TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Tarmo Põldme 142057

JURIIDILISE INFORMATSIOONI TSENTRAALSUSMÕÕTUDEL JA TEKSTILISEL RELEVANTSUSEL PÕHINEV OTSINGUMOOTOR

Magistritöö

Juhendaja: Innar Liiv
Ph. D.

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Tarmo Põldme

07.05.2017

Annotatsioon

Käesolev magistritöö jaguneb kahe suure eesmärgi vahel. Töö teoreetilises osas uuritakse, kas võrgustike teaduse teoreetilisi aluseid saab rakendada juriidilise informatsiooni struktuuri moodustavatele õigusaktidele relevantsuskaalude leidmiseks. Töö praktilise osa eesmärk on teoreetilises osas saadud teadmiste põhjal realiseerida efektiivsem otsingusüsteem.

Töö teoreetilise osa aluseks on etteantud algoritmi täiendamine, mis tugineb ühe seadusandliku akti sisemise struktuuri uurimisele ja sektsioonidevaheliste viidete kaardistamisele [1]. Käesolev töö laiendab algoritmi metoodika kogu seadusandlusele uurides viiteid seaduste vahel. Tsentraalsuskaalude leidmiseks kasutatakse vaheloleku mõõdikut kombineerituna konformismianalüüsiga. Teoreetilise osa tulemiks on kaardistatud seaduste võrgustik koos aktidele määratud tsentraalsuskaaludega.

Töö praktilises osas analüüsitakse otsingusüsteemide põhimõttelisi puudusi ning kaardistatakse kaasaegse otsingumootori nõuded. Nõuetele tuginedes, valitakse välja sobilik otsingutehnoloogia ning realiseeritakse prototüüplahendus, mis kasutab töö teoreetilises osas leitud tsentraalsuskaalusid. Praktilise osa tulemiks on otsingu prototüüp, mis arvestab tsentraalsusel põhinevat relevantsust koos valitud tehnoloogia tekstilise relevantsusega. Töö sisendiks on Riigi Teatajast kättesaadavad baasaktid, kuid tulemuse saab laiendada mistahes riigi seadusandlusele või struktuuri moodustavale informatsioonile.

Magistritöö on kirjutatud eesti keeles ning sisaldab teksti 68 leheküljel, 47 peatükki, 15 joonist, 10 tabelit.

Abstract

Legal information search engine based on centrality measures and textual relevance

The present master thesis is divided between two primary goals. The aim of the theoretical part is to research whether network science theoretical bases could be used in finding relevance weights for legislative acts of juridical information. The objective of the practical part is to build more efficient search system by using knowledge, gained from the theoretical part.

The base of the thesis is to complement given algorithm, which is founded on analyze of the inner-structure of the legislative act, constructed by references between the sections [1]. Current thesis expands the methodology to entire legislation by analyzing references between legislation acts. *Betweenness* centrality measure from the area of social networks is combined with matrix *conformity analyzes* to find numerical importance value for the vertices of the legislation graph. The result of the theoretical part is mapped legislation network with centrality weights assigned to each legislation act inside the network.

In the practical part of the thesis, principal deficiencies of search systems are analyzed and requirements of modern search engine are determined. Based on the requirements, appropriate search technology is chosen and search prototype is being built, which uses centrality weights found in the theoretical part of the thesis. The result of the practical part is search prototype, which considers relevance, based on centrality weights along with textual relevance of chosen technology. Prototype is validated against two different target groups – lawyers and ordinary citizens. The aim of the validation is to get qualitative feedback from experts and possible end users.

The input of the thesis relies on base legislation acts of Riigi Teataja, but the results are extendable to any other legislation as well as to information, which forms structure.

The thesis is in Estonian and contains 68 pages of text, 47 chapters, 15 figures, 10 tables.

Lühendite ja mõistete sõnastik

API	*Application Programming Interface,* reeglistik suhtlemaks olemasoleva tarkvaraga
Betweenness	Tsentraalsuse mõõdik vahelolek
Closeness	Tsentraalsuse mõõdik lähedus kõigile
Conformity analyses	Konformismianalüüs, andmete ümberkorrastamise metoodika [2] sagedusteisenduse abil tuletatud konformismiskaala põhjal
CSV	*Comma Separated Value,* faili formaat, talletamaks tabuleeritud andmeid
Degree	Tsentraalsuse mõõdik suhete arv
HTTP	*Hyper Text Transfer Protocol,* protokoll info edastamiseks arvutivõrkudes
JavaScript	Objektorienteeritud programmeerimiskeel, mida kasutatakse peamiselt veebilehtede programmeerimisel
JSON	*Javascript Object Notation,* inimkeeles loetav andmeobjektide formaat
Kasutaja	Otsingusüsteemi kasutav inimene
KB	*Kilobyte*, mälumahu mõõtühik, 1024 baiti
Lemmatiseerimine	Sõnade algvormile taandamine, kasutades vormi moodustamist (morfoloogia) ja sõnaraamatut
Otsingumootor	Veebipõhine klient-server rakendus, mis võimaldab sisestada otsingu parameetreid ja kasutaja käskluse peale teostada serveris olevatest andmetest otsingu ning tagastada need kliendile
PHP	*Hypertext Preprocessor*, üldotstarbeline programmeerimiskeel, mis leiab eelkõige kasutust veebiarenduses
Rakendus	Tarkvaraline toode, mis on kasutatav sisevõrgus või üle avaliku interneti
REST	*Representational state transfer,* arhitektuuriline lahendus, pakkumaks koostalitlusvõimet arvutisüsteemide vahel üle arvutivõrgu või interneti
SaaS	*Software (Search) as a Service,* (otsingu)tarkvara, mida pakutakse tsentraalse teenusena enamasti üle interneti, tellimuse põhiselt

Seaduste võrgustik	Seaduste ning neid siduvate viidete kaardistus
SQL	*Structured Query Language*, andmebaasi suhtluse keel
Stemmimine	Sõnade algvormile taandamine ilma sõna vormide moodustamiseta (morfoloogiata)
Täisteksti otsing	Tehnoloogia, otsimaks digitaalset dokumenti või dokumentide kogumit täisteksti andmebaasist, kus iga otsitavat sõna võrreldakse eraldiseisvalt
Võrgustike teadus	Akadeemiline uurimisvaldkond, mis tegeleb keeruliste võrgustike uurimisega, nagu telekommunikatsiooni võrgud, arvutivõrgud, bioloogilised võrgud, sotsiaalsed võrgud jt
XML	*Extended Markup Language*, standardne ja üldotstarbeline märgistuskeel, mille eesmärgiks on struktureeritud info jagamine infosüsteemide vahel

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Juriidilises informatsioonis orienteerumine on keeruline ja aeganõudev tegevus, mis enamasti on jõukohane ainult vastava erialase taustaga inimestele. Kuigi infotehnoloogilised lahendused erinevates eluvaldkondades on plahvatuslikult arenenud, on juriidika sellest paljus puutumata jäänud. Suur osa tööst seadusandlikest tekstidest info leidmisel tehakse manuaalselt või algelisi otsingusüsteeme kasutades. Põhjuseid on mitmeid. Esiteks on vastava eriala inimesed harjunud nii töötama ja see kuulub juristi traditsioonilise töö juurde. Teiseks on juriidilise info sisu keerulise struktuuriga ning infokildude vahelised seosed süsteemides kaardistamata. Efektiivsemate otsingusüsteemide jaoks oleks vaja arusaamist, kuidas seadusandlik info omavahel seotud on ning kas seostele saaks tähtsuse järgi teatava mõõdiku anda. See võib olla üheks põhjuseks, mis takistab infotehnoloogiliste otsingusüsteemide arengut juriidika valdkonnas.

Samas on nõudlus juriidilise info tarbimise järele ühiskonnas, sh tavakodanike seas, oluliselt kasvanud. Tihti on juriidilist nõu vaja lihtsate igapäevaste tegevuste käigus nagu kinnisvara ost, ettevõtlusega seotud toimingud, suhtlus tööandjaga jne. Kuigi internetiajastu ja avaliku sektori areng on seadusandliku info ligipääsetavaks teinud väga lihtsalt ja kõigile, siis paraku ei ole tavainimesel sellest kasu, kui vajamineva info leidmine vajab juriidilisi ekspertteadmisi. Oluliselt suurem tõenäosus on otsitava informatsioonini jõuda, kui infosüsteemid omaksid teatavat intelligentsust ning oskaksid otsingutulemustes eelistada olulisemat vähemolulisemast. Juriidilise info leidmise lihtsustamine ei puuduta seega ainult vastava valdkonna spetsialiste, vaid ühiskonda laiemalt ja sügavuti [3].

1.1 Töö eesmärgid

Töö eesmärgid jagunevad kaheks osaks, mis kokku moodustavad ühe terviku.

Esmane eesmärk on analüüsida uuritava algoritmi [1] abil seadusandliku info struktuuri, kaardistada see ning selle põhjal leida struktuuri moodustavatele seadustele relevantsuse

kaalud, mille põhjal on võimalik otsingutulemusi järjestada. Algoritmi uurimise käigus selgub, kas mingi osa sellest vajab täiendamist või ümberlükkamist, kuid eelistatud oleks relevantsuse kaalu leidmine tsentraalsusmõõdikuid kasutades.

Töö teine eesmärk on kaardistada kaasaegse otsingumootori nõuded, valida nõuetele vastav tehnoloogia ning realiseerida prototüüplahendus, mis tugineb töö uurimuslikus osas leitud relevantsuse kaaludele.

Töö üldisem eesmärk on saada kvalitatiivset tagasisidet töö uurimuslikust poolest, et kasutada seda efektiivsema tarkvaralise tulemi realiseerimisel.

Töö praktilise tulemuse huvigruppide hulka saab eelkõige lugeda juriidikat mitte tundvad kodanikud, kelle jaoks hetkel on juriidilise info leidmine kõige keerukam. Samas ei välistata ka juristide huvi, kes saavad oma tööd mõnevõrra mugavamaks muuta. Uurimusliku ning prototüüpimise poole peamiseks sihtgrupiks on tarkvarainsenerid, kelle kasutada on analüüsitud ja kaardistatud tulemused uute infosüsteemide projekteerimisel ning ehitamisel.

1.2 Ülevaade tööst

Käesolev magistritöö käsitleb võrgustike teaduse ja otsingumootorite temaatikat, tuginedes juriidilisele informatsioonile ning selles leiduvatele seaduspärasustele. Töö esimeses osas antakse ülevaade uuritavast algoritmist ja otsingutehnoloogiatest. Töö teises osas analüüsitakse seaduste vahelisi viiteid ja neist moodustuvat andmete võrgustikku. Kasutades ära võrgustikku iseloomustavaid parameetreid, püütakse täiendada metoodika osas kirjeldatud algoritmi, et määrata igale seadusele relevantsuse kaal. Kolmandas osas analüüsitakse kaasaegse juriidilise informatsiooni otsingumootori aluseid ja kaardistatakse nõuded uue põlvkonna otsingumootori tarbeks. Töö viimases osas viiakse kokku algoritmi uurimise tulemused ning kaardistatud otsingumootori nõuded ja prototüübitakse lihtsustatud näidislahendus koos näidispäringutega.

Töö sisendiks on Riigi Teataja infosüsteemist vabalt kättesaadavad õigusaktid, kuid töö tulemus on laiendatav mistahes riigi seadusandlusele või informatsioonile, kus andmed moodustavad omavahel viidete kaudu võrgustiku.

2 Metoodika ja otsingutehnoloogiad

Töö sissejuhatavas osas, tehakse ülevaade uurimuslikus pooles kasutatavast algoritmist ning otsingutehnoloogiatest, mille rakendamist kaaluda prototüüpimisel. Metoodilise peatüki eesmärk ei ole laskuda detailidesse, vaid anda lugejale üldine ülevaade töös käsitletavast temaatikast ja realisatsiooni vahenditest.

Algoritmi ülevaatega tutvustatakse töö teoreetilist poolt ja selgitatakse selle kasutamist käesoleva töö uurimisobjekti seisukohast lähtuvalt. Otsingutehnoloogiate kirjeldus aitab saada ettekujutuse prototüüpimise osas plaanitavast lahendusest ja põhimõttelistest tehnoloogilistest valikutest uue otsingu ehitamisel. Iga tehnoloogia puhul tehakse lühiülevaade ja antakse esmane hinnang kasutamise sobivuse kohta prototüübi ehitamisel.

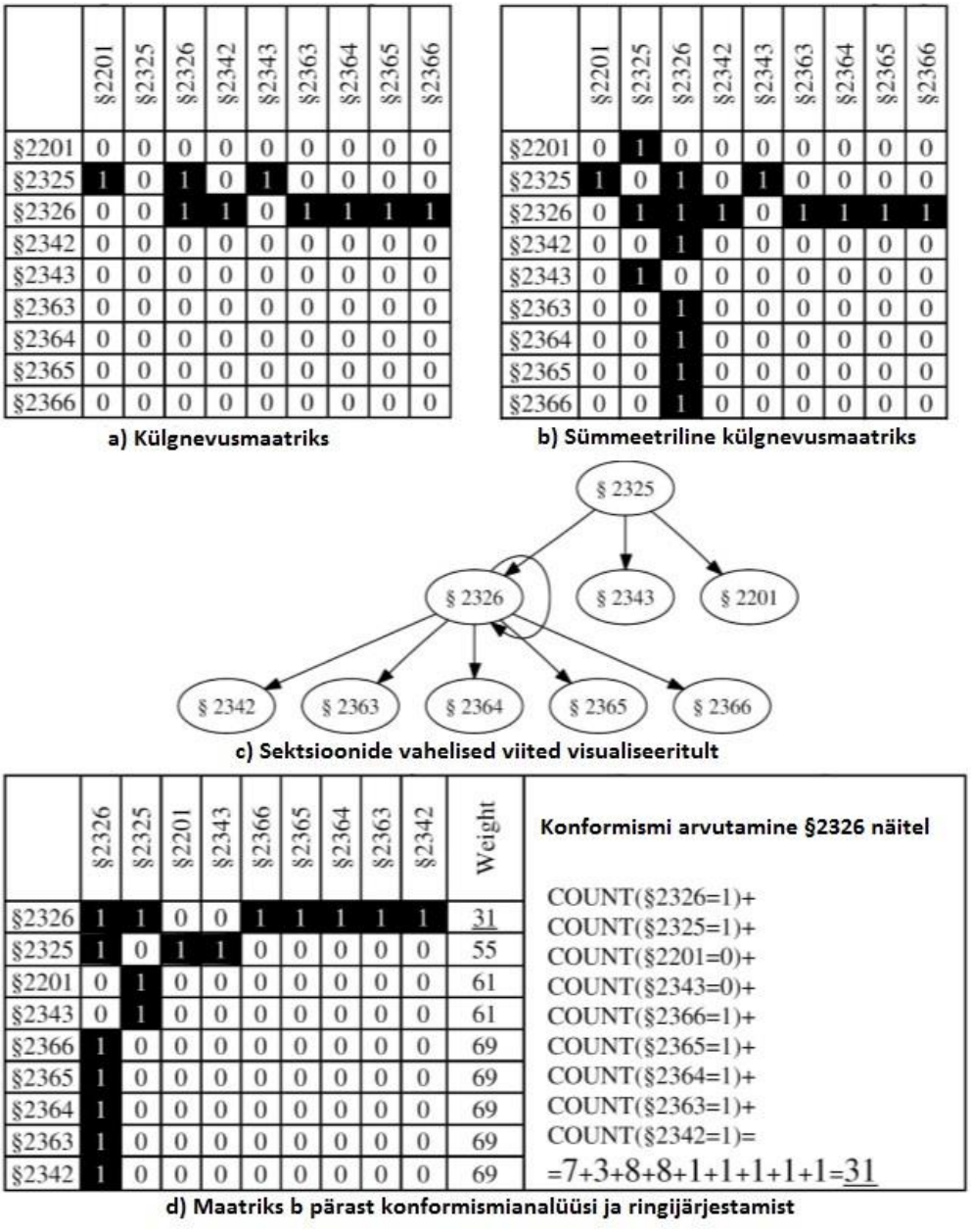
2.1 Ülevaade relevantsusmõõdiku leidmise algoritmist

Uuritav algoritm tugineb ühe seadusandliku akti sisemise struktuuri analüüsile ja visualiseerimisele ning pakub välja metoodika sektsioonidele relevantsuskaalude arvutamiseks sektsioonide vahelisi viiteid uurides.

Algoritmi põhiidee seisneb selles, et ühe seadusandliku akti puhul kaardistatakse ära akti sektsioonide vahelised seosed ning leitud seoste põhjal määratakse sektsioonidele kaalud. Viidete leidmisel on lubatud ka viited iseendale. Viiteid alamsektsioonidele käsitletakse kui viiteid alamsektsiooni vanemale. Leitud viited kaardistatakse maatriksina.

Numbriliste relevantsuskaalude arvutamiseks klassikalise võrgustike teaduse põhjal soovitatakse kasutada mõõdikut vahelolek (*betweenness*). Sektsioonide tegelike kaaluda arvutamiseks kasutatakse aga hoopiski konformismianalüüsi (*conformity analyses*) [2], mis järjestab maatriksi elemendid ringi nende sidemete esinemissageduse järgi – kõige rohkem ühendusi omavad sektsioonid eespool. Kaalu numbrilise väärtuse saamiseks kasutatakse tavalist liitmistehet, kus iga maatriksi rea puhul summeeritakse veerus oleva numbri vastete koguarv.

Kirjeldatud metoodika ülevaade on piltlikustatud joonisel 1 [1], kus maatriks a esitleb kaardistatud viiteid sektsioonide vahel, maatriks b peegeldab leitud viited sümmeetrilisse külgnevusmaatriksisse ning maatriks d puhul on seosed ringi järjestatud esinemissageduste järgi. Viidete struktuur on näha graafil c. Kirjeldatud arvutuskäik on välja toodud maatriksi d kõrval.



Joonis 1. Ülevaade relevantsusmõõdiku leidmise algoritmist.

Käesolevas töös püütakse algoritmi laiendada kogu seadusandlusele ning ühe akti asemel võetakse uurimise alla seadustevaheliste viidete analüüs, et selle abil igale seadusele

määrata relevantsuse kaal. Võrgustiku analüüsi käigus selgub, kas mõistlik on kasutada soovitatud vaheloleku mõõdikut või sobib relevantsuse määramiseks algoritmis olev konformismianalüüsiga leitud kaal. Välistada ei saa ka mõlemast mõõdikust kombineeritud kaalu kasutamist.

2.2 Ülevaade otsingutehnoloogiatest

Tehnoloogiate valik otsingusüsteemide ehitamisel, on käesoleval hetkel lai. Kasutada saab nii vabavaralisi tasuta tooteid (PostgreSQL, MySQL, ElasticSearch jt) kui ka tasulisi kommertslahendusi (Oracle, Azure Search, MS SQL jt). Täieliku ülevaate tegemiseks erinevatest lähenemistest, oleks vaja eraldiseisvat võrdlevat tööd, kuid oma olemuselt saab otsingu tehnoloogilised lahendused veebirakenduste puhul liigitada kolmeks:

- Lokaalne SQL andmebaasimootor (MySQL, PostgreSQL jt);
- Lokaalne andmebaasist sõltumatu spetsiifiline otsingutehnoloogia (ElasticSearch, Apache Solr jt);
- Otsing *SaaS* teenusena – otsing pilveteenusena, mis töötab üle API (Algolia, Azure Search jt).

Kuna enamasti tuginevad veebipõhised rakendused SQL andmebaasile, siis tavapärane lähenemine on otsing ehitada otse rakenduse aluseks oleva andmebaasi peale ehk kasutada lokaalset SQL tehnoloogiat. Lihtsamate süsteemide puhul on see üldjuhul ka kõige mõistlikum variant, kuna rakenduse äriloogika ei pruugi nõuda keerukat otsingusüsteemi ja SQL katab soovitud funktsionaalsed vajadused. Samas on viimasel ajal tarkvarainseneride poolt kiirelt tunnustust leidnud andmebaasist sõltumatud tehnoloogiad. Nende eeliseks on rakendusest eraldiseisev arhitektuur, parem täisteksti otsingu tugi, kiirus ning võimalus seadistada relevantsusalgoritmi. Viimase kategooriana välja toodud *SaaS* on samuti populaarsust kogumas, kuna pilveteenuste levikuga on see klientidele mugav alternatiiv muude variantidega võrreldes – puudub vajadus ise tegeleda hooldus- või jõudlusprobleemidega.

Järgnevates alapeatükkides antakse ülevaade tehnoloogiatest, mille hulgast saab teha valiku töö teises pooles realiseeritava prototüübi tarbeks. Iga eespool kirjeldatud kategooria puhul on välja toodud vähemalt üks tehnoloogia. Valik on tehtud lähtudes

sellest, milliseid tehnoloogiad praktikas kõige rohkem kasutatakse ja mis on hetkel populaarsed või populaarsust kiirelt kogumas [4], [5]. Tehnoloogiate detailsem analüüs tehakse töö teoreetilise poole järel, kui selgunud on nõuded planeeritavale otsingusüsteemile.

2.2.1 SQL

SQL on kõige levinum otsingutehnoloogia veebirakenduste juures, mis andmete hoiustamiseks kasutavad andmebaasi tuge. Kuigi erinevaid andmebaasimootoreid on palju, siis kõige tuntumad neist on MySQL [6] ja PostgreSQL [7], jättes kõrvale kallid kommertslahendused nagu Oracle või MS SQL. Märgitud andmebaasimootorite populaarsus on tingitud erinevatest asjaoludest. MySQL puhul võib tuua välja järgnevad põhjused:

- Vabavaralisus ja lai kogukond;
- Disainitud veebirakenduste jaoks;
- Madalaimad halduskulud;
- Ühilduvus erinevate operatsioonisüsteemidega;
- Skaleeritavus ja heal tasemel SQL standardi jälgimine.

Sarnaseid omadusi rõhutatakse ka PostgreSQL juures. Lisaks peetakse PostgreSQLi maailma kõige eesrindlikumaks vabavaraliseks andmebaasimootoriks, eelkõige tehnoloogiliste uuenduste tõttu nagu objektide relatsioonilisus, kastomiseeritavad andmetüübid ja kvaliteetne täisteksti otsingu tugi [8].

Jõudluse aspekte silmas pidades selgub, et mõlemad mainitud SQL tehnoloogiad toetavad *master-slave* replikeerimist ehk võimalust arhitektuuriliselt eraldada andmete säilitamise ja otsingu baasid ning neid mugavalt sünkroonis hoida.

Otsinguspetsiifilisi omadusi kõrvutades jääb PostgreSQL silma võimekama tootena kui MySQL [9]. Kui MySQL toetab täisteksti otsingu baasfunktsionaalsust, siis PostgreSQL puhul on selgelt välja toodud täiendavad võimalused nagu mitmekeelsus, stemmimise tugi, relevantsuse määramine ja hägusotsing (*fuzzy search*).

Miinustena võib erinevate SQL tehnoloogiate puhul välja tuua väga spetsiifilisi põhjuseid, kuid kvaliteetsete otsingute ehitamisel on enamasti takistusteks vajadus andmebaase denormaliseerida, mis suurte süsteemide puhul võib keeruliseks osutuda. Andmebaasimootorid ei ole esmajoones mõeldud otsingu tarbeks, vaid andmete struktuurseks hoiustamiseks. Kuigi MySQL ja PostgreSQL toetavad täisteksti otsingut, siis tihti võib täisteksti otsingu puudulik tugi otsingu kvaliteedile saatuslikuks saada – eriti, kui võrrelda seda alternatiivi pakkuvate spetsiifiliste otsingutehnoloogiatega. Lisaks ei võimalda SQL tehnoloogiad enamasti määrata otsingu relevantsusparameetreid või lubavad seda teha väga algelisel kujul.

Kuna juriidilise info otsing taandub ennekõike tekstiotsingule, siis kirjeldatud SQL põhised tehnoloogiad, koos täisteksti otsingu toega, on ilma detailsemate nõueteta piisavad otsingu realiseerimisel. Mainitud kahe toote puhul paistab esmapilgul eelistatum PostgreSQL. Seda eelkõige täisteksti otsingu parema toe tõttu.

2.2.2 ElasticSearch

ElasticSearch on avatud lähtekoodi ja HTTP (REST) veebiliidesega JSON formaadil tuginev otsingumootor, mis toetab täisteksti otsingut ning tugineb Apache Lucene otsingu tarkvarale [10]. ElasticSearchi peetakse tihti kõige populaarsemaks mitte SQL põhiseks otsingumootoriks [5].

ElasticSeachi põhilised plussid on:

- Spetsiaalselt mõeldud otsingu jaoks (erinevalt SQL'ist) koos täisteksti otsingu toega;
- Lihtne kasutada ja liidestada erinevate tarkvaraliste platvormidega tänu REST API'le;
- Lihtne otsingu konfigureerimine läbi REST API;
- Sissehitatud otsingu jaotamine (*distribution*), mis töötab üle API ja võimaldab vajadusel kiiret andmete replikeerimist uutesse ElasticSearch instantsidesse;
- Relevantsusparameetrite seadistamise võimalus ja detailne konfigureerimine;
- Kiirus ja võimekus otsida suurest hulgast andmetest (*Big Data support*);

- Lemmatiseerimise tugi.

Miinustena saab välja tuua:

- Dokumentatsioon kohati puudulik;
- Relevantsuse seadistamine võib nõuda ekspertteadmisi.

Juriidilise info otsingu ehitamisel tundub ElasticSearch tugeva kandidaadina, kuna relevantsusparameetrite seadistamine on üks võtmekohti otsingu ehitamisel ja nende parem seadistamise võimalus annab ElasticSearchile SQL'i ees eelise.

Selguse mõttes olgu siinkohal mainitud, et ElasticSearch on saadaval ka *SaaS* teenusena erinevate teenusepakkujate poolt, kuid käesolevas töös vaadeldakse ElasticSearchi siiski lokaalse installatsioonina.

2.2.3 Algolia

Algolia on pilveteenusena (*SaaS*) pakutav mitmeotstarbeline suletud lähtekoodiga otsingutehnoloogia, mis omab täisteksti otsingu tuge ja pakub otsingutulemusi esimesest klahvi vajutusest alates [11]. Võtmeomadustena võib välja tuua:

- Reaalajas otsing;
- Vigade tolereerimine koos seadistamisega (*fuzzy search*);
- Otsingutulemuste esile tõstmine (*highlight*);
- Relevantsusalgoritmi kastomiseerimine;
- REST API tugi ja liidestus enimlevinud programmeerimiskeeltes (PHP, Java, Python, Ruby jt);
- Mitmekeelsuse tugi;
- *SaaS*ist tulenevalt puudub vajadus hoolduse või skaleeritavusega tegelemiseks;
- Kasutajaliides konfigureerimiseks.

Algolia ise rõhub eriti oma kiiruslikele omadustele ning relevantsusalgoritmi lihtsale konfigureerimisele võrreldes otseste konkurentidega nagu ElasticSearch või Apache Solr.

Miinusteks võib Algolia puhul välja tuua:

- *SaaS* teenus maksab ja seda saab tasuta kasutada vaid piiratud andmemahtude korral;
- Lemmatiseerimise tugi on puudulikult kaetud;
- Suletud lähtekood.

Sarnaselt ElasticSearchiga tundub Algolia olevat hea alternatiiv klassikalisele SQL otsingule. Prototüüpimise juures peab kindlasti võrdlema relevantsusalgoritmi seadistamise võimalusi ja kaardistame erinevused ElasticSearchist.

3 Seaduste võrgustiku tsentraalsuse analüüs

Käesolevas peatükis võetakse vaatluse alla võrgustike teaduse teoreetilised alused ning püütakse uuritava seaduste võrgustiku jaoks valida mõõdikud, mida relevantsuse määramisel kasutada. Vastavate numbriliste väärtuste leidmiseks tutvustatakse esmalt tsentraalsusmõõdikuid, mis iseloomustavad võrgustiku sõlmpunkte ehk tippe ja aitavad paremini mõista kaalude leidmiseks valitavate mõõdikute sisu. Seejärel analüüsitakse, kuidas valitud mõõdikud kajastuvad käesoleva töö uurimisobjektiks oleval seaduste võrgustikul ning püütakse sobitada need uurimise all oleva algoritmi konteksti.

Tsentraalsuse analüüsi tulemuseks on kaardistatud seaduste võrgustik koos välja valitud relevantsuse mõõdiku ning leitud kaaludega, mida hilisemas prototüüpimise faasis kasutada. Samuti on teoreetilise osa väljundiks seisukoht ja muudatusettepanekud uuritava algoritmi sobivusest käesoleva töö subjektiks oleva võrgustiku analüüsimisel.

3.1 Tsentraalsusest üldiselt

Võrgustike teaduse seisukohast on tsentraalsusmõõdikute ülesanne tuvastada võrgustiku kõige olulisemad sõlmpunktid [12]. Mõõdik iseloomustab sõlmpunkti teatud omaduste järgi ja annab sõlmele numbrilise väärtuse ehk kaalu. Olulisusel on tsentraalsuses võrdlemisi lai tähendus, mis viib mitmete erinevate tsentraalsuse tõlgendusteni. Kõige üldisemalt kategoriseeritakse olulisus kaheks:

- Võrgustiku sõlmede sidususe hindamise järgi [12];
- Võrgustiku voo või ülekannete tüübi järgi [13].

Esimesel juhul iseloomustatakse võrku kui teed, mida mööda midagi liigub. Sellisel juhul saab tsentraalsuse klassifitseerida levimise tüübi järgi, mis antud kontekstis on oluline, arvestades leviva info või reaalse eseme atribuute. Siia kategooria alla saab liigitada info hargnemise inimeste vahel, kauba liikumise punktist A punkti B, viirusturundus jt.

Teine lähenemine uurib, kuidas tsentraalsus on üles ehitatud, analüüsides sõlmpunktide seost võrgustiku läbimise struktuurist (*walk structure*) lähtuvalt. Siin eristatakse kahte klassifikatsiooni [13]:

- Radiaalne;
- Mediaalne.

Radiaalne tsentraalsus loeb läbimisi, mis algavad või lõpevad kindlas võrgustiku sõlmpunktis. Mediaalse lähenemise puhul loetake sõlmpunkti läbimiste arvu teatud tingimuste alusel.

3.1.1 Klassikalised tsentraalsuse mõõdikud

Klassikalises käsitluses on kõige tuntumad kolm tsentraalsusmõõdikut [14], [15]:

- Suhete arv (*degree*) – iseloomustab tipu või sõlme suhete arvu teistesse tippudesse. Kui sõlmpunkti ühendavatel joontel on suunad, saab eraldi vaadelda sissetulevaid ja väljaminevaid seoseid. Sotsiaalvõrgustiku puhul saab suundadele omakorda tähendusi omistada (populaarsus, seltskondlikkus). Iga sissetulev või väljaminev sõlm annab tsentraalsuse mõõdu jaoks ühe punkti;
- Vahelolek (*betweenness*) – iseloomustab mitu korda sõlmpunkt käitub vahesillana mistahes kahe teise tipu ühendamisel kõige lühemat teed pidi. Tipul, millel on kõrge vaheloleku tsentraalsus, võib olla suurem mõju võrgule, kuna läbi nende liigub ja on kontrollitav teiste sõlmede jaoks oluline info. Selliste sõlmede eemaldamine võrgust häirib kõige rohkem infovahetust teiste sõlmede vahel;
- Lähedus kõigile (*closeness*) – summa lühimatest kaugustest vaadeldava tipu ja kõigi teiste tippude vahel. Mida väiksem on väärtus antud mõõdiku puhul, seda suurem on lähedus ja tsentraalsem on tipp. Eristatakse ka normaliseeritud lähedust, mis on keskmine kaugus vaadeldava tipu ja teiste tippude vahel. Antud mõõdiku järgi leitud tsentraalsed tipud võivad omada paremat ligipääsu võrgustikus olevale infole või suuremat otsest mõju teistele tippudele. Läheduse puhul on oluline ka sõlmpunkti ühendavate joonte suund.

Eelmises peatükis mainitud klassifitseerimise puhul kategoriseerub esimene kirjeldatud mõõdik radiaalseks ja kaks viimast mediaalseteks mõõdikuteks.

Lisaks võib tuntumatest mõõdikutest ära märkida *eigenvector*i tsentraalsuse. *Eigenvector* on teatud mõttes suhete arvu edasiarendus. Kui suhete arvu puhul arvestatakse kõiki suhteid, mida tipp omab, siis *eigenvector* kasutab tsentraalsuse arvutamiseks ainult teatud olulisi seoseid ja peab sõlmpunkti tähtsaks, kui ta on seotud teiste tähtsate sõlmpunktidega. Kehtib printsiip, et sõlm on oluline, kui tal on olulised naabrid [16].

3.1.2 Mõõdikute korrelatsioon

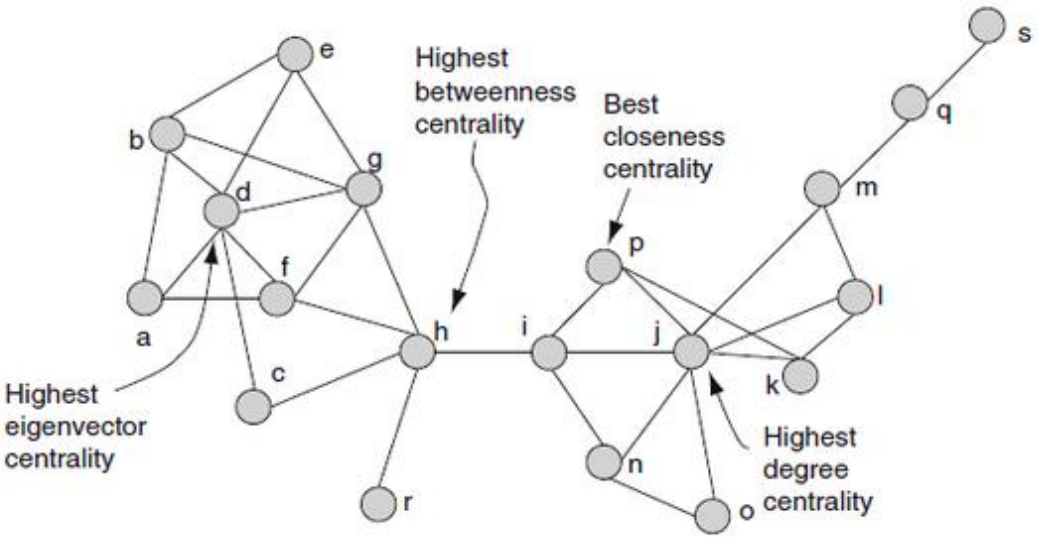
Tsentraalsusmõõdikute uurimisel ja konteksti asetamisel tuleb arvestada ka nende korrelatsiooni üksteisesse. Mõõdikud tuginevad andmetele sarnasest külgnevusmaatriksist ja nende väärtus tuleneb erinvatest matemaatilistest arvutuskäikudest. Mõõdikute liiga suure korrelatsiooni puhul võib erinevate mõõdikute kasutamist pidada üleliigseks. Sellele vastupidiselt peaks erinevad mõõdikud väiksema korrelatsiooni puhul näitama selgelt eristatavaid ja tõlgendatavaid tulemusi [17].

Erinevate käsitluste kohaselt on üle keskmise suuremas omavahelises seoses suhete arv ja vahelolek. Samuti on suur korrelatsioon suhete arvu ja *eigenvector*i tsentraalsuse vahel [16], [18].

Käesoleva töö kontekstist lähtuvalt on mõistlik pärast seaduste võrgustiku analüüsi hinnata, kui suures korrelatsioonis viidatud mõõdikud on ning suure korrelatsiooni puhul valida vaadeldavast paarist ainult üks mõõdik.

3.1.3 Piirangud tsentraalsusmõõdikute uurimisel

Tsentraalsust ja selle mõõdikuid ei saa käsitleda ühtemoodi erinevate võrgustike peal. Lähenemine, mis sobib ühele situatsioonile, võib vastupidiseid tulemusi anda teistes olukordades. Siin tuleb arvestada sellega, mida tsentraalsuse seisukohast „oluliseks“ peetakse. Kui vaadelda kuulujutu levimist inimeste vahelises võrgustikus, siis enamasti sellel ei ole konkreetset sihtpunkti või lühimat teed. Isegi sihtpunkti olemasolul on ülimalt vähetõenäoline, et see jõuab sinna lühimat teed pidi. Pigem hajub kuulujutt võrgustike erinevates sõlmpunktides ja sellisel juhul ei omaks eespool mainitud vaheloleku mõõdik suurt väärtust tsentraalsuse kirjeldamisel, kuigi võrgustiku struktuur võib sellele kõrge kaalu anda. Samamoodi ei pruugi vahelolek kirjeldada võrgustiku kõige populaarsemat inimest, kelle kaudu kuulujutt kõige kiiremini leviks, kuid kes võib olla ainukeseks sidemeheks teise suure kogukonnaga. Kirjeldatud olukorda iseloomustab hästi joonis 2 [19]:



Joonis 2. Erinevad tsentraalsuse mõõdikud ühe võrgustiku näitel.

Eelneva teoreetilise osa saab kompaktselt ja kokkuvõtlikult sõnastada allolevate punktidena:

- Võrgustiku tsentraalsust analüüsides on oluline teada, mida antud võrgustiku kontekstis oluliseks peetakse;
- Tsentraalsusmõõdikuid ei saa käsitleda õigete või valedena, vaid need vastavad erinevatele küsimustele, iseloomustades erinevaid omadusi;
- Tsentraalsusmõõdikud on omavahel teatavas korrelatsioonis, kusjuures väiksem korrelatsioon kahe mõõdiku vahel näitab erinevat sõlmpunkti iseloomuomadust kummagi mõõdiku seisukohast lähtuvalt.

Käesolev töö lähtub tsentraalsuse käsitlemisel samuti loetletud printsiipidest.

3.2 Võrgustiku konstrueerimine

Võrgustiku analüüsimiseks on esmalt vaja võrgustik konstrueerida. Selleks tuleb sisendandmed Riigiteataja veebiväljaandest XML kujul alla laadida ning seadustevaheliste viidete leidmiseks teostada andmete tekstiline parsimine, kuna

Riigiteataja infosüsteem automaatselt viiteid tuvastada ei võimalda. Baasaktide nimekiri on kättesaadav aadressilt: https://www.riigiteataja.ee/lyhendid.html

Aktide allalaadimiseks ning töötlemiseks kasutatakse Symfony PHP raamistikku [20]. Viidete parsimiseks ning hiljem prototüüpimise faasis kasutamiseks, salvestatakse need maha andmebaasi. Pärast andmete allalaadimist teostatakse iga akti kohta seoste parsimine läbi regulaaravaldise, millega otsitakse läbi kõik viited kogu võrgustiku moodustavate seaduste kohta pealkirja järgi. Viidete otsimisel tuvastatakse ka viited iseendale. Viiteks iseendale loetakse ka viidet seaduse siseste paragrahvide vahel või lõikude omavahelisel viitamisel.

Esmase andmetöötluse tulemuseks on kahe tabeliga normaliseeritud andmebaas, kus ühes tabelis on aktid ja teises tabelis viited aktide vahel. Viidete puhul on teada ka viitamise suund. Sisendandmed arvudes on näha allolevas tabelis 1:

Tabel 1. Sisendandmed arvudes.

Nimetus	**Arv**
Aktid	365
Viited aktide vahel	4678
Aktid, mis ei viita teistele aktidele peale iseenda	6
Aktid, millele ei viidata teiste aktide poolt peale iseenda	42
Aktid, mis ei viita teistele aktidele peale iseenda ja millele ei viidata teiste aktide poolt peale iseenda	2

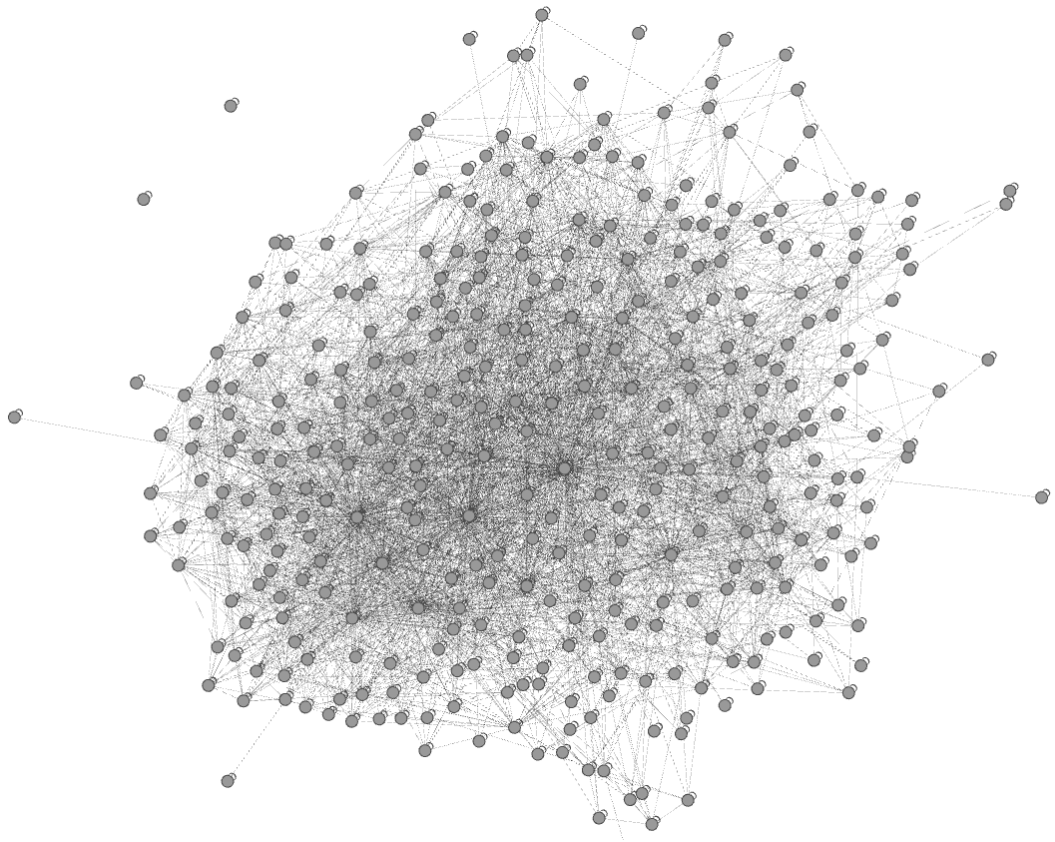
Lisaks tuvastati parsimise käigus ka korduvad viited samade aktide vahel, kuigi tsentraalsuse seisukohast korduv viidete arv ei oma tähendust, sest arvutused tehakse külgnevusmaatriksis olevate andmete põhjal, kus määravaks on lihtsal binaarväärtus, mis ütleb kas seos eksisteerib või mitte.

Andmete töötlemiseks ja parsimiseks ehitatud lähtekood on vabavarana kättesaadav aadressilt: https://github.com/tarmopoldme/legislation

3.3 Ülevaade koostatud võrgustikust

Tabelis 1 esitatud andmetest on näha, et üle 99% aktidest on omavahel seotud. See tähendab, et enamusele aktidest saab määrata tsentraalsuse mõõdiku. Seoseid mitte omavate aktide kaalud saab esialgu taandada nulliks. Täpsuse huvides on siinkohal korrektne ära märkida, et kuigi kõik aktid omavad vähemalt viidet iseendale, siis tsentraalsuse ja võrgustiku seisukohast lähtudes ainult seda arvestada ei saa, sest viited iseendale elimineerivad vastavad aktid võrgustiku mõistes eraldiseisvateks saarekesteks.

Andmete visualiseerimiseks ja mõõdikute tuvastamiseks analüüsitakse konstrueeritud võrgustikku Gephis [21]. Selleks tuleb tuvastatud viited konverteerida Gephi jaoks sobilikku CSV formaati ja rakendada võrgustiku visualiseerimise valikuid. Esmane võrgustiku struktuur, ilma mõõdikute arvutamist rakendamata, on näha alloleval joonisel 3:



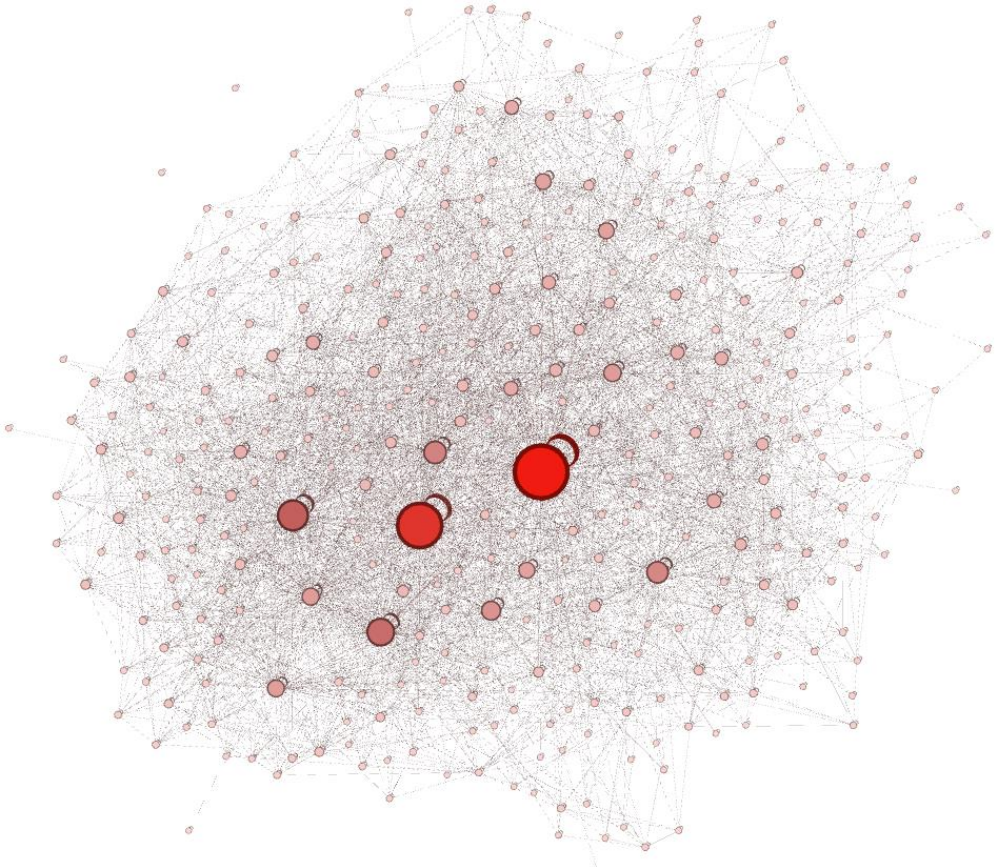
Joonis 3. Seaduste võrgustiku esmane struktuur.

Joonisel on vasakul üleval selgelt eristatavad 2 akti, mis peale iseenda viidete ei viita mujale ega oma ka sissetulevaid viiteid teistelt aktidelt. Samuti on visuaalselt aimatavad

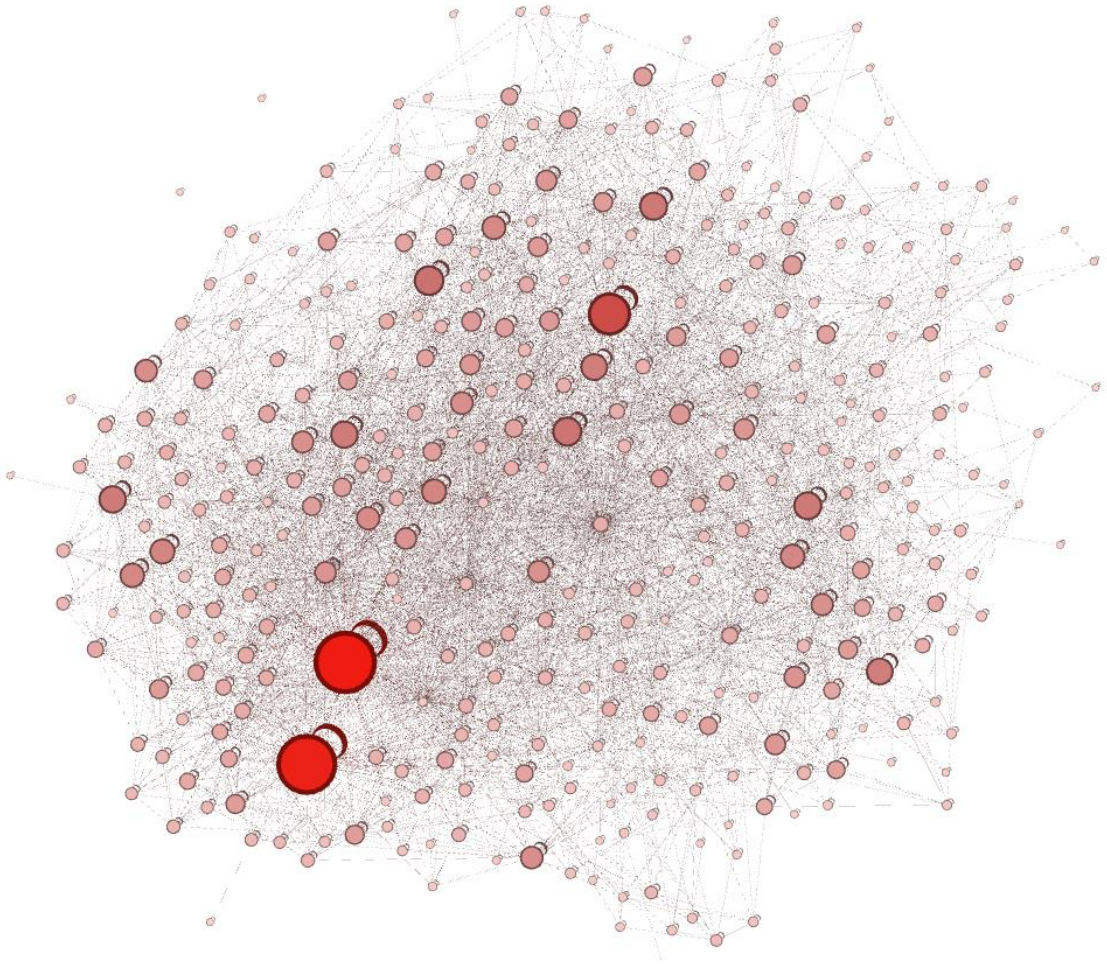
võrgustiku keskel olevad suurte sidemete arvuga aktid. Võrgustiku äärealadele on koondunud aktid, mis omavad minimaalselt otseseid seoseid teiste aktidega.

3.3.1 Võrgustiku esmane visualiseerimine ja mõõdikute tõlgendus

Parema ülevaate saamiseks on mõistlik koheselt rakendada Gephis tsentraalsusmõõdikute arvutamist ning tulemused visualiseerida. Gephi võimaldab uuritavat võrgustikku visualiseerida vaadeldava mõõdiku väärtuse järgi, muutes vastavat tippu proportsionaalselt suuremaks või väiksemaks. Spetsiifiliste võrgustike puhul võib taoline visualiseerimine anda juba uurimise algjärgus infot, milline mõõdik on uuritava võrgustiku jaoks oluline, arvestades võrgustiku iseloomu. Parema visualiseerimistulemuse saamiseks valitakse tippudele ka värviskaala. Mida intensiivsema värviga on tipp, seda suurem tsentraalsus vastava mõõdiku lõikes tal on, v.a läheduse puhul, kus tulemus on vastupidi tõlgendatav. Erinevad mõõdikud on kajastatud visualiseeringuna allolevatel joonistel 4, 5, 6, 7 ja 8. Mõõdikuid analüüsitakse peatükis 3.1.1 kirjeldatud järjekorras.



Joonis 4. Sissetulevate suhete arv visualiseeritult.

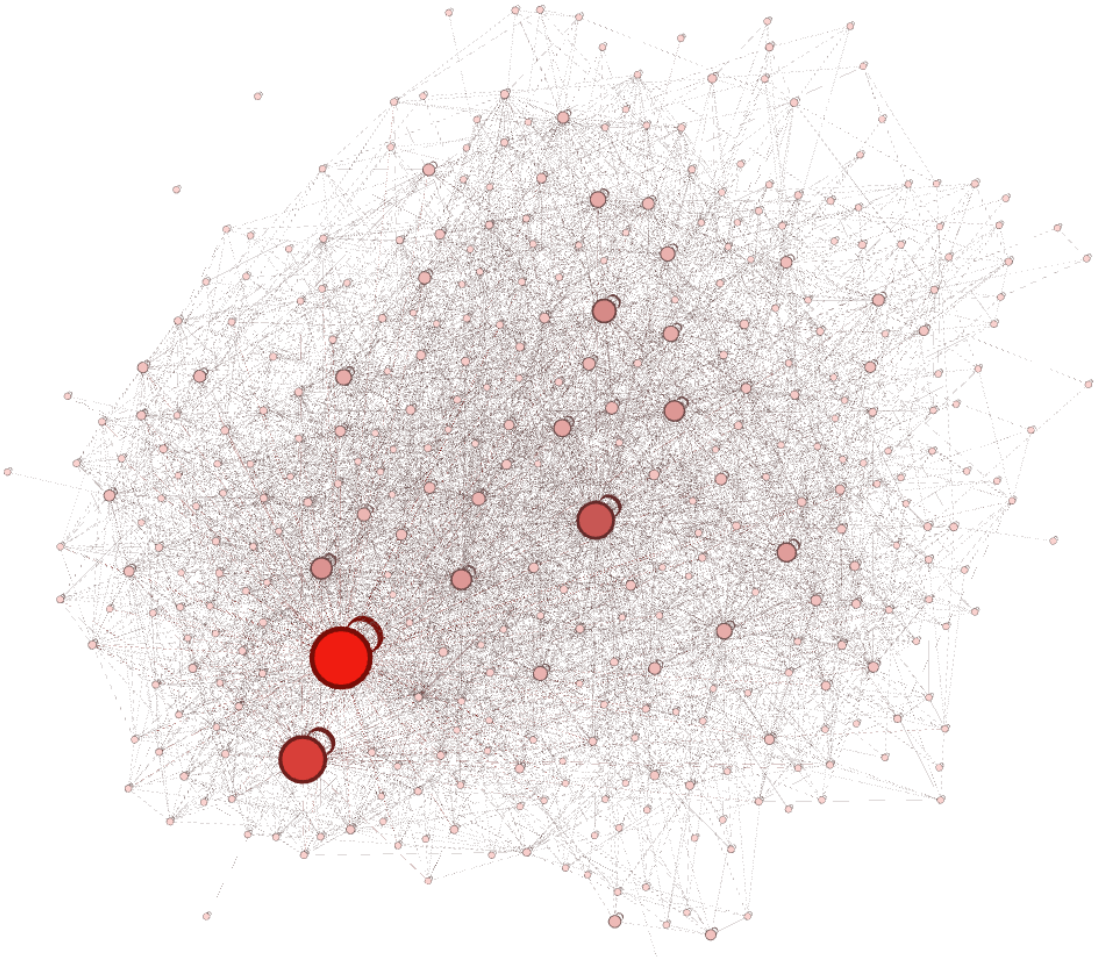


Joonis 5. Väljaminevate suhete arv visualiseeritult.

Kummagi mõõdiku tulemust visualiseeritult vaadates on näha, et tegu ei ole peegelpildiga, nagu pealiskaudsel süvenemisel eeldada võiks. Mõlema mõõdiku puhul on väga tugevalt esile kerkivaid tippe vähe. Märgatav on, et väljaminevate suhete arv on suurem kui sissetulevate suhete arv ühe akti piires ning sissetulevad suhted on ühtlasemalt jagunenud erinevate aktide vahel.

Semantiline tõlgendus suhete arvu mõõdikule, uuritava seaduste võrgustiku seisukohast, on loogiliselt tuletatav. Antud võrgustikus on teatud arv väga olulisi akte, mis pakuvad infot paljudele teistele aktidele (sissetulevad viited). Väljaminevate suhete arv on ühtlasemalt jagunenud, sest kõik aktid ei saa olla ühtemoodi infoallikaks vaadeldava akti seisukohast. Sarnast tuletuskäiku võib näha analoogses töös ka teiste autorite käsitluses,

kus näitena analüüsitakse USA seadusandluse sisemist struktuuri võrgustike teaduse mõõdikuid kasutades [22].

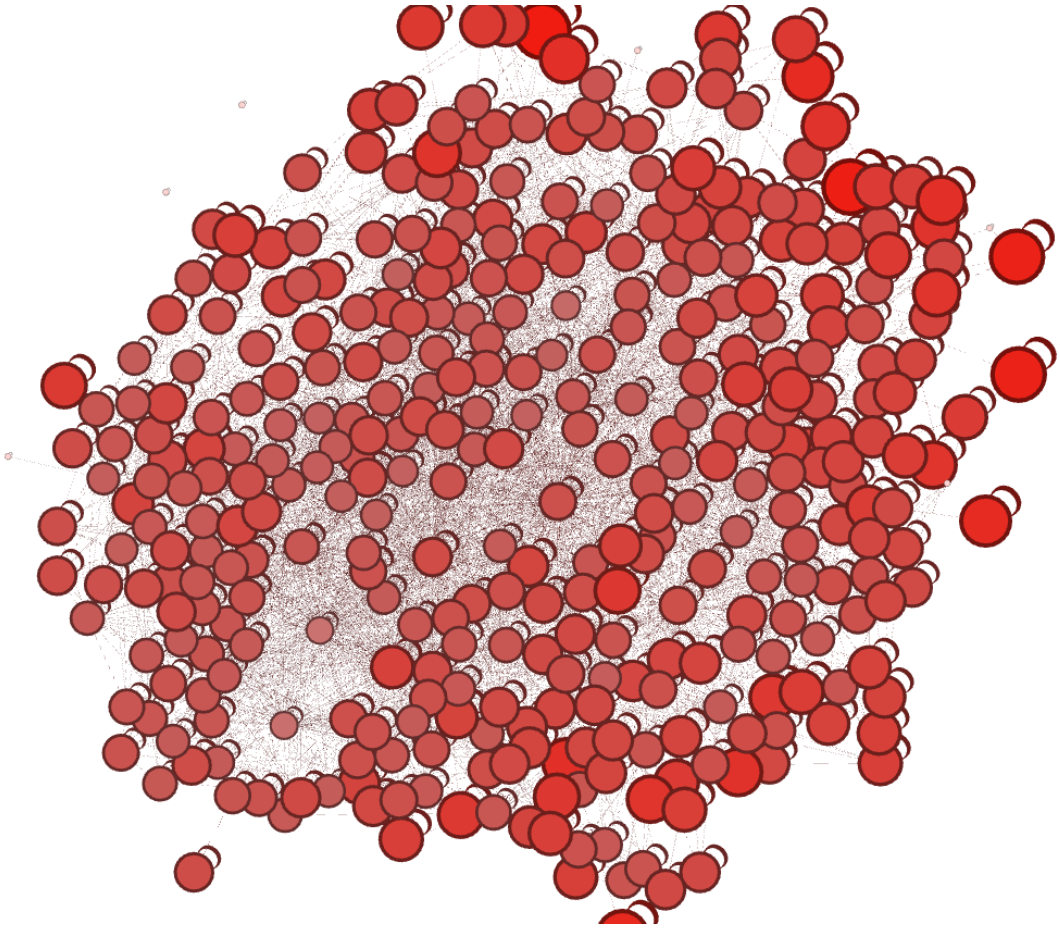


Joonis 6. Vahelolek visualiseeritult.

Vaheloleku mõõdiku puhul on märgatav, et olulisi sõlmpunkte on vähem kui suhete arvu puhul ja madalama kaaluga sõlmpunkte on oluliselt rohkem võrgustiku äärealadel. Täheldatav ei ole ka visuaalne sarnasus kummagi suhete arvu mõõdikuga, kuigi käesoleva töö alapeatükis 3.1.2 tõdeti, et mainitud mõõdikute vahel võib olla keskmisest suurem korrelatsioon.

Kuna seaduste võrgustikus on sõlmpunkte ühendavatel joontel suunad, siis suurem vahelolek tähendab, et antud akt on ühtemoodi oluline teistele aktidele konteksti andmisel kui ka iseenda konteksti loomisel, sõltudes teistest. Suurema väärtusega sõlmed on

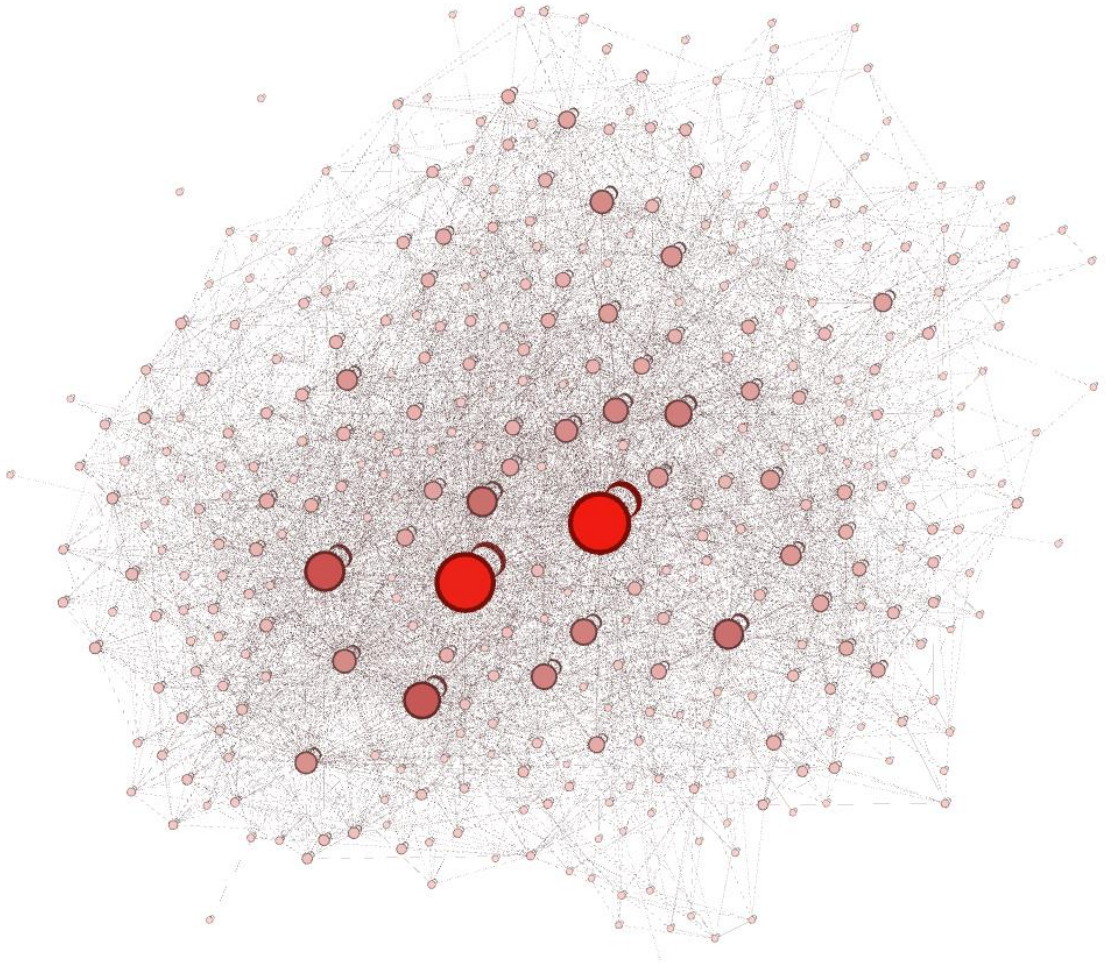
kitsaskohtadeks võrgustiku läbimisel, kuid ilma nendeta puuduks võrgustiku eri osade või sõlmpunktidest moodustuvate gruppide vahel ühendustee. Lisaks võib suurem vahelolek seaduste võrgustikus viidata erinevate valdkondade kokkupuutepunktidele [22].



Joonis 7. Lähedus kõigile visualiseeritult.

Läheduse mõõdiku tõlgendamisel ning joonise lugemisel on oluline arvestada, et pisem sõlm ja vähem erksam värv tähendab suuremat lähedust, sest läheduse algoritm arvutab teatavasti kaugust – mida väiksem väärtus, seda parem tsentraalsus.

Nagu jooniselt näha, ei sobi lähedus kõigile väga hästi tsentraalsusmõõdikuna relevantsuse hindamiseks seaduste võrgustiku jaoks, mis omab paljusid erinevaid viiteid. Nii värv kui sõlmpunktide suurus vihjab, et mõõdiku kaal on väga ühtlaselt jagunenud üle kogu võrgustiku, välja arvatud mõned vaevu märgatavad erandid võrgustiku keskel. Selle põhjuseks on asjaolu, et võrgustik omab ühtlaselt jagunenud suunatud ühendusteid sõlmpunktide vahel, mis ei anna ühelegi neist märkimisväärset eelist mõõdiku tsentraalsuse seisukohast.



Joonis 8. *Eigenvector* visualiseeritult.

*Eigenvector*i tsentraalsuse joonis on väga sarnane sissetulevate suhete arvuga joonisel 3. Nagu käesoleva töö alapeatükis 3.1.2 tõdeti, on kahe mainitud mõõdiku vahel keskmisest suurem korrelatsioon, mida visuaalselt on jooniselt lihtne näha. Antud tulemi põhjenduseks saab pidada kummagi mõõdiku sõltuvust otsestest seostest teiste võrgustiku sõlmpunktidega [18]. Sissetulev suhete arv on kõige olulisem indikaator tuvastamaks e*igenvector*i mõõdiku väärtust [19].

Käesoleva töö kontekstist lähtuvalt on sissetulevate suhete arv ja *eigenvector* liiased ning mõistlik ei ole neid mõlemat edasi uurida. Seega välistatakse siinkohal e*igenvector*i edasine analüüs.

3.4 Tulemuste analüüs ja tsentraalsusmõõdiku valik

Uurimise all olev algoritm soovitab kasutada relevantsuse määramiseks vaheloleku mõõdikut [1]. Senine sisendandmetel rakendatud analüüs Gephis seda soovitust ümber lükanud ei ole. Gephis tehtud visualiseeringu ning esmaste tulemuste tõlgendmise järel saab klassikalistest tsentraalsusmõõdikutest välistada läheduse kõigile ning liiase mõõdikuna e*igenvector*i tsentraalsuse. Valikusse jäävad alles suhete arv ja vahelolek.

Suhete arvu puhul tundub mõistlik käsitleda mõõdikut ühe tervikuna, mitte eristada sissetulevaid ja väljaminevaid suhteid. See tähendab, et relevantsuskaalu arvutamisel on sissetulevad ja väljaminevad viited summeeritud. Seaduste võrgustikku ei saa võtta kui klassikalist sotsiaalset võrgustikku, kus sõlmpunkte ühendavate joonte suund näitab populaarsust või seltskondlikkust ja puudub ka otsene teadmine, mis kinnitaks, et üks suund oleks teisele eelistatum. Lisaks annab suhete arvu ühtne tõlgendamine koos summeerimisega parema kaalu jaotuse väärtuse seisukohast – sarnase kaaluga akte on vähem.

Järelduste tegemiseks vaadeldakse esmalt kummagi mõõdiku kõrgema relevantsusega akte tabelis 2.

Tabel 2. Relevantsusskoor sissetulevate suhete arvu ning vaheloleku lõikes.

Nr	**Suhete arv**	**Suhete arvu kaal**	**Vahelolek**	**Vahel-oleku kaal**
1.	Vabariigi valitsuse seadus	285	Riigilõivuseadus	19 915
2.	Haldusmenetluse seadus	232	Majandustegevuse seadustiku üldosa seadus	14 786
3.	Korrakaitseseadus	165	Vabariigi Valitsuse seadus	11 262
4.	Riigilõivuseadus	160	Maksukorralduse seadus	6 367
5.	Majandustegevuse seadustiku üldosa seadus	154	Korrakaitseseadus	5 544
6.	Asendustäitmise ja sunniraha seadus	124	Täitemenetluse seadustik	5 231
7.	Võlaõigusseadus	106	Haldusmenetluse seadus	5 174
8.	Maksukorralduse seadus	105	Tsiviilkohtumenetluse seadustik	4 676
9.	Karistusseadustik	100	Kriminaalmenetluse seadustik	4 017

Nr	Suhete arv	Suhete arvu kaal	Vahelolek	Vahel-oleku kaal
10.	Täitmenetluse seadustik	90	Avaliku teenistuse seadus	3 464

Leitud kaalude täielik tabel on kättesaadav käesoleva töö käigus ehitatava prototüübi avalikust repositooriumist https://github.com/tarmopoldme/legislation ja „data“ alamkataloogist.

Tabelist selgub, et 10 kõrgema skooriga akti hulgas on 7 akti mõlema mõõdiku lõikes kattuvad, kuigi kattumine ei ole samadel positsioonidel. Kattuvus valikusse jäänud mõõdikute osas on selgitatav võrgustiku suhteliselt pisikeste mõõtmete ning suure arvu ühendusteedega. Positsioonide mittekattuvust selgitab mõõdikute erinev iseloom.

Aktide nimetusi semantiliselt uurides saab järeldada, et vahelolek eelistab mõnevõrra erinevamaid õigusvaldkonna akte. See võib tuleneda juba viidatud vaheloleku omadusest olla suurem ühendaja erinevate ainevaldkondade puhul [22]. Samas, ainult aktide semantilisi tähendusi võrreldes põhjapanevaid järeldusi teha ei saa ja tabeli tipus figureerivad kummagi mõõdiku lõikes küllaltki sarnased tulemused. Otsingu tarbeks sobiksid esmapilgul mõlemad mõõdikud.

Nagu tsentraalsuse teoreetilise osa analüüsis konstateeriti, tuleb mõõdiku valimisel lähtuda sellest, mis uuritava võrgustiku jaoks on oluline. Käesoleva töö peatükis 3.1.3 tõdeti, et üheselt ei ole võimalik defineerida, mis on õige või vale mõõdik. Eristada tuleks konteksti ja valdkonda, mille raames tsentraalsust tõlgendatakse.

Uuritava töö seisukohast on oluline, et akti tsentraalsust saaks maksimaalselt eristada. See tähendab, et leitud tsentraalsuse numbriline väärtus peaks iga akti lõikes olema võimalikult erinev, sest muidu ei saa akte eelistada otsingutulemuste järjestamisel. Kui analüüsida suhete arvu puhul leitud mõõdiku väärtusi Gephis, on näha, et esimene kaalude kattuvus tekib juba 17. positsiooni saanud aktide vahel. Liikudes tulemuste tabelis allapoole, selgub, et kattuvus suureneb madalamate kaalude lõikes veelgi ning kohati on sarnase kaaluga akte rohkem kui 10. Antud resultaati selgitab asjaolu, et suhete arvu leidmisel ei tehta keerulist matemaatilist arvutust, vaid liidetakse sissetulevate viidete arv üheks täisarvuks. Kui vaadelda suhete arvu kahte mõõdikut eraldiseisvatena,

on sissetulevate ja väljaminevate viidete lõikes kattuvus veelgi suurem. Selle põhjal ei saaks suhete arvu kaalu väga efektiivselt otsingutulemuste järjestamisel kasutada.

Võrreldes sama kriteeriumit vaheloleku mõõdiku peal, selgub, et vahelolek diferentseerib aktid paremini kaalude lõikes. Sarnased väärtused ilmnevad alles juhul, kui vaheloleku väärtuseks on 0. Selliseid akte on kokku 50 ja nende määramata väärtus tuleneb nende spetsiifilisest omadusest võrgustikus – puuduvad viited teistele või neile ei viidata, mis tähendab, et nende olulisust vaheloleku tsentraalsuse mõttes hinnata ei saa. Sisendandmete tabelist 1 on näha, et arv 50 klapib täpselt sellise kriteeriumi järgi eristatavate aktidega.

Sissetulevate suhete arvu kahjuks räägib ka asjaolu, et otsingusüsteemi jaoks oleks oluline minimaliseerida suhteline relevantsus. See tähendab, et kui otsingut teostatakse üle kogu võrgustiku, siis ei peaks eelistama tulemusi, mis on prominentsel kohal mingis võrgustiku alamosas. Näiteks on suhete arvu põhjal „Asendustäitmise ja sunniraha seadus“ 6. kohal, kuid vaheoleku põhjal kaugemal kui 100. positsioon.

Vastupidiselt eelistamisele võib suhete arvu puhul juhtuda ka see, et tähtsa sõlmpunktina funktsioneerivat akti ei hinnata piisavalt. „Kaubamärgi seadus“ on vaheloleku järgi 21. positsioonil, kuid sissetulevate suhete arvu järgi omab kaalu 8 ning on sellise väärtusega relevantsustabeli lõpus.

Arvestades tulemuste analüüsis tehtud avastusi, on parem ja diferentseeritum relevantsuskaal ning olemus vaheloleku mõõdikul. Seega on uuritava algoritmi esmane soovitus, seda mõõdikut kasutada, asjakohane.

3.5 Maatriksi konformismianalüüs

Eelmises alapunktis jõuti järeldusele, et tsentraalsusmõõdikutest on mõistlik relevantsuse määramiseks kasutada vaheloleku mõõdikut, kuna vahelolek diferentseerib klassikalistest mõõdikutest kõige paremini relevantsuse kaalud ning võimaldab otsingutulemusi selle alusel järjestada. Samas ilmnes, et 50 akti puhul on vaheloleku kaaluks 0 ning nende relevantsust otsinguga määrata ei saaks. Arvestades võrgustiku suurust, tähendaks see, et ligikaudu 14% aktide puhul ei oleks võimalik planeeritavas otsingusüsteemis eelistust määrata. Seetõttu oleks relevantsuse määramatuse minimaliseerimiseks mõistlik sisendandmetele rakendada ka konformismianalüüsi, nagu uuritav algoritm esmalt ette

nägi. Nagu näha töö metoodika peatükis olevalt jooniselt 1, siis konformismianalüüs võib kaalu määrata ka võrgustiku sõlmele, mis ise edasisi viiteid ei oma, kuid asub teatavas hierarhiliselt kõrgemas positsioonis teiste tipmiste sõlmpunktidega võrreldes.

Konformismianalüüs peaks lisaks relevantsuse määramatuse vähendamisele andma ka tagasisidet tsentraalsusmõõdikutega määratud relevantsuse tulemuste erinevusest. See aitab otsustada, kas vaheloleku mõõdikut on üldse mõistlik rakendada uuritava seaduste võrgustiku peal või piisab algoritmi esialgsest kujust kaalude määramisel. Seda juhul, kui mõlema mõõdiku tulemused on väga sarnased.

Konformismianalüüsiks on vaja programmilist lähenemist. Vastav arvutuskäik joonisel 1 on uuritava seaduste võrgustiku jaoks liiga mastaapne muul viisil teostamiseks. Kuna käesoleva töö peatükis 3.3 sai andmete esmaseks töötlemiseks juba PHP veebiraamistik seadistatud ning arvutamiseks vajalikud viited aktide vahel leitud, saab seda mugavalt teha seal ning leitud kaalud salvestada kohe andmebaasi aktide esmaste andmete juurde.

Parima relevantsuse saanud aktid konformismianalüüsi metoodikat kasutades on esitatud tabelis 3.

Tabel 3. Konformismianalüüsiga leitud prioriteetsemad aktid.

Nr	**Akt**	**Konformismianalüüsiga leitud kaal**
1.	Vabariigi valitsuse seadus	38 078
2.	Haldusmenetluse seadus	55 784
3.	Korrakaitseseadus	80 015
4.	Asendustäitmise ja sunniraha seadus	87 943
5.	Riigilõivuseadus	90 674
6.	Majandustegevuse seadustiku üldosa seadus	95 053
7.	Karistusseadustik	96 191
8.	Võlaõigusseadus	97 209
9.	Avaliku teabe seadus	101 174
10.	Maksukorralduse seadus	102 433

Konformismianalüüsi puhul on oluline silmas pidada, et väiksem kaal tähendab paremat relevantsuse positsiooni. Kogu tabel täpsemaks analüüsiks on kättesaadav juba viidatud prototüübi repositooriumist: https://github.com/tarmopoldme/legislation.

Võrreldes leitud tulemusi tsentraalsusmõõdikute tulemustega tabelist 2, selgub, et kattuvus suhete arvu resultaadiga on 90%. Lisaks on suhete arvu puhul mitmed positsioonid ka üks ühele kattuvad. Sellist tulemust selgitab kummagi mõõdiku tuletuskäik, mis lähtub otsestest suhetest sõlmpunktide vahel. See tähendab, et suhete arvu mõõdik antud algoritmi täiendusena relevantsuse määramiseks ei oleks otstarbekas, kuna kategoriseerub liiasuseks.

Vaheloleku puhul on tulemus oluliselt erinevam ja kattuvaid akte kõrgema relevantsuse saanud tulemuste seas on ainult 60%. See on isegi väiksem kui suhete arvu ja vaheloleku omavaheline kattuvus. Analüüsides leitud kaale edasi tabeli keskosas, väheneb kattuvus veelgi. Seega on vaheloleku mõõdiku kasutamine ja tsentraalsuse kaudu relevantsusele lähenemine asjakohane ning sobilik algoritmi edasiarenduseks. Vaja on vaid minimaliseerida määramata kaaludega aktide arv, et otsingu teostamisel relevantsust maksimaalselt rakendada.

Vahelolekuga määramata kaalude tuletamiseks läbi konformismianalüüsi, peavad konformismianalüüsiga leitud väärtused olema võimalikult vähe kattuvad. Kaalude esmane analüüs koos grupeerimisega andmebaasi tasandil kinnitab, et kattuvaid konformismianalüüsi kaalusid, mis vaheloleku puhul said kaaluks 0, nende 50 akti hulgas ei ole. Seejärel tuleks leida kõige viimasel positsioonil olev vaheloleku väärtus, mis on suurem kui 0 ning omistada kõigile määramata kaaluga aktidele leitud minimaalsest väärtusest väiksem väärtus, arvestades juba mainitud asjaolu, et konformismianalüüsi puhul tuleb omistamisjärjekord määrata kaalu põhjal kasvavalt.

Kõige väiksema määratletud vaheloleku kaaluga akt on hetkel „Teenetemärkide seadus“, väärtusega 0,024. Mõistlik oleks kaalu vähendamise samm valida selliselt, et relevantsuse skaala alumine ots ei läheks nullist väiksemaks, kuna semantiliselt ja tehniliselt võib taoline käsitlemine hilisemas prototüüpimise faasis arusaamatusi tekitada. Seega oleks sobilik vähendamise samm väärtusega 0,0001, mis jätab võimaluse positiivsete kaalude tõlgendamiseks uuritaval seaduste võrgustikul.

3.6 Kokkuvõte võrgustiku analüüsile ja algoritmi täiendamisele

Töö teoreetilise osa eesmärgiks oli analüüsida uuritava algoritmi sobivust seaduste võrgustiku suhtes ning arvestades tsentraalsusmõõdikute omadusi ja planeeritavat otsingusüsteemi, valida välja sobilik mõõdik relevantsuse määramiseks, mis võimaldaks otsingutulemusi järjestada.

Analüüsi põhjal välistati klassikalistest mõõdikutest suhete arv, lähedus kõigile ja *eigenvector*i tsentraalsus. Uurimise tulemusena selgus, et algoritmis välja pakutud vaheloleku tsentraalsuse mõõdik on sobilik relevantsuskaalu määramiseks. Relevantsuse määramatuse vähendamiseks võrgustiku äärealadele jäävate aktide osas pakuti välja lahendus kombineerida vaheloleku kaal koos konformismianalüüsiga selliselt, et kõik võrgustikus olevad aktid saaks unikaalse relevantsuse kaalu, mis võimaldab maksimaalselt ära kasutada relevantsust hilisemas otsingusüsteemis.

Juhul, kui uuritava võrgustiku struktuur on hajusam ja vahelolekuga leitav kaal jääb 50% juhtudest määramata, on tõenäoliselt mõistlikum rakendada ainult konformismianalüüsi, kuna mõõdikute kombineerimine nii suures ulatuses ei ole enam ilma täiendava uurimuseta usaldusväärne

.

4 Otsingumootori nõuete väljatöötamine

Selleks, et ehitada efektiivset ja ajakohast otsingumootorit, peab esmalt ära kaardistama nõuded plaanitavale otsingusüsteemile. Nõuete järgi on võimalik võrrelda potentsiaalsete tehnoloogiate sobivust, nende kasutamise tehnilist keerukust, realiseeritavuse otstarbekust ja maksumust. Samuti on mõistlik nõuete väljatöötamise käigus uurida, millised on olemasolevate otsingusüsteemide puudused, et hiljem verifitseerida, kas ehitatud süsteem on parem ja kui, siis mille võrra.

Alljärgnevas peatükis vaadeldaksegi esmalt otsingusüsteemide põhimõttelisi puudusi ja kaardistatakse ära nõuded, mida uue põlvkonna otsingumootor endas sisaldama peaks.

Selguse mõttes olgu siinkohal välja toodud, et käesolev töö ei käsitle otsingumootorite tehnoloogiaid, mis põhinevad erinevatelt veebilehtedelt leitud andmete indekseerimises ja nende serveerimises nagu Google. Töö keskendub uue põlvkonna veebipõhise otsingu ehitamisele, mis tugineb etteantud andmetele ehk antud juhul seadusandlikele aktidele.

4.1 Otsingusüsteemide puudustest

Otsingusüsteemi puudustena käsitletakse tihtipeale väga erinevaid ja kohati spetsiifilisi probleeme, kuid kokkuvõttes saab need kategoriseerida kolmeks eraldiseisvaks valdkonnaks:

- Halvasti disainitud kasutajaliides – liiga palju otsinguväljasid, arusaamatud selgitused väljade juures, puudulik dünaamiline veebidisain erinevates seadmetes, täiendavad toimingud otsingu teostamiseks jne [23]. Kasutajad ei peaks nuputama kuidas otsingut kasutada, vaid see peaks olema intuitiivne;

- Jõudlus – otsing ei tööta piisavalt kiiresti, eriti kui kasutatakse üldist tekstiotsingut ja lühikest otsingufraasi või rakendus sisaldab optimeerimata päringuid relatsioonilistesse andmebaasidesse [24];

- Puudulikud otsingutulemused – tulemused ei sisalda otsitavat infot, kuna tekstiotsingu implementatsioon/tehnoloogia ei võimalda osaliste vastete

(täisteksti otsing) puhul midagi tagastada või on vasted tagastatud relevantsust arvestamata [25].

Kasutajaliidese teemad on lai valdkond ja nende täielikuks katmiseks oleks vaja eraldi uurimuslikku tööd. Eriti seetõttu, et nutiseadmete suur mitmekesisus viimastel aastatel on kasutajaliideste nõuded uutesse kõrgustesse tõstnud. Kuna antud töö eesmärgiks ei ole disainialaste küsimuste lahendamine, siis detailselt seda järgnevas ei analüüsita ja põhiline rõhk on otsingu sisemise arhitektuuri uurimisel.

Teise kategooriana välja toodud jõudluse probleemid saavad enamasti alguse halvast otsingusüsteemi arhitektuurist, valest otsingutehnoloogiast või vähesest riistvaralisest võimekusest. Halvemal juhul on tegu kombinatsiooniga kõigist kolmest tegurist. Kui riistvaralist võimekust saab skaleeritava süsteemi puhul alati tõsta, siis halva otsingu tarkvaralise arhitektuuri ringitegemine võib olla oluliselt keerukam ja majanduslikult kulukam. Halva arhitektuuri põhjusteks võib SQL põhistes otsingusüsteemides lugeda denormaliseerimata andmebaasi struktuuri, mille puhul on vaja keerukaid tabeleid ühendavaid päringuid või juhul, kui denormaliseerimine on teostatud, ei teata, mida otsingu indeksis talletada [24].

Otsingutulemuste halb kvaliteet või selle puudumine võib olla põhjustatud samuti erinevatest teguritest. Näiteks ei pruugi tavaline SQL tehnoloogial tuginev otsing funktsionaalselt võimaldada kvaliteetselt osaliste vastete tagastamist või relevantsuse määramist leitud tulemustele. Kuigi alati on võimalik võrgustiku moodustavate andmete puhul tsentraalsusmõõtude kaudu kaalud leida ja neid kasutada, siis tavapraktikas kohtab autori isiklikele töökogemustele tuginedes selliseid lahendusi harva. Lisaks ei ole ainult tsentraalsusmõõtude relevantsust kasutavad otsingusüsteemid kuigi efektiivsed, kuna täisteksti otsingut need ei asenda.

4.1.1 Riigiteataja otsingu analüüs ja puudused

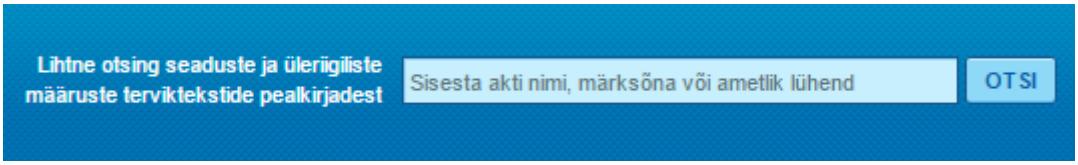
Järgnevalt uuritakse lähemalt töö sisendandmeteks oleva Riigiteataja otsingu funktsionaalsust, et hilisemas prototüüpimise etapis oleks seda võimalik võrrelda ehitatava lahendusega. Analüüsimisel pööratakse põhiliselt tähelepanu tekstiotsingu funktsionaalsusele, kuna antud töö skoopi ei mahu täiemõõtmelise otsingusüsteemi projekteerimine.

Riigiteataja otsing jaguneb kaheks:

- Kiirotsing, mille otsinguvorm on välja toodud joonisel 9 ja mis on nähtav rakenduse igas vaates paremal üleval;
- Detailotsing, mis on kujutatud joonisel 10 ja kuhu kasutaja pääseb vastava menüüpunkti kaudu peamenüüst.

Siinkohal on ääremärkusena korrektne ära mainida, et detailotsingu puhul vaadeldakse aktide terviktekstide otsinguvormi, mille sisendandmetele tugineb ka käesolev töö.

Parema ülevaate Riigiteataja otsingust saab vastavaid otsinguvorme analüüsides ning praktiliselt katsetades.



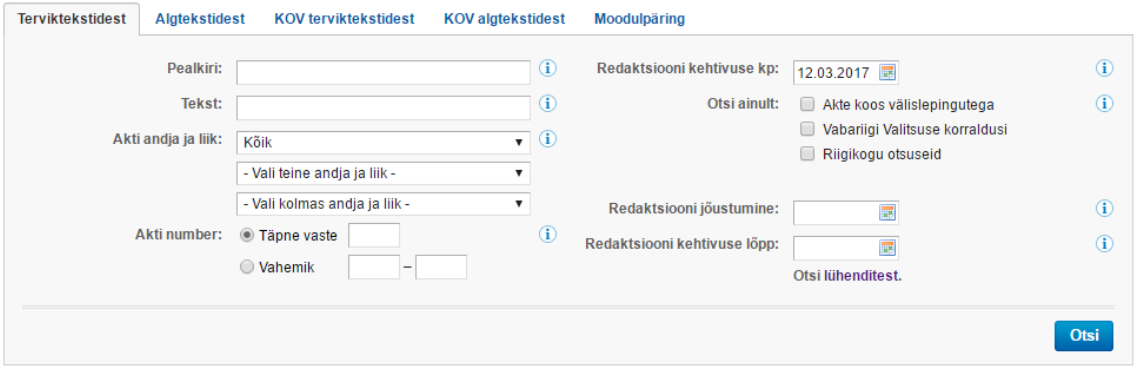
Joonis 9. Riigiteataja kiirotsingu vorm.

Kiirotsingu puhul jääb esimese puudusena silma, et otsing teostatakse ainult üle pealkirjade. Üldiselt võib eeldada, et kasutaja, kes teostab otsingut, ei taha leida aktide pealkirju, vaid süveneda akti sisusse ja pealkirja järgi otsimine vähendab märkimisväärselt tõenäosust otsitava infoni jõuda. Aktide pealkirjad on enamasti lakoonilised ja lühikesed ning otsitav infokild võib olla väga spetsiifiline.

Teise suure puudusena selgub praktilise katsetamise järel, et kiirotsingul ei ole täisteksti otsingu tuge. Näiteks, kui otsida sõnapaari „abielu lahutamine“, ei leia süsteem ühtegi akti ning kasutaja, kes tegeleb abielu lahutamisega, mis kahtlemata on keeruline nii isiklikust kui tihtipeale ka juriidilisest seisukohast, jääb ilma otsitava info ning nõuandeta. Kõige frustreerivam nende kasutajate jaoks võib olla see, kui mingi teise nurga alt see info siiski süsteemist välja ilmub. Selline puudulik funktsionaalsus võib vähendada kasutajate usaldust avalikesse infosüsteemidesse ja seadusandlusse tervikuna [3].

Relevantsuse rakendamise kohta otsingusüsteemi kasutajaliides vihjeid ei anna. Erinevaid märksõnu testides („tööandja“, „abielu“ jne), ei ole sellele kahjuks võimalik ka vastust saada, sest reeglipärasust silma ei jää.

Joonisel 10 näha olev detailotsing sisaldab oluliselt rohkem funktsionaalsust kui kiirotsing.



Joonis 10. Riigiteataja detailotsingu vorm.

Kuna detailotsingu vorm on ainuke koht, kus kasutaja saab akte sisutekstide järgi otsida, võib siinkohal esimese puudusena välja tuua vormi liigse keerukuse. Väljad, mis viitavad redaktsiooni kehtivusele või akti numbrile, ei ole vajalikud tavakasutajale, kes otsib infot teksti järgi. Kuna eeltoodu taandub kasutajaliidese disainile, siis siinkohal seda täpsemalt ei analüüsita, kuid märkimist väärib veel kord põhimõte, et kasutaja ei peaks otsima, kuidas otsingut kasutada. Kuigi tegu on detailotsinguga ning otsinguväljade rohkus on mõistetav, siis hea tava kohaselt võiks teisejärgulised väljad olla eraldatult või peidus ja kasutatavad vajadusel.

Püüdes tuvastada otsingutulemuste relevantsuse kriteeriumit, selgub sisuteksti välja järel olevast infoaknast, et otsingutulemused järjestatakse aktides leiduvate sõnade rohkuse järgi. Paraku ilmneb testimise käigus, et antud väide ei pea paika. Näiteks sõna „tööandja“ puhul on esimeseks vasteks „Töötajate usaldusisiku seadus“, kus otsitav sõna esineb 65 korda ning teisel positsioonil „Töölepingu seadus“, kus otsitav sõna figureerib 306 korda. Analoogne situatsioon on ka sõnaga „abielu“. Isegi, kui antud relevantsuse määramine töötaks, oleks küsitav, kas sellest on kasutajale praktilist kasu. Siin võib tekkida olukord, et esile kerkib suhteline relevantsus ja prominentsel positsioonil on tulemus, mis ei ole

keskne – täpselt nagu töö teoreetilises osas suhete arvu mõõdiku analüüsis tõdeti (peatükk 3.4).

4.2 Kaasaegse otsingumootori tunnused

Võttes arvesse eelnevas peatükis kajastatud puudusi, olemasolevate otsingutehnoloogiate võimalusi ja üldtuntud häid tavasid otsingusüsteemide ehitamisel, tuuakse järgnevas peatükis välja kaasaegset otsingumootorit iseloomustavad tunnused.

Tunnused on kaardistatud tuginedes otsingusüsteemide temaatikat käsitlevatele artiklitele, praktilistele näidetele internetis ning autori erialasele töökogemusele otsingusüsteemide ehitamisel. Märkimist väärib ka nüanss, et erialased teadusartiklid on tihtipeale antud valdkonna puhul natukene aegunud ega kajasta ajakohaseid trende. Seetõttu on allolevas tuginetud ka erinevatele praktilistele allikatele ja blogidele, mis uuritavat teemat on sügavuti analüüsinud:

- Täisteksti otsing – otsingusüsteem peaks oskama leida otsitavat teksti paindlikult üle erinevate tekstiliste atribuutide, arvestades iga sisestatud sõna eraldi ja kombineeritult. Kui võtta näiteks juriidilise informatsiooni otsingusüsteem, siis otsingufraas „Eesti Vabariigi haridus“ peaks tagastama kõik aktid, kus pealkirjas või sisus on antud sõnad, kas eraldiseisvalt või erinevates järjekordades;

  Selguse mõttes tuleb siinkohal tõdeda, et osade vanemate käsitluste kohaselt [26], ei peeta täisteksti otsingut kvaliteetseks, kuid kaasaegsed täisteksti otsingut toetavad tehnoloogiad sellist seisukohta ei jaga [10], [11];

- Struktureeritud otsingu indeks – otsitav info peaks otsitavate atribuutide osas olema struktureeritud lähtuvalt ehitatava süsteemi semantikast ning vajadustest. See tähendab, et otsinguindeks ei tohiks olla üks suur tekst või väljade kogum, kuhu kõik info on kokku pandud, vaid iga otsitav väli peaks indeksis eraldi kajastuma [24]. Selline lähenemine võimaldab otsingutulemuste relevantsusloogika määramisel atribuudipõhist lähenemist, andes otsitavatele väljadele prioriteedid, mis saavad esmase relevantsuse aluseks [27].

Tõmmates antud tunnuse osas paralleele juriidilise informatsiooni otsingusüsteemiga ja täisteksti otsinguga, peavad näiteks akti pealkiri ja sisu olema indeksis eraldi adresseeritud;

- Relevantsuse kasutamine – kaasaegne otsingusüsteem peaks sisaldama algoritmi, mis oskab otsingutulemusi relevantsuse järgi järjestada, arvestades otsingule edastatud sisendit (märksõna või fraasi), süsteemi semantikat ja otsinguindeksi konfiguratsiooni. Kasutajad on harjunud üldtuntud veebiotsingumootorite, nagu Google või Yahoo, omadusega olulist infot otsingutulemustes eespool kuvada ning ootavad samasugust käitumist ka väikeste infosüsteemide otsingutelt [28];

  Korrektne on antud alapunkti juures välja tuua, et antud töö kontekstis käsitletakse kahte eraldiseisvat relevantsust. Esimene on töö teoreetilises osas analüüsitud tsentraalsusega seotud fikseeritud relevantsus. Teine on otsingutehnoloogia sisemine dünaamiline relevantsus otsingutulemustele skoori arvutamisel, sõltuvalt otsingu sisendist ning indeksile antud struktuurist;

- Vigade tolereerimine – otsingusüsteem peaks olema piisavalt intelligentne, et aru saada ja „taluda“ (*tolerate*) elementaarseid sisestatud kirjavigu. Kasutajad eksivad otsingusõnade sisestamisel tihti [27], eriti kui kasutatakse mobiilseid seadmeid, kus ekraanipind on väike. Juhul, kui otsingusse sisestatakse märksõna „apielu“ või „tööandia“, siis kaasaegne otsingumootor võiks sisestatud kirjaveast aru saada ja leida tulemused, kus esineb otsitava sõna korrektne vorm;

- Kiirus – kasutajad on harjunud, et otsitav info leitakse reaalajas. Sellele ei aita kaasa ainult relevantse info ettepoole toomine otsingutulemustes, vaid otsing peab töötama minimaalse ajakuluga. Kui otsinguvormi sisestatakse otsingusõna, siis tavapärane ootus on, et kvaliteetne süsteem juba oskab selle põhjal soovitusi pakkuda (*search as you type*) [27], mitte ei oota kasutajapoolset käsklust otsingu teostamiseks.

Kuigi eelnevas loetelus said kvaliteedi tunnused välja toodud eraldiseisvate punktidena, siis praktikas on mitmed neist suuremal või vähemal määral seotud ja kattuvad. Näiteks on struktureeritud otsinguindeks seotud otseselt relevantsusega, sest indeksi struktuur on tihti aluseks relevantsusskoori arvutamisel – väljade järjekord määrab kuidas relevantsuskaal tulemustele leitakse. Vähem oluline ei ole ka täisteksti otsingu ja

relevantsuse seos, kuna relevantsusskoor üldjuhul arvutataksegi täisteksti otsinguga tuvastatud sõna või fraasi põhjal.

4.2.1 Naturaalse keele töötlemine kui kvaliteedi tunnus

Tekstiliste otsingusüsteemide puhul jõutakse varem või hiljem arusaamisele, et otsing töötaks oluliselt kvaliteetsemalt, kui tehnoloogia võimaldaks otsitavat sõna analüüsida tema algvormi vastu. Sisendi „abielluma“ puhul tundub loogiline, et otsing võiks tagastada ka tulemused, kus tekstis esineb sõna „abielu“ või „abielus“. Keeleteaduses on sõnade algvormidele taandamine tuntud kahe metoodikaga:

- Stemmimine (*stemming*);
- Lemmatiseerimine (*lemmatizing*).

Mõlema lähenemise puhul kasutatakse sõna erinevate infektiivsete vormide grupeerimist, et neid saaks analüüsida kui ühte tervikut [29], [30]. Näiteks „abieluline“, „abielluma“, „abielus“ puhul saaks otsingu taandada sõnale „abielu“, mis on mainitud kolme sõna algvorm. Erinevus metoodikate vahel tuleneb sellest, et lemmatiseerimise puhul tuginetakse ka sõnaraamatule ja morfoloogiale (vormi moodustamine) algvormi tuletamisel.

Juriidilise infosüsteemi otsingule annaks naturaalse keele töötlemine kindlasti palju juurde, kuna aktide tekstid on massiivsed ja sisaldavad paljusid erinevaid sõnavorme.

Naturaalse keele töötlemise miinuseks on enamasti see, et üldtuntud otsingutehnoloogiad ei toeta seda eksootiliste keelte puhul, mida eesti keel maailma mastaabis kindlasti on. Kuigi tihti on võimalik kasutada kolmandate osapoolte arendatud lingvistilist tarkvara, siis integreerimise keerukus või vastava tarkvara kvaliteet ei pruugi soovitud kvalitatiivset tulemust kokkuvõttes anda.

4.3 Nõuded juriidilise informatsiooni otsingule

Tuginedes kirjeldatud otsingusüsteemide puudustele ning kaasaegse otsingumootori tunnustele, dokumenteeritakse järgnevates alapeatükkides ära nõuded ehitatavale otsingu prototüübile.

Nõuete kaardistamisel lähtutakse põhimõttest, et nõue peab hiljem olema lihtsalt testitav ning mõõdetav. Nõuded on jagatud mittefunktsionaalseteks ja funktsionaalseteks. Nõuete jagamisel kahte kategooriasse kehtib üldtuntud põhimõte, et mittefunktsionaalsed nõuded määravad, kuidas süsteem toimib ning funktsionaalsed nõuded kirjeldavad konkreetsemalt süsteemi omadusi.

4.3.1 Mittefunktsionaalsed nõuded

Allolevas tabelis 4 on välja toodud otsingu mittefunktsionaalsed nõuded. Mittefunktsionaalsed nõuded on üldised ja kehtivad ka prototüübi üleselt võimaliku reaalse otsingusüsteemi jaoks. Nõuded on tähistatud inglise keelest tuletatud tähekombinatsiooniga NF (*non-functional*) ning sisaldavad järjekorra numbrit. Nõuete tähiseid kasutatakse edasises töös viitamiseks.

Tabel 4. Otsingu mittefunktsionaalsed nõuded.

Tähis	**Nõue**	**Kommentaar**
NF1	Otsingutehnoloogia peab toetama täisteksti otsingut.	Otsingu sisendina edastatud teksti võrreldakse otsinguindeksis täisteksti otsingutehnoloogiat kasutades. Iga sõna vaadeldakse indeksis eraldiseisvana otsingusisendi vastu.
NF2	Otsinguindeks peab võimaldama indekseeritavatele väljadele prioriteeti määrata.	Vaja on võimalust otsing seadistada nii, et akti pealkiri on suurema kaaluga kui akti sisu jne. See tähendab, et kui vaste leitakse pealkirjast, siis selline tulemus saab kõrgema relevantsuse.
NF3	Otsingutehnoloogia peab võimaldama relevantsusloogika seadistamist.	Kasutatav tehnoloogia peab võimaldama aru saada otsingutulemuste relevantsuse määramisest (see ei tohi olla läbipaistmatu sisemine algoritm) ning vajadusel seda lubama seadistada.
NF4	Otsingutehnoloogia peab võimaldama hägusotsingut.	Tehnoloogia peab toetama kasutaja sisestatud kirjavigadest arusaamist nii, et elementaarsete kirjavigade puhul säilitab otsing oma funktsionaalsuse.
NF5	Otsingutehnoloogia peab võimaldama kastomiseeritud relevantsuse seadistamist.	Otsingutehnoloogia peab lisaks sisemisele relevantsusele, võimaldama rakendada välise metoodikaga leitud relevantsust – antud töö kontekstis kehtib see nõue tsentraalsusmõõdiku kaudu leitud akti kaalu kaasamisele otsingusse.
NF6	Otsing peab kasutama lemmatiseerimist.	Lisaks täisteksti otsingule peab otsing kasutama sõnade algvormidele taandamist.

Mittefunktsionaalsete nõuete peatükis on asjakohane ära märkida, et reaalse otsingusüsteemi ehitamisel peaks vältima olemasoleva rakenduse arhitektuuri ja andmebaasi mastaapset ringi restruktureerimist. Kuigi käesolevas töös on prototüübi jaoks kasutatav andmemudel väga triviaalne ja koosneb kahest tabelist, siis reaaleluline süsteem võib olla oluliselt keerulisem ning selle denormaliseerimine mõistliku aja ning kuluga, teostamatu. Seega võib üldise funktsionaalse nõudena välja tuua ka omaduse, et kaasaegset otsingusüsteemi iseloomustab suhteline eraldatus rakenduse muust funktsionaalsusest. Ideaalis võiks otsing olla eraldiseisev arhitektuuriline kiht või hoopis väline teenus, mis rakenduse sisse implementeeritud.

4.3.2 Funktsionaalsed nõuded

Funktsionaalsed nõuded on kaardistatud tabelis 5 ja lähtuvad otseselt ehitatavast prototüübist. Nõuded on tähistatud tähe F ning vastava järjekorra numbri kombinatsiooniga.

Tabel 5. Otsingu funktsionaalsed nõuded.

Tähis	**Nõue**	**Kommentaar**
F1	Prototüüp sisaldab ühte tekstiotsingu välja analoogselt Riigiteataja kiirotsinguga.	Prototüüpimise faasis ei ole eesmärk ehitada keerukat otsingut erinevate väljadega, vaid demonstreerida töös uuritud relevantsuse ja täisteksti otsingu seadistamise võimalusi ning paremust võrreldes olemasolevate otsingusüsteemidega (näiteks Riigiteataja otsinguga).
F2	Tekstiotsing peab otsima vasteid nii akti pealkirjast kui akti tekstist.	Otsing peab töötama kõigi indeksis olevate tekstiliste väljade vastu. Prototüübis on esialgu lihtsuse mõttes kaks tekstilist välja: pealkiri ja akti sisu.
F3	Otsinguvorm peab töötama reaalajas.	Otsingutulemused peavad ilmuma hetkeliselt alates 1 sümboli sisestamisest. Kuna aktide andmeid on indeksis suhteliselt vähe, peaks päringu saatmine, otsingu teostamine ning vastuse töötlemine jääma alla 100 millisekundi.
F4	Kirjavigade puhul peab otsing suutma käsitleda kuni 2 eksitud tähte ühe sõna piires.	Hägusotsingu puhul peab olema lubatud kuni kaks viga ühes sõnas. Viga võib olla nii kõrvuti esinevate tähtede kui ka muude positsioonide osas.

Tähis	Nõue	Kommentaar
F5	Otsingutulemustes on eespool kõrgemat tsentraalsuskaalu omavad aktid.	Otsingutulemuste järjestamisel arvestatakse lisaks otsingutehnoloogia sisemisele relevantsusele käesoleva töö teoreetilises pooles leitud tsentraalsuskaalu nii, et kõrgema kaaluga aktid on eespool.
F6	Otsingu sisend on otsingutulemustes kuvatud muust tavatekstist eristuvalt.	Otsingu sisend on otsingutulemustes kuvatud teise värviga ja selgelt eristuvalt. Ka sõnaosa või osaline vaste on tulemustes eristuv.
F7	Otsingutulemused kuvatakse nimekirjana otsinguvormi all.	Otsingutulemused ilmuvad otsingu trükkimisel vastavalt nõudele F3 otsinguvormi all järjestatud nõude F5 järgi. Iga leitud tulemus sisaldab akti pealkirja, mis lingib Riigiteatajas vastava akti detailvaatesse.

Funktsionaalsed nõuded ei sea käesolevas töös piiranguid kasutajaliidesele, kuna antud töö ei keskendu kasutajaliideste temaatikale. Prototüüp on mõeldud toimima veebipõhiselt enimlevinud uuemates brauserites.

4.4 Nõuete kokkuvõte

Nõuete peatükis analüüsiti veebipõhiste otsingumootorite põhimõttelisi vigu ja kirjeldati Riigiteataja otsingu näitel ühe reaalse otsingusüsteemi nõrkuseid. Selle käigus tõdeti, et probleemseteks kohadeks on ootuspäraselt täistekstiotsingu puudumine ning vigane relevantsusalgoritm (või selle puudumine) otsingutulemuste järjestamisel.

Puuduste analüüsi järel kaardistati kaasaegse otsingumootori tunnused. Erinevatele allikatele tuginedes leiti, et uue põlvkonna otsingumootor võiks sisaldada täisteksti otsingu tuge, kasutada relevantsusalgoritmi otsingutulemuste leidmisel, olla toimiv reaalajas ja osata tolereerida kasutajate sisestatud kirjavigu.

Tuginedes tehtud analüüsile, töötati välja otsingu funktsionaalsed ja mittefunktsionaalsed nõuded, mida rakendada järgnevas prototüüpimise faasis.

5 Otsinguprototüübi ehitus

Selleks, et veenduda töö teoreetilises pooles analüüsitud tsentraalsusmõõtudega leitud kaalude ning nõuete praktilises väärtuses, realiseeritakse järgnevas peatükis otsingu prototüüplahendus. Prototüüpimise vahend valitakse käesoleva töö metoodika peatükis 2.2 kajastatud tehnoloogiate hulgast tulenevalt püstitatud nõuetest ja töö teemaks olevast subjektist.

Prototüüpimise tulemuseks on lihtsustatud otsingusüsteem, mis võimaldab sisestada otsingusõna või –fraasi ja kuvada otsingutulemused nimekirjana. Prototüüp peab arvestama kõiki nõudeid käesoleva töö tabelitest 4 ja 5. Realiseeritud tulemuse kvaliteeti võrreldaks näidispäringute komplektiga olemasoleva Riigiteataja otsingu vastu.

Prototüübi lähtekood on kättesaadav ja vabalt kasutatav aadressilt: https://github.com/tarmopoldme/legislation ning prototüübi toimiv veebirakendus hakkab asuma aadressil: http://tarmo.brainart.ee/legislation/web.

5.1 Tehnoloogiate analüüs lähtuvalt nõuetest

Tehnoloogia valimisel tuleb esmalt lähtuda otsingule seatud nõuetest, et selgeks teha, milliste pakutud vahenditega neid katta saab. Tabelis 6 on välja toodud kõik metoodika peatükis 2.2 kirjeldatud tehnoloogiad ja eelmises peatükis püstitatud nõuded risttabelina. Nõuete puhul on kasutatud tähiseid, mille täpsem selgitus on samuti eelmises peatükis vastavalt tabelites 4 ja 5. Kõigi tehnoloogiate puhul on lähtutud ametlikust dokumentatsioonist [6], [7], [10], [11] või täiendavatest allikatest, millistele viidatakse järgnevas analüüsis. Nõude realiseeritavust iseloomustab värvikood: roheline tähistab väga head, kollane osalist ning punane tugevalt piiratud võimalusi.

Tabel 6. Tehnoloogiate vastavus nõuetele.

Tehnoloogia	**Mittefunktsionaalsed nõuded**						**Funktsionaalsed nõuded**						
	NF1	NF2	NF3	NF4	NF5	NF6	F1	F2	F3	F4	F5	F6	F7
MySQL													
PostgreSQL													
ElasticSearch													
Algolia													

Järgnevates alapeatükkides uuritakse tabelis 6 visualiseeritud iga tehnoloogia aspekte detailsemalt, et nende põhjal prototüübi realiseerimiseks lõplik valik teha.

5.1.1 MySQL

Koostatud tabeli 6 visuaalsel analüüsil on näha, et realisatsioon MySQLis oleks kõige keerulisem ja ebaefektiivsem. MySQL ei võimalda tagada hägusotsingu tuge (NF4 ja F4), lemmatiseerimist (NF6) ja otsingu sisendi eristamist otsingutulemustes (F6). Vastavat funktsionaalsust ei ole ilma täiendavate lisadeta MySQLi pakendatud. Dokumentatsioon ei kajasta ka seda, kuidas saaks lihtsalt otsingutulemuste relevantsust määrata või mis seda kõige rohkem mõjutab (NF3). Lisaks selgub, et vanemates MySQLi versioonides on täisteksti otsingu tugi toetatud ainult juhul, kui tabelite tüübiks ei ole valitud InnoDB, mis on tavapärane lähenemine andmete terviklikkuse tagamise arhitektuuri puhul. Kergendavaks asjaoluks kriitiliste puuduste hulgas on see, et lemmatiseerimine ei ole toetatud ka teiste tehnoloogiate puhul või kui, siis väikeste eranditega.

MySQLi plussidena saab välja tuua tema laia leviku, mistõttu on suur tõenäosus, et sellele tehnoloogiale tugineb ka rakenduse andmebaas. Elementaarsed otsingusüsteemi funktsionaalsed nõuded (F1, F2, F3 ja F5) saaks teostada ilma täiendavaid tehnilisi vahendeid kaasamata, mis kokkuvõttes ei suurendaks süsteemi keerukust.

5.1.2 PostgreSQL

PostgreSQL on võrreldes eelmises punktis kirjeldatud konkurendiga paremal positsioonil. Ainukese suurema miinusena saab välja tuua lemmatiseerimise puudumise (NF6). Selle alternatiiviks on PosgreSQLis olemas stemmimise tugi, kuigi dokumentatsiooni põhjal ei ole aru saada, mis keeltes see olemas on. Kollasega märgitud nõuded, mis puudutavad hägusotsingut (NF4 ja F4), on pigem välja toodud seetõttu, et

nende seadistamine on keerukas, kuid mitte võimatu. Konfigureerimist raskendab enamasti see, et kõike tuleb teha SQL päringute tasemel ilma kasutajaliideseta, mis vähendab läbipaistvust ning vajab ekspertteadmisi.

PostgreSQLi kasuks räägib tema vabavaralisus, võimalus relevantsust määrata ja heal tasemel täisteksti otsingu olemasolu, mis töötab ilma täiendavate lisadeta [9]. Kõige üllatuslikum ongi vast võimalus relevantsuse algoritmi seadistada (NF5), mida andmebaasimootorilt oodata ei oskaks. Tarkvara puhul, mille põhifunktsioon ei seisne otsinguvõimaluste pakkumises, on PostgreSQL kindlasti tugevaks kandidaadiks valikulaual, eriti kui rakenduse andmebaas on juba selle peale ehitatud.

5.1.3 ElasticSearch

ElasticSearchi puhul on üks tugevalt piiratud võimalus (F5) nõuete realiseerimisel ja mõned spetsiifilised probleemid, mis ei luba teostada nõuete maksimaalset rakendamist.

Suurimaks takistuseks on otsingutulemuste kombineeritud järjestamistingimuse seadistamise puudulikkus. Käesoleva töö konteksti arvestades ei saaks otsingutulemusi järjestada tsentraalsuse kaudu leitud kaalu ning ElasticSearchi sisemise relevantsusskoori järgi koos (F5). Võimalik on kasutada ainult ühte järjestamistingimust. Kuna äriloogilise relevantsuse kaasamine otsingusse ei ole midagi haruldast (näiteks populaarsemad või enim vaatamisi saanud tooted e-poes jms), siis sellest tulenevalt on nõude NF5 katmine tähistatud tabelis 6 „punasega“. Hägusotsingu puhul (NF4 ja F4) ei ole võimalik seadistada tolereeritavat vigade arvu sõnas koos metamärgi otsinguga (*wildcard search*) [27]. Samuti ei arvesta ElasticSearch hägusotsingu puhul tulemuste sorteerimisel sisestatud kirjavigu, mis võib relevantsuse loogika arusaamatuks teha. Otsing sõnaga „riikliku“, juhul kui on lubatud üks kirjaviga igas otsitavas sõnas, kuvaks tulemustes eespool kirjed, mis sisaldavad sõna „riiklike“, sest „e“ on eespool kui „u“ ja see taandatakse hägusotsingu puhul kirjaveaks. Lemmatiseerimise tugi on ElasticSearchis olemas kolmandate osapoolte lisana, kuid selle seadistamine on keerukas.

ElasticSearchi eeliseks kirjeldatud andmebaasimootoritega võrreldes on otsene spetsialiseerumine otsingusüsteemide tarbeks. Sellest tulenevalt on otsinguindeksi konfigureerimine võrreldes SQL tehnoloogiatega, oluliselt paindlikum, läbipaistvam ja mugavam. Kuigi käesoleva töö kontekstis ei ole otsingu kiirus primaarne, siis suure hulga

kirjete puhul on ElasticSearchil tuginev otsing oluliselt kiirem andmebaasi otsingutest, eriti kui kasutatakse täisteksti otsingut.

5.1.4 Algolia

Algolia *SaaS* teenusena pakutav otsing on mõnevõrra üllatuslikult kõige paremini püstitatud nõuetele vastav tehnoloogia. Üllatuslikult seetõttu, et kirjeldatud tehnoloogiatest on ta kõige vähem tuntud. Ainukeseks miinuseks on lemmatiseerimise toe puudumine (NF6). Kuigi Algolia dokumentatsiooni kohaselt on lemmatiseerimise asendamiseks mõeldud hägusotsingu funktsionaalsus, siis keeruka ja paljude vormidega eesti keele puhul tundub see omadus nõrga alternatiivina.

Algolia suurimaks plussiks on väga heal tasemel ja arusaadavalt dokumenteeritud relevantsusalgoritmi toimimine. Kui ElasticSearchi puhul on relevantsuse tulemuseks läbipaistmatu ujukomaarv, siis Algolia puhul on relevantsus leitud vaikimisi seitsme lihtsalt mõistetava reegliga, mis kõik annavad tulemuseks täisarvu [11], [31]:

- Kirjavigade arv sõnas (*typo*) – eelise saavad sõnad, kus ei esine kirjavigu;
- Geograafiline asukoht (*geo*) – arvestab geolokatsioonilist infot (kaugus/ vahemaad) relevantsuse määramisel;
- Jooksvate filtrite rakendamine (*filters*) – relevantsuse arvutamisel kasutatakse indeksis defineeritud fikseeritud filtrite kaale (näiteks „eesti“=2, „soome“=1 juhul, kui soovitakse alati Eestiga seotud kirjed ette poole tuua);
- Otsingule vastav sõnade arv (*words*) – mida rohkem sõnu vastab kasutaja sisestatud päringule otsingu indeksis, seda parem relevantsusskoor;
- Sõnade lähedus (*proximity*) – arvestab otsitavate sõnade kaugust üksteisest teksti sees (mitmesõnalise sisendi puhul) ja annab lähemal olevatele sõnadele parema skoori;
- Indeksi atribuutide prioriteet (*attribute*) – arvestab indeksis defineeritud atribuutide järjekorda ja lisaks saavad atribuudi alguses esinevad sõnad kõrgema skoori;

- Täpne vaste (*exact*) – otsingu sisendi täpse vaste puhul antakse samuti parem skoor relevantsusele.

Neid reegleid kasutab Algolia eraldiseisvalt relevantsuse määramisel ja tulemuste järjestamisel, mis annab eelise ühe ujukomaarvu ees nagu ElasticSearchis või PostgreSQLis. Erinevad relevantsusatribuudid võimaldavad võrdse relevantsusskoori puhul võrdlemiseks kasutada järgmise atribuudi skoori, mis vähendab võimalust, et relevantsus on võrdne ja tulemuste järjekorda ei ole võimalik määrata.

Sarnaselt ElasticSearchiga on Algolia eeliseks spetsialiseerumine otsingusüsteemidele ning kiirus. Pilveteenusena pakutav Algolia on lihtsasti skaleeritav ka juhul kui andmemahud peaks kasvama ja otsing vajab rohkem riistvaralist ressurssi. Mainimata ei saa jätta ka mugavat veebipõhist kasutajaliidest ja väga heal tasemel dokumentatsiooni.

5.2 Tehnoloogia valik ja põhjendus

Kirjeldatud tehnoloogiatest saab esimesena kõrvale jätta MySQLi, kuna püstitatud nõudeid saab sellega kõige halvemini katta. Samuti võib valikulaualt välistada ElasticSearchi. Põhjuseks relevantsusalgoritmi seadistamisega seotud probleemid. Kui otsingut ei saa konfigureerida tsentraalsuskaalu ning sisemise relevantsusega koos, siis antud töö eesmärki arvestades, ei ole sellist tehnoloogiat mõistlik kasutada.

Valikulauale alles jäänud PostgreSQL ja Algolia, on tabeli 6 lõikes suhteliselt võrdsel positsioonil. Kuigi PostgreSQLi puhul on paari nõude realiseerimine keerukam kui Algolia puhul, siis välistava põhjusena on need liiga lihtsad. Käesoleva töö mittefunktsionaalsete nõuete alapeatükis 4.3.1 toodi välja asjaolu, et kaasaegset otsingusüsteemi iseloomustab eraldatus rakenduse muust funktsionaalsusest. Selles valguses on Algolia tunduvalt eelistatum, kuna *SaaS* teenusena mõjutaks see olemasolevat rakenduse arhitektuuri minimaalselt. PostgreSQL tunduks mõistliku variandina juhul, kui rakenduse andmebaas ei ole mõnele teisele andmebaasimootorile ehitatud.

Otsingu konfigureerimine on samuti mugavam Algolias, kuna seadistamine on võimalik läbi kasutajaliidese. Algolia pakub ka API tuge PHPs, millele tugineb prototüüpimisel kasutatav veebiraamistik Symfony. Lisaks on Algolia puhul eeliseks see, et otsingu skaleeritavuse ja saadavuse eest hoolitseb teenusepakkuja.

Kokkuvõttes tundub töö autorile kõige parema variandina realiseerida prototüüp Algoliat kasutades, kuna see võimaldab püstitatud nõudeid maksimaalselt katta, tagada arhitektuurilise eraldatuse rakenduse muust funktsionaalsusest (tulevikule mõeldes), pakkuda mugavat otsinguindeksi seadistamist ning ei sea täiendavaid tehnilisi nõudeid otsingusüsteemi lokaalselt töökorda seadmisel.

5.3 Sisendandmete indekseerimine

Algolia *SaaS* teenuse kasutamiseks tuleb kõigepealt registreerida veebipõhine konto. Selle kaudu saab ligipääsu teenuse veebiliidesele ning teada API võtmed, mida läheb vaja plaanitava rakenduse avaliku poole ehitamisel.

Otsinguprototüübi ehitust tuleb alustada sisendandmete indekseerimisest. Otsinguindeksi seadistamiseks on Algolias mitu võimalust. Üks variant on seda teha läbi veebipõhise kasutajaliidese, mis on mõistlik juhul, kui indekseeritavad andmed on juba soovitud kujul JSON formaadis olemas. Teine võimalus on kasutada programmilist lähenemist. Algolia toetab kõiki enim levinud programmeerimiskeeli omapoolse API toega. Kuna käesoleva töö teoreetilise osa peatükis 3.2 sai seaduste võrgustiku analüüsiks seadistatud Symfony PHP raamistik ning andmed talletatud kohalikku andmebaasi, siis saab seda mugavalt ära kasutada andmete indekseerimiseks.

Indekseeritavateks väljadeks on prototüübi tasemel akti pealkiri, akti sisu, viide veebiaadressina aktile Riigiteatajas ning tsentraalsust iseloomustav kaal. Kolm esimest atribuuti on tekstilised ja tsentraalsuse kaal ujukomaarv, mis kombineeritud vaheloleku ning konformismianalüüsiga leitud väärtusest (vaata käesoleva töö peatükki 3.4 ja 3.5). Väljade tüübid on indeksi koostamisel olulised, kuna näiteks stringi tüüpi andmevälju ei saa kasutada sorteerimise konfigureerimisel.

Algolia seab indekseeritavatele objektidele ka ühe olulise piirangu. Objekti maksimaalne suurus koos kõigi atribuutidega võib olla kuni 10KB. Piirangu mõte on vältida suuri tekstilisi väljasid indeksis, mille relevantsuse määramine on raskendatud või annab valepositiivseid tulemusi. Kuna seadusandlike aktide tekstid võivad olla väga pikad, siis on mõistlik indekseerimisel lähtuda järgnevatest põhimõtetest:

- Kui akti sisu koosneb osadest (näiteks Võlaõigusseadus), siis indekseeritakse iga osa all olevad peatükid indeksisse eraldi kirjetena;

- Kui akti sisu ei sisalda osasid, kuid koosneb eraldi peatükkidest (näiteks Vabariigi Valitsuse seadus), siis indekseeritakse peatükid indeksisse eraldi kirjetena;

- Kui puuduvad nii osad kui peatükid (näiteks Eesti Vabariigi omandireformi aluste seadus), siis indekseeritakse kogu akti sisu tervikuna.

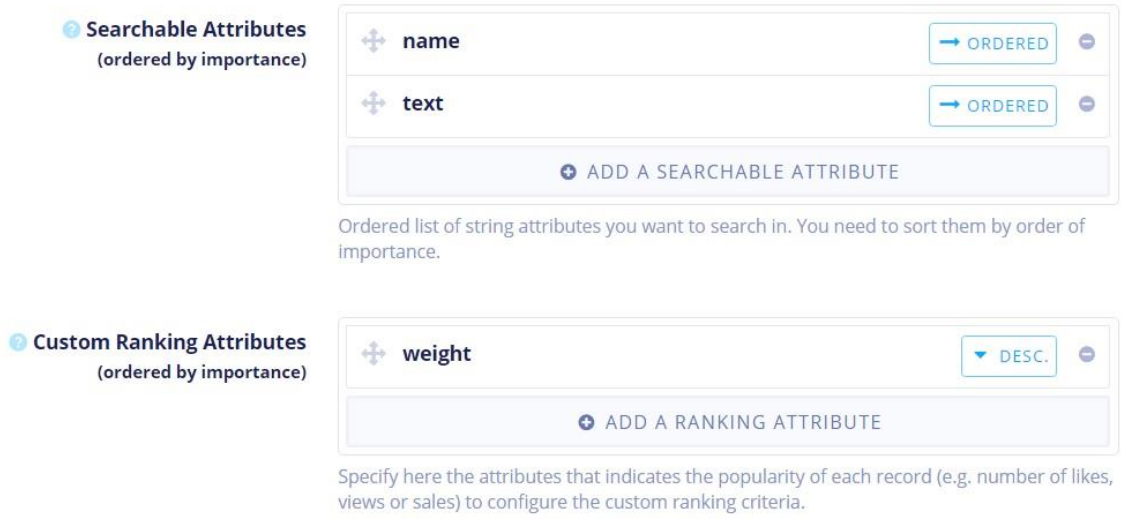
Kui eelnevaid nõudeid rakendades mõne objekti suurus ületab siiski 10KB piiri, rakendatakse täiendavalt akti teksti poolitamist ja indekseeritakse poolitatud tekstid eraldi objektidena. Selline lähenemise puhul ei tohi ära unustada, et indeksisse tekib palju duplikaate, mis erinevad ainult sisuteksti poolest. Prototüübi veebiväljundi ehitamisel tuleb seetõttu hiljem lahendada kirjete grupeerimine nii, et tulemustes ei oleks kajastatud korduvad kirjed.

Esmase indekseerimise tulemusena on Algolias 365 aktist saanud 2969 kirjega otsinguindeks. Otsingu praktiliseks kasutamiseks vajab indeks detailset konfigureerimist, mida käsitletakse järgnevas peatükis.

5.4 Otsinguindeksi seadistamine

Otsinguindeksi konfigureerimist on Algolias kõige mugavam teha läbi veebipõhise kasutajaliidese, kuigi kõik seaded on muudetavad ka vastava programmilise API kaudu. Käesoleva töö prototüübi jaoks tehakse vastav protsess selguse mõttes läbi kasutajaliidese.

Andmete indekseerimise järel tuleb esmalt seadistada otsitavad parameetrid, mille alusel hakkab otsing tulemusi filtreerima ning järjestama (relevantsust määrama). Vaikimisi ei ole ükski indekseeritud väli otsingusse kaasatud. Joonisel 11 näha olevalt seadete paneelil, on otsitavate väljade hulka määratud akti nimi ja akti tekst.

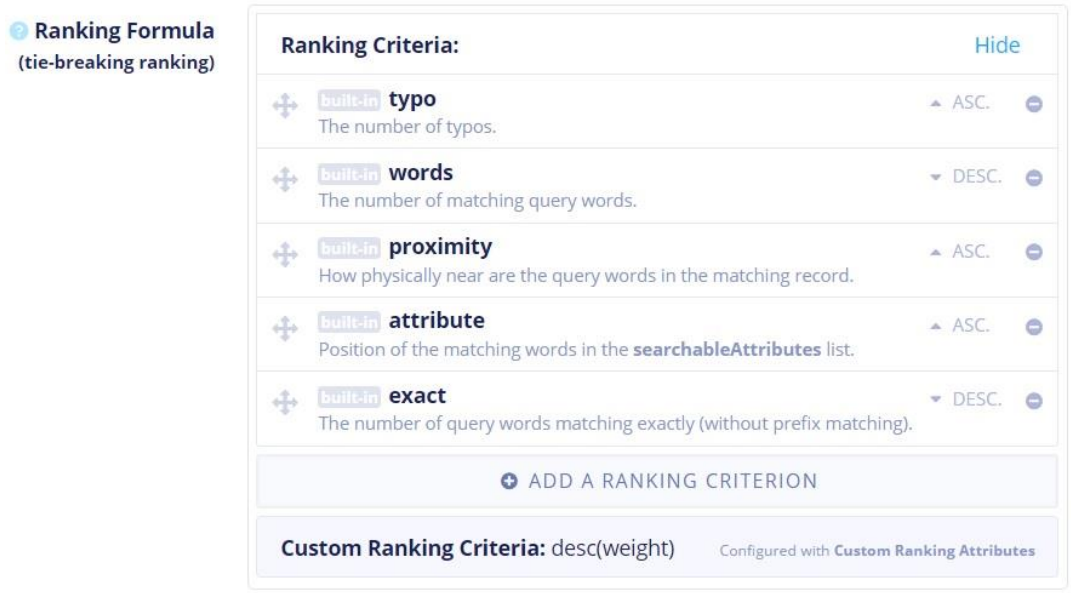


Joonis 11. Otsinguparameetrite määramine Algolias.

Tulenevalt eespool püstitatud nõuetest NF2 ja F2, on akti nimi prominentsemal kohal kui akti tekst. Seadetes määratud andmeväljade järjekord on otseselt aluseks tekstilise relevantsuse arvutamisel. Vajadusel võimaldab kasutajaliides otsitavaid välju mugavalt ringi järjestada. Mistahes muudatuse tegemisel indeksi konfiguratsioonis, ei tohi ära unustada indeksi üle salvestamist, mis käivitab reindekseerimise.

Joonise 11 alumises osas näha olev *Custom Ranking Attributes* sektsioon sisaldab äriloogiliste relevantsusparameetrite definitsioone. Käesolevas töös on ainsaks selliseks parameetriks akti tsentraalsuskaal, mille kasutamine peaks rahuldama nõuete NF5 ja F5 täitmise.

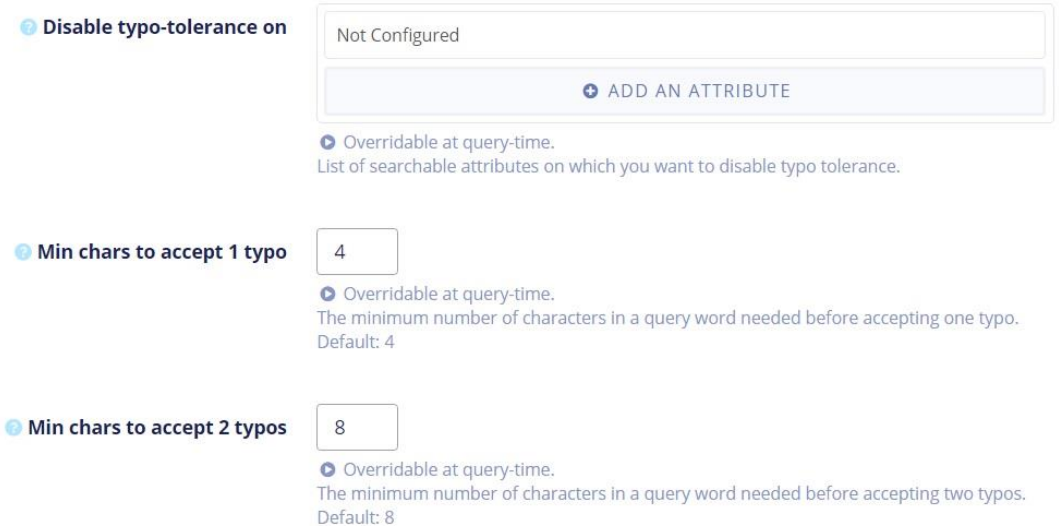
Kui indekseeritud parameetrid on seadistatud, tuleb üle vaadata relevantsusvalemi konfiguratsioon. Nagu kirjeldatud käesoleva töö peatükis 5.1.4, koosneb vastav algoritm Algolia puhul 7 reeglist, mida analüüsitakse kindlas järjekorras ülevalt alla. Käesoleva töö sisu arvestades on 7 reegli asemel kasutusel 5 reeglit. Joonisel 12 on näha vastav reeglite blokk:



Joonis 12. Relevantsusvalemi konfiguratsioon Algolias.

Konfiguratsioonist on eemaldatud geolokatsiooni ning filtri tingimusi arvestavad read, mis ehitatava prototüübi jaoks ei ole vajalikud (vaata reeglite täpsemaid selgitusi peatükk 5.1.4). Joonise allosas on näha eelmises seadete blokis defineeritud tsentraalsuskaalu parameeter, mida samuti relevantsuse arvutamisel nüüd kasutatakse.

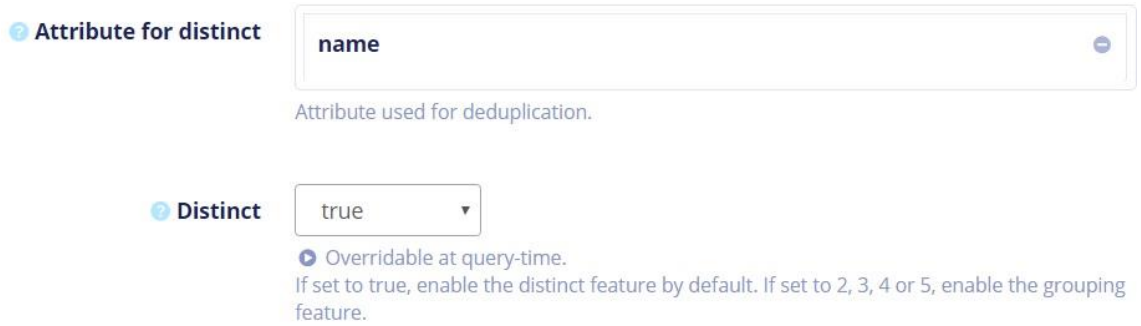
Järgmiseks tuleb konfigureerida hägusotsingu tuge pakkuvad seadistused nagu näha joonisel 13.



Joonis 13. Hägusotsingu konfiguratsioon Algolias.

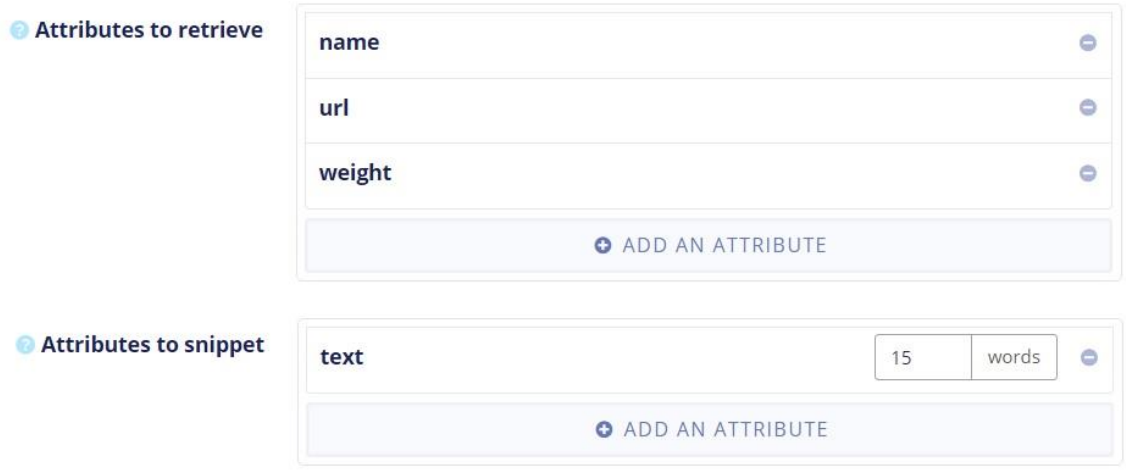
Algolia võimaldab relevantsust arvutada vaikimisi 2 kirjavea puhul sõnas, kusjuures minimaalne sõna pikkus ühe vea puhul on vaikimisi 4 tähemärki ning 2 vea puhul 8 tähemärki. Esialgu neid piiranguid muuta ei ole vaja. Vajadusel saab hägusotsingu mõne otsitava atribuudi jaoks indeksis ka välja lülitada (*disable typo-tolerance on*). Kaaluda tasub seda numbriliste atribuutide puhul, juhul, kui need on otsitavate väljade hulka lülitatud. Antud juhul indeksis numbrilisi välju otsingusse kaasatud ei ole ja vastav seade muutmist ei vaja.

Relevantsust puudutavate seadete järel tuleb konfigureerida andmete kuvamisega seotud parameetrid. Kõige olulisem nendest on grupeerimine. Tulenevalt indeksi objektile seatud mahupiirangust 10KB tükeldati aktide tekstid vastavalt peatükis 5.3 kirjeldatud reeglitele tuginedes. See tähendab, et mõne suurema akti puhul võib indeksis kirjeid olla kümneid. Duplikaatide välistamiseks otsingutulemustes saab Algolias selleks tarbeks seadistada grupeerimise parameetri nagu näha joonisel 14. Prototüübi indeksis piisab grupeerimiseks akti nimest. Sobiks ka akti veebiviide (url parameeter), mis samuti on indeksis olevate kirjete lõikes korduv.



Joonis 14. Indeksis olevate objektide grupeerimine Algolias.

Viimaseks on mõistlik ära määrata indeksi poolt tagastatavad atribuudid. Suure tõenäosusega ei vaja brauseris jooksev rakendus alati kõiki andmeid indeksist, nagu vaikimisi seadistus tagastab. Näiteks ei ole ehitatavas prototüübis oluline akti terviktekst, kuna otsingutulemustes kuvatakse ainult akti pealkiri, viide Riigiteatajasse ja otsingu tsentraalsuse kaal. Vastav seadete blokk on välja toodud joonise 15 esimesel paneelil. Selline optimeerimine võib oluliselt parandada otsingu kiiruslikke omadusi kuna andmemahud, mida on vaja serverist klientideni toimetada, vähenevad.



Joonis 15. Otsingu poolt tagastatavad objekti atribuudid Algolias.

Kuigi akti tervikteksti kuvamine otsingutulemuste lehel ei ole mõistlik, siis kasutajale on arusaadavuse mõttes siiski vaja kuvada lauset või tükki tekstist, mille põhjal talle tulemus tagastati. Seda võimaldab joonise 15 alumisel seadete paneelil näha olev konfiguratsioon (*Attributes to snippet*), mis leiab otsingu sisendile vastava relevantseima lause või teksti lõigu. Antud näites on see kuni 15 sõna pikk.

Lisaks kirjeldatud seadetele saab otsingut Algolias veel mitmeti peenhäälestada, kuid prototüübi ehitamiseks ei ole nende vaikeväärtusi esialgu mõtet muuta.

5.5 Prototüübi veebiliides

Funktsionaalsete nõuete peatükis 4.3.2 püstitatud eesmärkide täitmiseks tuleb prototüübile ehitada ka veebiliides. Selleks saab ära kasutada töö esimeses pooles seadistatud Symfony veebiraamistikku, mida kasutati ka andmete esmaseks indekseerimiseks Algoliasse.

Veebiliidese välimus on lihtne ning koosneb üheväljalisest otsinguvormist ning selle alla kuvatavast otsingutulemuste sektsioonist. Vastav ekraanipilt on välja toodud töö Lisas 1. Iga tagastatud otsingutulemuse puhul kuvatakse pealkiri, link Riigiteatajasse, sisutekstist vastena leitud relevantseim tekstiline lõik ning tsentraalsuse kaal. Juhul, kui vaste ei pärine sisutekstist, siis tekstilist lõiku ei kuvata. Otsingu sisend ja tulemustes tuvastatud vaste on rõhutatult kuvatud punasega. Veebiliides töötab kõigis enimlevinud uuemates

brauserites ning on ligipääsetav avalikult veebiaadressilt: http://tarmo.brainart.ee/legislation/web/

Veebiliidese tehniline pool taandub Symfony raamistikku pakendatud JavaScripti lahenduseks, mis saadab brauserist API päringuid Algolia *SaaS* teenuse vastu. Päringu tulemused renderdatakse samuti JavaScripti abil kohe ekraanile. Kuna päringud lähevad otse Algolia serveritesse, siis arhitektuuriliselt lokaalne veebirakendus praktiliselt mingisugust muutmist ei vaja ja otsing on sellise lahenduse puhul eraldiseisev arhitektuuriline kiht nagu soovitati käesoleva töö peatükis 4.3.1. Reaalelulises süsteemis oleks vaja vaid implementeerida andmete automaatne indekseerimine, kui neid kohalikus rakenduses muudetakse või juurde lisatakse.

5.6 Prototüübi vastavus nõuetele

Valitud tehnoloogia, selle peal seadistatud otsinguindeks ning ehitatud prototüübi veebiväljund katavad ära kõik nõuded, mis on püstitatud käesoleva töö peatükkides 4.3.1 ja 4.3.2. Ainukese puudusena ei ole kaetud mittefunktsionaalset nõuet NF6, mis pidi tagama lemmatiseerimise toe otsingule. Põhjustena saab välja tuua:

- Lemmatiseerimist ei ole Algoliasse sisse ehitatud. Võimalik oleks kasutada kolmandate osapoolte tarkvara, kuid otsingupäringute töötlemine (lemmatiseerijast läbi laskmine) kohalikus serveris aeglustaks otsingu tööd märkimisväärselt ja ei pruugiks katta enam nõude F3 (vaata ka Lisa 2) püstitatud reaalajas toimimise vajadust;

- Eesti keele lemmatiseerijate valik on piiratud ja parimad tooted on tasulised;

- Hägusotsing koos lemmatiseerimise toega vajaks detailsemat analüüsi (mis ei mahu töö skoopi), kuna võib tekitada raskesti tõlgendatavaid relevantsuse tulemusi.

Prototüübi kvalitatiivseid vastavusi nõuetele hinnatakse järgnevas peatükis, kus reaalsete sihtgruppide peal hinnatakse näidispäringute tulemusi ning võrreldakse neid Riigiteataja otsinguga.

5.7 Näidispäringud ja tulemuste analüüs

Näidispäringute tegemiseks kaasati kaks konkreetset huvigruppi, kellele käesoleva töö väljund on suunatud. Eesmärk oli saada kvalitatiivset tagasisidet oma ala ekspertidelt ja samuti potentsiaalsetelt lõppkasutajatelt. Esimeseks huvigrupiks olid juristid ja teiseks tavakodanikud, kes igapäevaselt seadustega kokku ei puutu. Lisaks koostati tsentraalsuse testimiseks komplekt päringuid kahe huvigrupi koostöös ning lõpetuseks valiti autori poolt välja päringud hägusotsingu funktsionaalsuse testimiseks. Tulemuste tõlgendamisel on lähtutud töö sisendiks olnud baasaktidest: https://www.riigiteataja.ee/lyhendid.html

5.7.1 Näidispäringud juristide huvigrupiga

Juristide huvigrupi valimi hulgas oli 4 tegevjuristi ja üks vandeadvokaat, kes said ülesandeks panna kirja kuni 10 märksõna/fraasi, mis seejärel vahetasid omanikku. Iga jurist pidi kellegi teise sisendi põhjal välja pakkuma nende hinnangul relevantseima seaduse, mis iga sõna või fraasiga seostub.

Saadud tulemusi valideeriti prototüübi ja Riigiteataja detailotsingu vastu, nagu näha tabelis 7, et võrrelda otsingute kvaliteeti tegelike ootustega. Kuna Riigiteataja detailotsing ei võimalda otsida korraga pealkirjast või sisust, siis teostati testpäring eraldi pealkirja ja sisu järgi nagu selgitatud tabeli vastava veeru päises sulgudes. Vastete puudumine on tähistatud "-" märgiga nagu ka kõigis järgnevates näidispäringute tabelites.

Tabel 7. Näidispäringute tulemused juristide sisendi põhjal.

Jurist	**Otsingu sisend**	**Oodatav vastus**	**Mitmendal positsioonil tulemustes**	
			Prototüüp	**Riigiteataja (pealkirjast / sisust)**
Jurist 1	pere	Perekonnaseadus	1	1/-
	abielu	Perekonnaseadus	2	-/-
	testament	Pärimisseadus	1	-/1
	elatis	Perekonnaseadus	10	-/3
	kahju kinnitamine	Võlaõigusseadus	16	-/2
	kindlustus	Liikluskindlustuse seadus	4	1/-
	korter	Korteriomandi seadus	2	2/1
	laen	Võlaõigusseadus	2	-/9

Jurist	**Otsingu sisend**	**Oodatav vastus**	**Mitmendal positsioonil tulemustes**	
			Prototüüp	**Riigiteataja (pealkirjast / sisust)**
Jurist 2	lepingu lõpetamine	Võlaõigusseadus	5	-/2
	trahv	Väärteomenetluse seadustik	1	-/-
	kahju hüvitamine	Võlaõigusseadus	1	-/10
	riigihange	Riigihangete seadus	1	1/-
	vallasasi	Asjaõigusseadus	2	-/4
	lepingu rikkumine	Võlaõigusseadus	1	-/2
	tariif	Elektrituruseadus	10	-/-
	elatis	Perekonnaseadus	10	-/3
	suhtluskord	Tsiviilkohtume-netluse seadus	-	-/-
	hüpoteek	Asjaõigusseadus	1	1
Jurist 3	pärimine	Pärimisseadus	1	-/1
	maks	Maksukorralduse seadus	1	8/4
	elatis	Perekonnaseadus	10	-/3
	viivis	Võlaõigusseadus	1	1
Jurist 4	kriminaalne joove	Korrakaitseseadus	-	-/-
	kaasomand	Asjaõigusseadus	6	-/3
	põlvnemine	Perekonnaseadus	3	-/7
	euroopa inimõiguste kohus	Haldusmenetluse seadustik	4	-/2
	Diskret-siooniõigus	Riigihangete seadus	-	-/-
Jurist 5	ideekonkurss	Riigihangete seadus	1	-/1
	isikut tõendav dokument	Isikut tõendavate dokumentide seadus	2	-/2
	kaubamärk	Kaubamärgi seadus	1	-/1
	eriline isikutunnus	Karistusseadustik	1	-/1
	hea usu põhimõte	Võlaõigusseadus	3	-/3

Tulemuste tõlgendamisel peab arvestama teatava subjektiivse faktoriga, kuna puudub ühtne mõõdik, sest juristide tegevusvaldkond võib olla väga spetsiifiline ning nende valikud tingitud sellest lähtuvalt. Tabelit statistiliselt analüüsides selgub, et püstitatud ootused vastasid mõnevõrra paremini prototüübi poolt pakutud vastustele. Oodatud akt oli tulemustes esimesel positsioonil 13 korral võrreldes Riigiteataja 11 korraga. Vastust ei tagastatud prototüübi puhul 2 korral ja Riigiteataja otsinguga 5 korral. Samas selgus, et Riigiteataja otsing tagastas parem tulemuse ainult 6 korral. Ülejäänud päringud olid prototüübis vähemalt samaväärsed. Lisaks on näha, et Riigiteataja pealkirja otsingu kasutamine jätab kasutaja väga tihti ilma vastusteta.

5.7.2 Näidispäringud tavakodanike huvigrupiga

Tavakodanike puhul ei saa otsingu kvaliteedi määramisel lähtuda samadest kriteeriumitest nagu juristide puhul. Inimesed ei tunne üldjuhul seaduste ametlikke nimetusi ning ei oska seetõttu öelda, milline tulemus võiks eespool olla. Oluline oleks, et otsing pakuks neile akte, mis sisaldavad otsitavat infot.

Sihtgrupis olnud 3 tavakodanikku said ülesandeks selgitada, millist teavet nad seadusandlusest leida soovivad ning panna kirja märksõnad, mida nad otsides kasutaksid. Täiendava nõudena pidi sisend olema vähemalt kahe sõnaline. Iga valimis olnud isik kirjeldas 3 olukorda. Otsingu sisend koos selgituste ja tulemustega on välja toodud tabelis 8. Riigiteataja otsingu puhul valideeriti ainult sisuteksti järgi tulemusi, kuna sisendiks olid kõigi 3 isiku puhul vähemalt kahesõnalised fraasid. Mõlema otsingu juures on välja toodud kuni 3 relevantset akti ja nende positsioonid otsingutulemustes akti nime järel sulgudes. Valik kummagi otsingusüsteemi pakutud aktide hulgast tehti esimeses huvigrupis olnud juristi abiga.

Tabel 8. Näidispäringute tulemused tavakodanike sisendi põhjal.

Isik	**Sisend**	**Selgitus otsingu kohta**	**Prototüüp**	**Riigiteataja**
Isik 1	oma firma	Eesmärgiks teada saada, millised seaduslikud piirangud on ettevõtlusega alustamiseks.	Äriseadustik (1)	Äriseadustik (1)
	ajateenistuse kohustus	Eesmärk teada saada, mis tingimustel saab	Riigikaitseseadus (3)	Kaitseväeteenis-tuse seadus (1)

Isik	Sisend	Selgitus otsingu kohta	Prototüüp	Riigiteataja
		ajateenistusest vabastust.	Kaitseväeteenistuse seadus (4)	Riigikaitseseadus (2)
	riiklikud haiguse toetused	Eesmärk teada saada, mille eest saab kroonilise haiguse puhul riikliku toetust.	Töövõimetoetuse seadus (1) Sotsiaalhoolekande seadus (4) Töötervishoiu ja tööohutuse seadus (5)	-
Isik 2	dividendide maksusta-mine	Eesmärk saada infot dividendidega saadud tulu maksustamise kohta.	Tulumaksuseadus (1) Investeerimisfon-dide seadus (2)	Tulumaksuseadus (1) Investeerimisfon-dide seadus (2)
	omavastutus liikluses	Eesmärk teada saada vastutuse määr liiklusõnnetuse puhul.	Liikluskindlustuse-seadus (1)	-
	keele eksam	Eesmärk saada infot eesti keele taseme hindamise eksami kohta.	Keeleseadus (1)	-
Isik 3	laste toetus	Eesmärk saada infot riiklike lastetoetuste kohta.	Sotsiaalhoolekande seadus (1) Perehüvitiste seadus (4)	Perehüvitiste seadus (2) Sotsiaalhoole-kande seadus (5)
	isapuhkuse saamine	Eesmärk saada infot isapuhkuse kohta.	Töölepingu seadus (1)	-
	nime andmine	Eesmärk saada teada, millised seaduslikud piirangud on lapsele nime andmisel.	Nimeseadus (1)	Nimeseadus (1)

Tabelist selgub, et kolmandikule päringutest ei suutnud Riigiteataja tekstiotsing vastet leida. See võib tuleneda asjaolust, et hägusotsingu funktsionaalsus on Riigiteataja otsingus halval tasemel või puudub täielikult. Enamike päringute puhul paranes Riigiteataja otsingu tulemus juhul, kui otsingut kahandada mingi spetsiifilise sõna peale. Näiteks „isapuhkuse saamine“ asemel andis tulemusi otsing „isapuhkus“. Sarnaselt toimis ka „omavastutus liikluses“ taandamine „omavastutus“ peale. Samas ei olnud selline lihtsustus alati lahenduseks. Püüdes otsida Riigiteatajast seadusi, mis selgitaksid toetuste saamist puude või haiguse korral, pakutakse vastusteks konteksti mitte sobituvaid akte.

Tulemust valideerinud juristi hinnangul oli 9 päringu puhul 6 vastused heal tasemel ja prototüüp pakkus võimalikest aktidest relevantsemaid esimese 5 otsingutulemuse hulgast.

5.7.3 Näidispäringud tsentraalsuse võtmes

Kuna kahes viimases peatükis tehtud päringud ei olnud mõeldud tsentraalsuse mõõdiku kaasamise kontrollimiseks, siis koostati mõlemast sihtgrupist valitud ühe inimese abiga mõned lihtsustatud elementaarpäringud selle testimiseks. Päringute koostamisel arvestati asjaoluga, et tsentraalsuskaal pääseb otsingutulemusi mõjutama kõige rohkem siis, kui otsingu sisend on väga üldine ja tekstiline relevantsus eelist ei anna. Sellisteks sisenditeks klassifitseeruvad suhteliselt lühikesed sõnad, mis esinevad mitmetes erinevates aktides ja eelistatult akti pealkirjas. Katse sisendiks valiti kirjeldatud tingimusi arvestades 10 märksõna, mis oleks mõlemale sihtgrupile mõistetavad. Iga märksõnaga seostati oodatav akt kooskõlas sihtgrupist valitud inimestega. Tulemused on näha tabelis 9, kus mõlema otsingu puhul on välja toodud kõige esimesena pakutud vaste. Riigiteataja puhul teostati seekord otsing ainult pealkirja järgi.

Tabel 9. Näidispäringute tulemused tsentraalsuse võtmes.

Sisend	**Oodatav tulemus**	**Prototüüp**	**Riigiteataja**
pere	Perekonnaseadus	Perekonnaseadus	Perekonnaseadus
abielu	Perekonnaseadus	Abieluvararegistri seadus	Abieluvararegistri seadus
töö	Töölepingu seadus	Töölepingu seadus	Eestisse lähetatud töötajate töötingimuste seadus
kuritegu	Karistusseadustik	Kriminaalmenetluse seadustik	-
eesti	Eesti Vabariigi põhiseadus	Eesti Vabariigi põhiseadus	Eesti Teaduste Akadeemia seadus
liiklus	Liiklusseadus	Liiklusseadus	Liikluskindlustuse seadus
maks	Maksukorralduse seadus	Maksukorralduse seadus	Raskeveokimaksu seadus
laps	Perekonnaseadus	Perekonnaseadus	-
majandus	Majandustegevuse seadustiku üldosa seadus	Majandustegevuse seadustiku üldosa seadus	Põllumajandusloomade aretuse seadus

Sisend	Oodatav tulemus	Prototüüp	Riigiteataja
euroopa	Euroopa Liidu kodaniku seadus	Euroopa Liidu ühise põllumajanduspoliitika rakendamise seadus	Euroopa Parlamendi valimise seadus

Katsest selgus, et juristi prognoositud akt vastas prototüübi pakutud variandile 70% juhtudest testimisel kasutatud sõnade korral. Riigiteataja otsingu puhul jäi sarnaselt tabeliga 7 silma, et pealkirja järgi otsimine on halva kvaliteediga ja pakub suure tõenäosusega variante, mida kasutaja ei otsi. Selle põhjal saab kinnitust oletus, et üldise tekstilise sisendi puhul võib tsentraalsusmõõdiku kaasamine otsingu arhitektuuri vägagi tulemuslik olla, eriti siis, kui sisendandmed sisaldaks peale baasaktide lisaks veel madalama taseme akte ja määrusi. Siiski peab teadvustama ka asjaolu, et otsitav info võib olla spetsiifiline ning üldise sisendi ja tsentraalsuse koostöös ei ole võimalik alati kasutajat rahuldada.

5.7.4 Näidispäringud hägusotsingu võtmes

Viimase testina tehti näidispäringud hägusotsingu funktsionaalsuse hindamiseks. Sisendi puhul kasutati ära juristide huvigrupi tabelit 7, kus mõlema otsingu puhul tulid oodatud seadused esimesele positsioonile. Katse eesmärk oli selgitada, kas tulemuste positsioon säilib ka kirjavea puhul.

Tabel 10. Näidispäringute tulemused hägusotsingu võtmes.

Sisend	Vigane sisend	Oodatav tulemus	Mitmendal positsioonil prototüübi tulemustes
pere	peree	Perekonnaseadus	6
testament	tesdament	Pärimisseadus	1
riigihange	riikihange	Riigihangete seadus	1
hüpoteek	hupöteek	Asjaõigusseadus	1
pärimine	parimne	Perekonnaseadus	1
viivis	vivvis	Võlaõigusseadus	1
kaubamärk	gaubamarg	Kaubamärgi seadus	1
eriline isikutunnus	erilline isikutunus	Karistusseadustik	1

Hägusotsingu päringute puhul võrdlusmomenti ei tekkinud, kuna Riigiteataja otsing hägusotsingu funktsionaalsust ei toeta ja seetõttu vastavaid tulemusi tabelis ka pole. Prototüübi puhul säilitasid 7 päringut samaväärse tulemuse ja ainult ühel juhul tulemus halvenes, kuid oli endiselt leitav. Hägusotsingu tugi annab seega süsteemile märkimisväärse lisaväärtuse.

5.8 Näidispäringute kokkuvõte

Tehtud päringute kokkuvõttena saab välja tuua järgnevad seisukohad, mis leidsid kinnitust:

- Täisteksti otsing toimib kvaliteetsemalt ja pakub otsitavaid tulemusi eespool kui rakendus võimaldab otsingut teostada üle kõigi tekstiliste atribuutide, arvestades atribuutide prioriteeti indeksis;

- Tsentraalsuse kaasamine otsingusse on mõistlik ja see pääseb kõige paremini esile üldiste otsingusõnade puhul. Isegi juhul, kui otsingu sisend ja oodatav tulemus tsentraalsuse kontekstis ei ühti. Sellisel juhul säilib suur tõenäosus, et tsentraalsuse järgi tulemuste järjestamine viib kiirema info leidmiseni;

- Hägusotsingu tugi lisab otsingule märkimisväärse lisaväärtuse, suutes otsingutulemusi leida ka väga ebatäpsete sisendite puhul.

Kuigi relevantsusel põhineva otsingusüsteemi päringuid saab alati koostada ka nii, et need välja toodud väiteid teataval määral kahtluse alla seavad, on antud tulemused siiski heaks aluspinnaks edasisteks uuringuteks

.

6 Kokkuvõte

Käesolev magistritöö jagunes võrdselt kahe suure eesmärgi vahel. Teoreetilise osaga sooviti kaardistada seadusandliku info struktuur ning määrata struktuuri moodustavatele aktidele relevantsuse kaalud, kasutades selleks võrgustike teaduse metoodikaid. Praktilise osaga plaaniti realiseerida juriidilise informatsiooni uue põlvkonna otsingumootor, mis oskaks relevantsuskaale kasutades otsingutulemusi paremini järjestada.

Töö teoreetilise osa käigus täiendati uurimise all olnud algoritmi ja leiti, et võrgustike teaduse vaheloleku mõõdikut saab sõlmpunktide tsentraalsuskaalude leidmiseks edukalt kombineerida konformismianalüüsiga. Praktilise eesmärgi saavutamiseks kaardistati kaasaegse otsingumootori tunnused, võrreldi erinevaid otsingutehnoloogiaid ning valiti välja parim püstitatud nõuete katmiseks. Valitud tehnoloogiaga realiseeriti otsingu prototüüp, mida valideeriti kahe sihtgrupi abil. Selle tulemusena saadi kinnitus, et tekstiline relevantsus koos tsentraalsuskaaludega suudab otsingutulemusi kvaliteetsemalt järjestada ja olulisemat infot eespool serveerida.

Käesoleva magistritöö peamised tulemused on:

- Uuritud algoritmi täiendamine tsentraalsuskaalude leidmiseks seaduste võrgustikku moodustavatele õigusaktidele;
- Erinevate otsingutehnoloogiate võrdlev kaardistus tehnilisest aspektist lähtuvalt;
- Tsentraalsusmõõtudel ja tekstilisel relevantsusel põhineva efektiivsema otsingumootori tehniline arhitektuur ja prototüüplahendus.

Lisaks võib töö kaaspanuseks lugeda tulemuste valideerimise ja mõõtmise katsed lõppkasutajate peal.

Töös kasutatud metoodikaid, praktilisi lahendusi ning tehtud järeldusi saab edaspidi rakendada uute otsingusüsteemide ehitamisel ja/või olemasolevate kaasajastamisel. Töö võib samuti olla sisendiks edasiste uurimuste teostamisteks vähe käsitletud juriidilise infotehnoloogia valdkonnas.

Kasutatud kirjandus

[1] Liiv, I. Vedeshin, A. Täks, E. Visualization and structure analysis of legislative acts: a case study on the law of obligations – *Proceedings of the 11th international conference on Artificial intelligence and law: 4-8th June 2007, Stanford, California*, 189-190.

[2] Võhandu, L. Some Methods to Order Objects and Variables in Data Systems (in Russian) – *Transactions of Tallinn University of Technology*. Tallinn : Tallinn Universety of Technology, 1980, (402), 43-50.

[3] When laws become too complex. [WWW] https://www.gov.uk/government/publications/when-laws-become-too-complex (10.01.2017)

[4] DB-Engines Ranking - popularity ranking of database management systems. [WWW] http://db-engines.com/en/ranking (05.01.2017)

[5] 8 best search engines for web applications as of 2017 – Slant. [WWW] https://www.slant.co/topics/95/~best-search-engines-for-web-applications (12.01.2017)

[6] MySQL :: MySQL Documentation. [WWW] https://dev.mysql.com/doc (12.01.2017)

[7] PostgreSQL: Documentation. [WWW] https://www.postgresql.org/docs (11.02.2017)

[8] Smith, L. What PostgreSQL has over other open source SQL databases. [WWW] https://www.compose.com/articles/what-postgresql-has-over-other-open-source-sql-databases (18.01.2017)

[9] Belaid, R. Postgres full-text search is Good Enough! [WWW] http://rachbelaid.com/postgres-full-text-search-is-good-enough (23.01.2017)

[10] Elasticsearch: RESTful, Distributed Search & Analytics | Elastic. [WWW] https://www.elastic.co/products/elasticsearch (12.02.2017)

[11] Algolia. [WWW] https://www.algolia.com/doc (12.02.2017)

[12] Borgatti, S. P. Everett, M. G. A Graph-theoretic perspective on centrality. – *Social Networks*. 2006, 28 (4), 466–484. [*Online*] Science Direct (20.02.2017)

[13] Borgatti, S.P. Centrality and network flow. – *Social Networks*. 2005, 27 (1), 55-71. [*Online*] Science Direct (22.02.2017)

[14] Freeman, L. C. Centrality in social networks: Conceptual clarification. – *Social Networks*. 1979, 1 (3) 215-239. [*Online*] Science Direct (04.01.2017)

[15] Hanneman, R. Introduction to social network methods Chapter 10: Centrality and power. [WWW] http://www.faculty.ucr.edu/~hanneman/nettext/C10_Centrality.html (15.02.2017)

[16] Domhoff, W. G. Centrality in networks and how it is measured. [WWW] http://www2.ucsc.edu/whorulesamerica/power_elite/centrality.html (15.02.2017)

[17] Valente, T. W. Coronges, K. Lakon, C. Costenbader, E. How correlated are network centrality measures? [WWW] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2875682/ (24.02.2017)

[18] Lee, C-Y. Correlations among centrality measures in complex networks. [WWW] https://www.researchgate.net/publication/2175985_Correlations_among_centrality_measures_in_complex_networks (01.03.2017)

[19] Iyer, S. What are the limitations of graph centrality measures? [WWW] https://www.quora.com/What-are-the-limitations-of-graph-centrality-measures (01.03.2017)

[20] Symfony, High Performance PHP Framework for Web Development. [WWW] https://symfony.com (02.03.2017)

[21] Gephi - The Open Graph Viz Platform. [WWW] https://gephi.org (02.03.2017)

[22] Lyte, A. Slater, D.M. Michel, S. Network measures of the United States Code. [WWW] https://www.mitre.org/publications/technical-papers/network-measures-of-the-united-states-code (14.03.2017)

[23] Intahchomphoo, C. Jeske, M. Landriault, E. Brown, M. Law Student Views on the Principles of a Legal Research Website: a User Experience Study. [WWW] https://doi.org/10.1017/S1472669616000384 (14.03.2017)

[24] Nelson, P. The Six Commandments of Search Implementation. [WWW] http://www.searchtechnologies.com/blog/six-search-engine-commandments (15.03.2017)

[25] Durnbull, D. What is search relevancy? [WWW] http://opensourceconnections.com/blog/2014/06/10/what-is-search-relevancy (15.03.2017)

[26] Beall, J. The Weaknesses of Full-Text Searching. – *The Journal of Academic Librarianship*. 2008, 34 (5), 438-444. [*Online*] Science Direct (17.03.2017)

[27] Full Text Search in your Database: Algolia vs Elasticsearch - Milliseconds Matter. [WWW] https://blog.algolia.com/full-text-search-in-your-database-algolia-versus-elasticsearch (18.03.2017)

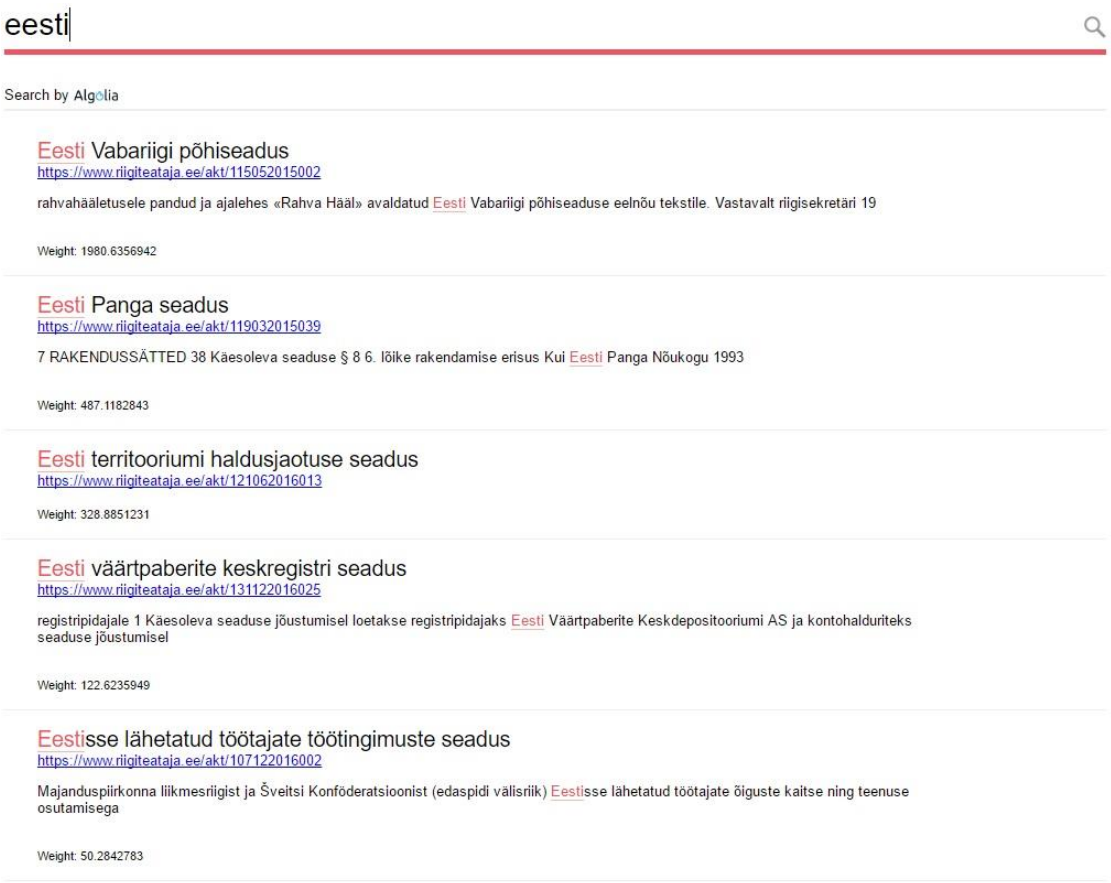
[28] Behnert, C. Lewandowski, D. Ranking Search Results in Library Information Systems - Considering Ranking Approaches Adapted From Web Search Engines – *The Journal of Academic Librarianship*. 2015, 41 (6), 725-735. [*Online*] Science Direct (19.03.2017)

[29] Petmanson, T. Ülevaade Eesti keele tarkvarast ning ressurssidest. [WWW] https://courses.cs.ut.ee/MTAT.03.277/2013_fall/uploads/Main/opinion-mining-ii.pdf (20.03.2017)

[30] Ismailov, A. Abdul Jalil, M.M. Abdullah, Z. A Comparative Study of Stemming Algorithms – *2016 3rd International Conference on Computer and Information Sciences (ICCOINS): 15-17th August 2016, Kuala Lumpur, Malaysia*, 2016, 7-12.

[31] Baissas, N. How Algolia tackled the relevance problem of search engines. [WWW] https://blog.algolia.com/how-algolia-tackled-the-relevance-problem-of-search-engines (01.04.2017)

Lisa 1 Prototüübi ekraanivaade otsinguga „Eesti“



Lisa 2 Prototüübi päringute jõudlus