



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Mindaugas Grenda

**IŠ ŠABLONŲ SUGENERUOTŲ HIPERTEKSTINIŲ DOKUMENTŲ
KLASTERIZAVIMAS PAGAL STRUKTŪRINĮ PANAŠUMĄ
CLUSTERING OF TEMPLATES BASED HYPERTEXT DOCUMENTS BY
STRUCTURAL SIMILARITY**

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa, valstybinis kodas 621E14007

Informatikos inžinerijos studijų kryptis

Vilnius, 2013

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Mindaugas Grenda

**IŠ ŠABLONŲ SUGENERUOTŲ HIPERTEKSTINIŲ DOKUMENTŲ
KLASTERIZAVIMAS PAGAL STRUKTŪRINĮ PANAŠUMĄ
CLUSTERING OF TEMPLATES BASED HYPERTEXT DOCUMENTS BY
STRUCTURAL SIMILARITY**

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa, valstybinis kodas 621E14007

Informatikos inžinerijos studijų kryptis

Vadovas

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Konsultantas

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Lietuvių kalbos konsultantas

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2013

Anotacija

Darbe pasiūlyti keturi nauji bendrų kelių metodo pagerinimai, skirti hipertekstinių dokumentų, sugeneruotų iš šablonų, klasterizavimui: atstumų skaičiavimas, remiantis XPATH, prieš tai pašalinus gretimų DOM medžio elementų indeksus iš XPATH; XPATH rinkinio dalies pašalinimas prieš skaičiuojant atstumus; XPATH kelių pradžios nukirtimas iki tam tikro gylio prieš skaičiuojant atstumus; XPATH, kurie yra bendri visiems klasterizuojamiems dokumentams, pašalinimas. Eksperimentų metu patikrinta, kad visi pasiūlyti metodai pranoksta bazinį metodą, kuriame atstumai skaičiuojami naudojant elementų XPATH. Labiausiai klasterizavimo kokybę pagerina elementų indeksų pašalinimas iš XPATH. Taip pat pasiūlytas metodas, kaip identifikuoti reikiamą kiekį klasterių, iš klasterizavimo metu gautos hierarchinės klasterių struktūros. Šis metodas klasterius identifikuoja geriau nei plačiai žinomi metodai: klasterių parinkimo pagal ribinę reikšmę ir klasterių parinkimo pagal medžio viršūnės gylį.

Darbo apimtis – 58 p. teksto be priedų, 17 pav., 8 lent., 25 bibliografiniai šaltiniai.

Atskirai pridedami darbo priedai.

Reikšminiai žodžiai: klasterizavimas, html, xpath, struktūrinis panašumas.

Annotation

There are four new improvements of common paths method proposed for clustering of templates based hypertext documents by structural similarity in the thesis. The methods include: measuring the distance using XPATH with removed sibling indexes; removing the part of the set of XPATH before measuring the distances; cutting of the beginnings of the paths to a certain depth before measuring the distances; and removing XPATHs that are common to all clustered documents. During the experiments there has been found that all proposed methods surpass the effectiveness of basic method that measures the distances by using simple XPATH. The method that increases the quality of clustering of documents the most is the removal of sibling indexes XPATH. Moreover, there is a method proposed for the identification of the required amount of clusters from the hierarchical structure obtained from clustering process. Proposed method identifies clusters more accurate than well-known methods like selection of clusters by the threshold value or selecting the clusters by the depth of the clusters in the tree.

The volume of thesis – 58 pages of text without appendixes, 17 pages of images, 8 tables, 25 references in bibliography.

Keywords: clustering, html, xpath, structural similarity.

Turinys

Paveikslų sąrašas	7
Lentelių sąrašas.....	7
Santrumpos, sąvokos	8
Įvadas.....	9
Temos aktualumas ir naujumas	9
Darbo objektas.....	10
Problema.....	10
Darbo tikslas	10
Uždaviniai.....	10
Darbo rezultatai	11
1. Literatūros apžvalga	12
1.1. Hipertekstinių dokumentų panašumas.....	13
1.1.1. RTDM – medžio koregavimo atstumas.....	14
1.1.2. CP – bendri keliai	14
1.1.3. CPS – bendros XPATH dalys.....	15
1.1.4. Dokumentų panašumas pagal URL	16
1.2. Klasterizavimas	16
1.2.1. Klasterizavimas į k klasterių.....	16
1.2.2. Hierarchinis klasterizavimas.....	16
2. Metodų pagerinimo būdai.....	17
2.1. Darbe naudojami kitų autorių metodai	17
2.1.1. Atstumo skaičiavimas.....	17
2.1.2. Klasterizavimas	18
2.1.3. Klasterių kokybės įvertinimas	18
2.1.4. Medžio perėjimas	19
2.1.5. DOM medžio elementų tipų parinkimas	20
2.1.6. Lapų XPATH.....	20
2.2. Metodų pagerinimas	21
2.2.1. XPATH be elementų indeksų.....	21
2.2.2. Bendrų dokumentų dalių pašalinimas.....	22
3. Rezultatai.....	24
3.1. Pasiruošimas	24
3.1.1. Naudojami duomenys	24
3.1.2. Pirminis klasterizavimas.....	24
3.1.3. Atstumų matrica	26
3.1.4. Atstumų matricos grafinis vaizdavimas	27
3.1.5. Dendrogramos	27
3.1.6. Gautų klasterių identifikavimas.....	27
3.2. XPATH parinkimo įtaka klasterizavimui.....	30
3.2.1. Geriausias XPATH parinkimo metodas pagal Rand indeksą.....	30
3.2.1. Geriausias XPATH parinkimo metodas pagal vidutinį grynumą.....	31
3.2.1. Geriausias XPATH parinkimo metodas pagal Dunn indeksą	33

3.3.	Bendrų dokumentų dalių pašalinimas.....	35
3.3.1.	XPATH rinkinio pradžios ir pabaigos nukirtimas.....	36
3.3.2.	XPATH pradžios nukirtimas	40
3.3.3.	Bendrų visiems dokumentams XPATH pašalinimas.....	42
3.4.	Darbe nagrinėtų metodų palyginimas	45
3.5.	Metodų taikymo pavyzdžiai	47
3.5.1.	XPATH be indeksų ir bendrų dokumentų XPATH pašalinimas	47
3.5.2.	XPATH be indeksų.....	51
3.6.	Rezultatų apibendrinimas	53
	Išvados	55
	Literatūra	56
	Priedai	59
1.	Priedas. Naudota programinė įranga	59
2.	Priedas. Naudotos atviro kodo bibliotekos	59
3.	Priedas. Eksperimentams naudota mašina.....	59

Paveikslų sąrašas

1. pav. Klasterių identifikavimas iš dendrogramos.	29
2. pav. <i>Rand</i> indekso priklausomybė nuo XPATH sudarymo būdo.	30
3. pav. Vidutinio grynumo priklausomybė nuo XPATH sudarymo būdo.	32
4. pav. <i>Dunn</i> indekso priklausomybė nuo XPATH sudarymo būdo.	33
5. pav. <i>Rand</i> indekso priklausomybė nuo XPATH sudarymo būdo.	35
6. pav. Vidutinių indeksų reikšmių priklausomybė nuo pašalinamos XPATH rinkinio dalies naudojant lapų XPATH metodą.	37
7. pav. Vidutinių indeksų reikšmių priklausomybė nuo pašalinamos XPATH rinkinio dalies naudojant XPATH be indeksų metodą.	39
8. pav. <i>Rand</i> indekso ir vidutinio grynumo priklausomybė nuo nukirstos XPATH dalies naudojant lapų XPATH metodą.	40
9. pav. <i>Rand</i> indekso ir vidutinio grynumo priklausomybė nuo nukirstos XPATH dalies naudojant XPATH be indeksų metodą.	41
10. pav. <i>Dunn</i> indekso priklausomybė nuo nukirstos XPATH dalies.	42
11. pav. Bendrų XPATH pašalinimo įtaka klasterizavimui naudojant XPATH be indeksų metodą.	43
12. pav. Bendrų XPATH pašalinimo įtaka klasterizavimui naudojant lapų XPATH metodą.	44
13. pav. Indeksai naudojant įvairius metodus.	46
14. pav. Svetainės „3“ klasterių dendrograma, gauta naudojant XPATH metodą.	48
15. pav. Svetainės „3“ klasterių dendrograma, gauta naudojant XPATH be indeksų metodą, pašalinus bendrus XPATH.	49
16. pav. Svetainės „9“ klasterių dendrograma, gauta naudojant XPATH metodą.	51
17. pav. Svetainės „9“ klasterių dendrograma, gauta naudojant XPATH be indeksų metodą.	52

Lentelių sąrašas

1 lentelė. Pirminio klasterizavimo sąlygos.	25
2 lentelė. Maksimalūs <i>Rand</i> indeksai svetainėse.	31
3 lentelė. Maksimalus vidutinis grynumas svetainėse.	32
4 lentelė. Maksimalūs <i>Dunn</i> indeksai svetainėse.	34
5 lentelė. Indeksų reikšmės taikant XPATH rinkinio dalies nukirtimo ir lapų XPATH metodus.	36
6 lentelė. Indeksų reikšmės taikant XPATH rinkinio dalies nukirtimo ir XPATH be indeksų metodus.	38
7 lentelė. Indeksų reikšmės taikant bendrų XPATH pašalinimo metodą.	43
8 lentelė. Geriausių, darbe nagrinėtų metodų indeksų reikšmės.	45

Santrumpos, sąvokos

SQL – sąryšinių duomenų bazių užklausų kalba

URL – universalusis adresas

HTML – hiperteksto ženklavimo kalba

XML – universali dokumentų ženklavimo kalba, skirta dokumentų struktūrai aprašyti

XPATH – specializuota kalba, skirta XML elementams aprašyti

DOM – dokumento objektų modelis

JVM – *Java* virtuali mašina

RDTM – hierarchinis medžių palyginimas iš viršaus į apačią

CP – bendri keliai

CPS – bendros kelių dalys

Medis – hierarchinė duomenų struktūra

Medžio viršūnė – hierarchinės duomenų struktūros mazgas, kuriame susijungia struktūros

Elementai – DOM medžio viršūnės, kurių tipas yra *Element*

Elementų atributai – DOM medžio viršūnės, kurių tipas yra *Attr*

Lapas – medžio viršūnė, kuri neturi vaikinių elementų

Kelias – XPATH, sudarytas tik iš elementų pavadinimų ir indeksų

Hipertekstas – elektroninio dokumento tekstas, kurio atskiros dalys siejamos saitais

Klasterizavimas – grupavimas, skirstymas į klasterius

Ivadas

Temos aktualumas ir naujumas

Pastaruoju metu dokumentų kiekis internete labai sparčiai didėja. Kartu auga ir poreikis iš jų automatiškai gauti reikiamą informaciją. Hipertekstiniai¹ dokumentai yra struktūrizuoti, tačiau jų struktūra gali būti labai įvairi.

Vienodą struktūrą turintys dokumentai dažnai turi tos pačios kategorijos turinį. Geras pavyzdys yra internetinės parduotuvės prekių puslapiai. Šie puslapiai yra sugeneruoti iš to paties šablono. Prekės aprašymas, kaina, nuotrauka, komentarai struktūros atžvilgiu visuose vienos parduotuvės prekių puslapiuose yra randami tose pačiose vietose. Taigi atpažinus, kad šie puslapiai yra sugeneruoti iš to paties šablono, iš jų galima lengvai išgauti norimą informaciją. Išgautos informacijos taikymo galimybės yra gana plačios. Geri pavyzdžiai yra duomenų analizės atlikimas, paieškos variklio dokumentų indeksavimas pagal tam tikras dokumento, sugeneruoto iš konkretaus šablono, dalis. Paiešką atlikusiam naudotojui gali būti parodoma tik tam tikra dokumento struktūros dalis. Taigi nustatymas, kad dokumentai yra sugeneruoti iš to paties šablono, turi didelę reikšmę internete esančių duomenų gavimui.

Nemažai darbo yra atlikta dokumentų turinio panašumui nustatyti, tačiau, kadangi hipertekstiniai dokumentai yra struktūrizuoti, struktūrinis panašumas daugeliu atvejų yra ne mažiau svarbus už turinio panašumą. Siekiant nustatyti, ar dokumentai yra panašūs ar ne, reikia apibrėžti atstumo funkciją. Pagal šią funkciją skaičiuojami atstumai tarp dokumentų. Dokumentai klasterizuojami pagal gautus atstumus. Idealiu atveju turi būti gaunami klasteriai, kur kiekviename iš jų yra dokumentai, sugeneruoti iš to paties šablono. Šiuo metu egzistuojantys hipertekstinių dokumentų struktūrinio panašumo nustatymo metodai ne visada tiksliai identifikuoja dokumentų struktūrinį panašumą.

¹ Hipertekstas – elektroninio dokumento tekstas, kurio atskiros dalys siejamos saitais. Hipertekstiniai dokumentai rašomi HTML kalba.

Išnagrinėtoje literatūroje yra aprašyti metodai, naudojami hipertekstinių dokumentų struktūriniam panašumui nustatyti. Literatūroje taip pat yra pristatytas kai kurių metodų palyginimas ir identifikuoti tiksliausiai veikiantys metodai. Šiame darbe yra pasirenkamas geriausiai veikiantis panašumų identifikavimo metodas, kuris remiasi hipertekstiniame dokumente esančių elementų XPATH². Pasirenkamas metodas yra pagerinamas keliais aspektais: pašalinami indeksai iš elementų XPATH, pasirenkami tik lapų XPATH, bandoma pašalinti dokumento dalis, kurios yra bendros visiems klasterizuojamiems dokumentams.

Darbo objektas

Hipertekstiniai dokumentai.

Problema

Darbo problema – iš šablonų sugeneruotų hipertekstinių dokumentų klasterizavimas pagal jų struktūrinį panašumą.

Darbo tikslas

Darbo tikslas – pasiūlyti iš šablonų sugeneruotų hipertekstinių dokumentų klasterizavimo pagal struktūrinį panašumą metodą, kuris skirsto dokumentus į klasterius tiksliau nei esami metodai.

Uždaviniai

Nurodytam tikslui pasiekti reikia atlikti šiuos uždavinius:

- 1) išnagrinėti susijusioje literatūroje aprašytus:
 - a) esamus panašumų tarp dokumentų skaičiavimo metodus;
 - b) esamus klasterizavimo metodus;

² XPATH – specializuota kalba, skirta XML elementams aprašyti.

- 2) pasiūlyti būdą, kaip pagerinti esamus metodus;
- 3) išanalizuoti atvirojo kodo bibliotekų panaudojimo galimybes;
- 4) įgyvendinti pasiūlytą metodą;
- 5) atlikti eksperimentus ir palyginti pasiūlyto metodo efektyvumą su esamų metodų efektyvumu.

Darbo rezultatai

Darbo metu yra gauti tokie rezultatai:

- 1) išanalizuota literatūra, susijusi su:
 - a. dokumentų klasterizavimu,
 - b. panašumo tarp XML, HTML dokumentų skaičiavimu;
- 2) išanalizuotos atviro kodo Java bibliotekos, skirtos klasterizavimui;
- 3) pasiūlyti būdai kaip galima pagerinti esamus dokumentų klasterizavimo metodus:
 - a. atstumų skaičiavimas, remiantis XPATH, prieš tai pašalinus gretimų DOM medžio elementų indeksus iš XPATH,
 - b. XPATH rinkinio dalies pašalinimas prieš skaičiuojant atstumus,
 - c. XPATH pradžios nukirtimas iki tam tikro gylio prieš skaičiuojant atstumus,
 - d. XPATH, kurie yra bendri visiems klasterizuojamiems dokumentams, pašalinimas prieš skaičiuojant atstumus;
- 4) įgyvendinti siūlomi metodai;
- 5) atlikti eksperimentai, kurių metu patikrinti ir palyginti visi siūlomi metodai, aptarti ir įvertinti bandymų rezultatai.

1. Literatūros apžvalga

Šio skyriaus pradžioje apžvelgiami įvairių autorių siūlomi hipertekstinių dokumentų klasterizavimo metodai. Plačiau aprašomi su darbu susiję dokumentų panašumo skaičiavimo ir dokumentų klasterizavimo metodai.

D. Gibsonas (Gibson *et al.* 2005) parodė, kad apie pusė visų internete randamų hipertekstinių dokumentų yra sukurti pagal šablonus. Šis skaičius augo apie šešis procentus per metus.

Šablonų radimo hipertekstiniuose dokumentuose problema pirmą kartą buvo aptarta (Yossef *et al.* 2002). Jų pasiūlytas šablonų atpažinimo metodas remiasi DOM³ medžio segmentavimu ir segmentų parinkimu.

Yra nemažai darbų, kuriuose aprašomi praktiniai šablonų aptikimo sprendimai. Toliau paminėsime keletą jų.

Šablonų aptikimo metodą sukūrė L. Blanco (Blanco *et al.* 2011). Metodas remiasi idėja, kad dokumentai, sugeneruoti naudojant šablonus, turi pasikartojančio turinio. Siekiant aptikti šį dažnai pasikartojantį turinį, įvedama turinio blokų entropijos sąvoka, kuria remiantis pašalinami pertekliniai turinio blokai.

S. Debnathas (Debnath *et al.* 2005) naudoja panašią perteklinių blokų prielaidą, tačiau atsižvelgia ne tik į turinio blokus, bet ir į kitas dokumento savybes, tokias kaip: scenarijai (angl. *scripts*), paveiksliukai ir kt.

L. Yi (Yi *et al.* 2003) nenagrinėja atskirų DOM medžio elementų. Autoriai teigia, kad dokumentas yra sugeneruotas iš šablonų, jei turi identiškai sudarytą DOM medžio šakų.

Panašus, bet kiek universalesnis sprendimas yra aprašomas D. C. Reiso (Reis *et al.* 2004). Darbe atstumo skaičiavimui tarp medžių yra naudojamas „hierarchinio medžių palyginimo iš viršaus į apačią“ (angl. *top-down tree mapping*) algoritmas. Dokumentų klasterizavimui vartojama „medžių koregavimo atstumo“ sąvoka. Tai yra veiksmų (elementų perstatymo, ištrynimo bei sukūrimo) skaičius, kuriuos reikia atlikti transformuojant vieną medį į kitą. Dokumentai skirstomi į klasterius

³ DOM – dokumento objektų modelis (angl. *Document Object Model*).

pagal medžio koregavimo atstumą. Gautuose klasteriuose esantys dokumentai yra sugeneruoti pagal tą patį šabloną.

S. Flesca (Flesca *et al.* 2002) išveda dokumentų panašumo skaičiavimo uždavinį į *Fourier* transformacijų skaičiavimo uždavinį. Toks algoritmas yra mažesnio sudėtingumo.

J. F. Cruzas (Cruz *et al.* 1998) apibūdina atstumo skaičiavimo metrikas internetiniams puslapiams. Išskiriamos kelios metrikų rūšys, paremtos elementų vektoriais, parametrizuotomis funkcijomis, medžių koregavimo atstumais.

Bendresniu atveju, lyginant XML dokumentus, medžio koregavimo atstumas yra bene geriausias, bet labai brangus laiko atžvilgiu metodas (Buttler 2004). Darbe pristatomas dokumento XPATH dalių panaudojimas skaičiuojant atstumus tarp dokumentų (dalių metodas). Dalių (angl. *shingles*) metodas yra aprašytas A. Z. Broder (Broder *et al.* 1997b). Šį metodą įprasta naudoti teksto panašumo skaičiavimui. Atstumo skaičiavimas naudojant XPATH dalis yra daug efektyvesnis, nei medžio koregavimo atstumo skaičiavimas. DOM medžių struktūros palyginimo būdas naudojant kelius yra aprašytas anksčiau (Joshi *et al.* 2003).

T. Lindholmas (Lindholm *et al.* 2006) aptaria galimybes pagreitinti XML medžių skirtumų skaičiavimus. L. Shi (Shi *et al.* 2006) pasiūlė hipertekstinių dokumentų palyginimą, kuris remiasi supaprastintu DOM medžiu. Šie autoriai naudoja standartinius klasterizavimo metodus. Gerą metodų apžvalgą ir palyginimą galima rasti B. Liu darbe (Liu 2007).

J. B. Kruskalis (Kruskal 1964) apibūdina ne metrinės daugiamatės sistemos uždavinį ir pasiūlo būdą, kaip jį išspręsti. D. Chakrabartis (Chakrabarti *et al.*) siūlo, atliekant hipertekstinių dokumentų klasterizavimą, atsižvelgti į reikšmines dokumento dalis. Siūlomas metodas į panašumo tarp dokumentų skaičiavimą įtraukia paieškos variklio užklausų istoriją.

K. Vieira (Vieira *et al.* 2006) pristato metodą, kuri aptinka internetinių puslapių šablonus nedidelėje dalyje dokumentų, o vėliau pašalina rastus šablonus iš visų dokumentų.

Žemiau plačiau aprašyti metodai, kurie bus naudojami darbe.

1.1. Hipertekstinių dokumentų panašumas

Šiame poskyryje trumpai pristatysime keletą plačiai žinomų ir naudojamų panašumo tarp hipertekstinių dokumentų skaičiavimo būdų.

1.1.1. RTDM – medžio koregavimo atstumas

Medžio koregavimo atstumo (angl. *tree edit distance*) metrikos yra pagrįstos skaičiavimu laiko, reikalingo vienos medžio struktūros transformavimui į kitą medžio struktūrą. Yra skaičiuojamos reikiamos atlikti medžio koregavimo operacijos, tokios kaip: elementų arba medžių šakų pašalinimas, pakeitimas kitomis, perstatymas ar įterpimas. Kalbant apie HTML DOM medžius, problema yra šiek tiek paprastesnė, kadangi visada yra žinomas šakninis elementas, o gretimi elementai dažnai yra vienodos medžio šakos (ypač, kai kalbama apie dokumentus, sugeneruotus pagal tą patį šabloną). Taigi dviejų medžių palyginimo problema yra supaprastinama į hierarchinį medžių palyginimą iš viršaus į apačią – RTDM (Reis *et al.* 2004). Parodoma, kad RDTM algoritmas HTML failams veikia gerai. Šis algoritmas daugelyje straipsnių pripažįstamas, kaip tinkamiausias medžių skirtumams rasti, tačiau naudojantis labai daug skaičiavimo resursų.

1.1.2. CP – bendri keliai

CP (angl. *common paths*) būdas palyginti hipertekstinių dokumentų struktūrą remiasi kelių nuo šakninio DOM medžio elemento iki kiekvieno kito medžio elemento palyginimu. Kelias yra pilnas elemento XPATH, sudarytas iš elementų pavadinimų, einant nuo pasirinkto DOM medžio elemento iki šakninio elemento. Elementų atributai šiuo atveju į XPATH nėra įtraukiami. Paprastos struktūros hipertekstinio dokumento XPATH rinkinys galėtų atrodyti taip:

```
/html  
/html/head  
/html/head/title  
/html/body  
/html/body/p  
/html/body/p[1]
```

Tokiu būdu suradus XPATH iki kiekvieno dokumento D DOM medžio elemento, yra gaunama XPATH aibė $P(D)$. Panašumas tarp dokumentų D_1 ir D_2 yra skaičiuojamas, kaip panašumas tarp aibių $P(D_1)$ ir $P(D_2)$. Panašumui tarp aibių skaičiuoti dažnai naudojama *Jaccard* funkcija:

$$J(P(D_1), P(D_2)) = \frac{|P(D_1) \cap P(D_2)|}{|P(D_1) \cup P(D_2)|} \quad (1)$$

Jaccard funkcija yra aprašyta P. Jaccardo (Jaccard 1901). T. Gottronas (Gottron 2008) pateikia alternatyvų skaičiavimo būdą:

$$panašumas(D_1, D_2) = \frac{|P(D_1) \cap P(D_2)|}{\max(|P(D_1)|, |P(D_2)|)} \quad (2)$$

XPATH radimo dokumente laikas tiesiškai priklauso nuo elementų skaičiaus dokumentuose D_1 ir D_2 .

1.1.3. CPS – bendros XPATH dalys

Bendros XPATH dalys (angl. *common path shingles*) yra CP metodo kombinacija su „dalių“ metodu, kuris yra pristatytas D. Buttlerio (Buttler 2004). Pagrindinė XPATH dalių metodo idėja yra lyginti ne pilnus XPATH, o suskaidyti juos į mažesnes, vienodo ilgio dalis, ir lyginti būtent tas dalis. Šio būdo privalumas tas, kad jei du XPATH yra labai panašūs ir skiriasi tik maža dalimi, tai šiuo metodu gaunamas nemažas panašumas tarp jų. Tuo tarpu, taikant CP metodą, XPATH panašumas yra lygus nuliui, jei jie nėra identiški.

Tarkime, kad $PS(D_1)$ ir $PS(D_2)$ yra atitinkamai dokumentų D_1 ir D_2 XPATH dalių aibės. Tuomet panašumas tarp šių dokumentų gali būti skaičiuojamas kaip panašumas tarp aibių $PS(D_1)$ ir $PS(D_2)$

$$panašumas(D_1, D_2) = \frac{|PS(D_1) \cap PS(D_2)|}{\max(|PS(D_1)|, |PS(D_2)|)} \quad (3)$$

A. Z. Broderis (Broder 1997a) parodė, kad, skaičiuojant dokumentų panašumą, galima gauti pakankamai gerą rezultatą, jei taikomas dalių metodas pasirenkant iš dalių rinkinių atsitiktinius elementus.

1.1.4. Dokumentų panašumas pagal URL⁴

(Blanco *et al.* 2011) pateikia alternatyvų dokumentų panašumo skaičiavimo ir klasterizavimo būdą. Šis metodas remiasi idėja, kad dokumentai, sugeneruoti iš to paties šablono dažnai turi ne tik panašią vidinę struktūrą, bet ir panašią URL struktūrą. Metodo pagrindinis privalumas yra greitis. Skaičiuojant panašumą tik pagal URL, nereikia skaityti ir analizuoti dokumento turinio, todėl per tą patį laiką galima apdoroti daug didesnius kiekius dokumentų, nei kitais metodais. Pagrindinis metodo trūkumas yra tikslumas. Ne visais atvejais apie dokumento struktūrą galima spręsti vien tik pagal URL, todėl metodas efektyviai veikia tik su tais dokumentais, kurių URL gerai atspindi struktūrą.

1.2. Klasterizavimas

Šiame skyriuje trumpai apžvelgiami klasterizavimo metodai.

1.2.1. Klasterizavimas į k klasterių

Klasterizavimas į k klasterių yra klasikinis klasterizavimo metodas, pirmą kartą pristatytas (MacQueen 1967). Pagrindinė metodo idėja yra kiekvieną dokumentą atsitiktine tvarka priskirti į vieną iš k klasterių. Toliau pradedamas klasterizavimo procesas. Kiekvienam klasteriui yra suskaičiuojamas centrinis dokumentas – dokumentas, kurio atstumų suma iki kitų klasterio dokumentų yra mažiausia. Dokumentai priskiriami tam klasteriui, iki kurio centrinio dokumento atstumas yra mažiausias. Atlikus šį priskyrimą, kiekvienam klasteriui iš naujo suskaičiuojamas centrinis dokumentas. Toliau procesas vėl kartojamas. Procesas nustojamas kartoti, kai klasterių konfigūracija nustoja keistis arba kai pasikeitimai tampa minimalūs.

1.2.2. Hierarchinis klasterizavimas

Hierarchinis klasterizavimas, priešingai nei aprašytas klasterizavimas į k klasterių, nereikalauja žinoti klasterių skaičiaus iš anksto. Hierarchinis klasterizavimas gali būti dviejų rūšių:

⁴ URL – universalusis adresas (angl. *Uniform Resource Locator*).

sujungiamasis ir dalijamasis. Dalijamasis klasterizavimas yra pradedamas nuo vieno klasterio, kuris toliau dalijamas į smulkesnius klasterius. Šiame darbe naudosime tik sujungiamąjį klasterizavimą. Sujungiamojo klasterizavimo atveju, kiekvienas dokumentas yra priskiriamas į atskirą klasterį. Toliau klasteriai pagal panašumą yra jungiami į aukštesnio lygio klasterius. Procesas kartojamas, kol sujungiami visi klasteriai. Taip sujungti klasteriai gali būti pavaizduoti medžio diagrama – dendrograma.

Yra keletas būdų (jungimo kriterijų), nuspręsti, kurie klasteriai turi būti jungiami į vieną. Pavyzdžiui dažnai naudojamas vieno jungimo kriterijus. Šiuo atveju, jungiami tie du klasteriai, tarp kurių atstumas yra mažiausias iš visų atstumų tarp klasterių.

2. Metodų pagerinimo būdai

Šiame skyriuje pirmiausia aprašomi darbe naudojami kitų autorių metodai. Toliau yra pasiūlomi kai kurių metodų galimi pagerinimo būdai.

2.1. Darbe naudojami kitų autorių metodai

Šiame poskyryje aprašoma, kokie egzistuojantys metodai yra pasirinkti šiam darbui atlikti bei pagrindžiamas jų pasirinkimas.

2.1.1. Atstumo skaičiavimas

T. Gottronas (Gottron 2008) ir D. Buttleris (Buttler 2004) pateikia įvairių atstumo skaičiavimo metodų eksperimentinį palyginimą. Eksperimentuose parodoma, kad medžio koregavimo atstumo metodas reikalauja žymiai daugiau resursų nei kiti metodai, todėl, nors teoriškai šis metodas yra geras, jo taikymas praktikoje yra sunkiai įmanomas esant dideliems duomenų kiekiams. Pagal D. Buttlerį (Buttler 2004) didžiausias klasterizavimo tikslumas gaunamas naudojant bendrą XPATH ir bendrą XPATH dalių atstumo skaičiavimo metodus. Taigi toliau darbe remsimės šiais metodais.

Atstumas tarp dokumentų D_1 ir D_2 yra skaičiuojamas kaip vienetas minus *Jaccard* panašumas tarp aibių $P(D_1)$ ir $P(D_2)$.

$$atstumas = 1 - J(P(D_1), P(D_2)) \quad (4)$$

Čia $P(D_1)$ ir $P(D_2)$ yra, atitinkamai, dokumentų D_1 ir D_2 , XPATH ir XPATH dalių aibės. $J(P(D_1), P(D_2))$ yra *Jaccard* panašumas, apibrėžtas lygybe (1).

2.1.2. Klasterizavimas

Pastebima, kad hierarchinis klasterizavimas veikia tiksliau, nei klasterizavimas į k klasterių (Gotttron 2008), todėl, nors šis metodas yra brangesnis skaičiavimo apimties atžvilgiu, šiame darbe naudojamas būtent šis metodas. Be to hierarchinius klasterius galima patogiai pavaizduoti grafiškai – dendrogramomis, o tai palengvina analizę.

2.1.3. Klasterių kokybės įvertinimas

Šiame poskyryje trumpai aprašomos metrikos, naudojamos klasterių kokybei įvertinti.

2.1.3.1. Dunn indeksas

Dunn indeksas palygina didžiausią atstumą d_{max} , tarp dokumentų, priskirtų tam pačiam klasteriui su mažiausiu atstumu d_{min} , tarp dokumentų, priskirtų skirtingiems klasteriams

$$D = \frac{d_{max}}{d_{min}} \quad (5)$$

Taigi šis indeksas parodo, kaip gerai atstumo skaičiavimo metrika atskiria dokumentus, sugeneruotus iš vieno šablono, nuo dokumentų, sugeneruotų iš kitų šablonų. Kuo didesnis *Dunn* indeksas, tuo geriau atstumo metrika atskiria dokumentus.

2.1.3.2. Rand indeksas

Rand indeksas pirmą kartą aprašytas W. M. Rando (Rand 1971). Šis indeksas remiasi „teisingu“ dokumentų suskirstymu į klasterius, kuris turi būti žinomas iš anksto, ir palygina jį su gautu dokumentų suskirstymu į klasterius. *Rand* indeksas parodo, kaip dažnai gauta klasterių sąranka (angl. *configuration*) sutampa su „teisinga“, iš anksto apsibrėžta sąranka.

$$R = \frac{A}{A + D} \quad (6)$$

Čia A yra skaičius dokumentų, kurie, remiantis „teisingu“ klasterizavimu, yra suklasterizuoti teisingai. D yra skaičius dokumentų, kurie pagal „teisingą“ klasterizavimą yra suklasterizuoti neteisingai.

Taigi šis indeksas parodo, kaip dažnai gauta klasterių sąranka sutampa su „teisinga“ iš anksto apibrėžta sąranka. Kuo didesnis *Rand* indeksas, tuo dažniau gautas klasterizavimas sutampa su „teisingu“, taigi tuo geresnė klasterizavimo kokybė.

2.1.3.3. Klasterių grynumas

Šis indeksas taip pat remiasi „teisingu“ dokumentų suskirstymu į klasterius, kuris turi būti žinomas iš anksto, ir palygina jį su gauto dokumentų suskirstymu į klasterius. Grynumas parodo, kiek dokumentų iš gauto klasterio patenka į vieną klasterį remiantis „teisinga“, iš anksto apibrėžta klasterių konfigūracija. Tarkime, turime klasterį c_k . $n_k^{(i)}$ yra dokumentų skaičius jame, kurie pagal „teisingą“ klasterizavimą turėtų priklausyti klasteriui i . Vieno klasterio c_k grynumas apskaičiuojamas taip:

$$P(c_k) = \frac{1}{\sum_i n_k^{(i)}} \cdot \max_i n_k^{(i)} \quad (7)$$

Norint įvertinti klasterių rinkinio, kurį sudaro m klasterių, $C = \{c_k : k = 1..m\}$, grynumą, yra skaičiuojamas visų klasterių grynumų vidurkis:

$$P(C) = \frac{1}{m} \cdot \sum_{k=1}^m P(c_k) \quad (8)$$

Taigi klasterių grynumas parodo, kiek dokumentų iš gauto klasterio turi priklausyti vienam klasteriui remiantis „teisinga“, iš anksto apibrėžta klasterių konfigūracija. Kuo didesnis grynumas, tuo mažiau gautuose klasteriuose neteisingai priskirtų dokumentų.

2.1.4. Medžio perėjimas

Visų hierarchinės struktūros elementų perėjimą galima įgyvendinti dviem būdais: *gylis pirmiau* (angl. *depth-first*), *plotis pirmiau* (angl. *breadth-first*). Blogiausiu atveju abu algoritmai turi tokį patį sudėtingumą. Šiame darbe medžio perėjimas naudojamas tik XPATH sudarymui, todėl nėra esminio skirtumo, kurį algoritmą geriau naudoti. Pasirenkamas naudojamas metodas *gylis pirmiau*, nes šis metodas yra įgyvendintas naudojamoje bibliotekoje *Jsoup*, kuri yra skirta darbui su HTML failais.

Klasterių medžio perėjimui taip pat naudojamas algoritmas *gylis pirmiau*.

2.1.5. DOM medžio elementų tipų parinkimas

Remiantis žiniatinklio konsorciumo (W3C) standartu (HTML 5.1), DOM medį gali sudaryti tokių tipų viršūnės:

Document;

DocumentFragment;

DocumentType;

ProcessingInstruction;

EntityReference;

Element;

Attr;

Text;

CDATASection;

Comment;

Entity;

Notation;

Sudarant viršūnių XPATH, darbe yra naudojamos tik *Element* tipo viršūnės. Kitų tipų viršūnės šiame darbe nėra aktualios, nors kitais būdais skaičiuojant skirtumus tarp dokumentų gali būti naudinga ir kitų viršūnių informacija.

2.1.6. Lapų XPATH

Šis metodas yra aprašytas D. Buttlerio (Buttler 2004). Autorius aprašo bendrų kelių metodą. Bendrų kelių metodui taikyti, D. Buttler pasirenka lapų XPATH, tačiau XPATH pasirinkimo plačiau nenagrinėja. Šiame darbe plačiau panagrinėsime XPATH pasirinkimo įtaką klasterizavimo kokybei.

Didelė dalis iš hipertekstinio dokumento gautų XPATH yra kitų XPATH sudedamoji dalis. Jei skaičiuojant atstumą tarp puslapių, imamas kiekvieno DOM medžio elemento XPATH, tai dokumentų tarpusavio panašumas gali dirbtinai padidėti. Kiek padidės panašumas, priklauso nuo DOM medžio gylio.

Panagrinėkime pavyzdį. Turime medžio elementą *a*, kuris žymi nuorodą į kitą puslapį. Šio elemento XPATH be indeksų yra:

`/html/body/header/div/div/a`

Tačiau skaičiuojant kiekvieno medžio elemento XPATH yra gaunamas visas sąrašas XPATH, kurie rodo kelią iki kurio nors aukščiau nei nagrinėjamas *a* esančio elemento:

```
/html  
/html/body  
/html/body/header  
/html/body/header/div  
/html/body/header/div/div  
/html/body/header/div/div/a
```

Struktūriškai šie elementai egzistuoja DOM medyje, tačiau jie turi vaikinių elementų. Pašalinus iš sąrašo elementų, kurie turi vaikinių elementų, XPATH, lieka tik tie XPATH, kurie rodo mums svarbiausius elementus (tuos, kurie atspindi tikrą, dažnai naudotojui matomą, puslapio struktūrą). Be to, puslapiai yra geriau atskiriami vienas nuo kito, nes nelieta elementų, kurie dažnai pasikartoja visuose dokumentuose: /html, /html/body ir pan.

2.2. Metodų pagerinimas

Šiame poskyryje aprašomi sugalvoti būdai kitų autorių metodams pagerinti.

2.2.1. *XPATH be elementų indeksų*

Hipertekstiniuose dokumentuose yra daug pasikartojančių struktūrinių dalių. Tarkime, turime internetinės parduotuvės prekės puslapį skirtingu laiko momentu. Struktūriškai puslapis yra beveik toks pat, tačiau einant laikui jame atsirado keli nauji atsiliepimai apie prekę. Taigi atsirado rinkinys naujų XPATH, kurie skiriasi nuo šalia esančių XPATH tik indeksais. Lyginant XPATH, gautus iš šių puslapių, atsiranda skirtumas, nes kiekvienas atsiliepimas apie prekę prideda naujų XPATH.

Pavyzdžiui, žemiau išvardinta XPATH aibė aprašo hipertekstinio dokumento lentelės eilutes:

```
/html/body/table/tr  
/html/body/table/tr[1]  
/html/body/table/tr[2].
```

Jei į lentelę pridedama nauja eilutė, atsiranda XPATH:

```
/html/body/table/tr[3],
```

nors šie puslapiai turi tokią pačią struktūrą.

Šis pastebėjimas motyvuoja pašalinti elementų indeksus iš XPATH. Pašalinus indeksus iš XPATH, iš aukščiau išvardytų kelių lieka tik vienas:

```
/html/body/table/tr
```

Taigi šiuo pavyzdžiu parodoma, kad iš XPATH pašalinus indeksus ir lyginant puslapius, struktūrinių skirtumų neberandama. Bendresniu atveju hipertekstiniuose dokumentuose gali būti daug pasikartojančių struktūrinių dalių, kurios skiriasi tik XPATH indeksais. Šiuos indeksus pašalinus, atstumas tarp dokumentų, sugeneruotų iš to paties šablono, turėtų sumažėti.

Indeksų pašalinimo metodas taip pat turi ir neigiamą įtaką panašumo skaičiavimui. Puslapiuose gali būti elementų, kurių XPATH be indeksų sutampa, tačiau struktūriškai jie yra skirtingose vietose. Tokiu būdu šiek tiek sumažėja metodo tikslumas.

2.2.2. Bendrų dokumentų dalių pašalinimas

Klasterizuojant dokumentus, kurie priklauso vienai interneto svetainei, susiduriama su problema, kad visi puslapiai, tiek sugeneruoti iš vieno šablono, tiek sugeneruoti iš skirtingų šablonų, turi daug bendros informacijos. Pavyzdžiui, dauguma dokumentų iš vienos internetinės svetainės gali turėti tokį patį šoninį ar viršutinį meniu. Apatinė dokumento dalis dažnai būna bendra visiems puslapiams.

Siekiant padidinti atstumą tarp skirtingų klasterių, reikia pašalinti tas dokumento dalis, kurios yra bendros visiems klasterizuojamiems dokumentams. Toliau aprašoma keletas metodų, kurie turėtų sumažinti bendrų struktūrinių dalių kiekį dokumentuose.

2.2.2.1.XPATH rinkinio pradžios ir pabaigos nukirtimas

Analizuojant hipertekstinius dokumentus, priklausančius vienai internetinei svetainei, yra nesunku pastebėti, kad visi dokumentai turi bendrų dalių. Bendros dalys dažniausiai būna šios: meniu juostos, antraštės, esančios viršuje, kontaktų juosta, esanti apačioje. Todėl dauguma dokumento struktūros dalių, bendrų visiems tos pačios svetainės dokumentams, turėtų būti dokumentų pradžioje ir pabaigoje.

Taigi jei prieš skaičiuojant atstumus tarp dokumentų, iš kiekvieno dokumento XPATH rinkinio pašalinsime rinkinio pradžią ir pabaigą, tai galime gauti klasterius, kurie yra geriau atskirti

vienas nuo kito. Norint gauti geriausią rezultatą, reikia tinkamai parinkti, kokia dalis XPATH rinkinio turi būti pašalinama.

2.2.2.2.XPATH pradžios nukirtimas

Atidžiau panagrinėjus hipertekstinių dokumentų struktūrą matyti, kad pagrindinė dokumento dalis (gauta atmetus pasikartojančius elementus: meniu juostą, apačioje esančią kontaktų juostą ir kt.) gali būti gaunama imant ne vidurinę XPATH rinkinio dalį, o nukirtus XPATH iki tam tikro lygio.

Pavyzdžiui, turime tokius XPATH:

```
/html  
/html/body  
/html/body/header  
/html/body/header/div  
/html/body/header/div/div  
/html/body/header/div/div/a
```

Tikėtina, kad pirmieji trys keliai yra randami visuose internetinės svetainės puslapiuose. - Todėl nukertame kiekvieną XPATH iki trečio lygio ir gauname tokį XPATH rinkinį:

```
/div  
/div/div  
/div/div/a
```

Labiau tikėtina, kad šis XPATH rinkinys bus randamas ne visuose svetainės puslapiuose, o tik sugeneruotuose iš vieno šablono.

2.2.2.3.Bendrų visiems dokumentams XPATH pašalinimas

Siekiant pašalinti bendras dalis iš dokumentų, galima tiesiog pašalinti visus bendrus XPATH, kurie priklauso visiems dokumentams iš vienos svetainės.

Tarkime turime n dokumentų. D_i , $i = 1..n$. $P(D_i)$ yra i –tojo dokumento XPATH aibė. Tuomet bendri XPATH gali būti randami taip:

$$P_c = \bigcap_{i=1}^n P(D_i) \quad (9)$$

Atstumas skaičiuojamas iš kiekvieno dokumento XPATH aibės pašalinus XPATH, kurie yra bendri visiems dokumentams:

$$atstumas = 1 - J(P(D_1)/P_c, P(D_2)/P_c) \quad (10)$$

Tokiu būdu suskaičiavus atstumą, dokumentai, sugeneruoti iš skirtingų šablonų, yra geriau atskiriami nuo dokumentų, sugeneruotų iš kitų šablonų. Be to, privalumas yra toks, kad nereikia euristiniu būdu nustatytų parametrų, tokių kaip nukertant dalį XPATH rinkinio arba dalį XPATH.

3. Rezultatai

Šiame skyriuje aprašomi eksperimentai, kurių metu buvo tikrinami pasiūlyti nauji metodai.

3.1. Pasiruošimas

3.1.1. Naudojami duomenys

Eksperimentams atlikti reikalingiems duomenims surinkti buvo panaudotas atviro kodo naršymo roboto karkasas *crawler4j*. Šis naršymo robotas buvo pakeistas taip, kad parsisiunčiamus HTML failus saugotų nurodytame kataloge, o papildomus duomenis apie failus saugotų *MySQL* duomenų bazėje. Duomenų bazėje saugomas kiekvieno failo pavadinimas, URL adresas, identifikacinis numeris ir svetainės identifikacinis numeris.

Automatizuotai naršant internetinius puslapius, svarbu laikytis mandagumo. Naršymo robotas turi siųsti ribotą kiekį užklausų į vieną serverį per laiko vienetą. Dar vienas svarbus dalykas automatiniam naršymui yra *robots.txt* failas. *Robots.txt* failas yra žiniatinklio serverio šakniniame kataloge. Serverio administratoriai šiame faile gali nurodyti, kokius puslapius leidžiama lankyti robotams, o kokių ne. Siunčiant failus šiam darbui, buvo laikomasi mandagumo taisyklių ir *robots.txt* faile nurodytų taisyklių.

Duomenys buvo surinkti iš kelių skirtingų interneto svetainių. Tolimesniame poskyryje pateikiamas sąrašas interneto svetainių, kurių dokumentai buvo klasterizuojami eksperimentuose.

3.1.2. Pirminis klasterizavimas

Siekiant įvertinti eksperimentinių klasterizavimo metodų efektyvumą, reikia turėti išankstinį, „teisingą“, klasterizavimą, su kuriuo bus lyginami gauti klasteriai. Pirminį klasterizavimą atliekame rankiniu būdu, remiantis dokumentų URL adresais. Dokumentams klasterizuoti apsibrėžiame

taisykles, išvardintas 1 lentelėje. Eksperimentų metu buvo klasterizuojama po tris šimtus dokumentų kiekvienoje interneto svetainėje.

1 lentelė. Pirminio klasterizavimo sąlygos pateikiamos pirminio klasterizavimo sąlygos.

1 lentelė. Pirminio klasterizavimo sąlygos.

Svetainės id	Svetainė	SQL sąlyga
4	azon.lt	url LIKE '%&addcompare=1%'
4	azon.lt	url LIKE '%.html'
2	bigbox.lt	url REGEXP '.html'
2	bigbox.lt	url REGEXP '&aid=[a-z,0-9,&=]{1,}pgNr='
2	bigbox.lt	url REGEXP 'searchparam=&cnid=[a-f,0-9]{1,}'
9	citylights.com	url REGEXP '(/book/?GCOI=[0-9]+\$) &fa=description'
9	citylights.com	url REGEXP '/catalog/?category_id=[0-9]+\$'
9	citylights.com	url REGEXP 'citylights.com/?fa=event&event_id=[0-9]+\$'
9	citylights.com	url REGEXP '&fa=details'
9	citylights.com	url REGEXP '?GCOI=[0-9]+&fa=recommend'
9	citylights.com	url REGEXP '/printable/?GCOI=[0-9]+\$'
3	elshop.lt	url LIKE '%/prekiu-katalogas/%'
3	elshop.lt	url LIKE '%/preke/%'
3	elshop.lt	url LIKE '%/nusiusti-draugui/%'
3	elshop.lt	url LIKE '%/atsiliepimas/%'
7	ilterzogirone.it	url REGEXP '/informazioni.php'
7	ilterzogirone.it	url REGEXP '/minisito/scheda_home.php'
7	ilterzogirone.it	url REGEXP '/vetrina.php'

Svetainės id	Svetainė	SQL sąlyga
7	ilterzogirone.it	url REGEXP '/cerca_bedandbreakfast.php/cerca_hotel_alberghi.php/cerca_agenzie_di_viaggio.php/cerca_rist.php'
7	ilterzogirone.it	url REGEXP '/card.php'
8	iristorante.it	url REGEXP 'r[0-9]+.html\$'
8	iristorante.it	url REGEXP '_lista_l[0-9]+.html'
8	iristorante.it	url REGEXP '/utenti_ristoranti.html'
8	iristorante.it	url REGEXP '_t[0-9]+c[0-9]+.html\$'
8	iristorante.it	url REGEXP 'iristorante.it/ristorante/'
8	iristorante.it	url REGEXP '/ricerca_ristoranti_*.html'
8	iristorante.it	url REGEXP '/profilo_utente_*.html'
5	varle.lt	url REGEXP '.html'
5	varle.lt	url REGEXP 'varle.lt/[a-z,-]{1,}/(\$)(\?p=[0-9]{1,})(\?tipas=)(\?gamintojas=)'
5	varle.lt	url REGEXP '/lesson/'
5	varle.lt	url REGEXP '/cart/add-item/'

3.1.3. Atstumų matrica

Klasterizavimas yra atliekamas remiantis atstumų matrica, kuri suskaičiuojama prieš klasterizuojant. Atstumų matricoje eilutės ir stulpeliai atspindi dokumentus, o langeliuose įrašomi atstumai, tarp atitinkamų stulpelyje ir eilutėje esančių dokumentų. Atstumų matrica yra simetrinė. Pagrindinės įstrižinės elementai yra lygūs nuliui, kadangi šie elementai yra atstumai tarp tų pačių dokumentų. Taigi skaičiuojant atstumų matricą, užtenka suskaičiuoti pusę šios matricos be pagrindinės įstrižinės. Jei matrica yra $n \times n$ dimensių, tai reikia atlikti $\frac{n^2-n}{2}$ atstumo skaičiavimų.

3.1.4. Atstumų matricos grafinis vaizdavimas

Atstumų matricą patogų pavaizduoti grafiškai. Kiekvienas matricos elementas grafiškai pavaizduojamas kvadratinio langeliu. Reikšmė tame langelyje pavaizduojama spalva. Atstumų matricoje reikšmės gali kisti nuo nulio iki vieneto, todėl nuliui patogų priskirti baltą spalvą, o vienetui – juodą spalvą. Visoms tarpinėms reikšmėms priskiriamos tolygiai šviesėjančios spalvos, kai einama nuo vieneto iki nulio.

3.1.5. Dendrogramos

Dendrograma yra medžio diagrama, dažnai naudojama iliustruoti hierarchinio klasterizavimo metu gautą medžio struktūrą. Darbe naudojamos dendrogramos yra sugeneruotos naudojant atviro kodo programą *MultiDendrograms* ir atviro kodo *Java* bibliotekoje *hierarchical-clustering* esančią klasę. Taip pat buvo sukurtas šios bibliotekos plėtinys, kuris išsaugo dendrogramą rinkmenoje.

Dendrogramos suteikia pirminę informaciją apie klasterizavimą, be to, pagal dendrogramas galima gerai parinkti klasterių identifikavimo ribinę reikšmę arba gylį (žiūrėti poskyrį 3.1.6. Gautų klasterių identifikavimas).

3.1.6. Gautų klasterių identifikavimas

Suklasterizavus turimus dokumentus, gaunama hierarchinė struktūra – klasterių medis. Iš gauto medžio reikia pasirinkti, kurios medžio viršūnės yra ieškomi klasteriai. Tai galima padaryti dviem būdais: pagal ribinę atstumo reikšmę arba pagal klasterio gylį.

Identifikuojant klasterius pagal ribinę atstumo reikšmę, tikrinama sąlyga: ar klasterio atstumas iki kito klasterio yra mažesnis arba lygus ribinei reikšmei, ir ar tėvinio klasterio atstumas iki kito klasterio yra didesnis už ribinę reikšmę:

$$\text{cluster.getDistance()} \leq \text{threshold AND cluster.getParent().getDistance()} > \text{threshold}$$

Identifikuojant klasterius pagal gylį, tikrinama sąlyga: ar klasterio gylis medyje yra lygus pasirinktam gyliui.

Pamėginus atlikti klasterizavimą, greit galima pastebėti, kad iš anksto atspėti ribinę reikšmę yra gana sudėtinga, o imant tik tam tikro gylio klasterius visada gaunamas klasterių skaičius 2^n , kur n yra gylis. Taigi abiem atvejais sunku tiksliai identifikuoti klasterius.

Pažvelgus į dendrogramą, galima pastebėti, kad jei turimas klasterių medis, tai ribinę reikšmę galima suskaičiuoti kiekvienam atvejui atskirai. Pavyzdžiui, tarkime, kad turime klasterių medį. Klasterių medis pavaizduotas 1 paveiksle. Paveikslo viršuje esanti horizontali ašis rodo atstumą tarp klasterių, kairėje pusėje yra šakninis klasteris, o dešinėje išsidėstę lapai – dokumentai, kurie yra klasterizuojami. Jei iš turimo klasterių medžio norima gauti keturis klasterius, reikia medį „pjauti“ vertikalia linija, taip kaip parodyta 1 paveiksle. „Atpjaautos“ šakos ir yra ieškomi klasteriai.

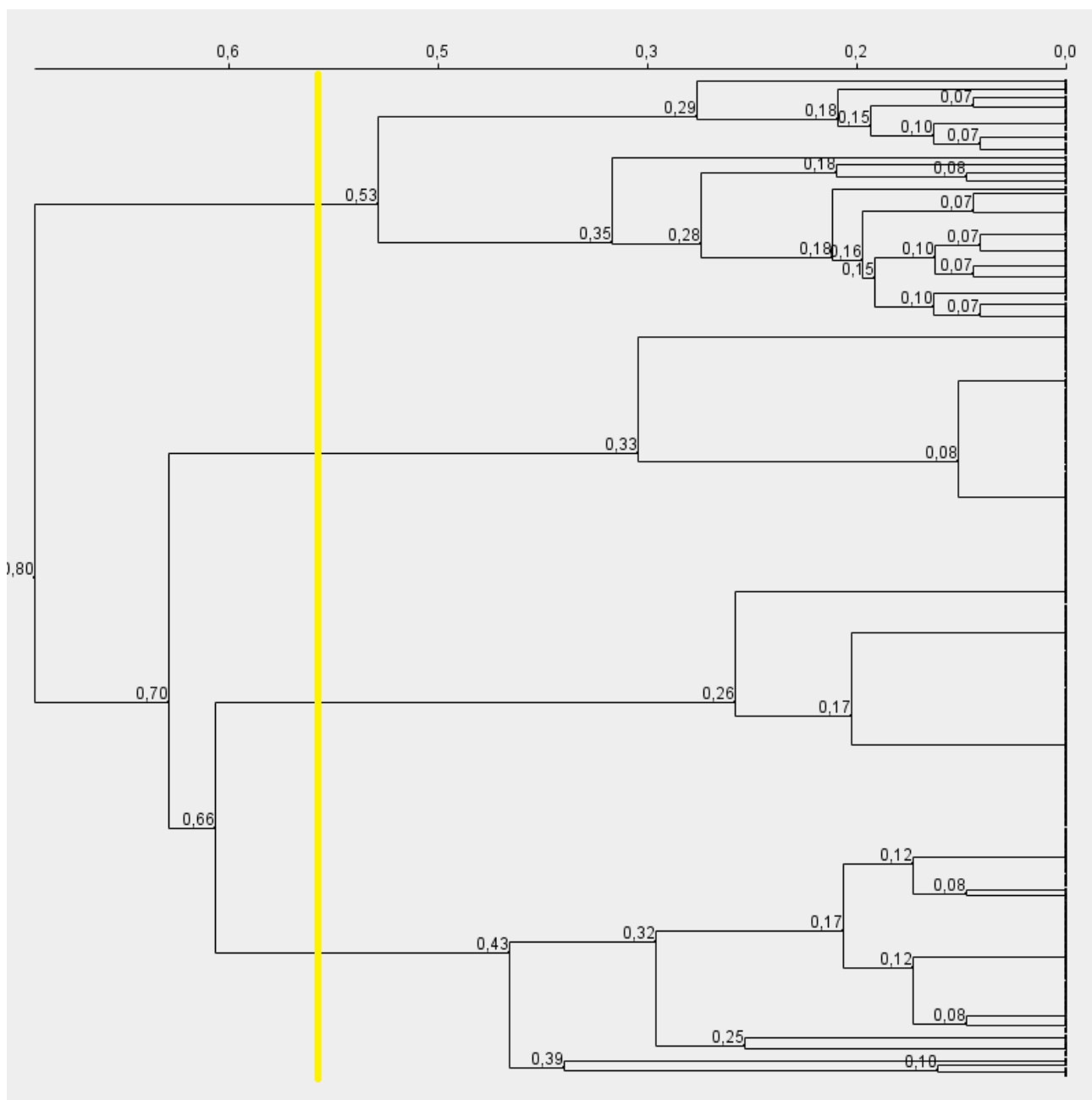
Norint surasti ribinę reikšmę, ties kuria „atpjovus“ gaunami keturi klasteriai, reikia mažėjimo tvarka išrūšiuoti turimas medžio šakas pagal atstumą tarp jų. Išrūšėjęs ribinė reikšmė visada bus tarp trečio ir ketvirto klasterio atstumo. Bendresniu atveju, jei norima gauti n klasterių, tai ribinė reikšmė bus tarp n – tojo ir $n - 1$ – ojo klasterio.

Mūsų atveju yra naudojama tokia „pjovimo“ sąlyga:

`cluster.getDistance() <= threshold && cluster.getParent().getDistance() > threshold`

Todėl pjovimo riba imamas n – tojo klasterio atstumas tarp jo vaikinių klasterių.

„Atpjovus“ klasterius ties tokiu būdu parinkta reikšme, daugeliu atveju gaunamas norimas klasterių skaičius. Problemų kyla kraštutiniais atvejais, pavyzdžiui, jei yra keli klasteriai su vienodais atstumais arba yra labai daug klasterių, tarp kurių atstumas yra lygus vienetui. Tokiais atvejais naudojame klasterių identifikavimą pagal gylį.



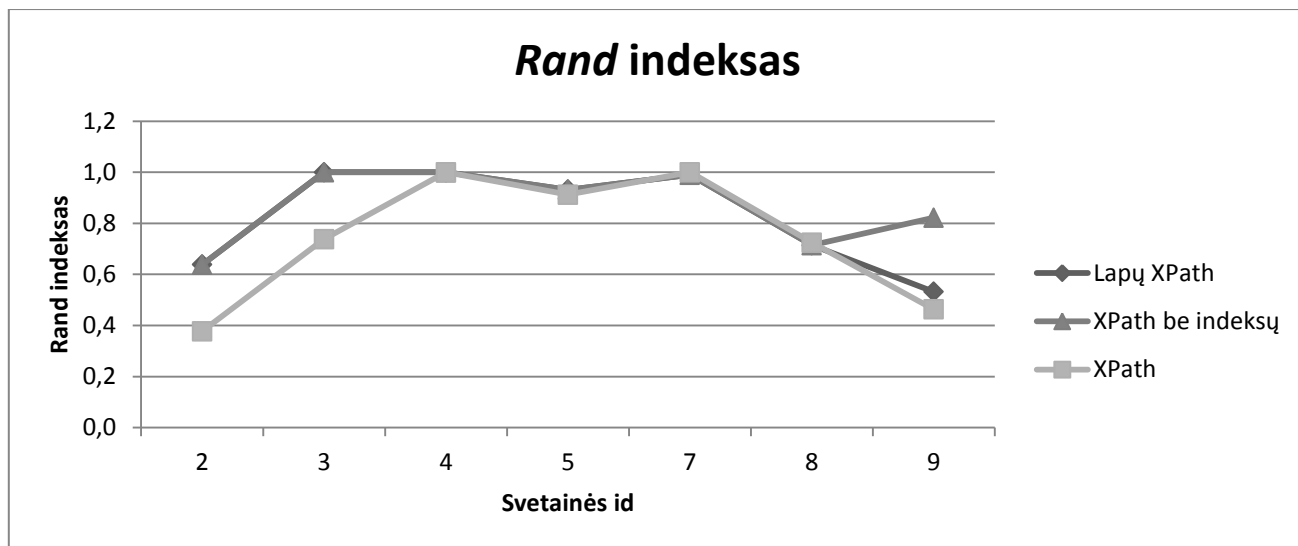
1. pav. Klasterių identifikavimas iš dendrogramos.

3.2. XPATH parinkimo įtaka klasterizavimui

Šiame skyriuje aprašomi eksperimentai, kuriuose klasterizavimo kokybė lyginama naudojant skirtingus XPATH parinkimo būdus: XPATH, XPATH be indeksų, lapų XPATH be indeksų.

3.2.1. Geriausias XPATH parinkimo metodas pagal Rand indeksą

2 paveiksle pavaizduota, kaip kiekvienoje svetainėje kinta *Rand* indeksas priklausomai nuo XPATH parinkimo būdo. Vertikalioje ašyje yra *Rand* indeksas, horizontalioje ašyje yra svetainės identifikacinis numeris, o patys taškai ir linijos grafike vaizduoja indekso priklausomybę nuo XPATH parinkimo būdo. Ašys yra parinktos būtent tokiu būdu, nes mus domina, kuriuo XPATH parinkimo būdu gaunamos didžiausios indekso reikšmės visose svetainėse



2. pav. *Rand* indekso priklausomybė nuo XPATH sudarymo būdo.

2 lentelėje nurodyta, kokių XPATH kelių parinkimo būdų kiekvienoje svetainėje gaunamas didžiausias *Rand* indeksas. Kai kurioms svetainėms didžiausias *Rand* indeksas gaunamas keliais atvejais, todėl tos svetainės lentelėje turi po kelias eilutes.

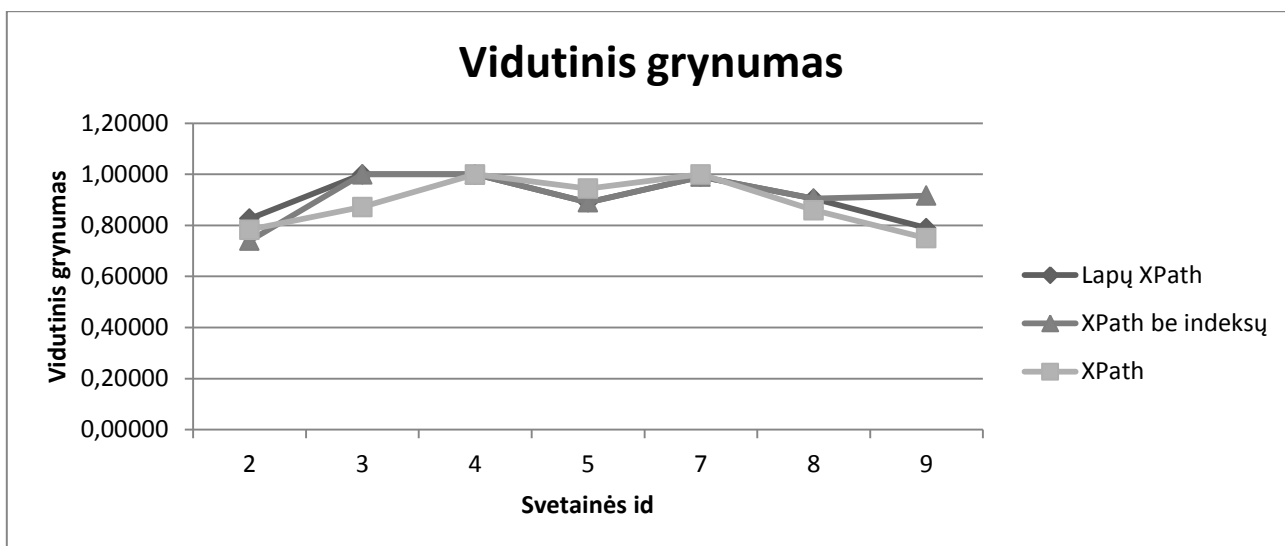
2 lentelė. Maksimalūs *Rand* indeksai svetainėse.

Svetainės id	Maksimalus <i>Rand</i> indeksas svetainėje	XPATH skaičiavimo tipas, kuriuo pasiekiamas maksimalius <i>Rand</i> indeksas
2	0.63855	Lapų XPATH
3	1.00000	XPATH be indeksų
3	1.00000	Lapų XPATH
4	1.00000	XPATH
4	1.00000	XPATH be indeksų
4	1.00000	Lapų XPATH
5	0.93187	XPATH be indeksų
5	0.93187	Lapų XPATH
7	1.00000	XPATH
8	0.72435	XPATH
9	0.82227	XPATH be indeksų

Matome, kad svetainėse: „2“, „3“ ir „9“, didžiausios *Rand* indekso reikšmės gaunamos naudojant XPATH be indeksų ir lapų XPATH. Kitose svetainėse indekso reikšmės yra panašios naudojant visus tris XPATH gavimo būdus. Taigi galima daryti išvadą, kad didžiausias *Rand* indeksas gaunamas naudojant lapų XPATH kelius arba XPATH be indeksų.

3.2.1. Geriausias XPATH parinkimo metodas pagal vidutinį grynumą

3 paveiksle pavaizduota, kaip, priklausomai nuo XPATH parinkimo būdo, kiekvienoje svetainėje kinta vidutinis grynumas.



3. pav. Vidutinio grynumo priklausomybė nuo XPATH sudarymo būdo.

3 lentelėje nurodyta, kokių XPath parinkimo būdų kiekvienoje svetainėje gaunamas didžiausias vidutinis grynumas. Kai kurioms svetainėms didžiausias vidutinis grynumas gaunamas keliais būdais, todėl tos svetainės lentelėje turi po kelias eilutes.

3 lentelė. Maksimalus vidutinis grynumas svetainėse.

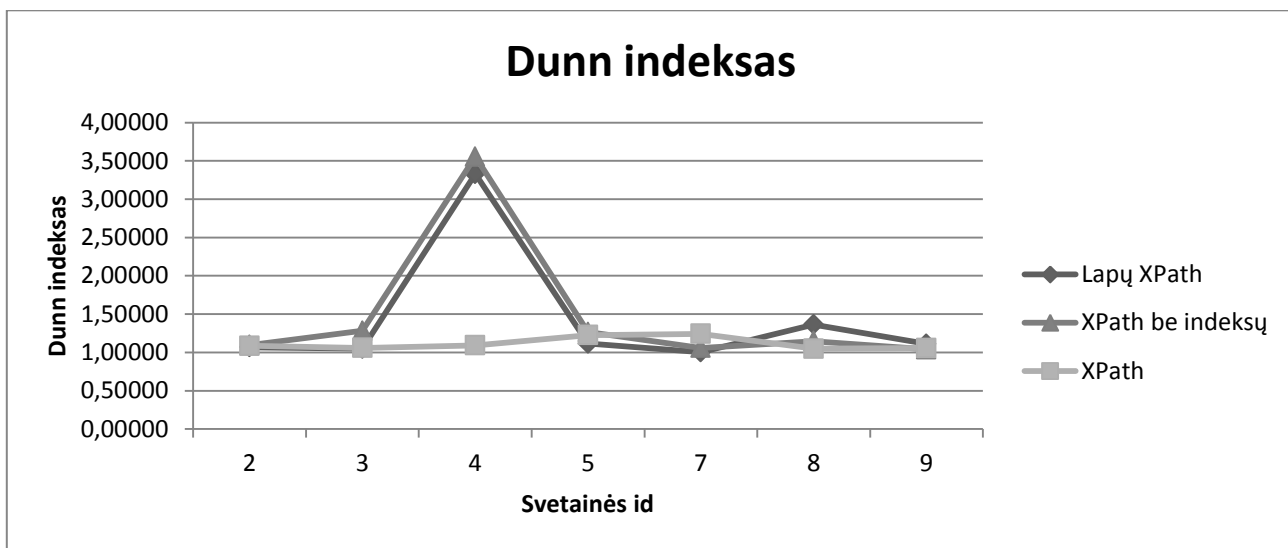
Svetainės id	Maksimalus vidutinis grynumas svetainėje	XPATH skaičiavimo tipas, kuriuo pasiekiamas maksimalus vidutinis grynumas
2	0.82659	Lapų XPath
3	1.00000	XPATH be indeksų
3	1.00000	Lapų XPath
4	1.00000	XPATH
4	1.00000	XPATH be indeksų
4	1.00000	Lapų XPath
5	0.94410	XPATH

Svetainės id	Maksimalus vidutinis grynumas svetainėje	XPATH skaičiavimo tipas, kuriuo pasiekiamas maksimalius vidutinis grynumas
7	1.00000	XPATH
8	0.90476	XPATH be indeksų
8	0.90476	Lapų XPATH
9	0.91667	XPATH be indeksų

Matome, kad vidutinis grynumas nežymiai priklauso nuo XPATH parinkimo būdo. Visose svetainėse indekso reikšmės yra panašios naudojant visus tris XPATH gavimo būdus. Tačiau XPATH be indeksų ir lapų XPATH rodo šioji tokį pranašumą kai kuriose svetainėse.

3.2.1. Geriausias XPATH parinkimo metodas pagal Dunn indeksą

4 paveiksle pavaizduota, kaip kinta *Dunn* indeksas priklausomai nuo kelių parinkimo būdo kiekvienoje svetainėje.



4. pav. *Dunn* indekso priklausomybė nuo XPATH sudarymo būdo.

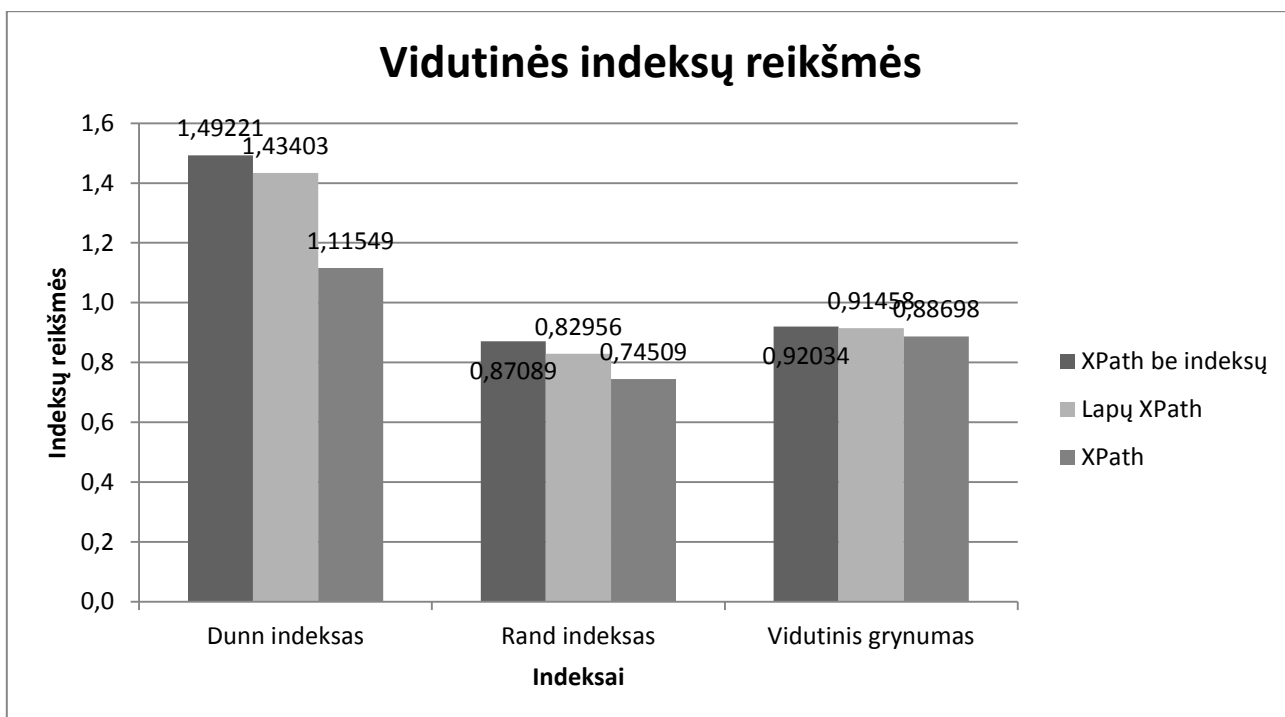
4 lentelėje nurodyta, koku XPATH parinkimo būdu gaunamas didžiausias *Dunn* indeksas kiekvienoje svetainėje.

4 lentelė. Maksimalūs *Dunn* indeksai svetainėse.

Svetainės id	Maksimalus <i>Dunn</i> indeksas svetainėje	XPath skaičiavimo tipas, kuriuo pasiekiamas maksimalius <i>Dunn</i> indeksas
2	1.09689	XPath be indeksų
3	1.28314	XPath be indeksų
4	3.55428	XPath be indeksų
5	1.26533	XPath be indeksų
7	1.24281	XPath
8	1.36388	Lapų XPath
9	1.11019	Lapų XPath

Matome, kad akivaizdų pranašumą turi XPATH be indeksų ir lapų XPATH.

5 paveiksle pateikiamos vidutinės indeksų reikšmės, suskaičiuotos remiantis indeksų reikšmėmis, gautomis suklasterizavus skirtingų internetinių svetainių dokumentus.



5. pav. *Rand* indekso priklausomybė nuo XPATH sudarymo būdo.

Matome, kad vidutiniškai didžiausios indeksų reikšmės gaunamos naudojant XPATH be indeksų, nedaug nuo jų atsilieka ir lapų XPATH. XPATH su indeksais duoda kiek blogesnius rezultatus. Be to, svetainėse „2“, „3“, „4“ ir „5“ didžiausios *Dunn* indekso reikšmės gaunamos naudojant XPATH be indeksų. Svetainėse „8“ ir „9“ didžiausios *Dunn* indekso reikšmės gaunamos naudojant lapų XPATH. Vienintelėje „7“ svetainėje pilnų XPATH *Dunn* indeksas yra nežymiai didesnis už XPATH be indeksų ir lapų XPATH *Dunn* indeksą. Be to, nesunku pastebėti, kad svetainėje „4“ indekso reikšmė, gaunama naudojant XPATH be indeksų ir lapų XPATH, yra žymiai didesnė, nei naudojant pilnus XPATH. Taigi galima daryti išvadą, kad didžiausias *Dunn* indeksas gaunamas naudojant XPATH be indeksų arba lapų XPATH.

3.3. Bendrų dokumentų dalių pašalinimas

Šiame skyriuje nagrinėsime, kokią įtaką klasterizavimo kokybei turi bendrų dokumentų dalių pašalinimas skaičiuojant atstumus tarp dokumentų. Ankstesniame skyriuje parodėme, kad geriausi

XPATH parinkimo būdai yra XPATH be indeksų ir lapų XPATH. Taigi šiame skyriuje naudosime tik šiuos du būdus.

3.3.1. XPATH rinkinio pradžios ir pabaigos nukirtimas

Siekiant palyginti XPATH rinkinio pradžios ir pabaigos nukirtimo metodą su kitais bendrų dokumentų dalių pašalinimo metodais, pirmiausia reikia nustatyti, kokią dalį XPATH rinkinio reikia nukirsti, kad gautume geriausią rezultatą.

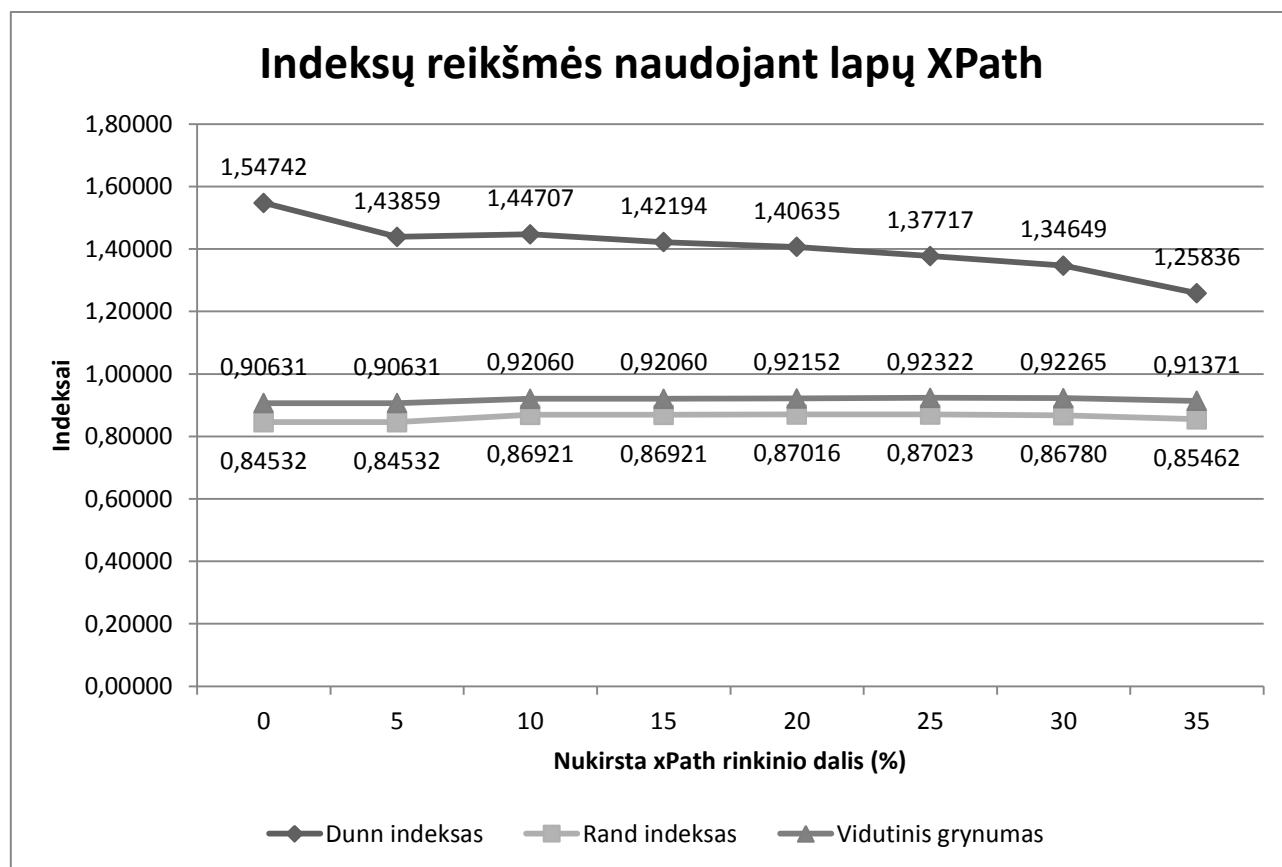
Buvo atliktas eksperimentas, kuriuo bandyta nukirsti nuo nulio iki trisdešimt penkių procentų XPATH rinkinio pradžios ir pabaigos. Daugiau nei trisdešimt penkių procentų nukirsti nėra tikslinga, kadangi tuomet lieka tik mažiau nei trisdešimt procentų pradinio XPATH rinkinio, kuris jau nelabai gerai atspindi dokumento struktūrą.

5 lentelėje pateikiamos indeksų reikšmės taikant XPATH rinkinio dalies nukirtimo metodą lapų XPATH.

5 lentelė. Indeksų reikšmės taikant XPATH rinkinio dalies nukirtimo ir lapų XPATH metodus.

Pašalinama dalis (%)	Dunn indeksas	Rand indeksas	Vidutinis grynumas
0	1,54742	0,84532	0,90631
5	1,43859	0,84532	0,90631
10	1,44707	0,86921	0,92060
15	1,42194	0,86921	0,92060
20	1,40635	0,87016	0,92152
25	1,37717	0,87023	0,92322
30	1,34649	0,86780	0,92265
35	1,25836	0,85462	0,91371

6 ir 7 paveiksluose pavaizduota, kaip vidutinės indeksų reikšmės priklauso nuo nukirstos dokumento dalies, kai naudojami lapų XPATH ir XPATH be indeksų metodai. Procentais nurodyta, kokia dalis XPATH rinkinio yra nukertama nuo pradžios ir pabaigos.



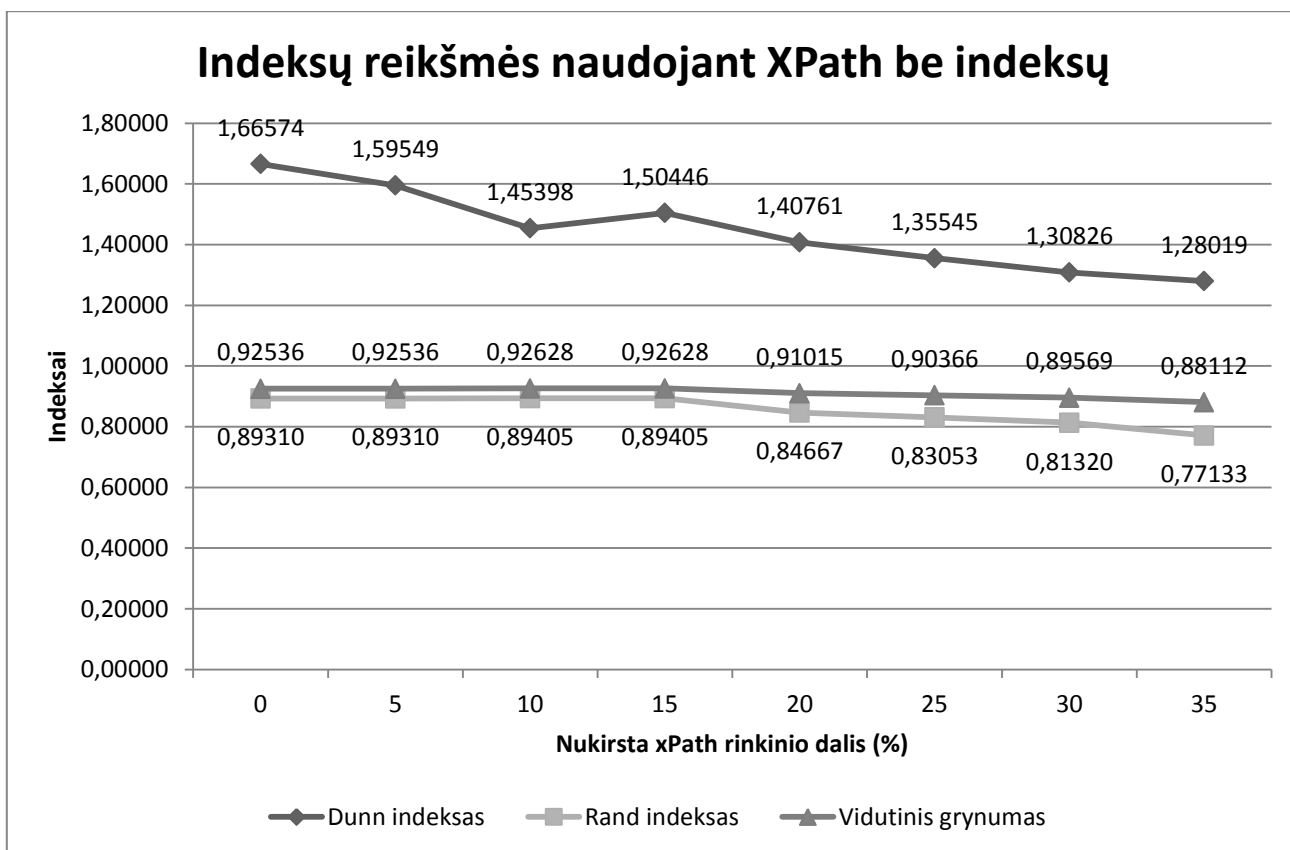
6. pav. Vidutinių indeksų reikšmių priklausomybė nuo pašalinamos XPATH rinkinio dalies naudojant lapų XPATH metodą.

Iš diagramos matome, kad didinant nukertamą XPATH rinkinio dalį, *Dunn* indeksas mažėja. Tai reiškia, jog atskirtis tarp gautų klasterių mažėja. Klasterizavimo tikslumą parodantys indeksai *Rand* indeksas ir vidutinis grynumas kinta labai nežymiai. Didžiausias reikšmes šie indeksai įgyja, kai nukertama dvidešimt penki procentai lapų XPATH nuo XPATH rinkinio pradžios ir pabaigos.

6 lentelėje pateikiamos indeksų reikšmės, taikant XPATH rinkinio dalies nukirtimo ir XPATH be indeksų metodus kartu.

6 lentelė. Indeksų reikšmės taikant XPATH rinkinio dalies nukirtimo ir XPATH be indeksų metodus.

Pašalinama dalis (%)	<i>Dunn</i> indeksas	<i>Rand</i> indeksas	Vidutinis grynumas
0	1,66574	0,89310	0,92536
5	1,59549	0,89310	0,92536
10	1,45398	0,89405	0,92628
15	1,50446	0,89405	0,92628
20	1,40761	0,84667	0,91015
25	1,35545	0,83053	0,90366
30	1,30826	0,81320	0,89569
35	1,28019	0,77133	0,88112



7. pav. Vidutinių indeksų reikšmių priklausomybė nuo pašalinamos XPATH rinkinio dalies naudojant XPATH be indeksų metodą.

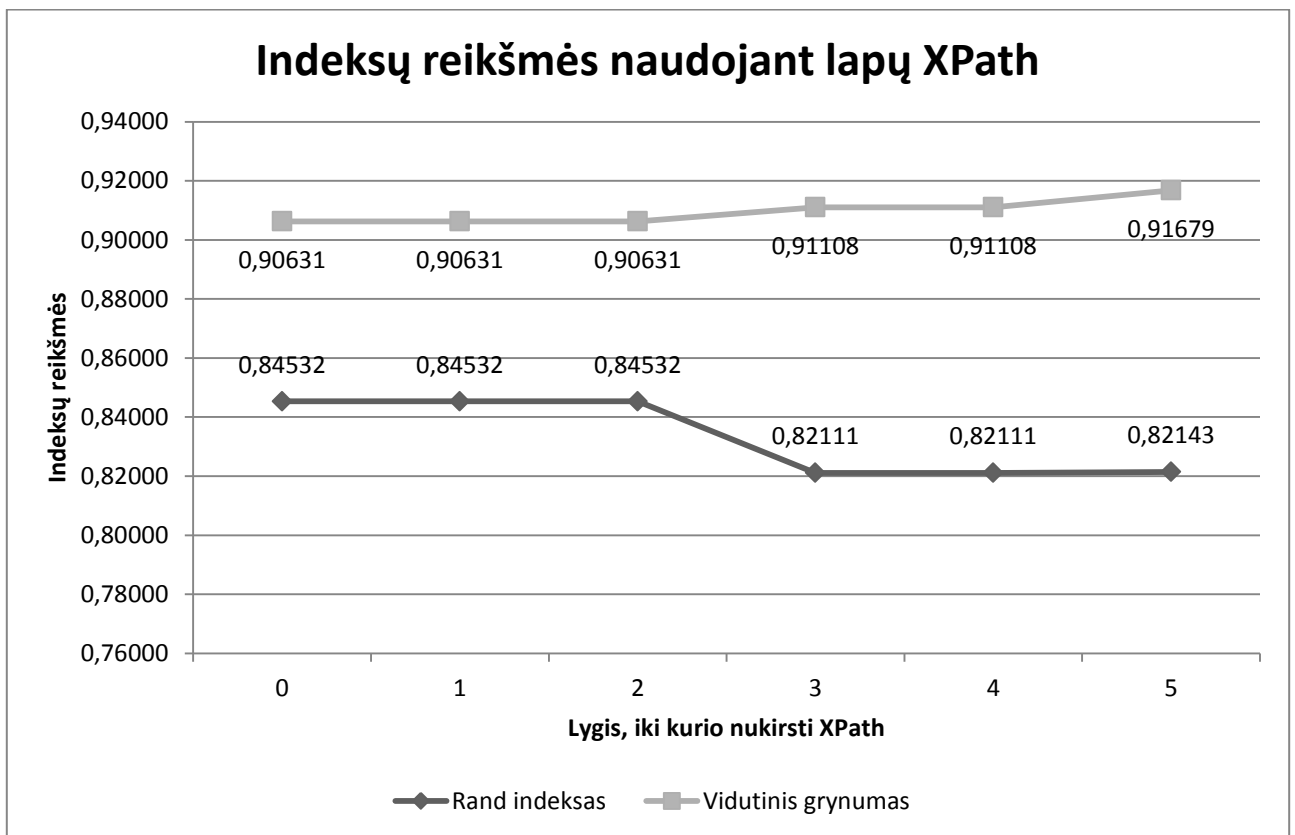
Iš 7 paveiksle pavaizduotos diagramos matome, kad didinant nukertamą XPATH rinkinio dalį, *Dunn* indeksas taip pat mažėja. *Rand* indeksas ir vidutinis grynumas iš pradžių kinta nežymiai, o paskui pradeda mažėti. Didžiausias reikšmės *Rand* indeksas ir vidutinis grynumas įgyja, kai nukertama penkiolika procentų XPATH be indeksų nuo kelių rinkinio pradžios ir pabaigos.

XPATH rinkinio dalies nukirtimas neduoda akivaizdžios naudos hipertekstinių dokumentų klasterizavimui. Be to, šis metodas mažina *Dunn* indeksą, o tai reiškia, kad mažėja atskirtis tarp gautų klasterių. Tokiu būdu skaičiuojant atstumus, dokumentų skirtumai mažėja. Vis dėlto pastebėjome, kad *Rand* indeksas ir vidutinis klasterių grynumas nežymiai padidėja, kai yra nukertama dvidešimt penki procentai lapų XPATH nuo kelių rinkinio pradžios ir pabaigos arba kai yra nukertama penkiolika procentų XPATH be indeksų nuo kelių rinkinio pradžios ir pabaigos.

3.3.2. XPATH pradžios nukirtimas

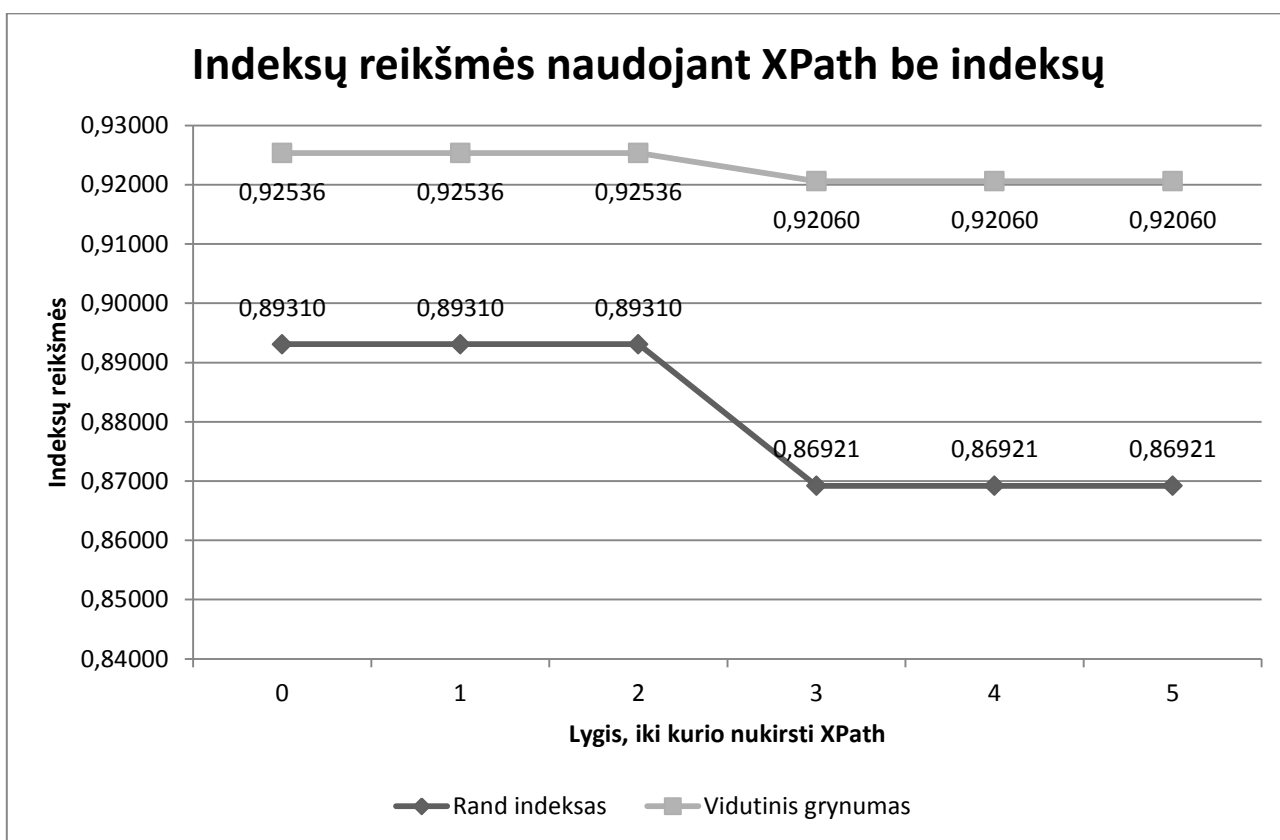
Šiame poskyryje nagrinėjamas XPATH nukirtimas iki tam tikro gylio. Aprašomas atliktas eksperimentas, kurio metu buvo tikrinama, kokią įtaką klasterizavimo kokybei turi XPATH pradžios nukirtimas. Eksperimento metu buvo nukertama nuo nulio iki penkių pirmųjų XPATH esančių elementų pavadinimų. Daugiau nukirtinėti nėra prasmės, kadangi gauti XPATH nebeatspindės realios dokumento struktūros.

8, 9, 10 paveiksluose pavaizduota, kaip vidutinės indeksų reikšmės priklauso nuo nukirstos XPATH dalies, kai naudojami lapų XPATH ir XPATH be indeksų. Horizontalioje ašyje nurodyta, kiek elementų nuo XPATH pradžios yra pašalinta.



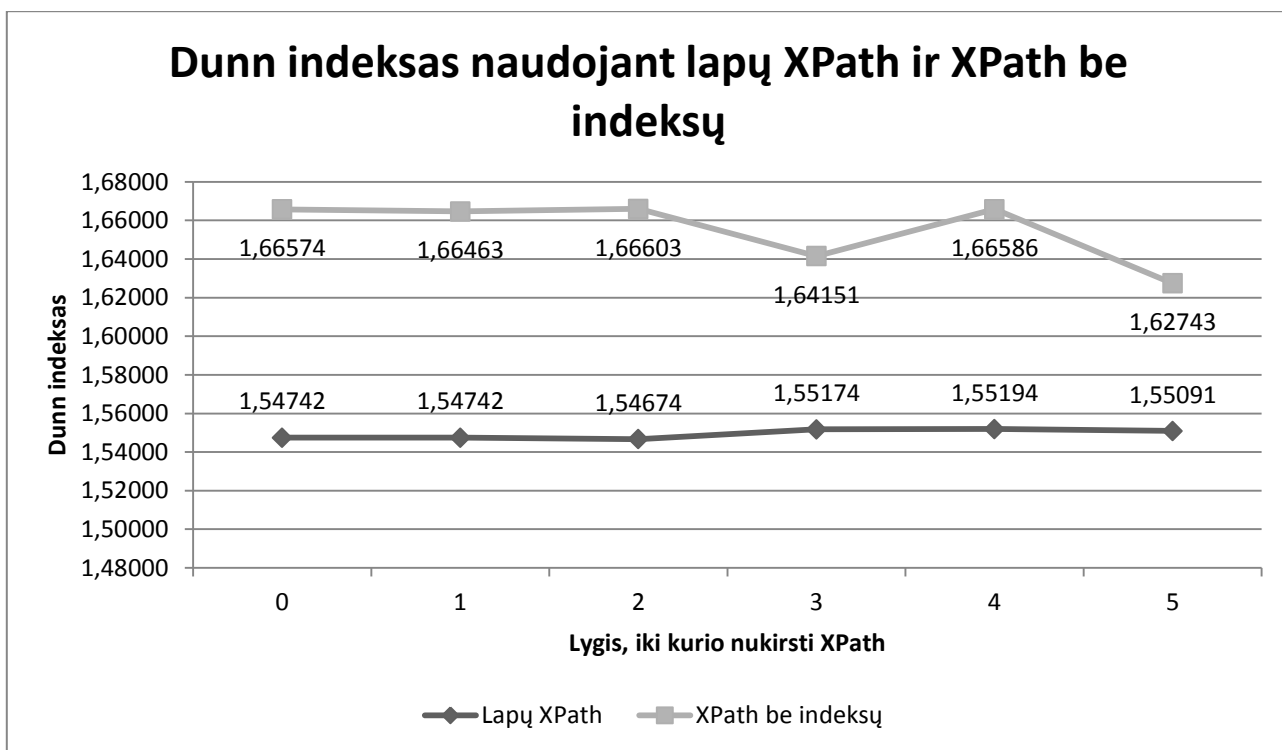
8. pav. Rand indekso ir vidutinio grynumo priklausomybė nuo nukirstos XPATH dalies naudojant lapų XPATH metodą.

8 paveiksle esanti diagrama vaizduoja vidutinių indeksų reikšmių priklausomybę nuo nukirstos XPATH dalies naudojant lapų XPATH. Matome, kad, didinant nukertamą XPATH dalį, gaunamų klasterių vidutinis grynumas truputį didėja. Tačiau *Rand* indeksas, parodantis, kaip gerai gauta klasterių konfigūracija atitinka „teisingą“, iš anksto apibrėžtą sąranką, mažėja didinant nukertamą XPATH dalį.



9. pav. *Rand* indekso ir vidutinio grynumo priklausomybė nuo nukirstos XPATH dalies naudojant XPATH be indeksų metodą.

Naudojant XPATH be indeksų, galima nesunkiai pastebėti (9 paveikslas), kad *Rand* indeksas ir vidutinis grynumas turi tendenciją mažėti didinant nukertamą XPATH dalį.



10. pav. *Dunn* indekso priklausomybė nuo nukirstos XPATH dalies.

Vidutinis *Dunn* indeksas kinta nežymiai (10 paveikslas). Tiek naudojant lapų XPATH, tiek XPATH be indeksų, *Dunn* indeksas šiek tiek kinta. Galima įžvelgti ir nežymią tendenciją mažėti, kai XPATH nukirtimo lygis didėja.

Visi nagrinėjami indeksai (*Rand* indeksas, *Dunn* indeksas, vidutinis grynumas) didinant nukertamą XPATH dalį, mažėja arba kinta labai nežymiai. Todėl, jei siekiama pagerinti klasterizavimą, šis metodas nėra tinkamas pasikartojančių dokumento dalių pašalinimui.

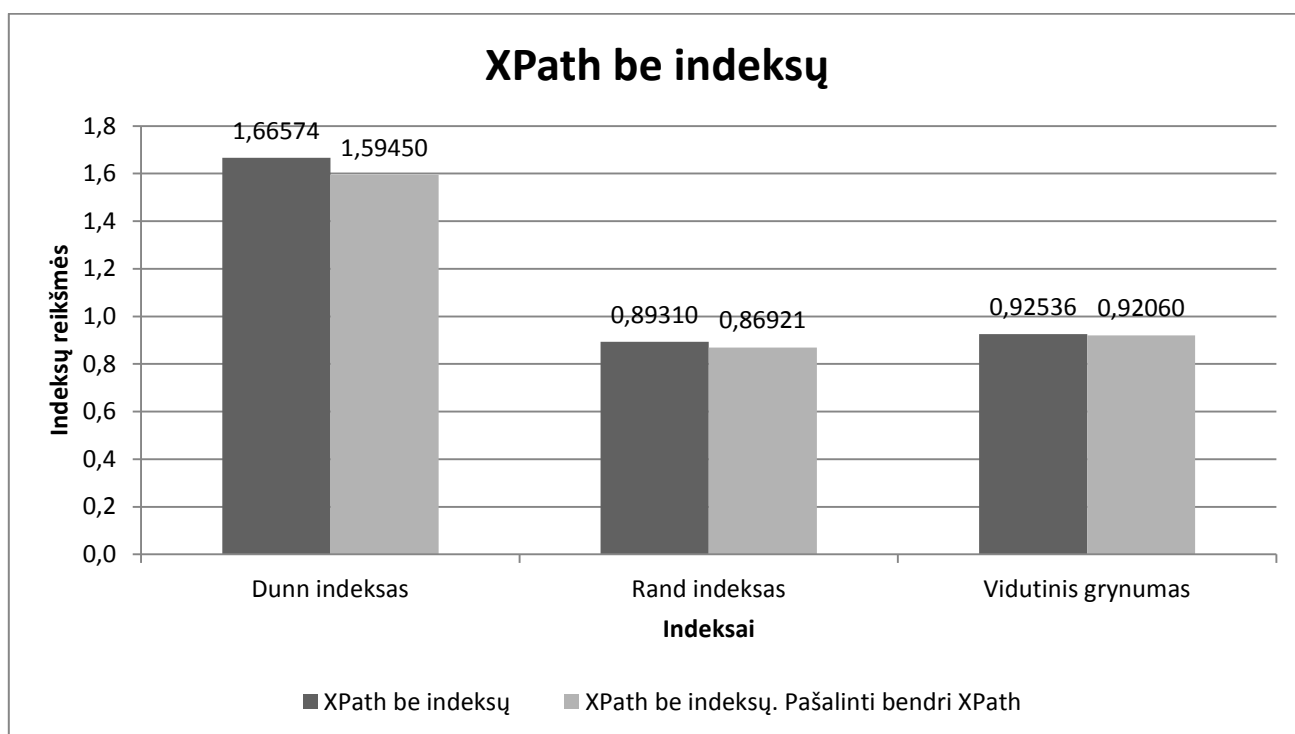
3.3.3. Bendrų visiems dokumentams XPATH pašalinimas

Šiame poskyryje aprašomas eksperimentas, kurio metu buvo tikrinta, kokią įtaką klasterizavimo kokybei turi bendrų XPATH pašalinimas iš visų dokumentų. Eksperimento metu skaičiuojant atstumą, buvo neatsižvelgiama į XPATH, kurie bent vieną kartą patenka į kiekvieną iš klasterizuojamų dokumentų.

7 lentelėje pateikiamos indeksų reikšmės, kai naudojami bendrų dokumento dalių pašalinimo metodai iš XPATH be indeksų ir lapų XPATH.

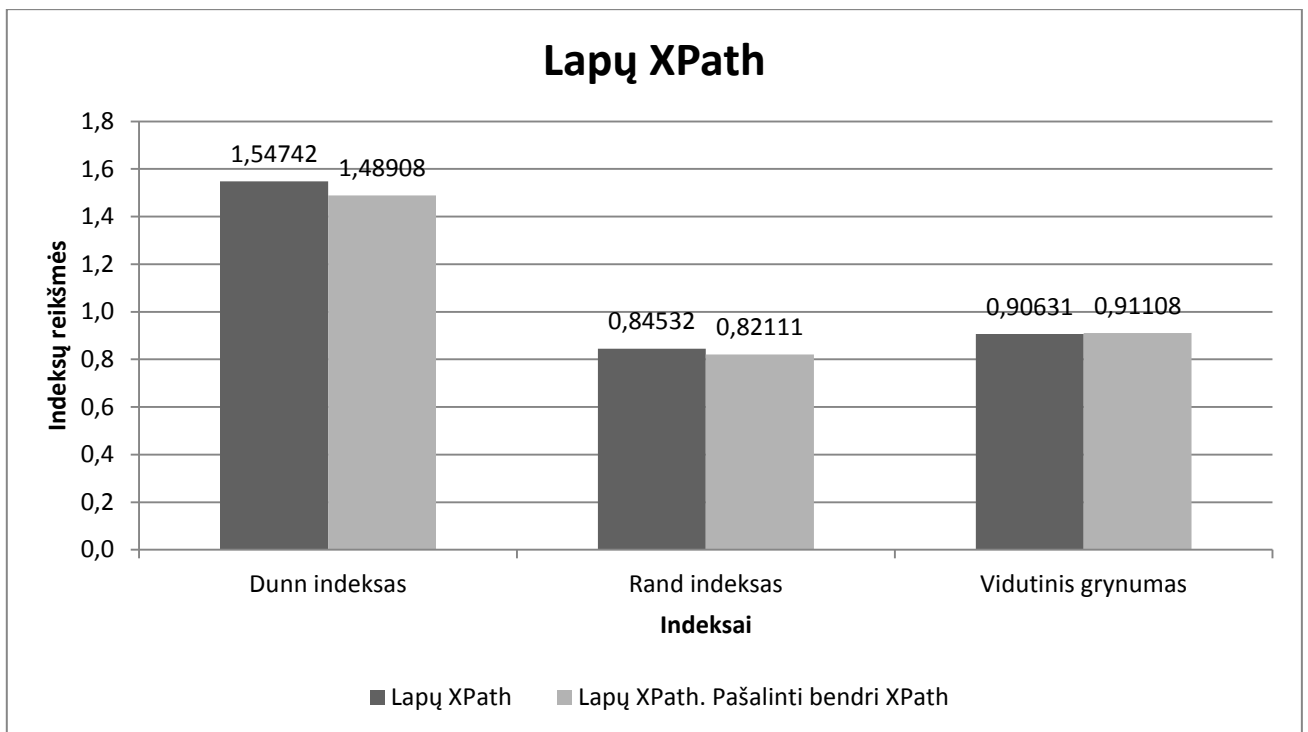
7 lentelė. Indeksų reikšmės taikant bendrų XPATH pašalinimo metodą.

Metodas	<i>Dunn</i> indeksas	<i>Rand</i> indeksas	Vidutinis grynumas
XPath be indeksų	1,66574	0,89310	0,92536
XPath be indeksų. Pašalinti bendri XPath	1,59450	0,86921	0,92060
Lapų XPath	1,54742	0,84532	0,90631
Lapų XPath. Pašalinti bendri XPath	1,48908	0,82111	0,91108



11. pav. Bendrų XPATH pašalinimo įtaka klasterizavimui naudojant XPATH be indeksų metodą.

11 paveiksle matome, kad iš XPATH be indeksų rinkinio pašalinus tuos elementus, kurie yra bendri visiems dokumentams, klasterizavimo kokybė nežymiai pablogėja. Vidutinis grynumas beveik nepasikeičia.



12. pav. Bendrų XPATH pašalinimo įtaka klasterizavimui naudojant lapų XPATH metodą.

Panaši situacija yra su lapų XPATH (12 paveikslas). Šiuo atveju *Dunn* ir *Rand* indeksai taip pat truputį sumažėja pašalinus iš dokumentų bendrus XPATH. Vidutinis grynumas truputį padidėja, tačiau labai nežymiai. Galima sakyti, kad bendrų XPATH pašalinimas iš dokumentų, neturi didelės įtakos klasterizavimo kokybei.

3.4. Darbe nagrinėtų metodų palyginimas

Šiame poskyryje palyginami darbe nagrinėti metodai. 8 lentelėje pateikiamos indeksų reikšmės, taikant visus darbe nagrinėtus metodus.

8 lentelė. Geriausių, darbe nagrinėtų metodų indeksų reikšmės.

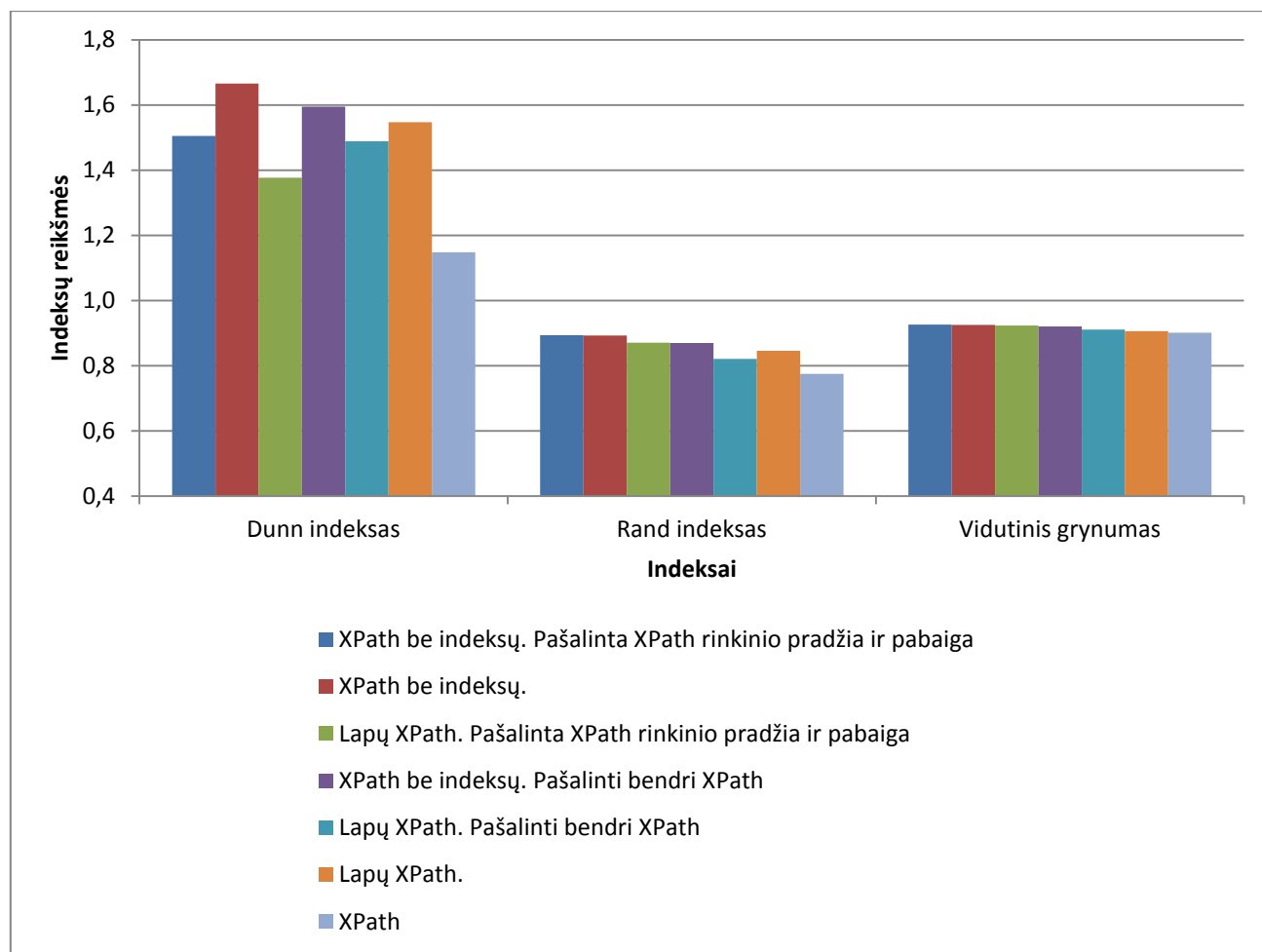
Metodas	Indeksai		
	<i>Dunn</i>	<i>Rand</i>	vidutinis grynumas
XPATH be indeksų. Pašalinta XPATH rinkinio pradžia ir pabaiga	1,50446	0,89405	0,92628
XPATH be indeksų.	1,66574	0,89310	0,92536
Lapų XPATH. Pašalinta XPATH rinkinio pradžia ir pabaiga	1,37717	0,87023	0,92322
XPATH be indeksų. Pašalinti bendri XPATH	1,59450	0,86921	0,92060
Lapų XPATH. Pašalinti bendri XPATH	1,48908	0,82111	0,91108
Lapų XPATH.	1,54742	0,84532	0,90631
XPATH	1,14795	0,77521	0,90099

Lapų XPATH su pašalinta XPATH rinkinio pradžia ir pabaiga metode yra nukertama dvidešimt penki procentai XPATH rinkinio pradžios ir pabaigos, nes, nagrinėjant metodą, buvo nustatyta, kad su šia reikšme, metodas duoda geriausius rezultatus. Taikant XPATH be indeksų su pašalinta XPATH rinkinio pradžia ir pabaiga metodą, yra nukertama penkiolika procentų XPATH kelių rinkinio pradžios ir pabaigos. XPATH pradžios nukirtimo metodas čia nėra lyginamas su kitais metodais, nes jis duoda blogesnius rezultatus, nei kiti metodai.

Didžiausias *Dunn* indeksas gaunamas naudojant XPATH be indeksų. Didžiausias *Rand* indeksas gaunamas naudojant XPATH be indeksų, iš kurių pašalinta penkiolika procentų rinkinio

pradžios ir pabaigos. Didžiausias vidutinis grynumas gaunamas naudojant XPATH be indeksų, iš kurių taip pat pašalinta penkiolika procentų rinkinio pradžios ir pabaigos.

Mažiausios visų indeksų reikšmės gaunamos naudojant paprastus XPATH su indeksais (8 lentelė).



13. pav. Indeksai naudojant įvairius metodus.

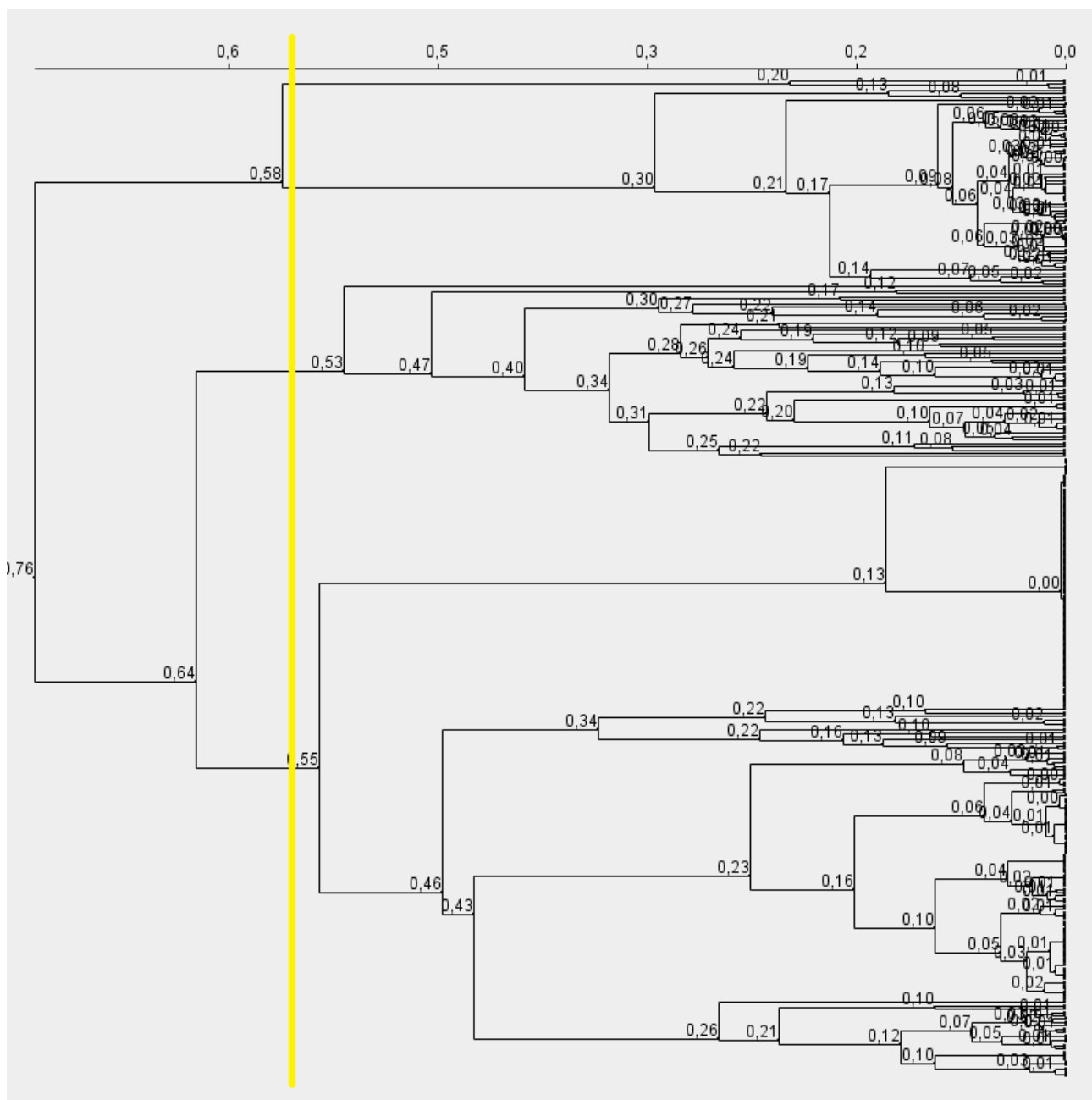
13 paveiksle pateikiamos indeksų reikšmės naudojant įvairius metodus. Metodai išrūšiuoti mažėjimo tvarka pagal vidutinį grynumą.

3.5. Metodų taikymo pavyzdžiai

Šiame poskyryje pateikiamos kelios tos pačios svetainės dokumentų klasterizavimo dendrogramos panaudojant skirtingus metodus.

3.5.1. XPATH be indeksų ir bendrų dokumentų XPATH pašalinimas

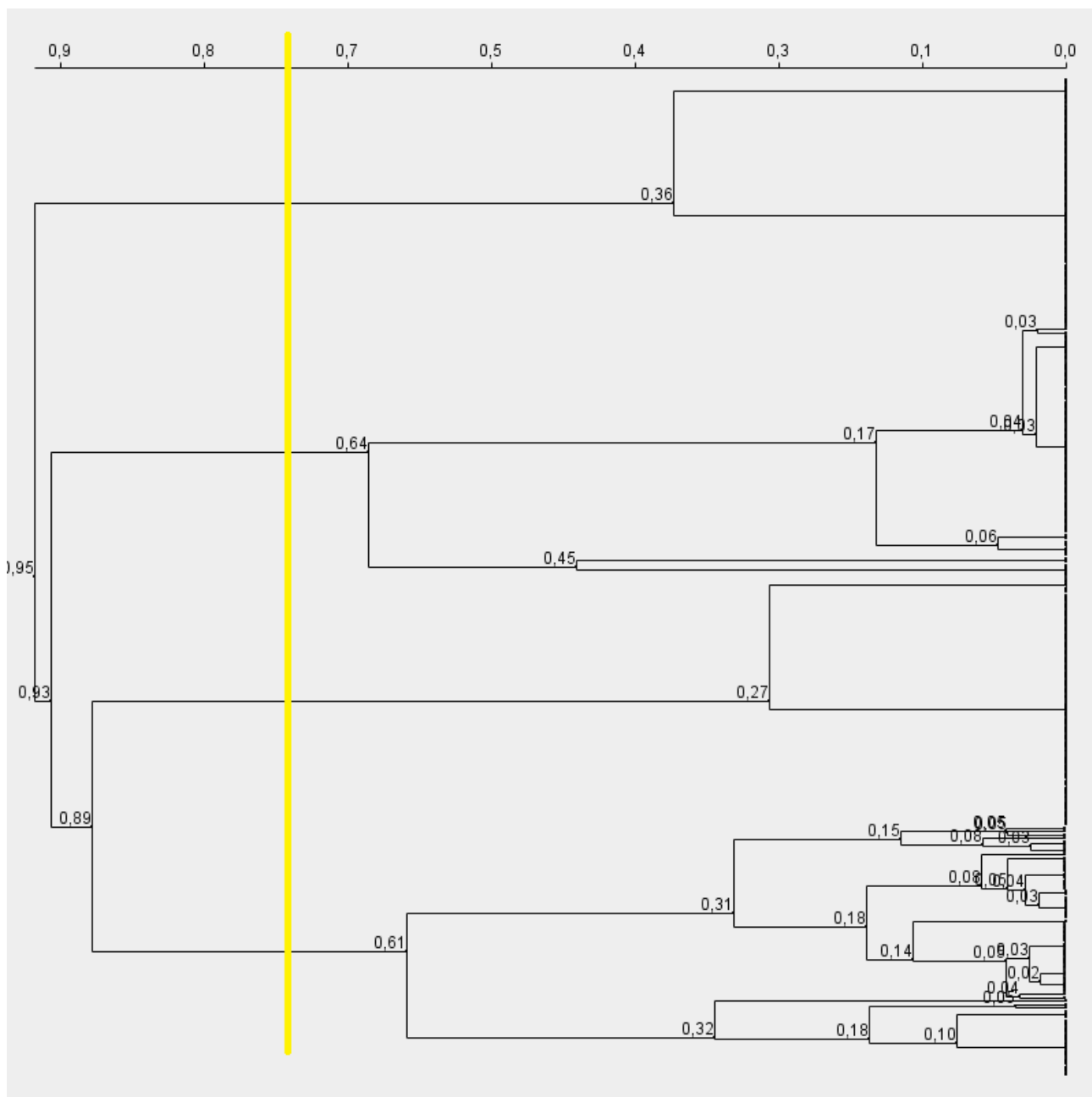
Šiame pavyzdyje parodoma, kaip XPATH be indeksų metodo ir bendrų XPATH pašalinimo metodo kombinacija išsprendžia svetainės „3“ klasterizavimo problemą.



14. pav. Svetainės „3“ klasterių dendrograma, gauta naudojant XPATH metodą.

14 paveiksle pavaizduotoje dendrogramoje matome, kad nukirtus medžio šakas ties ribine reikšme, taip kad būtų gauti keturi klasteriai, į kiekvieną klasterį patenka labai ne vienodas skaičius dokumentų, nors eksperimento metu buvo imama po vienodą skaičių dokumentų, priklausančių

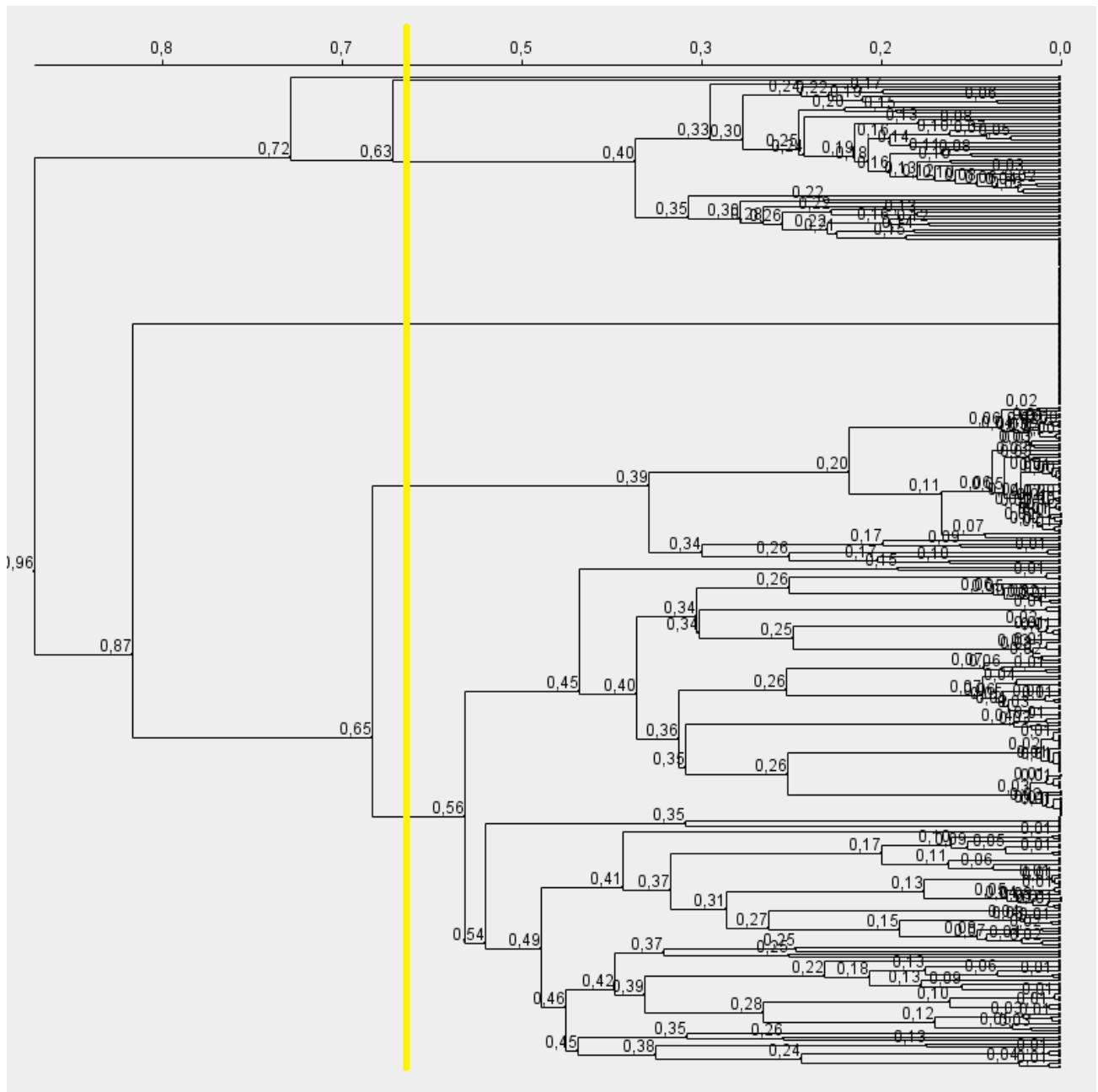
kiekvienam klasteriui. Šios klasterių konfigūracijos indeksai yra tokie: *Dunn*: 1.04985, *Rand*: 0.63000, vidutinis grynumas: 0.85081.



15. pav. Svetainės „3“ klasterių dendrograma, gauta naudojant XPATH be indeksų metodą, pašalinus bendrus XPATH.

15 paveiksle pavaizduotoje dendrogramoje matome, kad nukirtus medžio šakas ties ribine reikšme, taip kad būtų gauti keturi klasteriai, į kiekvieną klasterį patenka vienodas skaičius dokumentų. Taigi XPATH be indeksų ir bendrų XPATH pašalinimo metodų kombinacija išsprendžia svetainės „3“ klasterizavimo problemą. Šios klasterių konfigūracijos indeksai yra tokie: *Dunn*: 1.39543, *Rand*: 1, vidutinis grynumas: 1. Taigi dokumentai suklasterizuoti idealiai.

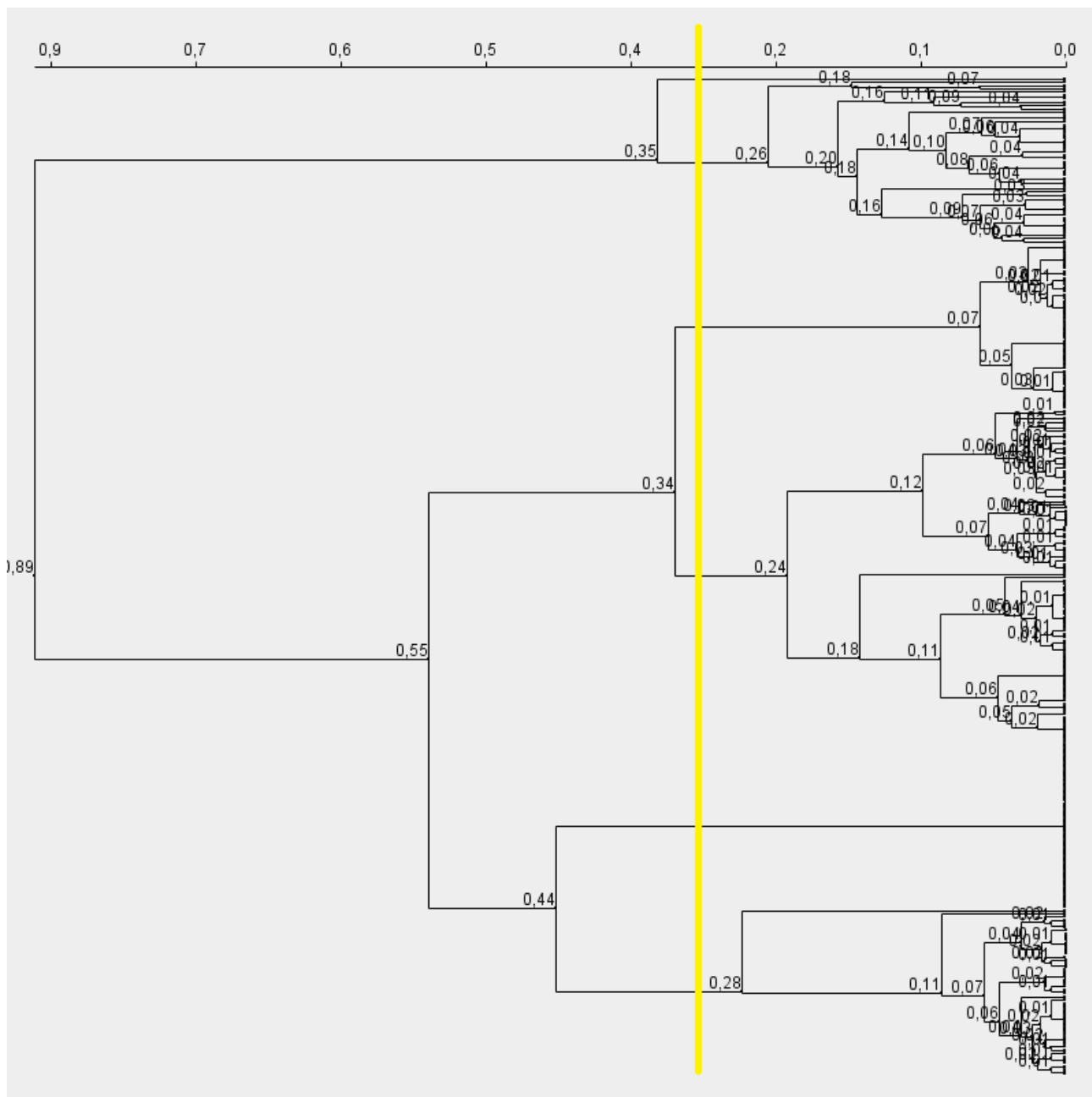
3.5.2. XPATH be indeksų



16. pav. Svetainės „9“ klasterių dendrograma, gauta naudojant XPATH metodą.

16 paveiksle pavaizduotoje dendrogramoje matyti ta pati problema, kaip ir ankstesniame pavyzdyje. Nukirtus medžio šakas ties ribine reikšme, taip kad būtų gauti, šiuo atveju, šeši klasteriai,

į kiekvieną klasterį patenka labai nevienodas skaičius dokumentų. Šios klasterių konfigūracijos indeksai yra tokie: *Dunn*: 1,12120, *Rand*: 0,65772, vidutinis grynumas: 0,83224.



17. pav. Svetainės „9“ klasterių dendrograma, gauta naudojant XPATH be indeksų metodą.

17 paveiksle pavaizduotoje dendrogramoje matyti, kad į kiekvieną klasterį patenka vienesnis skaičius dokumentų, nei taikant XPATH metodą, tačiau vis tiek nėra suklasterizuojama idealiai, kaip 15 paveiksle pavaizduotoje dendrogramoje. Problema kyla dėl to, kad kai kurie hipertekstinių dokumentų šablonai yra labai panašūs vienas į kitą. Šioje dendrogramoje galima aiškiai matyti tris klasterius, tačiau „teisingoje“ klasterių sąrankoje jų turėtų būti šeši. Vis dėlto, XPATH be indeksų metodas pagerino šios svetainės klasterių sąranką. Šios klasterių sąrankos indeksai yra tokie: *Dunn*: 1,20928, *Rand*: 0,83278, vidutinis grynumas: 0,9.

3.6. Rezultatų apibendrinimas

Darbe aprašyti keturi metodai, kuriais siekiama pagerinti geriausius literatūroje rastus metodus. Sukurti nauji metodų pagerinimai yra šie:

- 1) Atstumų skaičiavimas, remiantis XPATH, prieš tai pašalinus gretimų DOM medžio elementų indeksus iš XPATH.
- 2) XPATH rinkinio dalies pašalinimas prieš skaičiuojant atstumus.
- 3) XPATH kelių pradžios nukirtimas iki tam tikro gylio prieš skaičiuojant atstumus.
- 4) XPATH, kurie yra bendri visiems klasterizuojamiems dokumentams, pašalinimas prieš skaičiuojant atstumus.

Be to, darbo eigoje buvo sugalvotas metodas, kaip identifikuoti reikiamą kiekį klasterių iš klasterizavimo metu gautos hierarchinės klasterių struktūros. Šis metodas geriau identifikuoja klasterius, nei gerai žinomi metodai, tokie kaip klasterių parinkimo pagal ribinę reikšmę ir klasterių parinkimo pagal medžio viršūnės gylį.

Darbo metu buvo atlikti eksperimentai, kuriais siekiama patikrinti, ar sugalvoti metodai pagerina klasterizavimo kokybę. Klasterizavimo kokybė darbe vertinama trimis parametrais: *Rand* indeksas, *Dunn* indeksas, vidutinis grynumas. Eksperimentuose buvo tikrinti keturi aukščiau išvardinti metodai.

Atlikus eksperimentus paaiškėjo, kad XPATH be indeksų parinkimo metodas pagerina klasterizavimo kokybę, t.y., naudojant šį metodą, kokybę rodantys indeksai įgyja didesnes reikšmes, nei naudojant paprastus XPATH su indeksais arba lapų XPATH.

Antras, trečias ir ketvirtas metodai yra skirti pašalinti dokumentų dalis, kurios yra bendros visiems svetainės dokumentams. Nei vienas iš šių metodų neparodė akivaizdžios naudos, klasterizuojant hipertekstinius dokumentus, sugeneruotus iš šablonų. Tačiau šių metodų kombinacijos su lapų XPATH ir XPATH be indeksų metodais vis tiek duoda geresnius rezultatus, nei, paprastų XPATH su gretimų elementų indeksais metodas, kuris darbe naudojamas kaip pagrindinis.

Išvados

Darbe pasiūlyti keturi nauji metodų pagerinimai, skirti hipertekstinių dokumentų, sugeneruotų iš šablonų, klasterizavimui. Visi pasiūlyti metodai pranoksta lapų XPATH bei bazinį metodą, kuriame atstumai skaičiuojami naudojant elementų XPATH.

Labiausiai klasterizavimo kokybę pagerina elementų indeksų pašalinimas iš XPATH.

Bandymai pašalinti dokumentų dalis, kurios yra bendros visiems klasterizuojamiems dokumentams, nedavė pastebimos naudos klasterizavimo kokybei. Visais atvejais dokumentų dalių pašalinimas mažina skirtumus tarp skirtinguose klasteriuose esančių dokumentų. Netgi tiesiogiai pašalinus XPATH, kurie yra bendri visiems klasterizuojamiems dokumentams, gauti rezultatai nebuvo geresni nei klasterizuojant tuos pačius dokumentus, prieš tai nepašalinus bendrų XPATH. Taigi, pasiūlyti bendrų dokumentų dalių pašalinimo metodai nepagerino klasterizavimo kokybės. Pašalinus dalį dokumento, jo atstumas iki kitų dokumentų sumažėja. Šis atstumo sumažėjimas rodo, kad analizuojami metodai netiksliai identifikuoja dokumento dalį, kurią reikia pašalinti.

Hipertekstinių dokumentų klasterizavimui pagal struktūrinį panašumą, labai didelę įtaką turi patys klasterizuojami dokumentai, nes jų struktūra gali būti labai įvairi. Kadangi dažniausiai yra klasterizuojami konkrečios internetinės svetainės puslapiai, pasirenkant metodą reikėtų atsižvelgti į konkrečios svetainės puslapių specifiką.

Literatūra

- [1] Blanco L.; Dalvi, N.; Machanavajjhala, A. 2011. Highly Efficient Algorithms for Structural Clustering of Large Websites, in *WWW '11 Proceedings of the 20th international conference on World wide web*. New York: ACM, 437-446.
- [2] Broder, A. Z. 1997a. On the Resemblance and Containment of Documents, in *Proceedings of Compression and Complexity of Sequences 1997*. Salerno: 21-29.
- [3] Broder, A.Z.; Glassman, S.C.; Manasse, M.S.; Zweig, G. 1997b. Syntactic clustering of the web, in *Selected papers from the sixth international conference on World Wide Web*. Amsterdam: Computer Networks 1157–1166.
- [4] Buttler, D. 2004. A short survey of document structure similarity algorithms, in *IC '04: Proceedings of the International Conference on Internet Computing*. Las Vegas: 3–9.
- [5] Chakrabarti, D.; Kumar, R.; Punera, K. 2007. Page-level template detection via isotonic smoothing, in *WWW '07: Proceedings of the 16th International Conference on World Wide Web*. New York: ACM Press, 61–70.
- [6] Chakrabarti, D.; Mehta, R. 2010. The Paths More Taken: Matching DOM Trees to Search Logs for Accurate Webpage Clustering in *WWW '10 Proceedings of the 19th international conference on World wide web*. New York: 211-220.
- [7] Cruz, I.F.; Borisov, S.; Marks, M.A; Webbs, T.R.: 1998. Measuring structural similarity among web documents: preliminary results, in: *EP '98: Proceedings of the 7th international Conference on Electronic Publishing, Artistic Imaging, and Digital Typography*. Malo: 513 – 524.
- [8] Debnath, S.; Mitra, P.; Giles, C.L. 2005. Automatic extraction of informative blocks from webpages, in: *SAC '05: Proceedings of the 2005 ACM Symposium on Applied Computing*. New York: ACM Press, 1722–1726.
- [9] Flesca, S.; Manco, G.; Masciari, E.; Pontieri, L.; Pugliese, A. 2002. Detecting structural similarities between XML documents, in *Fifth International Workshop on the Web and Databases*. Rende:

- [10] Gibson, D.; Punera, K.; Tomkins, A. 2005. The volume and evolution of web page templates. In: *WWW'05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*. New York: ACM Press, 830–839.
- [11] Gottron, T. 2008. Clustering Template Based Web Documents, in *ECIR '08: Proceedings of the 30th European Conference on Information Retrieval*. Berlin: 40–51.
- [12] HTML 5.1. A vocabulary and associated APIs for HTML and XHTML, (W3C), [interaktyvus]. [Žiūrėta: 2013-06-03]. Prieiga internetu: <http://www.w3.org/TR/html51/>
- [13] Yi, L.; Liu, B.; Li, X. 2003. Eliminating noisy information in web pages for data mining, in *KDD '03: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 296–305.
- [14] Yossef, Z.; Rajagopalan, S. 2002. Template detection via data mining and its applications, in *WWW '02: Proceedings of the 11th International Conference on World Wide Web*. New York: ACM Press, 580–591.
- [15] Jaccard, P. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura, in *Bulletin de la Société Vaudoise des Sciences Naturelles* 37. 547–579.
- [16] Joshi, S.; Agrawal, N.; Krishnapuram, R.; Negi, S. 2003. A bag of paths model for measuring structural similarity in web documents, in: *KDD '03: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 577–582.
- [17] Kruskal, J.B. 1964. Nonmetric multidimensional scaling: A numerical method, in *Psychometrika*. Berlin: 115–129.
- [18] Lin, S.H.; Ho, J.M. 2002. Discovering informative content blocks from web documents, in *KDD '02: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 588–593.
- [19] Lindholm, T.; Kangasharju, J.; Tarkoma, S. 2006. Fast and simple XML tree differencing by sequence alignment, in *DocEng '06: Proceedings of the 2006 ACM Symposium on Document Engineering*. New York: ACM Press, 75–84.
- [20] Liu, B. 2007. Web Data Mining – Exploring Hyperlinks, Contents, and Usage Data. 1-st ed. Berlin: Springer.

- [21] MacQueen, J. B. 1967. Some Methods for classification and Analysis of Multivariate Observations, in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1*. Berkeley: University of California Press. 281–297.
- [22] Rand, W.M.: 1971. Objective criteria for the evaluation of clustering methods, in *Journal of the American Statistical Association*. 846–850.
- [23] Reis, D.C.; Golgher, P.B.; Silva, A.S.; Laender, A.F. 2004. Automatic web news extraction using tree edit distance, in *WWW '04: Proceedings of the 13th International Conference on World Wide Web*. New York: ACM Press, 502–511.
- [24] Shi, L.; Niu, C.; Zhou, M.; Gao, J. 2006. A DOM tree alignment model for mining parallel data from the web, in: *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*. Morristown: Association for Computational Linguistics 489–496.
- [25] Vieira, K.; Silva, A.S.; Pinto, N.; Moura, E.S. 2006. A fast and robust method for web page template detection and removal, in *CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management*. New York: ACM Press, 258-267

Priedai

1. Priedas. Naudota programinė įranga

MySql WorkBench

MySql Database Server

Eclipse

Java VM

MultiDendrograms

2. Priedas. Naudotos atviro kodo bibliotekos

Hierarchical-clustering

Crawler4j

JDBC

Jsoup

3. Priedas. Eksperimentams naudota mašina

Intel® Core™ i3 CPU 3.06GHz

8 GB RAM

Windows7 64bit operacinė sistema

JVM versija 1.7.0_17