presentation Based on Concepts*; Proceeding : 8th International Conference on Application of Information and Communication Technologies – AICT 2014. Kazakhstan, Astana. Pp. 559-569
- **M.khachidze; M.Tsintsadze; M.Archuadze; G.besiashvili**; *The Method of Concept Formation for Semantic Search*; Conference Proceeding :International Conference on APPLICATION of INFORMATION and COMMUNICATION TECHNOLOGIES (2013) Baku, Azerbaijan. Pp: 132-137.
- **M.Archaudze,D.Khachidze,N.Ninoshvili,N.Khachidze**; *Dental Self-diagnostic Information System Based on the Natural Language Processing.* eRA-11 International Scientific Conference. Greece, Pireus. 2016.
- **M.Khachidze,M.Tsintsadze,M.Archaudze,G.Besiashvili**; *Short Text Classification Application in Automated Workflow Management Systems.* eRA-11 International Scientific Conference. Greece, Pireus. 2016.

1. ინფორმაციული ძებნა

საუკუნეების მანძილზე ინფორმაციული ძებნის ამოცანის მნიშვნელობა დღითი დღე იზრდებოდა და 21-ე საუკუნეში, როდესაც ინტერნეტ სივრცე უზარმაზარ ციფრულ ბიბლიოთეკად გადაიქცა, ინფორმაციის ძებნის პროცესი ადამიანის ყოველდღიური ცხოვრების ერთ-ერთი მთავარი ნაწილი გახდა.

ინფორმაციის წერილობითი სახით შენახვის ტრადიცია უკავშირდება ქრისტეშობამდე 3000 წელს, შუმერების ეპოქას. ისინი სპეციალურად განკუთვნილ ადგილებში ინახავდნენ თიხის ფირფიტებს ლურსმნული წარწერით. შუმერები აცნობიერებდნენ, რომ წესრიგი და არქივების ხელმისაწვდომობა აუცილებელი იყო შრომისუნარინობის გაზრდისათვის. მათ მოიფიქრეს კლასიფიკაციის გზა ფირფიტებზე გაკეთებული წარწერების იდენტიფიკაციისათვის.

ინფორმაციის შენახვისა და მოძიების საჭიროება მკვეთრად გაიზარდა ისეთი გამოგონებების შედეგად, როგორიცაა ქაღალდი და ბეჭდური პრესა. კომპიუტერების გამოგონებიდან ცოტა ხნის შემდეგ, ანალიზის საფუძველზე, გასაგები გახდა, რომ კომპიუტერის გამოყენება დიდი ზომის ინფორმაციის შენახვის, დამუშავების და მექანიკურად მოპოვების შესაძლებლობას არსებითად გაზრდიდა. 1945 წელს ვანევარ ბუშმა(Vannevar Bush) გამოაქვეყნა ინოვაციური სტატია სახელწოდებით “As We May Think”, რომელმაც სათავე დაუდო იდეას ტექსტების არქივების ავტომატური ძებნის შესახებ (2). 50-იან წლებში ეს იდეა დაკონკრეტდა ტექსტის მექანიკური ძებნის მეთოდების აღწერით. ამ პერიოდში ჩატარდა ტექსტების მექანიკური მეთოდებით ძებნის სამეცნიერო, თეორიული და პრაქტიკული კვლევები. მათ შორის ერთ–ერთი ყველაზე მნიშვნელოვანი იყო 1957 ლუჰის(H.P.Luhn) მიერ შესრულებული სამუშაოები, სადაც მან ძებნის კრიტერიუმად აიღო ინდექსირებული სიტყვების და გადაფარვის არის (word overlap) ზომა (3).

რამდენიმე მნიშვნელოვანი ნაბიჯი გადაიდგა ამ სფეროში 60–იან წლებში. განსაკუთრებით აღსანიშნავია გერარდ სალტონის (Gerard Salton) და მისი სტუდენტების მიერ პირველად ჰარვარდის უნივერსიტეტში, ხოლო მოგვიანებით - კორნელის უნივერსიტეტში შექმნილი SMART სისტემა (4) , აგრეთვე კლევერდონისა (C.W.Clewerdon) და მისი გუნდის მიერ გრანფილდის აერონავტიკის კოლეჯში გაკეთებული ძებნის სისტემების შეფასების ტესტები (5). გრანფილდის კოლექცია - ეს არის კოლექცია, რომელიც გამოიყენება ელემენტარული საპილოტე ექსპერიმენტების დროს. შეიცავს 1398 სტატიის ანოტაციას აეროდინამიკის სფეროში, 225 მოთხოვნას და რელევანტურობის შეფასების კრიტერიუმებს მოთხოვნა-დოკუმენტი წყვილისათვის.

გრენფილდის ტესტებმა განავითარეს ინფორმაციის მოძიების მექანიზმებში შეფასების სისტემა. ისინი დღესაც გამოიყენება ინფორმაციული ძებნის სისტემებში, მეორე მხრივ, SMART სისტემამ შესაძლებელი გახადა მკვლევართათვის ძებნის ხარისხის გაუმჯობესება.

ექსპერიმენტების სისტემამ და შეფასების მეთოდოლოგიამ ხელი შეუწყო ამ სფეროს სწრაფ განვითარებას.

ტერმინი „ინფორმაციული ძებნა“ პირველად შემოიტანა კელვინ მურიმ (C.Mooers) 1948-1950 წლებში (6). ამის შემდეგ ეს იდეა თანდათან ვითარდება სხვადასხვა

სამეცნიერო სტატიებში და 1957 წელს უკვე გაჩნდა დოკუმენტების კოლექციაში სიტყვების (ცნებების) საშუალებით ძებნის იდეა (7).

მომდევნო 70-80-იანი წლების კვლევებმა განვითარება ჰპოვეს 60–იანი წლებში მიღწეული შედეგების საფუძველზე. განვითარებული იყო დოკუმენტების მოძიების სხვადასხვანაირი მოდელი და მეთოდი, რომელთა საშუალებითაც სამეცნიერო კვლევები ხორციელდებოდა ძებნის პროცესის ყველა ეტაპზე. ეს ახალი მოდელები (ხერხები) ეფექტური იყო იმ მცირე რაოდენობის ტექსტებთან სამუშაოდ (რამდენიმე ათასი ნაშრომი), რომლებიც მკვლევართათვის იმ დროისათვის იყო ხელმისაწვდომი. თუმცა, ტექსტების დიდი კოლექციების ნაკლებობის გამო, უპასუხოდ რჩებოდა კითხვა: გამოდგებოდა თუ არა ეს მეთოდები ტექსტების დიდ კორპუსზე სამუშაოდ.

არსებული მდგომარეობა შეიცვალა 1992 წელს **TREC**–ის **(Text Retrieval Evaluation conference)** წყალობით (8). შეიქმნა NIST-ის მიერ (U.S. National Instituteof Standart and Technology). იგი წარმოადგენს სტანდარტულ სატესტო კოლექციას, რომელიც შეიცავს 1,89 მილიონ დოკუმენტს და რელევანტურობის შეფასების კრიტერიუმებს 450 მოთხოვნისათვის.

TREC-ის საფუძველზე, რომელიც იძლევა დიდი ზომის ტექსტებთან ეფექტური მუშაობის საშუალებას, შეიცვალა არსებული მეთოდები და ამავე დროს ჩამოყალიბდა (ახლაც ყალიბდება) ბევრი ახალი. TREC-მა ინფორმაციული ძებნა ისეთ მნიშვნელოვან სფეროებად დაყო, როგორიცაა მონაცემების ფილტრაცია, კლასიფიკაცია და ა.შ. ინფორმაციული ძებნის სფეროში ჩამოყალიბებული ალგორითმები იყო პირველი ალგორითმები, რომელიც 1996-1998 წლებში გამოიყენეს World Wide Web-ში ძებნის პროცესისათვის (9), (1).

კლასიკური განსაზღვრის თანახმად, ინფორმაციული ძებნის მიზანია მომხმარებლისათვის „ინფორმაციული მოთხოვნილებების“ დაკმაყოფილება (10), რომელიც გულისხმობს არასტრუქტურირებული მონაცემთა ბაზიდან მომხმარებლის მოთხოვნის შესაბამისი ინფორმაციის დაბრუნებას (11). ინფორმაციულ ძებნას ეკუთვნის აგრეთვე ისეთი ამოცანებიც, რომლებიც არ ექვემდებარებიან ისეთ განსაზღვრებას, როგორიცაა ინფორმაციის წარმოდგენა, მონაცემების დამუშავება, ფილტრაცია, შენახვისა და მენეჯმენტის საკითხები (10). ძებნის პროცესის ამოცანა არის ინფორმაციული ძებნის ისეთი მეთოდების შემუშავება, რომელთა გამოყენებითაც მოხდება ადამიანის ინფორმაციული მოთხოვნილების შესაბამისი რელევანტური ინფორმაციის მიღება. ბუნებრივია, ძებნის შედეგად შესაძლებელია როგორც რელევანტური, ასევე არარელევანტური შედეგის დაბრუნება. ინფორმაციული ძებნის მიზანს წარმოადგენს სწორედ იდეალური კრიტერიუმების შერჩევა წარმატებული შედეგის მისაღწევად, რაც გულისხმობს ძებნის შედეგების მეტ სიზუსტეს და სისრულეს.

ინფორმაციული ძებნის სისტემები კლასიფიცირდება მათი მოქმედების მასშტაბების მიხდვით. ძირითადად გამოიყოფა სამი დონე:

- ძებნა ინტერნეტ სივრცეში, რომელიც წარმოადგენს ყველანაირი ტიპის მონაცემთა უნივერსალურ საცავს და მოითხოვს მუშაობას ინფორმაციის დიდი ზომის მასივებთან;
- პერსონალური ძებნა, რომელიც მოიცავს ელექტრონულ ფოსტაში „სპამების“ და წერილების ფილტრაციის, დაჭდევებისა და დახარისხების ამოცანებს.

– კორპორატიული ძებნა, რომელიც ორიენტირებულია საგნობრივი სფეროების მიხედვით და მუშაობს კორპორაციების შიდა ბაზებთან.

ინტერნეტში ინფორმაცია წარმოდგენილია სხვადასხვა ფორმით: ტექსტი, გრაფიკა, აუდიო, ვიდეო. ტექსტი არის ინფორმაციის წარმოდგენის ერთადერთი ფორმა, რომელიც მოითხოვს ფუნქციონალური დამუშავების სრულ პროცესს. ყველა დანარჩენი ტიპის მონაცემები განიხილება, როგორც მაღალი დონის ინფორმაციის წყაროები, რომელთა თავდაპირველი ძებნაც ხორციელდება ტექსტების საფუძველზე. ამიტომ ძებნის პროცესი, ძირითადად, ფოკუსირებულია ტექსტური ფორმით წარმოდგენილ ინფორმაციაზე, რაც გულისხმობს ძებნას დოკუმენტების კოლექციაში. ძებნის პროცესი მოიცავს აგრეთვე დოკუმენტების ფილტრაციის, მოდელირების, კლასიფიკაციის, კლასტერიზაციის, ამავე დროს საძიებო სისტემის არქიტექტურის დაპროექტების, სამომხმარებლო ინტერფეისის და მოთხოვნათა ენის განსაზღვრის ამოცანებს.

ინფორმაციის ძებნა ძირითადად ხორციელდება არასტრუქტურირებულ მონაცემებზე. ამ ტიპის მონაცემებს არ გაჩნიათ მკვეთრად გამოხატული, კომპიუტერზე ადვილად რეალიზებადი სტრუქტურა, არ არსებობს განსაზღვრული სინტაქსური კანონზომიერება, რითაც ძებნის სისტემას ექნება საშუალება მოძებნოს მოთხოვნის შესაბამისი სემანტიკის ჩანაწერები. არ არსებობს წესი, რომლის მიხედვითაც შესაძლებელია სინტაქსურად ერთნაირი ჩანაწერების ძებნა. თუმცა ზოგიერთ შემთხვევაში შესაძლებელია მოხდეს დოკუმენტებიდან სტრუქტურისა და სემანტიკის გამოყოფა. ამ გზით ხდება დოკუმენტების ბაზის ნახევრადსტრუქტურირება. მართალია, ამ შემთხვევაში არ არსებობს რელაციური ცხრილები კონკრეტული ველებით, მაგრამ შესაძლებელია საგასაღებო სიტყვების საშუალებით მოხდეს მათი წარმოდგენა სტრუქტურირებულ ცხრილებში. არსტრუქტურირებული დოკუმენტები ამავე დროს შეიცავს სტრუქტურირებულ ნაწილებსაც, მაგ. მეტამონაცემებს (ავტორის გვარსა და სახელს, შექმნის თარიღს და ა.შ.). დოკუმენტები ხშირად სტრუქტურირებულია სხვა გზითაც. მათ გააჩნიათ ,,დოკუმენტის“ სტრუქტურა, რაც გულისხმობს სარჩევს, წინასიტყვაობას, თავებად დაყოფილ ტექსტს, რომლებიც შეიცავს სურათებს, ცხრილებს, გრაფიკას. დოკუმენტის გარკვეულ ობიექტებს აქვთ კონკრეტული დასახელებები (მაგ. სურათი, ფიგურა, ცხრილი). პროგრამული უზრუნველყოფის ინსტრუმენტებით შესაძლებელია მოხდეს ამ სტრუქტურის ცალკეული ნაწილების ძებნა საგასაღებო სიტყვებით.

განსხვავებით არასტრუქტურირებული და ნახევრადსტრუქტურირებული ბაზებისაგან **სტრუქტურირებული ბაზა** შეიცავს გარკვეული სინტაქსური კანონზომიერებით ორგანიზებულ სახელდებულ კომპონენტებს, მაგ. რელაციურ მონაცემთა ბაზის ცხრილში შეიძლება იყოს ბევრი ჩანაწერი და ყველა სტრიქონს ჰქონდეს ერთი და იგივე ველები. გარდა ამისა ჩანაწერის თითოეულ ატრიბუტს გააჩნია განსაზღვრული მნიშვნელობა და იგი არის ერთი და იმავე სემანტიკის მატარებელი ყველა ჩანაწერში. ასეთ ბაზებთან მუშაობა უკვე შესაძლებელია მონაცემთა ბაზების მართვის არსებული სისტემებით (მბმს, DBMS), რომელიც ეძებს კონკრეტულ კომპონენტს სინტაქსის მიხედვით და აბრუნებს მოთხოვნის შესაბამის ზუსტ შედეგს.

ზოგადად არსებობს ორი ტიპის ძებნის სისტემა: ინფორმაციული ძებნის სისტემა და მონაცემთა ბაზების მართვის სისტემა. პირველი მუშაობს არასტრუქტურირებული

„ინფორმაციის“ მოძიებაზე, ხოლო მეორე - სტრუქტურირებულ მონაცემებზე. მათ შორის განსხვავებაა დაბრუნებული შედეგის სტატუსის განსაზღვრაშიც: ინფორმაციული ძებნის შედეგი შესაძლებელია იყოს მოთხოვნასთან მიმართებაში სრულად ან ნაწილობრივ რელევანტური მაშინ, როდესაც მბმს მომხმარებლის მოთხოვნაზე ზუსტ შედეგს აბრუნებს (1).

დღეისათვის, პრაქტიკული გამოყენებიდან გამომდინარე, ინფორმაციული ძებნის სისტემებისა და მონაცემთა ბაზების მართვის სისტემების ერთმანეთში ინტეგრაცია უმნიშვნელოვანესი ამოცანაა. კომერციული მონაცემთა ბაზების კომპანიებს უკვე აქვთ ერთმანეთში ინტეგრირებული ორივე ტიპის სისტემა. პირველი ასეთი სისტემა, რომელიც უკვე 15 წელია არსებობს, არის InQuira[1] იგი შეიქმნა ORACLE-ის მიერ. ეს სისტემა თეზაურუსის საფუძველზე ახორციელებს ბაზაში განთავსებული ინფორმაციის თემებად (კლასებად) სტრუქტურიზაციას.

1.1 ინფორმაციული ძებნის მოდელები

ძებნის სტრატეგიები იდენტიფიცირებულია, როგორც ძებნის მოდელები. არსებობს ინფორმაციული ძებნის სხვადასხვა მოდელი, რომლებიც ერთმანეთისაგან განსხვავდებიან დოკუმენტის და მოთხოვნის წარმოდგენის ფორმებით და მათი ერთმანეთთან მსგავსების განსაზღვრის მეთოდებით. ყველა მოდელი ფოკუსირებულია შემდეგ კომპონენტებზე:

- დოკუმენტი: დოკუმენტში იგულიხმება ობიექტი, რომელიც შეიცავს ინფორმაციას ფიქსირებულ ფორმატში. დოკუმენტი შეიძლება შეიცავდეს ბუნებრივ ან ფორმალურ ენებზე დაწერილ ტექსტებს, გრაფიკულ ობიექტებს, ხმოვან ინფორმაციას;
- მოთხოვნა: მოთხოვნა არის მომხმარებლის მიერ ფორმალიზებული სახით წარმოდგენილი ინფორმაციული საჭიროება. ამისათვის გამოიყენება მოთხოვნათა წარმოდგენის ენები;
- რანჟირების ფუნქცია, რომელიც განსაზღვრავს შესაბამისობის ხარისხს მოთხოვნასა და დაბრუნებულ დოკუმენტს შორის.

ძებნის კლასიკურ მოდელებს მიეკუთვნება: ბულის მოდელი, ვექტორული სივრცის მოდელი და ალბათური მოდელი.

1.1.1. ბულის მოდელი

ბულის ძებნის მოდელი - ეფუძნება ბულის ლოგიკას. იგი ტერმინების შერჩევისათვის იყენებს „სიტყვების ჩანთის“ (Bag of words) პრინციპს. ამ მოდელში მოთხოვნა ფორმირდება ტერმინებით და ბულის ცნობილი ოპერატორებით: AND, OR, NOT. ბულის მოდელის დადებითი მხარე მისი იმპლიმენტაციის სიმარტივეა (12). ბულის ძებნა „ზუსტი“ ძებნაა, რაც გულისხმობს დაბრუნებული შედეგის „ზუსტ“ დამთხვევას მოთხოვნაში წარმოდგენილ პირობასთან. თუმცა რეალურად შესაძლებელია მეტად ან ნაკლებად რელევანტური დოკუმენტების არსებობაც, მაგრამ ასეთი შედეგის მიღება ამ მოდელით ვერ ხერხდება, რაც მის ერთ-ერთ ნაკლად ითვლება. მოდელს აქვს სხვა

[1] http://www.oracle.com/us/corporate/Acquisitions/inquira/index.html

ნაკლიც - რანჟირების პროცესი არ არსებობს. ძებნის შედეგად დაბრუნებული დოკუმენტები არის დალაგებული სხვადასხვა პარამეტრით: მაგ. თარიღით და არა მოთხოვნასთან მსგავსების სიდიდით, ანუ არ არსებობს რანჟირებული სია. ამიტომ ამ მოდელის ეფექტურობა უფრო დაბალია, ვიდრე სხვა ძებნის მოდელების, რომლებიც ორიენტირებულია რანჟირებულ სიაზე. ასეთ მოდელებს მიეკუთვნება ვექტორული სივრცისა და ალბათური ძებნის მოდელები.

ბულის მოდელის პრობლემაა სინონიმების არსებობაც. მისთვის მოთხოვნაში წარმოდგენილი ტერმინების შესაბამისი სინონიმების არსებობა დოკუმენტში არ აღიქმება. თუმცა მარტივ დონეზე ეს პრობლემა მოდელში გადაწყვეტილია OR ოპერატორით. პრობლემას წარმოდგენს, აგრეთვე, მოთხოვნის ბულის ოპერატორებით წარმოდგენის აუცილებლობაც, რომელიც მოითხოვს ამ სფეროში სპეციფიკურ ცოდნას, რაც მომხმარებელთა უმეტეს ნაწილს არ გააჩნია (13). მისი გამოყენება უფრო ეფექტურია ძებნის ბოლო ეტაპზე, როდესაც საჭიროა შედეგის უფრო მეტად დაზუსტება. ბულის მოდელი უფრო მონაცემების ძებნაზეა ორიენტირებული, ვიდრე ინფორმაციულ ძებნაზე.

1.1.2. ვექტორული სივრცის მოდელი

ბულის მეთოდის პრობლემები ნაწილობრივ გადაიჭრა ძებნის ისეთი სტატისტიკური მოდელებით, როგორიცაა ვექტორული სივრცისა და ალბათური ძებნის მოდელები. ორივეს ერთი მიზანი აქვს - დაბრუნებული დოკუმენტების რანჟირებული სია, თუმცა ამ მიზნის მიღწევის გზები განსხვავებულია. ორივე მათგანი მოთხოვნასთან მიმართებაში დოკუმენტის რელევანტობის დასადგენად იყენებს ტერმინების სიხშირის შესაბამის სტატისტიკურ ინფორმაციას, თუმცა განსხვავებული მიდგომებით. ამიტომ პრობლემებიც, რომლებიც სტატისტიკურ მეთოდებშია, განსხვავებულია.

ვექტორული სივრცის მოდელი წარმოადგენს ინფორმაციული ძებნის მეთოდების ფუნდამენტს და საკამოდ ეფექტურად გამოიყენება კლასიფიკაციის ამოცანებში (14). ვექტორული სივრცის მოდელში მოთხოვნაც და დოკუმენტიც წარმოიდგინება ტერმინების ვექტორების სახით. იგი მოიცავს სამ ეტაპს: ინდექსაციის პროცესი, რომელიც გულისხმობს ტექსტიდან ტერმინების ამოღებას, შემდეგ ამ ტერმინებისათვის წონების მინიჭება და ბოლო ეტაპი არის წონებით წარმოდგენილი დოკუმენტისა და მოთხოვნის ვექტორებს შორის მსგავსების სიდიდის განსაზღვრა. ამ მოდელში ყველაზე მნიშვნელოვანი პარამეტრია ტერმინის წონა, რომელიც დამოკიდებულია დოკუმენტში ტერმინის შეხვედრის სიხშირეზე. დოკუმენტიდან ამოღებული ტერმინების სიმრავლეების გაერთიანებით მიიღება ტერმინთა სივრცე, რომელშიც თითოეული განსხვავებული ტერმინი ამ სივრცის განზომილებას წარმოდგენს. კოლექციაში ძებნის დროს მიიღება მაღალგანზომილებადი ვექტორული სივრცე - „დოკუმენტების სივრცე“ რომელშიც ტერმინების დადგენილი წონა განიხილება, როგორც დოკუმენტის კოორდინატები ამ სივრცეში (9).

ანალოგიურად, წონითი ვექტორების პრინციპით ხდება მოთხოვნის წარმოდგენაც. ძებნის შედეგის მისაღებად გამოითვლება „დოკუმენტისა“ და „მოთხოვნის“ ვექტორებს შორის მსგავსების ზომა, რომელიც ამ ორი ვექტორის სკალარული ნამრავლის ტოლია.

$D = (d_{i1}, d_{i2,} \dots d_{ij})$ ვექტორის სახით წარმოდგენილი t განსხვავებული ტერმინის შემცველი დოკუმეტების მსგავსების სიდიდე, ტერმინის w წონებით ჩაწერილ $Q = (w_{q1}, w_{q2,} \dots w_{qt})$ მოთხოვნის ვექტორთან გამოითვლება შემდეგი ფორმულით (15):

$$score(Q, D) = \sum_{j=1}^{t} w_{qj} \times d_{ij} \quad (1)$$

დოკუმენტის და მოთხოვნის ვექტორებს შორის მსგავსების განსაზღვრისათვის ეფექტურად გამოიყენება ორ ვექტორს შორის მსგავსების კოსინუსური ზომა. ბუნებრივია, მსგავსების განსაზღვრისას ორ ვექტორს შორის სხვაობა არაეფექტურია, რადგან ორი ერთნაირი დოკუმენტის შემთხვევაში სხვაობა შესაძლებელია იყოს დიდი, თუ ერთ დოკუმენტში სიტყვების რაოდენობა საკმაოდ დიდია მეორესთან შედარებით. ამიტომ დოკუმენტის სიგრძის გავლენის კომპენსირებისათვის შემოღებულ იქნა ორ ვექტორს შორის კუთხის კოსინუსის ზომა.

$$SQ(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj}\, d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}} \quad (2)$$

არსებობს მსგავსების განსაზღვრის სხვა მეთოდებიც (16):

- Dice კოეფიციენტი:

$$SQ(Q, D_i) = \frac{2 \sum_{j=1}^{t} w_{qj}\, d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2} \quad (3)$$

- Jaccard კოეფიციენტი:

$$SQ(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj}\, d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 + \sum_{j=1}^{t} (w_{qj})^2 - \sum_{j=1}^{t} w_{qj}\, d_{ij}} \quad (4)$$

ვექტორული სივრცის მოდელის ერთ-ერთი უპირატესობა, ბულის მოდელთან შედარებით, რანჟირებული შედეგია, ხოლო ნაკლად განიხილება შემდეგი ფაქტორები:

- მოდელში არ განიხილება NOT ოპერატორის შედეგები, რაც იმას ნიშნავს, რომ ძებნისას არ მიიღება მხედველობაში ის ტერმინები, რომელთა არ არსებობაც შესაძლებელია განმსაზღვრელი ფაქტორი იყოს ძებნის შედეგისათვის;
- ტერმინები განიხილება ერთმანეთისაგან დამოუკიდებლად;
- მოდელი მოითხოვს ინტენსიურ გამოთვლებს.

1.1.3. ალბათური ძებნის მოდელი

ვექტორული სივრცის მოდელთან ერთად, სტატისტიკური ძებნის მოდელს მიეკუთვნება ალბათური ძებნის მოდელი, რომელიც პირველად 60-იან წლებში გაჩნდა (17). ეს მოდელი ეფუძნება მონაცემების ალბათური რანჟირების პრინციპს (18), რომლის

თანახმადაც საძიებო სისტემის მიერ შედეგის ყველაზე მაღალი ეფექტურობა მიიღწევა იმ შემთხვევაში, თუ მოხდება ძებნის შედეგად დაბრუნებული დოკუმენტების რანჟირება მათი მოთხოვნასთან შესაბამისობის ალბათობის შემცირების მიხედვით. იგი აფასებს რელევანტურობის სიდიდეს წყვილისათვის „დოკუმენტი-მოთხოვნა“ და ეძებს მათ შორის საუკეთესო შესაბამისობას. დოკუმენტის რელევანტურობა ფასდება ბაიესის თეორემის საფუძველზე (19):

$$P(R|D = \frac{P(D|R)}{P(D)}P(R) \quad (5)$$

$P(R|D)$ - დაბრუნებული დოკუმენტების რანჟირების ალბათობის სიდიდე;

$P(D|R)$- D რელევანტური დოკუმენტის ალბათობა

$P(R)$ - აპრიორული ალბათობა იმისა, რომ D კოლექციიდან შემთხვევით აღებული დოკუმენტი არის რელევანტური.

$P(D)$-პირობითი ალბათობა იმისა, რომ k ტერმინის მქონე დოკუმენტი გამოჩნდება D კოლექციაში.

ისევე, როგორც ძებნის სხვა მოდელებს, ამ მოდელსაც აქვს გარკვეული ნაკლოვანებები:

- მოდელი მოითხოვს წინასწარ მიღებულ გადაწყვეტილებათა ცოდნის ბაზას;
- მოითხოვს ალბათობების შეფასებას;
- ემყარება ვარაუდებს;
- უგულვებელყოფს ბულის ლოგიკას.

არსებობს დადებითი და უარყოფით თვისებები, რომლებიც ერთნაირად ახასიათებს ორივე სტატისტიკურ მოდელს:

მეთოდების დადებით მხარეები:

- რანჟირებული შედეგი და მომხმარებლისათვის რანჟირების ზღვარის განსაზღვრის შესაძლებლობა;
- მოთხოვნის მარტივი ფორმულირება, რომელიც არ მოითხოვს მოთხოვნების განსაზღვრის ენების ცოდნას და, შესაბამისად, შესაძლებელია მისი ჩამოყალიბება ბუნებივი ენის ტერმინებით;
- მოთხოვნების წარმოდგენის მრავალფეროვნება.

უარყოფითი მხარეები:

- გარკვეული შეზღუდვები სტატისტიკური სიდიდეების განსაზღვრისას;
- გამოთვლების სირთულე;
- სისტემის ეფექტურობის გაზრდისათვის საჭირო ტერმინების დიდი რაოდენობა.

ბუნებრივია, ეს ფაქტორები გავლენას ახდენს ძებნის სისტემის მუშაობის ეფექტურობაზე, რომელიც გარკვეული კრიტერიუმებით ფასდება.

1.2. ძებნის სისტემის შეფასება

ძებნის სისტემის ობიექტური შეფასების სიდიდეების განსაზღვრა, მრავალი სამეცნიერო ექსპერიმენტის კვლევის სფეროს წარმოდგენს. გრანფილდის ტესტების საფუძველზე 1960 წელს გამოქვეყნდა თვისებები, რომელთა გამოყენებაც შესაძლებელი იყო ძებნის სისტემის შეფასებისათვის. მიუხედავად ამისა, მრავალი წლის განმავლობაში ამ თემაზე კვლავ მიმდინარეობდა კვლევები და ექსპერიმენტები. საბოლოოდ, სამეცნიერო საზოგადოებამ მიიღო შეფასების ორი ძირითადი სიდიდე: სისრულე (Recall) და სიზუსტე (Precision) (20).

სისრულე - ძებნის შედეგად დაბრუნებული დოკუმენტების წილი მთლიანად მოძებნილ რელევანტური დოკუმენტების რაოდენობასთან მიმართებაში.

$$\textit{სისრულე} = \frac{\textit{დაბრუნებული რელევანტური დოკუმენტების რაოდენობა}}{\textit{რელევანტური დოკუმენტების სრული რაოდენობა}}$$

სიზუსტე - რელევანტური დოკუმენტების წილი მთლიანად მოძებნილ დოკუმენტებში.

$$\textit{სიზუსტე} = \frac{\textit{დაბრუნებული რელევანტური დოკუმენტების რაოდენობა}}{\textit{მთლიანად დაბრუნებული დოკუმენტების რაოდენობა}}$$

ყველაზე კარგ შემთხვევაში, კარგმა ძებნის სიტემამ უნდა უზრუნველყოს ბევრი რელევანტური დოკუმენტის დაბრუნება, რაც ნიშნავს მაღალი სიდიდის „სისრულეს“, ამავე დროს დაბრუნებულ დოკუმენტებში უნდა იყოს მცირე რაოდენობის არარელევანტური დოკუმენტები, რაც ნიშნავს მაღალი სიდიდის „სიზუსტეს“, მაგრამ ეს ორი სიდიდე ერთმანეთთან სრულ წინააღმდეგობაში არის: მეთოდები, რომლებიც აუმჯობესებენ ძებნის სიზუსტეს, ამცირებენ სისრულეს და პირიქით. ნებისმიერ სიტუაციაში ორივე მათგანი დამოკიდებულია ტექსტის შინაარსზე.

დოკუმენტები	რელევანტური	არარელევანტური
დაბრუნებული	TP	FP
არდაბრუნებული	FN	TN

$$PRECISION = \frac{TP}{TP + FP} \qquad (6)$$

$$RECALL = \frac{TP}{TP + FN} \qquad (7)$$

$TP(True\ Positive)$-დაბრუნებული რელევანტური დოკუმენტების რაოდენობა;

$FP\ (False\ Positive)$ –დაბრუნებული არარელევანტური დოკუმენტების რაოდენობა;

$FN\ (False\ Negative)$–არდაბრუნებული რელევანტური დოკუმენტების რაოდენობა;

$TN\ (True\ Negative)$ –არდაბრუნებული არარელევანტური დოკუმენტების რაოდენობა.

არსებობს კიდევ კლასიფიკაციის სისტემის ეფექტურობის შეფასების ორი სტანდარტული მეთოდი: მიკრო-გასაშუალება (micro-averaging) და მაკრო-გასაშუალება (macro-averaiging). ორივე მათგანი გამოითვლება გლობალურ ამოცანებში (მაგ. კატეგორიზაცია).

$$P_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \quad (8)$$

P_{micro} *(მიკროსიზუსტე)*- ყველა კატეგორიის სიზუსტეების ჯამი(6).

$$R_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \quad (9)$$

R_{micro}*(მიკროსისრულე)*-ყველა კატეგორიის სისრულეების ჯამი (7).

$$P_{macro} = \frac{1}{|C|}\frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \quad (10)$$

P_{macro}*(მაკროსიზუსტე) ყველა კატეგორიის სიზუსტეების ჯამის საშუალო (6).*

$$R_{macro} = \frac{1}{|C|}\frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \quad (11)$$

R_{micro}*(მაკროსისრულე)*-ყველა კატეგორიის სისრულეების ჯამის საშუალო (7).

$i = 1, \dots, |C|$ $(C - დოკუმენტების კატეგორია)$

არაერთ სიტუაციაში აღმოჩნდება ხოლმე, რომ ერთი უფრო მნიშვნელოვანია, ვიდრე მეორე. მაგ. ინტერნეტ სივრცეში ინფორმაციის ძებნისას მომხმარებლისათვის მნიშვნელოვანია სიზუსტე, ხოლო ინფორმაციის დისკზე ძებნის შემთხვევაში - სისრულე. თუ გაიზრდება კოლექციაში დოკუმენტების რაოდენობა და, შესაბამისად, მოძებნილი დოკუმენტების რიცხვი, ეს არ გამოიწვევს სისრულის შემცირებას, ამ შემთხვევაში იკლებს სიზუსტე (21).

არსებობს მეთოდები, რომლებიც ამ ორი სიდიდის საფუძველზე ითვლიან გაზომვის სიზუსტეს (accuracy)-ს. ეს სიდიდე განსაზღვრავს გაზომვის შედეგად მიღებული სიდიდის სიახლოვეს მის სტანდარტულ ან უკვე ცნობილ სიდიდესთან.

$$Acc_i = \frac{TP_i + NT_i}{TP_i + FP_i + FN_i + TN_i} \tag{12}$$

ფორმულაში მრიცხველი წარმოადგენს იმ დოკუმენტების რაოდენობას, რომლებისთვისაც კლასიფიკატორმა მიიღო სწორი გადაწყვეტილება, ხოლო მნიშვნელი არის დასასწავლ კატეგორიაში დოკუმენტების საერთო რაოდენობა.

რაც შეეხება სიდიდეს $Err(Error\ Rate)$ იგი არის არარელევანტური დოკუმენტების რიცხვის ფარდობა დოკუმენტების საერთო რაოდენობასთან. საუკეთესო მნიშვნელობა მისი არის 0.0, ხოლო ყველაზე უარესი 1.0;

$$Err_i = \frac{FP_i + FN_i}{TP_i + FP_i + FN_i + TN_i} \tag{13}$$

სისრულესა და სიზუსტეს შორის დამოკიდებულების დასაბალანსებლად გამოიყენება ბალანსირებული F ზომა (9), (22)

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{14}$$

β არის წონითი ფაქტორი სიზუტისა და სისრულის განსაზღვრისათვის (1). $\beta \in [0, \infty)$ თუ β იღებს მნიშვნელობას დიაპაზონში $0 < \beta < 1$, მაშინ პრიორიტეტი აქვს სიზუსტეს, ხოლო თუ $\beta > 1$, მაშინ უპირატესობა ენიჭება სისრულეს, ხოლო $\beta = 1$ შემთხვევაში გვაქვს ბალანსირებული ზომა და ფორმულა იღებს სახეს :

$$F = \frac{2PR}{P + R} \tag{15}$$

ზირითადად, ძებნის სისტემის შედეგების შეფასება ხდება ორ სიდიდეზე დაყრდნობით: სიზუსტე და სისრულე.

1.3. ტერმინის წონა

ძებნის მოდელები ეფუძნება ტერმინების წონის დათვლის პრინციპს. ტერმინს წონა არის სტატისტიკური სიდიდე, რომელიც განისაზღვრება დოკუმენტში ტერმინის შეხვედრის სიხშირით და განსაზღვრავს ტერმინის მნიშვნელოვნებას.

ტერმინის წონის დათვლის განსხვავებული მეთოდები არსებობს: მათი ნაწილი მუშავდება ალბათური ძებნის მოდელების საზღვრებში, ხოლო ნაწილი რეალიზდება ვექტორული სივრცის მოდელის ფარგლებში. ალბათურ მოდელებზე დაფუძნებული წონები, ეფუძნებიან სხვადასხვა ალბათობების შეფასებებს (23), ხოლო ვექტორული

სივრცის შემთხვევაში წონების შეფასება ხდება სამეცნიერო კვლევებისა და ექსპერიმენტების საფუძველზე (24).

ნებისმიერ მოდელში წონების დათვლა ეფუძნება სამ ძირითად ფაქტორს: ტერმინის სიხშირე (tf), დოკუმეტების შებრუნებული სიხშირე (idf) და დოკუმენტების სიგრძე.

ტერმინის სიხშირე- ტერმინები, რომლებიც ტექსტში ბევრჯერ მეორდება, ითვლებიან განსაკუთრებულად. დოკუმენტში მათი შეხვედრის სიხშირე არის ამ ტერმინების წონა. ტერმინის სიხშირეზე დამოკიდებული წონა პირველად გამოყენებული იქნა ვექტორული სივრცის მოდელში 1960 წელს.

ტერმინების წონის დათვლის სტანდარტული სქემაა - ტერმინის სიხშირე $tf_{t,d}$. ამ სქემის მიხედვით ტერმინის წონა დოკუმენტში მისი გამოჩენის სიხშირის ტოლია. თუმცა, შესაძლებელია, დოკუმენტის წარმოდგენა სხვადასხვა წონითი ფუნქციით განსაზღვრული წონების სიმრავლით. ასეთი სახით წარმოდგენილი დოკუმენტის მოდელი არის BOW მოდელი, რომელშიც ტერმინების თანმიმდევრობა უგულებელყოფილია და ძირითადი დატვირთვა მოდის შეხვედრის რაოდენობაზე.

დოკუმენტების შებრუნებული სიხშირე - დოკუმენტში ტერმინის მნიშვნელოვნება პროპორციულად იზრდება მისი შეხვედრის სიხშირის ზრდასთან ერთად, მაგრამ არის ტერმინები, რომლებიც ბევრ დოკუმენტში გვხვდებიან და ითვლებიან საერთო ტერმინებად, ამიტომ ისინი ნაკლებად განსაზღვრავენ დოკუმენტის შინაარსს და შესაბამისად მათი გავლენა კოლექციაში დოკუმენტის რანჟირების განსაზღვრისათვის მცირდება.

შებრუნებული სიხშირის შემოტანა დაკავშირებულია იმ ფაქტთან, რომ ტერმინის სიხშირის განსაზღვრას აქვს სერიოზული პრობლემა: მოთხოვნის შესაბამისი დოკუმენტის რანჟირებისას ყველა ტერმინი ითვლება ერთნაირად მნიშვნელოვნად. სინამდვილეში არის ტერმინები, რომლებიც მეტად უმნიშვნელოა რელევანტურობის განსაზღვრისათვის. ძებნის (მოთხოვნის დამუშავების) პროცესში, ამ პრობლემის თავიდან ასაცილებლად, შემოდის იმ ტერმინის „გავლენის“ შემცირების მექანიზმი, რომელიც ყველაზე ხშირად გვხვდება კოლექციაში. ბუნებრივია, ასეთი ტერმინი ვერ მოახდენს გავლენას დოკუმენტის რელევანტურობის შეფასებისას. მაგალითად, თუ განვიხილავთ დოკუმენტების კოლექციას „კომპიუტერული მეცნიერება“ ან „ საინფორმაციო ტექნილოგიები“, ტერმინი „კომპიუტერი“ გვხვდება თითქმის ყველა დოკუმენტში, ამიტომ ის არ შეიძლება იყოს რელევანტურობის განმსაზღვრელი, მაგრამ არის საუკეთესო ტერმინი, რომელიც განასხვავებს დოკუმენტებს „კოპიუტერული მეცნიერებების“ შესახებ ისეთი დოკუმენტებისაგან, რომლების ამ სფეროს საერთოდ არ ეკუთვნის. ამიტომ გაჩნდა იდეა, რომ კოლექციაში მაღალი წონის ტერმინები შემცირდეს რაღაც კოეფიციენტით. აქედან გამომდინარე, შემოვიდა სიდიდე ,,დოკუმენტების შებრუნებული სიხშირე“. იგი იმ დოკუმენტების რაოდენობაა, რომლებიც შეიცავენ ტერმინს. წონითი მეთოდები, რომლებიც ამ პრინციპზეა დაფუძნებული დაკავშირებულია დოკუმენტის შებრუნებულ სიხშირესთან (IDF). პირველად იგი გამოყენებულ იქნა 1972 წელს (25).

დოკუმენტების შებრუნებული სიხშირე გამოითვლება ფორმულით:

$$idf_t = log \frac{N}{df_t} \quad (16)$$

დოკუმენტების სიგრძე- როდესაც კოლექციაში გვაქვს ცვალებადი სიგრძის დიკუმენტები, მათგან დიდი მოცულობის დოკუმენტებისათვის ძებნის ეფექტურობის შეფასების სიდიდე უფრო მაღალია, რადგან ისინი შეიცავენ ბევრ ტერმინს, რაც განაპირობებს ტერმინების სიხშირის გაზრდას.

დოკუმენტში თითოეულ ტერმინს ენიჭება $tf_{t,d} - idf_{t,d}$ წონითი სქემით განსაზღვრული წონა, რომელიც სტანდარტული სქემით გამოისახება:

$$tf_{t,d} - idf_t = tf_{t,d} \times idf_t \quad (17)$$

აქედან გამომდინარე, ტერმინის წონა:

- არის მაქსიმალური, თუ იგი გვხვდება ბევრჯერ მცირე რაოდენობის დოკუმენტებში;
- არის მინიმალური, თუ იგი გვხვდება კოლექციის ყველა დოკუმენტში.

ნათელია, რომ რაც უფრო ერთგვაროვანია ტერმინის განაწილება, მით ნაკლებად სპეციფიურია ის დოკუმეტისათვის.

მას შემდეგ, რაც შეიქმნა პირველი TREC კოლექცია, კორნელის უნივერსიტეტის მკვლევარების მიერ რეალიზებულ სამეცნიერო კვლევებში გამოიკვეთა, რომ tf-ის გამოყენება იყო არაოპტიმალური (26) მოგვიანებით რობერტსონის (S.E.Robertson) და მისი გუნდის მიერ ალბათური მოდელის ფარგლებში შემუშავდა წონის დათვლის ეფექტური სქემა (27), ხოლო მოგვიანებით კი კორნელის უნივერსიტეტის მკვლევარებმა შეიმუშავეს ახალი სქემა, რომელიც მოიცავდა დოკუმენტის სიგრძის გავლენას წონის დათვლისათვის (28) , შემდეგ გაჩნდა okapiBM25.

ლოგარითმული:

$$f(t_i, d_j) = 1 + \log f(t_i, d_j), \quad (18)$$

შევსებითი(Augmented):

$$f(t_i, d_j) = 0.5 + \frac{0.5 \times f(t_i, d_j)}{max\{f(w, d_j) : w \in d_j\}}. \quad (19)$$

ბულის:

$$f(t_i, d_j) = \begin{cases} 1, თუ\ tf > 0 \\ 0, სხვა\ შემთხვევაში \end{cases} \quad (20)$$

ტერმინების წონის დათვლის განსხვავებული მეთოდების გამოყენებამ შესაძლებელია გამოიწვიოს ძებნის სისტემის ეფექტურობის არსებითი ცვლილება. მათი შერჩევა ძირითადად დამოკიდებულია ტექსტის სიგრძეზე (სიტყვების რაოდენობაზე) (28).

1.4. ბუნებრივი ენების ანალიზი ინფორმაციული ძებნის ამოცანებში

ბუნებრივი ენის ანალიზი მოიცავს სინტაქსსა და სემანტიკაზე დაყრდნობით ცოდნის ამოღებას ბუნებრივ ენაზე დაწერილი დოკუმენტებიდან. ასეთი მიდგომა შეიძლება განვიხილოთ, როგორც „სემანტიკური“ მიდგომა იმ ლოგიკით, რომ დოკუმენტის შინაარსი და სტრუქტურა განისაზღვრება არასტატისტიკური მეთოდებით.

ადამიანისათვის მარტივია სინტაქსსა და სემანტიკაზე დაყრდნობით დოკუმენტების რელევანტურობის განსაზღვრა. ავტომატური სისტემებისათვის სტატისტიკური გამოთვლა მარტივია, მაგრამ ამ მეთოდებით, შესაძლებელია, მოუძიებელი დარჩეს მნიშვნელოვანი დოკუმენტები. ამიტომ დღეისათვის სტატისტიკური/ალბათური მეთოდებისა და სინტაქსური/სემანტიკური მეთოდების ინტეგრაცია იდეალური გამოსავალია ძებნის პროცესის ეფექტურობის გაზრდისათვის (29) .

ბუნებრივი ენის ანალიზის მეთოდები კლასიფიკაციას ახდენს ტექსტის ლინგვისტური ერთეულების დამუშავების პრინციპებისა და სირთულის დონის მიხედვით (30). კლასიფიკაციის პროცესში განიხილება ტექსტების დამუშავების შემდეგი ეტაპები: ფონოლოგიური, მორფოლოგიური, სინტაქსური, სემანტიკური, მსჯელობის და პრაგმატული. ფონოლოგიური ეტაპი არის საუბრის ხმების, ფონემების გადამუშავების ეტაპი. ის ძირითადად გამოიყენება საუბრის ტექსტად კვალიფიკაციისას. ტრადიციული ინფორმაციული ძებნის მეთოდები იყენებს NLP მიდგომას, მაგრამ მხოლოდ მორფოლოგიურ და ლექსიკურ ეტაპებზე.

მორფოლოგიური ეტაპი მოიცავს მოცემული ტერმინის სხვადასხვა ფორმის ანალიზს მისი შემადგენლობის მიხედვით: პრეფიქსები, ფუძეები და სუფიქსები. შესაბამისად, ტრადიციული ეგრეთ წოდებული სტემინგ[2] მეთოდები (31), რომლებიც დოკუმენტისა და მოთხოვნის მსგავსების განმსაზღვრელი ტერმინების რაოდენობას ამცირებენ ტექსტში სიტყვების ფუძის ამოღების გზით, ინფორმაციის მორფოლოგიური დამუშავების მაგალითია. *ლექსიკური ეტაპი* მოიცავს სიტყვის ანალიზს მის სტრუქტურასა და მნიშვნელობაზე დაყრდნობით. მაგალითად, ტრადიციულ ლექსიკურ-ინფორმაციულ ძებნას აქვს სია „stop“ სიტყვებისა, რომელთაც აქვთ დაბალი სემანტიკური მნიშვნელობა (29). შესაბამისად, სიტყვათა სიის ინდექსაციისა და მოთხოვნათა ფორმულირებისთვის განმარტებითი ლექსიკონების გამოყენება და წარმოება ლექსიკური ინფორმაციული ძებნის პროცესის სხვა მაგალითია. ბუნებრივი ენის ანალიზის ერთ-ერთი ამოცანაა ტექსტიდან მეტყველების ნაწილების გამოყოფა, რომელსაც ტექსტის ლინგვისტური დამუშავება მოიცავს. ეს პროცესი იშვიათია ტრადიციულ ინფორმაციულ ძებნაში[3] (Information Retraval – IR).

სინტაქსურია ეტაპი, სადაც წინადადების სინტაქსური წყობა განისაზღვრება სიტყვების ადგილმდებარეობის მიხედვით. პრაქტიკაში ერთ წინადადებას შესაძლოა ჰქონდეს ბევრი შესაძლო სტრუქტურა. სწორი სტრუქტურის დადგენა მოითხოვს მაღალი დონის ცოდნას (ან სტატისტიკას ტრენინგის საფუძველზე). ამის გამო სინტაქსური ეტაპი ან საერთოდ არაა, ან იშვიათად გამოიყენება ტრადიციულ IR-ში.ზოგჯერ ტექსტის

[2] სტემინგი-მორფოლოგიური დამუშავების პროცესი სიტყვიდან ფუძის მისაღებად.

[3] შემდგომში ტერმინის „ინფორმაციული ძებნა“ აღსანიშნავად გამოვიყენებთ IR

სინტაქსური ანალიზი გამოიყენება სიტყვაზე დიდი ერთეულების ამოსაცნობად. მაგალითად, ფრაზები, მაგრამ აქაც, სტატისტიკური თანხვედრა და ვარაუდი უკეთესი და უფრო მეტად გამოსაყენებელი მეთოდია IR–ში.

გამომდინარე იქიდან, რომ სიტყვისათვის ორაზროვნების მოცილება შესაძლებელია მხოლოდ ფრაზიდან, წინადადებიდან ან ხანდახან ტექსტის უფრო დიდი ერთეულიდან, სემანტიკური არის ეტაპი, რომელიც მოიცავს ტექსტის შინაარსის დადგენას არა ერთეული სიტყვების, არამედ წინადადებების მიხედვით და შესაბამისად, სიტყვათა ორაზროვნების ამოცნობაც სემანტიკური ამოცანაა. სირთულისა და NLP სიზუსტის საჭიროების გამო ტრადიციული IR თავს არიდებს სემანტიკური ეტაპის გადამუშავებას და სიტყვათა შეთავსების სტატისტიკური მეთოდებით განსაზღვრას ამჯობინებს.

მსჯელობა ის ეტაპია, რომელიც მოიცავს ტექსტის სტრუქტურისა და მნიშვნელობის ამოცნობას უფრო დიდი ერთეულების საფუძველზე. მაგალითად, პარაგრაფების, ფრაზებისა და წინადადებების დახმარებით.

პრაგმატულია ეტაპი, როდესაც დოკუმენტის დამუშავებისათვის გამოყენებულია ზოგადი ცოდნა, რომელიც კონკრეტულ სიტუაციაში დოკუმენტისათვის უცხოა. მაგ. შესაძლებელია განისაზღვროს, მოცემულ მოთხოვნაში მომხმარებელს რა ინფორმაციის მიღება აქვს მიზნად.

ტრადიციულ ინფორმაციულ ძებნაში სემანტიკის განსაზღვრის ყველაზე მნიშვნელოვანი წყარო არის უკუკავშირი, რომელიც მოიცავს მოთხოვნის დაზუსტებასა და გაფართოებას ადამიანური ცოდნის საფუძველზე. ეს პროცესი დამყარებულია მომხმარებლის მიერ ტექსტის სემანტიკის გაგებაზე, ინფორმაციის გაანალიზებაზე. შესაბამისად, უკუკავშირი არის უკვე NLP მაღალ დონეზე. თუმცა ტრადიციულ ინფორმაციულ ძებნაში ეს პროცესი ხორციელდება ტექსტის აღმწერი ტერმინებითა და მათი შესაბამისი წონებით. ძებნის მეთოდები შესაძლებელია გაუმჯობესდეს და გაფართოვდეს სხვადასხვა გზით: მაგალითად, შესაძლებელია ტექსტის აღმწერი იყოს არა დოკუმენტიდან ამოღებული სიტყვა, არამედ უფრო მაღალი დონის ლინგვისტური ერთეული, მაგალითად, ფრაზა ან კონცეპტი, რომელიც ბუნებრივად დოკუმენტში არ არსებობს.

სემანტიკური მეთოდები შესაძლებლობას იძლევა ამოღებული იქნეს ტექსტიდან ისეთი ტერმინები, რომელთა საშუალებითაც განისაზღვრება ტექსტის სემანტიკა. ეს მეთოდები გამოიყენება ორაზროვნების პრობლემის გადასაჭრელად. მათ შეუძლიათ გავლენა მოახდინონ კატეგორიის განსაზღვრაზე ისეთი ტერმინებით, რომლებიც არ ფიგურირებს მომხმარებლის მოთხოვნაში და არც დოკუმენტში არ გვხვდება. ისინი გამოიყენება მოთხოვნაში ტერმინების სინონიმებით ჩასანაცვლებლად. ასევე შესაძლებელია მათი გამოყენება სიტყვებსა და ფრაზებს შორის სემანტიკური ურთიერთობის დასადგენად.

პირველი თავის დასკვნა

პირველ თავში განხილულ იქნა თანამედროვე მონაცემთა ბაზების ტიპები (სტრუქტურირებული, არასტრუქტურირებული, ნახევრადსტრუქტურირებული) და ის პრობლემები, რომლებიც წარმოიშვა არასტრუქტურირებულ მონაცემთა ბაზებში ინფორმაციული ძებნის პროცესთან დაკავშირებით. განვიხილეთ ინფორმაციული ძებნის მიზანი, რომელიც წარმოადგენს იდეალური კრიტერიუმის შერჩევას ძებნის პროცესის წარმატებული შედეგის მისაღწევად.

აღვწერეთ ძებნის სტრატეგიები, რომლებიც იდენტიფიცირებულია, როგორც ძებნის მოდელები. ისინი ერთმანეთისაგან განსხვავდებიან დოკუმენტის და მოთხოვნის წარმოდგენის ფორმებით და მათი ერთმანეთთან მსგავსების განსაზღვრის მეთოდებით. თუმცა, ყველა მოდელი ფოკუსირებულია სამ კომპონენტებზე: დოკუმენტი, მოთხოვნა, რანჟირების ფუნქცია.

ჩვენ განვიხილეთ ინფორმაციული ძებნის ძირითადი მოდელები: ბულის მოდელი, ვექტორული სივრცის მოდელი და ალბათური ძებნის მოდელი, მათი მუშაობის თავისებურებანი და ის დადებითი და უარყოფითი თვისებები, რომლებიც ახასიათებს თითოეულ მათგანს. ძებნის მოდელები ეფუძნება ტერმინების წონის დათვლის პრინციპს. ტერმინს წონა არის სტატისტიკური სიდიდე, რომელიც განისაზღვრება დოკუმენტში ტერმინის შეხვედრის სიხშირით და განსაზღვრავს ტერმინის მნიშვნელოვნებას. შესაბამისად, ძებნის მოდელების ფუნქციონირების ჩარჩოებში, განხილულ იქნა ტერმინის წონის განსაზღვრის მეთოდები, რომელთა ნაწილი მუშავდება ალბათური ძებნის მოდელების საზღვრებში, ხოლო ნაწილი რეალიზდება ვექტორული სივრცის მოდელის ფარგლებში.

ამასთანავე, განვიხილეთ ინფორმაციული ძებნის ამოცანებში ბუნებრივი ანალიზის მეთოდები, მათი ძირითადი ეტაპები და ის მნიშვნელოვანი თვისებები, რომელიც ახასიათებს თითოეულ მათგანს. ბუნებრივი ენის ანალიზი მოიცავს სინტაქსსა და სემანტიკაზე დაყრდნობით ცოდნის ამოღებას ბუნებრივ ენაზე დაწერილი დოკუმენტებიდან. ასეთი მიდგომა შეიძლება განვიხილოთ, როგორც „სემანტიკური“ მიდგომა იმ ლოგიკით, რომ დოკუმენტის შინაარსი და სტრუქტურა განისაზღვრება არასტატისტიკური მეთოდებით. ავტომატური სისტემებისათვის სტატისტიკური გამოთვლა მარტივია, მაგრამ ამ მეთოდებით, შესაძლებელია მოუძიებელი დარჩეს მნიშვნელოვანი დოკუმენტები. ამიტომ, დღეისათვის სტატისტიკური/ალბათური მეთოდებისა და სინტაქსური/სემანტიკური მეთოდების ინტეგრაცია იდეალური გამოსავალია ძებნის პროცესის ეფექტურობის გაზრდისათვის.

2. კლასიფიკაცია, როგორც ინფორმაციული ძებნის ამოცანა

ინფორმაციის ძებნის პროცესი არ წარმოადგენს მხოლოდ ერთი სახის ოპერაციის შედეგს. მისი წარმატებულობა და რელევანტურობა დამოკიდებულია ძებნის ციკლის ადეკვატურობაზე და სისრულეზე. ამ ციკლში ერთ-ერთი მნიშვნელოვანი ადგილი უკავია კლასიფიკაციის ეტაპს, რომლითაც, როგორც წესი, იწყება ძებნის პროცესი.

კლასიფიკაცია წარმოადგენს ძებნის პროცესს, რომლის მიზანია ავტომატურად მიანიჭოს კლასები დოკუმეტებს წინასწარ განსაზღვრული სიმრავლიდან. ტექსტების კლასიფიკაცია დამოუკიდებელი ამოცანაა, მაგრამ იგი გამოიყენება საძიებო სისტემებში ძებნის შედეგების გასაუმჯობესებლად. მაგ. კლასიფიკატორების გამოყენება ძებნის შედეგების დაჯგუფებისათვის. კლასიფიკაციაში გამოიყენება იგივე მოდელები, რაც ინფორმაციულ ძებნაში. თუმცა არის განსხვავებული მიდგომაც - კლასიფიკაციის მოდელები გამოიყენება დოკუმენტებისათვის შესაბამისი კლასის წინასწარი ამოცნობისათვის. ამისათვის დოკუმეტების სიმრავლე წინასწარ დამუშავებამდე იყოფა დასასწავლ და ტესტირების სიმრავლეებად. თავდაპირველად სისტემა საჭიროებს დასასწავლ დოკუმენტებს, რათა, მათზე დაყრდნობით, შექმნას კლასიფიკატორები. ხოლო შემდეგ, სატესტო დოკუმენტების სიმრავლის საფუძველზე, პროგნოზირებენ მათი მიკუთვნების კლასს და ახდენენ კლასიფიკაციას.

კლასიფიკაციის ამოცანა ზოგადი სახით შეიძლება ასე ჩამოვაყალიბოთ (32):

კლასიფიკაციის ამოცანაა მიანიჭოს ბულის მნიშვნელობა წყვილს $\langle d_j, c_i \rangle \in \mathcal{D} \times C$, სადაც D წარმოადგენს დოკუმენტების დომენს $\mathcal{D} = \{d_1, \dots d_{|D|}\}$ და $C = \{c_1, \dots c_{|C|}\}$ არის წინასწარ განსაზღვრული კატეგორიების სიმრავლე.

$\langle d_j, c_i \rangle$ წყვილს მიენიჭება მნიშვნელობა True („1“) თუ იქნება გადაწვეტილება d_j მიეკუთვნოს c_i კლასს, ხოლო მნიშვნელობა False („0“), არის მაჩვენებელი იმისა რომ, d_j არ მიეკუთვნოს c_i კლასს.

უფრო ფორმალურად - მიახლოებით განისაზღვროს უცნობი მიზნის ფუნქცია (რომელიც განსაზღვრავს, როგორ უნდა იქნას დოკუმენტი კლასიფიცირებული): $\Phi: D \times C \to \{True, \text{False}\}$ ისეთი Φ' ($\Phi': D \times C \to \{\text{True}, \text{False}\}$) კლასიფიკატორის მეშვეობით, რომელიც „მაქსიმალურად თანხვედრილი“ იქნება Φ-სთან.

თანხვედრის/დამთხვევის (რომელსაც „ეფექტურობას უწოდებენ“) შესაფასებლად იყენებენ ინფორმაციული ძებნის ისეთ კლასიკურ სიდიდეებს როგორიცაა: სიზუსტე (precision) და მთლიანობა (recall).

არსებობს დოკუმენტების კლასიფიკაციის მრავალი მეთოდი, თუმცა ყოველი მათგანი ეფუძნება ტექსტის დამუშავების პროცესს. ტექსტის დამუშავება კი, თავის მხრივ, დამოკიდებულია ენის თავისებურებებსა და სირთულეებზე. კლასიფიკაციის ალგორითმები მუშაობენ უკვე რაღაც მეთოდით დამუშავებულ ტექსტებზე. ტექსტის დამუშავების სხვადასხვა გზა შეიძლება არსებობდეს, თუმცა კლასიფიკაციის თითქმის ყველა ცნობილი ალგორითმი იყენებს მის მინიმიზაციას, მასში მხოლოდ ყველაზე ხშირად განმეორებადი სიტყვების დატოვების გზით.

კლასიფიკაციის საწყისი ამოცანაა დოკუმენტის დამუშავების პროცესი, რომელიც მის განსაზღვრული ნორმალიზებული ფორმით წარმოდგენას გულისხმობს. ეს პროცესი რამდენიმე ეტაპად ხორციელდება:

1. ტექსტის საწყისი დამუშავება
   - თვისებების ამოღება;
     - ტოკენიზაცია;
     - „სტოპ“ სიტყვების წაშლა
     - სტემინგი, ლემატიზაცია;
   - ნიშან-თვისებების შერჩევა;
2. კლასიფიკაციის მეთოდის შერჩევა/ფორმირება.

კლასიფიკაციის ამოცანებში ინფორმაციული ძებნის ტექნოლოგიები გამოიყენება სამ ფაზაში: ინდექსაცია, კლასიფიკატორის ფორმირება, შეფასების განსაზღვრა.

2.1. ტექსტის საწყისი დამუშავება

ზოგადად, დოკუმეტი შეიცავს არა მხოლოდ იმ სიტყვებს, რომლებიც მნიშვნელოვანია დოკუმენტის შინაარსობრივი მხარის წარმოდგენისათვის, არამედ ისეთსაც, რომლებსაც არანაირი დატვირთვა ამ კუთხით არ გააჩნიათ. ამიტომ მისაღებია და დასამუშავებლად მოსახერხებელია დოკუმენტის ისეთი სახით წარმოდგენა, სადაც ასეთი არამნიშვნელოვანი ტერმინები არ იქნება.

კლასიფიკაციის პროცესში დოკუმენტის წარმოდგენის ერთ-ერთი ფართოდ გავრცელებული მეთოდია „სიტყვების ჩანთა“ (bag of words – BOW) (11) , რომელიც გულისხობს დოკუმენტის სიტყვების სიმრავლის სახით ჩაწერას. არსებობს დოკუმენტების წარმოდგენის სხვა საშუალებებიც, რომელთა გამოყენებაც ხდება კონკრეტული სიტუაციების შესაბამისად. მაგ:

- კონცეპტუალური თვისებები - გამოხატავენ ორიგინალური დოკუმენტების შინაარსს, კონტექსტუალური თვისებები - შეიცავენ ტერმინების შესაბამის კონტექსტუალურ ინფორმაციას (მაგ ბიგრამები, ტრიგრამები და ა. შ.) (33);
- მექანიკურად ამოღებული თვისებები - მათი ამოღება დოკუმენტიდან ხდება დოკუმენტის შინაარსისა და ენის სტრუქტურის შესახებ ყოველგვარი ცოდნის გარეშე (34);
- დოკუმენტის სტრუქტურის შესაბამისი თვისებები (სიტყვების საერთო რაოდენობა, წინადადებების რაოდენობა, სიგრძე და ა. შ) (35). თუმცა ყველაზე უფრო ფართოდ გამოყენებადი არის „სიტყვების ჩანთა“(BOW).

2.1.1. ნიშან-თვისებების ამოღება

დამუშავების პირველი ეტაპი ტექსტიდან თვისებების ამოღებაა. ეს ეტაპი მოიცავს ტოკენიზაციას, ანუ ტექსტის ლექსემებად[4] დაყოფას. ტექსტის ლექსემებად დაყოფის ყველაზე მარტივი გზა არის ტექსტის გახლეჩა „ჰარის“ პოზიციაში, შედეგად მიიღება

[4] სალექსიკონო ერთეული, რომელშიც გაერთიანებულია სიტყვის სხვადასხვა პარადიგმული ფორმა

სიტყვების სიმრავლე, რომელთაგან ბევრი გამოუყენებელია, ზოგი კი საკმაოდ მნიშვნელოვანია დოკუმენტის კატეგორიის განსაზღვრისათვის. ამიტომ ასეთი „უსარგებლო“ სიტყვები დოკუმენტიდან ამოიღება, დარჩენილი სიტყვები კი მორფოლოგიურად მუშავდება, რაც მოიცავს სტემინგის და ლემატიზაციის პროცესს.

სტემინგი ევრისტიკული პროცესია, რომელიც მოიცავს სიტყვიდან თანდართული აფიქსების[5] ჩამოცილებას.

ლემატიზაცია კი სიტყვის მორფოლოგიური ანალიზის დამუშავების სრული ეტაპია, რომლის დროსაც სიტყვიდან ხდება გრამატიკული მნიშვნელობების მქონე დაბოლოებების ჩამოცილება და ბრუნდება ძირითადი, უცვლელი ლექსიკონური ფორმა - ლემა.

როგორც ცნობილია, ზოგიერთი სიტყვა შესაძლოა შეგვხვდეს სხვადასხვა გრამატიკული ფორმით. ლემატიზაციის პროცესის განხორციელების მიზანია სიტყვათა ფორმების წარმოდგენა ერთი კანონიკური ფორმით.

2.1.2. სტემინგი და ლემატიზაცია

სტემინგისა და ლემატიზაციის პროცესი ინფორმაციის დამუშავების უნიშვნელოვანესი ეტაპია. ძებნის პროცესის გამარტივებისათვის საკმაოდ ეფექტური გამოსავალია დოკუმეტების ავტომატური ანალიზი, გამოუსადეგარ მონაცემთა ნაწილის მოშორება და მხოლოდ სასარგებლო ინფორმაციის დატოვება. ენის თავისებურებებიდან გამომდინარე, სხვადასხვა ბუნებრივი ენებისათვის, ეს პროცესი განსხვავებულია, ორივე პრიცესის მიზანია სიტყვიდან ფუძის მიღება, მაგრამ ამას ახორციელებენ სხვადასხვა გზით: სტემინგი ემყარება წესებს და არ ითვალისწინებს მეტყველების ნაწილებს, ხოლო სიტყვის უცვლელი ნაწილის მიღება მეტყველების ნაწილის კატეგორიის გათვალისწინებით, ლემატიზაციაა.

სტემინგ ალგორითმს ან იგივე სტემერს აქვს სამი დანიშნულება: პირველია სიტყვათა დაჯგუფება თემის მიხედვით. ბევრი სიტყვა წარმოიქმნება ერთი და იმავე ფუძისგან და ერთსა და იმავე არსს გულისხმობს (მაგ. სწავლა, მო-სწავლ-ე, მა-სწავლ-ებ-ელ-ი). ეს სიტყვები ნაწარმოებია პრეფიქს-სუფიქსებით. სტემერები, რომლებიც ინგლისურ ენაზე მუშაობენ, ძირითადად სუფიქსების მოცილებას ანხორციელებენ, რადგან ზოგიერთ შემთხვევაში პრეფიქსები და ინფიქსები სიტყვის მნიშვნელობის შეცვლას იწვევს (36) გამონაკლისები გვხვდება ისეთ ფლექტიურ ენებში, როგორიცაა გერმანული და დანიური (37), აგრეთვე კონკრეტული სფეროსათვის დამახასიათებელ დოკუმენტებში, მაგ: მედიცინა, ქიმია, სადაც პრეფიქსები და სუფიქსები განსაზღვრავენ სიტყვის არსს. სუფიქსებში გამოიყოფა ორი ძირითადი სახე: “უღლებადი“ წარმონაქმნი, რომლის ძირითადი ფუნქცია გრამატიკული ინფორმაციაა, მოიცავს ინფორმაციას სიტყვის სქესის, რაოდენობის, ხასიათის ან დროის შესახებ. და დერივაციული წარმონაქმნები, რომლებიც არ იწვევენ საწყისი სიტყვის მნიშვნელობის ცვლილებას, ისინი ქმნიან ახალ სიტყვებს უკვე არსებული სიტყვებიდან (38). წარმოქმნილი სიტყვისგან სუფიქსების მოშორებით მიიღება ფუძე, რომელიც თითქმის იდენტურია

[5] "მიმაგრებული" სიტყვის ნაწილი, რომელსაც აქვს გრამატიკული (და არა საგნობრივი) მნიშვნელობა (პრეფიქს-სუფიქსი)

მისი მორფოლოგიური ძირისა, რის შემდეგაც თემატურად ერთმანეთთან დაკავშირებული სიტყვები შესაძლებელია განისაზღვროს მათი ფუძეების დამთხვევით.

მეორე დანიშნულება სტემინგ ალგორითმისა უკავშირდება ინფორმაციის ამოცნობის პროცესს, რომელიც ამ ეტაპზე სიტყვიდან ფუძის ამოღებას გულისხმობს. ჩვენ შეგვიძლია დოკუმენტების უკეთ ინდექსაცია ტერმინებით, ხოლო ტერმინები კი სწორედ ფუძის საშუალებით ჯგუფდება.

მესამე დანიშნულება საერთო ფუძის მქონე სიტყვების გაერთიანებაა. ეს, ზოგადად, ამცირებს ლექსიკონის ზომას. პროცესის შედეგად შესაძლებელია მთელი მონაცემების ფუძეებზე დაყვანა, რაც ამცირებს გამოსაყენებელ სივრცეს და ამსუბუქებს სისტემის დატვირთვას.

სტემინგის გავრცელებული მიდგომებია:

1. ალგორითმზე დაფუძნებული სტემინგი. ასეთი სტემერები არ ითვალისწინებენ ისეთ ლინგვისტურ დამოკიდებულებას, როგორიცაა სქესი, დრო, ისინი იყენებენ წინასწარ დადგენილ წესებს, რათა გაიგონ მოაშორონ თუ არა აფიქსები. ვინაიდან არაა გათვალისწინებული ლინგვისტური კანონები, შედეგი ხშირად არასწორად აწყობილი სიტყვაა, რომელსაც არ აქვს არანაირი მნიშვნელობა.
2. ლინგვისტიკაზე დამყარებული მიდგომა, სადაც სიტყვის დამუშავება ხდება ლინგვისტური წესებით .

სტემინგის ალგორითმები

ლიტერატურაში ნახსენები პირველი სტემერი ლოვინსის (J.B.Lovins) ალგორითმია (39), რომელიც შეიქმნა ჯერ ინგლისური, ხოლო შემდეგ, სხვადასხვა სამეცნიერო კვლევებში, სხვადასხვა ენისათვის განხორციელდა მისი მოდიფიცირება (40).

ალგორითმი შედგება ორი ეტაპისაგან: სუფიქსების მოშორება და დარჩენილი ძირის დახარისხება. ალგორითი ითვალისწინებს 294 სუფიქსს, რომელიც სიტყვასთან 29 ვარიანტით გამოიყენება. ამის შემდეგ დარჩენილ ძირს სჭირდება რამდენიმე ისეთი ლინგვისტური პრობლემის მოგვარება, როგორიცაა ორმაგი დაბოლოება (ორი „ლ“ ან ორი „დ“). ამ საფეხურს „ჩაწერის ფაზა“ ეწოდება. ალგორითმში ჩადებული 35 წესი არკვევს, ძირის დარჩენილი ნაწილი მოდიფიცირებული უნდა იყოს, თუ - მთლიანად წაშლილი. ბოლოს, სიტყვათა გაერთიანება ხდება ალგორითმით, რომელიც ეძებს არა ზუსტ, არამედ მსგავს ფუძეებს, რომლებიც მაინც ახლოსაა ერთმანეთთან მნიშვნელობით. ამ ალგორითმით შესაძლებელია ბევრი შეცდომის დაშვება და შემცირება „სიზუსტისა” და „სისრულის”. აღნიშნული პრობლემების თავიდან ასაცილებლად დავსონმა (J.Dawson) შექმნა ლოვინსის სტემერის მოდიფიცირებული ვარიანტი, მის ალგორითმში სუფიქსების ამოღების თავდაპირველი ეტაპი გაუქმებულია და სიტყვები პირდაპირ ჯგუფდება მსგავსი ძირის მიხედვით. ამავე დროს სუფიქსების სია 1200-მდე გაიზარდა (41).

იმის მიუხედავად, რომ ლოვინსის ალგორითმი ყველაზე ხშირად გამოიყენება, პორტერის სტემერი ყველაზე პოპულარულია ინფორმაციული ძებნის ამოცანებში (42) . იგი აბალანსებს სიმარტივესა და სიზუსტეს. პორტერს აქვს 5 საფეხურიანი ალგორითმი, რომელიც ლექსიკონში არსებულ ყველა სიტყვასთან შეთავსებადია. ალგორითმი ემყარება 60 წესს, რომლებიც შემდეგ იყოფა 5 საფეხურად. ამ ალგორითმის ძირითადი

იდეაა, რომ სუფიქსი (ინგლისურ ენაში) თავად შედგება პატარა და მარტივი სუფიქსებისაგან. ისინი სიტყვას ეტაპობრივად (ხუთ ეტაპად) სცილდება. ყოველი წინა ეტაპის დასრულების შემდეგ - ახალი იწყება.

კიდევ ერთი ცნობილი შემოთავაზებაა პაის/ჰასკის (Paice/Husk) სტემერი ინგლისური ენისათვის (43), რომელიც იყენებს 120 წესს სიტყვის ბოლო სიმბოლოსა და თვითონ სიტყვებს შორის დამოკიდებულების დასადგენად. იგი ხორციელდება რამდენიმე ეტაპად. სიტყვის დამუშავების ყველა ეტაპზე ალგორითმი უზრუნველყოფს დაბოლოების ან მოცილებას, ან შეცვლას. თუ აღმოჩნდა პირობა, რომელიც წესს არ შეესაბამება, ალგორითმი წყდება (44).

გარდა ინგლისური სტემერებისა, ზოგიერთ მკვლევარს აქვს მოდიფიცირებული მიდგომა ან შემოთავაზება სხვა ენებისთვის. ლინგვისტიკაზე დაყრდნობით ენა შესაძლებელია გაიყოს ორ ძირითად კატეგორიად, მათი მორფოლოგიური სტრუქტურის მიხედვით: ანალიზური ენა და სინთეზური ენა

ანალიზური ისეთი ტიპის ენებია, რომლებშიც სიტყვათა შორის გრამატიკული მიმართებები გამოიხატება დამხმარე სიტყვების და ნაწილაკების მეშვეობით, ინტონაციით (ჩინური) ან სიტყვათა თანმიმდევრობით. ანალიზურია ინგლისური, ფრანგული, ახალი სპარსული, ჩინური და ვიეტნამური ენები.

სინთეზურია ენები, რომლებშიც სიტყვათა შორის გრამატიკული მიმართებები გამოიხატება მორფოლოგიური ხერხებით, თვით სიტყვის ფარგლებში აფიქსებისა და ფუძის ფლექსიის (სიტყვათწარმოება) საშუალებით. სინთეზურია ძველი ბერძნული, ლათინური, რუსული, ქართული და სხვა ენები). მეორე კატეგორია კიდევ იყოფა 3 ქვეკატეგორიად : აგლუტინატიური ენა, ფლექტიური ენა, პოლისინთეზური ენა. ეს კლასიფიკაცია იდეალურია, რადგან ყველა ენა სხვადასხვა კატეგორიას ეკუთვნის. შემოღებულია ორი მუდმივი ცვლადი, სინთეზურობისა და ფლექტიურობის ინდექსი იმისთვის, რომ გვიჩვენოს, რამდენად მიეკუთვნება ერთი ენა რომელიმე კატეგორიას. (45).

სინთეზურობის ინდექსი აღწერს როგორ და რა დონეზე ემატება სიტყვას აფიქსები. ერთი უკიდურესობაა ყველაზე ანალიტიკური ენები, სადაც ყველა მორფემა[6] თავისუფალია. ხოლო მეორე - პოლისინთეზური ენაა, სადაც ენას აქვს მიდრეკილება შეიცავდეს წინადადებებს, რომლებიც ერთი სიტყვისგან და სხვადასხვა აფიქსებისგან შედგება.

მეორე ცვლადი, ფლექტიურობის ინდექსი, აღწერს, თუ რამდენად ადვილია სიტყვების მორფემებად დანაწილება. ამ შკალაზე ერთი უკიდურესობაა ენები, სადაც მარტივად ნაწილდება და ნათლად აღიქმევა მორფემები, ხოლო მეორე - მოიცავს სიტყვებს, რომელიც რთულად აღსაქმელი ან გასარჩევი მორფემებისგან შედგება.

ენების სიმრავლე განაპირობებს პრობლემების სიმრავლეს და მათი მოგვარება მხოლოდ არატრადიციული ხერხებითაა შესაძლებელი, რადგან კლასიკური ხერხები ამ

[6] მორფემა - სიტყვის ნაწილი (სუფიქსი, პრეფიქსი, ძირი), რომელსაც აქვს ლექსიკური ან გრამატიკული მნიშვნელობა.

შემთხვევაში არ გამოირჩევიან იმ სიზუსტითა და ეფექტურობით, როგორც ინგლისურში ან მის მსგავს ენებში.

ფინური და თურქული ცვალებადი მორფოლოგიის მქონე ენების მაგალითია. თურქულში 23000 ძირია და სიტყვები ფორმირდება მათი გრამატიკული ფუნქციის მიხედვით, სუფიქსების დახმარებით. ეს შედეგი თეორიულად უსასრულო სიტყვებია. სხვადასხვა მიდგომით ცდილობენ ამ პრობლემის შემცირებას, სიზუსტის შენარჩუნებას, მაგალითად „N" გრამების გამოყენებით და აფიქსების მოშორებით, წესებითა და კარგად ცნობილი სუფიქსების სიით, როგორც პორტერის სტემერში.

თუ შევხედავთ ფლექტიურ ენებს, ამ ჯგუფს ძირითადად მიეკუთვნება ინდოევროპული ენები. მათი სტემერის უმრავლესობა დაფუძნებულია პორტერის მიდგომაზე, რადგან ის იდეალურად ჯდება მათ მორფოლოგიურ სტრუქტურაში. ერთი მაგალითია ალგორითმული სტემერი „პოპოვიჩი და ვილეტი სლოვენური ენისათვის", რომელიც იყენებს 5276 კარგად ნაცნობ სუფიქს. ეს ციფრი გაცილებით მაღალია პორტერის ორიგინალურ შემოთავაზებასთან, რადგან სლოვენური მორფოლოგიურად ინგლისურზე გაცილებით მდიდარია. ამ სტემერის საშუალებით ჩატარებული ექსპერიმენტები კარგი შედეგებითა და სიზუსტით დასრულდა. ბევრი მკვლევარი ამტკიცებს, რომ სტემერის ეფექტურობა იზრდება ენის კომპლექსურ სირთულესთან ერთად. მაგ. არაბული (46), ბერძნული (47) ენებისათვის. ეს უკანასკნელი კიდევ უფრო მეტი სირთულით გამოირჩევა, ამიტომაც ავტორებმა შექმნეს მეტყველების ნაწილებით დაჭდევება (part-of-speech tagging phase), რათა სუფიქსების მოშორებასთან ერთად გაიგონ, სიტყვა რომელ გრამატიკულ კატეგორიას ექვემდებარება. შემდეგ კი, წესისამებრ, შორდება მორიგი სუფიქსები ფრაზის ნაწილებითი დაჭდევების მიხედვით, ამიტომაც ბერძნული სტემერების 96,7% ლექსიკონის ძირს სწორად იღებს.

არსებობს პოლისინთეზური ენები, რომელთათვისაც სტემინგის პროცესი არ განხორციელებულა. მაგალითად, ჩუკოტკურ-კამჩატკური, აფხაზურ-ადიღური და სხვა მრავალი ენა სამხრეთ-ამერიკულ ენათა ოჯახიდან . სავარაუდოდ, ამ ენებში ის ფაქტი, რომ არსებობს მრავალი მორფემა, ართულებს ტერმინის იდენტიფიკაციის პროცესს.

ამის მიუხედავად, ყველაზე მეტად გამოყენებადი მაინც პორტერის ალგორითმია მისი სიმარტივის, შეგუებადობისა და გაფართოების გამო. ამასთან ერთად, პორტერმა გაამარტივა საქმე და შექმნა „თოვლის გუნდა" (Snowball) პლატფორმა, სადაც ყალიბდება ახალი სტემერები (48). Snowball, რომელიც წარმოდგენილია ადვილად სასწავლი ენით, იძლევა სტემერისათვის ANSI[7], C ან JAVA ვერსიის გამოყენების შესაძლებლობას. დღეისათვის 25 სხვადასხვა ენაა დამუშავებული Snowball-ის საშუალებით, რომელთაგან უმრავლესობა აგლუტინატიური ან ფლექტიური ენებია.

2.1.3. ნიშან-თვისებების შერჩევა

ნიშან-თვისებათა მონიშვნის პროცესის იდეა მიღებული ტერმინების სიმრავლიდან მნიშვნელოვანის გამოყოფაა, რის შედეგადაც მიიღება მრავალგანზომილებიან ნიშან-თვისებათა სივრცე. ყველაზე დიდი სირთულე და პრობლემა ტექსტების კატეგორიზაციის პროცესში ნიშან-თვისებათა სივრცის მაღალი განზომილებაა. ეს ნიშან-თვისებები, რომლებიც მოიცავს ტერმინებსა ან ფრაზებს,

[7] American National *Standards* Institute

შეიძლება ასეულობით და მეტიც იყოს. მათი ასეთი დიდი რაოდენობა მრავალი სასწავლო ალგორითმისათვის გარკვეულ პრობლემას წარმოადგენს. ამიტომ მთავარ მიზანს კლასიფიკაციის პროცესში წარმოადგენს ნიშან-თვისებათა სივრცის შემცირება, შედეგის სიზუსტის გაუარესების გარეშე. ეს შეიძლება განხორციელდეს ავტომატურად ან ხელით. თვისებათა შერჩევის ავტომატური მეთოდი მოიცავს არაინფორმაციული ტერმინების მოცილებას კოლექციიდან და სივრცის ფორმირებას მნიშვნელოვანი ტერმინებით.

ნიშან-თვისებათა შერჩევის ორი ძირითადი მოდელი არსებობს: ფილტრაციის (Filter) და შეფუთვის (Wrapper) (49).

- ფილტრაციის მოდელის გამოყენებისას ნიშან-თვისებათა შერჩევა ხორციელდება დასწავლის ალგორითმებისაგან დამოუკიდებლად და წარმოადგენს წინასწარი დამუშავების საფეხურს. იგი საკმაოდ სწრაფია, მაგრამ ის კრიტერიუმები, რომლებიც შეირჩა წინასწარი დამუშავებისას, შესაძლებელია არ გამოდგეს დასწავლის ალგორითმის უკუსვლის პროცესის დროს.
- შეფუთვის მოდელი ძირითადად მოიცავს შერჩევის პროცესის დროს პროგნოზირების ოპტიმიზაციას. თუკი ფილტრაციის მეთოდები ეყრდნობიან დასასწავლი მონაცემების ძირითად მახასიათებლებს იმისათვის, რომ ამოკრიბონ თვისებები პროგნოზირებისაგან დამოუკიდებლად, შეფუთვის მოდელი გამოიყენება მასწავლებლით (Superwised) დასწავლის დროს. იგი საჭიროებს დიდ გამოთვლებს. თვისებათა ქვესიმრავლის შერჩევა ხდება სწავლებისას გამოყენებული დასწავლის ალგორითმზე დაყრდნობით.

იმ შემთხვევაში, როდესაც მახასიათებლების რაოდენობა ძალიან დიდია, ფილტრაციის მოდელი შეუცვლელია თვისებათა შემცირებული სიმრავლის მისაღებად. არსებობს თვისებათა შერჩევის რამდენიმე მეთოდი: "დოკუმენტების სიხშირე" (Document Frequency), „ინფორმაციის დაგროვება" (Information Gain), „გადაწყვეტილებათა ხე" (Decision Tree), „საერთო ინფორმაცია" (Mutual Information), „ნეირონული ქსელები" (Neural Netvork), χ^2. (მათი დეტალური აღწერა მოცემულია სტატიებში: (50), (51), (52)). ეს მეთოდები განეკუთვნებიან „ზღვრული მეთოდების" კლასს იმიტომ, რომ მათი გამოყენებისას ტექსტებიდან ამოიღება ის თვისებები, რომელთა სიხშირეებიც მეტია ან ნაკლებია წინასწარ განსაზღვრულ სიდიდეზე (53). ეს მეთოდები მეტად პოპულარულია მათი სისწრაფის გამო, თუმცა აქვთ ნაკლიც: დოკუმენტისათვის მაღალი სიხშირის ტერმინი ცუდი შემფასებელია კლასიფიკაციის პროცესში. ეს პრობლემა ცნობილია, როგორც „თვისებათა ჩალაგებულობა" (Feature nesting) (54).

თვისებათა შერჩევის საკითხი საკმაოდ აქტუალურია სხვადასხვა კვლევაში. მათი გამოყენება კლასიფიკაციის სხვადასხვა ალგორითმებში სხვადასხვა შედეგს იძლევა. მაგალითად: ბაიესის ალგორითმით კლასიფიკაციის ამოცანაში თვისებათა შერჩევისათვის გამოიყენეს „ინფორმაციის დაგროვება", ხოლო ბინარული კლასიფიკაციისათვის - „გადაწყვეტილებათა ხეები" (55); ნეირონული ქსელებით კლასიფიკაციისას თვისებათა შერჩევა განხორციელდა „საერთო ინფორმაცია" და χ^2 მეთოდებით (56), (57) და. ა. შ.

მიუხედავად მრავალი ექსპერიმენტებისა, კვლავ პრობლემად რჩება პასუხი კითხვაზე, თუ რა დონემდეა შესაძლებელი თვისებათა შერჩევა, რომ არ მოხდეს სასარგებლო ინფორმაციის დაკარგვა და გაუმჯობესდეს კლასიფიკატორის სიზუსტე. იმის მიუხედავად, რომ კლასიფიკაციის ამოცანებში თვისებათა შერჩევის სხვადასხვა მეთოდი გამოიყენეს, კლასიფიკაციის ამოცანა დიდი ზომის ტექსტებისათვის არ განხორციელებულა. ეს, ნაწილობრივ, იმის გამოა, რომ კლასიფიკაციის ალგორითმები მაღალ განზომილებიან ნიშან-თვისებათა სივრცეში არ რეალიზდება. კვლევა, რომელიც განხორციელდა ყველაზე დიდი რაოდენობის თვისებებით (6-დან 180-მდე), გადაწყვეტილებათა ხის ალგორითმისათვის იყო. თუმცა, ტექსტების კლასიფიკაციის ამოცანისათვის ამგვარი მასშტაბის ანალიზი რეალობიდან საკმაოდ შორსაა (58).

სამეცნიერო კვლევების ძირითადი ნაწილი, რომელიც ტექსტების კლასიფიკაციის ამოცანას განიხილავს, თვისებათა შერჩევისათვის იყენებს ტერმინის წონის TF-IDF დათვლის სქემას, რომელიც წინა თავში იყო განხილული.

მეორე თავის დასკვნა

მეორე თავში აღვწერეთ კლასიფიკაციის პროცესი, როგორც ინფორმაციული ძებნის ერთ-ერთი ქვეამოცანა. ინფორმაციის ძებნის პროცესი არ წარმოადგენს მხოლოდ ერთი სახის ოპერაციის შედეგს. მისი წარმატებულობა და რელევანტურობა დამოკიდებულია ძებნის ციკლის ადექვატურობაზე და სისრულეზე. ამ ციკლში ერთ-ერთი მნიშვნელოვანი ადგილი უკავია კლასიფიკაციის ეტაპს, რომლითაც, როგორც წესი, იწყება ძებნის პროცესი.

ავღწერეთ ტექსტის საწყისი დამუშავების პროცესები, რომელთა განხორციელება აუცილებელია კლასიფიკაციის საწყის ეტაპზე. განვიხილეთ სტემინგისა და ლემატიზაციის პროცესი, რომელიც წარმოადგენს დოკუმენტების დამუშავების უმნიშვნელოვანეს ეტაპს. ვისაუბრეთ სტემინგის პოპულარულ ალგორითმებზე-ლოვინსის, პორტერის და პაის/ჰასკის ალგორითმებზე, რომლებიც შეიქმნა ძირითადად ინგლისური და მორფოლოგიურად მისი მსგავსი ენებისათვის. აღვწერეთ ალგორითმების შემადგენელი ეტაპები. ამასთანავე, განვიხილეთ სტემინგის ალგორითმების გამოყენების თავისებურებანი ანალიზურ და სინთეზურ ენებში და, ამ თავისებურებათა გათვალისწინებით, მათი მოდიფიკაციის აუცილებლობა სხვადასხვა ენის კორპუსისათვის, თუმცა არსებობს ისეთი ენებიც, რომელთა დამუშავება მოითხოვს საერთოდ ახალი სტემერის შექმნას.

სტემინგის პროცესის მიზანია კლასიფიკაციისათვის მნიშვნელოვან თვისებათა შერჩევა. განსხვავებულია თვისებათა შერჩევის მეთოდები, რომლებიც უზრუნველყოფენ თვისებათა მრავალგანზომილებიანი სივრცის ფორმირებას. ამ პროცესში უმნიშვნელოვანესია ამ სივრცის განზომილების შემცირება, რომელიც სხვადასხვა გზით ხორციელდება. თუმცა, მიუხედავად მრავალიცხოვანი სამეცნიერო კვლევების და ექსპერიმენტებისა, კვლავ პრობლემად რჩება ნიშან-თვისებათა შერჩევის ზღვრული რაოდენობის განსაზღვრა, რომელიც შესაძლებლია წარმატებით იქნას გამოყენებული ძებნის პროცესისათვის.

3. ბუნებრივი ენის დამუშავების მეთოდები კლასიფიკაციის ამოცანებში

კონცეპტებზე დაფუძნებული ინფორმაციული ძებნა

ინფორმაციული ძებნის სისტემები ტრადიციულად დამოკიდებულია სიტყვათა შეთანხმებაზე დოკუმენტების ინდექსირების და ძებნის პროცესის განსახორციელებლად. ასეთი მიდგომა, დიდი ალბათობით, არაზუსტია. ზოგადად, სიტყვათა შეთანხმება უფრო სემანტიკურია, ვიდრე სინტაქსური და იგი მოითხოვს „ადამიანურ ცოდნას“. საერთო მიდგომებზე დაფუძნებული ინფორმაციული ძებნის მეთოდები ამ სირთულეს ხელით შექმნილი ლექსიკონებით ან არამნიშვნელოვანი სიტყვების დოკუმენტიდან მოშორებით ეჭიდებიან.

ადამიანის ინფორმაციული მოთხოვნა განისაზღვრება კონცეპტუალურ სივრცეში. აქედან გამომდინარე, გასაღებ სიტყვებზე დაფუძნებული ძებნისას კონცეპტების გამოყენება საკმაოდ ეფექტურია. კონცეპტის აგების ერთ-ერთი მიდგომა სემანტიკური კავშირების გამოყენებაა. კონცეპტებით ძებნა ერთის მხრივ უზრუნველყობს შედეგის რელევანტურობას, ხოლო მეორე მხრივ, კონცეპტის არდამთხვევის შემთხვევაში, ამცირებს არარელევანტური დოკუმენტების დაბრუნების ალბათობას.

კონცეპტი წარმოადგენს მენტალურ სტრუქტურას. სიტყვები და ფრაზები კონცეპტის ლინგვისტური წარმოდგენაა. ბუნებრივი ენის თავისებურებებიდან გამომდინარე, ერთი და იგივე სიტყვა შეიძლება სხვადასხვა შინაარსით შევიდეს სხვადასხვა კონცეპტში, ან სხვადასხვა სიტყვით შესაძლებელია ერთნაირი (ერთი დონის) ან ერთმანეთთან სემანტიკურად ახლოს მდგომი კონცეპტების შექმნა, ხოლო შემდეგ ონტოლოგიის[8] გამოყენებით „გასაღები“ კონცეპტის ფორმირება.

ინფორმაციული ძებნის ყველაზე მნიშვნელოვანი პრობლემა სინონიმია[9] და პოლისემიაა[10], რომლებიც სხვადასხვა კვლევაში განსხვავებულად გადაიჭრება.

მკვლევარების მიერ სინონიმების პრობლემა გადაიჭრა იმით, რომ მომხმარებლის მიერ ჩაწერილი მოთხოვნა გაფართოვდა არსებული ტერმინების შესაბამისი სინონიმებით (59). თუმცა, ხშირად, მოთხოვნაში და საძიებო დოკუმენტში წარმოდგენილი ტერმინები სცილდება სინონიმიის ჩარჩოებსაც. მეთოდი გაუმჯობესდა შემდეგი პრინციპით: იდენტიფიცირდა ის ტერმინები, რომლებიც დაემთხვა მოთხოვნისა და რანჟირებული სიის ყველაზე მაღალი რელევანტურობის ხარისხის დოკუმეტის ტერმინებს (60), მაგრამ ეს მიდგომა, ძებნის სისტემის შედეგის გაუარესების თავიდან ასაცილებლად, მოითხოვს ტერმინების სიის ხელით შექმნას, რაც მის ნაკლს წარმოადგენს.

[8] ონტოლოგია - მეცნიერება ყოფიერების ანუ არსებობის, არსის, სტრუქტურისა და კანონზომიერების შესახებ.

[9] სინონიმია - ერთსახელიანობა. სხვადასხვა სიტყვის მსგავსება მნიშვნელობის მიხედვით.

[10] პოლისემია - მრავალმნიშვნელობიანი. მრავალმნიშვნელიანობა სიტყვისა (სიტყვას რომ ერთზე მეტი, რამდენიმე მნიშვნელობა აქვს).

პოლისემიის პრობლემა გადაწყდა Wordnet თეზაურუსის შექმნით, რომელმაც შესაძლებელი გახადა მოძებნილიყო სიტყვის ყველა შესაძლო მნიშვნელობა მისი სწორი კონტექსტით გამოყენებისათვის (61).

ინფორმაციულ ძებნაში კონცეპტების გამოყენება ერთ-ერთი ალტერნატიული მიდგომაა, რომლის მიზანიც აღნიშნული პრობლემების გადაწყვეტაა. კონცეპტებით ძებნა მოიცავს ტექსტის სემანტიკურ ანალიზს.

ბუნებრივი ენის ანალიზთან დაკავშირებული სტატისტიკური მეთოდების პრობლემების გადაჭრა განხორციელდა სემანტიკური ძებნის მოდელებით. ამ მოდელებმა განსაკუთრებული როლი ითამაშეს ბუნებრივი ენის ტექსტების ანალიზის პროცესში.

3.1. ლატენტური სემანტიკური ანალიზი (LSA)

ლატენტური სემანტიკური ანალიზი იყენებს სტატისტიკურ მიდგომებს ძებნის პროცესში, მაგრამ აღწერილი მეთოდებისაგან განსხვავებით, მისი მიზანია მოთხოვნის და დოკუმენტის ფორმირება ისეთ დონეზე, რომ მათ შინაარსი გასაგები იყოს (62). ეს მეთოდი უზრუნველყოფს ტერმინების მსგავსებაზე დამყარებული სტატისტიკური ძებნის მოდელების პრობლემების გადაჭრას. ერთ-ერთი პრობლემაა სინონიმებისა და ომონიმების არსებობა. მომხმარებელი აყალიბებს თავის მოთხოვნას კონცეპტუალურ სივრცეში, სადაც სხვადასხვა სიტყვა შეიძლება ერთნაირი სემანტიკის იყოს ან პირიქით, ერთი და იგივე სიტყვა იყოს განსხვავებული სემანტიკის სხვადასხვა კონცეპტუალურ სივრცეში. ბუნებრივია, ამ დროს, ძებნის ეფექტური შედეგის მისაღებად, ტერმინების მსგავსების გამოყენება ნაკლებად ეფექტურია.

ლატენტური სემანტიკური ანალიზი მნიშვნელოვანია არა მარტო ლინგვისტური კვლევების თვალსაზრისით, არამედ მარკეტინგული კვლევებისთვისაც. უკანასნელ პერიოდში სულ უფრო ხშირად გამოიყენება ახალი პროდუქციის შესასწავლად ე. წ. შეფასებითი ტექსტების შემცველი კითხვარები. ასეთი ტიპის კითხვარებში რესპოდენტები აფასებენ პროდუქციას არა წინასწარ მოცემული პარამეტრების ნაკრების და მათი ქულობრივი მნიშვნელობით, არამედ აღწერენ თავიანთ მოსაზრებას პროდუქტის ხარისხზე, მის დადებითსა და უარყოფით თვისებებზე. ასეთი სახით მიღებული დოკუმენტების დამუშავების პროცესი უშუალო კავშირშია ტექსტის ლატენტურ სემანტიკურ ანალიზთან.

ტექსტის ლატენტური სემანტიკური ანალიზი ასევე მნიშვნელოვანია სხვადასხვა ექსპერტული სისტემისა და გადაწყვეტილების მხარდამჭერი საინფორმაციო სისტემების ცოდნის ბაზის შემუშავებისთვის. რიგ დარგებში ექსპეტული ცოდნის ამოღება გართულებულია იმის გამო, რომ მისი ფორმალიზების შესაძლებლობის ხარისხი დაბალია. ექსპერტის შეფასება, რიგ შემთხვევაში, წარმოადგენს ტექსტს, რომელიდანაც ცოდნის ამოსაღებად აუცლებელია მისი ლატენტური სემანტიკური ანალიზი.

ლატენტური სემანტიკური ანალიზის მეთოდი განსაზღვრავს დიდი მოცულობის მონაცემების სტატისტიკური მეთოდებით დამუშავების შედეგად მიღებულ ტერმინებს შორის სემანტიკურ კავშირებს (63). მეთოდის თანახმად, მონაცემების სემანტიკა, ტერმინების შემთხვევითობის პრინციპზე შერჩევის შემთხვევაში, ნაწილობრივ არაცხადი, დაფარული ხდება. მეთოდი უზრუნველყოფს ერთი და იმავე სიტყვის

აზრობრივი განსხვავებების გარკვევას შინაარსთან მიმართებაში და სრულად იღებს ლატენტურ სემანტიკას. ეს პროცესი მთლიანად ავტომატიზირებულია და არ მოითხოვს ხელით შექმნილი ლექსიკონების გამოყენებას. იგი ემყარება Singular-Value Decomposition პრინციპს (64). ლატენტური სემანტიკური ანალიზის შედეგად მიღებული „ფუნქციონალური ბირთვი“ წარმოადგენს დიდი ზომის მატრიცას, რადგან მის ფორმირებაში მონაწილეობს კოლექციაში არსებული ყველა განსხვავებული ტერმინი.

A მატრიცის სტრიქონები შეესაბამებიან i-ურ ტერმინებს, ხოლო სვეტები - j დოკუმენტებს, მატრიცის ელემენტი წარმოადგენს i-ური ტერმინის j -ურ დოკუმენტში შეხვედრის რაოდენობას.

A მატრიცის გახლეჩვით მიიღება რამდენიმე მატრიცა:

$$A = U\Sigma V^T \tag{21}$$

სადაც U და V წარმოადგენენ ორთოგონალურ მატრიცებს, ხოლო Σ არის დიაგონალური მატრიცა, რომლის ელემენტებიც წარმოადგენენ სინგულარულ რიცხვებს, რომელთა დალაგებაც შემდგომში ხდება მაგნიტუდის მიხედვით. ამ მატრიცის პირველი k მნიშვნელობა A მატრიცის ლატენტური სემანტიკაა .

რაც უფრო დიდია კოლექცია, მით უფრო დიდი ზომისაა მატრიცა და, შესაბამისად, კლასიფიკაციის პროცესს უფრო დიდი დრო ესაჭიროება. რაც რიგი კლასიფიკაციის ამოცანებისათვის შემაფერხებელია.

მეთოდის დადებითი მხარეა ის, რომ მოთხოვნის შედეგად შესაძლებელია ისეთი რელევანტური დოკუმენტების დაბრუნება, რომლებიც მოთხოვნილ ტერმინს არ შეიცავენ. იგი მუშაობს სტრუქტურირების უფრო ღრმა დონეებზე, ვიდრე სტატისტიკური მოდელები. ამ მოდელის განსხვავება ვექტორული სივრცის მოდელისაგან სივრცის შემცირებული განზომილებაა, რომელიც მიიღწევა ინდექსირებული ტერმინების რაოდენობის შემცირებით.

3.2. ზუსტი სემანტიკური ანალიზი (ESA)

ზუსტი სემანტიკური ანალიზი ეფუძნება ადამიანის მიერ შექმნილი ცოდნის ბაზიდან ავტომატურად ამოღებული ტერმინების საფუძველზე კონცეპტების ფორმირებას (65). ამ მეთოდის გამოყენებით, ინფორმაციის ძებნის მაღალი ხარისხის სიზუსტისათვის, ტექსტიდან მაღალი მნიშვნელოვნების თვისებებს ავტომატურად იღებენ. კლასიკური მეთოდებისაგან განსხვავებით, ამ მეთოდით ხორციელდება დაჭდევებული დასასწავლი მონაცემების ავტომატური გენერაცია და შედეგად მიიღება მაღლგანზომილებიანი კონცეპტების სივრცე. ეს მეთოდი წარმტებით გამოიყენეს ძებნის ამოცანებისათვის სხვადასხვა სამეცნიერო კვლევაში (66), (67), (68).

ვიკიპედიაზე დაფუძნებულ ზუსტ სემანტიკურ ანალიზში სიტყვის სემანტიკა აღიწერება ვექტორით, რომელიც გასაზღვრავს სიტყვის მნიშვნელოვნებას კონცეპტში. კონცეპტი გამომუშავდება ვიკიპედიის ერთი სტატიიდან და ის წარმოადგენს იმ ტერმინების tf-idf წონითი ვექტორების სიმრავლეს, რომლებიც გვხვდება სტატიაში. კონცეპტის ფორმირების შემდეგ, მასში შემავალი თითოეული ტერმინისათვის შექმნილი

ინვერტირებული ინდექსები მიმართავს იმ კონცეპტებს, რომლებთანაც ის ასოცირდება. ტერმინის წონა წარმოადგენს კონცეპტსა და ტერმინს შორის შესაბამისობის ხარისხს.

ESA გამოირჩევა საკმაო მოქნილობით ტექსტების კლასიფიკაციისას. სემანტიკის ამოსაცნობად იგი არაა დამოკიდებული ტექსტების სტრუქტურაზე, მუშაობს ადამიანის ცნობიერების და არა დოკუმენტის ან სიტყვის ლექსიკურ დონეზე. იგი განსაზღვრავს ტექსტის შინაარსს დოკუმენტიდან ამოღებული კონცეპტების საფუძველზე, ამიტომ ის ფართოდ გამოიყენება სემანტიკური კავშირების დასადგენად კლასიფიკაციის ამოცანებში.

ლატენტური სემანტიკური ანალიზისგან განსხვავებით, ეს მეთოდი იყენებს ვიკიპედიის კოლექციას და შედეგად იძლევა ვექტორების სახით წარმოდგენილ კონცეპტს. კლასიფიკაციის პროცესი უფრო სწრაფია, შედეგებიც საკმაოდ მაღალი. მიუხედავად მისი ეფექტურობისა, მისი გამოყენება ქართული ტექსტებისათვის თითქმის შეუძლებელია საწყისი წყაროს, ვიკიპედიის, სიმწირის გამო.

მესამე თავის დასკვნა

განვიხილეთ ბუნებრივი ენის დამუშავების მეთოდები კლასიფიკაციის ამოცანებში, კერძოდ, კონცეპტებზე დაფუძნებული ინფორმაციული ძებნა. კონცეპტი წარმოადგენს მენტალურ სტრუქტურას. ტერმინები არის კონცეპტების ლინგვისტური წარმოდგენა. ძებნის პროცესში კონცეპტების გამოყენება საკმაოდ ეფექტურია. კონცეპტის აგების ერთ-ერთი დადებითი მხარეა სემანტიკური კავშირების გამოყენება რელევანტური შედეგის მისაღებად, ხოლო მეორე მხრივ არარელევანტური დოკუმენტების დაბრუნების ალბათობის შემცირება კონცეპტის არდამთხვევის შემთხვევაში.

აღვწერეთ ინფორმაციული ძებნის ორი მნიშვნელოვანი პრობლემა: სინონიმია და პოლისემია, გადაჭრის სხვადასხვა მეთოდი და მათი ქმედითი ასპექტები. აღნიშნული მეთოდებიდან გამოვყავით ლატენტური სემანტიკური და ზუსტი სემანტიკური ანალიზის მეთოდები, როგორც საუკეთესო აღნიშნული პრობლემების გადასაჭრელად. ლატენტური სემანტიკური ანალიზი იყენებს სტატისტიკურ მიდგომებს ძებნის პროცესში. იგი უზრუნველყოფს ტერმინების მსგავსებაზე დამყარებული სტატისტიკური ძებნის მოდელების პრობლემების გადაჭრას. ეს მეთოდი მნიშვნელოვანია არა მარტო ლინგვისტური კვლევების თვალსაზრისით, არამედ მარკეტინგული კვლევებისთვისაც. ტექსტის ლატენტური სემანტიკური ანალიზი ასევე მნიშვნელოვანია სხვადასხვა ექსპერტული სისტემისა და გადაწყვეტილების მხარდამჭერი საინფორმაციო სისტემების ცოდნის ბაზის შემუშავებისთვის. იგი განსაზღვრავს დიდი მოცულობის მონაცემების სტატისტიკური მეთოდებით დამუშავების შედეგად მიღებულ ტერმინებს შორის სემანტიკურ კავშირებს.

ზუსტი სემანტიკური ანალიზი ეფუძნება ადამიანის მიერ შექმნილი ცოდნის ბაზიდან ავტომატურად ამოღებული ტერმინების საფუძველზე კონცეპტების ფორმირებას. ამ მეთოდით ხორციელდება დაჭდევებული დასასწავლი მონაცემების ავტომატური გენერაცია და შედეგად მიიღება მაღალგანზომილებიანი კონცეპტების სივრცე.

ორივე მეთოდი ფოკუსირებულია სემანტიკურ ძებნაზე, რომელიც დაფუძნებულია დოკუმენტისა და მოთხოვნის სემანტიკურ შესაბამისობაზე, ეს პროცესი ხორციელდება ბუნებრივი ენის ანალიზის მეთოდებით და მაღალი ხარისხით განსაზღვრავს დაბრუნებული დოკუმენტების მოთხოვნასთან სემანტიკურ მსგავსებას. მიუხედავად აღნიშნული თვისებების, არსებობს გარკვეული პრობლემები მეთოდების რეალიზაციისას. ერთ-ერთი პრობლემა, რომელიც ახასიათებს ლატენტურ სემანტიკურ ანალიზს, არის ნიშან-თვისებათა სივრცის მაღალი განზომილება. ნიშან-თვისებათა საწყისი მატრიცის ფორმირება ხორციელდება ტექსტში არსებული ყველა ტერმინის საფუძველზე, ამიტომ, ბუნებრივია, ტექსტის ზომის გაზრდასთან ერთად იზრდება მატრიცის ზომაც, რაც ართულებს ძებნის პროცესს. რაც შეეხება ზუსტ სემანტიკურ ანალიზს, მისი რეალიზაცია გართულებულია ქართულენოვანი ტექსტებისათვის, რომლის მიზეზადაც შეიძლება დავასახელოთ ქართულენოვანი ვიკიპედიის სიმწირე.

4. მანქანური სწავლება და კლასიფიკაციის მეთოდის შერჩევა/ფორმირება

დოკუმენტების ავტომატური კლასიფიკაცია, ინფორმაციული ძებნის ერთ-ერთი ძირითადი მიზნის ჭრილში შეიძლება განიმარტოს, როგორც ბუნებრივი ენის ანალიზის სახით წარმოდგენილი მანქანური სწავლების შედეგი.

უკანასკნელი წლების განმავლობაში შინაარსზე დაფუძნებულმა ინფორმაციული ძებნის ამოცანებმა, ელექტრონულ ფორმატში წარმოდგენილი დოკუმენტების ხელმისაწვდომობის და საჭიროების მოთხოვნილების ზრდასთან ერთად, საკმაოდ დიდი მნიშვნელობა შეიძინეს.

ტექსტების კლასიფიკაცია ადრეულ 60-იან წლებში დაიწყო. 80-იან წლებამდე ამ ამოცანის გადაწყვეტისათვის ყველაზე პოპულარული მიდგომა იყო ცოდნის ინჟინერია (69), რომელიც იყენებდა ხელით შედგენილ წესებს ტექსტების კატეგორიზაციისათვის. 90-იან წლებში ამ მიდგომამ დაკარგა პოპულარობა მანქანური სწავლების განვითარების გამო.

ციფრული დოკუმენტები, რომლებიც წარმოადგენენ შემავალ ინფორმაციას ინდექსირების პროცესისათვის, წარმოდგებიან როგორც ბაიტების ნაკრები ფაილში ან ვებ გვერდზე. საწყის ეტაპზე ხდება მათი გარდაქმნა სიმბოლოთა წრფივ თანმიმდევრობში. ბუნებრივია, შემავალი ტექსტი სხვადასხვა კოდირებისაა და საჭიროა სწორი კოდირების განსაზღვრა. ეს პროცესი შესაძლებელია მანქანური სწავლების ამოცანის ინტერპრეტაციით. პრაქტიკაში ამ ამოცანის გადაწყვეტა ხდება ევრისტიკული მეთოდებით.

მანქანური სწავლების მიზანი არის ისეთი ალგორითმების შემუშავება, რომელთაც შეუძლიათ ექსპერტების მიერ შემუშავებული ცოდნის ბაზის ცვლილება სიტუაციების შესაბამისად, აგრეთვე, ინფორმაციული ძებნის ისეთი პროცესების ავტომატიზაცია, როგორიცაა კლასიფიკაცია, მომხმარებლის მოთხოვნის ფორმალიზაცია და ა. შ. მათ შეუძლიათ პროცესების „შემსუბუქება“ და ადამიანის მიერ დაშვებული შეუსაბამობების აღმოფხვრა.

მანქანური სწავლება, წინასწარ კლასიფიცირებული დოკუმენტების კატეგორიების დასწავლის საფუძველზე, ეფუძნება კლასიფიკატორის ავტომატურად აგების პროცესს. მანქანური სწავლების მეთოდების გამოყენებით უფრო სწრაფია ავტომატური ფორმირება ისეთი მოდელებისა, რომლებიც მოცულობითი და რთული მონაცემების ხარისხიან ანალიზს მოახდენს. იგი, ადამიანის ჩარევის გარეშე, მოსალოდნელ შედეგსაც განსაზღვრავს და, შესაბამისად, გადაწყვეტილების მიღების შესაძლებლობასაც.

მანქანური სწავლების მეთოდებით შესაძლებელია:

- ისეთი სისტემების შემუშავება, რომლებსაც შეუძლიათ თავისი თავის ადაპტირება და შეცვლა ინდივიდუალური მომხმარებლისათვის (ნიუსები და ფილტრები);
- ახალი ცოდნის ამოღება მოცულობითი ბაზებიდან;
- გარკვეული მონოტონური ამოცანებისა და ადამიანის სხვა უნარების იმიტირების შესაძლებლობა, რომელიც გარკვეულ ინტელექტს მოითხოვს (ხელნაწერის ამოცნობა);

– ძალიან რთული სისტემების შემუშავება (სისტემები, რომლებიც მოითხოვენ სპეციფიურ უნარებს შესაბამისი ამოცანების შესასრულებლად ე.წ. ,,ცოდნის ინჟინერია“);

არსებობს მანქანური სწავლების სამი ტიპი: ინდუქციური, დედუქციური, სწავლება გამყარებით (reinforcement).

ინდუქციურია სწავლება „პრეცედენტებით“ (ობიექტი-პასუხი), ამ შემთხვევაში მოცემულია ობიექტების სიმრავლე და პასუხების შესაძლო ვარიანტები, უცნობია მათ შორის კავშირები. ცნობილია მხოლოდ „პრეცედენტების“ სასრული რაოდენობა, რომელსაც ჰქვია „დასასწავლი სიმრავლე“, რომლის საფუძველზეც ხდება ალგორითმის აგება, რომელიც უზრუნველყობს სისტემის მიერ ნებისმიერი ობიექტისათვის შესაბამისის პასუხის განსაზღვრას, რაც შეიძლება მეტი სიზუსტით.

დედუქციური სწავლება ემყარება ექსპერტების ცოდნის ფორმალიზაციას და მის საფუძველზე ცოდნის ბაზის აგებას. იგი გამოიყენება ექსპერტულ სისტემებში.

„სწავლება გამყარებით“ ხასიათდება არა სწავლების მეთოდებით, არამედ სწავლების „პრობლემებით“. ყველა მეთოდი, რომელიც შეეხება ამ პრობლემების გადაჭრას, არის „გამყარებული“ მეთოდი. კონკრეტული სიტუაციიდან გამომდინარე, იგი წინასწარ განსაზღვრავს სისტემის მოსალოდნელ ქცევას (70).

მანქანური სწავლების ალგორითმები ფართოდ გამოიყენება ინფორმაციული ძებნის ამოცანებში (მაგ. ტექსტების კლასიფიკაცია).

დღეისათვის კლასიფიკაციის ამოცანა შეიძლება განხილულ იქნას, როგორც მანქანური სწავლებისა და ინფორმაციული ძებნის მეთოდების ერთობლიობა. რთულია იმის განსაზღვრა, თუ სადაა ზუსტი საზღვარი ამ ორ სფეროს შორის.

განსაზღვრული იქნა მანქანური სწავლების კვლევების ხუთი ძირითადი პარადიგმა (71), რომლებიც გამოიყენება ინფორმაციული ძებნის სწავლების მიმართულებით. ეს პარადიგმებია: ინდუქციურობის წესი (rule-induction), ეგზემპლარზე დაფუძნებული სწავლება (instance-base learning), ნეირონული ქსელები, გენეტიკური ალგორითმები, ანალიტიკური სწავლება. აქედან პირველი ოთხის შესწავლა ხდება მარტივი სტრუქტურებით - კონცეპტებით, რომლებიც უფრო ხშირად აღიწერება სიმბოლოებით ან რიცხვითი ატრიბუტებით. ევრისტიკული მეთოდებით ხდება ამ სტრუქტურების რეალიზაცია. ისინი შეიცავენ არაცხად კავშირებს მონაცემებს შორის. ეს ოთხი პარადიგმა გამოიყენება „ინტელექტუალურ საინფორმაციო ძებნის სისტემებში“, რომლებშიც ობიექტების (დოკუმენტების) აღწერა ხორციელდება მარტივი თვისებებით, როგორიცაა „ტერმინი-სიხშირე“ ზომა. მეხუთე პარადიგმა-ანალიტიკური სწავლება, ძირითადად მოიცავს ტიპიური სიტუაციებისათვის წესების და განმარტებების შესწავლას საბაზო ცოდნის გამოყენებით.

4.1. მანქანური სწავლების ალგორითმები

მონაცემების კლასიფიკაცია მანქანური სწავლების ერთ-ერთი ამოცანაა, რომლის გადასაჭრელადაც იგი ინტენსიურად იყენებს სხვადასხვა მათემატიკურ მეთოდებს.

კლასიფიკაციის ამოცანა ხორციელდება ინდუქციური სწავლების მეთოდით. ამისათვის აქტიურად გამოიყენება მანქანური სწავლების სხვადასხვა ალგორითმები: მხარდამჭერი ვექტორების ალგორითმი (Support Vector Machines(SVM)) (72), K უახლოესი მეზობლის ალგორითმი (K-nearest neighbor KNN) (73), ბაიესის ალგორითმი (Bayesian classifier), გადაწყვეტილებათა ხეები (Decision Tree) (74), ნეირონული ქსელები (Neural Networks), გენეტიკური ალგორითმები (Genetic Algorithms) და ა.შ. მათი ეფექტურობა ფასდება ექსპერიმენტების საფუძველზე. თითოეულ მათგანს გააჩნია თავისი დადებითი და უარყოფითი მხარეები, რაც სხვადასხვა ფაქტორიებითაა განპირობებული. ზოგადად, იმის განსაზღვრა, თუ რომელი ალგორითმია უფრო კარგი, შეუძლებელია.

თუ გადავხედავთ სხვადასხვა მკვლევარის მიერ ჩატარებულ კვლევებს და ექსპერიმენტებს, ნათლად ჩანს, რომ კატეგორიზაციის შედეგების შეფასება, ერთი და იმავე ალგორითმის გამოყენების შემთხვევაში, სხვადასხვა ენისათვის განსხვავებულია. არსებობს კლასიფიკატორების შედარების ორი მეთოდი: პირდაპირი შედარება და ირიბი შედარება (75). პირველ შემთხვევაში კლასიფიკატორების შედარება ხდება ერთსა და იმავე კოლექციაზე, ერთი და იმავე მკვლევარის მიერ, ერთანირ პირობებში, ხოლო მეორე შემთხვევაში, კლასიფიკატორების შედარება ხდება ერთი და იმავე მკვლევარის მიერ სხვადასხვა კოლექციაზე, სხვადასხვა პირობებით ჩატრებული კვლევების შედეგების ანალიზით. ორივე მეთოდის შემთხვევაში შედეგები, ნებისმიერ სიტუაციაში, განსხვავებულია.

ბუნებრივი ენების თავისებურებების გათვალისწინებით განხორციელდა კლასიფიცირების ამოცანების გადაჭრა მანქანური სწავლების სხვადასხვა ალგორითმით სხვადასხვა ენისათვის, რიგ შემთხვევებში მოხდა მათი მოდიფიცირება შედეგების გაუმჯობესების მიზნით.

განვიხილოთ მანქანური სწავლების რამდენიმე ალგორითმი.

4.1.1. K-უახლოესი მეზობლის ალგორითმი (KNN)

K უახლოესი მეზობლის ალგორითმი ყველაზე მარტივი ალგორითმია, რომელიც გამოიყენება მანქანურ სწავლებაში. იგი უზრუნველყოფს დოკუმენტის მიკუთვნებას იმ კლასისათვის, რომელთანაც უფრო ახლოსაა k მეზობელი კლასებიდან ნიშან-თვისებათა მრავალგანზომილებიან სივრცეში. დასწავლის პროცესში უბრალოდ იმახსოვრებს ყველა ვექტორს და მათ შესაბამის კლასს, ხოლო რეალურ მონაცემებთან მუშაობისას, კლასების გამოცნობისათვის, იგი გამოითვლის ორ ვექტორს შორის მანძილს ევკლიდურ სივრცეში და k-ს (მეზობელი კლასების რაოდენობა) განსაზღვრის შემდეგ დოკუმენტის მიაკუთვნებს იმ კლასს, რომელიც ყველაზე უფრო ახლოსაა.

$$d(x;y) = \sqrt{\sum_{i=1}^{K}(x_i - y_i)^2} \tag{22}$$

KNN-ი, მუშაობის პრინციპებიდან გამომდინარე, მკვეთრად განსხვდება კლასიფიკაციის სხვა ალგორითმებისაგან. მეთოდში დასწავლის პროცესი K-ს

განსაზღვრაზე და დოკუმენტების წინასწარ დამუშავებაზე დაიყვანება. მაგრამ თუ K წინასწარ განსაზღვრულია და დოკუმენტების დამუშავება (ლექსემებად დაყოფა) არ განხორციელებულა, ალგორითმში დასწავლის ფაზა ფაქტიურად არ არსებობს.

ალგორითმს გააჩნია გარკვეული დადებითი და უარყოფითი თვისებები: დადებითია ის, რომ მას არ სჭირდება არანაირი პარამეტრის შეფასება, როგორც მაგალითად ბაიესის ალგორითმს (აპრიორული და პირობითი ალბათობები) და ამავე დროს, ადვილად რეალიზებადია, მაგრამ საჭიროებს დიდ დროს შედეგის დასაბრუნებლად. იგი წრფივადაა დამოკიდებული დასასწავლი დოკუმენტების სიმრავლის ზომაზე, რადგანაც კლასიფიკაციისას იგი ითვლის მანძილს სატესტო დოკუმენტსა და დასასწავლი სიმრავლის თითოეულ დოკუმენტს შორის. ამავე დროს პრობლემაა მეზობელი კლასების რაოდენობის (K -ს) სიდიდის განსაზღვრა. დიდი K-ს შემთხვევაში, მართალია, მცირდება "ხმაურის" ეფექტი კლასიფიკაციის პროცესში, მაგრამ რთულდება კლასებს შორის მკვეთრი საზღვრის დაფიქსირება (76).

ალგორითმის მთავარი პრინციპი ისაა, რომ უბრალოდ იმახსოვრებს ყველა დოკუმენტს დასასწავლი სიმრავლიდან, ხოლო შემდეგ ადარებს მათ თითოეულ სატესტო დოკუმენტს. ამის გამო ამ მეთოდს „ სწავლება დამახსოვრებით" (Memory based learning) უწოდებენ.

მანქანური სწავლებისას შედეგის გაუმჯობესებისათვის საჭიროა დასასწავლი დოკუმენტების დიდი რაოდენობა, KNN-ში კი ეს ფაქტი იწვევს ეფექტურობის გაუარესებას.

აღწერილი უარყოფითი თვისებების მიუხედავად, ალგორითმის გამოყენებით, სხვა მეთოდებთან ერთად, შესაძლებელია შედეგის სიზუსტის გაუმჯობესება (77).

4.1.2. ბაიესის ალგორითმი (NB)

ბაიესის კლასიფიკატორი მიეკუთვნება „პრეცედენტებით" სწავლის ალგორითმების ჯგუფს. იგი სტატისტიკური კლასიფიკატორია. ამ მეთოდის წარმატებით მუშაობისთვის საჭიროა დიაგნოსტიკური ინფორმაციის დიდი რაოდენობა. როდესაც ინფორმაციის მოცულობა საშუალებას გვაძლევს გამოვიყენოთ ბაიესის მეთოდი, მიზანშეწონილია მისი გამოყენება, როგორც საიმედო და ეფექტური მეთოდისა.

მეთოდი დამყარებულია ბაიესის მარტივ ფორმულაზე. თუ მოცემულია სავარაუდო შედეგი D_i და დიაგნოსტიკური ნიშნების კომპლექსი K^* ,მაშინ ბაიესის ალგორითმი ადგენს დიაგნოსტიკური ნიშნებისთვის კონკრეტული D_i შედეგის არსებობის ალბათობას.

ბაიესის ფორმულა:

$$P(D_i|K^*) = \frac{P(D_i)P(K^*|D_i)}{\sum_{s=1}^{n} P(D_s)P(K^*|D_s)} \qquad (23)$$

სადაც:

$P(D_i)$ არის კონკრეტული D_i კატეგორიის არსებობის ალბათობა. (კლასიფიკაციის შემთხვევაში ეს არის 1/n, სადაც n არის კატეგორიების რაოდენობა;

$P(K^*|D_i)$ არის კონკრეტული D_i კატეგორიისათვის K^* დიაგნოსტიკური ნიშნების არსებობის ალბათობა (ტექსტის ყოველი სიტყვის კონკრეტულ კატეგორიაში არსებობის ალბათობების ნამრავლი). მის დასათვლელად ტექსტის ყოველი სიტყვის კონკრეტულ კატეგორიაში არსებობის ალბათობა წინასწარ უნდა იყოს ცნობილი.

4.1.3. მხარდამჭერი ვექტორების ალგორითმი (SVM)

მხარდამჭერი ვექტორების ალგორითმი - SVM მიეკუთვნება კლასიფიკატორების ჯგუფს "ფართო ნაპრალით" და ეფუძნება ვექტორული სივრცის მოდელზე დამყარებულ მანქანურ სწავლებას.

SVM ახდენს ვექტორებად (წერტილებად) წარმოდგენილი დოკუმენტების სივრცის გაყოფას ჰიპერსიბრტყით, რომლის სხვადასხვა მხარეს არის სხვადასხვა კლასი (72). ეს სიბრტყე წარმოადგენს კლასიფიკატორის სიბრტყეს. ბუნებრივია, ასეთი სიბრტყე ბევრი შეიძლება გაივლოს, მაგრამ ამ შემთხვევაში აიღება ის სიბრტყე, რომლიდანაც უახლოეს წერტილებამდე დაშორება არის მაქსიმალური. ამ სიბრტყესთან ყველაზე ახლოს მდგომი ვექტორები „მხარდამჭერი" ვექტორებია. ნებისმიერი ახალი დოკუმენტი კლასიფიცირდება იმის მიხედვით, თუ სიბრტყის რომელ მხარეს ჩავარდება.

თუ დასასწავლი სიმრავლე კლასიფიკაციის განსახორციელებლად მოიცავს დოკუმენტების ორ კლასს, რომელთათვისაც შესაძლებელია განხორციელდეს წრფივი გაყოფა, მაშინ არსებობს დიდი რაოდენობა წრფივი კლასიფიკატორების, რომელთა საშუალებითაც შესაძლებელია ეს გაყოფა განხორციელდეს. ზოგიერთი სწავლების მეთოდი (როგორიცაა პერსეპტრონი) იძლევა საშუალებას, რომ განისაზღვროს თუნდაც ერთი წრფივი გამყოფი, ხოლო სხვა მეთოდები, მაგ. ბაიესი, ეძებს საუკეთესო გამყოფს კონკრეტული კრიტერიუმების გათვალისწინებით. SVM ეძებს გამყოფ საზღვარს, რომელიც ყველა წერტილიდან მაქსიმალურადაა დაშორებული.

არის შემთხვევები, როდესაც ასეთი წრფივი გაყოფა შეუძლებელია, ან საერთოდ არანაირი ტიპის გაყოფა არ ხერხდება. ამ შემთხვევაში გამოიყენება „მოქნილი საზღვრის" (Soft marging) ალგორითმი, რომელიც მოითხოვს დამატებითი პარამეტრების განსაზღვრას და ჰიპერსიბრტყის მაგივრად მრავალგანზომილებიანი გამყოფი სივრცის შემოტანას. SVM ალგორითმი ასეთ სიტუაციაში ახორციელებს უფრო მაღალგანზომილებიანი სივრცის ფორმირებას, სადაც წრფივი გაყოფა უკვე შესაძლებელი ხდება. ეს ოპერცია ცნობილია „ბირთვის ილეთის" (kernel trick) სახელით. სწორედ ეს ითვლება ამ ალგორითმის უმნიშვნელოვანეს ღირსებად.

ზოგადად, რომელიმე ალგორითმისათვის უპირატესობის მინიჭება ძნელია. მათი ეფექტურობა დამოკიდებულია სხვადასხვა ფაქტორზე. სამეცნიერო სტატიებში (78), (79) განხილულია კლასიფიკაციის ამოცანებში ყველაზე ხშირად გამოყენებადი ალგორითმების შედარებითი ანალიზი. საუკეთესო შედეგი მიიღეს SVM-ის გამოყენებისას, მას მოსდევს KNN-ი, ნეირონული ქსელები და ბაიესი.

მეოთხე თავის დასკვნა

წარმოდგენილ თავში განვიხილეთ მანქანური სწავლების ძირითადი პრონციპები და ალგორითმები. დღეისათვის კლასიფიკაციის ამოცანა შეიძლება განხილულ იქნას, როგორც მანქანური სწავლებისა და ინფორმაციული ძებნის მეთოდების ერთობლიობა. მანქანური სწავლების მიზანი არის ისეთი ალგორითმების შემუშავება, რომელთაც შეუძლიათ ექსპერტების მიერ შემუშავებული ცოდნის ბაზის ცვლილება სიტუაციების შესაბამისად, აგრეთვე, ინფორმაციული ძებნის ისეთი პროცესების ავტომატიზაცია, როგორიცაა კლასიფიკაცია, მომხმარებლის მოთხოვნის ფორმალიზაცია და ა. შ. მათ შეუძლიათ პროცესების „შემსუბუქება“ და ადამინის მიერ დაშვებული შეუსაბამობების აღმოფხვრა რომელებიც საკმაოდ ხშირად გამოიყენება კლასიფიკაციის ამოცნებში. გამოყენების თავისებურებები და მათი დამოკიდებულება სხვადასხვა ფაქტორზე.

მანქანური სწავლების ალგორითმები ფართოდ გამოიყენება ინფორმაციული ძებნის ამოცანებში (მაგ. ტექსტების კლასიფიკაცია). გამოიკვეთა ის ფაქტი, რომ რომელიმე ალგორითმის გამოყოფა, საუკეთესო შედეგის მისაღწევად, თითქმის შეუძლებელია. მათი შერჩევა ხდება კონკრეტული ამოცანებისათვის და კონკრეტული ბაზებისათვის. ცხადია ის ფაქტიც, რომ არარელევანტური და ზედმეტი თვისებები, რომლებიც ნაკლებად ინფორმატიულია კლასიფიკაციის ამოცანისათვის, ამცირებენ ალგორითის მუშაობის ეფექტურობას.

დავახასიათეთ სამი ალგორითმი, რომლებიც გამოყენებული იქნა ჩვენს მიერ განხორციელებულ კვლევებში. ესენია: უახლოესი მეზობლის ალგორითმი, მხარდამჭერი ვექტორების ალგორითმი და ბაიესის ალგორითმი. განვიხილეთ ყველა თავისებურება, რომელიც ახასიათებთ თითოეულს მუშაობის სხვადასხვა საფეხურზე.

ზოგადად, რომელიმე ალგორითმისათვის უპირატესობის მინიჭება ძნელია. მათი ეფექტურობა დამოკიდებულია სხვადასხვა ფაქტორზე.

5. კონცეპტის პატერნების ფორმირება დოკუმენტების კლასიფიკაციისათვის

5.1. კონცეპტების ფორმირების ანალიტიკური ევრისტიკების მეთოდი

როგორც მესამე თავში აღვნიშნეთ, კონცეპტებით ძებნა ყველაზე მეტად უზრუნველყობს შედეგის რელევანტურობას, ხოლო კონცეპტის არდამთხვევის შემთხვევაში, ამცირებს არარელევანტური დოკუმენტების დაბრუნების ალბათობას. კონცეპტი წარმოადგენს მენტალურ სტრუქტურას. სიტყვები და ფრაზები კონცეპტის (ცნების) ლინგვისტური წარმოდგენაა. ბუნებრივი ენის თავისებურებებიდან გამომდინარე, ერთი და იგივე სიტყვა შეიძლება სხვადასხვა შინაარსით შევიდეს სხვადასხვა კონცეპტში, ან სხვადასხვა სიტყვით შესაძლებელია ერთნაირი (ერთი დონის) ან ერთმანეთთან სემანტიკურად ახლოს მდგომი კონცეპტების შექმნა, ხოლო შემდეგ ონტოლოგიის გამოყენებით „გასაღები“ კონცეპტის ფორმირება.

ამ ფაქტის გათვალისწინებით მეთოდი, რომლის საშუალებითაც შესაძლებელი იქნება ამა თუ იმ ცნების (კონცეპტის) განზოგადოებული ჩარჩოს (პატერნის) შემუშავება, რომელიც შეიძლება წარმოდგენილ იქნას, როგორც დოკუმენტების კლასიფიკატორის ძირითადი საბაზო ელემენტი საძიებო სისტემაში.

ეს მეთოდი ეფუძნება აკად. ვლადიმერ ჭავჭანიძის მეთოდს, რომელიც ცნობილია კონცეპტების ფორმირების ანალიტიკური ევრისტიკების მეთოდის სახელით (80), (81).

აღვწეროთ ეს მეთოდი ზოგადად, როგორც ობიექტების კლასიფიკაციის ინსტრუმენტი.

მოცემულია ობიექტების კლასი C, რომელიც შედგება არაიგივური სასრული რაოდენობა ობიექტებისაგან $\{C_1, C_2, \dots C_n\}$. ყოველი C_i ობიექტი, $i = 1, \dots, n$, ხასიათდება ნიშანთვისებათა სასრული რაოდენობით $A = \{A_1, A_2, \dots A_m\}$ და შეფასებით, თუ რომელ კლასს განეკუთვნება ეს ობიექტი $C^+ \subset C$ თუ $C^- \subset C$. ყოველ ნიშანთვისებას A_j, j=1,…,m შეუძლია მიიღოს მნიშნელობა $b_{jk} \subset B, k = 1,2, \dots, n_j$. B არის $A-$ ს მნიშვნელობათა სიმრავლე.

$A_1, A_2, \dots A_m$ მნიშნელობათა დალაგებულ სიმრავლეს C_i ობიექტის აღწერის შემთხვევაში ვუწოდოთ „ტრაექტორია“. C_i ობიექტი შეიძლება ჩავწეროთ „ტრაექტორიის“ სახით :

$$C_i = \{b_1(i), b_2(i), \dots, b_m(i)\}, b_j(i) \in B,\ j = 1, 2, \dots, m \qquad (24)$$

ქვეკლასებიდან ობიექტებზე დაკვირვების შედეგად სუბიექტმა უნდა ჩამოაყალიბოს ცნება, რომელიც შეესაბამება C^+ და C^- ქვეკლასებს.

ანალიტიკური ევრისტიკების მეთოდი საშუალებას იძლევა შეფასებული ობიექტების საფუძველზე ავაგოთ პატერნი რომელიც შეესაბამება C^+ და C^-.

პატერნის აგების პროცესი შედგება შემდეგი ეტაპებისაგან:

I. **ნიშანთვისებათა ბინარიზაცია მათი „მნიშვნელობათა სივრცის“ ბინარიზაციით.** ბინარიზაციის ყველაზე ზოგად მეთოდად შეიძლება ავიღოთ სიმრავლის გაყოფა ორ

ურთიერთშემავსებელ ქვესიმრავლედ „არის“ - „არ არის“. ამ აღნიშვნებში C_i ობიექტს შეიძლება გააჩნდეს A_k ან $\overline{A}_k$, $k = 1, 2, \dots, m$.

II. **ნიშანთვისებათა გადაკოდირება.**

შემოგვაქვს რიცხვითი სიმრავლე და ალ-სიმრავლე[11]:

ნაცვლად ნიშან-თვისებათა $A = \{A_1, A_2, \dots, A_m\}$ სიმრავლისა გვექნება $N_A = \{1, 2, \dots, m\}$,

ნაცვლად მნიშნელობათა $B = \{b_{11}, b_{12}, \dots, b_{m\,n_m}\}$ სიმრავლისა – $\underline{N}_B = \left\{\overset{\vee}{1}, \overset{\vee}{2}, \dots, \overset{\vee}{n}\right\}$,

სადაც

$$\overset{\vee}{k} = \begin{cases} k \\ \bar{k} \end{cases}, \quad k = 1, 2, \dots, n;$$

„ტრაქტორიის“ $C_i = \{b_1(i),\ b_2(i), \dots, b_m(i)\}$ ნაცვლად გვექნება $\underline{N}_{C_i} = \left\{\overset{\vee}{\alpha}_1(i), \overset{\vee}{\alpha}_2(i), \dots, \overset{\vee}{\alpha}_m(i)\right\}$,

სადაც

$\overset{\vee}{\alpha}_j(i) \in \underline{N}_B$, $j = 1, 2, \dots, m$.

III. **ორთონორმირებული ბინარული მდგომარეობის ვექტორების აგება.**

შემოდის V მატრიცა განზომილებით $n \times m$ ($2n = 2^m$). ამ მატრიცის სვეტებს წარმოადგენენ მდგომარეობის ორთონორმირებული ვექტორები (ანუ ფილტრები) ψ_i, $i = 1, 2, \dots, m$, რომელიც იქმნება $\underline{N}_B$ ელემენტებით *(დაწართი-ცხრილი 6)*

ფილტრაციის ოპერაცია - ყველა C_i ტრაექტორიის ორთონორმირებულ ფილტრებზე გატარება.

ყოველ ტრაექტორიას $C_i = \left\{\overset{\vee}{\alpha}_1(i), \overset{\vee}{\alpha}_2(i), \dots, \overset{\vee}{\alpha}_m(i)\right\}$ უთანადდება მდგომარეობათა ორთონორმირებულ ვექტორთა კონიუქციური ნამრავლი

$$\varphi(C_i) = \left(\overset{\vee}{\psi}_1 \overset{\vee}{\psi}_2, \dots, \overset{\vee}{\psi}_m\right)_i, \quad i = 1, 2, \dots, n \qquad (25)$$

სადაც

$\overset{\vee}{\psi}_j = \psi_j$ - თუ ტრაექტორიის j-ური ელემენტი შედის ψ_j ვექტორში „არის“ სახით, ანუ e_j, $j = 1, 2, \dots, m$

და

$\overset{\vee}{\psi}_j = \overline{\psi}_j$, თუ ტრაექტორიის j-ური ელემენტი შედის ψ_j ვექტორში „არ არის“ სახით, ანუ $\bar{e}_j$, $j = 1, 2, \dots, m$.

[11] დავუშვათ e ელემენტის მნიშვნელობების სიმრავლე შედის A სიმრავლეში. ალ-სიმრავლე (ალგებრაიზირებული სიმრავლე) N_A განსხვავდება A სიმრავლისაგან იმით, რომ მასში შეიძლება არსებობდეს, როგორც e ელემენტი, ასევე მისი უარყოფაც ē.

IV. **დიზიუნქციური სუპერპოზიციის ოპერაცია.**

$$\varphi_+ = \bigcup_{C_i \in C^+} \varphi(C_i) - C^+ \qquad \text{კონცეპტის პატერნი} \tag{26}$$

$$\varphi_- = \bigcup_{C_i \in C^-} \varphi(C_i) - C^- \qquad \text{კონცეპტის პატერნი} \tag{27}$$

იმ შემთხვევაში, თუ

- ობიექტების რაოდენობა C^+ და C^- ქვეკლასებთან მიმართებაში საკმაოდ დიდია, ხოლო თვით ობიექტები არაიგივურია და საკმაოდ ფართოდ წარმოადგენენ შესაბამის ქვეკლასს;
- სწორადაა გამოყოფილი საკმარისი რაოდენობის ნიშანთვისებები, ზუსტადაა განსაზღვრული მათი მნიშვნელობათა სიმრავლე და ჩატარებულია ამ სიმრავლეთა „წარმატებული“ ბინარიზაცია,

მაშინ პატერნები φ_+ და φ_- შეიცავენ სრულ ინფორმაციას C^+ და C^- შესახებ (როგორც ტიპიურის, ასევე იშვიათი წარმომადგენლების ქვეკლასიდან) და არ მოდიან ერთმანეთთან წინააღმდეგობაში.

დიდი n და m-ის შემთხვევაში შეუძლებელია გამოისახოს პატერნი მკვეთრი ლოგიკური ფორმულირებით. ამიტომ საჭიროა შემდგომი ეტაპი - პატერნის გამარტივება.

V. **ბულის ცვლადებზე პირობითი გადასვლის ოპერაცია.**

თუ ყოველ ψ_i ვექტორს შევცვლით x_i-ზე , ხოლო $\overline{\psi}_i$–ს $\bar{x}_i$-ზე, მაშინ φ_+ და φ_- ფუნქციონალები მიიღებენ სრულ დიზიუნქციურ ნორმალურ ფორმას (82), (83)

$$\varphi_+ = \bigvee_{I_+(\sigma_1 \sigma_2 \ldots \sigma_m)} x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_m^{\sigma_m} \tag{28}$$

$$\varphi_- = \bigvee_{I_-(\sigma_1 \sigma_2 \ldots \sigma_m)} x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_m^{\sigma_m} \tag{29}$$

სადაც

$$\sigma_i = \begin{cases} 1 & if \quad x_i \\ 0 & if \quad \bar{x}_i \end{cases} \quad i = 1, 2, \ldots, m, \tag{30}$$

$I_+(\sigma_1 \sigma_2 \ldots \sigma_m)$ – არის C^+ ქვეკლასის ტრაექტორიების ნაკრებების სიმრავლე;
$I_-(\sigma_1 \sigma_2 \ldots \sigma_m)$ – არის C^- ქვეკლასის ტრაექტორიების ნაკრებების სიმრავლე;

ნორმალური დიზიუნქციური ფორმების მინიმიზაციით მივიღებთ პატერნის ბინარულ ფორმას:

$$K_+ = f^+\left(\xi_1^{\sigma_1},\ \xi_2^{\sigma_2}\ \ldots \xi_l^{\sigma_l}\right) = \vee \xi_1^{\sigma_1}\ \ \xi_2^{\sigma_2}\ \ldots \xi_l^{\sigma_l}, \qquad (31)$$

სადაც $l < m$ და $\xi_1^{\sigma_1}, \xi_2^{\sigma_2}, \ldots, \xi_l^{\sigma_l}$ -ის საშუალებით აღნიშნულია ის $x_i^{\sigma_i}, x_j^{\sigma_j}, \ldots, x_k^{\sigma_k}$ ცვლადები, რომლებიც დარჩნენ φ_+ სრული დიზიუნქციური ნორმალური ფორმის მინიმიზაციის შემდეგ (ანალოგიური ფორმა მიიღება φ_- -თვის). დანარჩენი $\xi_{l+1}, \ldots, \xi_m$ ცვლადები არამნიშვნელოვანნი არიან, იმ აზრით, რომ მათი მნიშნელობა არ ახდენს გავლენას ობიექტის შეფასების შედეგზე.

პატერნის ბინარული ფორმა K_+ შეიცავდა მნიშვნელოვან ნიშან-თვისებებს და ასახავს იმ განსაკუთრებულობას, რომელიც დამახასიათებელია C^+ ქვეკლასის სასრული ნაკრების ობიექტებისათვის. საკმაოდ დიდი n და m -ისთვის პატერნი K_+ შეკუმშული ფორმით შეიცავს იმ წესებს (ზოგად შემთხვევაში ცოდნას), რომლითაც ხელმძღვანელობდა შემფასებელი ობიექტების სიმრავლის C^+ და C^- ქვეკლასებად დაყოფისას. შესაბამისად ბინარული K_+ პატერნის საშუალებით შეიძლება შეფასდეს ახალი ობიექტები, რომლებიც არ მონაწილეობდნენ პატერნის ფორმირებაში: ახალი ობიექტის C^+-ზე მიკუთვნებისათვის საკმარისია, რომ ახალი ტრაექტორიის $\xi_1, \ldots, \xi_l$ ცვლადებმა მიიღონ ზუსტად ის მნიშვნელობები, რომლებიც დაფიქსირებულია K_+ პატერნის თუნდაც ერთ იმპლიკანტში, $\xi_{l+1}, \ldots, \xi_m$ ცვლადების მნიშვნელობები ნებისმიერია. შევნიშნოთ, რომ ბინარული პატერნი ადვილად წარმოდგება პროდუქციული წესის სახით.

5.2. ანალიტიკური ევრისტიკების მეთოდი დოკუმენტების კლასიფიკაციისათვის

იმისათვის, რომ შევძლოთ ანალიტიკური ევრისტიკების მეთოდის გამოყენება, უპირველესად უნდა მოვახდინოთ ამ მეთოდში გამოყენებული ძირითადი ელემენტებისა და ცნებების განმარტება.

ნიშან-თვისებათა სიმრავლე - ამ სიმრავლეს შეუსაბამოთ ამა თუ იმ ენის სიტყვათა სიმრავლე $\mathcal{A}$. ყოველი ასეთი სიმრავლე შესაძლებელია თავის მხრივ წარმოვადგინოთ რვა-ათი A_i მტყველების ნაწილების შესაბამისი სიტყვების ქვესიმრავლის გაერთიანებად (1. არსებითი სახელი, 2. ზმნა, 3. ზედსართავი სახელი, 4. რიცხვითი სახელი, 4. ნაცვალსახელი, 6. ზმნიზედა, 7. კავშირი, 8. წინდებული (ან თანდებული), 9. ნაწილაკი, 10. შორისდებული). ქვესიმრავლეთა რაოდენობა დამოკიდებულია კონკრეტული ენის სპეციფიკაზე (მაგ. ქართული ენისთვის - ათი). ყოველი $A_i, (i = \overline{1,10})$, არის $a_{i,j}, j = \overline{1, N_i}$ ელემენტებისაგან შემდგარი სიმრავლე, სადაც N_i არის სიტყვათა რაოდენობა კონკრეტულ A_i-მეტყველების ნაწილში.

კონკრეტული C „ცნების“ ერთი რომელიმე „აღმწერი x ტექსტი“ წარმოვადგინოთ $T_C^x = \{w_{i,j}\}, i = \overline{1,10}\ , j = \overline{1,M}$, სიმრავლის სახით , სადაც $w_{i,j}$ არის ამ ტექსტში ყველა განსხვავებული სიტყვა, ხოლო M არის განსხვავებული სიტყვების რაოდენობა. რა თქმა უნდა, არსებობს კონკრეტული C „ცნების“ აღმწერი სხვა ტექსტიც C^y. ყველა ასეთი სიმრავლე წარმოადგენს საწყისი $\mathcal{A}$ სიმრავლის ქვესიმრავლეს. ჩვენ შეგვიძლია ეს

სიმრავლე გავაფართოვოთ ალ-სიმრავლემდე. ასეთ შემთხვევაში შევძლებთ სრულად გამოვიყენოთ ანალიტიკური ევრისტიკების მეთოდი.

როგორც აღვნიშნეთ, Explicit Semantic Analysis (ESA) მეთოდი მონაცემების წყაროდ იყენებს ვიკიპედიის საცავს (84). ძირითადად გამოიყენება ამა თუ იმ ცნების (აღვნიშნოთ ეს ცნება, როგორც C^{wik}) განმმარტავი ტექსტი (როგორც წესი ერთ ცნებას შეესაბამება ერთი ტექსტი) ავღნიშნოთ ეს ტექსტი, როგორც T_c^{wik}. ყოველი ასეთი ტექსტი წარმოდგება წონითი ნაკრებების სახით, სახელდობრ, ეგრეთწოდებული tf-idf სქემის საშუალებით. სემანტიკური გადამყვანი ახდენს ტექსტის სიტყვების იტერაციას, იღებს ინვერტირებული ინდექსებისაგან შემდგარ შესაბამის ჩანაწერს და აერთიანებს მას ცნების ვექტორში, რომელიც აღწერს ტექსტს.

ნაშრომი (84) შესაბამისად, წარმოვიდგინოთ C კონცეპტის აღმწერი ტექსტი w_i სიტყვების სიმრავლის სახით $T_c^{wik} = \{w_i\}$, i=1,…,M^{wik} (M^{wik} არის ვიკიპედიის საცავში ამ ცნების შესაბამის ტექსტში შემავალი სიტყვების რაოდენობა), რომელსაც შეესაბამება TF-IDF$\langle v_i^{wik}\rangle$ ვექტორი, სადაც ყოველ w_i სიტყვას შეესაბამება v_i^{wik} წონა.

ტერმინის წონების დასადგენად წარმატებით გამოიყენება წონების ავტომატური გენერაციის სქემა:

$$tf - idf_t = tf_{t,d} \times idf_t \qquad (32)$$

ასევე იქმნება ინვერტირებული ინდექსების $\langle k_i^{wik}\rangle$ ვექტორი, რომელშიც k_i^{wik} წარმოადგენს w_i სიტყვის ინვერტირებულ ინდექსს C კონცეპტის შესაბამისი ტექსტისათვის ვიკიპედიის საცავიდან. საცავში არსებული C კონცეპტის შესაბამისი T_C^{wik} ტექსტისათვის გვექნება V_C^{wik} წონების ვექტორი რომელიც განისაზღვრება როგორც:

$$\sum_{w_i \in T_C^{wik}} v_i^{wik} \cdot k_i^{wik} \qquad (33)$$

აღვწეროთ მიღებული C კონცეპტი ანალიტიკური ევრისტიკების მეთოდისათვის მისაღებ ფორმაში. ჩავთვალოთ, რომ ყოველი C-ზოგადად აღიწერება $w_1^{wik}, w_2^{wik}, \dots, w_N^{wik}$ "სიტყვებით". აღწერაში მონაწილე სიტყვების რაოდენობა მკვლევარის (კონცეპტის შემმუშავებლის) შეხადულებაზეა დამოკიდებული. ამ რაოდენობის განსასაზღვრად შესაძლებელია გამოყენებულ იქნას სხვადასხვა მიდგომა, რომელიც უმთავრესად დამოკიდებული იქნება ტექსტის მოცულობაზე, შინაარსობრივად განსხვავებული სიტყვების რაოდენობაზე და სხვა. ყოველი w_i^{wik} სიტყვის მოხვედრა აღწერაში განისზღვრება C კონცეპტის შესაბამისი T_j^{wik} ტექსტის V_j^{wik} წონების ვექტორით. სიტყვის მოხვედრას C კონცეპტის აღწერაში წონის გარდა განსაზღვრავს ის თუ მეტყველების რომელ ნაწილს განეკუთვნება ეს სიტყვა (ანუ $\mathcal{A}$ სიმრავლის რომელ A_i ქვესიმრავლის ელემენტია). როგორც წესი მეტყველების ნაწილის კავშირის შესაბამისი ქვესიმრავლის ელემენტები ასეთ აღწერაში თითქმის არ იღებენ მონაწილეობას. მკვლევართა უმეტესი ნაწილი უპირატესობას ანიჭებს არსებით სახელებს, ან სიტყვათა წყვილს „ზედსართავი სახელი"+„არსებითი სახელი", „არსებითი სახელი"+"ზმნა". სიტყვათა წყვილის გამოსაყოფად მხოლოდ ცალკეული სიტყვების წონების ვექტორი საკმარისი არაა. ეს შემთხვევა სხვა ჩვენი მომდევნო კვლევების საგანი გახდება.

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 კონცეპტის შესაბამისი T_C^x ტექსტი. შესაბამისად ამ საცავისთვისაც შეგვიძლია გამოვიყენოთ ESA მეთოდი და შემდგომ აღვწერთ $w_1^x, w_2^x, \dots, w_L^x$ "სიტყვებით" (L არის ამ საცავში შემავალი სიტყვების რაოდენობა). თუ ამ პროცედურას ჩავატარებთ ჩვენს ხელთ არსებული ყველა საცავისათვის, მივიღებთ ერთი C კონცეპტის შესაბამის რამდენიმე, შესაძლოა განსხვავებულ, აღწერას. C კონცეპტის ყოველი აღწერის შეესაბამისი T_C^x ტექსტისათვის გვექნება V_C^x წონების ვექტორი.

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

ცხრილი 1. C კონცეპტის აღმწერი ტექსტის წონების ვექტორი

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

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

შესაბამისად *ცხრილი 2* მიიღებს უნიფიცირებულ სახეს:

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

R^k	$\breve{w}_{1,k}$	$\breve{w}_{2,k}$	…	$\breve{w}_{i,k}$	…	$\breve{w}_{N,k}$
…	…	…	…	…	…	…
R^m	$\breve{w}_{1,m}$	$\breve{w}_{2,m}$	…	$\breve{w}_{i,m}$	…	$\breve{w}_{N,m}$

ცხრილი 3. C კონცეპტის აღმწერი ტექსტის წონების ვექტორი (უნიფიცირებული სახით)

სადაც

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

რადგან ყოველი C კონცეპტის აღწერა სასრულია, სასრულია $W = \{w_1, w_2, \dots, w_N\}$ სიმრავლეც, ჩვენ შეზღუდვის გარეშე შეგვიძლია ეს სივრცე ალ-სიმრავლედ წარმოვიდგინოთ, და მასში განვმარტოთ ყველა ის ოპერაციები, რაც განმარტებულია ასეთი ტიპის სიმრავლეებში. თუ დავუბრუნდებით ანალიტიკური ევრისტიკების მეთოდის აღნიშვნებს და განმარტებებს სრულად, ჩვენ შეგვიძლია ვთქვათ, რომ C კონცეპტი იგივეა, რაც ობიექტი და ყოველი w_i ამ ობიექტის აღმწერი ნიშანთვისებების A_i-ის შესაბამისია. ეს საშუალებას გვაძლევს სრულად გამოვიყენოთ ეს მეთოდი. აქედან გამომდინარე, C-ს ყოველი რეალიზაცია წარმოდგება ჩვეულებრივი იმპლიკანტის სახით, ხოლო კონცეპტის პატერნის მისაღებად მოვახდენთ ნორმალური დიზიუნქციური ფორმის მინიმიზაციას.

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	$\overline{w}_1 \& w_2 \& w_3 \& w_4$

ცხრილი 4 . კონცეპტის წარმოდგენა იმპლიკანტების სახით.

ჩავწეროთ ეს რეალიზაციები დიზიუნქციური ნორმალური ფორმის სახით,

$$(w_1\&w_2\&\overline{w}_3\&w_4)\vee(w_1\&\overline{w}_2\&\overline{w}_3\&w_4)\vee(w_1\&w_2\&\overline{w}_3\&w_4)\vee(w_1\&\overline{w}_2\&\overline{w}_3\&w_4)$$
$$\vee(w_1\&w_2\&\overline{w}_3\&w_4) \quad (34)$$

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

$$w_1\&\overline{w}_3\&w_4 \quad (35)$$

რა თქმა უნდა, რეალურად, კონცეპტის აღწერაში ბევრად უფრო დიდი რაოდენობის სიტყვები იღებენ მონაწილეობას, მაგრამ ჩვენ შეგვიძლია შევარჩიოთ ყველაზე მაღალი წონის შესაბამისი სიტყვები ზუსტი სემანტიკური ანალიზის (Explicit Semantic Analysis (ESA)) მეთოდით მიღებულ ვექტორზე დაყრდნობით. რაოდენობის განსაზღვრა შეიძლება დამოკიდებული იყოს ტექსტში სიტყვების რაოდენობის და განსხვავებული სიტყვების რაოდენობის თანაფარდობაზე. ასეთი მიდგომა გააძლიერებს საბოლოოდ მიღებული კონცეპტის მიერ სემანტიკურ მნიშვნელობის აღწერას, რადგან, რაც უფრო მეტი სიტყვა მიიღებს მონაწილეობას აღწერაში, მით უფრო სემანტიკურად ადეკვატური იქნება შედეგი. თუმცა, მეორე მხრივ სიტყვების დიდმა რაოდენობამ შესაძლოა გაართულოს კონცეპტის გამოყენება ინფორმაციის ძებნისათვის.

მეთოდის შესამოწმებლად ჩატარდა სატესტო შემოწმება. ტესტირება მოიცავდა ორ საფეხურს: 1. ცნებების ფორმირება; 2. ძებნა ფორმირებული ცნების შესაბამისად. რადგან კონცეპტი წარმოადგენს იმპლიკანტს (დიზიუნქცია, კონიუქცია), მას მიესადაგება ბულის ძებნის ალგორითმი. სწორედ ამ ალგორითმით მოხდა მეთოდის შემოწმება.

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 ცალკ-ცალკე . ძებნის სიზუსტე(Precision) მერყეობდა 0.81 დან 0.92 მდე.

მეხუთე თავის დასკვნა

ზემოთ აღწრილ თავში განვიხილეთ კონცეპტების ფორმირების ანალიტიკური ევრისტიკების მეთოდის, როგორც ობიექტების კლასიფიკაციისათვის საჭირო ინსტრუმენტის, ზოგადი აღწერა,

სემანტიკური მიდგომის თავისებურებას უმთავრესად წარმოადგენს ის ფაქტი, რომ გამოიყენება დოკუმენტების კონცეპტუალური წარმოდგენა, რომელიც იქმნება საგნობრივი არის ცოდნის სემანტიკურ მოდელებზე დაყრდნობით, ხოლო ცოდნის წარმოდგენის არსებულ ინსტრუმენტებს შორის ონტოლოგია წარმოადგენს ყველაზე გამოსახვით ხერხს. ჩვეულებრივ ონტოლოგიებში საგნობრივი არეების ცოდნა აღიწერება ცნებებისა და თვისებების იერარქიით, ასევე შეერთებული ცნებების ეგზემპლარების სემანტიკური ქსელებით. არსებულ მიდგომებთან შედარებით, ონტოლოგიის გამოყენებამ შესაძლოა მოგვცეს საშუალება გავაუმჯობესოთ ძებნის ხარისხი. ამიტომ მნიშვნელოვანია საგნობრივი არის აღმწერი ცოდნის ბაზის შემუშავების მოქნილი ალგორითმის შექმნა. ჩვენს მიდგომაში ეს ალგორითმი ეფუძნება ანალიტიკური ევრისტიკების მეთოდს.

მეთოდი მოიცავს სხვადასხვა ეტაპებს: ნიშანთვისებათა ბინარიზაცია; ნიშანთვისებათა გადაკოდირება; ორთონორმირებული ბინარული მდგომარეობის ვექტორების აგება; ფილტრაციის ოპერაცია; დიზიუნქციური სუპერპოზიციის ოპერაცია; ბულის ცვლადებზე პირობითი გადასვლის ოპერაცია.

ასევე აღვწერეთ ანალიტიკური ევრისტიკების მეთოდი დოკუმენტების კლასიფიკაციისათვის. მისი საშუალებითაც შესაძლებელი იქნება ამა თუ იმ ცნების (კონცეპტის) განზოგადოებული ჩარჩოს (პატერნის) შემუშავება, რომელიც შეიძლება წარმოდგენილ იქნას, როგორც დოკუმენტების კლასიფიკატორის ძირეული საბაზო ელემენტი საძიებო სისტემაში.

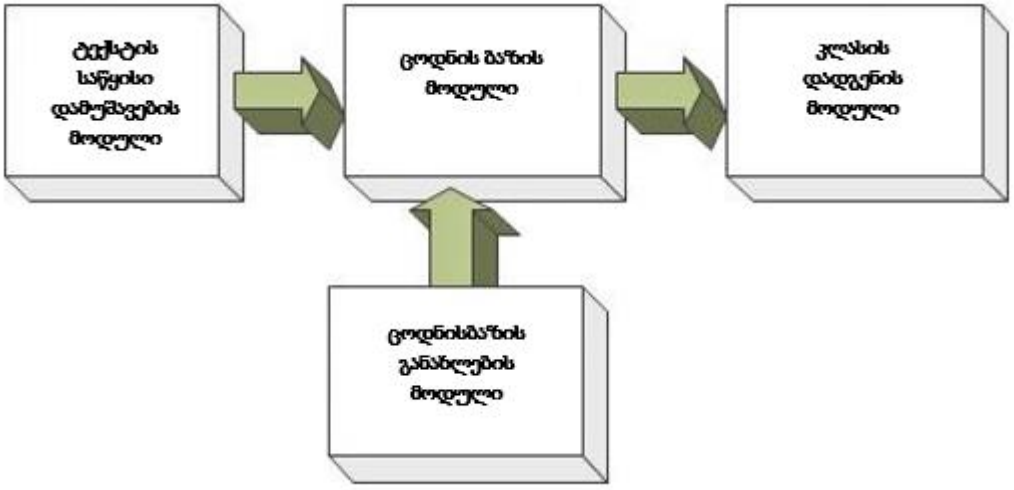
შეფასებულია მეთოდის სატესტო შემოწმება, რომლის საფუძველზე შესაძლებელია დავასკვნათ, რომ შემოთავაზებული მეთოდით კონცეპების შემუშავება განზოგადოებულად აღწერს მის სემანტიკურ არსს. ეს მეთოდი საშუალებას იძლევა კონცეპტის არასტრუქტურირებული აღმწერი მეტამონაცემების (ტექსტების) საფუძველზე მოხდეს განზოგადოებული სემანტიკური არსის მქონე სტრუქტურის ფორმირება. ეს სტრუქტურა წარმოადგება, როგორც ინფორმაციის ძებნის ერთ-ერთი ძირითადი კომპონენტი.

6. ტექსტების კლასიფიკაცია ცნების პატერნზე დაფუძნებული სისტემის გამოყენებით

6.1. ტექსტების კლასიფიკაციის სისტემის არქიტექტურა

ჩვენს მიერ შემუშავებული ტექსტების კლასიფიკაციის მეთოდის პრაქტიკული გამოყენება შესაძლებელია მრავალმოდულიანი სისტემის შემუშავების საშუალებით. ეს მოდულებია:

1. ტექსტების საწყისი დამუშავების მოდული;
2. ცოდნის ბაზის მოდული;
3. კლასის დადგენის მოდული;
4. ცოდნის ბაზის შემდგომი გამდიდრების მოდული.



ნახ 1. კლასიფიკაციის სისტემის არქიტექტურა

6.1.1. ტექსტების საწყისი დამუშავების მოდული

ეს მოდული მოიცავს კლასიფიცირებისათვის განკუთვნილი დოკუმენტების პირველად დამუშავებას. აღნიშნული პროცესი გულისხმობს ტექსტის სტემინგის და ლემატიზაციის პროცესს, რომლის შედეგადაც მიიღება ტექსტის „ცნების“ სახით აღწერისათვის ტერმინების ვექტორი. ტექსტის „ცნების ვექტორის“ სიგრძე დამოკიდებულია საწყისი ტექსტის მოცულობაზე. დაახლოებით 600 სიტყვიანი ტექსტისათვის იგი შედგება 10 კომპონენტისაგან (ტერმინისაგან); 600-დან 3000-სიტყვამდე - 20 ტერმინისაგან; 3000-დან 10 000-მდე სიტყვისათვის - 30 ტერმინი. ტერმინების რაოდენობა შერჩეულია ექსპერიმენტული შემოწმებების შედეგედ მიღებული მონაცემების ანალიზის საფუძველზე. ტექსტის აღმწერი ვექტორის ასაგებად გამოიყენება tf-idf წონითი სქემა .

განვიხილოთ ტექსტის დამუშავების ეტაპი უფრო დეტალურად

სტემინგის ალგორითმი ქართული ენისათვის

დოკუმენტის კლასიფიცირების ამოცანა ემყარება ტექსტის ინდექსაციას სტემინგის გამოყენების გზით, რომელიც მოსახერხებელია არა მარტო ინფორმაციის ძებნის, არამედ მანქანური სწავლების მოდელების შემუშავებისათვისაც. როგორც ზემოთ აღვნიშნეთ, არ არსებობს სტემინგის და ლემატიზაციის უნივერსალური ალგორითმი, რომელთა გამოყენებაც შესაძლებელია ყველა ტიპის ენებისათვის. აქედან გამომდინარე, გავითვალისწინეთ რა ქართული ენის თავისებურებები, აუცილებელი გახდა ქართული ენისათვის ახალი სტემინგის ალგორითმის შემუშავება, რომლის გამოყენებაც შესაძლებელი იქნება ტექსტის დამუშავების საწყის მოდულში.

სხვა ენებთან შედარებით, ქართულ ენაზე არსებული ტექსტების კლასიფიცირების შესახებ კვლევები ძალიან მცირეა.

ქართული მიეკუთვნება იმ აგლუტინაციურ ენათა ჯგუფს, რომელსაც სიტყვაში შეიძლება გააჩნდეს რვა მორფემაც კი, მაგრამ ასეთ ენებშიც კი იგი გამონაკლისია, რადგან ქართულში ზმნის უღლება მეტად თავისებურია და ანალოგი არ გააჩნია არცერთ სხვა ენასთან. ქართული ენის სირთულეზე წარმოდგენა მარტივად შეგვექმნება თუ ვიტყვით რომ ლინგვისტების მიერ ქართულისთვის გამოიყენება ტერმინი ,,screeve", რაც იმას გულისხმობს, რომ ერთი ზმნის პირისა და რიცხვის წარმოებისათვის ექვსი ფორმა გამოიყენება. მორფემებია: ზმნისწინი, სუბიექტური და ობიექტური პირისა და რიცხვის ნიშნები (სამი გრამატიკული პირიდან სამივეს სხვადასხვა ნიშანი აქვს), სავრცობი, თემის ნიშანი, გვარის ნიშანი, ქცევის ნიშანი და კონტაქტის მაწარმოებელი. მაგ. ზმნა „ჩაგეშენებინათ" ასე დაიყოფა: ჩა-გ-ე-შენ-ებ-ინ-ა-თ (85).

კლასიფიკაციის საწყის ეტაპზე, ხდება ტექსტის ლექსემებად დაყოფა მისი შემდგომი დამუშავებისათვის. დაყოფილ ტექსტს უნდა ჩამოცილდეს ისეთი სიტყვები, რომლებსაც ტექსტის შინაარსის განსაზღვრაში არანაირი დატვირთვა არ გააჩნიათ. ესენია ე.წ „სტოპ" სიტყვები. დარჩენილი სიტყვებისათვის კი უკვე ხორციელდება სტემინგის პროცესი, რომელიც მათ ნორმალიზაციას გულისხმობს. ნორმალიზაციის პროცესის აუცილებლობა გამოწვეულია იმ ფაქტით, რომ ტექსტში შესაძლებელია ერთი და იგივე სიტყვა შეგვხვდეს სხვადასხვა ფორმით, სწორედ ამიტომ საჭიროა მათი დამუშავება და ერთიან კანონიკურ ფორმაზე დაყვანა, რომ ერთი სიტყვის სხვადასხვა ფორმა განსხვავებულ სიტყვებად არ აღიქმებოდეს. სიტყვის დამუშავებაში იგულისხმება მისგან იმ ნაწილის დატოვება, რომელიც ფორმაწარმოების დროს უცვლელი რჩება. სიტყვის უცვლელ ნაწილს ფუძეს უწოდებენ. ფუძის გარკვევის სირთულე პირდაპირ კავშირშია ენაში სიტყვის ფორმათაწარმოების წესების სირთულეებთან. ინგლისური ენისათვის ეს ამოცანა უფრო მარტივად იჭრება, რადგან აქ ჩვეულებრივ პროცედურას წარმოადგენს სიტყვისათვის დაბოლოებების მოცილება. ქართულ ენაში წარმოიქმნება სიტყვის კუმშვისა და კვეცის პრობლემა, რომელსაც თან ერთვის მრავალმნიშვნელოვანობის ან საწყისი ფორმის არაერთგვაროვნობის პრობლემა.

ქართულ ენაში მეტყველების სხვადასხვა ნაწილები მათთვის დამახასიათებელი თავისებურებებით გამოირჩევა. მაგ. ზმნა არის გაცილებით მდიდარი ვიდრე დანარჩენი ფორმაცვალებადი მეტყველების ნაწილები და თანაც ხშირად ავლენს თავისებურებას უღლებაში. სხვა სახელები (მაგალითად რიცხვითი) იშვიათია და ვერ მოახდენს

მნიშვნელოვან გავლენას კლასიფიკაციაზე. ამიტომ ამ ეტაპზე შევისწავლეთ არსებითი სახელები.

ტექსტის ნორმალიზაციისათვის გამოყენებულ იქნა სტემინგის ალგორითმი, რომელიც წარმოადგენს „პორტერის“ ალგორითმის მოდიფიცირებულ ვარიანტს ქართული ენის თავისებურებების გათვალისწინებით.

კატეგორიზაციის ამოცანაში ტექსტის ინდექსაცია ეფუძნება სახელიდან ფუძის ამოღების პროცესს. სახელის ფორმაწარმოების დროს ცვლილებას განიცდის სუფიქსური ელემენტი (ბრუნვის ნიშანი, რიცხვის ნიშანი), ხოლო სახელის ფუძე (ლექსიკური ბაზისი) უცვლელია (ან მხოლოდ ფონეტიკურად იცვლება). ქართული არსებითი სახელის ფორმაწარმოება მარტივია, ერთტიპური; მისთვის ერთფუძიანობაა დამახასიათებელი.

ფორმაწარმოების (ბრუნებისა თუ რიცხვის წარმოების) პროცესში არსებითი სახელის ფუძე და ბრუნვის ნიშნები ურთიერთმოქმედებენ. ამ ურთიერთმოქმედებამ შეიძლება ერთ-ერთ მათგანში ფონეტიკური ცვლილება გამოიწვიოს.

ქართულ ენაში ფუძის დაბოლოების მიხედვით სახელები თანხმოვანფუძიანია ან ხმოვანფუძიანი. თანხმოვანფუძიანი სახელებდან ზოგი უკუმშველია, ზოგი - კუმშვადი. კუმშვა ეწოდება ფუძიდან ხმოვნის ამოვარდნას (კალამი-კალმის; კედელი-კედლის). იკუმშება ა, ე და ო ხმოვნები. კუმშვას ფუძეზე ხმოვნით დაწყებული ბოლოსართის დართვა იწვევს. **ო** ხმოვანი ან საერთოდ იკარგება (ობოლი-ობლის), ან ვ-დ იქცევა (ნიორი-ნივრის). კუმშვად სახელებს ფუძე ეკუმშებათ მხოლობითი რიცხვის სამ ბრუნვაში: ნათესაობითში, მოქმედებითსა ვითარებითში და ებ-იან მრავლობითის ყველა ბრუნვაში.

ფუძის ბოლო ხმოვნის დაკარგვას კვეცა ეწოდება. კვეცადია ხმოვანფუძიანი საზოგადო სახელები, რომელთა ფუძე ბოლოვდება **ა** და **ე** ხმოვნებზე. ხმოვანფუძიან სახელებს ფუძე ეკვეცებათ მხოლობითი რიცხვის მხოლოდ ორ ბრუნვაში – ნათესაობითსა და მოქმედებითში. ა-ზე დაბოლოებული სახელი ებ-იან მრავლობითში იკვეცება ყველა ბრუნვაში, ე-ზე დაბოლოებული სიტყვები კი არ იკვეცება.

რამდენიმე სახელს ფუძე ერთდროულად ეკუმშება და ეკვეცება. კუმშვად-კვეცადი სახელები (ქვეყანა, ქარხანა, ბეგარა, მოყვარე, ფანჯარა, ტომარა, პეპელა) მხოლობითი რიცხვის ნათესაობით, მოქმედებითსა და ვითარებით ბრუნვებში და ებ-იანი მრავლობითის ყველა ბრუნვაში შეკუმშულ-შეკვეცილი ფორმით უნდა ვიხმაროთ (ქვეყნის, მოყვრის, პეპლები…).

სიტყვის კვეცის პრობლემა ადვილად წყდება სიტყვების ბაზით, თუმცა კუმშვადობას სიტყვების ბაზა ვეღარ მოაგვარებს, საჭიროა სიტყვების მორფოლოგიური დახასიათება.

ქართული ენის სტემინგის ალგორითმში გამოყენებულია სიტყვების ბაზა, სადაც სიტყვებზე არანაირი მორფოლოგიური ინფორმაცია არ გაგვაჩნია. სიტყვები სახელობით ბრუნვაშია.

ამ ალგორითმის უარყოფით მხარედ შეიძლება ჩაითვალოს ის, რომ რამდენიმე ათეული ათასი სიტყვიდან თითოეულისთვის ვაწარმოეთ მისი ყველა ფორმა, რის შედეგადაც სიტყვათა ბაზის მოცულობა იზრდება.

ამგვარად არსებით სახელის სტრუქტურა ქართულ ენაში ასეთია: ფუძე, ბრუნვის ნიშანი, თანდებული, სავრცობი. ტექსტის მორფოლოგიური დამუშავებისას ხდება

ფუძიდან სავრცობის, თანდებულის და ბრუნვის ნიშნის ჩამოცილება. თუ სიტყვის ფორმა მიუთითებს ერთზე მეტ ფუძეს ავარჩევთ ისეთს რომელშიც ყველაზე ნაკლები მორფემაა (ბრუნვის ნიშანი და ნაწილაკი).

ფუძის ამოღების ამოცანა ფორმალურად ასე შეიძლება ჩამოვაყალიბოთ: შევიმუშავოთ მეთოდი, რომელიც ორი ნებისმიერი სიტყვისთვის გაარკვევს ეკუთვნიან თუ არა ისინი ერთი და იმავე ლექსიკური ერთეულის ფორმებს.

თუ (A და B) და (B და C) სიტყვები არიან ერთი ლემის ფორმები, მაშინ A და C შესაბამისად იქნებიან ერთი ლემის ფორმები. აქედან გამომდინარე, ყველა არსებითი სახელის ყველა ფორმის სიმრავლე არის ტრანზიტული და ქმნის ეკვივალენტურ კლასებს, სადაც თითოეული ექვივალენტური კლასი წარმოადგენს ერთი ლემის ყველა ფორმის სიმრავლეს.

ჩვენი მიზანია თითოეულ ეკვივალენტობის კლასს შევუსაბამოთ რაიმე უნიკალური მაიდენტიფიცირებელი, რომელიც მიიღება ამ კლასის ყველა წევრისგან. მაიდენტიფიცირებელ მნიშვნელობად ჩვენ ავიღეთ სიტყვის გრამატიკული ფუძე, რაც გავრცელებულ პრაქტიკას შეესაბამება.

ალგორითმი იყენებს ქართული არსებითი სახელების ბაზას (დაახლოებით 30000 სიტყვამდე) და სუფიქსების ბაზას (დაახლოებით 60-მდე). ბაზაში სიტყვები სახელობით ბრუნაშია. ალგორითმის მუშაობის პრინციპი მარტივია, იღებს სიტყვას და თუ ბაზაში მოიძებნება მოცემული სიტყვის შესაბამისი სახელობითი ბრუნვის ფორმა, აბრუნებს ამ სიტყვის გრამატიკულ ფუძეს.

ალგორითმი ახორციელებს შემდეგ ეტაპებს :

— თითოეულ შემომავალ სიტყვას ჩამოაშორებს სუფიქსს;
— ეძებს ბაზაში თუ არსებობს მოცემული სიტყვის შესაბამისი სახელობითი ბრუნვის ფორმა;
— თუ შესრულდა წინა პუნქტი აბრუნებს გრამატიკულ ფუძეს.

მეთოდი რეალიზებულია Java-ზე. სიტყვათა და სუფიქსების ბაზები წარმოდგენილია ცალკეული ფაილების სახით, რაც აადვილებს მის ეტაპობრივ შევსებას.

სიტყვიდან ფუძის მიღების პროცესი ხორციელდება კონკრეტული ოპერაციების შესრულების შედეგად.

1. საწყის ეტაპზე ალგორითმი ამოწმებს აქვს თუ არა რაიმე სუფიქსი სიტყვას -ძებნა ხორციელდება სუფიქსების ბაზაში. აღმოჩენის შემთხვევაში მოცემულ სიტყვას სუფიქსი სცილდება. ბაზაში სუფიქსები დალაგებულია სიგრძის კლებადობის მიხედვით, რაც იძლევა გარანტიას, რომ თუ მოცემული სიტყვის სუფიქსი სხვა სუფქსსაც შეიცავს, მთლიანად ჩამოაშორებს მას (მაგალითად: „მხეც-ებივით“ და „მხეც-ივით", „ებ-ი-ვით“ არის მთლიანი სუფიქსი და ამიტომ ერთად შორდება და არა ცალ-ცალკე „ებ-ი“-ს და „ვით“-ს). სუფიქსებად მიჩნეულია ყველაფერი ერთად (ბრუნვის ნიშნები, სავრცობები და ა.შ. მაგ. „ებ-ის-ა-თვის“ არის მთლიანად სუფიქსი);
2. მომდევნო ეტაპზე ხორციელდება მსგავსი სიტყვების შედარება - გადაეცემა ორი სიტყვა და ამოწმებს საწყისი პოზიციიდან არის თუ არა მეორე სიტყვა პირველის „ქვესტრიქონი“ (ან პირიქით) (მაგალითად: „კედელი“ და „კედლ“ და არა - „დედა“ და

„და“). ამის შემოწმება მარტივად ხორციელდება: სიმბოლოების შედარება იწყება პირველი პოზიციიდან, თუ იმ სიტყვაში, რომელშიც ხორციელდება „ქვესტრიქონის“ ძიება, არდამთხვევა ერთი სიმბოლოა, შედარების პროცედურა გრძელდება (მაგ: „კედელი“ და „კედლ“, „ლ“ != „ე“, ამიტომ შედარდება „კედელი“-ში „ე“-ს მომდევნო სიმბოლო და „ლ“);

3. შემდეგ ეტაპზე ძებნა მიმდინარეობს არსებითი სახელების ბაზაში - მოცემული ფუძის შესამაბისი სახელობითი ბრუნვის ფორმის აღმოსაჩენად. ამისათვის ფაილში თითოეული სიტყვისთვის მოწმდება გადაცემული სიტყვის ქვესტრიქონად ყოფნის დადგენის მიზნით (`contains` მეთოდის გამოყენებით). თუ `contains` მეთოდი წარმატებით შესრულდა, მოწმდება ბაზაში არსებული სიტყვის დაბოლოება - არის თუ არა „ი“, თუ კი - აბრუნებს სიტყვას „ი“-ს გარეშე, წინააღმდეგ შემთხვევაში - თვითონ სიტყვას;
4. წინა ეტაპის უშედეგოდ დასრულების შედეგად ბრუნდება null.

მოცემული მეთოდი მაქსიმალურად უზრუნველყოფს გადაცემული სიტყვისთვის გრამატიკული ფუძის დაბრუნებას. მუშაობს კუმშვად, კვეცად და კუმშვად-კვეცად არსებით სახელებზე. თუმცა მისი მუშაობა დამოკიდებულია იმაზე, თუ რამდენად სრულყოფილია სიტყვათა და სუფიქსების ბაზები.

მეთოდმა ტესტირება გაიარა ტექსტების კატეგორიზაციის სამ ალგორითმზე: ბაიესის, k-უახლოესი მეზობლის და მხარდამჭერი ვექტორების ალგორითმებზე. სამივე ალგორითმისთვის შედეგები იდენტურია.

ტექსტიდან ტერმინების სიმრავლის მიღების შემდეგ, უკვე შესაძლებელია ტერმინებისათვის წონების დათვლა. პროცედურა ხორციელდება ზემოთ აღწერილი წესების შესაბამისად: დამუშავების შედეგად მიღებული ტერმინების ნაკრებიდან ყველა განსხვავებული „ტერმინისათვის“ გამოითვლება tf წონა. წონის გამოთვლის წესი შეირჩევა საწყისი ტექსტის მოცულობის მიხედვით. თუ ტექსტის მოცულობა 600 სიტყვემდეა, ყოველი t_i ტერმინის წონის გამოსათვლელად გამოიყენება ფორმულა:

$$t_i f(t_i, d_j) = \frac{n_i}{\sum_{k_j} n_{k_j}} \tag{36}$$

სადაც n_i არის d_j დოკუმენტში t_i ტერმინის რაოდენობა, ხოლო - n_{k_j} ამ დოკუმენტში ყველა სიტყვის საერთო რაოდენობა.

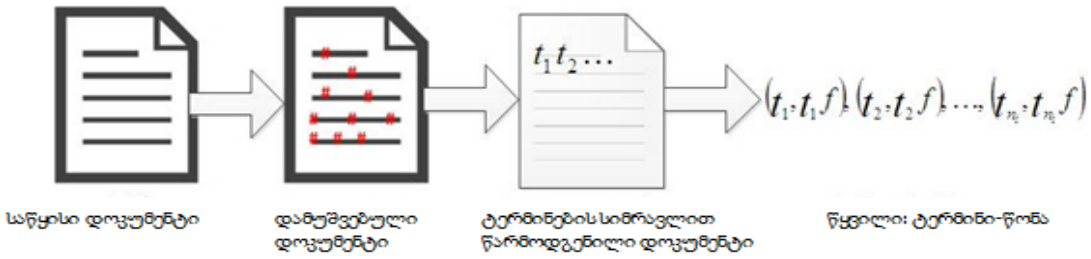
საშუალო ზომის ტექსტისათვის (600-დან 3000 სიტყვამდე) გამოიყენება:

$$t_i f(t_i, d_j) = 1 + \log f(t_i, d_j) \tag{37}$$

ხოლო დიდი ტექსტებისათვის :

$$t_i f(t_i, d_j) = 0.5 + \frac{0.5 \times f(t_i, d_j)}{max\{f(w, d_j) : w \in d_j\}}. \tag{38}$$

წონების დათვლის შემდგომ ყოველი ტექსტისათვის (ზომის მიხედვით) ამოირჩევა პირველი 10, 20 ან 30 მაღალი წონის ტერმინი, რაოდენობა განისაზღვრება ტექსტის სიგრძის შესაბამისად. აღნიშნული ოპერაციის ჩატარების შემდეგ ვღებულობთ ტექსტიდან ყველა განსხვავებულ ტერმინს თავისი შესაბამისი შეხვედრის სიხშირეებით, რომლებიც დოკუმენტში ამ ტერმინის წონის შესაბამისია.



ფიგურა 2 ტექსტის საწყისის დამუშავების პროცესი

6.1.2. ცოდნის ბაზის მოდული - ცნების პატერნების შემუშავება ცოდნის ბაზისთვის

ცოდნის ბაზის მოდული წარმოადგენს სისტემის ძირითად ნაწილს. მის სისრულესა და ლოგიკურობაზეა დამოკიდებული სისტემის ფუნქციონირების ადექვატურობა. ცოდნის ბაზა წარმოდგება კონკრეტული კლასის აღმწერი „კლასის ცნების პატერნების“ ერთობლიობის სახით. ყოველ კლასს აღწერს მინიმალურ ნორმალურ დიზიუნქციურ ფორმაში გამოსახული ჩანაწერი, რომლის იმპლიკანტები „ტერმინებია“. ცოდნის ბაზის შექმნა ცალკე პროცედურაა და ის არ წარმოადგენს უშუალოდ კლასიფიცირების სისტემის ფუნქციონალურ ნაწილს.

აღვწეროთ ცოდნის ბაზის შემუშავების პროცესი:

დავუშვათ გავაქვს $D = \{d_1, d_2, \cdots, d_N\}$ დოკუმენტთა სიმრავლე (კოლექცია) (N დოკუმენტების რაოდენობაა), რომლებიც გადანაწილებულნი არიან $C = \{c_1, c_2, \cdots, c_K\}$ კლასებში (K კლასების რაოდენობაა). ყოველ c_i, $i = \overline{1, \ldots, K}$, კლასს შეესაბამება დოკუმენტების D_i ქვესიმრავლე D-დან, $D_i = \cup_{j=\overline{1,\ldots J}} d_{i_j}$, J არის c_i კლასის შესაბამისი დოკუმენტების რაოდენობა. ყოველი D_i სიმრავლიდან შესაბამისი c_i კლასის აღმწერი კონცეპტის პატერნის შესამუშავებლად გამოვიყენოთ ანალიტიკური ევრისტიკების მეთოდი (86) ანალოგიურად ზემოთ აღწერილი მეთოდისა ყოველი d_{i_j} დოკუმენტისათვის შევადგინოთ ტერმინების ვექტორი (რადგან სხვადასხვა მოცულობის ტექსტებისათვის სხვადასხვა ზომის პატერნები გამოიყენება, ვექტორის სიგრძის შერჩევა მოხდება ამის შესაბამისად).

$$\begin{array}{ccc} d_{i_1} & \rightarrow & t_{i_1}^1, t_{i_2}^2, \ldots, t_{i_1}^M \\ d_{i_2} & \rightarrow & t_{i_2}^1, t_{i_2}^2, \ldots, t_{i_2}^M \\ \cdots & \cdots & \\ d_{i_J} & \rightarrow & t_{i_j}^1, t_{i_j}^2, \ldots, t_{i_j}^M \end{array} \qquad (39)$$

d_{i_j} არის c_i კლასის შესაბამისი დოკუმენტი (ტექსტი);

შენიშვნა: თუ ვიყენებთ განსხვავებული ზომის პატერნებს განსხვავებული ტექსტებისათვის ვექტორის სიგრძე იქნება განსხვავებული.

t_{ij}^{m} $(m = \overline{1, M})$არის d_{i_j} დოკუმენტის ტერმინების ვექტორში შემავალი პირველი M მაღალწონიანი სიტყვა. $M = \{10, 20, 30\}$. ყოველი განსხვავებული t_{ij}^{m} ტერმინი c_i კლასის შესაბამისი დოკუმენტების კოლექციიდან აღვნიშნოთ $w_{ij}^{n}, n = 1 \dots N$. მაშინ ანალიტიკური ევრისტიკების ტერმინებში აღიწერილ, c_i კლასის შესაბამისი d_{i_j}-ს „ტერმინების ვექტორს“ ექნება სახე:

$$d_{i_j} \rightarrow \breve{w}_{i_j}^{1} \& \breve{w}_{i_j}^{2} \& \dots \& \breve{w}_{i_j}^{N} \tag{40}$$

სადაც:

N - განსხვავებული მაღალი წონის მქონე ტერმინების რაოდენობაა;

$$\breve{w}_{i_j}^{n} = \begin{cases} {w_{i_j}}^{n}, \text{თუ ეს სიტყვა წარმოადგენს შესაბამისი ტექსტის მაღალწონიან სიტყვას} \\ \overline{{w_{i_j}}^{n}}, \text{თუ ეს სიტყვა არ წარმოადგენს შესაბამისი ტექსტის მაღალწონიან სიტყვას} \end{cases}$$

წარმოვადგინოთ „ტერმინების ვექტორების“ ერთობლიობა დიზიუნქციური ნორმალური ფორმის სახით:

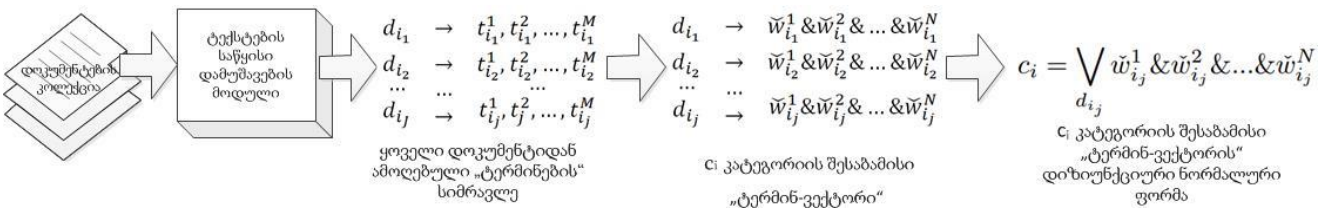
$$\mathrm{c}_i = \bigvee_{d_{i_j}} \breve{w}_{i_j}^{1} \& \breve{w}_{i_j}^{2} \& \dots \& \breve{w}_{i_j}^{N} \tag{41}$$

საბოლოოდ ცოდნის ბაზისათვის გვექნება 1 დონის პატერნი. ცოდნის ბაზაში სულ ასეთი პირველი დონის პატერნი გვექნება K ცალი. პატერნების სიმრავლე წარმოადგენს იმპლიკანტების ერთობლიობას და ისინი არიან ცოდნის ბაზის შემავსებელი ძირითადი ელემენტები.

გამოვსახოთ c_i კლასის შესაბამისი P^i პატერნი I_k^i იმპლიკანტების სახით.

$$P^i = I_1^i \vee I_2^i \dots \vee I_{j_i}^i \tag{42}$$

სადაც j_i წარმოადგენს იმპლკანტების რაოდენობას შესაბამისი c_i კლასიდან.



ნახ. 3. კონცეპტის ფორმირება

იმ შემთხვევაში, თუ კლასიფიკაცია მოითხოვს უფრო მეტ დაზუსტებას, (ქვედარგების, ქვეთემების და ა.შ. მითითებით), ანალოგიური პროცედურა ჩატარდება ყოველი c_i კლასისათვის.

დავუშვათ გვაქვს c_i კლასზე მიკუთვნებულ დოკუმენტტა $D_i = \{d_{i_1}, d_{i_2}, \cdots, d_{i_N}\}$ სიმრავლე (i_N ამ კლასს მიკუთვნებული დოკუმენტების რაოდენობაა). c_i დაყოფილია $c_{i_j}, i_j = \overline{\iota_1, \dots, \iota_k}$ ქვეკლასებად. შესაბამისად D_i ქვესიმრავლეში შემავალი დოკუმენტები გადანაწილებულია $D_{i_1}, \dots, D_{i_k}$ ქვესიმრავლეებში (ქვეკლასებზე მიკუთვნების შესაბამისობით). ყოველი D_{i_j} ქვესიმრავლის საფუძველზე შემუშავდება მეორე დონის პატერნები, რომლებიც ანალოგიურად პირველი დონის პატერნებისა შეავსებენ ცოდნის ბაზას. თუ დოკუმენტების კლასიფიკაცია მოითხოვს უფრო მეტ დაზუსტებას ჩატარდება ანალოგიური პროცედურა c_{i_j} ქვეკლასებისთვისაც და ა.შ.

ეს მოდული მოქმედებს ტექსტის საწყისი დამუშავების შედეგად მიღებული „ტერმინების ვექტორის“ საშუალებით. განისაზღვრება, რა ტექსტის მოცულობა საწყისი მოდულის მიერ დადგინდება ტერმინების ვექტორის სიგრძე და მოხდება მისი ფორმირება ზემოთ აღწერილი პროცედურების შესაბამისად.

შემდგომ ხდება ტერმინების ვექტორის შედარება ცოდნის ბაზაში. პროცესორი მიმართვას ცოდნის ბაზის იმ ნაწილს სადაც შესაბამისი სიგრძის პატერნებია განთავსებული. დასამუშავებული ტექსტის ვექტორი დარდება პატერნს და ფილტრაციის მეთოდის საშუალებით შეირჩევა ყველაზე ახლოს მდგომი პატერნი. იმ შემთხვევაში თუ ტექსტის შესაბამისი ზომის პატერნში შედარებამ ცუდი შედეგი მოგვცა, პროცესორი იყენებს სხვა მოცულობის ტექსტებისათვის განკუთვნილი პატერნებზე შედარებას. იმ შემთხვევაში თუ ვერ მოხდა დასამუშავებელი ტექსტის მიკუთვნება ვერცერთ ძირითად კლასზე, პროცესორს შეუძლია გამოიყენოს ქვეკლასების შესაბამის პატერნები. ამ ეტაპზეც უარყოფითი შედეგის მიღების შემთხვევაში ტექსტი გადაიგზავნება ცოდნის ბაზის გამდიდრების მოდულში მომავალი დამუშავებისათვის.

ავღწეროთ პროცესი მაგალითის საფუძველზე:

მაგალითად: თუ გვაქვს 3 განსხვავებული კლასი c_1, c_2, c_3. ყოველ c_i კლასს შეესაბამება 5 განსხვავებული დოკუმენტი (მოცულობით 600 სიტყვამდე) $d_{i_j}, j = 1, \dots 5$. ყოველი ტექსტისათვის გამოვყობთ 10 მაღალწონიან განსხვავებულ ტერმინს $t_{i_j}^m, m = 1, \dots, 10$. ყოველ c_i კლასისათვის გვექნება ასეთი ტერმინების 5 ნაკრები, ხოლო მთელი კოლექციისთვის 15 ნაკრები.

c_1 კლასისათვის გვექნება:

$$\begin{array}{lcl} d_{1_1} & \rightarrow & t_{11}^1, t_{11}^2, \dots, t_{11}^{10} \\ d_{1_2} & \rightarrow & t_{12}^1, t_{12}^2, \dots, t_{12}^{10} \\ \dots & \dots & \\ d_{1_5} & \rightarrow & t_{15}^1, t_{15}^2, \dots, t_{15}^{10} \end{array} \qquad (43)$$

c_2 კლასისათვის გვექნება:

$$\begin{array}{lll} d_{2_1} & \rightarrow & t_{21}^1, t_{21}^2, \dots, t_{21}^{10} \\ d_{2_2} & \rightarrow & t_{22}^1, t_{22}^2, \dots, t_{22}^{10} \\ \dots & \dots & \\ d_{2_5} & \rightarrow & t_{25}^1, t_{25}^2, \dots, t_{25}^{10} \end{array} \quad (44)$$

c_3 კლასისათვის გვექნება:

$$\begin{array}{lll} d_{3_1} & \rightarrow & t_{31}^1, t_{31}^2, \dots, t_{31}^{10} \\ d_{3_2} & \rightarrow & t_{32}^1, t_{32}^2, \dots, t_{32}^{10} \\ \dots & \dots & \\ d_{3_5} & \rightarrow & t_{35}^1, t_{35}^2, \dots, t_{35}^{10} \end{array} \quad (45)$$

ამ ნაკრებებიდან ყველა განსხვავებული $t_{i_j}^m$ ტერმინი აღვნიშნოთ $w_{i_j}{}^n$, n=1,…, N, სადაც N არის ტერმინების საერთო ნაკრებში განსხვავებული ტერმინების რაოდენობა ($10 \leq N \leq 150$). ვთქვათ N=20.

$$\begin{array}{c} c_1 \rightarrow \breve{w}_{11}^1 \& \breve{w}_{11}^2 \& \dots \& \breve{w}_{11}^{20} \vee \breve{w}_{12}^1 \& \breve{w}_{12}^2 \& \dots \& \breve{w}_{12}^{20} \vee \breve{w}_{13}^1 \& \breve{w}_{13}^2 \& \dots \& \breve{w}_{13}^{20} \\ \vee \breve{w}_{14}^1 \& \breve{w}_{14}^2 \& \dots \& \breve{w}_{14}^{20} \vee \breve{w}_{15}^1 \& \breve{w}_{15}^2 \& \dots \& \breve{w}_{15}^{20} \end{array} \quad (46)$$

$$\begin{array}{c} c_2 \rightarrow \breve{w}_{21}^1 \& \breve{w}_{21}^2 \& \dots \& \breve{w}_{21}^{20} \vee \breve{w}_{22}^1 \& \breve{w}_{22}^2 \& \dots \& \breve{w}_{22}^{20} \vee \breve{w}_{23}^1 \& \breve{w}_{23}^2 \& \dots \& \breve{w}_{23}^{20} \\ \vee \breve{w}_{24}^1 \& \breve{w}_{24}^2 \& \dots \& \breve{w}_{24}^{20} \vee \breve{w}_{25}^1 \& \breve{w}_{25}^2 \& \dots \& \breve{w}_{25}^{20} \end{array} \quad (47)$$

$$\begin{array}{c} c_3 \rightarrow \breve{w}_{31}^1 \& \breve{w}_{31}^2 \& \dots \& \breve{w}_{31}^{20} \vee \breve{w}_{32}^1 \& \breve{w}_{32}^2 \& \dots \& \breve{w}_{32}^{20} \vee \breve{w}_{33}^1 \& \breve{w}_{33}^2 \& \dots \& \breve{w}_{33}^{20} \\ \vee \breve{w}_{34}^1 \& \breve{w}_{34}^2 \& \dots \& \breve{w}_{34}^{20} \vee \breve{w}_{35}^1 \& \breve{w}_{35}^2 \& \dots \& \breve{w}_{35}^{20} \end{array} \quad (48)$$

ამ დიზუნქციების მინიმიზაცია მოგვცემს c_i კლასის შესაბამის „ბინარულ“ პატერნს. საბოლოოდ პატერნს ექნება ასეთი სახე:

$$\begin{array}{l} P^1 = I_1^1 \vee I_2^1 \vee I_{j_1}^1 \\ P^2 = I_1^2 \vee I_2^2 \vee I_{j_2}^2 \\ P^3 = I_1^3 \vee I_2^3 \vee I_{j_3}^3 \end{array} \quad (49)$$

სადაც I_{j_1}, I_{j_2}, I_{j_3}, მნიშვნელობები იცვლება 1-დან 5 მდე.

აღწერილი პროცესურის განხორციელების შემდეგ მიღებული ცოდნის „პატერნი“ აღწერს სხვადასხვა ცნებას განზოგადოებული სახით.

მანქანური სწავლების ალგორითმებზე დაყრდნობით განხორციელდა ამ მეთოდის პრაქტიკული რეალიზაცია დარგობრივი სფეროების ტექსტების ბაზებისათვის და შეფასდა შედეგები.

შემდეგ თავში ავღწერთ მეთოდის რეალიზაციას სხვადასხვა დარგობრივი სფეროებისათვის.

6.1.3. მეთოდის საცდელი შემოწმება.

ეს მოდული ოპერირებს ტექსტის საწყისი დამუშავების შედეგად მიღებული „ტერმინების ვექტორის“ საშუალებით. განისაზღვრება, რა ტექსტის მოცულობა საწყისი მოდულის მიერ დადგინდება ტერმინების ვექტორის სიგრძე და მოხდება მისი ფორმირება ზემოთ აღწერილი პროცედურების შესაბამისად.

ტესტირების საწყის ეტაპზე აღებული იქნა დოკუმენტების 6 კლასი (პოლიტიკა, იურისპრუდენცია, სოციოლოგია, კომპიუტერული მეცნიერება, სპორტი, ეკონომიკა). თითოეული კლასი შეიცავდა 200 დოკუმენტს. საბოლოოდ მიღებული იქნა დოკუმენტების კოლექცია, რომელიც შეიცავდა 1200 დოკუმენტს. თითოეული კლასიდან შემთხვევითობის პრინციპით აღებული იქნა 50 დოკუმენტი, რომელთა საფუძველზეც მოხდა სამდონიანი ცნებათა პატერნის აგება თითოეული კლასისათვის. სულ მიღებული იქნა 15 პატერნი, რომელთა საშუალებითაც მოხდა ცოდნის ბაზის ფორმირება. ტესტირების პროცესში მონაწილეობას იღებდა 900 დოკუმენტი.

კლასის დასახელება	დოკუმენტების საერთო რაოდენობა	დასასწავლი დოკუმენტების რაოდენობა	სატესტო დოკუმენტების რაოდენობა
პოლიტიკა	200	50	150
იურისპრუდენცია	200	50	150
სოციოლოგია	200	50	150
კომპიუტერული მეცნიერება	200	50	150
სპორტი	200	50	150
ეკონომიკა	200	50	150

ცხრილი 5 . მონაცემთა ბაზა

კლასიფიკაცია განხორციელდა მანქანური სწავლების სამი ალგორითმით: Bayes, KNN, SVM. შედეგების ანალიზმა ცხადყო KNN-ის და SVM -ის უპირატესობა Bayes ალგორითმთან შედარებით. თუმცა სისწრაფის მხრივ ეს უკანასკნელი გაცილებით სწრაფი აღმოჩნდა პირველ ორ ალგორითმთან შედარებით. კლასიფიკაციის შედეგები წარმოდგენილია ცხრილების და გრაფიკების სახით *(დანართი-ცხრილი 7, ცხრილი 8, ცხრილი 9, ცხრილი 10, სურათი 3. Precision-Recall მრუდი სამი ალგორითმისათვის)*

მეთოდის პრაქტიკული რეალიზაცია

რამდენიმე წელია საქართველოს შრომის, ჯანმრთელობისა და სოციალური დაცვის სამინისტრომ დაიწყო ჯანმრთელობის დაცვის ერთიანი საინფორმაციო

სისტემის დანერგვა, აქედან გამომდინარე პაციენტისათვის გაწეული სამედიცინო კვლევების შემცველი დოკუმენტების სტრუქტურირების და კლასიფიკაციის ამოცანა ერთ-ერთი აქტუალური ამოცანაა. წარმოდგენილი ნაშრომში აღწერილია სამდიცინო ჩანაწერების კლასიფიცირების მეთოდი ქართულენოვანი ტექსტებისათვის. ეს არის პირველი მცდელობა ასეთი ტიპის ტექსტების კლასიფიკაციის. კვლევებისათვის გამოყენებული იქნა 24.855 ჩანაწერი. დოკუმენტების კლასიფიკაცია განხორციელდა სამ ძირითად ჯგუფად (ულტრასონოგრაფია, ენდოსკოპია, რენტგენი) და 13 ქვეჯგუფად. ამოცანის გადაწყვეტისათვის გამოყენებული იქნა ორი კარგად ცნობილი მანქანური სწავლების ალგორითმი: მხარდამჭერი ვექტორების ალგორითმი (SVM) და K-უახლოესი მეზობლის ალგორითი (KNN). შედეგებმა აჩვენა რომ მანქანური სწავლების ორივე მეთოდი ეფექტურია, მაგრამ უკეთესი შედეგი გამოვლინდა SVM-ის გამოყენებისას. კლასიფიკაციის პროცესში ჩვენს მიერ შემუშავებული იქნა თვისებათა ამოკრების ჩვენეული მეთოდი, ე. წ. „ „შეკუმშვის“ მეთოდი, რომელმაც კლასიფიკაციის პირველ დონეზე საკმაოდ კარგი შედეგი მოგვცა. თუმცა მეორე დონეზე, რომელიც ქვეკლასებად კატეგორიას მოიცავდა შედეგი ცოტა გაუარესდა. დოკუმენტების 23%-ის მიკუთვნება ინდივიდუალური კლასებისათვის არ განხორციელდა. ბუნებრივია, აღნიშნული პრობლემის ძირითადი მიზეზი იყო ქვეკლასებში კვლევის შედეგების აღწერებში ერთნაირი ტერმინების არსებობა.

მთლიანობაში მეთოდის გამოყენებამ კარგი შედეგები აჩვენა და იგი წარმატებული აღმოჩნდა სამედიცინო ჩანაწერების კლასიფიკაციის ამოცანაში.

პირველი სამედიცინო მონაცემების დამუშავების სისტემის შექმნიდან ნახევარი საუკუნე გავიდა (87). ამ სისტემებმა საფუძველი ჩაუყარეს თანამედროვე სამედიცინო ინფორმატიკას და EMR სისტემების დანერგვა განვითარებას მსოფლიოს მრავალ ქვეყანაში (88). სამედიცინო მონაცემების დამუშავების ერთ ერთი მნიშვნელოვანი ამოცანა არის მონაცემების ჩანაწერების კლასიფიკაცია, რომლის განხორციელებისათვის არსებობს სხვადასხვა მეთოდები. შედეგების ანალიზი ხორციელდება ისეთი სიდიდეების შედარებით, როგორიცაა სიზუსტე, სისრულე, F ზომა. (89) ბუნებრივია შედეგებზე გავლენას ახდენს სხვადასხვა ფაქტორი, მაგალითად როგორიცაა ბაზის სტრუქტურა (მოცულობა, მონაცემთა ტიპები) თვისებების ამოკრების მეთოდები და ა.შ.

ზოგადად, არსებობს ჩანაწერების კლასიფიცირების რამდენიმე მეთოდი: კლასიფიცირება ხელით, კლასიფიცირება სისტემით, რომელიც დაფუძნებულია წესებზე და ლექსიკონებზე და ჰიბრიდული სისტემები, რომლების სხვადასხვა პროცესს ერთდროულად მოიცავს: დაფუძნებულია წესებზე და ამასთანავე იყენებს მანქანური სწავლების მეთოდებს. (90).

2011 წელიდან საქართველოს შრომის, ჯანმრთელობისა და სოციალური დაცვის სამინისტრომ დაიწყო ჯანმრთელობის დაცვის ერთიანი საინფორმაციო სისტემის დანერგვა, რომელიც მოემსახურება სამინისტროს, სადაზღვევო კომპანიების, სამედიცინო მომსახურების მიმწოდებლებისა და პაციენტების საინფორმაციო საჭიროებების უზრუნველყოფას. ამ სისტემის ერთერთ მთავარ რგოლს წარმოადგენს მოდული „ელექტრონული სამედიცინო ისტორია“, ელექტრონული სამედიცინო ისტორიების წარმოების სისტემის დანიშნულებაა პაციენტის სამედიცინო დახმარებისთვის მნიშვნელოვანი კლინიკური ინფორმაციის შეგროვების, შენახვის,

დამუშავებისა და ხელმისაწვდომობის უზრუნველყოფა. სისტემა ძირითადად ფოკუსირებულია კლინიკურ მონაცემებზე. სრული სახით ფუნქციონირებადმა სისტემამ უნდა მოიცვას კლინიკური ინფორმაცია სრულად, რომელიც კავშირშია პაციენტისთვის ადრე გაწეულ, ახლა მიმდინარე და რიგ შემთხვევაში მომავალში დაგეგმილ სამედიცინო დახმარებასთან. შესაბამისად სისტემაში უნდა განთავსდეს ინფორმაცია პაციენტის დემოგრაფიული მონაცემების, სამედიცინო პრობლემების, მედიკამენტების, ვიტალურ ნიშნების, სამედიცინო ანამნეზის, იმუნიზაციის, ლაბორატორიული და სხივური დიაგნოსტიკით მიღებულ მონაცემების შესახებ, რათა მოხდეს კლინიკური დოკუმენტბრუნვის ავტომატიზაცია და რაციონალიზაცია.

მონაცემები

დასამუშავებელ მასალას წარმოადგენს ეგრეთწოდებული „სამედიცინო ჩანაწერების დომენი“, რომელიც მოიცავს საქართველოს ერთერთ კლინიკაში 2012-2014 წელს სხვადასხვა დროს პაციენტებზე ჩატარებული ინსტრუმენტული დიაგნოსტიკის აღმწერ დოკუმენტებს. დოკუმენტები წარმოდგენილია მცირე მოცულობის (არაუმეტეს 300-350 სიტყვა) თავისუფალი ტექსტებით .doc ან .docx ფორმატის). დოკუმენტები აღწერენ სხვადასხვა ორგანოებზე ჩატარებულ ულტრასონოგრაფიის, რენტგენის და ენდოსკოპიის კვლევების ჩანაწერებს. რიგი ჩანაწერი განეკუთვნება ერთი და იმავე პაციენტზე სხვადასხვა დროს ჩატარებულ განსხვავებული და ასევე განმეორებითი ტიპის კვლევებს. ულტრასონოგრაფიის დომენი მოიცავს სხვადასხვა ორგანოს (ღვიძლისა და სანაღვლე სისტემის ულტრასონოგრაფია, თირკმელების და შარდ-სასქესო სისტემის ულტრასონოგრაფია, გინეკოლოგიური ულტრასონოგრაფია, ფარისებრი ჯირკვლის, სარძევე ჯირკვლის ულტრასონოგრაფია, სისხლძარღვების დოპლერ-ულტრასონოგრაფია) გამოკვლევის 12864 ჩანაწერს, რენტგენი - სხვადასხვა ორგანოს (გულმკერდის, მუცლის ღრუს, ხერხემლის, კიდურების, საყლაპავის და კუჭის, მსხვილი და წვრილი ნაწლავების რენტგენოლოგიური კვლევა) გამოკვლევის 10523 ჩანაწერს, ენდოსკოპია - გამოკვლევის 1468 ჩანაწერს. სხვადასხვა ქვეჯგუფებისათვის ჩანაწერების რაოდენობა 500-დან 1000-მდეა. გამონაკლისია ენდოსკოპია, რომელსაც ქვეჯგუფი არ აქვს. *(დანართი-ცხრილი 11)*

მეთოდები

სამედიცინო ჩანაწერების კლასიფიკაცია, შეიძლება განხილული იქნეს, როგორც ინფორმაციული ძებნის ერთ-ერთი ქვეამოცანა - ტექსტების კლასიფიკაცია, რომელიც ხორციელდება არასტრუქტურირებულ სამედიცინო ჩანაწერებზე ინფორმაციული ძებნის მეთოდების და მანქანური სწავლების ალგორითმებით. ბუნებრივი ენის ანალიზი (NLP) ამ ამოცანის გადაწყვეტისათვის ყველაზე უფრო მისაღები და გამოყენებადია (91), (92), (93). ბიოინფორმატიკის სფეროში სხვადასხვა სამეცნიერო ჯგუფების მიერ ჩატარებული კვლევების საფუძველზე დადგენილია, რომ მანქანური სწავლების ალგორითმების გამოყენება კლასიფიკაციის ამოცანისათვის არასტრუქტურირებული ელექტრონული ჩანაწერების ბაზიდან, რომელიც უამრავ სამედიცინო ისტორიას შეიცავს ერთ-ერთი საუკეთესო გზაა (94) (95), (96).

ამასთანავე კლასიფიკაციის ამოცანა მოიცავს ექიმის მიერ ბუნებრივ სალაპარაკო ენაზე გაკეთებული ჩანაწერიდან გარკვეული თვისებების ამოკრების პროცესს,

რომლებიც იღებენ მონაწილეობას ექიმის მიერ დიაგნოზის აღწერაში, ამიტომ მათი გამოყენება მნიშვნელოვანია კლასიფიკაციისათვის. (97).

კვლევების საფუძველზე დადგენილია, რომ ავტომატური მეთოდებით სამედიცინო ჩანაწერებიდან ინფორმაციის ამოღება უფრო ეფექტურია, ვიდრე ხელით. საინტერესო კვლევაა სტატიაში მოცემული: 5232 მონაცემის იდენტიფიკაციას ხელით დაჭირდა 176 საათი, ხოლო ავტომატურად მისი ამოღება სისტემით მოხდა 4,5 საათში. (98).

ინფორმაციული ძებნის მეთოდებით და მანქანური სწავლების ალგორითმებით წარმატებით განხორციელდა სამედიცინო ბაზიდან მონაცემების კლასიფიკაცია ბრაზილიის პედიატრიულ ინსტიტუტში (99).

ყველაზე კარგად ცნობილ ალგორითმებს შორის SVM არის ერთ-ერთი ყველაზე უფრო პოპულარული და ხშირად გამოყენებადი მოკლე ტექსტების ბაზისათვის (100). SVM-ის გამოყენებამ ისეთი დაავადებების კლასიფიკაციისას, როგორიცაა სიმსივნე და დიაბეტი საკმაოდ კარგი შედეგი აჩვენა (101), (102), (103).

დოკუმენტის კლასიფიცირების ამოცანა ემყარება ტექსტის ინდექსაციას სტემინგის გამოყენების გზით, რომელიც მოსახერხებელია არა მარტო ინფომაციის ძებნის, არამედ მანქანური სწავლების მოდელების შემუშავებისათვისაც. გავითვალისწინეთ რა ქართული ენის თავისებურებები, ჩვენ გამოვიყენეთ სპეციალურად ქართული ენისათვის შემუშავებული ახალი სტემინგის ალგორითმი ტექსტის დამუშავების საწყის მოდულში, რომელიც ზემოთ გვქონდა აღწერილი, და შემდგომ მოვარგეთ KNN და SVM (104), (105).

სრული სახით ჩვენი ამოცანა წარმოდგება, როგორც რამდენიმე დამოუკიდებელი ამოცანა:

- ჩანაწერების წინასწარი დამუშავება-დეიდენტიფიკაცია
- ტექსტების საწყისი დამუშავება - ტოკენიზაცია, ლემატიზაცია - ქართული ენის თავისებურებების გათვალისწინებით;
- თვისებების ამოკრების და წონების დათვლის პროცესი;
- ტექსტების კლასიფიკაცია - ჩატარებული კვლევის ტიპის მიხედვით სამედიცინო ჩანაწერების კლასიფიცირება სტრუქტურირება ელექტრონული სამედიცინო ისტორიების სისტემაში განსათავსებლად;
- ჩანაწერის იდენტიფიკაცია პაციენტის მიხედვით.

6.2.1. ჩანაწერების დეიდენტიფიკაცია

ელექტრონული სამედიცინო ჩანაწერების ზრდასთან ერთად, იზრდება კლინიკური მონაცემების რაოდენობაც. მიუხედავად ამისა, მათი გამოყენება კომპანიების, ორგანიზაციებისა და მკვლევარებისათვის შეზღუდულია, იმის გამო რომ ისინი შეიცავენ ადამიანის ჯანმრთელობასთან დაკავშირებულ პერსონალურ ინფორმაციას. დე-იდენტიფიკაცია არის აუცილებელი ეტაპი მონაცემების ხელმისაწვდომობისა და გამოყენების შესაძლებლობისათვის.

დე-იდენტიფიკაცია, შეიძლება განვიხილოთ როგორც ტრადიციული სახეთა ამოცნობის ამოცანა, თუმცა მას აქვს თავისებურებანი (მაგ. სიტყვა ან ფრაზა არის თუ არა

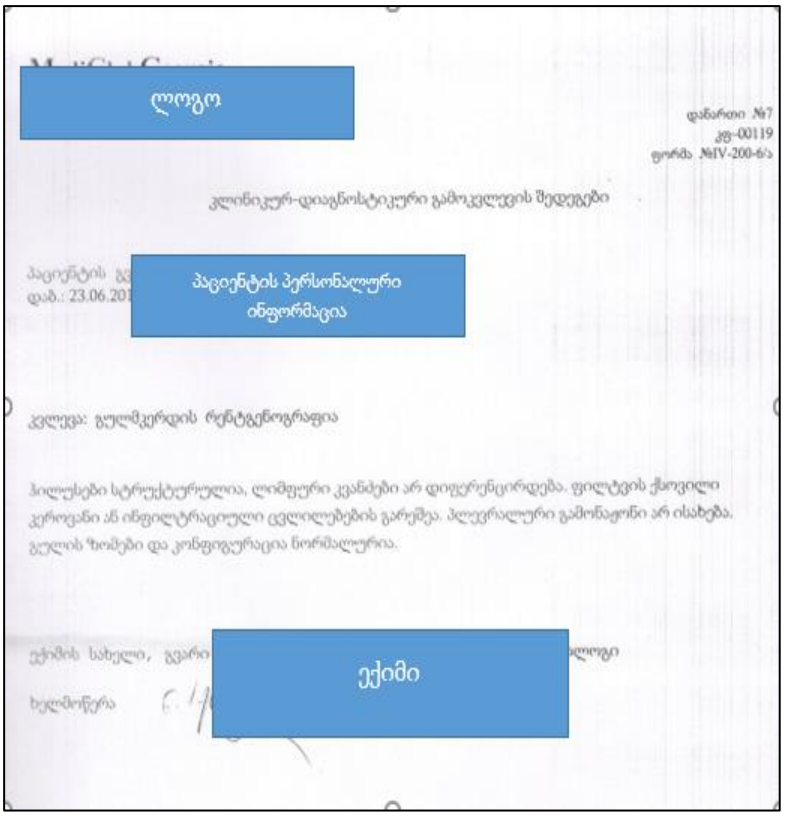
პერსონალური ინფორმაციის შემცველი). 1996 წელს შეიქმნა HIPPA[12] , სამედიცინო ჩანაწერების პორტატულობის და ნდობის აქტი, რომლითაც აღიწერა ყველა სახის სამედიცინო ჩანაწერი (106). მიუხედავად იმის, რომ უკანასკნელ წლებში შეიქმნა დე-იდენტიფიკაციის სხვადასხვა სისტემები (107) (108), ერთიანი პლატფორმა, HIPPA-ში განსაზღვრული ნებისმიერი ტიპის სამედიცინო ჩანაწერის წაკითხვის, არ არსებობს.

2014 წელს, i2b2-ზე დაყრდნობით განხორციელდა ბუნებრივი ენის ტექსტებით ჩაწერილი სამედიცინო ჩანაწერების დაყოფა 7 კატეგორიად და 25 ქვეკატეგორიად, რომელიც მოიცავს HIPPA-თი განსაზღვრულ 18 სახის ჩანაწერს. შემუშავდა სხვადასხვა მიდგომა დე-იდენტიფიკაციისათვის და შესაბამისად მოხდა სისტემის შეფასებები. აღმოჩნდა, რომ საუკეთესოს გამოყოფა შეუძლებელია, თუმცა გამოიკვეთა ის ფაქტი, რომ მანქანურ სწავლებაზე დაფუძნებული სისტემები უფრო ეფექტურია, ვიდრე ის სისტემები, რომლებიც იყენებენ წესებს (109).

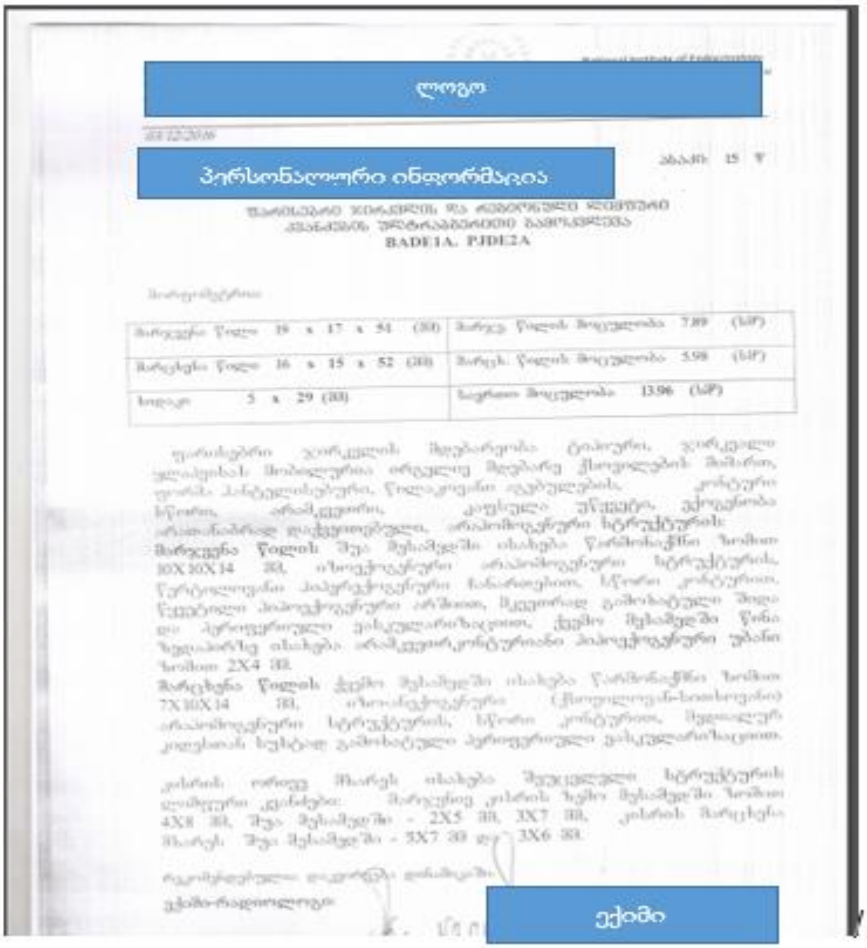
ჩვენს მიერ განხორციელებულ კლასიფიკაციის ამოცანაში, გამოყენებული დოკუმენტების (სამედიცინო ჩანაწერების) კოლექცია, შეიცავს პერსონალურ მონაცემებს როგორც ავადმყოფის, ასევე ექიმის შესახებ. საწყის ეტაპზე კლასიფიცირების ამოცანისათვის ხორციელდება ჩანაწერების დე-იდენტიფიკაცია. ექიმის მიერ პაციენტის მდგომარეობის შესახებ შესავსები ფორმის ანალიზით დადგინდა, რომ პაციენტის იდენტიფიცირება ძირითადად ხორციელდება შემდეგი ველებით: გვარი, სახელი, დაბადების თარიღი, ასაკი და კვლევის ჩატარების თარიღი. ავადმყოფის მონაცემები განთავსებულია ფორმის დასაწყისში რამდენიმე სტრიქონად, ხოლო ექიმის ვინაობის შესახებ ინფორმაცია ფორმის სულ ბოლო სტრიქონზე.

ძირითადად პაციენტის გასინჯვის ფორმა ასეთი სახისაა:

[12] Health Incurance Portability and Accountability Act



სურათი 1 პაციენტის გასინჯვის ფურცელი.



სურათი 2 პაციენტის გასინჯვის ფურცელი.

ინფორმაცია პაციენტის იდენტიფიცირებისათვის მოთავსებულია ტექსტის თავში, ხოლო მკურნალი ექიმის შესახებ ჩანაწერი გვხვდება ტექსტის ბოლოს. ჩანაწერების დე-იდენტიფიკაციისათვის ვახდენთ ტექსტიდან პირველი რამდენიმე სტრიქონის და სულ ბოლო სტრიქონის წაშლას შემდეგი წესით:

- ვეძებთ სიტყვებს: „გვარი“, „სახელი“, „ასაკი“, „დაბადების თარიღი“ და ვიმახსოვრებთ იმ სტრიქონის ნომრებს სადაც ისინი გვხვდებიან;
- მათგან ვარჩევთ უდიდეს რიცხვს რომელიც ნაკლები ან ტოლი იქნება 5-ზე (რადგან მაქსიმალური ოდენობა სტრიქონებისა, რაც განკუთვნილია პაციენტის იდენტიფიცირებისათვის არის 5);
- ვახდენთ შესაბამისი ოდენობის სტრიქონების წაშლას დოკუმენტის დასაწყისიდან.

ამ პროცედურების გავლის შემდეგ დოკუმენტიდან მოცილებულია პაციენტის და ექიმის მაიდენტიფიცირებელი ჩანაწერები. ასეთი დოკუმენტი უკვე შესაძლებელია გამოყენებულ იქნეს კვლევისათვის.

6.2.2. დოკუმენტების საწყისი დამუშავება

სამედიცინო ტექსტების დამუშავების ბევრი განსხვავებული მეთოდი არსებობს (75). ისინი ეფუძნებიან UMLS[13]-ის გამოყენებას, განსაკუთრებით მანქანური სწავლებისას (110), (111).

UMLS წარმოადგენს სპეციალურ სისტემას, რომელიც შეიცავს ბიოსამედიცინო და ჯანდაცვის სფეროში არსებული ტერმინების სრულ სიას. ამ ტერმინებზე დაყრდნობით შექმნილია სპეციალური ლექსიკონები, რომელთა გამოყენებაც აუმჯობესებს კომპიუტერული სისტემების მუშაობის ეფექტურობას. ამ სისტემის გამოყენება ჩვენს შემთხვევაში გართულებულია ქართულენოვანი შესატყვისის არ არსებობის გამო.

კლასიფიკაციის თითქმის ყველა ცნობილი ალგორითმი იყენებს მის მინიმიზაციას, მასში მხოლოდ ყველაზე ხშირად განმეორებადი სიტყვების დატოვების გზით. სიტყვების სიხშირის გამოსათვლელად საჭიროა სიტყვების დამუშავება, რომ ერთი სიტყვის სხვადასხვა ფორმა განსხვავებულ სიტყვებად არ მივიჩნიოთ. სიტყვის დამუშავებაში იგულისხმება: სიტყვის იმ ნაწილის დატოვებას, რაც ფორმაწარმოების დროს უცვლელი რჩება (112).

სტემინგის ყველაზე ცნობილი ალგორითმი - პორტერის ალგორითმი ამ ამოცანას წარმატებით ართმევს თავს, მაგრამ მისი გამოყენება განსხვავებული გრამატიკული სტრუქტურის გამო თითქმის შეუძლებელია ქართული ენის და მისი მსგავსი ენებისათვის (113), (114).

ახალი მეთოდი, რომელიც შემუშავდა თბილისის სახელმწიფო უნივერსიტეტის მიზნობრივ სამცნიერო პროექტში, საწყის ეტაპზე ახდენს ტექსტის დამუშავებას (115). ჩვენს მიერ გამოყენებული ყველა ტექსტი, რომელიც მონაწილეობს როგორც კლასიფიკაციის პროცედურის შემუშავებაში, ასევე სისტემის სატესტო შემოწმებაში, ექვემდებარება დამუშავების სრულ პროცესს, რაც გულისხმობს:

1. ე.წ. "სტოპ" სიტყვების ბაზის შექმნა (ქართულენოვანი ტექსტებისათვის ასეთი სიტყვებია კავშირები, შორისდებულები, ნაცვალსახელები) და მათ ჩანაცვლება "ჯოკერებით", აგრეთვე რიცხვითი მონაცემების წაშლა;
2. არაქართული ტექსტების იდენტიფიკაცია და მათი ჩანაცვლება ქართული შესატყვისებით (ზოგიერთი სამედიცინო ტერმინი მოცემული იყო ინგლისურ ენაზე);
3. სტემინგის ოპერაცია, რომელიც ქართული ენის სუფიქსების და არსებითი სახელების ბაზაზე დაყრდნობით უზრუნველყობს სიტყვიდან ყველაზე დიდი სუფიქსის მოცილებას და სიტყვიდან ფუძის ამოღებას.

პორტერის ალგორითმისგან განსხვავებით ჩვენს მიერ შემუშავებული მეთოდი ფუძის მიღებისას იყენებს სიტყვების ბაზას. თუმცა სამედიცინო ტექსტებისათვის ბაზის გამოყენება წარმოშობს რიგ პრობლემებს – ბაზის სისრულე სამედიცინო ტერმინების მიმართ, ბაზის საიმედობა (შინაარსის თვალსაზრისით) და ბაზის განახლებადობა. ამ სიძნელეების გადასალახად ქართული სიტყვების საწყისი ბაზა (99350 სიტყვა) გაფართოვდა ICD-10[14]-ში გამოყენებული ქართულენოვანი სამედიცინო

[13] Unified Medical Language System's
[14] the 10th revision of the International Statistical Classification of Diseases and Related Health Problems

ტერმინოლოგიით (116). ICD-10 წარმოადგენს მსოფლიო ჯანდაცვის ორგანიზაციის მიერ შემუშავებულ სამედიცინო ტერმინოლოგიის ბაზას.

6.2.3. თვისებების ამოღება

ტექსტების საწყისი დამუშავების შედეგს წარმოადგენს ინდექსირებული დოკუმენტი, რომელიც წარმოდგენილია ნიშან-თვისებათა სივრცეში გამოსახული ვექტორის სახით. ვექტორის ელემენტს წარმოადგენს წყვილი „ტერმინი/წონა“ წონის განსაზღვრა მოხდა სტნდარტული TF-IDF სქემით (24):

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{df_i} \qquad (50)$$

სადაც

$w_{i,j}$- არის i-ური ტერმიინის წონა j-ურ დოკუმენტში; $tf_{i,j}$- არის i-ური ტერმინის სიხშირე j-ურ დოკუმენტში; N - არის დოკუმენტების რიცხვი კოლექციაში. df_i- არის კოლექციაში იმ დოკუმენტების სიხშირე, რომლებიც შეიცავენ i-ურ ტერმინს.

როგორც ცნობილია, დოკუმენტების რაოდენობის ზრდა იწვევს სიტყვათა ლექსიკონის ზრდას (117). შესაბამისად იზრდება ნიშან-თვისებათა სივრცეც. ამიტომ, რიგ შემთხვევაში, მიმართავენ კლასიფიკაციისათვის საჭირო ტერმინების რაოდენობის შემცირებას.

ტერმინების სივრცის შესამცირებლად იყენებენ სხვადასხვა მეთოდებს. იმ ტერმინებს, რომლებიც ყველაზე უფრო მნიშვნელოვანი და ხშირად გამოყენებადია, ტოვებენ, ხოლო დანარჩენების იგნორირებას ახდენენ (118).

ნიშან-თვისებათა სივრცის განზომილების შესამცირებლად ჩვენს მიერ გამოყენებული მიდგომა განსხვავებულია. ქართული ენის თავისებურებიდან გამომდინარე, ზოგადად ყველა ტექსტში „სტოპ“ სიტყვების ხვედრითი წილი მეტად მცირეა (<< 1%). სამი ძირითადი კლასის, 14 განსხვავებული ქვეკლასის დოკუმენტების ტექსტების საწყისი დამუშავების შედეგად მიღებული ნიშან-თვისებათა სივრცეში გამოსახული ვექტორების ანალიზმა აჩვენა, რომ ყოველი ქვეკლასის აღმწერი ტერმინების სივრცეში გვხვდებიან ტერმინები, რომლებიც ერთი და იგივეა ყველა სამი კლასისათვის (სულ ასეთი ტერმინის რაოდენობაა 21). კლასებში შემავალი ქვეკლასების შესაბამისი ვექტორების ანალიზით აღმოჩნდა, რომ ერთი და იგივე ტერმინების რაოდენობა ულტრასონოგრაფიული გამოკვლევების შესაბამისი ტექსტებისათვის ტოლია 10-ის (მდებარეობა, სტრუქტურა, ფორმა, კონტური, ორგანო, მაჩვენებელი, ზომა, ქსოვილი, სიდიდე, კერა), ხოლო რენტგენული კვლევებისათვის - 17 (ქსოვილი, სტრუქტურა, ფორმა, კონტური, ორგანო, მაჩვენებელი, სეგმენტი, ზომა, სიდიდე, არე, სურათი, კვლევა, ცვლილება). ამ სიტყვებს ვუწოდეთ „ფსევდო სტოპ სიტყვები“ და ამოვიღეთ თვისებათა სივრციდან. შესაბამისად ვექტორს ვუწოდეთ **„შეკუმშული”** ვექტორი.

6.2.4. კლასიფიკატორი

კლასიფიკაციისათვის საჭირო მანქანური სწავლების ალგორითმის შერჩევისათვი გავითვალისწინეთ ჩვენს მიერ განხორციელებული კლასიფიკაციის ამოცანის შედეგები,

რომელზეც ზევით გვქონდა საუბარი. აქ შედეგები საუკეთესო აღმოჩნდა ორი ალგორითმისათვის KNN და SVM. ამიტომ ამ ეტაპზე, არჩევანი გავაკეთეთ სწორედ ამ ალგორითმებზე. კლასიფიკაციის ამოცანაში ამ ალგორითმების გამოყენების ერთ-ერთი მიზეზი იყო აგრეთვე ის თავისებურებანი, რაც ახასიათებთ ალგორითმებს: KNN -ს- სიმარტივე (112), ხოლო SVM -ს- მაღალი სიზუსტე და სისრულე (119). მხედველობაში მივიღეთ ის ფაქტიც, რომ ორივე მათგანი საკმაოდ წარმატებულად გამოიყენება სამედიცინო ჩანაწერების კლასიფიკაციის ამოცანებში (ეს ალგორიმები დაწვრილებით განვიხილეთ მე-4 თავში).

ჩვენი ამოცანა გაიყო ორ ნაწილად: თავდაპირველად საჭირო იყო ორი ალგორითმიდან შეგვერჩია ერთ-ერთი. ამიტომ ალგორითმების გამოყენებით განვახორციელეთ სამედიცინო ჩანაწერების ტექსტების კლასიფიკაცია თვისებათა შერჩევის კლასიკური მეთოდით, რათა კვლავ გამოგვეყო ჩვენთვის კიდევ უფრო მისაღები ვარიანტი.

ზემოაღნიშნული კლასიფიკატორების გამოყენებით ტექსტის კლასიფიკაცია განხორციელდა 2 ეტაპად: პირველ ეტაპზე მოხდა დოკუმენტების კატეგორიზაცია ჩატარებული კვლევის ტიპის მიხედვით: სულ სამი ტიპი - ულტრასონორგაფია, რენტგენი, ენდოსკოპია. აქვე უნდა აღინიშნოს, რომ ყოველი დოკუმენტი შეიძლება განეკუთვნებოდეს მხოლოდ ერთ კლასს. ამ ეტაპის გავლის შემდეგ, მეორე ეტაპზე, მოხდა დოკუმენტის მიკუთვნება შესაბამისი კლასის ქვეკლასზე (ულტრასონოგრაფიის შემთხვევაში - 7 ქვეკლასი, რენტგენი - 6 ქვეკლასი, ენდოსკოპიას ქვეკლასი არ გააჩნია). ტექსტის საწყისი დამუშავება ორივე მეთოდისათვის ერთნაირად განხორციელდა.

განისაზღვრა შედეგების შეფასების კრიტერიუმები და შესაბამისად მიღებული შედეგები წარმოდგენილია ცხრილის სახით (**R** - Recall, **P -** precision, **F -** F measure, **ERR -** Error Rate, **Acc -** Accuracy) (*დანართი - ცხრილი 12)*

SVM-ით განხორციელებული კლასიფიკაცია უკეთესი აღმოჩნდა, ამიტომ თვისებათა ამოკრების ორივე მეთოდის ტესტირება (კლასიკური და „შეკუმშული“) მოხდა SVM-ის შემთხვევაში. მანქანური სწავლებისათვის ჩვენ გამოვიყენეთ როგორც კლასიკური დოკუმენტების ვექტორი(რომელიც შეიცავს ყველა ნიშან-თვისებას) აგრეთვე „შეკუმშული“ (შემცირებული ნიშან-თვისებებით). შედეგები მოცემულია ცხრილებში *(დანართი-ცხრილი 13, ცხრილი 14, სურათი 4)*

შედეგებმა აჩვენეს, რომ კლასიფიკაციის პირველ ეტაპზე, ნიშან-თვისებათა შერჩევის ორივე მეთოდი (კლასიკური და შეკვეცილი) საკმაოდ წარმატებული იყო, შედეგებში მცირეოდენი განსხვავებით. თითოეულ დოკუმენტს განესაზღვრა მხოლოდ ერთი კლასი, თუმცა გამონაკლისს წარმოადგენდა ღვიძლის და სანაღვლე სისტემის გამოკვლევების შედეგები.

საბოლოოდ, შეიძლება ითქვას, რომ დოკუმეტების დაყოფა ქვეკლასებად ძირითადად წარმატებული აღმოჩნდა თუ არ ჩავთვლით ღვიძლის და სანაღვლე სისტემის გამოკვლევების შედეგებს.

შედეგები, რომელიც მივიღეთ კლასიფიკაციის მეორე ეტაპზე, კლასიკური სივრცისათვის იყო უკეთესი ვიდრე შეკვეცილისათვის. მეორე დონის კლასიფიკაციას,

რომელიც მოიცავდა მონაცემების მიკუთვნებას ქვეკლასებისათვის, დოკუმენტების 23%-ის მინიჭება ფაქტიურად არ მოხდა არცერთი ქვეკლასისათვის, რამაც გამოიწვია სისტემის შეფასების შედეგების გაუარესება. ამ შემთხვევაში თვისებათა ამოკრების კლასიკური მეთოდი „შეკუმშულ“ მეთოდთან შედარებით უფრო წარმატებული აღმოჩნდა (დანართი- *ცხრილი 15*).

ჩვენ მიგვაჩნია, რომ ნიშან-თვისებათა არის შეკვეცა შესაძლებელია იყოს მისაღები, როდესაც კლასების ბუნება განსხვავებულია. ჩვენს შემთხვევაში, მეორე დონის კლასიფიკაციისას, ქვეკლასების აღწერა ხდებოდა თითქმის ერთნაირი ტერმინებით, რამაც გაართულა ამოცნობის პროცესი. თუმცა, კლასიფიკაციის პირველ დონეზე ეს პრობლემა არ ყოფილა.

საბოლოოდ მივიღეთ, რომ SVM მეთოდი, რომელიც გამოვიყენეთ ქართულენოვანი სამედიცინო ტექსტების კლასიფიკაციისას უკეთესი შედეგის მომტანი იყო ვიდრე KNN, თუმცა ორივე მეთოდმა მეტ-ნაკლებად დადებითი შედეგები მოგვცა.

მეექვსე თავის დასკვნა

ამ თავში ავღწერეთ ქართულენოვანი ტექსტების კლასიფიცირების სისტემის ძირითადი მოდულები, რომლებშიც შეძლებისდაგვარად გათვალისწინებული იქნა ქართული ენის თავისებურებანი. ანალიტიკური ევრისტიკების მეთოდის გამოყენებით, დოკუმენტების კოლექციიდან მოხდა კონკრეტული ცნების აღმწერი კონცეპტ-პატერნების შემუშავება. პორტერის ალგორითმის მოდიფიცირებული ვარიანტით განხორციელდა კლასის აღმწერი ნიშან-თვისებების სიმრავლის გამოყოფა და შესაბამისად წონების დათვლა tf-idf სტანდარტული სქემით.

სისტემის ტესტირება განხორციელდა 6 კატეგორიის 900 დოკუმენტისაგან შედგენილ დოკუმენტების ბაზაზე. მიღებულმა შედეგებმა აჩვენა კონცეპტების გამოყენების უპირატესობა სხვა მეთოდებთან შედარებით კლასიფიკაციის ამოცანების განხორციელებისათვის.

ამ ამოცანის განხორციელების შემდეგ, მოხდა მისი პრაქტიკული რეალიზაცია ქართულენოვანი სამედიცინო ჩანაწერების კლასიფიკაციისათვის. წარმოდგენილი იქნა სამდიცინო ჩანაწერების კლასიფიცირების მეთოდი ქართულენოვანი ტექსტებისათვის. აღსანიშნავია, რომ ეს არის ასეთი ტიპის ტექსტების კლასიფიკაციის პირველი მცდელობა. სრული სახით ჩვენი ამოცანა შეიცავდა რამდენიმე მოდულს: ჩანაწერების წინასწარი დამუშავება-დეიდენტიფიკაცია, ტექსტების საწყისი დამუშავება - ტოკენიზაცია, ლემატიზაცია - ქართული ენის თავისებურებების გათვალისწინებით, თვისებების ამოკრების და წონების დათვლის პროცესი, ტექსტების კლასიფიკაცია - ჩატარებული კვლევის ტიპის მიხედვით სამედიცინო ჩანაწერების კლასიფიცირება სტრუქტურირება ელექტრონული სამედიცინო ისტორიების სისტემაში განსათავსებლად, ჩანაწერის იდენტიფიკაცია პაციენტის მიხედვით (უკანასკნელი მოდული სამომავლო ამოცანად განიხლება).

კვლევებისათვის გამოყენებული იქნა 24.855 ჩანაწერი. დოკუმენტების კლასიფიკაცია განხორციელდა სამ ძირითად ჯგუფად (ულტრასონოგრაფია, ენდოსკოპია, რენტგენი) და 13 ქვეჯგუფად. ამოცანის გადაწყვეტისათვის გამოყენებული იქნა ორი კარგად ცნობილი მანქანური სწავლების ალგორითმი: მხარდამჭერი ვექტორების (SVM) და უახლოესი მეზობლის (KNN) ალგორითმები. შედეგებმა აჩვენა რომ მანქანური სწავლების ორივე მეთოდი საკმაოდ შედეგინია, მაგრამ უკეთესი შედეგი გამოვლინდა SVM-ის გამოყენებისას. კლასიფიკაციის პროცესში ჩვენს მიერ შემუშავებული იქნა თვისებათა ამოკრების ჩვენეული, ე. წ. „ „შეკუმშვის“ მეთოდი, რომელმაც კლასიფიკაციის პირველ დონეზე, კლასიკურ მეთოდთან შედარებით, საკმაოდ კარგი შედეგი მოგვცა.

მთლიანობაში მეთოდის გამოყენებამ კარგი შედეგები აჩვენა და იგი წარმატებული აღმოჩნდა.

დასკვნა

მოცემული კვლევის სტრატეგია ეფუძნება სამომავლოდ უკეთესი განვითარების პერსპექტივაზე აქცენტირებულ ინფორმაციული ძებნის ტექნოლოგიას, რომლის რეალიზაციაც სემანტიკური ძებნის პირობებშია შესაძლებელი და კვლევის შედეგებიც ხვალინდელ დღეზეა ორიენტირებული.

ტექსტების კლასიფიკაციის ამოცანა არ არის ახალი, კვლევები ამ მიმართულებით საკმაოდ ინტენსიურად მიმდინარეობს, ძირითადად ინგლისურენოვანი ტექსტებისათვის. მთელი რიგი კვლევები მიეძღვნა სხვა ენებსაც, გამონაკლისია ქართული ენა, რომელზეც აღწერილი ტიპის კვლევები და შესაბამისად კლასიფიკაციის ამოცანა არ განხორციელებულა. ამიტომ კლასიფიკაციის ეს ამოცანა არის სრულიად ახალი.

ამოცანის განსახორციელებლად გამოყენებული მანქანური სწავლების ორი ალგორითმი: SVM და KNN საკმაოდ ეფექტური აღმოჩნდა კლასიფიკაციის ამოცანის გადაწყვეტაში, თუმცა SVM-მა, KNN- თან შედარებით, ცოტა უფრო უკეთესი შედეგები მოგვცა, ამიტომ კლასიფიკაციის პროცესი, თვისებათა ამოკრების „შეკუმშული“ მეთოდით, განვახორციელეთ SVM-ით.

ჩვენს მიერ განხორციელებული სამუშაო წარმოადგენს ინსტრუმენტს, რომლის საფუძველზე შესაძლებელია აიგოს სრულფასოვანი სისტემა ქართულენოვანი ჩანაწერების კლასიფიკაციისათვის, რომელიც შესაძლებელია გამოყენებულ იქნეს ქართულენოვანი საძიებო სისტემის ძრავის შემადგენელ კომპონენტად.

ბიბლიოგრაფია

1. **C.J.Van Rijsbergen.** *INFORMATION RETRIEVAL.* London : Butterworths, pp.1-153. 1979.

2. As We May Think. *Atlantic Monthly* **. Vannevar Bush**. The Atlantic Monthly. pp. 1-8. July 1945.

3. *A statistical approach to mechanized encoding and searching of literary information.* **H.P.Luhn.** IBM Journal of Research and Development, pp. 309-317.1957

4. **Gerard Salton.** *The SMART Retrieval System—Experiments in Automatic Document Retrieval.* Englewood Cliffs, NJ : Prentice Hall Inc. pp.1-556. 1971.

5. *The Cranfield tests on index language devices.* **C.W.Cleverdon.** Aslib Proceedings, pp.173-194.. 1967

6. **C.Mooers.** *The theory of digital handling of non-numerical information and its implications to machine economics.* Boston, Zator. pp.1-34.1950.

7. *A statistical approach to mechanized encoding and searching of literary information.* **H.P.Luhn**. IBM Journal of Research and Development, Vol. 1. pp.309-317,1957

8. *Overview of the first Text REtrieval Conference (TREC-1).* **D.K.Harman.** In Proceedings of the First Text Retrieval Conference (TREC-1), pp. 1–20. NIST Special Publication 500-207, March 1993.

9. **G.Salton and C.Buckley.** *Introduction to Modern Information Retrieval.* New York : McGraw Hill Publishing Company, pp.1-440. 1983.

10. **Ricardo Baeza-Yates and Ribeiro-Neto Bertheir.** *Modern Information Retrieval.* New-York : ACM Press, ISBN o-201-39829-X. pp.1-510. 1999.

11. **C.Manning, P.Raghavan and H.Schutze.** *Introduction to information retrieval.* Cambridge : Cambridge University Press, ISBN 0-521-86571-9. 2008.

12. **W.Frakes and R.Baeza-Yates.** *Information Retrieval:Data stucturies and Algorithms.* New Jersey : USA, pp. 1-237. 1992.

13. *Information Filtering and Information Retrieval : two sides of the same coin.* **N.J.Belkin and W.B.Croft.** Communication of the ACM, Vol. 35(12), pp. 29-38. 1992.

14. *A vector space model for information retrieval.* **G.Salton, A.Wong and S.Yang.** Journal of the American Society for Information Science, November 18(11), Vols. 18(11) pp.613-620, November 1975.

15. **D.A.Grossman and O.Frieder.** *Information Retrieval.* Dordrecht,The Netherlands : Springer, pp.1-330. 2004.

16. **G.Salton.** *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley Publishing Company, pp.1-530. 1989.

17. *On Relevance, Probabilistic Indexing and Information Retrieval.* **M.E.Maron and J.L.Kuhns.** Journal of the Association for Computing Machiner, Vol. 7(3), pp. 216-244. 1960.

18. *The probability ranking principle in IR.* **S.E.Robertson.** Journal of Documentation, Vol. 33(4), pp. 294-304. 1977.

19. **C.D.Manning and H.Schuetze.** *Foundations of Statistical Natural Language Processing.* MIT, pp.1-704. 1999.

20. *The Cranfield tests on index language devices.* **C.W.Cleverdon.**. Aslib Proceedings 19(6):173-194. 1967.

21. *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation.* **D.M.W.Powers.**, Journal of Machine Learning Technologies, pp. 37–63. 2011.

22. **C.Manning, P.Raghavan and H.Shutze.** Evaluation in information Retrieval. Cambridge University Press, pp.151-175. 2009.

23. *Relevance weighting of search terms.* **S.E.Robertson and Jones, K.Spark.** Journal of the American Society for Information Science, 27(3):129–146 , May-June 1976.

24. *Term-weighting approaches in automatic text retrieval.* **G.Salton and C.Buckley.** Information Precessing and Management, Vol. 24, pp. 513-523. 1988.

25. *A statistical interpretation of term specificity.* **K.S.Jones.** Computer Laboratory, University of Cambridge, UK, Journal of Documentation, Vol. 28, pp. 11-21. 1972.

26. *Automatic retrieval with locality information using SMART.* **C.Buckley, G.Salton and J.Allan.**. NIST Special Publication 500-207, In Proceedings of the First Text REtrieval Conference (TREC-1), pp. 59–72, March 1993.

27. *Some simple effective approximations to the 2–poisson model for probabilistic weighted retrieval.* **S.E.Robertson and S.Walker.**. In Proceedings of ACM SIGIR'94, pp. 232–241. 1994

28. *Pivoted document length normalization.* **A.Singhal, C.Buckley and M.Mitra.** New York, August 1996 : s.n. In Proceedings of ACM SIGIR'96, pp. 21–29. Association for Computing Machinery.

29. *A survey of information retrieval and filtering methods.* **C.Faloutsos and D.W.Oard.**University of Maryland, College park.pp.1-24. 1995.

30. *Discourse Level Structure of Abstracts.* **E.D.Liddy.** Boston : American Society for Information Science and Technology Meeting, Proceedings of the 50th ASIS Annual Meeting, pp.138-147. 1987.

31. *A survey of stemming algorithms in information retrieval.* **C.Moral, et al.** Madrid, Spain : IR Information Research, Vol. 19.pp.1-22. 2014.

32. *Machine learning in Automated text Clategorization.* **F.Sebastiane.** Italy : ACM Computing Surveys , Vols. 34(1). pp.1-47. 2002.

33. *Improving text classification by using conceptual and contextual features.* **l.S.Jensen and T.Martinez.** In Proceedings of the Workshop on Text Mining at the 6th ACM SIGKDD. pp.1-2. 2000

34. *Features extraction techniques of unintelligible texts.* **M.Fontaine and S.Matwin.** KDD-2000 Workshop on Text Mining, Boston, 2000, pp. 95-96.

35. *Mining e-mail authorship.* **Olivier de Vel.** Boston . KDD-2000 Workshop on Text Mining.pp.1-7. 2000. https://isis.poly.edu/kulesh/forensics/docs/mining-e-mail-authorship.pdf

36. *Stemming algorithms - a case study for detailed evaluation.* **A.Hull.** Journal of the American Society for Information Science, Journal of the American Society for Information Science, Vol. 47, pp. 70-84. 1996

37. **W.Kraaij and R.Pohlmann.** Porter's stemming algorithm for Dutch. [Online] 1994. https://www.researchgate.net/publication/2596750_Porter's_stemming_algorithm_for_Dutch.

38. *Viewing morphology as an inference process.* **R.Krovetz.** In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY: ACM Press pp. 191-202. 1993

39. *Development of a stemming algorithm.* **J.B.Lovins.** Mechanical Translation and Computational Linguistics, Vol. 11, pp. 22-31. 1968.

40. *How effective is suffixing?* **D.Harmann.** Journal of the American Society for Information Science, Vol. 42, pp. 7-15.1991.

41. **J.Dawson.** *Suffix removal and word conflation. ALLC Bulletin, 2(3), 33-46.* 1974.

42. *An algorithm for suffix stripping.* **M.F.Porter.** Program, Vols. 14(3). pp.130-137, pp. 130-137. 1980.

43. *Another stemmer.* **D Chris Paice.** ACM SIGIR Forum,, Vol. Volume 24, pp. 56-61. 1990.

44. *An evaluation method for stemming algorithms.* **D Chris Paice.** Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. pp.42- 50.1994.

45. **J.Whaley.** *Introduction to Typology: The Unity and Diversity of Language.*SAGE, pp. 1-323. 1992.

46. *ADAM: Analyzer for Dialectal Arabic Morphology.* **Wael Salloum and Nizar Habash.** Journal of King Saud University - Computer and Information Sciences, Vol. 26, pp. 372-378.2014.

47. *An Efficient Mechanism for Stemming and Tagging: The case of greek languge.* **Giorgos Adam, et al.** Proceedings of the 14th international conference on Knowledge-Based and intelligent information and Engineering Systems. Cardiff.Uk . Pp. 389-390. 2010.

48. *Snowball: A language for stemming algorithms.* **M.F.Porter.** 2001. http://snowball.tartarus.org/texts/introduction.html

49. *A survey on feature selection methods.* **G.Chandrashekar and F.Sahin.** Computers & Electrical Engineering, Vol. 40, pp. 16-28. 2014.

50. *A comparative study on feature selection in text categorisation.* **Y.Yang and J.O.Pedersen.** In Proceedings of the 14th International Conference on Machine Learning, ICML-97. *pp.412-420.* 1997. http://courses.ischool.berkeley.edu/i256/f06/papers/yang97comparative.pdf

51. *Feature selection and negative evidence in automated text categorization.* **L.Gallavotti, F.Sebastiani and M.Simi.** Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries. Lisbon: pp.1-2.2000. http://pages.di.unipi.it/turini/MURST/simi2.pdf

52. *Feature selection in text- learning.* **D.Mladenic.** pp.1-6. 1998. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.3219&rep=rep1&type=pdf

53. **K.Fuka and R.Hanka.** *Feature Set Reduction for Document Classification Problems.* Medical Informatics Unit.Univercity of Cambridge.pp. 1-7. http://www.cs.cmu.edu/~mccallum/textbeyond/papers/fuka.pdf

54. *Floating search methods in feature selection.* **P.Pudil, J.Novovi and J.Kittler.** In Pattern Recognition Letters, Vol. 15, pp. 1119-1125. 1994.

55. *Comparision of two learning algorithms for text categorization.* **Lewis, D.D and M.Ringuette.** SDAIR'94 : Proceedings of the third Annual Symposium of document analysis and Information Retrieval. pp. 1-14.1994. https://pdfs.semanticscholar.org/e9fd/1a7ae0322d417ab2d32017e373dd50efc063.pdf

56. *A comparasion of classifiers and document representations for the routing problem.* **H.Shutze, D.A.Hull and J.O.Pedersen.** SIGIR'95 : In 18th Ann Int ACM SIGIR Conferenceon Research and Development in Information Retrieval. pp. 229-237. 1995

57. *A neural network approach to topic spotting.* **E.Wiener, J.O.Pedersen and A.S.Weigend.** SDAIR'95 : In proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval. pp. 1-16. 1995. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.6608&rep=rep1&type=pdf

58. *Toward optimal feature selection.* **D.Koller and M.Sahami.** In proceedings of the Twelfth International Conference on Machine Learning. pp.1-9. 1995.
http://ai.stanford.edu/~koller/Papers/Koller+Sahami:ICML96.pdf

59. *Query expansion using lexical-semantic relations.* **E.M.Voorhees.** Dublin, Ireland : Springer-Verlag New York, In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 61-69. 1994.

60. *Improving the effectiveness of information retrieval with local context analysis.* **J.Xu and W.B.Croft.** ACM Transactions on Information Systems, pp. 79–112.2000,

61. *Using wordnet to disambiguate word senses for text retrieval.* **E.M.Voorhees.** In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Pittsburgh, PA, pp. 171–180.1993.

62. *Indexing by Latent Semantic Analysis.* **T. Susan Dumais*, W. George Furnas and K. Thomas Landauer.** Journal of the American Society for information Science, Vol. 41(6), pp. 391-407.1990,

63. *An Introdaction in Latent Semantic Analyses.* **T.Landauer, P.Foltz and D.Laham.** Discource processes, Vol. 25, pp. 259-284.1998

64. *Real Rectangular matrix.* **J.Callum and R.Willougby.** Lanczos algorithms for large symmetric eignvalue computations., Vol. 1. 1985.

65. *Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge.* **E.Gabrilovich and S.Markovitch**. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06). pp. 1-6. 2006. http://www.cs.technion.ac.il/~gabr/papers/wiki-aaai06.pdf

66. *Computing semantic relatedness using wikipediabased explicit semantic analysis.* **E.Gabrilovich and S.Markovitch.** In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). Morgan Kaufmann Publishers Inc., Hyder. 1606-1611. 2007.
https://isc.uqam.ca/upload/files/Linguistique_cognitive/gabrilovich-markovitch-2007.pdf

67. *Text categorization with knowledge transfer from heterogeneous data sources.* **R.Gupta and L.Ratinov.** In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. AAAI Press, Chicago, IL, pp. 842–847.2008.

68. *Concept-based feature generation and selection for information retrieval.* **O.Egozi, E.Gabrilovich and S.Markovitch.** In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. AAAI Press, Chicago, IL, 1132–1137. 2008.

69.Knowledge-Based Systems for Natural Language Processing. **K. Mahesh, S. Nirenburg.** Computing Research Laboratory . pp. 1-26.1996.
https://pdfs.semanticscholar.org/05b7/6b57c03ebac83ee0ab35e0ff7b577f6cd20f.pdf

70. **S.Sutton, Richard and G.Barton, Nndrew.** *Reinforcement Learning: An Introduction.* Cambridge, Massachusetts : The MIT Press, 2012. http://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html

71. *Applications of machine learning and rule induction.* **P.Langley and H.A.Simon.** Communications of the ACM , Vol. 38(11), pp. pp. 55–64. 1995

72. *Text categorization with support vector machines: Learning with many relevant features.* **T.Joachims.** In Proceedings of ECML-98, pp 137-142. 1998

73. *effective and efficient learning from human decisions in text categorisation and retrieval.* **Yiming Yang.** Dublin : In Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval. pp.13-22. 1994.

74. *Induction of decision trees.* **J.R.Quinlan.** Machine Learning, Vol.1. pages 81–106. 1986.

75. *Machine Learning in Automated Text Categorization.* **F.Sebastiani.** ACM Computing surveys, Vol. 34, pp. 1-47. 2002.

76. An Improved k-Nearest Neighbor Algorithm for Text Categorization . **Baoli Li, Shiwen Yu, Qin Lu** 2003. https://arxiv.org/abs/cs/0306099

77. *Improve text classification accuracy based on classifier fusion methods.* **Behzad Moshiri, Ali Danesh.** 10th International Conference on Information Fusion.2007.

78. *A re-examination of Text categorization Methods.* **Yiming Yang and Xin Liu**.IGIR-99. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 42-49. 1999.

79. **Aas Kjersti and Line Eikvil.** *Text Categorization: A Survey*. pp. 1-38. June, 1999.

80. *Towards the General Theory of Conceptual Systems (a New Point of View).* **V.Chavchanidze.** Kybernetes, Vol. 3, pp. 17-25. 1974

81. *К общей теории концептуальных систем.* **В.В.Чавчанидзе.** Сообщения Института кибернетики АН ГССР. Тбилиси. pp. 1-35. 1973.

82. *К проблеме распознавания образов и об универсальной природе концептуальной интеллектуальной активности.* **В.В.Чавчанидзе.** Материалы коллоквиума по "концептуальному системному анализу естественных и искусственных систем". (Медицина, наука, техника).pp.67-75. 1973.

83. *К началам теории принятия концептуальных решений в системе искусственного интеллекта.* **В.В.Чавчанидзе.** Сообщения АН ГССР, N2.pp. 1-40. 1973.

84. *Concept-Based Information Retrieval using Explicit Semantic Analysis.* **O.Egozi, S.Markovitch and E.Gabrilovich.** ACM Transactions on Information Systems. Vol. 29, No. 2. pp. 1-38.2011.

85. **H.Aronson.** *Georgian: A Reading Grammar.* University of Chicago, pp.1-538. 1990.

86. *Concept pattern formation in semantic search problem.* **M.Khachidze, et al.** GESJ. Georgian Electronic Scientific Journal, Computer Science and Telecommunications, pp. 13-20. 2014.

87. *Hospital Automation via Computer Time-Sharing, Computers and Biomedical Research.* **J.J.Baruch, R.W.Stacey and B.D.Waxman.** Academic Press New York, pp. 291-312. 1965.

88.**F.Morris and al.** *The History of Medical Informatics in the United States.* Springer, pp.1-755. 2015.

89. *An Evaluation of Statistical Approaches to Text Categorization.* **Y.Yang.** Journal of Information Retrieval, pp. 69-90.1999.

90. *Automatic construction of the rule-based ICD-9-CM coding systems.* **R.Farkas and G.Szarvas.** BMC Bioinformatics, pp. 3-10. 2008.

91. *Using natural language processing to extract mammographic findings.* **H.Gao, et al.** Journal of Biomedical Informatics, pp. 77-84. 2015.

92. *Portable automatic text classification for adverse drug reaction detection via multi-corpus training.* **A.Sarker and G.Gonzalez.** Journal of Biomedical Informatics, pp. 196-207.2015.

93. *A systematic comparison of feature space effects on disease classifier performance for phenotype identification of five diseases.* **C.Kotfila and O.Uzuner.** Biomed Inform.pp.92-102. 2015

94. *What can natural language processing do for clinical decision support?* **D.Demner-Fushman, WW.Chapman and CJ.McDonald.** Biomed Inform, (42), pp. 760-772.2009

95. *Natural language processing:an introduction.* **PM.Nadkarni, L.Ohno-Machado and WW.Chapman.** J Am Med Inform Assoc, pp. 544-551.2011

96. **C.Friedman and N.Elhadad.** Natural language processing in health care and biomedicine. *Biomedical informatics.* pp. 255-84.Springer, 2013,

97. *A review of feature selection techniques in bioinformatics.* **Y.Saeys.** Bioinformatics, pp. 2507-2517.2007

98. *Use of remind artificial intelligence software for rapid assessment of adherence to disease specific management Guidelines in Acute Coronary Syndromes.* **A.F.Sonel, et al.** AHRQ. Center For Health Equity Research and Promotion, 2006. www.cs.cmu.edu/~stefann/papers/AHRQ_06_poster.ppt

99. *Using Machine Learning Classifiers to Assist Healthcare-Related Decisions: Classification of Electronic Patient Records.* **JT.Pollettin, et al.** 2012, Journal of Medical Systems, Vol. 36(6), pp. 3861-3874.

100. *Text classification and classifiers.* **V.Korde.** 2012, International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 3, pp. 85-94.

101. *Accurate Cancer Classification using Expressions of Very few Genes.* **N.Revathy and R.Amalraj.** International Journal of Computer Applications, Vol. 14, pp. 19-22. 2011.

102. *N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit.* **BJ.Marafino, et al.** Med Inform Assoc. Vol. 21(5), pp. 871-875. 2014.

103. *Gene Selection for Cancer Classification using Support Vector Machine.* **I.Guyon, et al.** Machine Learning, Vol. 46, pp. 389-422. 2002.

104. *Negation scope delimitation in clinical text using three approaches: NegEx, PyConTextNLP and SynNeg.* **H.Tanushi, et al.** Sweden : Linköping University Electronic Press, Linköping; Proc 19th NODALIDA, NEALT. pp. 387-397. 2013.

105. *Detecting negation of medical problem in French clinical notes.* **G.Luo, et al.** New York. Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. pp. 697-702. 2012.

106. Health Insurance Portability and Accountability Act of 1996.Public Low https://aspe.hhs.gov/report/health-insurance-portability-and-accountability-act-1996

107. Evaluating the state-of-the-art in automatic de-identification. **O.Uzuner, Y.Luo and P.Szolovits.** *Med Inform Assoc 2,* 14(5), pp. 550-63.2007.

108. *Automatic de-identification of textual documents in the electronic health records: a review of recent research.* **SM.Meystre, et al.** BMC Med Res Methodol Vol. 10(70).pp.1-16. 2010.

109. *Automatic De-Identification of French Clinical Records: Comparison of Rule-Based and Machine-Learning Approaches.* **Cyril Grouin and Zweigenbaum, Pierre.** IMIA and IOS Press, MEDINFO . pp. 476-481. 2013

110. *The unified medical language system.* **D.Lindberg, B.Humphreys და A.McCray.** Inf Med. vol. 1, pp. 281-291.1993.

111. *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.* **AR.Aronson.** AMIA Annual Symposium. pp. 17-21. 2001.

112. **C.Manning, P.Raghavan and H.Shutze.** *Introduction to Iinformatiion Retrieval.* Cambridge University press, pp. 1-581.2008.

113. **D.A.Hull and G.Grefenstette.** A Detailed Analysis of English Stemming Algorithms. *Rank Xerox Research Centre.* 31, pp. 1-16. 1996.

114. **M.F.Porter.** Stemming algorithms for various European languages. [Online] 2004. http://snowball.tartarus.org/english/stemmer.html.

115. *Georgian Language Based Document Classification Method Development.* **M.Khachidze, M.Vardanidze and G.Dzamashvili.** The Fourth Annual Conference in Exact and Natural Sciences. Tbilisi.2016. http://conference.ens-2016.tsu.ge/en/lecture/view/514

116. **[Online] http://classifications.moh.gov.ge/Classifications/Pages/ViewICD10.asp.**

117**. H.S.Heaps.** Information Retrieval: Computational and Theoretical Aspects**.** Orlando.USA : Academic Press, pp. 1-344. 1978.

118**.** *A comparative study on feature selection in text categorization.* **Y.Yang and J.O.Pedersen. San Francisco.** Proceedings of ICML-97, 14th International Conference on Machine Learning. US : Morgan Kaufmann Publishers, pp. 412-420.1997.

119**.***An Introduction to Kernel-Based Learning Algorithms*.**Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf.** IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 12, NO. 2,pp. 1-21. MARCH 2001

120. *The Method of Concept Formation for Semantic Search.* **M.khachidze, et al.** proceedings : International Conference on APPLICATION of INFORMATION and COMMUNICATION TECHNOLOGIES. Baku, Azerbaijan. pp.132-137. 2013.

121. *Concept Pattern Based Text Classification System Development for Georgian Text Based Information Retrieval.* **M.Khachidze at al**. Baltic J. Modern Computing, vol 3, pp. 307-317. 2015

122**.** *Natural Language Processing based Instrument For Classification of Free Text medical Record.* **M.Khachidze, M.Tsintsadze and M.Archuadze.** BioMed Research international. Vol.2016(2016).pp.1-10.

123**.** *Complex System State Generalized Presentation Based on Concepts.* **M.Khachidze, et al**. 8th International Conference on Application of Information and Communication Technologies - AICT2014**.** Kazakhstan, Astana . pp. 559-569. 2014.

124. *Dental Self-diagnostic Information System Based on the Natural Language Processing.* **M.Khachidze, et al.** eRA-11 International Scientific Conference. Pireus, Greece , 2016. http://era.teipir.gr/conference-info

125. *ანალიტიკური-ევრისტიკული კონცეპტები სემანტიკური ძებნის ამოცანებში.* **მ.ხაჩიძე, მ.არჩუაძე და გ.ბესიაშვილი.** VI საერთაშორისო სამეცნიერო პრაქტიკული კონფერენცია "ინტერნეტი და საზოგადოება". გვ.11-14.ქუთაისი . 2013.

126. *Short Text Classification Application in Automated Workflow Management Systems.* **M.Khachidze, et al**. eRA-11 International Scientific Conference. Greece, Pireus . 2016. http://era.teipir.gr/conference-info

დანართი

N	ψ_0	ψ_1	ψ_2	ψ_3	...	ψ_m
1	1	1	1	1		1
2	2	2	2	2		$\overline{2}$
				$\vdots$		
	.	.	.	$\overline{2^{m-3}}$		
			.	$\overline{2^{m-3}+1}$		.
			.	$\overline{2^{m-3}+2}$		
				$\vdots$		
	.	.	2^{m-2}	$\overline{2^{m-2}}$		
			$\overline{2^{m-2}+1}$	$2^{m-2}+1$		.
			$\overline{2^{m-2}+2}$	$2^{m-2}+2$		
				$\vdots$		
	.	.	.	$\overline{2^{m-2}+2^{m-3}}$		
			.	$\overline{2^{m-2}+2^{m-3}+1}$		.
			.	$\overline{2^{m-2}+2^{m-3}+2}$		
				$\vdots$		
n	n	2^{m-1}	$\overline{2^{m-1}}$	$\overline{2^{m-1}}$		
n+1	$\overline{1}$	$\overline{2^{m-1}+1}$	$2^{m-1}+1$	$2^{m-1}+1$		.
n+2	$\overline{2}$	$\overline{2^{m-1}+2}$	$2^{m-1}+2$	$2^{m-1}+2$		
				$\vdots$		
			.	$\overline{2^{m-1}+2^{m-3}}$		.
	.	.	.	$\overline{2^{m-1}+2^{m-3}+1}$		
			.	$\overline{2^{m-1}+2^{m-3}+2}$		
				$\vdots$		
			$2^{m-1}+2^{m-2}$	$\overline{2^{m-1}+2^{m-2}}$		.
	.	.	$\overline{2^{m-1}+2^{m-2}+1}$	$2^{m-1}+2^{m-2}+1$		
			$\overline{2^{m-1}+2^{m-2}+2}$	$2^{m-1}+2^{m-2}+2$		
				$\vdots$		
			.	$\overline{2^{m-1}+2^{m-2}+2^{m-3}}$		.
	.	.	.	$\overline{2^{m-1}+2^{m-2}+2^{m-3}+1}$		
			.	$\overline{2^{m-1}+2^{m-2}+2^{m-3}+2}$		
				$\vdots$		
$2n$	$\overline{n}$	$\overline{2^m}$	$\overline{2^m}$	$\overline{2^m}$		$\overline{2^m}$

ცხრილი 6

სფერო	დაბრუნებული	რელევანტური (tp)	არა რელევანტური (fp)	fn	tn	არ დაბრუნებული	recall	precision	F measure	Accuracy	ERR
ეკონომიკა	**132**	**112**	**20**	38	747	**38**	**0,747**	**0,848**	**0,794**	0,937	0,063
პოლიტიკა	**159**	**117**	**42**	33	720	**33**	**0,780**	**0,736**	**0,757**	0,918	0,082
სპორტი	**173**	**117**	**56**	33	706	**33**	**0,780**	**0,676**	**0,724**	0,902	0,098
ისტორია	**180**	**118**	**62**	32	699	**32**	**0,787**	**0,656**	**0,715**	0,897	0,103
მედიცინა	**183**	**118**	**65**	32	696	**32**	**0,787**	**0,645**	**0,709**	0,894	0,106
იურისპრუდენცია	**197**	**124**	**73**	26	682	**26**	**0,827**	**0,629**	**0,715**	0,891	0,109

ცხრილი 7 შედეგები KNN ალგორითმისათვის

სფერო	დაბრუნებული	რელევანტური (tp)	არა რელევანტური (fp)	fn	tn	არ დაბრუნებული	recall	precision	F measure	Accuracy	ERR
ეკონომიკა	**125**	**104**	**21**	46	754	**46**	**0,693**	**0,832**	**0,756**	0,928	0,072
პოლიტიკა	**137**	**112**	**25**	38	742	**38**	**0,747**	**0,818**	**0,780**	0,931	0,069
სპორტი	**158**	**115**	**43**	35	721	**35**	**0,767**	**0,728**	**0,747**	0,915	0,085
ისტორია	**173**	**124**	**49**	26	706	**26**	**0,827**	**0,717**	**0,768**	0,917	0,083
მედიცინა	**172**	**123**	**49**	27	707	**27**	**0,820**	**0,715**	**0,764**	0,916	0,084
იურისპრუდენცია	**187**	**127**	**60**	23	692	**23**	**0,847**	**0,679**	**0,754**	0,908	0,092

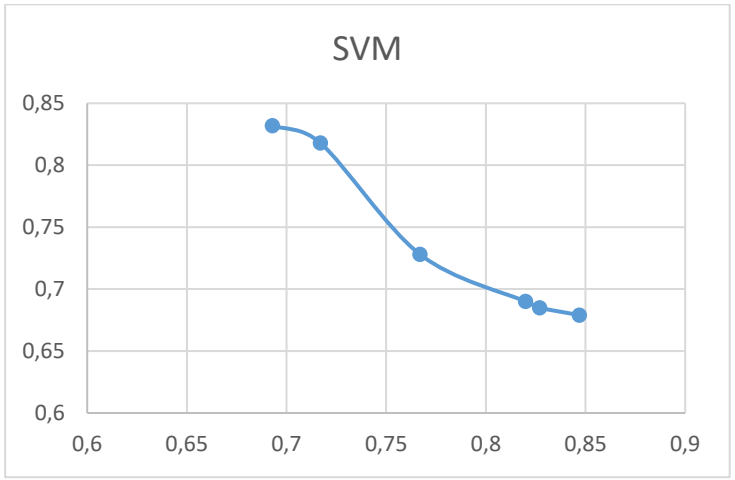
ცხრილი 8 შედეგები SVM ალგორითმისათვის

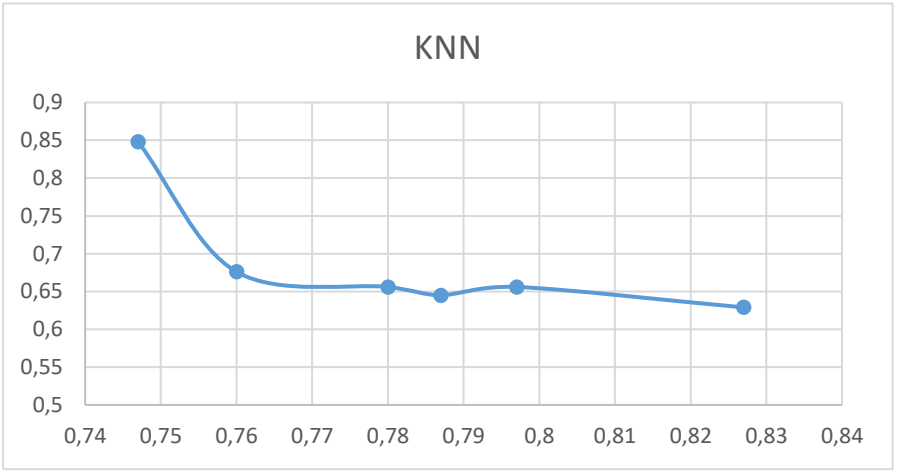
სფერო	დაბრუნებული	რელევანტური (tp)	არა რელევანტური (fp)	fn	tn	არ დაბრუნებული	recall	precision	F measure	Accuracy	ERR
ეკონომიკა	**120**	**75**	**45**	75	759	**75**	**0,500**	**0,625**	**0,556**	0,874	0,126
პოლიტიკა	**159**	**99**	**60**	51	720	**51**	**0,660**	**0,623**	**0,641**	0,881	0,119
სპორტი	**162**	**100**	**62**	50	717	**50**	**0,667**	**0,617**	**0,641**	0,879	0,121
ისტორია	**165**	**100**	**65**	50	714	**50**	**0,667**	**0,606**	**0,635**	0,876	0,124
მედიცინა	**209**	**117**	**92**	33	670	**33**	**0,780**	**0,560**	**0,652**	0,863	0,137
იურისპრუდენცია	**218**	**120**	**98**	30	661	**30**	**0,800**	**0,550**	**0,652**	0,859	0,141

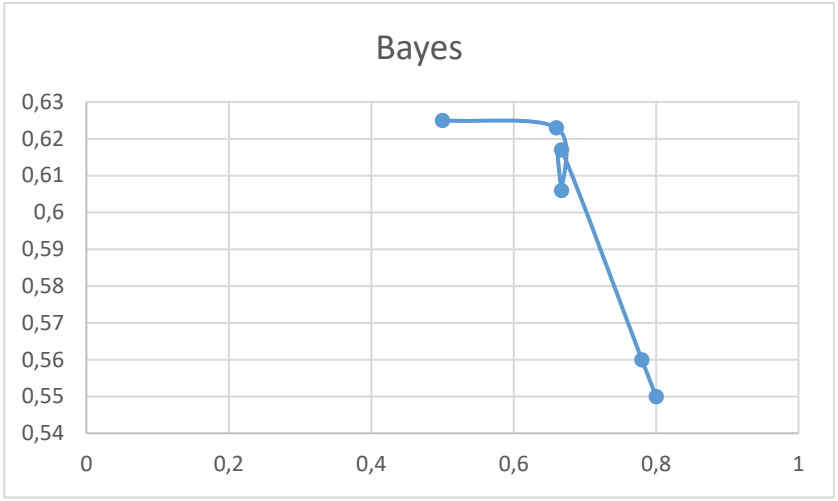
ცხრილი 9 შედეგები Bayes ალგორითმისათვის

სფერო	Recall			Precision			F Measure			Accuracy			ERR		
	KNN	SVM	Bayes	KNN	SVM	Bayes	KNN	SVM	Bayes	KNN	SVM	Bayes	KNN	SVM	Bayes
ეკონომიკა	**0,747**	0,693	0,500	0,848	0,832	0,625	0,937	0,756	0,556	0,937	0,928	0,874	0,072	0,072	0,126
პოლიტიკა	**0,780**	0,747	0,660	0,736	0,818	0,623	0,918	0,780	0,641	0,918	0,931	0,881	0,069	0,069	0,119
სპორტი	**0,780**	0,767	0,667	0,676	0,728	0,617	0,902	0,747	0,641	0,902	0,915	0,879	0,085	0,085	0,121
ისტორია	**0,787**	0,827	0,667	0,656	0,717	0,606	0,897	0,768	0,635	0,897	0,917	0,876	0,083	0,083	0,124
მედიცინა	**0,787**	0,820	0,780	0,645	0,715	0,560	0,894	0,764	0,652	0,894	0,916	0,863	0,084	0,084	0,137
იურისპრუდენცია	**0,827**	0,847	0,800	0,629	0,679	0,550	0,891	0,754	0,652	0,891	0,908	0,859	0,092	0,092	0,141

ცხრილი 10 შედარებითი ანალიზი (KNN, SVM,Bayes)







სურათი 3. precision-recall მრუდი სამი ალგორითმისათვის

მონაცემები		სულ	დასასწავლი	სატესტო
ულტრასონოგრაფია		**12864**	**7720**	**5144**
	ღვიძლი		1360	784
	სანაღვლე სისტემა,		1227	957
	თირკმელების და შარდ-სასქესო სისტემა		896	536
	გინეკოლოგიური		1372	942
	ფარისებრი ჯირკვლი		1101	851
	სარძევე ჯირკვლის		771	531
	სისხლძარღვების		993	543
რენტგენი		**10523**	**5262**	**5262**
	გულმკერდის,		849	849
	მუცლის ღრუს,		1057	1057
	ხერხემლის,		946	946
	კიდურების,		612	612
	საყლაპავის და კუჭის,		1224	1224
	მსხვილი და წვრილი ნაწლავების		574	574
ენდოსკოპია		**1468**	**734**	**734**
სულ	**დოკუმენტების რაოდენობა (ულტრასონოგრაფია, რენტგენი, ენდოსკოპია)**	**24855**	**13716**	**11140**

ცხრილი 11 სამედიცინო ჩანაწერების დოკუმენტების გაზა

I დონე	I I დონე	SVM					KNN				
		recall	precision	F meas	Accuracy	ERR	recall	precision	F measure	Accuracy	ERR
ულტრასონოგრაფია		**0,905**	**0,803**	**0,851**	**0,853**	**0,147**	**0,833**	**0,814**	**0,824**	**0,835**	**0,165**
	ღვიძლი	0,749	0,528	0,619	0,860	0,140	0,719	0,510	0,597	0,852	0,148
	სანაღვლე სისტემა,	0,572	0,436	0,495	0,783	0,217	0,555	0,396	0,462	0,760	0,240
	თირკმელების და შარდ-სასქესო სისტემა	0,942	0,813	0,873	0,971	0,029	0,784	0,591	0,674	0,921	0,079
	გინეკოლოგიური	0,908	0,871	0,889	0,958	0,042	0,699	0,631	0,663	0,870	0,130
	ფარისებრი ჯირკვლი	0,825	0,794	0,809	0,936	0,064	0,807	0,712	0,757	0,914	0,086
	სარძევე ჯირკვლის	0,904	0,769	0,831	0,962	0,038	0,870	0,589	0,703	0,924	0,076
	სისხლძარღვების	0,866	0,721	0,787	0,950	0,050	0,871	0,696	0,774	0,946	0,054
რენტგენი		**0,894**	**0,813**	**0,852**	**0,853**	**0,147**	**0,814**	**0,793**	**0,804**	**0,812**	**0,188**
	გულმკერდის,	0,888	0,852	0,870	0,957	0,043	0,775	0,838	0,805	0,940	0,060
	მუცლის ღრუს,	0,855	0,762	0,806	0,917	0,083	0,760	0,640	0,695	0,866	0,134
	ხერხემლის,	0,889	0,774	0,827	0,933	0,067	0,783	0,628	0,697	0,878	0,122
	კიდურების,	0,990	0,774	0,869	0,965	0,035	0,832	0,567	0,675	0,907	0,093
	საყლაპავის და კუჭის,	0,905	0,881	0,893	0,949	0,051	0,688	0,638	0,662	0,837	0,163
	მსხვილი და წვრილი ნაწლავების	0,911	0,944	0,927	0,984	0,016	0,841	0,657	0,738	0,935	0,065
ენდოსკოპია		**0,965**	**0,804**	**0,877**	**0,981**	**0,019**	**0,926**	**0,749**	**0,828**	**0,974**	**0,026**

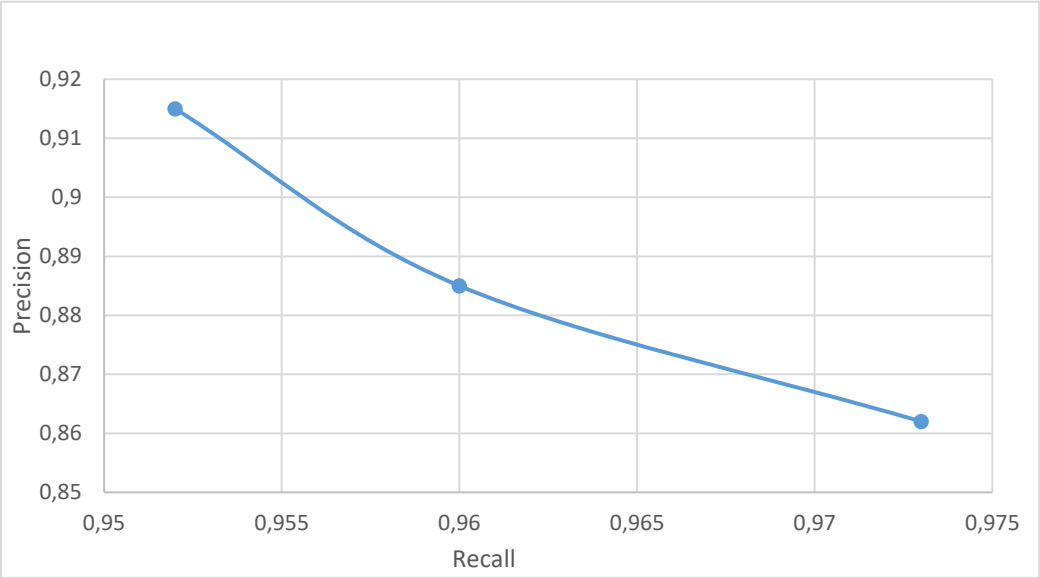
ცხრილი 12 SVM და KNN შედარებითი ანალიზი (თვისებათა ამოკრების კლასიკური მეთოდი)

გამოთვლები							tp/(tp+fn)	tp/(tp+fp)	2*P*R/ (R+P))	(tp+tn)/ (tp+fp+fn+tn)	(fp+fn)/n
ტესტირების კლასები	გასატესტი დოკუმენტების რაოდენობა	დაბრუნებული დოკუმენტები	True positive (tp)	False positive (fp)	False negative (fn)	True negative (tn)	Recall	Precision	F measure	Accuracy	ERR
ულტრასონოგრაფია	11140	5798	4657	1141	487	4855	0,905	0,803	0,851	0,854	0,146
რენტგენი		5784	4703	1081	559	4797	0,894	0,813	0,852	0,853	0,147
ენდოსკოპია		881	708	173	26	9698	0,965	0,804	0,877	0,981	0,019

ცხრილი 13 კლასიფიკაციის შედეგები (თვისებათა შერჩევის კლასიკური მეთოდი/SVM)

გამოთვლები							tp/(tp+fn)	tp/(tp+fp)	2*P*R/ (R+P)	(tp+tn)/ (tp+fp+fn+tn)	(fp+fn)/ n
ტესტირების კლასები	ტესტის დოკუმენტების რაოდენობა	დაბრუნებული დოკუმენტები	True positive (tp)	False positive (fp)	False negative (fn)	True negative (tn)	Recall	Precision	F measure	Accuracy	ERR
ულტრასონოგრაფია	11140	5802	5004	798	140	5198	0,973	0,862	0,914	0,916	0,084
რენტგენი		5478	5010	468	252	5410	0,952	0,915	0,933	0,935	0,065
ენდოსკოპია		872	705	167	29	9701	0,960	0,885	0,878	0,982	0,018

ცხრილი 14 კლასიფიკაციის შედეგები (თვისებათა შერჩევის „შეკუმშული" მეთოდი/SVM)



სურათი 4. precision-recall მრუდი (თვისებათა შერჩევის „შეკუმშული“ მეთოდი/SVM/I დონე)

ქვეკლასი/დონე 2										
ულტრასონოგრაფია (დოკუმენტების რაოდენობა)	**შეკუმშული მეთოდი**					**კლასიკური მეთოდი**				
	R	**P**	**F**	**Acc**	**ERR**	**R**	**P**	**F**	**Acc**	**ERR**
ღვიძლი (784)	0,610	0,398	0,481	0,800	0,200	0,749	0,528	0,619	0,860	0,140
სანაღვლე სისტემა(957)	0,416	0,360	0,386	0,753	0,247	0,572	0,436	0,495	0,783	0,217
ტირკმელების და შარდსასქესო სისტემა (536)	0,806	0,635	0,711	0,932	0,068	0,942	0,813	0,873	0,971	0,029
გინეკოლოგია (942)	0,781	0,738	0,759	0,909	0,091	0,908	0,871	0,889	0,958	0,042
ფარისებრი ჯირკვალი (851)	0,805	0,752	0,778	0,924	0,076	0,825	0,794	0,809	0,936	0,064
სარძევე ჯირკვალი (531)	0,793	0,594	0,679	0,923	0,077	0,904	0,769	0,831	0,962	0,038
სისხლძარღვები (543)	0,888	0,666	0,761	0,941	0,059	0,866	0,721	0,787	0,950	0,050
რენტგენი (დოკუმენტების რაოდენობა)	**შეკუმშული მეთოდი**					**კლასიკური მეთოდი**				
	R	**P**	**F**	**Acc**	**ERR**	**R**	**P**	**F**	**Acc**	**ERR**
გულმკერდი (849)	0,802	0,629	0,705	0,892	0,108	0,888	0,852	0,870	0,957	0,043
მუცლის ღრუ (1057)	0,729	0,634	0,678	0,861	0,139	0,855	0,762	0,806	0,917	0,083
ხერხემალი (946)	0,785	0,521	0,626	0,831	0,169	0,889	0,774	0,827	0,933	0,067
კიდურები (612)	0,822	0,403	0,541	0,838	0,162	0,990	0,774	0,869	0,965	0,035
საყლაპავის და კუჭის (1224)	0,714	0,679	0,696	0,855	0,145	0,905	0,881	0,893	0,949	0,051
მსხვილი და წვრილი ნაწლავის (574)	0,796	0,603	0,686	0,921	0,079	0,911	0,944	0,927	0,984	0,016

ცხრილი 15 კლასიფიკაციის შედეგები ქვედონეების მიხედვით (კლასიკური და „შეკუმშული“ მეთოდების გამოყენებით/ SVM)