Ville Pirttimäki

Yhteisöpalvelujen tägien hyödyntäminen ICT-alan muutosten ennakoinnissa ja havainnoinnissa

Tietotekniikan
pro gradu -tutkielma
17. huhtikuuta 2012

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Ville Pirttimäki
Yhteystiedot: ville.pirttimaki@jyu.fi
Työn nimi: Yhteisöpalvelujen tägien hyödyntäminen ICT-alan muutosten ennakoinnissa ja havainnoinnissa
Title in English: Using tags in social network services to notice and predict changes in the ICT field
Työ: Tietotekniikan pro gradu -tutkielma
Sivumäärä: 88
Tiivistelmä: Tägejä ja näiden emergenttiä ontologiaa, eli folksonomiaa on tutkittu jo vuodesta 2004 lähtien. Tänä aikana tägien käyttö on yleistynyt Internetin kautta tarjottavissa palveluissa ja kyseisten palveluiden suosio on kavanut, luoden suuren määrän tägidataa. Tätä dataa ollaan kuitenkin harvoin käytetty muuhun, kuin resurssien löytämiseen ja uudelleenlöytämiseen (jota varten data tietysti on luotukin). Tässä tutkielmassa pyritään luomaan katsaus tägidatan ja folksonomioiden analysointia varten kehitettyihin menetelmiin ja käyttämään kahta näistä valikoitua menetelmää ICT-alan termien analysointiin. Tavoitteena on löytää ja testata menetelmä tai menetelmiä, joiden avulla voidaan havaita ICT-alan muutoksia aikaisessa vaiheessa ja löytää epätriviaaleja yhteyksiä alan termien välillä.
English abstract: Tag and their emergent ontologies, folksonomies, have been studies all the way since 2004. In this time, the using of tags and Internet based services that use tagging have grown incresingly popular, creating vast amounts of tag data. However, this data is rarely used for anything else but discovering and rediscovering of resources (although this is the reason the data exists in the first place). This study aims to provide a brief introduction to the methods developed for analyzing of tag data and folksonomies, and to use selected two of these methods to analyze terms connected to the ICT field. The goal is to find and test a method or methods to be used for early detection of changes in the ICT field and to find non-trivial connections between the terms of the field.
Avainsanat: Delicious, folksonomia, tiedonlouhinta, tägi
Keywords: data mining, Delicious, folksonomy, tag

Sanasto

folksonomia Ihmisjoukon resursseille asettamista vapaavalintaisista tägeistä muodostuva emergentti rakenne, jossa on taustalla käyttäjien saamaan palautteeseen perustuva tägäyskäytäntöjen jonkinasteinen yhtenäistyminen. [Kos07]

graafi Matemaattinen malli, jossa tietyn joukon jäsenten (solmujen) välisiä yhteyksiä kuvataan (yksi- tai kaksisuuntaisilla) kaarilla.

hypergraafi Graafin yleistys, jossa yksi kaari voi yhdistää useampaa kuin kahta solmua.

ICT Information and Communication Technology. Tieto- ja viestintätekniikka.

metadata Data, joka kuvaa kohdedatan sisältöä tai informaatiolähteen ominaisuuksia, muotoa ja/tai ominaisuuksia.

ontologia Tietyn kohdealueen tai käsiterakenteen termien ja näiden välisten suhteiden kartoitus (ei yhteydessä ontologiaan filosofian alana).

resurssi Tässä tutkielmassa kaikille mahdollisille entiteeteille (yleensä WWW-sivu), joihin liittyvää metadataa on tarkoitus käsitellä, mutta joiden varsinaiseen sisältöön tai muotoon ei oteta kantaa.

taksonomia Termien välinen hierarkkinen luokitusjärjestelmä.

tägi (engl. *tag*) Objektia (lyhyesti, yleensä yhdellä sanalla) sanallisesti kuvaava metadata ilman eksplisiittistä ontologiaa.

tägäyspalvelu Erillinen palvelu, joka mahdollistaa tägien yhdistämisen resurssiin.

Merkistö

A: käyttäjäjoukko (actors) a_i: käyttäjä i (actor)
$C(x, y)$: painoarvofunktio luokituksessa
$c_{t|t'}$: approksimoi todennäköisyyttä, että tägi t kuuluu samaan konseptiin kuin tägi t' (SMM-mallin mukaan)
D: aikojen joukko d: ajan hetki
E: kaarijoukko
F: kolmipolvinen folksonomiajoukko
G: tägigraafi
I: resurssijoukko i_i: resurssi i
$I(x)$: informaatioarvofunktio i: indeksi a_i:lle ja p_i:lle
j: indeksi t_j:lle ja q_j:lle
K: konseptien lukumäärä
k_i ja k_j: muuttujia liitteen C todistuksessa
L: yhdessäesiintymien lukumäärä
n_x: tägin x esiintymislukumäärä
O: ontologia
p_i: resurssitodennäköisyys i SMM mallissa
q_j: tägitodennäköisyys j SMM mallissa
$R_{r\alpha}$: SMM mallin piilomuuttuja, todennäköisyys että konseptista z_α seuraa yhdessäesiintymä r
r: indeksi yhdessäesiintymiselle joukossa S
S: yhdessäesiintymien joukko
T: tägijoukko t_j: tägi j
w: painoarvo verkossa (t): iteraatioindeksi
Z: abstraktien konseptien joukko z_α: abstrakti konsepti α
α: indeksi π_α:lle ja z_α:lle
π_α: konseptin z_α todennäköisyys
$\mathcal{L}$: logaritminen uskottavuusfunktio
$n_x \| n_y$: tägien x ja y yhdessäesiintymien määrä
$\sum_{x:p(x)}$: summa yli kaikkien x:ien, joilla $p(x)$ on tosi

Kuvat

Taulukot

Algoritmit

Sisältö

1 Johdanto

Yksi monista ICT-alan yritysten haasteista on pysyä mukana alalle ominaisessa jatkuvassa ja nopeatempoisessa muutoksen ilmapiirissä. Alalla tehdään jatkuvasti tutkimusta ja kehitetään uusia teknologioita, joiden tunteminen on alan yrityksille elintärkeää. Jotta yritys pystyisi hyödyntämään uusia innovaatioita ja teknologioita, vaaditaan näihin liittyvää osaamista. Näin tämän osaamisen hankinta nousee tärkeäksi haasteeksi alan yrityksille.

Raportissaan [Sam10, luku 3] Nataliia Samoilenko kuvaa tarkemmin järjestelmän, jonka avulla yritys pystyy saamaan ja omaamaan tietämystä, määrittelemällä järjestelmään seuraavat osat: yrityksen, osaamisen, henkilön, tiedon lähteen ja koulutustapahtuman, sekä näiden keskinäiset suhteet. Lyhyesti kuvattuna tässä mallissa yritys työllistää henkilön. Tämä henkilö omaa jonkin osaamisen. Osaamista voidaan saada koulutustapahtumissa (jotka voidaan yleistää kattamaan henkilön omatoiminen oppiminen) ja näissä tapahtumissa osaamista voidaan myös täydentää. Lisäksi koulutustapahtumat perustuvat tiedonlähteisiin, joskin tämä yksityiskohta ei ole tarkastelun tässä vaiheessa merkittävä. Merkittävää tässä vaiheessa on huomata, että yrityksen omaama osaaminen on riippuvainen (tämän palkkaamien henkilöiden kautta) koulutustapahtumista. Ja täten myös edellä mainittu ICT-alan kehityksessä mukana pysyminen on riippuvainen koulutustapahtumista, joissa yrityksen työntekijät käyvät.

Jyväskylän yliopiston järjestämä PROFIT-projekti pyrkii avustamaan yrityksiä henkilöstönsä koulutuksessa järjestämällä koulutustapahtumia, joita yritykset määrittävät järjestettäviksi. Tällä lähestymistavalla on kuitenkin se heikkous, että koulutustapahtumat suunnitellaan ja järjestetään täysin reaktiivisesti. Projektin kyky vastata yritysten tarpeisiin (pelkkien pyyntöjen sijaan) paranisi, mikäli koulutustarpeita pystyttäisiin ennakoimaan ICT-alan kehityksen mukaan. Tällöin projekti pystyisi ehdottamaan yrityksille koulutuksia, joiden avulla yritykset pystyisivät lisäämään kompetenssiaan, ilman että projektin tarvitsee erikseen odottaa pyyntöä näiden koulutusten järjestämiseen.

Toisaalla, Internetissä, tekniikan kehityksestä seurannut sivustojen ja palveluiden kehittäjien ajattelutavan muutos, jonka Darcy DiNucci nimesi vuonna 1999 kirjoituksessaan [DiN99] nimellä Web 2.0, on yleistynyt ja ajattelutavan mukaisten sivustojen käyttäjäkunta on kasvanut valtavaksi. Web 2.0 -nimi kokoaa allensa useita suunnittelufilosofioita, jotka poikkeavat radikaalisti ennen 2000-lukua käytetyistä teknologian käyttötavoista. Keskeisenä ajatuksena Web 2.0 -mallissa on käyttäjiin ja heidän tuottamaan sisältöön tukeutuminen. Ideologian yleistystä auttoi myös se, että tämä uusi lähestymistapa todettiin useissa eri käyttöyhteyksissä toimivaksi, hyödylliseksi ja kannattavaksi. Web 2.0 -buumin seurauksena erilaisten yhteisöpalveluiden (engl. *social network sites*) ja muiden yhteisön osallistumista hyödyntävien sivujen määrä ja suosio on noussut räjähdysmäisesti 2000-luvulla.

Niin Samoilenkon raportissa kuin tässä tutkielmassa tarkoituksena on tutkia mahdollisuutta hyödyntää yhteisöpalveluissa, etenkin näissä sivustoissa käytetyissä tägeissä, piilevää joukon viisautta[1] (engl. *wisdom of the crowd*) pyrkiessä ennakoimaan ICT-alan koulutustarpeita. Konseptin pohjalla oleva idea tiivistyy hyvin Shirkyn tekstin *Ontology is Overrated: Categories, Links, and Tags* [Shi05] väliotsikossa, jossa sanotaan "The only group that can categorize everything is everybody", eli "vain kaikki pystyvät kategorisoimaan kaiken".

Tässä tutkielmassa yritetään käytännön toteutuksen eli konstruktion avulla hyödyntää yhtä suosituimmista tägeihin pohjautuvista yhteisöpalveluista sekä yhteisöpalveluiden tägeihin pohjautuvia tutkimuksia ja malleja ICT-alan koulutustarpeiden ennakoimiseksi. Tavoitteena on luoda sovellus, joka analysoi Delicious[2] -palvelussa tapahtuvaa tägäystä ja pyrkii olemassa olevaan tutkimukseen pohjautuvien algoritmien avulla luomaan kahdenlaista tietoa. Ensinnäkin pyritään löytämään tunnettuun aiheeseen vahvasti liittyviä, ennalta tuntemattomia aiheita. Toisekseen yritetään tunnistaa uusia, nousevia tekniikoita mahdollisimman varhaisessa vaiheessa.

Tunnettuun aiheeseen liittyviä aiheita toivotaan pystyttävän käyttämään hyväksi kahdella eri tavalla suunnitellessa ICT-alan koulutuksia. Ensinnäkin toivotaan tuloksena olevan aiheeseen kuuluvia, pienempiä ala-aiheita, joita voidaan ehdottaa sisällytettäväksi koulutukseen. Toisekseen tulokseksi toivotaan myös laaja-alaisempia konsepteja, joiden alle tunnettu aihe kuuluu. Näitä voitaisiin taas ehdottaa koulu-

[1] James Surowieckin vuonna 2004 julkaisema kirja *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations* toi termin aikanaan julkiseen tietoisuuteen.

[2] `http://www.delicious.com/`

tusaiheeksi, jos on olemassa tunnettu aihe, jolla tiedetään olevan koulutustarvetta, mutta aihe sinällään todetaan liian suppeaksi koulutustapahtuman järjestämiseksi. Nousevan suosion saamien tekniikoiden joukosta taas toivotaan löytyvän ehdotettavaksi koulutusaiheita, joita kohtaan yrityksillä on kiinnostusta, mutta joita kohtaan kiinnostusta ei olla vielä ehditty ilmaista.

Tutkielman alussa, luvussa 2, käydään läpi terminologiaa ja tutkimussuuntia, joihin tutkielma pohjautuu. Luvussa 3 keskitytään tarkemmin tiedon- ja tietämyksenlouhintaa, jota tutkielmassa periaatteessa suoritetaan. Luvussa 4 käydään läpi muutamia menetelmiä, joiden avulla tägidataa voidaan tutkia ja louhia. Luku 5 kuvaa tarkemmin tutkielman yhteydessä luodun konstruktion ja tämän käyttämät menetelmät toteutuksineen. Lopuksi, luvussa 6 tarkastellaan kerättyä dataa ja tästä datasta saatuja tuloksia.

2 Tausta

Ennen kuin voimme alkaa ratkaisemaan johdannossa esitettyjä tavoitteita, tulee ensin varmistaa että tutkielman lukijalla on riittävä ymmärrys käsiteltävistä konsepteista ja datasta, jota tutkielmassa käytetään hyväksi. Suurin osa käytetystä sanastosta oletetaan tunnetuksi ja näihin annetaan vain pikainen selitys ja/tai englanninkielinen käännös joko sanan ensimmäisen käyttökerran yhteydessä tai tutkielman alussa olevassa sanastossa. Näiden määritelmien ei ole niinkään tarkoitus selittää termiä lukijalle, vaan varmistaa, että sekä kirjoittaja että lukija ymmärtävät termit tämän tutkielman yhteydessä samoilla tavoilla.

On kuitenkin olemassa joitakin tutkielman kannalta keskeisiä käsitteitä, joihin on hyvä tutustua tarkemmin ennen näiden käyttöä. Tässä luvussa ja osittain myös luvussa 3 pyritään antamaan näistä käsitteistä tarvittava (joskaan ei kattava) tietämys. Lisäksi tämän luvun lopuksi käydään läpi tutkielman yhteydessä käytettyjä tieteellisiä lähteitä samasta aihealueesta tehdyistä tutkimuksista. Läpikäynnin otanta on varsin hajanainen eikä yritäkään korvata systemaattista katsausta aihealueen tutkimuksiin, mutta antaa toivon mukaan lukijalle yleisnäkemyksen aiheeseen liittyvän tutkimuksen tasosta ja näin myös kontekstin, johon liittää tämä tutkielma.

2.1 Metadata

Yksinkertainen metadatan määritelmä on usein “dataa datasta”, joskin David Haynes [Hay04, s. 6-8] pitää tätä määritelmää liian yksinkertaisena kuvaamaan metadataan liittyvää monimutkaisuutta ja käyttömahdollisuuksien kirjoa. Lisäksi hänen mielestään useat aiemmat määritelmät ovat liian sidottuja tiettyyn käyttötapaan. Hänen määritelmänsä mukaan metadata on dataa, joka kuvaa kohdedatan tai informaationlähteen sisältöä, muotoa tai ominaisuuksia. Lisäksi hän listaa metadatalle seuraavat käyttötarkoitukset [Hay04, s. 15-17]: 1) resurssin kuvaus, 2) informaation haun avustus, 3) resurssien hallinta, 4) omistajuuksien ilmaisu, ja 5) järjestelmien välisen yhteistoiminnan avustus.

Metadatan avulla pyritään ratkaisemaan ongelmia, jotka ilmenevät käsiteltäessä datajoukkoja, joissa käsittelyn vaatiman datan johtaminen kohdedatasta on joko

Ulkoinen	UDDI Open Directory Project	RSS Annotea
Sisäinen		HTML <META> RDDL
	Keskitetty	**Hajautettu**

Taulukko 2.1: Kaksiulotteinen metadatan jaottelu esimerkeillä

vaikeaa (esim. datan määrän takia) tai mahdotonta (joko koneellisesti tai jopa ihmisvoimin). Metadata ratkaisee tämän ongelman erottamalla hyödynnettävän datan kohdedatasta. Käytännössä metadataa käytetään mm. hakuaikojen parantamiseen, digitaalisten objektien hallintaan, datan aitouden varmistamiseen ja järjestelmien väliseen yhteistoimintaan. [Hay04, s. 11-12]

Jotta saataisiin yleiskäsitys siitä, millaisissa muodoissa metadataa ilmenee, tarkastellaan lähteessä *Professional XML Meta Data* [ARML+01] tehtyä erottelua eri tyyppisten metadatan muotojen välillä. Lähde erottelee resurssin sisäisen metadatan, joka on upotettu resurssiin itseensä, ja ulkoisen metadatan, joka on olemassa resurssista erillään, mutta on linkitetty resurssiin jollain tavoin. Lisäksi malli erottelee keskitetyn metadatan, jossa eri resursseihin liittyvä metadata on saatavilla eri tietovarastoista, ja hajautetun metadatan, jossa metadatalle ei ole vain yhtä lähdettä. Taulukossa 2.1 nähdään muutamia metadatan esitysmuotoja jaoteltuna näiden erotteluiden mukaan. On huomattavaa, että keskitetylle sisäiselle metadatalle ei löydy esimerkkiä. Lähde toteaakin, että sisäinen metadata on määritelmällisesti aina hajautettua, koska metadatajoukko on jaettu eri resurssien kesken.

Näiden lisäksi Hyvönen et al. [HHV02, 3.1.2] täydentävät jaottelua lisäämällä eron implisiittisen metadatan ja eksplisiittisen metadatan välille. Implisiittisessä metadatan esityksessä, kuten esimerkiksi skeematon tai muuten määrittelemätön XML, datan merkitys ei ole suoraan sidottuna sen rakenteeseen. Eksplisiittisessä esityksessä datan merkitys on taas algoritmisesti tulkittavissa ja muokattavissa, erottaen näin metadatan semantiikan datan käsittelyalgoritmeista.

2.1.1 Metatietorakenteita

Mihin metadata onkaan tallennettuna, hajautettuna sisäisenä metadatana tai keskiettynä ulkoisena, vaatii se myös tietorakenteen johon metadatan data voidaan tallentaa. Tämä rakenne voi olla yksinkertainen avain-arvo pari tai monimutkainen

graafirakenne. Metadatan tietorakenteiden pohjaksi (yhtenäisen käsittelyn mahdollistamisen ohella) on kehitetty useita standardeja, kuten avain-arvo pareihin pohjautuva Dublin Core sekä graafirakenteinen W3C-organisaation määrittelemä RDF (Resource Description Framework) [MM].

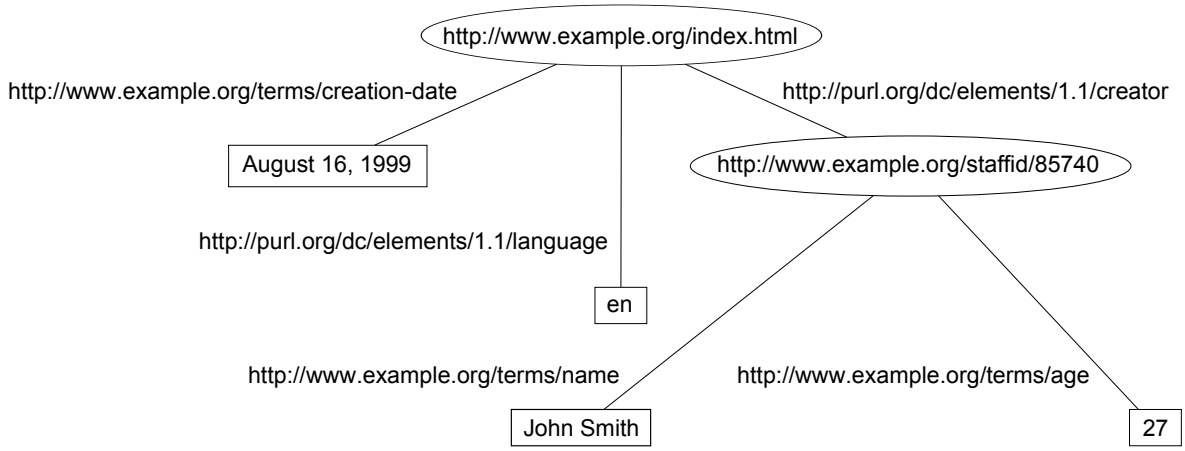
Edellä Dublin Core[1] kuvailtiin tietorakenteeksi. Tämä on jokseenkin harhaanjohtavaa, koska Dublin Core koostuu pääasiassa 22 elementistä (engl. *element*) eli avainsanasta, näihin liitettävien annettavien arvojen formaattisuosituksista ja näitä tarkentavista määre-elementeistä (engl. *qualifiers*). [Hil, luvut 4 ja 5] Näin ollen se on ennemminkin avainsanarajoitin kuin tietorakenne. Metadatan rakennetta Dublin Core ei rajoita, sillä kaikki elementit ovat vapaaehtoisia, toistettavia ja järjestyksellä ei ole merkitystä. [Hil, luku 3] Rajoitetun avainsanaston lisäksi Dublin Corea varten on määritelty[2] syntaksit datan esittämiseksi tekstimuodossa, HTML-dokumentissa, XML-dokumentissa sekä seuraavaksi käsiteltävässä RDF-metatietorakenteessa.

RDF perustuu URI:lla (Uniform Resource Identifier) kuvatun *resurssin*[3] ja joko URI:lla tai merkkijonoa esitetyllä literaalilla kuvatun *objektin* liittämiseen yhteen *predikaatilla*. URI-muotoisen objektin salliminen notaatiossa mahdollistaa objektin tarkemman kuvaamisen subjektin tavoin, liittämällä URI-muotoiseen objektiin lisää objekteja RDF-notaation avulla. Kuva 2.1 esittää esimerkin URL:lla kuvattuun Internet-sivuun `http://www.example.org/index.html` liitettävästä, sekä rekursiivisesti tämä sivun luojaan liitettävästä metadatasta. RDF tukee myös ominaisuuksia, kuten nimettömiä solmuja, ryhmien ja vaihtoehtojen listauksia muutamia esimerkkejä mainitakseni. Tietorakenteena RDF-data muodostaa kolmen elementin tripleteistä (subjekti, predikaatti, resurssi) koostuvan graafin, jossa yhden tripletin subjekti toimii toisen tripletin resurssina. Graafin tallentamiseksi voidaan käyttää esimerkiksi XML pohjaista RDF/XML-notaatiota, joka kuvaa RDF-rakenteisen metadatan XML-muotoiseksi ja täten helposti tallennettavaksi, siirrettäväksi ja uudelleentulkittavaksi. Kuvan 2.1 mukainen data on esitettynä tässä muodossa kuvassa 2.2. Näiden yksityiskohtiin ei tämän tutkielman osalta paneuduta, sillä tämä luvun tarkoitus on vain mainita esimerkkejä metatietorakenteista.

[1] `http://dublincore.org/`

[2] `http://dublincore.org/specifications/`

[3] "Resurssi" on tässä tutkielmassa myöhemmin käytettävä termi. *RDF Primer* käyttää termiä *subjekti* kuvaamaan samaa asiaa.



Kuva 2.1: Esimerkki siitä, kuinka RDF liittää objekteja resurssiin

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:dc="http://purl.org/dc/elements/1.1/"
             xmlns:exterms="http://www.example.org/terms/">

    <rdf:Description rdf:about="http://www.example.org/index.html">
         <exterms:creation-date>August 16, 1999</exterms:creation-date>
         <dc:language>en</dc:language>
         <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.example.org/staffid/85740">
         <exterms:name>John Smith</exterms:name>
         <exterms:age>27</exterms:age>
    </rdf:Description>
  </rdf:RDF>
```

Kuva 2.2: Esimerkki RDF/XML-notaatiosta

2.1.2 Taksonomiat ja ontologiat

Metadatan avainten ja rakenteen standardoimisen lisäksi joskus on hyödyllistä rajoittaa myös metadatan arvoja ennalta määrätyn standardin mukaiseksi. Tämän tutkielman kannalta ei ole merkityksellistä tutustua tiettyihin arvoja rajoittaviin standardeihin. Mutta tulevien aiheiden kannalta on hyvä tutustua kahteen termijoukkoa luokittelevaan järjestelmätyyppiin, taksonomiaan (engl. *taxonomy*) ja ontologiaan (engl. *ontology*).

Joukon taksonomia määritellään yleensä joukkoa koskevien kategorioiden joukoksi ja tavaksi, jolla joukko jaetaan näihin kategorioihin kattavasti ja yksikäsitteisesti. Määritelmään liitetään usein myös ryhmien jakaminen alaryhmiin. Ryhmitte-

lytapana taksonomia on iäisyyksiä vanha ja sitä käytetään usein esimerkiksi kirjastojen kirjojen ryhmittelyyn hyllyille. Kuuluisin taksonomia lienee eläin- ja kasvikuntaa kategorioiva Linnaealainen taksonomia, joka juontaa juurensa Carl Linnæusin teoksesta Systema Naturæ. Linnaelainen taksonomia on niinkin kuulu esimerkki taksonomiasta, että useimmat sanakirjamääritelmät sanasta viittaavatkin pääasiallisesti eläin- ja kasvikunnan luokittelun tieteeseen.

Taksonomian käsitettä voidaan laajentaa sallimaan monimuotoisempia relaatioita käsitteiden välille, jolloin usein puhutaan ontologiasta. Suosittu ja tiivis ontologian määritelmä tietotekniikan yhteydessä[4] on Tom Gruberin [Gru93] määritelmä "a specification of a conceptualization". Hän myöhemmin tarkensi määritelmää [Gru09][5] seuraavaksi (vapaasti käännettynä):

> Tietokone- ja informaatiotieteiden yhteydessä ontologia määrittää edustavien primitiivien joukon jolla tietämysalue tai diskurssi kuvataan. Edustavat primitiivit ovat tyypillisesti luokkia, ominaisuuksia ja suhteita. ...

Tärkeä huomioitava seikka on, että määritelmässä luokkien väliset suhteet ovat oma, vapaasti määriteltävä primitiivinsä. Verrattuna taksonomiaan, ontologia on vapaampi kuvaamaan termien suhteita muutenkin kuin hierarkkisesti, mahdollistaen näin käsitteiden välisten suhteiden monipuolisemman kuvauksen. Ontologioita käytetään lähteen [McG03] mukaan mm. rajoitetun sanaston (engl. *controlled vocabulary*) lähteinä tai sovellusten välisen kommunikoinnin apuna.

2.2 Tägit, tägäyspalvelut ja folksonomiat

"Tägi" (engl. *tag*) on luultavasti tutkielman keskeisin käsite. Kaikki tutkielmassa esitetyt dataa prosessoivat menetelmät käsittelevät tägejä. Luvussa 3 esiteltävän tietämyksenlouhintaprosessin toteutus ottaa syötteenä tägejä ja suurin osa tuloksena tulevasta datasta on myös tägejä tai tägipohjaista dataa. Tästä syystä on tarkoituksenmukaista selvittää tarkasti tämän käsitteen merkitys ja muut siihen liittyvät käsitteet.

Aloitetaan tarkastelemalla tägiä puhtaana datana. Useimmiten tägillä tarkoitetaan tämän kohdetta kuvaavaa lyhyttä tekstiä, esimerkkeinä `article`, `toread` ja

[4] Sanalla ontologia on oma merkityksensä filosofian kentällä, jossa se kuvaa olemassaoloa systemaattisesti käsittelevää filosofiaa.

[5] `http://tomgruber.org/writing/ontology-definition-2007.htm`

`cloud`. Näin ollen tägistä käytetään myös termiä “avainsana”. Tägin tarkka määritelmä vaihtelee eri käyttökohteiden välillä, mutta yleisiä piirteitä ovat ihmisluettavuus, lyhyys ja tägien keskenäisten eksplisiittisten rakenteiden puute. Esimerkkinä määritelmän vapaudesta mainittakoon että Gupta et al. [GLYH10, luku 1] sallivat Facebookissa[6] viestiin yhdellä hiiren painalluksella liitettävän ”Like”-kommentin määrittelemisen tägiksi. Tässä tutkielmassa tägejä käsitellään lyhyinä (tutkielmaa varten kerätyssä datassa tägin keskimääräinen pituus on 9,84 merkkiä), vapaamuotoisina merkkijonona, jotka on liitetty resurssiin. Vapaamuotoisuudella tarkoitetaan, että merkkijonon muotoa ei rajoiteta muutamia sääntöjä lukuun ottamatta. Yleisimmät rajoitukset liittyvät merkkijonon maksimipituuteen sekä rajoituksiin sallitussa merkistössä (ja nämäkin kiellot rajoittuvat yleensä tägien tallennuksessa käytettyihin erottimiin, kuten pilkkuun tai välilyöntiin).

Tägäyspalveluksi kutsutaan tässä tutkielmassa Internet-sivustoa, joka tarjoaa mahdollisuuden tägien liittämiseen resursseihin (esim. linkkeihin, musiikkiin, kuviin tai videoihin). Tutkiessa mitä metadata-arkkitehtuuria tägit edustavat tägäyspalvelujen osana luvussa 2.1 kuvattujen näkökulmien mukaan, voidaan voidaan sanoa että tägit ovat

- implisiittistä metadataa, koska (yksinkertaisesta rakenteestaan huolimatta) tägien varsinainen merkitys on hyvin tulkinnanvaraista ja tägien tulkitseminen koneellisesti on huomattavan haastava tehtävä,
- ulkoista metadataa, koska se on tallennettuna resurssin ulkopuolelle[7], ja
- (palvelukohtaisesti) keskitettyä metadataa, joskin tämä on hiukan tulkinnanvarainen määritelmä joka riippuu siitä, voidaanko tägäsypalvelu kokonaisuudessaan laskea asiakasohjelman kannalta yhdeksi lähteeksi, mikäli metadata on jakautunut palvelulla usealle sivulle.

Gupta et al. [GLYH10, 2.1 ja 2.2] listaavat tägien eri tyyppejä ja käyttötarkoituksia. Tägäyksen (eli tägien liittämiseen objektiin) motivaatioita (tämän lähteen mukaan) ovat resurssien uudelleen löytäminen tai jakaminen muille käyttäjille, huomion haku, leikki- tai kilpailumielinen käyttäytyminen, oman suhteen resurssiin

[6] `http://www.facebook.com/`

[7] Tämä ei ole tägeillä yleinen ominaisuus, vaan vaatii tägäyspalvelun kaltaisen palvelun, joka yhdistää tägin resurssiin. Muutoin tägit ovat yleensä resurssiin muun metadata ohessa tallennettua sisäistä metadataa.

ilmaisu, mielipiteen ilmaisu, tehtävien organisointi (joka johtaa hyvin yleisiin tägeihin `toread` ja `todo`), sosiaalinen viestintä ja jopa rahallinen motivaatio (esim. Squidoo[8] ja Amazon Mechanical Turk[9] maksavat tägäystoiminnasta). Eri käyttötarkoituksista seuraa luonnollisesti tyypeiltään erilaisia tägejä. Hyvin karkeasti ottaen tägit voidaan Körnerin [KÖ9] mukaan jakaa resursseja kategorisoiviin tägeihin ja resursseja kuvaaviin tägeihin. Gupta et al. tekevät tägien tyypeistä hiukan hienojakoisemman jaon, jossa tägit jaetaan seuraaviin ryhmiin:

- resurssin sisältöä kuvaavat tägit (esim. `cars`, `odyssey`, `batman`)
- resurssin kontekstia kuvaavat tägit (esim. `san-francisco`, `iccsm-2011`, `2005-10-19`)
- resurssin ominaisuuksia kuvaavat tägit (esim. tekijä tai tägit kuten `funny` tai `inpsirational`)
- resurssin omistajuutta kuvaavat tägit (esim. `msdn`, `jyu`, `spotify`)
- henkilökohtaisia mielipiteitä kuvaavat tägit (esim. `funny`, `cool`, `awful`, `boring`)
- henkilökohtaiseen organisointiin tarkoitetut tägit (esim. `toread`, `mywork`)
- resurssin tarkoitusta kuvaavat tägit (esim. `music-recommendations`, `learn-latex`)
- resurssiin liittyviä faktoja kuvaavat tägit
- henkilökohtaiset tägit, jotka on tarkoitettu vain tägääjän omaan käyttöön
- metatägit (Esimerkiksi Flickr-palvelussa[10] on yleisessä käytössä tägi `sometaithurts` eli ”so meta it hurts” kuvaamaan kuvia Flickr-palvelusta itsestään ja sen käyttäjistä.)
- ryvästävät tägit, joiden avulla luodaan ryhmityksiä (Esimekriksi Delicious-palvelussa on yleistä, että resurssin URL:n alkuosaa käytetään ryhmittävänä täginä, esimerkkitäginä `http://www.microsoft.com`)

[8] `http://www.squidoo.com/UsingSquidooToRaiseMoney`
[9] `https://www.mturk.com/mturk/welcome?variant=worker`
[10] `http://www.flickr.com/`

Huomaa että ominaisuuksia kuvaavissa tägeissä on tägejä, jotka voidaan ryhmitellä sisältönsä perusteella myös mielipiteitä ilmaiseviksi tägeiksi. Myös muissa ryhmissä on nähtävissä tägien päällekkäisyyksiä. Usein näiden samojen tägien käytössä on kuitenkin löydettävissä eri motivaatioita, jotka tukevat ryhmien erillisyyttä. Esimerkiksi henkilökohtaisia mielipiteitä ilmaisevat tägit johtuvat itseilmaisun halusta ja ovat enimmäkseen kuvaavia, kun taas samankaltaiset ominaisuuksia kuvaavat tägit luodaan usein kategorisointitarkoituksessa. Lisäksi on hyvä huomata, kuinka tägien eri käyttötavat kattavat useita Haynesin määrittelemiä metadatan käyttötarkoituksia (käytännössä kaikki paitsi viimeisen). Tämä on huomionarvoinen ominaisuus, sillä yleensä metadatan käyttötarkoitus on eksplisiittisesti tiedossa. Tägit ovat tässä poikkeuksellisia, sillä eri käyttötarkoituksia varten annetut tägit on vaikeaa, jos ei mahdotonta erottaa toisistaan.

Myös tägäyspalveluilla on useita ominaisuuksia, jotka tulee ottaa huomioon palveluita ja näistä saatavaa dataa käsitellessä. Marlow et al. [MNBD06, 4.1] listaavat useita ominaisuuksia, jotka vaikuttavat palvelussa tehtävään tägäykseen.

- Tägäysoikeudet
  Kenellä on oikeus liittää resurssiin tägejä? Tiukimmillaan tämä oikeus on vain yhdellä käyttäjällä tai hyvin rajatulla käyttäjäjoukolla, joko resurssin omistajalla tai ylläpidolla. Vapaimmillaan jokainen käyttäjä voi tägätä kaikkia resursseja vapaasti.

- Tägien ryhmittäminen
  Jaettavissa kahteen malliin.
  Joukko-mallissa (engl. *set-model*) resurssiin liitetään yksi tägijoukko, joka sisältää kaikki resurssiin liitetyt tägit, eliminoiden mahdolliset duplikaatit. Tätä mallia käytetään yleensä rajattujen tägäysoikeuksien yhteydessä. Lisäksi tägillä ja tägäyksen suorittajalla ei liitetä tägiin taikka liitosta ei korosteta (tägäyksen suorittaja on löydettävissä esimerkiksi muutoslokeista).
  Monijoukko-mallissa (engl. *bag-model* tai *multiset-model*) tägäysprosessi ei eliminoi duplikaatteja taikka duplikaatit eliminoidaan vain käyttäjäkohtaisesti. Tämä malli on yleisesti käytössä järjestelmissä, jossa tägäysoikeudet ovat vapaampia. Lisäksi malli sopii järjestelmään jossa tägäyksen ja tämän suorittaneen käyttäjän yhteyttä korostetaan.

- Tägäyksen tuki
  Järjestelmä voi tukea tägäyksen suoritusta useilla eri tavoilla tai antaa käyttä-

jän suorittaa tägäyksen täysin sokeasti. Mahdollisia tukijärjestelmiä ovat esimerkiksi erilaiset tägien suosittelujärjestelmät, yleensä samankaltaisiin resursseihin liitettyjen tägien perusteella, taikka monijoukko-mallia käyttävissä järjestelmissä resurssiin jo liitettyjen tägien ehdotus.

- Resurssin tyyppi
  Merkittävin jaottelu tässä ominaisuudessa on, onko resurssi tekstipohjainen vai ei. Tekstipohjaisten resurssien tägäyksessä tekstissä ilmenevien termien löytyminen tägeissä on varsin todennäköistä. Sama pätee jossain määrin myös ääntä sisältäviin resursseihin kuten musiikki tai videot, jos tämä sisältää puhetta.

- Resurssin lähde
  Resurssi voi olla järjestelmässä itsessään (esim. ESP Game[11], Last.fm[12], Yahoo! Podcasts[13]), käyttäjien järjestelmään lisäämä (esim. YouTube[14], Flickr, Upcoming[15]) tai yleisesti saatavilla oleva resurssi, johon järjestelmässä vain viitataan (esim. Delicious). Huomaa, että teknisestä näkökulmasta viimeinen ryhmä on identtinen toisen ryhmän kanssa, jos sen resurssiksi lasketaan linkitetyn sivun sijaan itse linkki. Erotus on kuitenkin hyvä pitää mielessä, kun pohditaan tämän jaottelun vaikutuksia järjestelmässä esiintyviin tägeihin.

- Resurssien liittyvyys
  Resurssit voivat liittyä toisiinsa eri tavoin (joko järjestelmässä itsessään tai sen ulkopuolella), jotka joko johtavat tägien samankaltaisuuteen järjestelmässä tai mahdollisuuteen löytää toisiinsa liittyviä resursseja järjestelmän ulkopuolella eriävistä tägeistä huolimatta. Marlow et al. luokittelevat liittyvyystyypit karkeasti ryhmiin linkitetyt, ryvästetyt ja liittymättömät.

- Käyttäjien liittyvyys
  Kuten resurssien, myös käyttäjien välillä voi olla useanlaisia liittyvyyssuhteita. Järjestelmän ulkopuolella olevien liittyvyyksien havaitseminen on kuiten-

[11] Poistunut verkosta, kuvaus järjestelmästä löytyy esim. sivulta `http://en.wikipedia.org/wiki/ESP_game`

[12] `http://www.last.fm/`

[13] Poistunut verkosta lokakuussa 2007

[14] `http://www.youtube.com/`

[15] `http://upcoming.yahoo.com/`

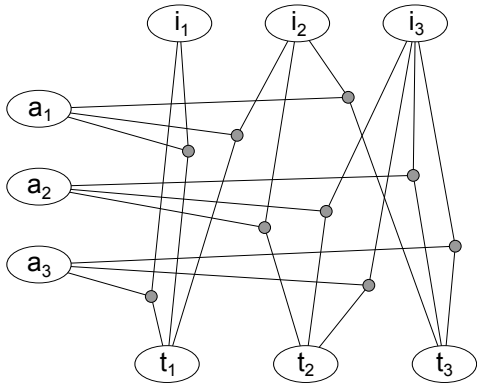
kin hankalampaa kuin resurssien tapauksessa, joten käyttäjien liittyvyyssuhteita tarkastellessa ei yleensä ole käytännöllistä ottaa huomioon järjestelmän ulkopuolisia liitoksia. Kuten resurssienkin kohdalla, Marlow et al. luokittelevat käyttäjien liittyvyystyypit karkeasti ryhmiin linkitetyt, ryvästetyt ja liittymättömät.

Kaikkien näiden ryhmittelyiden tarkoituksena on antaa lukijalle kuva erityyppisten tägien ja tägäyspalveluiden kirjosta. Tämä on tärkeä asia ottaa huomioon, sillä tägejä analysoidessa, muuten kuin hyvin yleisellä tasolla, ollaan usein kiinnostuneita vain tietyntyyppisistä tägeistä. On kuitenkin epätodennäköistä että tägäyspalvelu sisältäisi minkäänlaista erottelua erityyppisten tägien välillä. Näin ollen tägidataa analysoitaessa on menetelmien valinnassa, suorituksessa ja lopputulosten analysoinnissa otettava huomioon, että kaikki tägit eivät todennäköisesti tule olemaan haluttua tyyppiä. Menetelmien tulee joko pystyä tekemään erottelu erityyppisten tägien välillä ja karsimaan "väärän" tyyppiset tägit pois tarkasteltavasta datajoukosta, tai oltava tarpeeksi häiriönkestävä jotta eri tavoin käytettävät tägit eivät aiheuta ongelmia.

Tutkittaessa tägeihin liittyviä emergenttejä ontologioita, puhutaan yleensä folksonomioiden (engl. folksonomy) tutkimisesta. Sana juontaa juurensa Information Architecture Institution (silloin Asylomar Institute for Information Architecture) postituslistalla heinäkuussa 2004 käytyyn keskusteluun Furl, Flickr ja Delicious[16] -sivustojen tavasta käyttää tägejä informaation järjestämiseen. Thomas Vander Wal määritteli tuon keskustelun aikana ensimmäisen kerran termin "folksonomy". Muutamaa päivää myöhemmin Gene Smith esitteli termin blogissaan, josta termi levisi laajempaan käyttöön [VW]. Sanana "folksonomia" on sanojen "folk" (eli "kansa") ja "taxonomy" yhdistelmä.

Folksonomian täyteen formaaliin kuvaamiseen tarvitaan kolme joukkoa: $A = \{a_1, \ldots, a_l\}$ kuvaamaan käyttäjiä, $T = \{t_1, \ldots, t_j\}$ kuvaamaan tägejä ja $I = \{i_1, \ldots, i_i\}$ kuvaamaan tägättäviä resursseja, kuten kuvia tai linkkejä. Näiden avulla voidaan muodostaa folksonomiaa kuvaava joukko $F \subseteq A \times T \times I$. Kuva 2.3 kuvaa yhtä mahdollista hypergraafin esitystapaa. Mikäli folksonomiaa analysoiva sovellus sitä vaatii, kuvausta voidaan täydentää myös tägäyshetkeä kuvaavalla joukolla $D = \{d | d \text{ on ajan hetken kuvaus}\}$ ja/tai tägejä/resursseja yhdistävien aiheiden joukolla $Z = \{z_1, \ldots, z_K\}$.

[16] Tuolloin vielä "Del.icio.us".



Kuva 2.3: Esimerkki yksinkertaisesta hypergraafista

On huomattava, että edellä kuvattu määritelmä ei luo taksonomiaa tai ontologiaa vastaavaa tai ylipäätään minkäänlaista ymmärrystä tuottavaa rakennetta. Tällaisten rakenteiden tuottaminen folksonomiasta vaatii kuvatun rakenteen louhimista ja louhinnan tulosten esittämistä havainnollistavassa muodossa. Tämän louhinnan tekemistä ja tulosten tuottamista käsitellään tarkemmin seuraavissa luvuissa.

2.3 Tutkimuskatsaus ja tägäyspalveluesimerkkejä

Ennen kuin syvennymme tarkemmin tämän tutkielman lähestymistapaan tägien ja folksonomioiden suhteen, on syytä luoda pikainen yleiskatsaus aiheen tutkimuksen lähestymistapoihin yleisesti. Tässä luvussa ryhmitellään tutkielman pohjustamisen ja kirjoittamisen yhteydessä esiin tulleita artikkeleita karkeisiin ryhmiin lähestymistapojensa pohjalta. Kattavampaan katsaukseen tutkimussuunnasta suositellaan tätä varten koottua kirjallisuuskatsausta, esimerkiksi [GLYH10] tai [Tra09].

Tägejä ja folksonomioita käsitellään useissa artikkeleissa käsitteitä sen enempää soveltamatta. Lähteet [Vos07], [Mat04], [Shi05] ja [Gru] esimerkiksi käsittelevät tägäystä ja folksonomioita puhtaasti konseptuaalisesti. Vastapainoisesti taas lähteet [MNBD06], [SD09] ja [KÖ9] tutkivat myös tägäyksen todellisuutta tutkimalla eri palveluiden tägäysdataa.

Tutkiessa eri tapoja muokata ja hyödyntää tägidataa, nousee esille kolme yleistä lähestymistapaa. Ensimmäinen näistä pyrkii muodostamaan tägeista näiden välisiä kytkentöjä kuvaavan graafin. Yleensä tämä graafi on painotettu, kuten lähteissä [SW05], [Mik07], [BKS06] ja [HRS07], mutta myös suunnattu graafi on mahdollinen, kuten lähteessä [HRS06]. Toinen selvä suuntaus on tägien ryhmittäminen abstrak-

tien konseptien alle, jota tutkitaan lähteissä [WZY06], [ZWY06] ja [DLZ+10], kussakin eri tavoin. Kolmantena on tägien ennustus ja/tai suosittelu tietylle resurssille. Tätä suuntaa tutkitaan mm. lähteissä [WSZ09] ja [HRGM08].

Näiden kolmen lähestymistavan lisäksi on tietysti olemassa muitakin tapoja käsitellä tägidataa. Esimerkiksi [CCS08] kuvaa tavan järjestää tägejä hierarkiaan. Szomezor et al. taas yrittivät tutkielmassaan [SCA+07] käyttää tägejä varsin spesifiin käyttötarkoitukseen. Lokakuussa 2006 amerikkalainen elokuvien suoratoistoa tarjoava yritys Netflix julisti kilpailun paremman elokuvansuositusalgoritmin löytämiseksi. Kilpailun ohessa he julkaisivat mittavan toistojoukon algoritmin testaamista varten. [SCA+07, s. 7] Tutkielmassaan Szomszor et al. esittivät kaksi algoritmia jotka yrittivät ennustaa arvosanan, jonka käyttäjä antaa elokuvalle, käyttäen hyväkseen Netfixin antamaa tietojoukkoa (arvosanojen lähde) sekä IMDB:n tietokantaa elokuvista (avainsanojen lähde).

Tutkielmassa esitettiin kaksi avainsanoihin perustuvaa algoritmia ja yksi naiivi, keskiarvoon perustuva algoritmi tehokkuuden vertailemiseksi. Algoritmeista painottamaton antoi enemmän oikeita arvauksia (ero alle 2 prosenttiyksikköä), kun taas painotetun algoritmin tulosten virheiden neliöllinen keskiarvo oli pienempi (ero noin 0,1). Yleisesti ottaen algoritmit ennustivat arvostelmat arviot (1-3) pääsääntöisesti 3:ksi ja korkeat arvostelmat (4-5) 4:ksi. Molemmat algoritmit antoivat harvoin ennustuksien arvoiksi 1, 2 tai 5.

Lisäksi, tägeistä tehdyn tutkimuksen lisäksi on hyvä myös tarkastella olemassa olevien tägäyspaleluiden kirjoa. Taulukko 2.2 antaa joitakin esimerkkejä, kuinka luvussa 2.2 listatut tägäyspalveluiden ominaisuudet on toteutettu joissain tägäyspalveluita tarjoavissa sivustoissa. Listaan on koottu mahdollisimman monipuolinen otos tägäyspalveluista, ottamatta (usean tutkielman pohjalla olevaa Deliciouspalvelua lukuun ottamatta) kantaa palveluiden käyttöön tai käytettävyyteen tägien ja folksonomian tutkimuksessa.

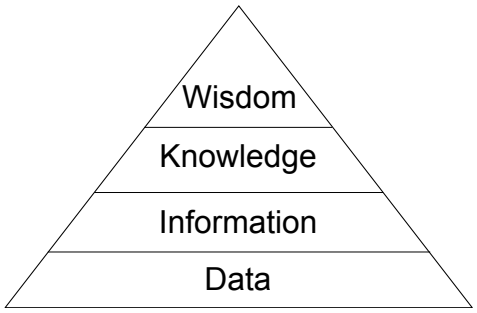
	Delicious	**YouTube**	**StackOverflow**	**Flickr**
Tägäys-oikeudet	yleinen	resurssin haltija	yleinen, mutta rajoituksilla	resurssin haltija
Tägien ryhmittäminen	monijoukko	joukko	joukko	joukko
Tägyksen tuki	ehdotus	ehdotus	ehdotus	ei tukea
Resurssin tyyppi	WWW-sivu	video	tekstimuotoinen kysymys	kuva
Resurssin lähde	yleinen	käyttäjät	käyttäjät	käyttäjät
Resurssien liittyvyys	mahd. linkitys	video-response	ei	ei
Käyttäjien liittyvyys	ystävälistat	seuraajat ja ystävälistat	ei	ystävälistat

Taulukko 2.2: Esimerkkejä eri tägäyspalveluista

3 Tietämyksen johtaminen tägeistä

Luvussa 2 käsiteltiin lähinnä datan ja metadatan erottelua. Mutta pelkkä data ei, jopa määritelmänsä mukaan [Row07, luku 5.2], ole hyödyllistä. Jotta tägidataa pystyttäisiin hyödyntämään, pitää siitä pystyä johtamaan arvokkaampia ja merkityksellisempiä muotoja.

Lähde [Row07] kuvaa tietämyksen hierarkian useita vuosikymmeniä vanhaksi käsitteeksi, mainiten hierarkian ensimmäiseksi ilmentymäksi T.S. Eliotin vuonna 1934 kirjoittaman runon *The Rock*. Kuuluisampi, modernimpi, tarkempi ja useammin käytetty määritelmä tästä hierarkiasta on kuitenkin R.L. Ackoffin artikkelista *From data to wisdom*. Tässä hierarkiassa ylimpänä on viisaus (engl. *wisdom*) ja tämän alla, järjestyksessä, ymmärrys (engl. *understanding*), tietämys (engl. *knowledge*), informaatio (engl. *information*) ja viimeisenä data.



Kuva 3.1: DIKW-hierarkia pyramidina

Määritellään edellä nimetyt termit tässä tiivistetysti lähteen [Row07, luku 5] kokoamia määritelmiä ja lähdettä [Ack89] käyttäen:

- Data:
  Havaintojen perusyksikköjä. Organisoimattomia ja prosessoimattomia. Usein arvotonta, koska ei sisällä kontekstia tai tulkintaa.

- Informaatio:
  Todellisuutta kuvaavaa dataa, joka on ihmiselle merkityksellisessä muodossa ja/tai yhteydessä. Prosessoitua dataa, jolle on tarkoituksenmukaisen prosessoinnin kautta annettu merkitys.

- Tietämys:
  Tietämys on taitotietoa tai asiantuntemusta siitä kuinka esim. jokin järjestelmä toimii. Tietämyksen tarkka määritelmä on vaikea tehtävä, mutta yksi merkittävä yksityiskohta on, että tietämys on aina opittua, joko toiselta sen omaavalta entiteetiltä tai kokemuksen tuoman informaation kautta.

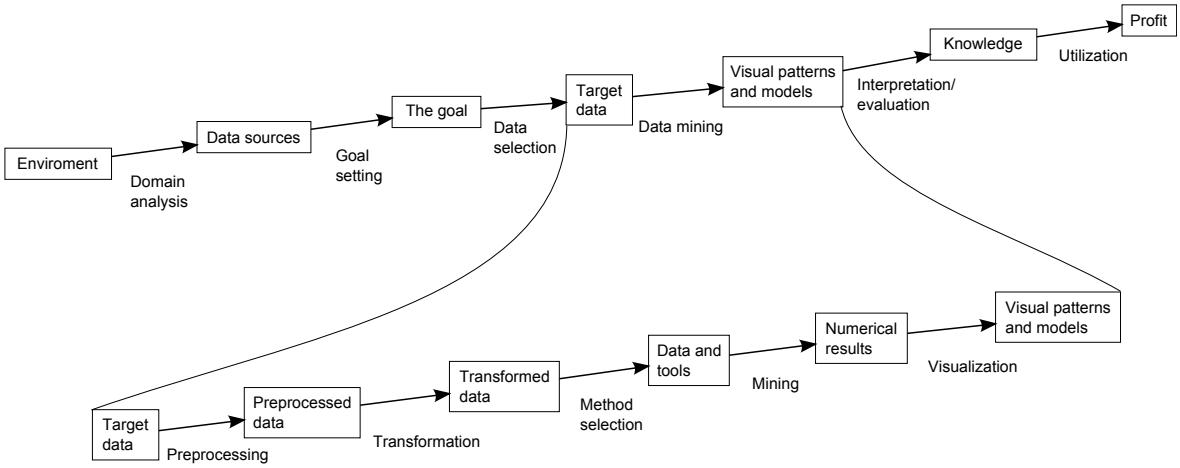
- Viisaus:
  Viisaus on helpointa kuvata vertaamalla sitä tietämykseen ja samalla kaikkiin aiempiin hierarkian tyyppeihin. Kaikki edelliset lisäävät toimimisen tehokkuutta (engl. *efficiency*), kun taas viisaus lisää toimimisen vaikuttavuutta (engl. *effectiveness*)

Hierarkia kuvataan kuvan 3.1 kaltaisena pyramidina, koska korkeampaa muotoa varten tarvitaan alempaa muotoa. Informaatio johdetaan datasta, tieto informaatiosta ja tietämys tiedosta. Tämä johtamisprosessi on ennemminkin karsiva kuin transformoiva (tosin, kuten Weinberger [Wei] argumentoi, ei pelkkää suodatusta). Tästä johtuen informaatio voidaan ajatella datan alalajina, tietämys informaation alalajina ja viisaus tietämyksen alalajina.

3.1 Tietämyksenlouhinta

Kuten johdannossa mainittiin, yhteisöpalveluiden, ja tätä myötä tägäyspalveluiden suosio ollut viime vuosina nousussa. Tästä johtuen saatavilla olevan folksonomiadatan määrä on valtava ja jatkuvasti kasvussa. Hand et al. [HMS01] määrittelevät tiedonlouhinnan (engl. *data mining*) datan (usein suuren määrän) analysoinniksi yllättävien suhteiden löytämiseksi ja/tai ymmärrettävien ja käyttökelpoisten visuaalisaatioiden luomiseksi. Näin ollen tiedonlouhinta kuulostaa luonnolliselta tavalta prosessoida folksonomiadataa. Tässä tutkielmassa tiedonlouhinta suoritetaan Sami Äyrämön määrittelemän tietämyksenlouhintaprosessin (engl. *Knowledge Mining*) [Äyr06, luku 2.3.2] avulla. Tämä prosessi muodostuu kahdesta osaprosessista, tietämyksen muodostuksesta (engl. *Knowledge Discovery*) ja tiedonlouhinnasta, joista jälkimmäinen on ensimmäisen osaprosessi kuvan 3.2 kuvaamalla tavalla.

Tietämyksen muodostus koostaa tietämyksenlouhintaprosessin yleiskuvan, alkaen ympäristön analyysista ja päätyen tulosten hyödyntämiseen. Kaikki prosessin askeleet kuvataan seuraavassa listassa:



Kuva 3.2: Äyrämön tietämyksenlouhintaprosessi

- Kohdealueanalyysi (engl. *Domain Analysis*):
  Tässä askeleessa analysoidaan ympäristöä (engl. *Environment*), jossa tiedonlouhinta tullaan suorittamaan. Tätä tutkielmaa varten ei erikseen tehty kohdealueanalyysia vaan analyysinä käytettiin Nataliia Samoilenkon raporttia [Sam10], joka kuvaa ICT-alan koulutusympäristöä tämänkin tutkielman näkökulmasta. Kohdealueanalyysin tuloksia ovat ympäristön parempi ymmärrys ja datan mahdolliset lähteet sekä näiden lähteiden ominaisuudet (esim. datan muoto ja saatavuus) seuraavaa askelta varten.

- Tavoitteen asetus (engl. *Goal Setting*):
  Kun kohdealueanalyysin tuloksena on saatu tieto mahdollisista datan lähteistä, voidaan louhintaprosessin tavoite valita realistisesti. Mitä luultavimmin tietämyksenlouhintaprosessin aluksi on jo asetettu jokin, mahdollisesti epämääräinen tavoite. Tässä askeleessa tavoitteen realistisuus voidaan varmentaa ja tavoite tarkentaa. Tässä tutkielmassa tavoite on kuvattu johdannossa.

- Datan valinta (engl. *Data Selection*):
  Kohdealueanalyysin ja asetetun tavoitteen pohjalta voidaan jo suorittaa askel, jossa kaikista ympäristöstä havaituista datan lähteistä valitaan ne lähteet, joita loppujen lopuksi tullaan käyttämään tiedonlouhintaprosessissa. Lisäksi kustakin lähteestä valitaan tarvittava tietojoukko käytettäväksi. Tämä voi olla joko kaikki lähteen sisältämä data tai valikoitu osa. Delicious valittiin tässä toteutuksessa pääasialliseksi datan lähteeksi seuraaviin syihin vedoten. Ensinnäkin monet aiemmat julkaisut (Kaikki luvussa 2.3 mainitut jotain datalähdettä

käyttävät julkaisut, lukuun ottamatta julkaisuja [SD09], [DLZ+10], [SCA+07], [WSZ09] ja [CCS08]) käyttävät kyseistä sivustoa datan lähteenä, joten sivuston antaman datan louhintaan on tunnetusti jo useita menetelmiä. Lisäksi sivuston käyttäjäkunta on huomattavan suuri, antaen näin tarvittavan suuren datan määrän, ja hyvin ICT-tietoinen (tämä todetaan mm. tutkielmassa [ZWY06, s. 179], jossa analyysin tuloksena huomataan, että tägiä `owl` käytetään enimmäkseen teknologisessa merkityksessä viitaten Web Ontology Laguage[1] (josta käytetään lyhennettä OWL) kuvailukielien perheeseen ja vähemmän viitatessa pöllöihin), joten ICT-alan kiinnostavat ilmiöt näkyvät datassa keskimääräistä suuremmalla todennäköisyydellä.

- Tiedonlouhinta:
  Varsinainen tekninen tiedonlouhintaprosessi, jossa valikoitu data muokataan visuaaliseksi esitykseksi tulkintaa varten. Tämän askeleen yksityiskohtiin palataan myöhemmin tässä luvussa.

- Tulkinta/arviointi (engl. *Interpetation/evaluation*):
  Tässä askeleessa tiedonlouhinnan tuloksia pyritään tulkitsemaan asetettujen tavoitteiden mukaan. Askeleen onnistumiskriteeri riippuu paljon asetetusta tavoitteesta, mutta yleensä tavoitteena on jonkinlainen tiedon lisääminen tutkitusta kohteesta. Tähän palataan myöhemmässä luvussa.

- Hyödyntäminen (engl. *Utilization*):
  Saavutettu tietämyksen lisäys on hyödytön, jos sitä ei hyödynnetä käytännössä. Tämä tutkielma ei käsittele tarkemmin, kuinka rakennetun menetelmän avulla saavutettua tietämystä tulisi hyödyntää. Toiveena kuitenkin on, että tietämys auttaisi PROFIT-projektia tarjoamaan yrityksille koulutusaiheita tai -aiheen osa-alueita, jotka ovat yrityksille entuudestaan tuntemattomia mutta silti hyödyllisiä.

Tiedonlouhinta on prosessin tekninen osa, jonka suoritukseen tässä tutkielmassa keskitytään. Johtuen prosessin teknillisestä luonteesta ja tietämyksen muodostusprosessiin verrattuna poikkeavasta asiantuntemuksen vaatimuksesta, Äyrämö suo-

[1] `http://www.w3.org/TR/owl2-overview/`

sittelee että prosessi on mahdollisimman automatisoitu. Tässä yhteydessä[2] tiedonlouhinta koostuu askelista:

- Esikäsittely (engl. *Preprocessing*):
  Datalähteistä valittu data ei välttämättä ole sellaisenaan tiedonlouhinta-algoritmien käytettävissä vaan vaatii datan esikäsittelyn algoritmin hyväksymään muotoon. Vaadittavat esikäsittelymenetelmät ovat pitkälle riippuvia käytettävistä algoritmeista. Ne sisältävät esimerkiksi puuttuvan datan imputoinnin sekä virheiden ja häiriödatan korjaukset tai poistot. Tägidatan esikäsittelyä sivutaan luvussa 3.2 ja tämän tutkielman esikäsittelyprosessia käsitellään luvussa 6.1.

- Transformointi (engl. *Data transformation*):
  Läheisesti esikäsittelyyn liittyvä askel on datan transformointi. Se eroaa esikäsittelystä siinä, että esikäsittelyn korjatessa yksittäisiä ongelmakohtia datassa, tranformointi muokkaa koko datajoukkoa esimerkiksi skaalaamalla, piirteitä erottelemalla tai datan ulottuvuuksia vähentämällä. Transformaation pääasialliset tavoitteet ovat datan yhtenäistäminen ja yksinkertaistaminen.

- Menetelmän valinta (engl. *Method selection*):
  Tässä askeleessa valitaan datalle ja tavoitteelle sopivat louhintamenetelmät.

- Louhinta (engl. *Mining*):
  Vaikka louhintavaihe onkin tiedonlouhintaprosessin ydin, on se pitkälle automaattinen ja sisältää lähinnä edellisessä askeleessa valittujen menetelmien soveltamisen esikäsiteltyyn ja transformoituun dataan.

- Visualisointi (engl. *Visualization*):
  Louhintaprosessin tuottama data on harvoin helposti ymmärrettävässä muodossa. Jotta tiedonlouhintaprosessin loppukäyttäjä pystyisi hyödyntämään louhinnan tuloksia, on tulokset usein syytä esittää jonkinlaisessa tiivistetyssä, visuaalisessa muodossa, josta louhinnan tulos on nähtävissä mahdollisimman intuitiivisesti.

[2]Määritelmissä, jotka eivät jaa tietämyksen saavuttamiseen tähtäävää prosessia tietämyksenmuodostukseen ja tiedonlouhintaa sisällyttävät usein tietämyksenmuodostuksen keskeisimmät askeleet tiedonlouhintaprosessin määritelmään.

	Rakenteen löytäminen	**Tiedonjyvien löytäminen**	
		tuntemattomia	*tunnettuja*
Ei-tekstuaalinen data			tietokantahaut
Tekstuaalinen data	laskennallinen lingvistiikka	tekstitiedon louhinta	tiedonhakua

Taulukko 3.1: Tiedonlouhinnan alalajeja jaoteltuna tavoitteen ja datan mukaan

3.2 Tiedonlouhinnan alalajit

Edellä kuvattu prosessi kuvaa tiedonlouhintaa yleisellä tasolla, joka ei ota kantaa käytetyn datan muotoon eikä haluttuihin tuloksiin. On kuitenkin olemassa useita tiedonlouhinnan alalajeja, jotka keskittyvät tiettyyn lähdedatan muotoon ja haettuun tulokseen. Eriytymällä omaksi tutkimusalueekseen nämä alalajit pystyvät paremmin ottamaan huomioon oman erikoistumisalueensa erikoisvaatimukset.

Marti Hearst esittää lähteessä [Hea99] yhden tiedonlouhinnan alalajien jaottelun. Datan osalta Hearst keskittyy tekstuaaliseen ja ei-tekstuaaliseen dataan, kun taas tavoitteissa jaottelu tehdään rakenteen (engl. *pattern*) löytämisen sekä tiedonjyvien, joko ennalta tunnettujen tai tuntemattomien (engl. *novel nuggets* ja *non-novel nuggets*), löytämisen välillä.[3] Hearstin mukaan esimerkiksi tunnettujen tiedonjyvien löytäminen tekstidatasta on tiedonhakua (engl. *information retrieval*) kun taas rakenteiden löytäminen tekstidatasta kuuluu laskennallisen lingvistiikan piiriin (engl. *computational linguistics*). Taulukossa 3.1 kuvataan eri tiedonlouhinnan alalajien sijoittuminen edellä kuvattuun jaotteluun Hearstia mukaillen.

Tarkastellaan folksonomiadatan louhintaa edellä kuvattujen tavoitteiden mukaan. Tunnettujen tiedonjyvien löytäminen tägejä hyväksikäyttäen on oikeastaan vain tiedonhakua ilman tarvetta esikäsitellä haun kohteena olevia dokumentteja/resursseja avainsanojen toivossa. Tiedonhaun yleisesti tunnetuista malleista vektoriavaruuteen (engl. *vector space*) pohjautuva malli on helposti sovellettavissa tägipohjaiseen dataan. Rakenteiden louhintaa folksonomia- ja tägidatasta on tutkittu yleisellä tasolla useasti ja laajalti. Tämä todettiin jo luvussa 2.3, jossa listataan useita lähteitä, jotka tutkivat tägijakaumia eri tägäyspalveluissa. Tiedonlouhinnan tavoitteiden lisäksi folksonomiadatan louhintaa voidaan verrata myös muihin louhittaviin

[3]Hearstin tiedonlouhinnan alalajien asetusta tähän jaotteluun on kritisoitu esimerkiksi Kroeze et al. kirjoittamassa julkaisussa [KMB03]. Jaotteluperusteet (datan muoto ja tavoitteet) ovat tässäkin julkaisussa tosin samat.

datan muotoihin, etenkin näihin sovellettavien, datatyypille ominaisten prosessien osalta. Tarkastellaan tässä esimerkiksi tekstitiedonlouhintaa ja etenkin tekstidatan esikäsittelyä. Kirja [BYRN99, luku 7.2] (esimerkiksi) listaa tekstitiedon louhinnassa käytettävän dokumentin esikäsittelyprosessin seuraavasti: 1) leksikaalinen analyysi (engl. *lexical analysis*), 2) sulkusanojen (engl. *stopwords*) poistaminen, 3) stemmaus (engl. *stemming*), 4) indeksitermien valinta, ja 5) sanakirjan (engl. *thesaurus*) kokoaminen. Useat askeleista eivät ole tärkeitä tägipohjaiselle datalle, koska tägit ovat valmiiksi jaoteltuna sanoittain, kun taas tekstidatasta sanat ovat tallennettuna merkkijonoihin, joista sanat pitää erikseen erotella. Lisäksi tägidatan sanasto on huomattavasti luonnollista tekstiä rajoittuneempi.

Tämä ei kuitenkaan tarkoita, etteikö tägipohjaisen datan louhintaprosessi voisi käyttää joitakin tekstidatan louhinnan tekniikoita. Otetaan esimerkiksi edellisen listan kohta 1, leksikaalinen analyysi. Tekstin käsittelyssä tämän askeleen pääasiallinen tehtävä on jakaa merkkijono sanoiksi. Kuten edellä mainittiin, tälle tehtävälle ei tägipohjaisessa datassa ole tarvetta. Mutta tämän lisäksi askeleessa käsitellään myös sanoihin liittyviä poikkeamia, kuten isoja kirjaimia ja välimerkkejä. Näitä esiintyy usein tägidatassa ja niiden oikeaoppinen käsitteleminen auttaa varmistamaan, että näitä poikkeamia eliminoitaessa tägin merkitys ei muutu. Myös kohdan 3 stemmausta voidaan soveltaa tägeihin, joskaan taivutetut muodot eivät ole tägidatassa niin yleisiä kuin luonnollisessa tekstissä.

Tekstidokumenttien esikäsittelyprosessista kohta 2, sulkusanat, on luultavasti se, josta tägipohjaisen datan käsittely hyötyisi eniten. Esimerkiksi luvussa 2.2 listattiin eri tägityyppejä tägäyksen motiivien mukaan. Mitä todennäköisimmin tägidatan tietämyksenlouhintaprosessissa voidaan jo hyvin aikaisessa vaiheessa karsia pois tägityyppejä, joiden sisältö ei missään nimessä voi olla kiinnostavaa tietämyksenlouhinnan tavoitteen kannalta. Hyvänä esimerkkinä tästä ovat henkilökohtaiseen organisointiin liittyvät tägit, kuten `toread`, jotka ovat relevantteja vain tägäyksen tehneelle henkilölle.[4] Kuten luvussa 6.1 todetaan, yksinkertainen sulkusanalista karsi tässä tutkielmassa käytetystä datajoukosta viidesosan tägidatasta.

[4]Mainittakoon kuitenkin että, kuten jo luvussa 2.2 mainittiin, listauksessa määritellyt tägiryhmät voivat limittyä sisältämiensä tägien osalta. Näin ryhmäkohtaisessa tägien karsimisessa tulisi olla tarkkana, että samalla datasta ei karsita myös mielenkiintoiseen ryhmään kuuluvia tägejä.

4 Menetelmiä

4.1 Lyhyesti ryvästyksestä

Koska myöhemmissä luvuissa tullaan puhumaan ryvästävistä (kts. luku 4.2.2) tai ryvästystä muistuttavista (kts. luku 4.3) algoritmeista, on aiheellista kuvata ryvästämisen käsite lyhysti ennen näihin määritelmiin paneutumista.

Tirozzi et al. [TBF07, luku 6.1] kuvaavat ryvästyksen prosessiksi, jossa samankaltaiset entiteetit ryhmitellään yhteen. Ryhmiä, joihin entiteettejä ryhmitetään kutsutaan ryppäiksi tai klustereiksi (engl. *cluster*). Ryhmittely suoritetaan jonkinlaisen samankaltaisuusmetriikan perusteella, pohjautuu usein datapisteestä johdettuun n pituiseen vektoriin. Huomattavana poikkeuksena tähän mainittakoon graafien ryvästys (jota käsitellään tarkemmin luvussa 4.2.2 ja lähteessä [Sch07]), jossa käsitellään useimmiten samankaltaisuuslukujen matriisia. Tirozzi et al. huomauttavat myös, että itse ryvästettävät objektit eivät ole merkityksellisiä ryvästysalgoritmille, kunhan näistä vain saadaan samankaltaisuusmetriikat.

Kuten lähde [Sch07, luku 1] mainitsee, ryvästyksen on hyvin lähellä hahmontunnistuksen ohjaamattoman oppimisen (engl. *unsupervised learning*) tekniikkaa, jonka tarkoituksena on luokitella dataa ilman *a prioria* informaatiota. Myös ryvästyksessä tarvittavan a priorin informaation määrää pyritään minimoimaan. Esimerkiksi suosittu K-means -menetelmä [TBF07, luku 6.5] vaatii informaatiota haluttujen ryppäiden keskipisteistä, joko prototyyppien muodossa tai, mikäli ryppäät halutaan tehdä pelkällä alkuarvauksella, ryppäiden määränä.

4.2 Graafeihin perustuvat menetelmät

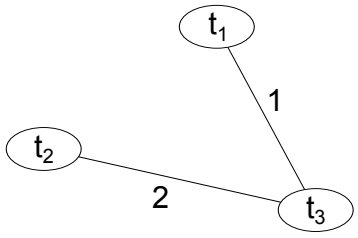
Folksonomiadatan hypergraafi $F \subseteq A \times T \times I$ voidaan esittää painotettuna yksinkertaisena graafina ottamalla yksi joukoista tutkittaviksi solmuiksi (usein joukko T eli tägit) laskemalla kahdesta jäljelle jääneistä joukoista yhdessäesiintymiset solmujen välisten kaarien painoiksi.

Matemaattisesti tägejä kuvaavasta graafista muodostuu joukko $G = (T, E), E = (t_1, t_2, w) : t_1, t_2 \in T, t_1 \neq t_2, w = n_1 \| n_2$, missä käytetään merkintää n_x kuvaamaan

tägin t_x esiintymislukumäärää ja merkintää $n_x \| n_y$ kuvaamaan tägien t_x ja t_y yhdessäesiintymisien lukumäärää. Algoritmisesti graafi on helpointa muodostaa käymällä läpi kukin tägäystapahtuma (käyttäjä $\hat{a}$ tägää resurssin $\hat{i}$ tägijoukolla $\hat{T}$) ja käymällä läpi kunkin tägäystapahtumaan kuuluvan tägijoukuon $\hat{T}$ 2 jäseniset alijoukot $t_x, t_y \subseteq \hat{T}$. Kukin alijoukko käsitellään seuraavasti. Jos suuntaamaton painotettu graafi G sisältää kaaren tägien t_x ja t_y välillä, lisätään kaaren painoa yhdellä. Jos graafi ei sisällä tätä kaarta, luodaan kaari painoarvolla 1 ja mahdollisesti tarvittavat solmut.

Algoritmi 1 Tägien yhdessäesiintymisgraafiin rakennus

for all $taggingEvent \in taggingEvents$ **do**
  $tags \leftarrow taggingEvent_{tags}$
  for $i = 1 \rightarrow tags_{lenght} - 1$ **do**
    for $j = i + 1 \rightarrow tags_{lenght}$ **do**
      if $(tags_i, tags_j) \notin graph_{edges}$ **then**
        $graph_{(tags_i, tags_j)} \leftarrow 1$ {Luo solmut $tags_i$ ja $tags_j$, jos tarpeen ja luo 1:n arvoinen kaari näiden välille}
      else
        $graph_{(tags_i, tags_j)} \leftarrow graph_{(tags_i, tags_j)} + 1$
      end if
    end for
  end for
end for



Kuva 4.1: Kuvan 2.3 tägäystapahtumista johdettava tägien kytkeytymisgraafi

Kuvassa 4.1 kuvataan, miltä edellä kuvatun algoritmin tulos olisi käyttäen kuvassa 2.3 kuvattuja tägäystapahtumia[1]. Lisäksi, algoritmien toteutuksen helpottami-

[1]Olettaen, että kunkin käyttäjän yhteen resurssiin liittämät tägit muodostavat yhden tägäystapahtuma, ovat tägäsytapahtumat resurssille r_1 {[t_1],[t_1]}, resurssille r_2 {[t_1, t_3],[t_2]} ja resurssille r_3 {[t_2, t_3],[t_2, t_3]}

seksi on yleensä hyvä idea tallentaa samalla tägien esiintymisten lukumäärä graafin solmujen painoarvoksi, mikäli tämä on käytetyssä graafikirjastossa mahdollista.

4.2.1 Taksonomian johtaminen graafista

Lähteessä [HRS06, luku 6] kuvataan lyhyesti ehdotus menetelmään, jonka avulla folksonomiadatasta voidaan johtaa hierarkkisia taksonomioita. Menetelmä käyttää tulosten esityksessä hyväkseen W3C:n RDF (Resource Description Framework)-mallissa esiteltyä predikaattia *rdfs:subClassOf*, johonka sopivaksi se kokoaa tägejä ja tägeistä muodostettuja konsepteja toistensa ylä- ja alaluokiksi. Määritellään funktiot $C(y,x) = \frac{n_x \| n_y}{n_y}$ ja $I(x)$ tai $I(x,y)$, jotka kuvaavat tägin t_x tai tägiparin t_x ja t_y informaatioarvoa [HRS06, luku 3] laskemalla, montako sivua tägin tai tägiparin avulla löytyy. Funktion yläraja $I(t_x) = 1$ kuvaa tilannetta, jossa tägillä t_x löytyy tarkin (eli pienin) mahdollinen määrä sivuja, tässä yksinkertaisuuden vuoksi 1. Funktion alaraja $I(t_x) = 0$ kuvaa tilannetta, jossa tägillä t_x haettaessa löydetään jokainen mahdollinen sivu, tai vaihtoehtoisesti ei ainuttakaan sivua. Yleisesti ottaen pienempi informaatioarvo kuvaa suurempaa määrää löydettyjä sivuja.

Menetelmässä käydään läpi kaikki graafin G kaaret ja luo mahdollisesti kunkin kaaren yhdistämien tägien t_x ja t_y välille ylä- ja alaluokkayhteyden seuraavien sääntöjen ja raja-arvon ϵ perusteella:

- Jos $I(y) < I(x)$ ja jos $1 > C(y,x) > \epsilon$ niin t_y *rdfs:subClassOf* t_x.
- Jos $I(y,x) \leq I(y) < I(x)$ ja jos $C(y,x) = 1$ niin "t_y t_x" *rdfs:subClassOf* t_x.

4.2.2 Graafin ryvästys

Begelman et al. [BKS06] kuvaavat ehdotuksen graafin ryvästystä varten. Ehdotettu menetelmä käyttää edellä kuvattua graafikuvausta pohjanaan. Koska kappaleen alussa kuvatun tägigraafin perusmuodossa kaarien painoarvot ovat tägien yhteisesiintymiä, suosittuihin tägeihin liittyvillä kaarilla on tässä graafissa yleensä huomattavan suuri painoarvo. Jotta suositut tägit eivät tästä syystä hallitsisi ryvästystä, vaan ryvästys toimisi tägien samankaltaisuuden perusteella, artikkeli ehdottaa kaarien painojen normalisointia jonkin esiintymistodennäköisyyksiin pohjautuvan samankaltaisuusmitan avulla. Esimerkkeinä näistä mitoista annetaan mm. Dicen samankaltaisuus $\frac{2|t_x \cap t_y|}{|t_x|+|t_y|}$ ja Jaccardin samankaltaisuus $\frac{|t_x \cap t_y|}{|t_x \cup t_y|}$ joissa $|t_x|$-merkinnällä kuvataan tägin esiintymistodennäköisyyttä. Mikäli esimerkiksi kuvan 4.1 kaarien

painot tasattaisiin Dicen samankaltaisuusmitan avulla, saadaan kaaren $\{t_1, t_3\}$ painoksi $\frac{1}{3}$ ja kaaren $\{t_2, t_3\}$ painoksi $\frac{4}{5}$. Näin tägin t_2 muita pienempi esiintymismäärä skaalaa tägiin liittyvän kaaren painoarvon painottamattomasta kaksinkertaisesta 2,4-kertaiseksi.

Varsinaista ryvästysmenetelmää artikkelissa ei kuvata, mutta menetelmän valinnalle annetaan joitakin ohjeita, suosituksia ja rajoituksia. Eritoten mainitaan tägigraafin ryvästyksen keskeinen ongelma, eli ryppäiden määrä. Huomattavana ongelmana tägien ryvästyksessä on, että ryppäiden määrästä ei ole aprioria tietoa. Siksi artikellissa suositellaan välttämään algoritmeja, jotka vaativat ryppäiden määrää alkuarvaukseksi. Ja koska monet ryvästysalgoritmit perustuvat ennalta määrättyyn ryppäiden määrän arvaukseen, rajoittaa tämä merkittävästi algoritmin valintaa. Begelman et al. itse suosittelevat spektraalista ryvästysalgoritmia (engl. *spectral clustering*). Toinen lähestymistapa ryvästykseen on hierarkkinen ryvästys. Hierarkkisuuden avulla pyritään välttämään apriori ryppäiden määrän asetus jakamalla graafi sisäkkäisiin ryppäisiin lukumäärään sitomattomien sääntöjen perusteella. Tutkielman yhteydessä ei kuitenkaan löytynyt esimerkkiä graafin hierarkkisesta ryvästyksestä tägidatan yhteydessä.

4.3 SMM

Lei Zhangin, Xian Wun ja Yong Yun tutkielma [ZWY06] emergenteistä semantiikoista käyttää hyväkseen seuraavaa todennäköisyyspohjaista mallia URL:ien ja tägien ryvästämiseksi abstraktien konseptien[2] alle:

1. Käyttäjä kohtaa satunnaisesti URL:n i_i todennäköisyydellä p_i.
2. URL i_i ja tämän sisältö saavat käyttäjän ajattelemaan konseptia z_α todennäköisyydellä $p_{\alpha|i}$.
3. Konsepti z_α saa käyttäjän ajattelemaan tägiä c_j todennäköisyydellä $q_{j|\alpha}$.

Tämä malli vastaa Hofmannin ja Puzichan [HP98] määrittelemää asymmetristä SMM-mallia (Separable Mixture Model) yhdessäesiintymisdatalle. Tämä malli

[2]Konseptit ovat tässä käytännössä folksonomian tägien sumea jaottelu K:hon ryhmään ja samalla folksonomian resurssien jaottelu samoihin K:hon ryhmään. Käyttäjä on tässä mallissa vain jaottelun mahdollistava tekijä, johon malli ei tarkemmin ota kantaa. Mallista on olemassa muunnelma, joka ottaa myös käyttäjät huomioon. Tämä malli esitellään julkaisussa [WZY06]

on laskennallisesti ja täten myös laskennan tulosten kannalta yhtenevä symmetrisen SMM-mallin kanssa. Mikäli edellinen prosessi toimisi symmetrisesti, näyttäisi se tältä:

1. Käyttäjä ajattelee satunnaisesti konseptia z_α todennäköisyydellä π_α.
2. Käyttäjä valitsee URL:n i_i konseptista z_α todennäköisyydellä $p_{i|\alpha}$.
3. Käyttäjä valitsee tägin c_j konseptista z_α todennäköisyydellä $q_{j|\alpha}$.

Symmetrinen malli voidaan sovittaa tunnettujen yhteisesiintymien joukkoon maksimoimalla seuraava logaritminen uskottavuusfunktio [HP98, luku 2.2]:

$$\mathcal{L} = \sum_{r=1}^{L} \sum_{\alpha=1}^{K} R_{r\alpha} (\log \pi_\alpha + \log p_{i(r)|\alpha} + \log q_{j(r)|\alpha})$$

Tässä $R_{r\alpha}$ kuvastaa todennäköisyyttä, että URL:n i_i ja tägin c_j yhdessäesiintymä (i_i, c_j, r) (joks kuuluu joukkoon $S = (i_{i(r)}, c_{j(r)}, r) : 1 \leq r \leq L$) on seurausta konseptista z_α, L on näiden yhdessäesiintymien kokonaismäärä ja R on annettu konseptien määrä (yleensä pelkkä arvio). Uskottavuusfunktiolle voidaan löytää lokaali maksimi jollakin alkuarvauksella käyttäen hyväksi EM-algoritmia (Expectation Maximation).[3] SMM-mallille sovitettuna, EM-algoritmi saa seuraavanlaisen muodon:

E-askel

$$\langle R_{r\alpha} \rangle^{(t+1)} = \frac{\hat{\pi}_\alpha^{(t)} \hat{p}_{i(r)|\alpha}^{(t)} \hat{q}_{j(r)|\alpha}^{(t)}}{\sum_{v=1}^{K} \hat{\pi}_v^{(t)} \hat{p}_{i(r)|v}^{(t)} \hat{q}_{j(r)|v}^{(t)}}$$

M-askel

$$\hat{\pi}_\alpha^{(t)} = \frac{1}{L} \sum_{r=1}^{L} \langle R_{r\alpha} \rangle^{(t)}$$

$$\hat{p}_{i|\alpha}^{(t)} = \frac{1}{L\hat{\pi}_\alpha^{(t)}} \sum_{r:i(r)=i} \langle R_{r\alpha} \rangle^{(t)}$$

$$\hat{q}_{j|\alpha}^{(t)} = \frac{1}{L\hat{\pi}_\alpha^{(t)}} \sum_{r:j(r)=j} \langle R_{r\alpha} \rangle^{(t)}$$

Jos $\langle R_{r\alpha} \rangle$:aa ei haluta laskea ja tallentaa erikseen, voidaan nämä matemaattiset kaavat tiivistää listauksessa 5.5.2 kuvatuksi algoritmiksi.

[3] Alkuarvausta $R_{r\alpha} = C, C$ *on vakio kaikilla* $1 \leq r \leq L$ *ja* $1 \leq \alpha \leq K$ ei suositella, koska on todistettavissa ettei EM-algoritmi konvergoi, mikäli edellä kuvattu tilanne toteutuu. Todistus löytyy liitteestä C

Tulee myös huomata, että $q_{j|\alpha}$ kuvaa todennäköisyyttä että konseptista z_α aiheutuu tägin c_j valinta. Mikäli tägejä on huomattava määrä, on tämä luku keskimääräisesti huomattavan pieni. Jotta saataisiin helpommin tulkittava luku $q_{\alpha|j}$, joka kuvaa todennäköisyyttä että tägi c_j assosioituu konseptiin z_α, joudutaan $q_{\alpha|j}$ laskemaan Bayesin teoreemaa käyttäen kaavalla

$$q_{\alpha|j} = \frac{q_{j|\alpha}\pi_\alpha}{p(c_j)} = \frac{q_{j|\alpha}\pi_\alpha}{\sum_{v=1}^{K} q_{j|v}\pi_v}$$

ja vastaavasti luvulle $p_{\alpha|i}$. Näitä lukuja hyväksikäyttäen voidaan laskea kunkin URL:n ja tägin semantiikkaa SMM-mallin avulla luoduissa konsepteissa kuvaavat vektorit $\overrightarrow{semantics(i_i)} = \langle p_{\alpha|i} | \alpha = 1, 2, \ldots, K \rangle$ ja $\overrightarrow{semantics(c_j)} = \langle q_{\alpha|j} | \alpha = 1, 2, \ldots, K \rangle$, joita tarkastelemalla voidaan arvioida tägien ja URL:ien semantiikkaa. Lisäksi luvuista voidaan laskea tägin tai URL:n epämääräisyyttä (mallin mukaisella ryhmityksellä) kuvaava luku seuraavilla kaavoilla:

$$ambiguity(i_i) = -\sum_{\alpha=1}^{K} p_{\alpha|i} \log p_{\alpha|i}$$

$$ambiguity(c_j) = -\sum_{\alpha=1}^{K} q_{\alpha|j} \log q_{\alpha|j}$$

Nämä arvot, antaen yksittäisen numeroarvon kunkin tägin ja URL:n levinneisyydelle eri konseptien kesken, kuvaavat yksinkertaisella tavalla kuinka selvästi tägi tai URL liittyy johonkin konseptiin tai konsepteihin (luku ei kerro, mikä tai mitkä konseptit nämä ovat). Näin ollen luvun avulla voidaan arvioida, kuinka selvästi tägi tai URL liittyy rajoitettuun määrään annettuja konsepteja. Arvo on pienimmillään 0, jolloin tägi tai URL kuuluu yhteen konseptiin todennäköisyydellä 1. Arvon maksimi (tapaus, jossa tägi tai URL kuuluu kaikkiin konsepteihin yhtä todennäköisesti) riippuu konseptien määrästä K, mutta se ei voi olla missään tapauksessa yli $\frac{K}{e}$[4].

4.4 Tägipilven muutosten seuranta

Tämän menetelmän idea pohjautuu Terrell Russellin toteuttamaan Cloudalicious-palveluun [Rus06], joka analysoi tiettyyn URL:ään Delicious-palvelussa liitettyjen

[4]Funktion $-x \log x$ maksimi välillä $[0, 1]$ on $\frac{1}{e}$, joten summan $-\sum_{i=1}^{K} x_i \log x_i$ maksimi on $\frac{K}{e}$, kun $0 \geq x_i \geq 1 | i \in 1, \ldots, K$.

tägien määrää ja tämän määrän kehitystä ajan myötä. Koska palvelu ei kirjoitushetkellä ollut enää toimintakunnossa eikä toiminnasta ollut saatavilla tarkkaa kuvausta, pohjautuu tässä annettu menetelmänkuvaus Pietro Speroni di Fenizion blogikirjoitukseen [SdFc] ja videoituun luentoon [SdFa], jotka käsittelevät Cloudaliciouspalvelun toimintaa sekä palvelun tuottamien visalisaatioiden tulkintaa.

Menetelmän perusideana on seurata muutoksia tägipilvessä, jonka Speroni di Fenizion määrittelee [SdFa, 32:47] joukoksi tägejä jossa joka tägillä on tietty monilukuisuus. Kustakin tägistä seurattava arvo on tägin paino, joka lasketaan jakamalla tägiä käyttävien käyttäjien lukumäärä käyttäjien kokonaislukumäärällä [SdFb]. Graafista, jossa kunkin tägin painoarvoa seurataan ajan suhteen, voidaan erottaa piirteitä joista voidaan tehdä johtopäätöksiä tunnettuuden muutosten syistä. Speroni di Fenizio kuvailee lähteessä [SdFc] joitakin tyypillisiä piirteitä näissä graafeissa ja näiden mahdollisia tulkintoja. Tämän tutkielman kannalta kiinnostavin piirre on graafissa nopeasti nouseva käyrä, joka implikoi yhteisössä kasvavaa tietoisuutta käyrää vastaavasta (käyttäjäkunnalle) uudesta sanasta.

4.4.1 Tägijakauman stabiilius

Artikkeli [HRS07] on yksi monista, jotka tutkivat tägien ja tägipivien käyttäytymistä ajan myötä. Kuten monet muutkin tutkimukset[5], myös tämä tutkielma toteaa tägien jakautumisen noudattavan potenssilakia (engl. *power law*) siten, että käytetyimpiä tägejä käytetään eksponentiaalisesti verrattuna vähemmin käytettyihin. Tämän käyttäytymisen syihin ei tässä (eikä usein muissakaan) julkaisussa perehdytä eikä niitä pyritä selittämään, mutta useimmat julkaisut hyväksyvät tämän yleisenä ominaisuutena luonnollisesti syntyneelle ja stabiilille tägidatalle.

Lisäksi lisäksi Halpinin et al. tutkielmassa pyrittiin mittaamaan, kuinka nopeasti tägipilvi stabiiloituu tähän muotoon. Tutkimuksessa käytettyä stabiliuden mittaa kuvaillaan tässä tarkemmin, koska kyseistä mittaa käytetään tässä tutkielmassa visualisaation alkupisteen löytämiseksi tutkittaessa tägipalveluiden muutoksia. Tulee kuitenkin huomata, että tämä menetelmä ei ole sidottu tähän käyttötarkoitukseen vaan sitä voidaan soveltaa yleisesti tutkittaessa tägipilven stabiloitumista.

Tutkielmassa käytettiin Kullback-Leibler -etäisyyttä [HRS07, luku 5] kuvaamaan tägipilven stabiiliutta. Kullback-Leibler -etäisyyden määritelmä, jossa P ja Q ovat

[5]Esimerkkeinä muista julkaisuista, joissa on tehty sama havainto, mainittakoon [SW05] ja [ZWY06].

todennäköisyysjakaumia kuvaavia vektoreita, on seuraava:

$$D_{KL}(P,Q) = \sum_{x:P(x)>0=>Q(x)>0} P(x)\log(\frac{P(x)}{Q(x)})$$

Tämä funktio on aina positiivinen, konveksi funktio, eli $D_{KL}(P,Q) \geq 0, \forall P,Q$. Lisähuomautuksena funktio ei ole symmetrinen, eli $D_{KL}(P,Q) \neq D_{KL}(Q,P)$ suurimmassa osassa tapauksia ja $D_{KL}(P,Q) = 0$ jos ja vain jos $P \equiv Q$.

Tägien tutkimisen yhteydessä Kullback-Leibler -etäisyysmittaa voidaan käyttää vertailemaan kahden ajanhetken välisten tägipilvien eroa asettamalla P ensimmäisen ajanhetken tägipilven tägien esiintymistodennäköisyysvektoriksi ja Q vastaavasti toiselle ajanhetkelle. Stabiiliusmittaukseen soveltuvasti etäisyysmittaa käytettiin Halpinin et al. artikkelissa [HRS07] kahdella eri tavalla. Ensinnäkin mittaa käytettiin vertailemaan kunkin mittaushetken tägipilveä lopulliseen tägipilveen, jolloin tuloksena saatiin mitta tägipilven stabiiliudesta kunakin ajahetkenä, olettaen että viimeinen mittaushetki on jo stabiili. Toisekseen mittaa käytettiin vertaillessa kunkin peräkkäisten mittaushetkien tägipilviä keskenään, jolloin mittauksesta käyvät ilmi äkilliset lokaalit muutokset. Kummassakin tapauksessa pienempi etäisyys tulkitaan suuremmaksi stabiiliudeksi.

5 Konstruktio

Toteutuskieleksi valittiin Java vedoten kielen yleiskäyttöisyyteen. Lisäksi kielelle on olemassa runsas määrä valmiiksi toteutettuja ja vapaaseen levitykseen julkaistuja luokkakirjastoja, joiden avulla kaikkia konstruktion ominaisuuksia ei tarvitse toteuttaa alusta alkaen. Tutkielman tekemisessä käytettiin apuna seuraavia kirjastoja:

- JSON in Java[1]:
  Tämä kirjasto sisältää rajapinnan JSON-muotoisen tekstin käsittelyyn. Kirjastoa käytettiin Delicious-palvelusta haetun datan käsittelemiseksi ja tallentamiseksi. Lisätietoa datan hausta ja tallentamisesta on luvussa 5.3.

- JGraphT[2]:
  Yksinkertainen ja helposti muokattava graafikirjasto. Kirjastoa käytettiin graafeihin perustuvien menetelmien testaamisen ja tutkimisen apuna. Lopullisessa toteutuksessa kirjastoa ei enää käytetty.

Lisäksi prosessin alkuvaiheissa testattiin myös kirjastoa HypergraphDB[3], koska tietomalliltaan hypergraafimuotoisen folksonomian tallentamisen hypergraafimuotoiseen tietokantaan tuntui luonnolliselta idealta. Käytännössä kirjasto osoittautui kuitenkin liian monimutkaiseksi ja raskaaksi käytettäväksi näin yksinkertaisessa toteutuksessa.

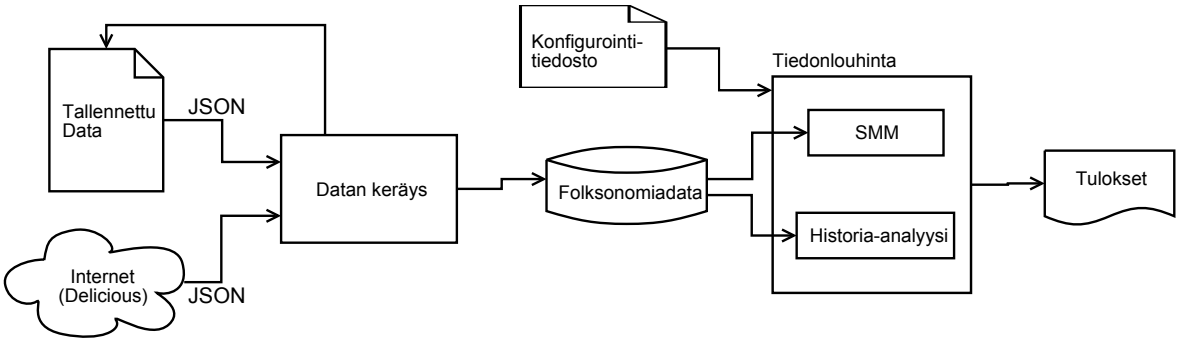
5.1 Prosessi

Konstruktion arkkitehtuuri on jaettavissa kolmeen osaan. Nämä osat ovat datan keräävä osa, datan kokoava osa ja tiedonlouhinnan suorittava osa. Datan keräävä osa kerää käsiteltävän datan Delicious-palvelusta ja yhdistää sen aiemmilla datankeräyskerroilla tallennettuun dataan. Haluttaessa keräys voidaan rajoittaa vain valmiiksi tallennettuun dataan, esim. prosessin nopeuttamiseksi. Datan kokoavan osan

[1] `http://json.org/java/`
[2] `http://www.jgrapht.org/`
[3] `http://www.hypergraphdb.org/`

tehtävä on toimia datan hakevan osan ja tiedonlouhintaosan välisenä rajapintana, jolloin hakevan osan ei tarvitse ottaa huomioon, kuinka dataa tullaan käyttämään ja tiedonlouhintaosan ei tarvitse välittää datan lähteestä tai lähteessä esiintyvän datan muodosta.

Kuva 5.1 kuvaa tämän prosessin yksinkertaistetussa muodossa.



Kuva 5.1: Yksinkertainen kuvaus konstruktion toiminnan periaatteesta

5.2 Toteutus

Datan hakua varten luotiin luokat `DeliciousReader` datan hakemiseksi Delicious-palvelusta ja `SavedDataReader` kovalevylle tallennetun datan lukemiseksi. Näiden toimintaa käsitellään tarkemmin luvussa 5.3.

Datan kokoava osana toimii käytännössä `FolksonomyStorage`-luokka, johon perehdytään tarkemmin luvussa 5.4.

Tiedonlouhintaa varten ei luotu omia luokkia vaan louhinta-algoritmit integroitiin `FolksonomyStorage`-luokan metodeiksi. Mikäli algoritmit halutaan kuitenkin erottaa tästä luokasta, on yksi suhteellisen yksinkertainen mahdollisuus tämän tekemiseksi toteuttaa algoritmit `FolksonomyStorage`-luokan alaluokkien metodeissa, jolloin algoritmeilla on edelleen yksinkertainen pääsy `FolksonomyStorage`-luokan dataan, vaikka metodit on erotettu itse yläluokasta. Toteutettuja algoritmeja käsitellään tarkemmin luvussa 5.5.

Kuvassa 5.2 kuvataan näiden luokkien tärkeimmät metodit ja riippuvuudet.

5.3 Datan keräys ja tallennus

Konstruktion ainoa ulkoinen datan lähde on Delicious-palvelu ja datan hausta täältä huolehtii `DeliciousReader`-luokka. Luokka käyttää hyväkseen Deliciousin julkisia JSON-syötteitä ja tarjoaa julkiset metodit tiettyyn tägiin, tägiryhmään ja URL:aan liittyvien syötteiden lukemiseksi. Luokka käsittelee syötteiden dataa JSON in Java -luokkakirjaston olioiden avulla.

Koska konstruktiolla on vain yksi ulkoisen datan lähde, on luontevaa tallentaa data levylle samassa muodossa kuin se luetaankin, eli JSON-muotoisena tekstitiedostona. Kutakin toteutettua ja tallennettavaa hakua kohden luodaan yksi tekstitiedosto. `DeliciousReader`-luokkaan on toteutettu ominaisuus, jossa luokka täydentää hakuparametreja vastaavan tiedoston JSON-syötteen uudella haetulla datalla käyttäen hyväkseen SavedData-luokkaa. Ominaisuus on toteutettu yksinkertaisesti lukemalla tallennetun datajoukon ensimmäisen kirjanmerkin tallennushetki. Tämän jälkeen Delicious-palvelusta haetusta syötteestä luetaan vain tätä hetkeä uudemmat kirjanmerkit, jotka lisätään tiedoston alkuun järjestyksessä uusimmasta vanhimpaan.

Kuva 5.3 kuvaa esimerkinomaisesti sekvenssin, jossa URL:aan liittyvä JSON-syöte luetaan Delicious-palvelusta ja yhdistetään tallennettuun dataan. Tägi- ja käyttäjäsyötteet luetaan vastaavalla tavalla. Kussakin toteutuksessa on omat huomioonotettavat ominaisuudet, kuten URL:n syötteen haku Delicious-palvelusta sen MD5-koodin mukaan tai mahdollisuus hakea tägiyhdistelmien syöte.

Toteutus on riittävä tämän tutkielman osalta, mutta ei optimaalinen. Siinä on kaksi ongelmaa, jotka tulisi mittavammassa toteutuksessa ratkaista. Ensinnäkin eri tiedostot voivat hyvinkin sisältää duplikaatteja toistensa sisältämästä datasta. Tämä pitää etenkin paikkansa tägiryhmien hakuihin liitetyistä tiedostoista, jotka sisältävät suurella todennäköisyydellä samoja resursseja kuin ryhmän yksittäisiin tägeihin kohdistuneiden hakujen tiedostot (mikäli näitä on olemassa). Toisekseen, mikäli sovellukseen halutaan liittää toisenlaisia datalähteitä, tulee joko uudesta lähteestä saatava data muokata Deliciousin JSON-feedeja vastaavaan muotoon tallentamista varten, tai vaihtoehtoisesti tulee luoda uusi persistentti tietorakenne, joka pystyy tallentamaan kaikkien käytettyjen lähteiden datan.

5.4 Datan kokoaminen

FolksonomyStorage-luokan tallentama data on periaatteessa joukkojen *käyttäjät*(A), *tägit*(T), *resurssit* (I) ja *tägäyshetket* (D) muodostama joukko $F \subseteq A \times T \times I \times D$. Java ei kuitenkaan sisällä tietorakennetta, joka vastaisi tarkasti hypergraafia. Tämän vuoksi datan kokoamiseksi käytettiin `Map<K, V>`-assosiaatiotaululuokkaa, jossa `K` on avainluokka ja `V` on arvoluokka. Hypergraafirakenteen emuloimiseksi `Map<K, V>`-oliota käytettiin rekursiivisesti, luoden `Map<K, V>`-olio, jossa avaimena on URL ja arvona `Map<K, V>`-olio, jossa avaimena on käyttäjä ja arvona `Map<K, V>`-olio, jossa avaimena on tägi ja arvona tägäyshetkeä kuvaava olio. Koska tägäyshetkeä kuvataan järjestelmässä `Calendar` luokan oliolla ja sekä käyttäjää, URL:aa että resurssia `String`-luokan olioilla, on lopullinen folksonomian hypergraafia kuvaava olio muotoa `Map<String, Map<String, Map<String, Calendar>>>`.

5.5 Louhinta-algoritmit

5.5.1 Louhintamenetelmien valinta

Keskitytään aluksi hylättyihin menetelmiin. Luvussa 4.2.1 kuvattu menetelmä antoi lupaavia alustavia tuloksia, kun menetelmästä tehtiin alustava, lähteen [HRS06, luku 6] ideaa mukaileva toteutus. Valitettavasti menetelmän kuvauksen tarkempi tarkastelu paljasti tästä joitakin sisäisiä ristiriitoja ja epäselvyyksiä, joiden vuoksi menetelmä hylättiin. Kuvatun menetelmän perusidea on kuitenkin tarkastelemisen arvoinen, mikäli menetelmä ollaan valmis purkamaan perusideaansa ja rakentamaan tämän päälle uusi, ristiriidaton säännöstö.

Luvussa 4.2.2 kuvattu menetelmä puolestaan vaikutti sisäisesti ristiriidattomalta. Mutta kuten lähde [BKS06] toteaakin, on spektraalisen ryvästysalgoritmin pohjalla toimiva matemaattinen malli varsin monimutkainen ja on suositeltavaa käyttää tätä varten valmiiksi kehitettyä kirjastoa. Käytännössä menetelmä osoittautui liian työlääksi toteutettavaksi tässä tutkielmassa ja ilman lähteen suosittelemia valmiita kirjastoja menetelmä jouduttiin toteutuksen puuttumisen vuoksi hylkäämään.

Luvuissa 4.3 ja 4.4 esitellyt menetelmät toteutettiin seuraavissa luvuissa esitetyillä tavoilla.

5.5.2 SMM-mallin mukainen EM-algoritmi

SMM-mallin mukainen louhinta käyttää luvussa 4.3 kuvattua algoritmia datassa esiintyvien tägien ryvästämiseen. Tätä algoritmia ei kuitenkaan kannata toteuttaa suoraan matemaattisen mallin mukaan, sillä kaikkien $\langle R_{r\alpha} \rangle$-muuttujien tallentamiseen vaadittaisiin αL:n numeroarvon tallentamista. Algoritmi, jossa näiden arvojen tallentamisen tarve on ohitettu, on kuvattu algoritmilistauksessa 2.

Huomionarvoista on myös, että algoritmi vaatii alkuarvauksen ennen ensimmäistä iteraatiota. Kuten liitteessä C todetaan, tasainen todennäköisyysjakauma ei sovellu alkuarvaukseksi. Ilman loogisesti datasta johdettavaa alkuarvausta todennäköisyysjakaumasta on perusteltua käyttää satunnaista alkuarvausta. Valitettavasti alkuarvauksen vaikutus algoritmin tulokseen on merkittävä, joten satunnaisuuden tuottama variaatio tuloksissa tulee mitätöidä tavalla tai toisella.

Tässä tapauksessa variaation mitätöintiä varten yhden suorituskerran tulos yksinkertaistettiin tarkasteltavan tägin t' suhteen määrittelemällä muuttuja

$$c_{t|t'} = \sum_{\alpha : q_{\alpha|t'} \geq W/K} q_{\alpha|t} q_{\alpha|t'},$$

jossa t' on tarkasteltava tägi, t tägi, jonka relevanssia tarkasteltavaan tägiin tutkitaan, K on konseptien määrä. W on kerroin, joka kuvaa kuinka monta kertaa yli keskimääräisen arvon $q_{\alpha|t'}$:n on oltava, jotta se otetaan mukaan laskuihin (esim. edellä mainittu 1,2). Käytännössä tämä muuttuja approksimoi todennäköisyyttä, että tägi t kuuluu samaan konseptiin kuin tägi t' (SMM-mallin mukaan), muuttaen algoritmin antaman K:n konseptin joukon yhdeksi, tägille t' relevantiksi konseptiksi.

Varsinainen variaation vaikutuksen mitätöinti tehtiin toistamalla algoritmi 7 kertaa eri alkuarvauksilla. Iteraatioiden tuloksista laskettiin yhteen kunkin tägin t edellä määriteltyjen muuttujien $c_{t|t'}$ summa. Tämän jälkeen tägijoukosta karsittin pois kaikki, joiden muuttujien summa oli alle 4 epärelevanttien tägien karsimiseksi. Luvut 7 ja 4 ovat suoritukselle tässä tutkielmassa annettuja parametreja. Kertauksen määränä 7 todettiin sopivaksi ollen tarpeeksi suuri kertausten lukumäärä satunnaisuuden vaikutuksen minimoimiseksi ja tarpeeksi pieni, jotta algoritmin suoritusaika olisi käytännöllinen. Tulokset ovat luettavissa liitteessä A, jossa $c_{t|t'}$:en summista on laskettu keskiarvo (jakamalla summat 7:llä), jotta tulokset olisivat verrattavissa todennäköisyysarvoihin.

Algoritmi 2 SMM-malliin sovitettu EM-algoritmi, 1 iteraatio

$logsum \leftarrow 0$
for $r = 1 \rightarrow L$ **do**
  $divsum_r \leftarrow 0$
  for $v = 1 \rightarrow K$ **do**
    $divsum_r \leftarrow divsum_r + \pi_r p_{i(r)|v} q_{j(r)|v}$
  end for
end for
$p^{(new)} \leftarrow 0$ { muuttujilla $p^{(new)}$ ja $q^{(new)}$ on p:n ja q:n dimensiot}
$q^{(new)} \leftarrow 0$ { alustetaan muuttujat $p^{(new)}$ ja $q^{(new)}$ nollilla}
for $\alpha = 1 \rightarrow K$ **do**
  $\pi_{old} \leftarrow \pi_\alpha$
  $\pi_\alpha \leftarrow 0$
  for $r = 1 \rightarrow L$ **do**
    $s \leftarrow p_{r|\alpha} q_{r|\alpha} / divsum_r$
    $\pi_\alpha \leftarrow \pi_\alpha + s$
    $p^{(new)}_{i(r)|\alpha} \leftarrow p^{(new)}_{i(r)|\alpha} + s$
    $q^{(new)}_{j(r)|\alpha} \leftarrow q^{(new)}_{j(r)|\alpha} + s$
  end for
  $\pi_\alpha \leftarrow \pi_\alpha \pi_{old} / L$
  $m \leftarrow \pi_{old} / L \pi_\alpha$
  for all $p_x \in p^{(new)}$ **do**
    $p_{x|\alpha} \leftarrow p_{x|\alpha} m$
  end for
  for all $q_y \in q^{(new)}$ **do**
    $q_{y|\alpha} \leftarrow q_{y|\alpha} m$
  end for
  for $r = 1 \rightarrow L$ **do**
    $logsum \leftarrow logsum + \frac{\pi_{old} p_{i(r)|\alpha} q_{j(r)|\alpha}}{divsum_r} (\log \pi_\alpha + \log p^{(new)}_{i(r)|\alpha} + \log q^{(new)}_{j(r)|\alpha})$
  end for
end for
$p \leftarrow p^{(new)}$
$q \leftarrow q^{(new)}$

5.5.3 Tägipilven muutosten seuranta

Kuten luvussa 4.4 todetaan, tälle menetelmälle ei ole tarkasti määriteltyä algoritmia valmiiksi saatavilla. Algoritmi voidaan kuitenkin johtaa olemassa olevista tiedoista suhteellisen yksinkertaisesti. Yksinkertaistettuna algoritmin askeleet ovat seuraavat:

1. valitaan aikamääre, jonka välein dataa tarkastellaan (esim. yksi päivä)
2. lasketaan datasta tägipilven stabiiliusarvo (kts. luku 4.4.1) aikamääreen välein
3. valitaan stabiliusarvojen avulla tarkasteltavan ajanjakson alkamiskohta (tässä tutkielmassa käytettiin arvoa 0.15)
4. lasketaan kunkin tägin painoarvo (kts. luku 4.4) aikamäärein välein tarkasteltavassa ajanjaksossa

Viimeisen askeleen tulokset voidaan piirtää luvussa 4.4 kuvattuna graafina. Jotta graafista pystyttäisiin karsimaan turhat kuvaukset pois, voidaan piirrettäviä viivoja rajoittaa. Ottaen huomioon tutkielman tavoitteen, rajattiin tarkasteltavat tägit niihin, joiden painoarvo kasvoi tarkastelujakson aikana. Näitä rajattiin vielä karsimalla pois ne tägit, joiden painoarvo ei lopussa ylittänyt ennalta asetettua rajaa (0.02). Lopuista tägeistä valittiin ne, joilla muutos painoarvossa tarkasteltavan ajanjakson alussa ja lopussa on suurin. Viimeisellä ehdolla tarkasteltavien tägien määrä rajattiin enintään 20:een.

5.6 Käytännössä todettuja ongelmia

Konstruktion toteutus käytännössä ei ollut aivan ongelmaton prosessi (joitakin ongelmia mahdollisesti tätä ennen jo esitetty). Tässä luvussa esitetään esiin nousseita ongelmia ja mahdollisia ratkaisuja, jotta nämä voitaisiin ottaa huomioon muissa vastaavissa toteutuksissa.

5.6.1 Muistin käyttö

Suuri käsiteltävän datan määrä johtaa myös huomattavaan muistin käyttöön dataa käsitellessä. Ongelmaa voidaan lievittää tämän konstruktion arkkitehtuurista poikkeavilla arkkitehtuureilla, joissa vältellään kaiken käytettävän datan tallentamista

muistiin. Lisäksi muistia vievät eri muodot, joihin data muutetaan algoritmien ajon aikana, tässä toteutuksessa esimerkiksi SMM-mallin todennäköisyystaulukot $p_{i|\alpha}$ ja $q_{j|\alpha}$ sekä tägipilven historiadata.

Arkkitehtuurin valinnasta huolimatta toteutus joutuu luultavasti käyttämään huomattavia määriä muistia. Käytettäessä Javaa toteutuskielenä, on otettava huomioon JVM:n (Java Virtual Machine) rajoitettu kekomuisti. Sunin JVM-toteutuksissa kekomuistin määrä ei ole dynaaminen, vaan se kiinnitetään käynnistyksen yhteydessä. Tämä määrä on joko vakioarvo tai `-Xmx` argumentilla asetettu luku.

5.6.2 SMM

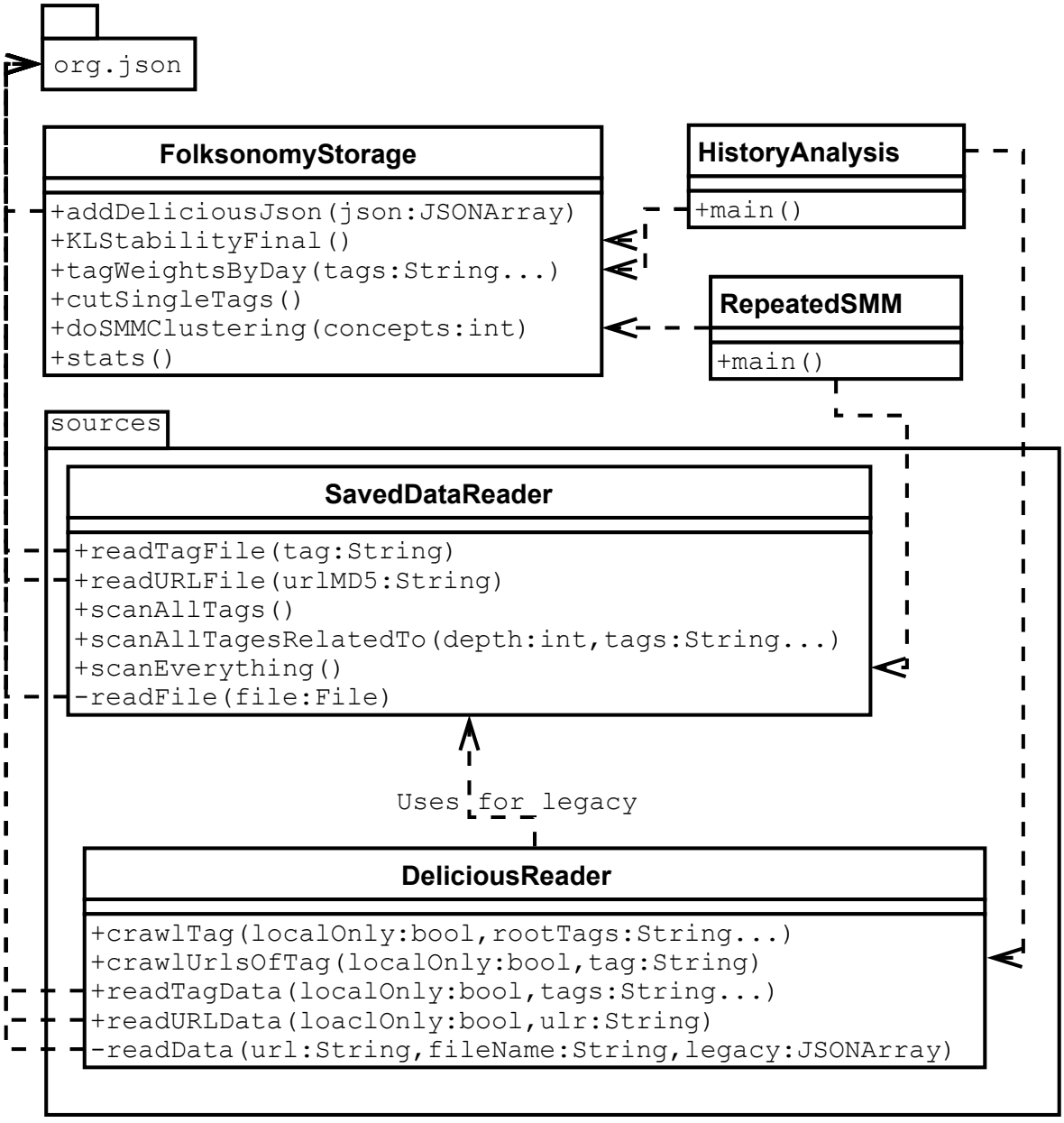
Menetelmää käytettäessä törmättiin ongelmaan, jossa ohjelmointikieli ei enää kyennyt käsittelemään ryvästyksessä tapahtuvia pieneneviä todennäköisyyksiä vaan alkoi tallentamaan liian pieniä lukuja nollina. Tämä taas johti ongelmiin logaritmisumman laskennassa. Tässä tapauksessa logaritmi nollasta tulkittiin negatiiviseksi äärettömäksi, joka johti koko logaritmisumman negatiiviseen äärettömyyteen. Ratkaisuna logaritmien arvo rajoitettiin negatiiviseen kolmannekseen suurimmasta mahdollisesta arvosta.

5.6.3 Tägipilven muutokset

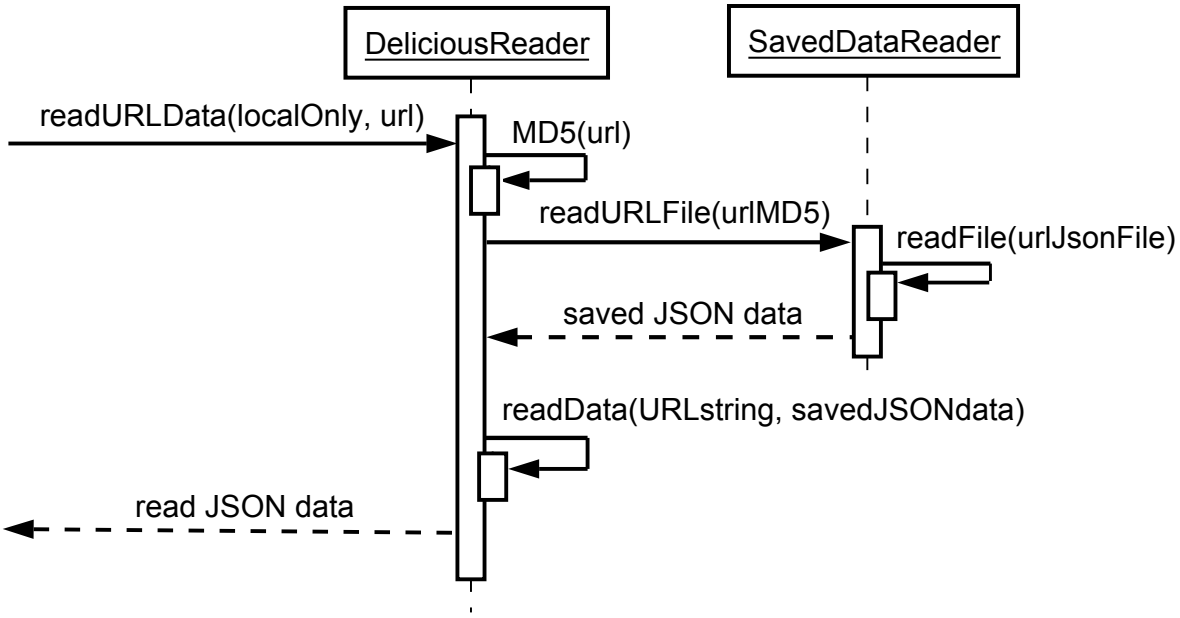
Tiedonkeräystapa on haavoittuva tapauksille, joissa yksi käyttäjä tägää suositun sivun tarkastellulla tägillä, vaikka tämä tägi ei sivuun liitykään. Tämän seurauksena, kun dataa haetaan tältä sivustolta, seuraa piikki epäolennaisten tägien tunnettavuudessa. Esimerkiksi 18. kesäkuuta käyttäjä tägää sivun `http://www.delicious.com/help/bookmarklets` tägillä `ax`. Tämän seurauksena vahva nousu tägin `delicious` tunnettuuden nousussa.

5.7 Kehitysehdotus

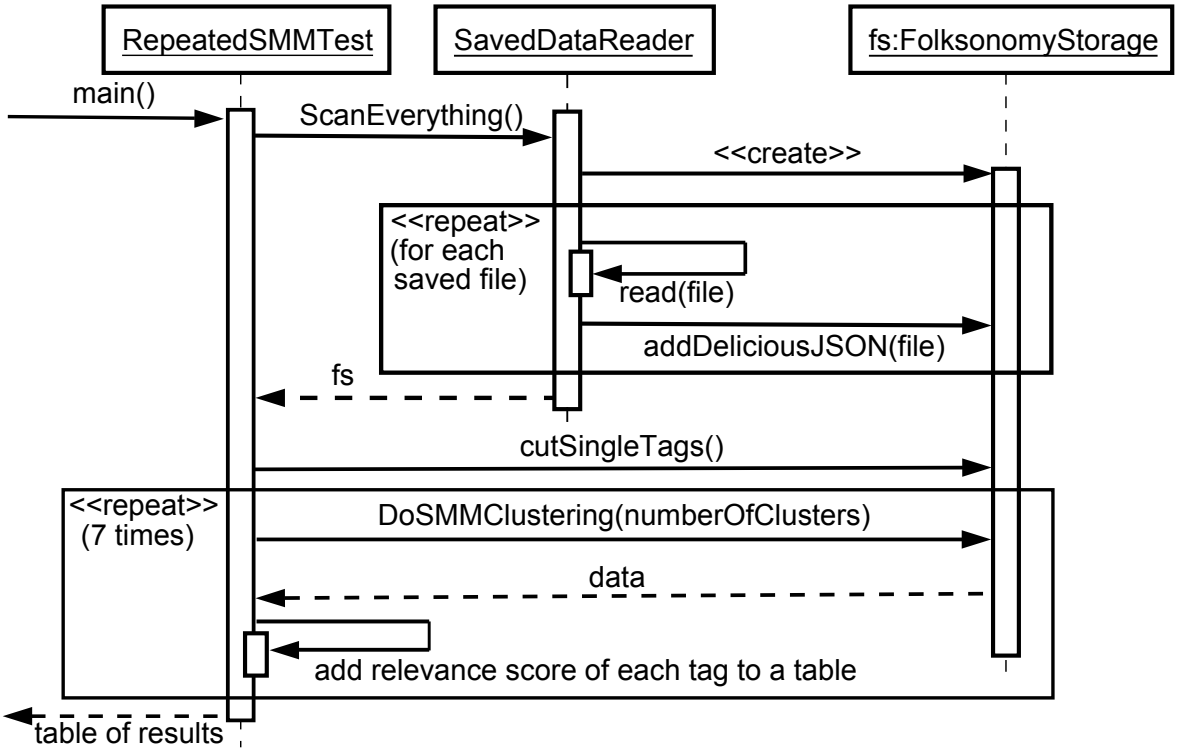
Tällä hetkellä konstruktiossa käytetään vain yhtä tägäyspalvelua. Lähde [Gru] kuvaa tarkemmin yhtä vaihtoehtoa folksonomian mallintamisesta, kun tägien lähteitä on useita. Lähteessä paneudutaan myös joihinkin lähteiden yhdistämisestä seuraaviin ongelmiin ja niiden ratkaisuihin.



Kuva 5.2: Yksinkertaistettu UML-luokkakaavio toteutuksessa käytetyistä luokista



Kuva 5.3: URL-datan hakuprosessi UML-muotoisena sekvenssikaaviona

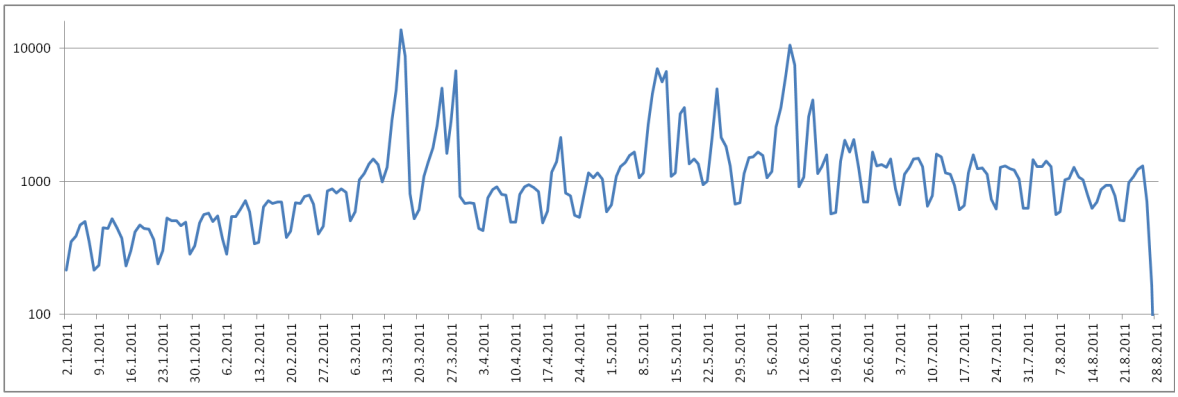


Kuva 5.4: SMM-mallin mukaisen EM-algoritmin UML-mallinen sekvenssikaavio

6 Tulokset

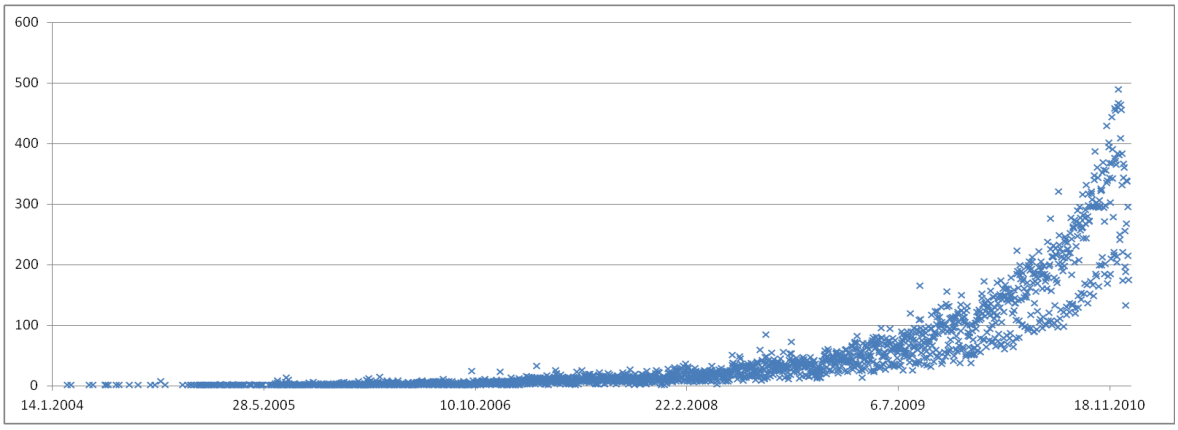
6.1 Data

Ennen tuloksien tarkempaa analyysia, on aiheellista tarkastella tarkemmin käytettyä dataa. Data haettiin Delicious-palvelun tarjoamista JSON-syötteistä vuoden 2011 helmikuun ja syyskuun välisenä aikana. Data koostuu 1831 eri tägistä saadusta syötteistä, 13574 eri URL:n syötteestä ja yhden käyttäjän syötteestä. Mikäli syöte on luettu useaan kertaan, tulokset on yhdistetty tallennusta varten. Kokonaisuudessaan tallennettu syötedata on kooltaan 153 MB. Kerätty data sisältää 174930 eri URL:aa, 164383 käyttäjää ja 119740 tägiä, jotka on kerätty 446640:sta kirjanmerkistä ja muodostavat 1723186 tägäystapahtumaa. Tallennuksen yhteydessä datasta poistettiin 32663 tägiä, jotka esiintyivät tekstitiedonlouhintaan tarkoitetulla, 429 sanaa sisältävällä sulkusanalistalla[1]. Kuva 6.1 kuvaa kerättyjen kirjanmerkkien sijoittumista eri päiville vuoden 2011 aikana. Kuva 6.2 kuvaa ennen vuotta 2011 lisättyjä kirjanmerkkejä. Näitä jälkimmäisiä on datassa mukana lähinnä harvoin käytettyjen tägien ja URL:ien johdosta. Kuvassa 6.1 näkyy myös kirjanmerkkien määrän väheneminen säännöllisin väliajoin, 7 päivän välein. Tarkastelemalla päiviä, jolloin nämä pienemeniset tapahtuvat, huomataan niiden tapahtuvan lauantaisin ja sunnuntaisin.



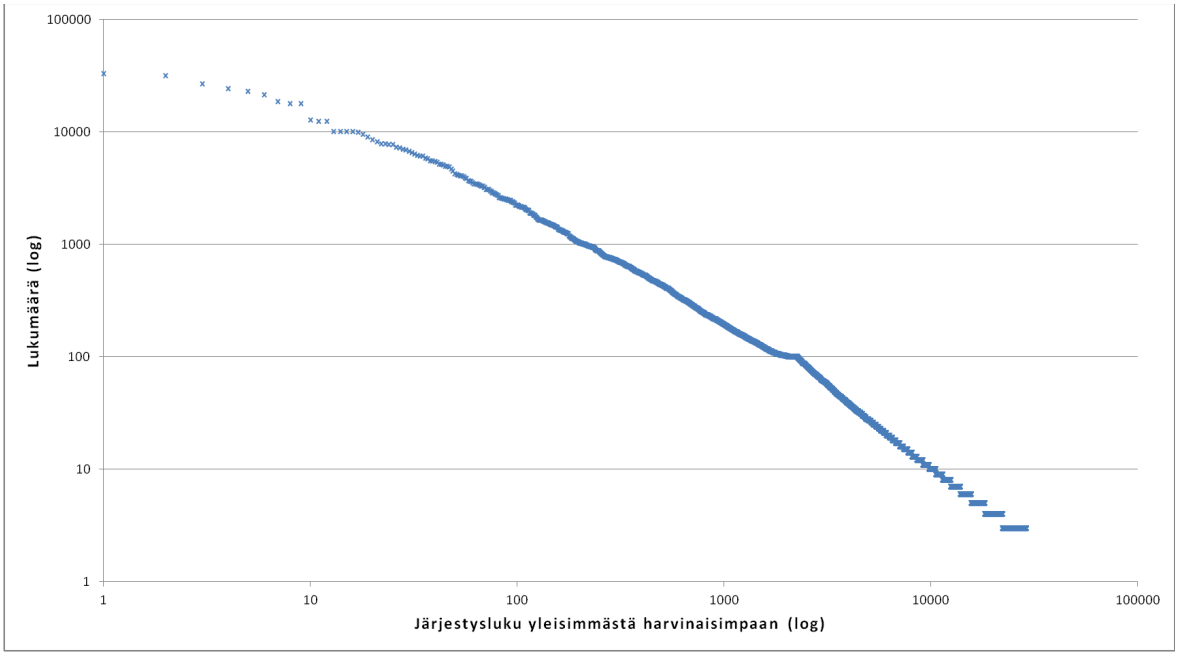
Kuva 6.1: Kirjanmerkkejä per päivä (2011)

[1] `http://www.lextek.com/manuals/onix/stopwords1.html`



Kuva 6.2: Kirjanmerkkejä per päivä (ennen vuotta 2011)

Kuten oli (luvussa 4.4.1 mainittujen aiempien havaintojen perusteella) odotettavissa, tägit noudattavat lukumäärältään potenssilakia, niin kuin kuvasta 6.3 on nähtävissä. Kuvan graafista on runsaslukuisuutensa vuoksi jätetty pois 15132 kahdesti esiintyvää tägiä ja 75697 kerran esiintyvää tägiä, jotka muodostavat jakauman pitkän hännän huipun.



Kuva 6.3: Tägien lukumäärä

Käsitelty tägi	Relevantteja tägejä	% tägijoukosta
ax	5	0,0011
android	433	1,0058
cloud	228	0,5177
iphone	293	0,6653
maemo	61	0,1385
meego	15	0,0341
qt	143	0,3247
silverlight	490	1,1125
wp7	100	0,2271

Taulukko 6.1: Yleiskuvaus SMM-ryvästyksen tuloksista

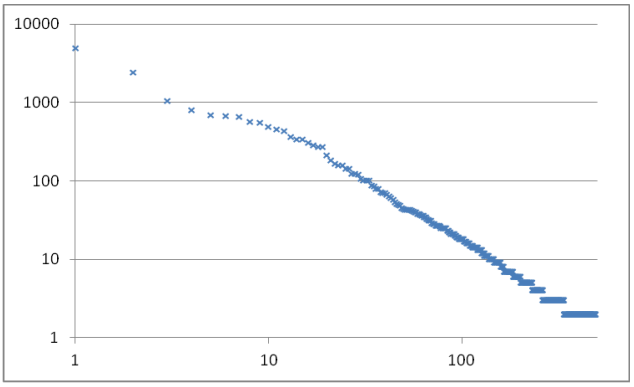
6.2 SMM-ryvästys

Algoritmi suoritettiin koko kerätylle datalle, josta oli karsittu pois vain kerran esiintyvät tägit. Tämä operaatio jättää käyttäjien, URL:ien ja kirjanmerkkien joukot samankokoisiksi, mutta tägien määrä supistuu 44043:een ja tägäyksien määrä 1647489:ään. Konseptien määräksi asetettiin $K = 20$. Luvun 5.5.2 kuvauksen mukaisesti, ryvästysalgoritmi suoritettiin kutakin tarkasteltavaa tägiä kohden 7 kertaa, joka kerta uudella satunnaisella alkuarvauksella. Tuloksista poimittiin relevanteiksi tägiehdokkaiksi tägit, joiden $c_{t|t'}$:en summa on yli 4. Kukin suoritus toistoineen kesti noin 9 tuntia.

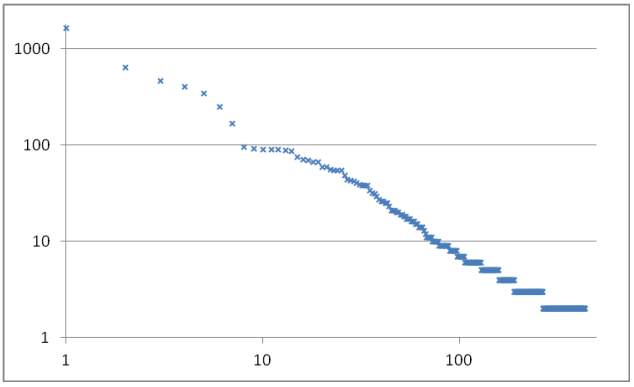
Algoritmi suoritettiin tägeille `qt`, `wp7`, `meego`, `maemo`, `ax`, `silverlight`, `cloud`, `iphone`, `android`. Yleiskuvaus suoritusten tuloksista esitetään taulukossa 6.1.

Tarkasteltaessa algoritmin relevanteiksi ehdottamia tägejä, nämä näyttävät valikoituneen tasaisesti yleisestä tägijoukosta. Tämä on nähtävissä tarkasteltaessa relevanttien tägien esiintymislukumääriä. Kuten lähdedatan tägit, tämän tägijoukon esiintymiskerrat noudattavat potenssisääntöä (esim. tägiin `silverlight` liitetyissä tägeissä kuvassa 6.4), joskin tägien `android`, `cloud`, `maemo` ja `wp7` vastaavissa kuvauksissa on havaittavissa poikkeama, jossa jakauma ei noudata (molempien akselien ollessa logaritmiskaalassa) yhtä suoraa, vaan kahta erisuuntaista suoraa, jotka kohtaavat (vaaka-akselilla) lähellä järjestyksessä kymmenettä tägiä. Esimerkkinä kuvauksen mukaisesta visualisaatiosta on kuvassa 6.5 esitetyt tägiin `android` liittyvien tägien esiintymismäärät.

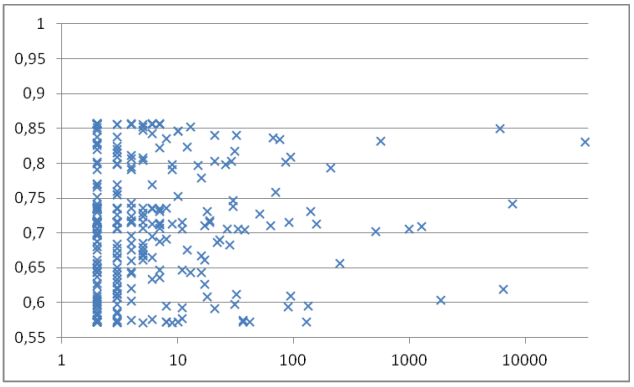
Mainittakoon vielä, että relevanttien tägien lukumäärällä ja tägin t ryvästäytymistä yhteen tägin t' kanssa kuvaavalla $c_{t|t'}$-arvolla ei vaikuta olevan vahvaa korrelaatiota. Tämä on nähtävissä kuvassa 6.6, joka kuvaa tägiin `iphone` liitettyjen tägien lukumäärän ja keskimääräisen $c_{t|iphone}$ arvojen suhteita. Kuvasta nähdään, että arvot ovat jakautuneet tasaisesti tarkastellulle välille kaikilla tägien lukumäärillä. $c_{t|iphone}$ arvojen rajoittuminen tietyn lukuarvon (tässä tapauksessa 8,5:n) tienoille on ilmiö, joka on havaittavissa myös tägeihin `cloud`, `silverlight` ja `qt` liittyvissä tuloksissa. Tämä johtunee arvon laskentatavasta, joka jättää täysin huomioimatta SMM-mallin konseptit, joihin tarkasteltavan tägin kuulumisen todennäköisyys on alle tietyn raja-arvon (tässä tutkielmassa 6%). Korkean, mutta rajan alle jäävän konseptin poisjättäminen voi näin aiheuttaa systemaattisen arvojen alenemisen.



Kuva 6.4: Tägiin `silverlight` liitettyjen tägien esiintymislukumäärät (vaaka-akselilla järjestyksessä suurimmasta pienimpään)



Kuva 6.5: Tägiin `android` liitettyjen tägien esiintymislukumäärät (vaaka-akselilla järjestyksessä suurimmasta pienimpään)



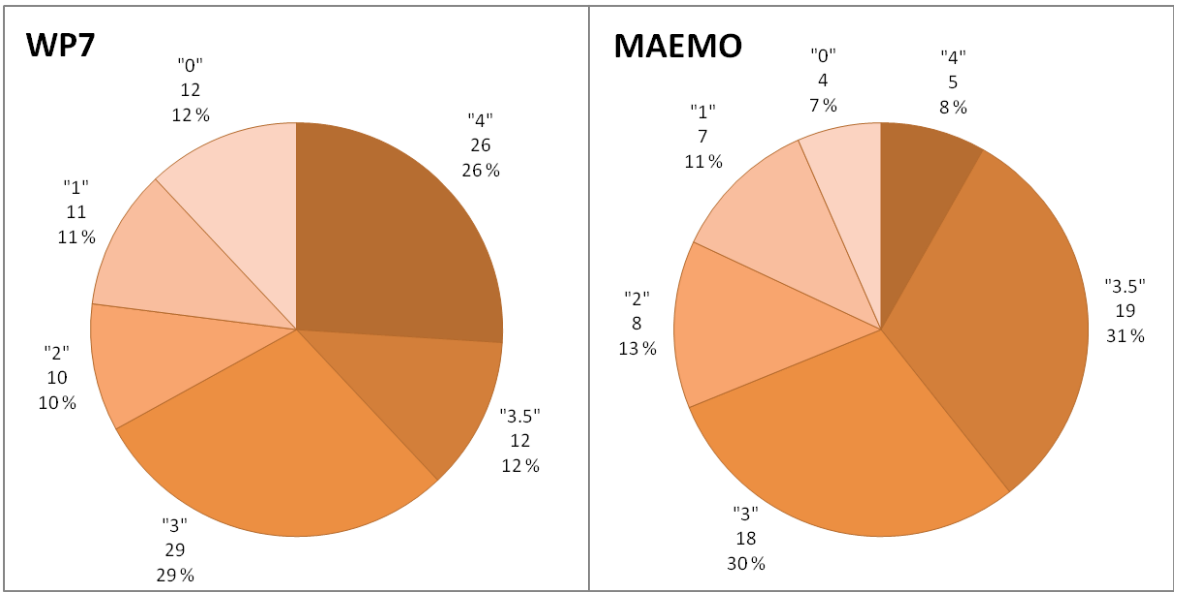
Kuva 6.6: Tägiin `iphone` liitettyjen tägien lukumäärien ja keskimääräisten $c_{t|iphone}$ arvojen suhteet

Vaikka edellä algoritmin valikoimia tägejä onkin kutsuttu nimellä "relevantit tägit", tägien varsinaisen relevanttiuden varmistaminen on varsin haastava työtehtävä. Kuten taulukosta 6.1 on nähtävissä, vaikka algoritmi rajaakin datan tägijoukosta hyvin rajallisen osan (n. 1,1% - 0,01%), on tämä tägijoukon koon takia silti varsin suuri joukko, jonka tägien määrä on usein laskettavissa sadoissa. Jotta voisimme kuitenkin arvioida algoritmin kykyä löytää relevantteja tägejä, tarkastellaan tägeihin `wp7` ja `maemo` liittyviä tägejä, jotka on listattu liitteessä A.

Tarkastelua varten kunkin ryppään tägin relevanttiutta arvioitiin asteikolla 0-4, jossa 4 kuvaa tägin synonyymia, 3 kuvaa vahvasti tägin aiheeseen liittyvää avainsanaa, 2 kuvaa yleiskäyttöistä avainsanaa, joka kuitenkin on yhdistettävissä tägiin tavalla tai toisella, 1 kuvaa lähdettä kuvaavaa avainsanaa ja 0 kuvaa avainsanaa, jonka yhteyttä tägiin ei saatu selville. Skaala ei ole absoluuttinen ja on usein tulkinnanvaraista kuuluuko jonkin avainsana esimerkiksi ryhmään 2 vai 3. Lisäksi, koska kummastakin esimerkistä löytyi suuri määrä hyvin samanlaisia tägejä (Windows Phone 7 -ohjelmistokehitykseen viittaavia tägejä tägin `wp7` yhteydessä ja Nokiaan ja tämän kännykkämalleihin viittaavia tägejä tägin `maemo` yhteydessä), luotiin näitä varten vielä ryhmä 3.5.

Algoritmin valikoimat tägit ryhmittyvät edellä kuvattuun luokitteluun kuvan 6.7 kuvaamalla tavalla. Epärelevanttien luokkien 0 ja 1 osuus on molemmissa esimerkeissä pieni, noin 10% kutakin luokkaa kohden. Aidosti relevanttien ja kiinnostavien tägien osuus, ryhmä 3, on molemmissa esimerkeissä kiitettävän suuri, noin 30%. Kuvajoukko 6.8 taas kuvaa, mikä on ryhmän tägien yleinen lukumäärä ja keskimääräisen $c_{t|t'}$ arvon suhde (vrt. kuva 6.6). Kuvissa esiintyvät tägit ovat sekä tägille `wp7` että tägille `maemo` relevanttien tägien joukkojen yhdiste. Näistä kuvista on

nähtävissä, että ryhmien 0, 1 ja 2 tägejä on käytetty melko vähän, yhtä poikkeusta lukuun ottamatta enintään 10 kertaa. Rymissä 3.5 ja 4 taas tägien $c_{t|t'}$ arvo on muita ryhmiä suuremmalla todennäköisyydellä yli 0,8 tägin käyttömäärästä riippumatta. Mainittakoon, että nämä johtopäätökset eivät välttämättä yleisty tarkasteltua alemmille $c_{t|t'}$ arvoille. Jotta johtopäätösten yleistettävyys voitaisiin vahvistaa, tulisi analyysi toistaa datalle, jossa pienempi $c_{t|t'}$ arvoiset tägit ottaa analyysiin mukaan.



Kuva 6.7: Tägeihin `wp7` ja `maemo` liitettyjen tägien relevanttiusjakaumat

6.3 Tägipilven muutokset

Tätä algoritmia ei suoritettu koko datalle. Sen sijaan data kerättiin valitsemalla mielenkiintoinen avainsana. Tämän jälkeen haettiin avainsanan mukaisen tägin syöte. Tästä datasta käytiin läpi vielä kaikki kerätyt URL:t ja näihin liittyvät syötteet lisättiin vielä dataan. Edellä mainituiksi mielenkiintoisiksi avainsanoiksi valittiin `android`, `ax`, `ce`, `cloud`, `hibernate`, `iphone`, `maemo`, `meego`, `n9`, `qml`, `qt`, `silverlight`, ja `wp7` sekä tägien `cloud` ja `computing` yhdistelmä. Datan valitsemistavan tarkoituksena oli karsia tutkittavasta datajoukosta pois tägejä, joita ei voitu pitää kiinnostavina. Tarkastellessa liitteessä B esiteltyjä tuloksia, tavoitteeseen päästiin pääosin, joskin datankeruutavasta löytyi luvussa 5.6.3 kuvattu heikkous, joka ilmenee graafissa B.2.

Tägi	URL:eja	käyttäjiä	tägejä	kirjanmerkkejä	tägäyksiä
android	2187	29549	3210	45753	135334
ax	232	1483	266	1707	2872
ce	439	5125	938	5981	14189
cloud	2242	38331	4559	58525	186770
+computing	892	13540	1783	16464	47181
hibernate	744	8384	1086	13890	42367
iphone	2349	42115	4656	68443	220026
maemo	340	2694	525	3561	11206
meego	621	3247	642	4049	13242
n9	241	692	200	828	2622
qml	163	295	106	439	1253
qt	1276	8478	1321	12306	39976
silverlight	1710	9896	1840	21332	65304
wp7	1815	7854	1323	13130	38627

Taulukko 6.2: Yleiskuvaus datajoukoista, joiden tägipilvien historiaa tutkittiin

Tavoitteena oli löytää lähteessä [SdFc] kuvattuja nopeita kulttuurin muutoksia (engl. *rapid cultural changes*), jotka ilmenevät tulosgraafeissa nousevina käyrinä. Huomattavana erona lähteessä kuvattuun toteutukseen, lähteessä olevat graafit kuvaavat muutoksia yhden URL:n tägipilvessä, verrattuna tässä tutkielmassa dataan (systemaattisesti) kerättyihin satoihin URL:eihin.

Lähteessä kuvatun nousevan `ajax`-tägin tapaisia nousuja oli selvästi havaittavissa vain tägin `silverlight` yhteydessä (kts. kuva B.13). Graafista on nähtävissä selvä, (tarkkailun päättymiseen saakka) jatkuva, noin 1.4. alkava ja muut tägit ohittava tunnettuuden nousu tägien `jquery`, `javascript`, `html5` ja `video` yhteydessä. Myös tägi `wp7` nousee selvästi tarkkailuajan alusta, mutta nousu tyrehtyy toukokuun aikana.

Tägien `android` (kuva B.1), `cloud` (kuva B.4), `hibernate` (kuva B.9) ja `qml` (kuva B.11) suhteen tägipilvissä ei ollut havaittavissa merkittäviä muutoksia. Tägien tunnettuus pysyi joko stabiilina tai nousu tapahtui jo valmiiksi hyvin tunnetulle tägille. Tägien `maemo` (kuva B.7, noin 10.4. ja 20.6., mm. tägissä `webdesign`), `meego` (kuva B.8, noin 20.6., mm. tägissä `nokia`[2]), `n9` (kuva B.10, noin 5.8., mm. tägissä `meego_phone`), `qt` (kuva B.12, useissa kohdin, mm. 6.4. tägiss `pdf`) ja `wp7` (ku-

[2]Nokian N9-älypuhelinmalli julkistettiin 21.6.2011. [Nok]

va B.14, noin 20.6., mm. tägissä `icons`[2]) sekä tägiyhdistelmän `cloud+computing` (kuva B.5, noin 27.7, tägeissä `internet` ja `technology`) kohdalla tägipilvissä oli havaittavissa hyppäyksiä, tai sosiaalisia järistyksiä (engl. *social quakes*) [3] lähteen [SdFc] mukaan, jotka voitaneen johtaa johonkin mielenkiintoiseen, tägeihin liittyvään tapahtumaan. Tällainen tapahtuma voi olla esimerkiksi että uusi, laaja käyttäjäkunta on vastaikään tullut tutuksi tarkasteltavan avainsanan kanssa.

6.4 Tulosten hyödynnettävyys

Edellisissä luvuissa käsiteltiin tutkielmaa varten kerättyä dataa ja tälle suoritetun tiedonlouhinnan tuloksia mahdollisimman monesta näkökulmasta. Nämä analyysit eivät kuitenkaan vastaa luultavasti tärkeimpään kysymykseen, jonka louhinnan tuloksista voi esittää; ovatko nämä tulokset hyödyllisiä? Johdannossa tutkielman perimmäisenä tavoitteena oli ICT-alan koulutustapahtumien suunnittelussa hyödyllisen tiedon automaattinen muodostus. Tätä varten pyrittiin löytämään kahta erilaista tietoa. Ensinnäkin pyrittiin löytämään ennalta tunnettuun tägiin epätriviaalisti liittyviä tägejä, joiden avulla suunnitteilla olevaan koulutustapahtumaan voidaan halutessa liittää lisäaiheita. Toiseksi haluttiin automaattisesti havaita tägejä, joiden tunnettuus on vastaikään lähtenyt nousuun, jotta alalle relevantteja koulutustapahtumia uusista, nousevista aihealueista voitaisiin ehdottaa ja suunnitella hyvissä ajoin.

Kuten luvussa 6.2 todetaan, SMM-mallin ryppäistä löytyy runsaasti (kahdessa käsitellyssä esimerkissä 77% ja 82%) kiinnostuksen kohteena olevaan avainsanaan liittyviä tägejä. Valitettavasti näiden tägien hyödyllisyys koulutustapahtumien suunnittelussa on marginaalinen. Molemmissa esimerkeissä tägiryppäässä on suuri joukko synonyymeja (`wp7`-ryppäässä tägin `wp7` synonyymeja ja `maemo` ryppäässä sekä tägin `maemo` että Nokian puhelinmallien synonyymeja), jotka eivät tuo lisäinformaatiota kiinnostuksen kohteen olevan avainsanan ympärillä olevista aiheista. Erottelussa ryhmä 2:ksi nimetty ryhmä (joka esimerkeissä sisältää 22% ja 13% tägeistä) sisältää lähinnä tutkitun avainsanan käyttökohteita tai tägejä, kuten `codesamples`, jotka eivät myöskään anna lisäinformaatiota itse avainsanasta.

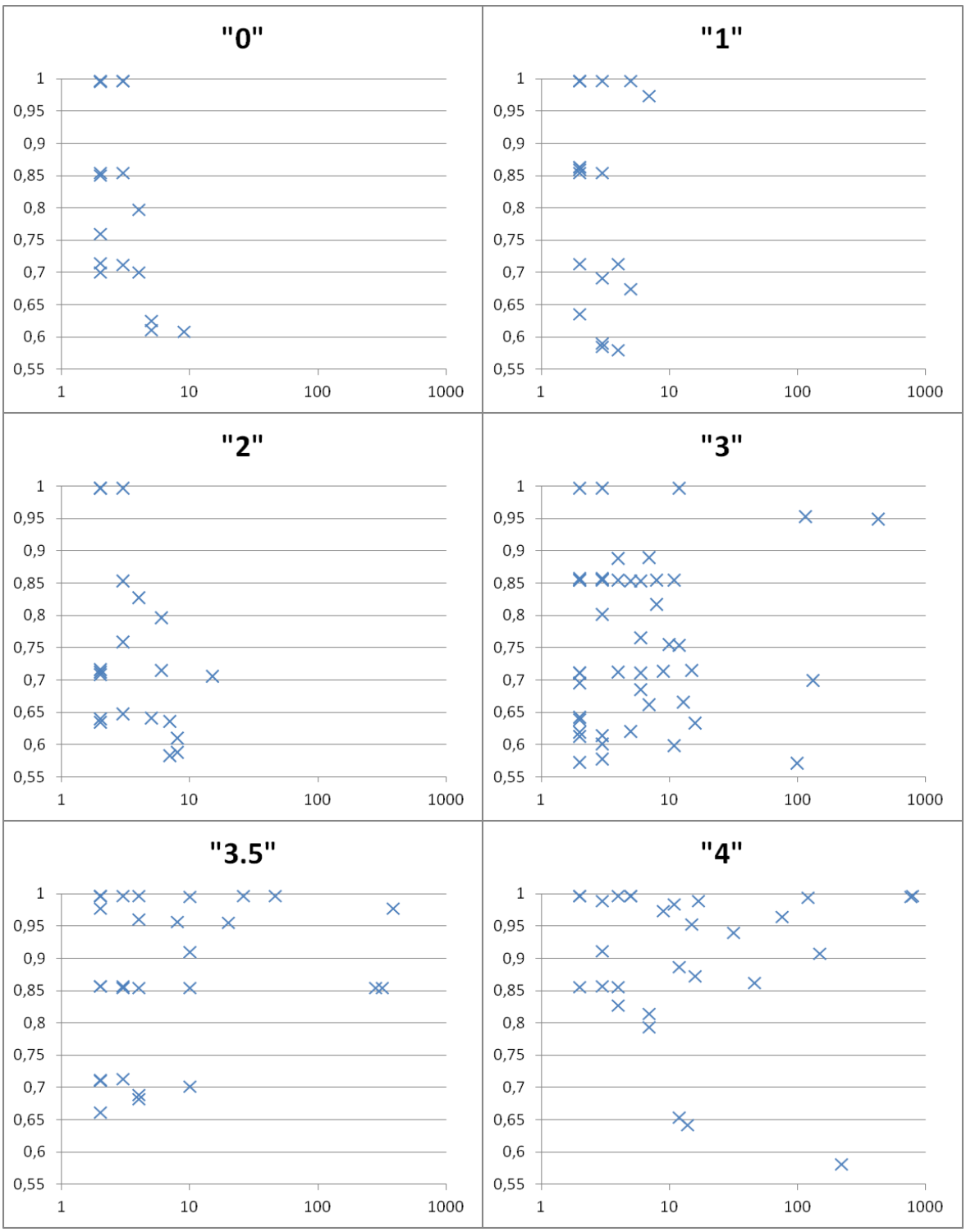
[3] Lyhyellä aikavälillä tapahtuva nousu tai lasku tägin tunnettuudessa, joka laantuu nopeasti. Speroni di Fenizion mukaan mahdollinen syy tähän on, että käyttäjäjoukko, joka käyttää keskimääräistä enemmän tägejä linkin kuvaamiseen, on tuona aikana löytänyt linkin. Tästä johtuen linkkiä kuvaavien tägien joukko ei muutu, mutta tiettyjä tägejä aletaan käyttää aiempaa useammin.

Luvun 6.2 ryhmistä mielenkiintoisimman ryhmän, selvästi juuri käsiteltyyn avainsanaan liittyvien tägien ryhmän 4, hyödyllisyys riippuu siitä, kuinka tarkasti tai yleisesti avainsanan aihetta halutaan käsitellä. Ryhmä sisältää runsaasti tägejä, jotka viittaavat avainsanaan liittyviin koodikirjastoihin, ohjelmistojen versioihin tai yksittäisten käyttöliittymäkomponenttien nimiin. Koulutustapahtumissa, joissa esimerkiksi juuri käytettävissä olevia kirjastoja halutaan esitellä, tästä tägilistasta voi olla hyötyä, mutta mikäli avainsanan aihetta halutaan kuvata yleisemmällä tasolla, eivät useimmat tämän ryhmän tägeistä ole mainitsemisen arvoisia. Tägiin `wp7` liittyen löytyy vain kaksi laajempaa konseptia kuvaavaa tägiä, `executionmodel` ja `tombstoning`.

Tägipilven historian seuranta ei myöskään tuottanut sen enempää haluttuja tuloksia. Vain yhdessä tarkastelluista 14:stä tägipilvestä oli havaittavissa tägejä, joiden tunnettavuus tägipilvessä oli vastaikään alkanut nousta (monissa tägipilvissä tunnetuimpien tägien tunnettavuus oli edelleen nousussa). Edellä mainitut ainoat nousevat tägit viittasivat enimmäkseen vanhoihin teknologioihin (`jquery`, `javascript` ja `video`), joskin näiden teknologioiden käytössä Silverlight-alustalla (tägipilvi oli rakennettu tägin `silverlight` ympärille) on saattanut tapahtua edistystä. Tässäkin tapauksessa löytyi kuitenkin yksi tägi, joka kuvaa suhteellisen uutta teknologiaa: `html5`.

Edellä mainitut tavoitteet huomioonottaen tuloksia ei voida pitää merkittävän hyödyllisinä. Mutta jos tutkielman taustalla olevat spesifit tavoitteet jätetään hetkeksi sivuun ja keskitytään tulosten yleishyödyllisyyteen, voidaan SMM-mallin mukaisen ryvästyksen tuloksissa nähdä runsaasti potentiaalia. Tulokset sisälsivät runsaasti synonyymeja, joiden automaattista tunnistamista on helppo hyödyntää esimerkiksi Deliciouksen kaltaisissa palveluissa, kun suoritetaan johonkin tiettyyn aiheeseen liittyviä hakuja. Tällöin synonyymien tuntemusta voidaan hyödyntää antamalla käyttäjälle tuloksia hänen antamansa hakusanan lisäksi myös hakusanan synonyymien antamista tuloksista. Lisäksi tägien suosittelualgoritmi pystyisi hyödyntämään ryhmiin 3, 3.5 ja 4 ryhmittyviä tägejä.

Ongelmana kuitenkin on, että algoritmin nykyisessä muodossa eri ryhmiin kuuluvien tägien tunnistaminen algoritmin tuloksista ei ole triviaali tehtävä ja jouduttiin tämänkin tutkielman puitteissa tekemään ihmisvoimin. Kuvasarjasta 6.8 voidaan johtaa ominaisuuksia kullekin ryhmälle, mutta ryhmien tägijoukkojen sijoittuminen päällekkäin tarkastellussa koordinaatistossa tekee ryhmien tarkasta erottelusta annetun datan perusteella mahdotonta.



Kuva 6.8: Eri tägiryhmien lukumäärien ja keskimääräisten $c_{t|t'}$ arvojen suhteet

7 Yhteenveto

Tutkielmassa perehdyttiin yksinkertaiseen, mutta yhteisöpalveluissa ja muissa Web 2.0 -mallin mukaisissa sivustoissa yleisesti käytössä olevaan metadatan muotoon, tägeihin. Tutkielma kartoitti niin tägien tyypitystä, teoreettista mallintamista kuin käyttöä tiedonlouhintaprosessin kohteena. Lisäksi tutustuttiin tarkemmin useisiin menetelmiin, joiden avulla tägidatasta voidaan louhia informaatiota. Tällä kaikella luotiin pohjaa tutkielman empiiriselle osuudelle, jossa rakennettiin edellä mainittuja louhintamenetelmiä käyttävä konstruktio, joka analysoi Delicious-palvelusta noin seitsemän kuukauden aikana kerättyä dataa. Konstruktion tavoitteena oli etsiä kerätystä datasta ensinnäkin tägien välisiä ontologisia yhteyksiä ja toisaalta löytää datasta uusia tekniikoita kuvaavia tägejä, joiden suosio on vastikään alkanut kasvaa. Näistä tuloksista toivottiin olevan apua ICT-alan yrityksille suunnattujen koulutustapahtumien suunnittelussa.

Konstruktion tuottamista tuloksista voidaan todeta, että SMM-mallin mukainen ryvästys on validi tapa löytää tiettyyn avainsanaan liittyviä tägejä. Valitettavasti perimmäinen tavoite ei toteutunut ja analysoiduista tuloksista löytyi enemmän synonyymeja kuin koulutuksen kannalta oleellisia avainsanoja. Myöskään tägipilvien historian seurannassa ei saatu haluttuja tuloksia. Useimmat tarkastellut tägipilvet olivat lähes täysin vakaita tai sisälsivät vain sosiaalisia järistyksiä. Näihin tuloksiin voi toki vaikuttaa seuranta-ajan lyhyys, käytetyn datan määrä ja rajoittunut keräysmenetelmä sekä seurattujen tägipilvien suhteellisen pieni määrä. Lisäksi ryvästysalgoritmin tuloksille on löydettävissä käyttötarkoituksia tutkielman omien tavoitteiden ulkopuolelta. Tosin synonyymisten ja relevanttien tägien joukkojen tarkka tunnistus koko tulosjoukosta on ongelma, joka tulee ratkaista, ennen kuin tutkielmassa esiteltyjä menetelmiä voidaan tehokkaasti käyttää näiden löytämiseen. Mahdollisesti rajaamalla tulosjoukkoa (tässä tutkielmassa käytetyn) $c_{t|t'}$ arvon lisäksi myös lukumäärän perusteella.

Ryvästyksen tulosten heikkous voi osin selittyä tägien kirjaamisen takana olevilla motiiveilla. Joshua Porter vertaa blogikirjoituksessaan *The Del.icio.us Lesson* [Por] Delicous-palvelun[1] tägejä muiden metadatamallien tägeihin. Hän huomauttaa, että

[1]Blogin kirjoiuksen aikaan Delicious-palvelun nimi oli vielä Del.icio.us. Käytän tässä palvelusta

toisin kuin jossain muunlaisessa metadatassa, jossa tägäyksen motiivina on toimia resurssin löytämisen ja ymmärtämisen apuna muille, Delicious-palvelussa tägäyksen motiivina on toimia resurssin (uudelleen)löytämisen apuna käyttäjälle itselleen.

Tutkielman loppuvaiheilla, datan keräämisen loputtua, Delicious-palvelu vaihtoi omistajaa. Omistajanvaihdoksen yhteydessä palvelua uudistettiin ja siihen lisättiin kirjanmerkkien yhteen, julkiseen listaan kokoamisen mahdollistava "pino" (engl. *stack*) -ominaisuus. Kirjanmerkkien tägääminen on edelleen palvelussa mahdollista, mutta käyttöliittymän painotus on suuntautunut huomattavasti enemmän tämän uuden ominaisuuden suuntaan. Tämän muutoksen vaikutus sivustolla tapahtuvaan tägäyskäyttäytymiseen voi mahdollisesti vaikeuttaa tutkielmassa tehtyjen testien uusimista tai Delicious-palvelun käyttämistä tägitiedonlouhintaan käytettävän datan lähteenä.

nykyisin käytettävää nimeä Delicious.

8 Lähteet

[Ack89] Russel L. Ackoff. From data to wisdom. *Journal of Applied Systems Analysis*, 16:3–9, 1989.

[ARML+01] Kal Ahmed, Daniel Rivers-Moore, Joshua Lubell, Andrew Watt, Mark Birbeck, Jay Cousins, Rob Worden, Miloslav Nic, Danny Ayers, and Ann Wrightson. *Professional XML meta data*. WroxPress, 2001.

[BKS06] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, May 2006.

[BYRN99] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[CCS08] Luigi Di Caro, K. Selçuk Candan, and Maria Luisa Sapino. Using tagflake for condensing navigable tag hierarchies from tag clouds. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1069–1072. ACM, 2008.

[DiN99] Darcy DiNucci. Fragmented future. *Print*, 53, 1999.

[DLZ+10] Ali Daud, Juanzi Li, Lizhu Zhou, Lei Zhang, Ying Ding, and Faqir Muhammad. Modeling ontology of folksonomy with latent semantics of tags. In Jimmy Xiangji Huang, Irwin King, Vijay V. Raghavan, and Stefan Rueger, editors, *Web Intelligence*, pages 516–523. IEEE, 2010.

[GLYH10] Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *SIGKDD Explorations*, 12(1):58–72, 2010.

[Gru] Thomas R. Gruber. Ontology of folksonomy. URL: `http://tomgruber.org/writing/ontology-of-folksonomy.htm`, viitattu 13.12.2011.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Gru09] Thomas R. Gruber. *Encyclopedia of Database Systems*, chapter Ontology. Springer-Verlag, 2009.

[Hay04] David Haynes. *Metadata for Information Management and Retrieval*. Library Assn Pub Ltd, 2004.

[Hea99] Marti A. Hearst. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10, Morristown, NJ, USA, 1999. Association for Computational Linguistics.

[HHV02] Eero Hyvönen, Petteri Harjula, and Kim Viljanen. Representing metadata about web resources. In Eero Hyvönen, editor, *Semantic Web Kick-Off in Finland - Vision, Technologies, Research, and Applications*, number 2002-001 in HIIT Publications, 2002.

[Hil] Diane Hillmann. Using dublin core. URL: `http://dublincore.org/documents/usageguide/`, viitattu 10.12.2011.

[HMS01] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, Cambridge, MA, 2001.

[HP98] Thomas Hofmann and Jan Puzicha. Statistical models for co-occurrence data. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, February 1998.

[HRGM08] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[HRS06] Harry Halpin, Valentin Robu, and Hana Shepard. The dynamics and semantics of collaborative tagging. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW'06)*, 2006.

[HRS07] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th*

international conference on World Wide Web, pages 211–220, New York, NY, USA, 2007. ACM.

[KÖ9] Christian Körner. Understanding the motivation behind tagging. ACM Student Research Competition - Hypertext 2009, July 2009.

[KMB03] Jan H. Kroeze, Machdel C. Matthee, and Theo J. D. Bothma. Differentiating data- and text-mining terminology. In Jarr Eloff, Andries Engelbrecht, Paula Kotzä, and Mariki Eloff, editors, *Proceedings of the 2003 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement Through Technology*, pages 93–101. South African Institute for Computer Scientists and Information Technologists, 2003.

[Kos07] Riikka Koskinen. Folksonomioiden semantiikka. Luk-tutkielma, Helsingin yliopisto, 2007.

[Mat04] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata, December 2004.

[McG03] Deborah L. McGuinness. Ontologies come of age. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.

[Mik07] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, March 2007.

[MM] Frank Manola and Frank Miller. Rdf primer. URL: `http://www.w3.org/TR/rdf-primer/`, viitattu 14.12.2011.

[MNBD06] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM.

[Nok] Nokia Oyj. Nokia esittelee strategiansa edistymistä. URL: `http://web.nokia.fi/nokia/lehdisto/tiedotteet/arkisto/arkisto-show?newsid=1524777`, viitattu 13.12.2011.

[Por] Joshua Porter. The del.icio.us lesson. URL: http://bokardo.com/archives/the-delicious-lesson/, viitattu 14.12.2011.

[Row07] Jennifer Rowley. The wisdom hierarchy: representations of the DIKW hierarchy. *J. Information Science*, 33(2):163–180, 2007.

[Rus06] Terrell Russell. Cloudalicious: Folksonomy over time. In *JCDL 06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 364–364, New York, NY, USA, 2006. ACM Press.

[Sam10] Nataliia Samoilenko. Ict anticipation model - report on the development. Technical report, University of Jyväskylä (PROFIT-project), 2010.

[SCA+07] Martin Szomszor, Ciro Cattuto, Harith Alani, Kieron O'Hara, Andrea Baldassarri, Vittorio Loreto, and Vito D.P. Servedio. Folksonomies, the semantic web, and movie recommendation. In *4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0*, 2007.

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007.

[SD09] Aixin Sun and Anwitaman Datta. On stability, clarity, and co-occurrence of self-tagging. In Ricardo A. Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu, editors, *WSDM (Late Breaking-Results)*. ACM, 2009.

[SdFa] Pietro Speroni di Fenizio. Observing society through tags: Using tags to help society. URL: http://videolectures.net/cvss08_fenizio_ostt/, viitattu 13.12.2011.

[SdFb] Pietro Speroni di Fenizio. On tag clouds, metric, tag sets and power laws. URL: http://blog.pietrosperoni.it/2005/05/25/tag-clouds-metric/, viitattu 13.12.2011.

[SdFc] Pietro Speroni di Fenizio. Tagclouds and cultural changes. URL: http://blog.pietrosperoni.it/2005/05/28/tagclouds-and-cultural-changes/, viitattu 13.12.2011.

[Shi05] Clay Shirky. Ontology is overrated: Categories, links, and tags, 2005.

[SW05] Kaikai Shen and Lide Wu. Folksonomy as a complex network, September 2005.

[TBF07] Brunello Tirozzi, Daniela Bianchi, and Enrico Ferraro. *Introduction To Computational Neurobiology and Clustering*. World Scientific, 2007.

[Tra09] Jennifer Trant. Studying social tagging and folksonomy: A review and framework. *Journal of Digital Information*, 10(1), 2009.

[Vos07] Jakob Voss. Tagging, folksonomy & co. - Renaissance of manual indexing? *10th international Symposium for Information Science*, 2007.

[VW] Thomas Vander Wal. Folksonomy. URL: `http://www.vanderwal.net/folksonomy.html`, viitattu 13.12.2011.

[Wei] David Weinberger. The problem with the data-information-knowledge-wisdom hierarchy - the conversation - harvard business review. URL: `http://blogs.hbr.org/cs/2010/02/data_is_to_info_as_info_is_not.html`, viitattu 13.12.2011.

[WSZ09] Robert Wetzker, Alan Said, and Carsten Zimmermann. Understanding the user: Personomy translation for tag recommendation. In Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors, *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497, pages 275–284, Bled, Slovenia, September 2009. CEUR Workshop Proceedings.

[WZY06] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM Press.

[ZWY06] Lei Zhang, Xian Wu, and Yong Yu. Emergent semantics from folksonomies: A quantitative study. *Journal on Data Semantics VI*, 2006.

[Äyr06] Sami Äyrämö. *Knowledge Mining Using Robust Clustering*, volume 63 of *Jyväskylä Studies in Computing*. University of Jyväsylä, 2006.

A Listaus tägeistä, jotka SMM-mallin toteutuksen mukaan liittyvät tägeihin `wp7` ja `maemo`

Tämä liite sisältää luvuissa 4.3 ja 5.5.2 kuvatun SMM-mallin mukaisen luokittelun tulokset tägeille `wp7` ja `maemo`. Luku $\overline{c_{t|t'}}$ kuvaa luvussa 5.5.2 kuvatun $c_{t|t'}$ luvun keskiarvoa 7 algoritmin kertauksen välillä, kun t on taulukon rivin tägi ja t' on joko `wp7` tai `maemo`. Lukumäärä on tägin esiintymismäärä lähdedatassa. Käytännössä kumpikin taulukko kuvaa tarkastellun tägin ympärillä havaittua konseptia (kts. luku 4.3) josta on karsittu pois mahdollisesti (algoritmin mukaan, tämän antamien tulosten yhtäläisyyksiä ja eroja vertaillen) epärelevantit tägit.

”Ryhmä” kuvaa ihmisarviota tägin relevanttiudesta. Ryhmä 4 sisältää joko `wp7`:n tai `maemo`:n synonyymeja. Ryhmä 3.5 sisältää joko Windows Phone 7:n ohjelmistokehitystä (tägiin `wp7` yhteydessä) tai Nokiaa ja tämän puhelinmalleja (tägin `maemo` yhteydessä) kuvaavia tägejä synonyymeineen. Ryhmä 3 kuvaa tarkasteltuun tägiin läheisesti kuuluvia tägejä. Ryhmä 2 kuvaa yleisistä sanoista johdettuja tägejä, jotka voidaan liittä tarkasteltavaan tägiin. Ryhmä 1 kuvaa lähteitä, esim. Internet sivuja tai ihmisiä. Ryhmä 0 kuvaa tägejä, joille ei löydetty suoraa yhteyttä tarkasteltavaan tägiin.

A.1 `wp7`

Tägi	Lukumäärä	$\overline{c_{t\|wp7}}$	Ryhmä
`davidfrincon`	2	0,99680	1
`silverlight,windowsmobile`	2	0,99680	3
`windows7mobile`	5	0,99680	4
`nokiawindows`	3	0,99680	3
`wp7tutorials`	4	0,99680	3.5
`windowsphone7developmenttutorials`	2	0,99680	3.5
`windowsphone7development`	2	0,99680	3-5
`mdcm400-02`	3	0,99680	0
`petzold`	5	0,99680	1

herewego	3	0,99680	0
windows-phone-7-dev	2	0,99680	3.5
wpdev	3	0,99680	3.5
phoneytools	2	0,99680	3
design_help	2	0,99680	2
windows+phone	4	0,99680	4
windows7-phone	2	0,99680	4
windowsphone7	797	0,99680	4
wp7-dev	3	0,99680	3.5
windows.phone.7	2	0,99680	4
virtualizing	3	0,99679	2
blogsforums	2	0,99679	1
wp7.1	5	0,99678	4
windowsphone	766	0,99516	4
wp7	5423	0,99363	4
wp7_1	3	0,98920	4
wm7	17	0,98907	4
winpho7	11	0,98377	4
wp7dev	388	0,97790	3.5
iphone-to-wp7	2	0,97754	3.5
windowsphones	9	0,97405	4
charla	7	0,97330	1
winphone	76	0,96408	4
wp7contrib	4	0,96108	3.5
windows7phone	15	0,95281	4
mango	426	0,94917	3
winphone7	32	0,94052	4
phone7	149	0,90760	4
coding.app	7	0,88968	3
windowphone7	12	0,88682	4
windows.phone	16	0,87228	4
windows_phone	46	0,86179	4
wp71	4	0,85582	4
adduplex	3	0,85576	3
watermarkedtextbox	2	0,85502	3

para_leer	3	0,85487	0
sketchfkow	3	0,85421	3
blogs_wp7	2	0,85414	1
windowsphone-zielgruppe	2	0,85394	0
shelltileschedule	6	0,85394	3
windowsphonemango	5	0,85394	3
readread	2	0,85041	0
wphone	4	0,82767	4
codesamples	4	0,82753	2
chevron	8	0,81677	3
win7phone	7	0,81456	4
wp7nl	3	0,80201	3
homer	4	0,79714	0
windowsmobile7	7	0,79429	4
gadget.hack	6	0,76494	3
parkingbot	2	0,76039	0
liveid	3	0,75912	2
touchstudio	10	0,75498	3
jumpstart	2	0,71617	2
"windows	6	0,71418	2
windowsphonegeek	2	0,71323	1
agfx	9	0,71313	3
silverlightzone	4	0,71308	1
angol	3	0,71283	0
preemptive	2	0,71227	2
livetile	4	0,71227	3
wpconnect	6	0,71127	3
executionmodel	2	0,71127	3
isolated	15	0,70511	2
windows_phone_dev	10	0,70191	3.5
nexttime	4	0,70103	0
to_be_read	2	0,70045	0
tombstoning	133	0,69881	3
panoramascrennshots	2	0,69571	3
imagine	3	0,69091	1

windows-phone-development	4	0,68861	3.5
textblock	6	0,68456	3
jesse	5	0,67426	1
wp7design	2	0,66157	3.5
bing.maps	7	0,66154	3
winmobile	12	0,65443	4
windows_phone7	14	0,64259	4
pushpins	2	0,64207	3
bing_maps	2	0,63900	3
tech.blog	2	0,63622	1
obfuscator	7	0,63600	2
winmo	16	0,63288	3
.favorite	5	0,62488	0
isolated-storage	5	0,62025	3
chevronwp	2	0,61940	3
windoz	5	0,61161	0
==	9	0,60839	0
wince	11	0,59800	3
samplecode	8	0,58813	2
windows-phone-7	221	0,58197	4
msdnblog	4	0,58077	1
netcf	3	0,57738	3

A.2 maemo

Tägi	Lukumäärä	$\overline{t_{t\|maemo}}$	Ryhmä
pluthon	2	0,99681	0
nokia770	46	0,99681	3.5
flasher	12	0,99681	3
wiki-page	2	0,99681	1
masterproj	2	0,99681	0
770	26	0,99681	3.5
nokia_n900	10	0,99618	3.5
informática_móvil	2	0,99585	0
maemo5	121	0,99402	4

maemo	1583	0,99369	4
n900	1161	0,96949	3.5
n8x0	8	0,95675	3.5
nokian900	20	0,95496	3.5
hildon	116	0,95356	3
maemon	3	0,91182	4
3c	10	0,91068	3.5
aplicacionesmaemo	4	0,88825	3
arijaaksi	2	0,86342	1
jaaksi	2	0,86320	1
ari_jaaksi	2	0,85967	1
nokia900	3	0,85714	3.5
nokia.n900	2	0,85714	3.5
fmtx	2	0,85714	3
nokia-n800	2	0,85714	3.5
meamo	3	0,85714	4
easydebian	3	0,85712	3
maemo4	2	0,85529	4
maemoflashing	4	0,85407	3
n810	282	0,85396	3.5
n800	317	0,85396	3.5
n900os	3	0,85395	3.5
tweet_en	3	0,85395	1
nitdroid	8	0,85395	3
n770	10	0,85395	3.5
nokia810	4	0,85395	3.5
os2008	11	0,85395	3
maemoqt	2	0,85395	3
remapping	3	0,85321	2
internettablet	6	0,79630	2
scratchbox	12	0,75315	3
via:@grundsignal	2	0,71429	0
pymaemo	15	0,71428	3
nokia800	3	0,71388	3.5
wayfinder	2	0,71264	3.5

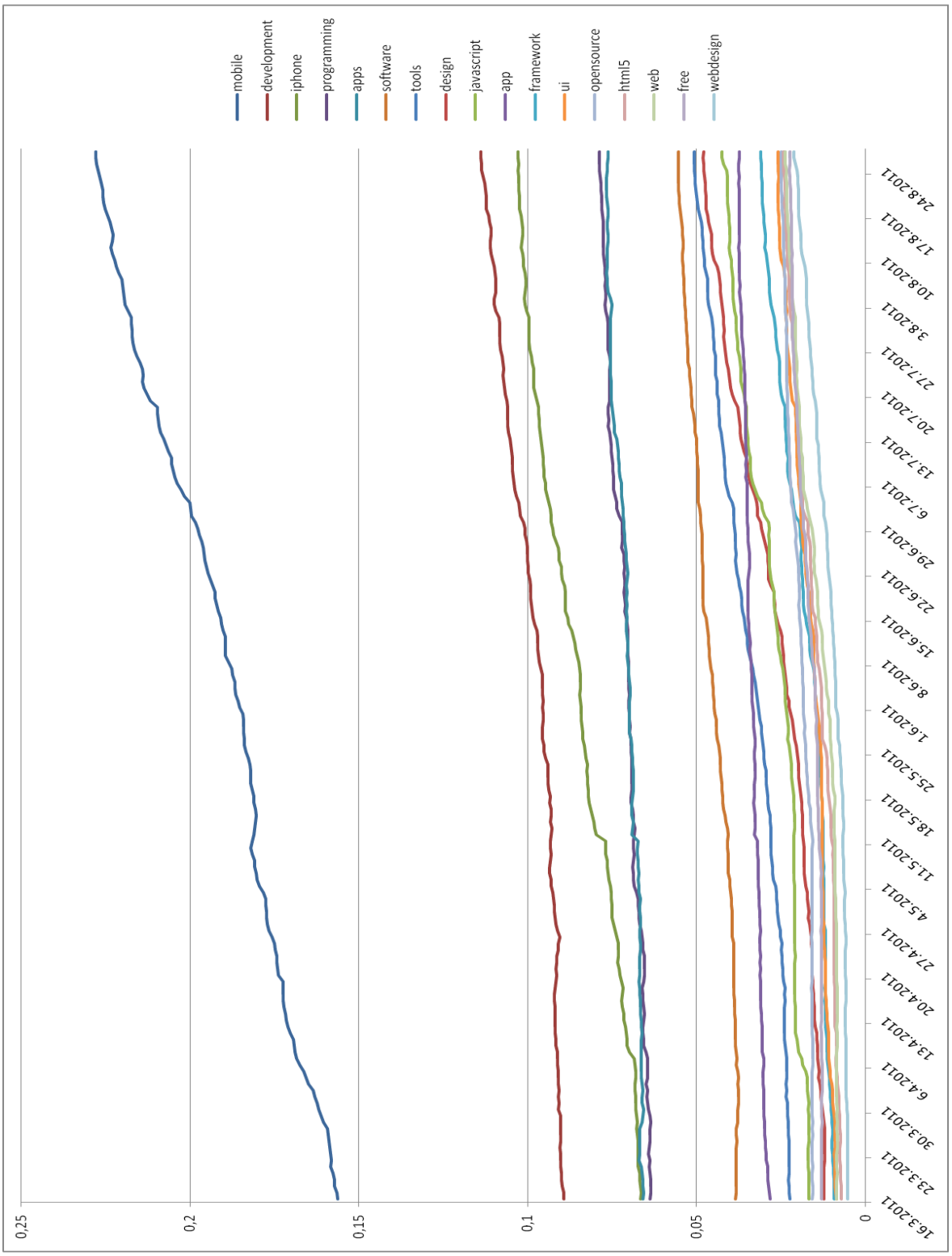
n9000	2	0,71112	3.5
emmc	2	0,71110	3
palmtop	2	0,70795	2
nokian810	4	0,68269	3.5
diablo	13	0,66561	3
mobile-developments	3	0,64788	2
extras	5	0,64101	2
carpc	2	0,63944	2
cs_lang:python	2	0,63502	2
nit	3	0,61432	3
meegoinstall	2	0,61232	3
developertools	8	0,60969	2
mfe	3	0,60040	3
carl	3	0,59074	1
blog-posting	3	0,58577	1
updating	7	0,58256	2
smscon	2	0,57249	3
microb	100	0,57144	3

B Tägihistorian tutkinnan tulokset

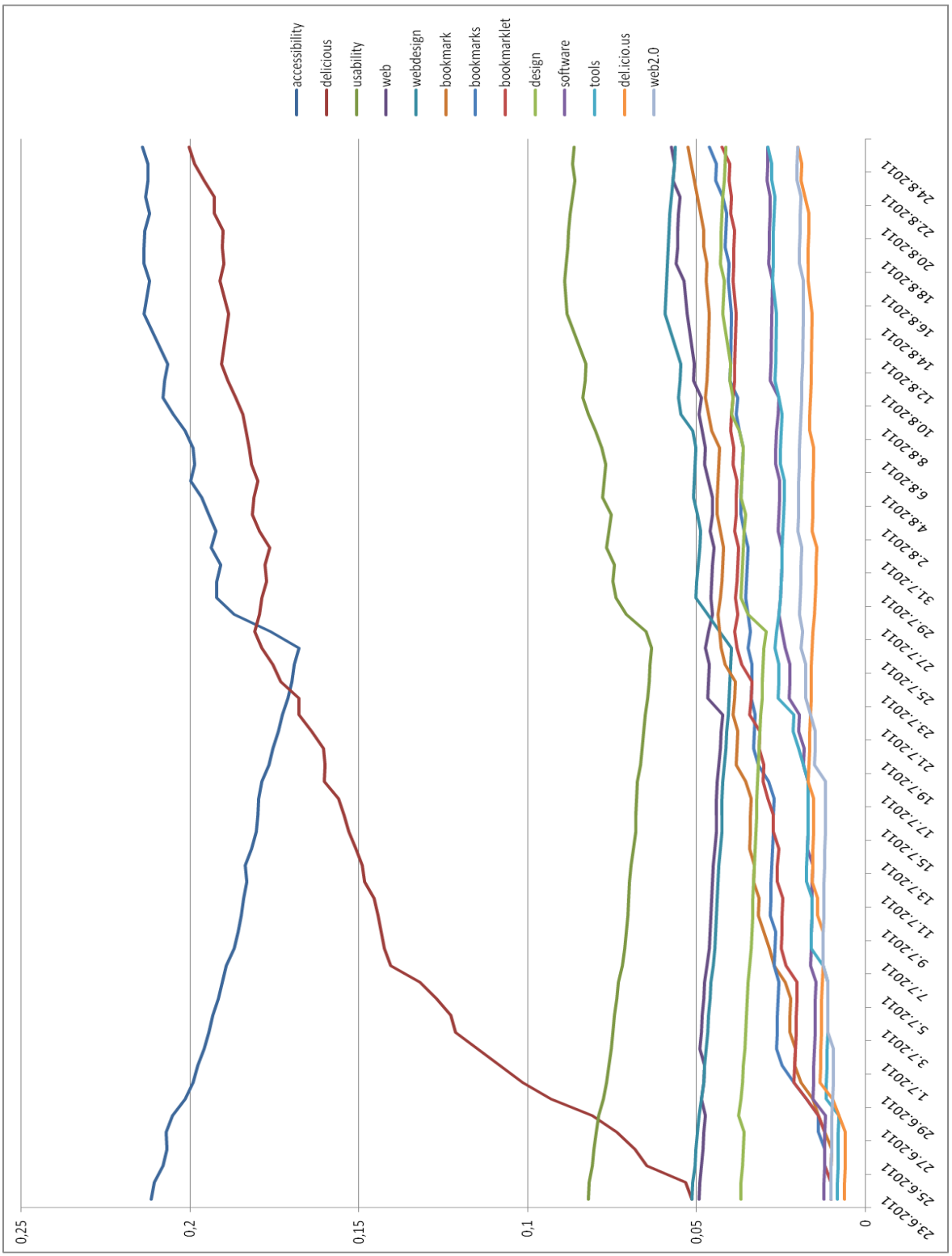
Tämä liite sisältää kaikki luvuissa 4.4 ja 5.5.3 kuvatun tägipilven historian seurantaalgoritmin, jotka tämän tutkielman yhteydessä suoritettiin, tulokset. Suoritusten pohjana olevaa dataa kuvataan tarkemmin luvussa 6.2. Tulokset ovat aakkosjärjestyksessä sen tägin mukaan, jonka pohjalta suoritukset tehtiin. Tuloksia tarkastellessa tulee ottaa huomioon luvussa 5.6.3 mainittu heikkous datanhakualgoritmissa.

Graafeissa vaaka-akseli kuvaa tarkasteltuja ajanhetkiä, esitettyinä päivämäärinä. Pystyakseli kuvaa kerrointa tägin tunnettuudesta datassa esiintyvien käyttäjien keskuudessa, jossa tägi lasketaan tunnetuksi, jos käyttäjä on käyttänyt sitä kerrankin ennen havaintopistettä. Esimerkiksi kerroin 0.3 tarkoittaa, että kaikista tähän ajanhetkeen mennessä esiintyvistä käyttäjistä 30% on käyttänyt kyseistä tägiä vähintään kerran. Graafien käyrät ovat kunkin kuvan selitteksteissä samassa järjestyksessä kuin itse käyrät ovat viimeisessä mittauspisteessä graafin oikeassa laidassa.

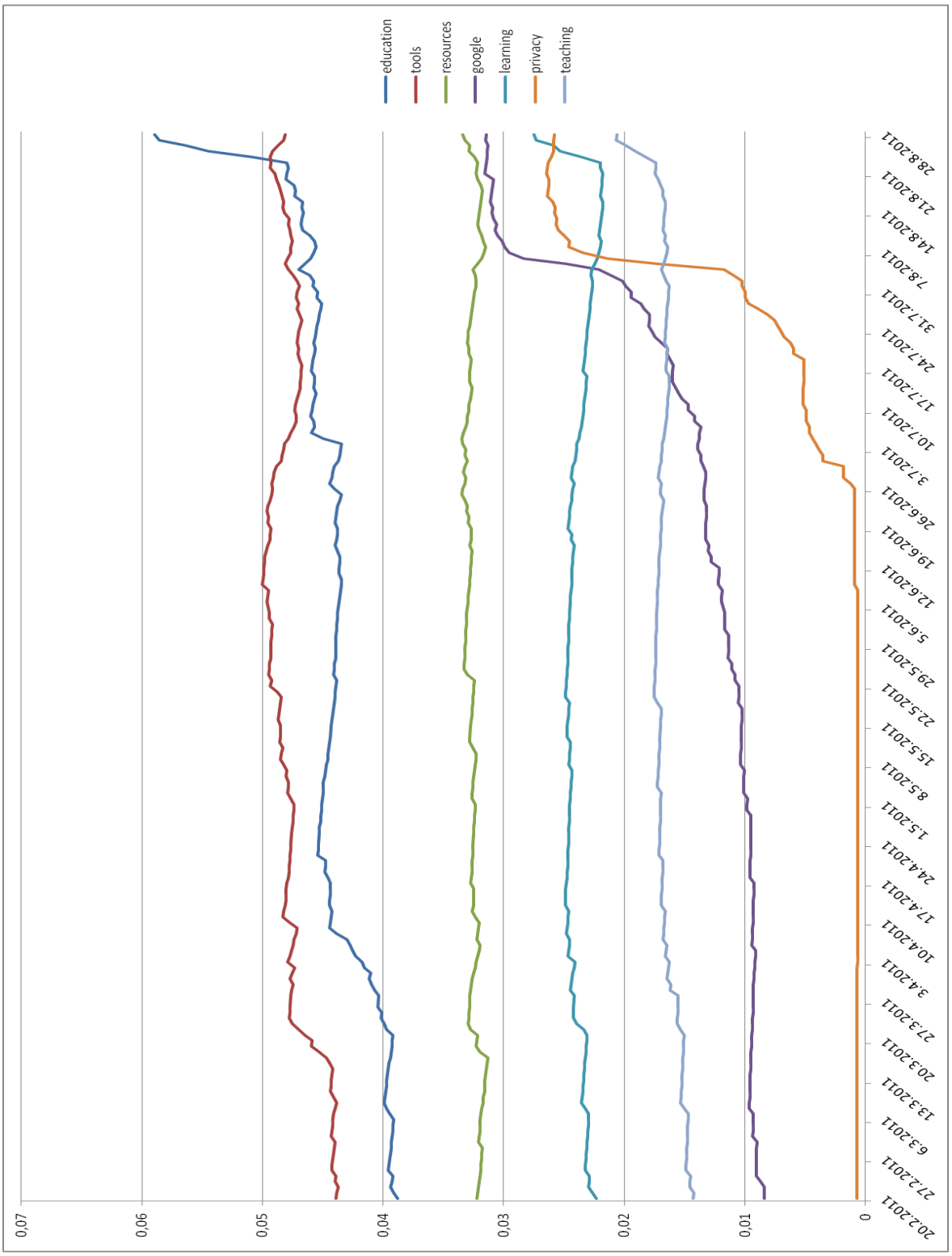
Tarkasteltaviksi tägeiksi valittiin vain tägejä, joiden tunnettuus on kasvanut tarkastelujakson alun ja lopun välillä. Lisäksi näistä tägeistä karsittiin ne, joiden tunnettuuskerroin oli alle 0.02. (Eli käytännössä alle 2% tarkastelluista käyttäjistä on käyttänyt tägiä.) Tarkastelujakson aluksi valittiin päivä, jona seuratun tägipilven stabilius (samankaltaisuus viimeisen havaintopisteen tägipilven kanssa) luvun 4.4.1 kaavan mukaan esitettynä on 0.15.



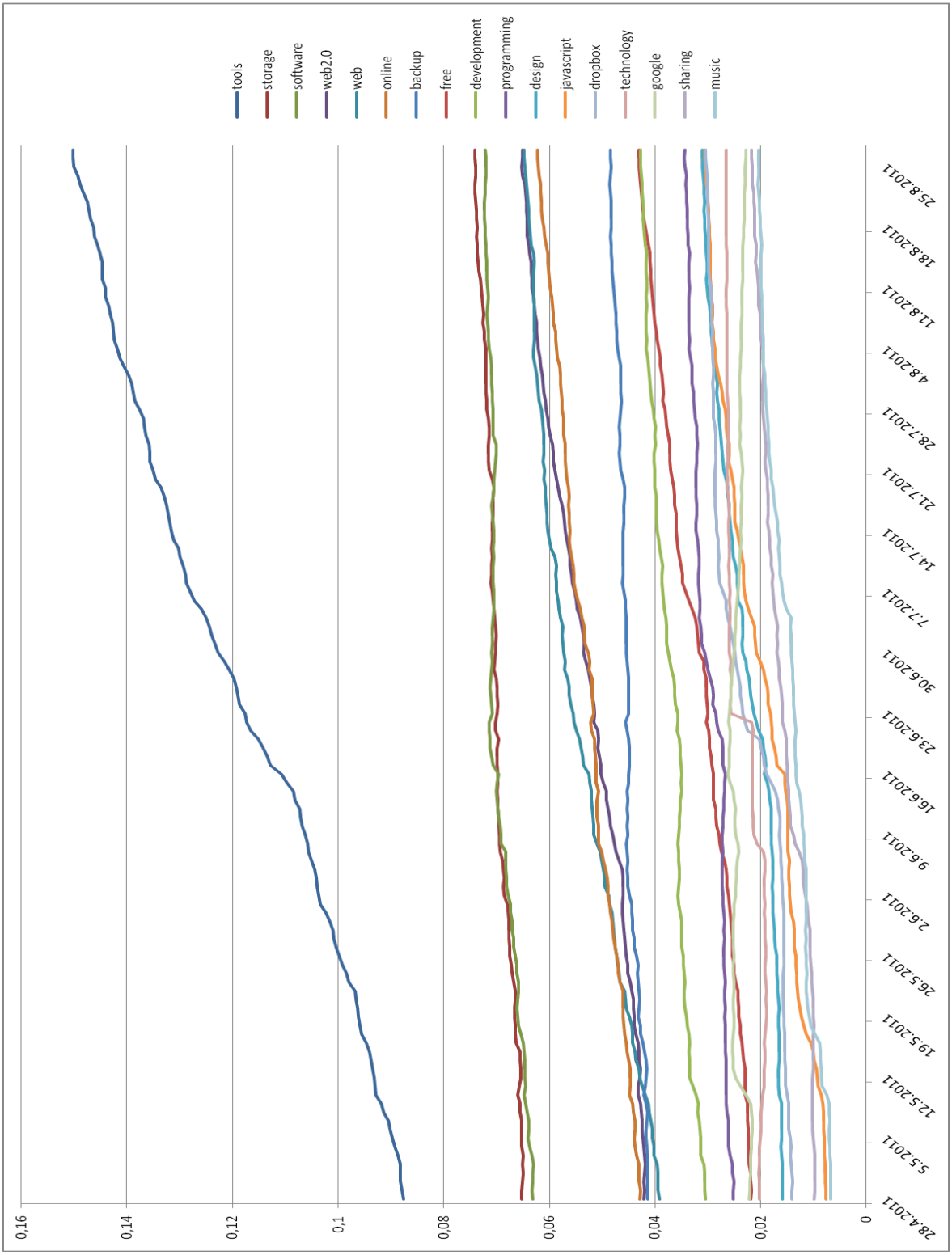
Kuva B.1: Tägihistoria tägiin `android` liittyen



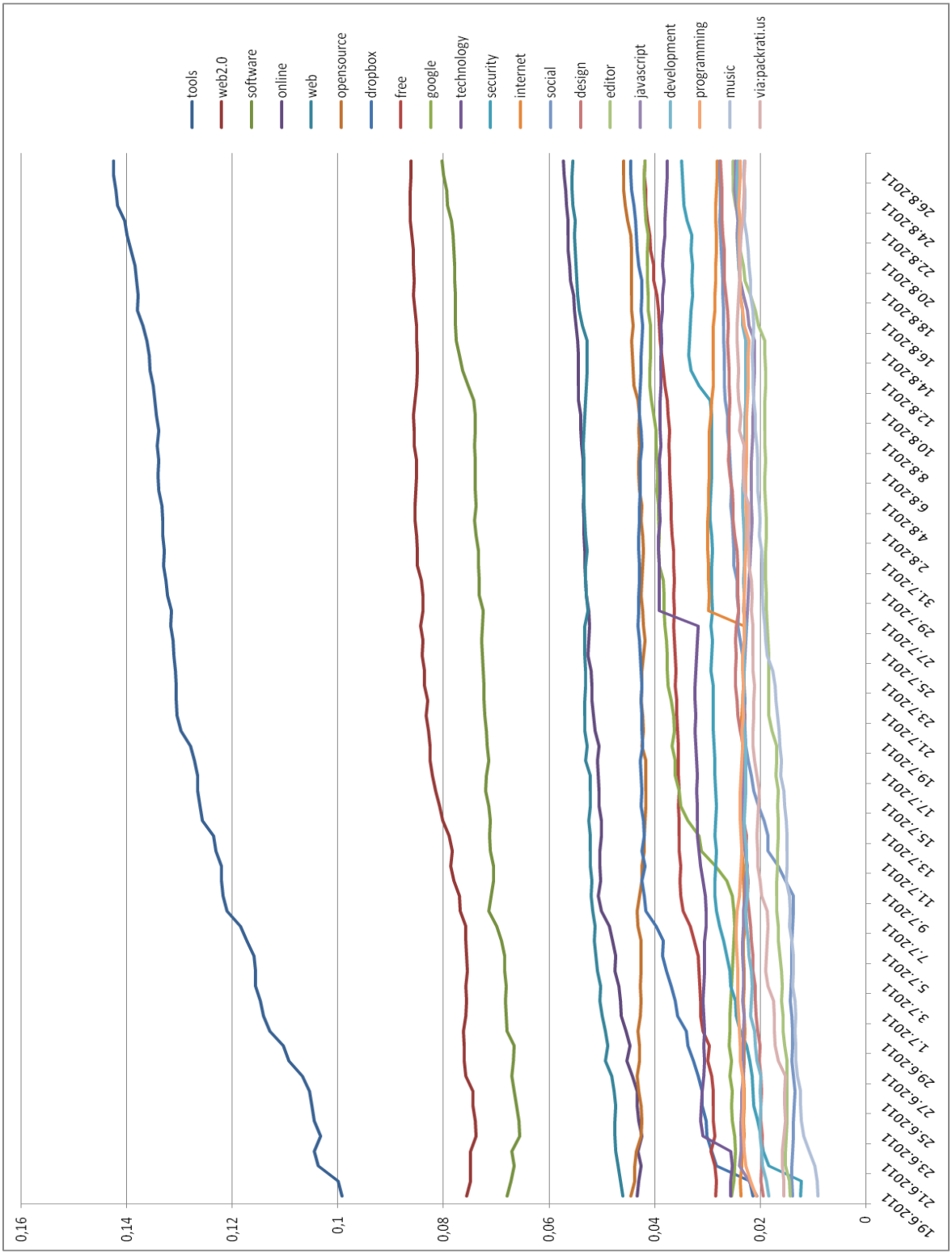
Kuva B.2: Tägihistoria tägiin `ax` liittyen



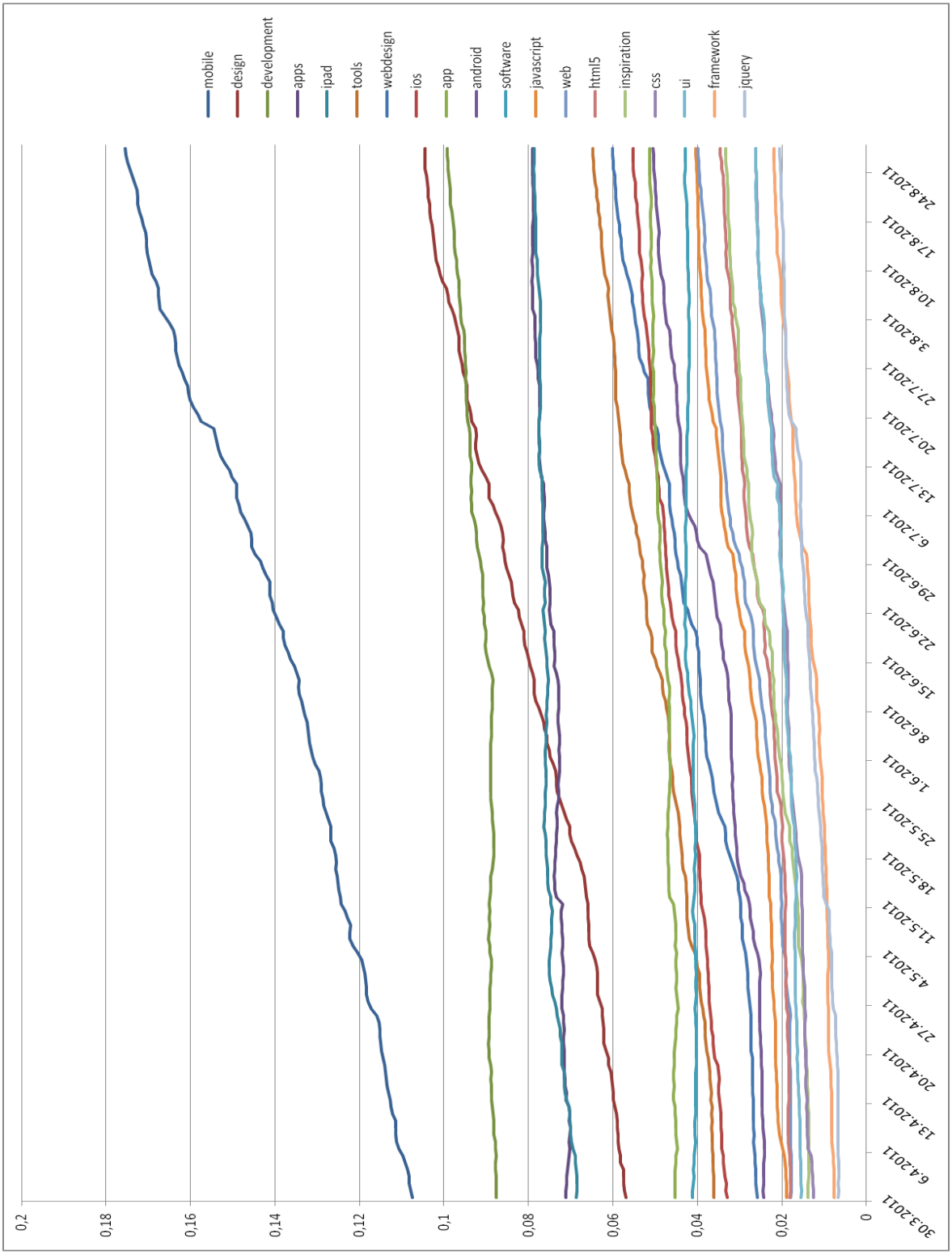
Kuva B.3: Tägihistoria tägiin `ce` liittyen



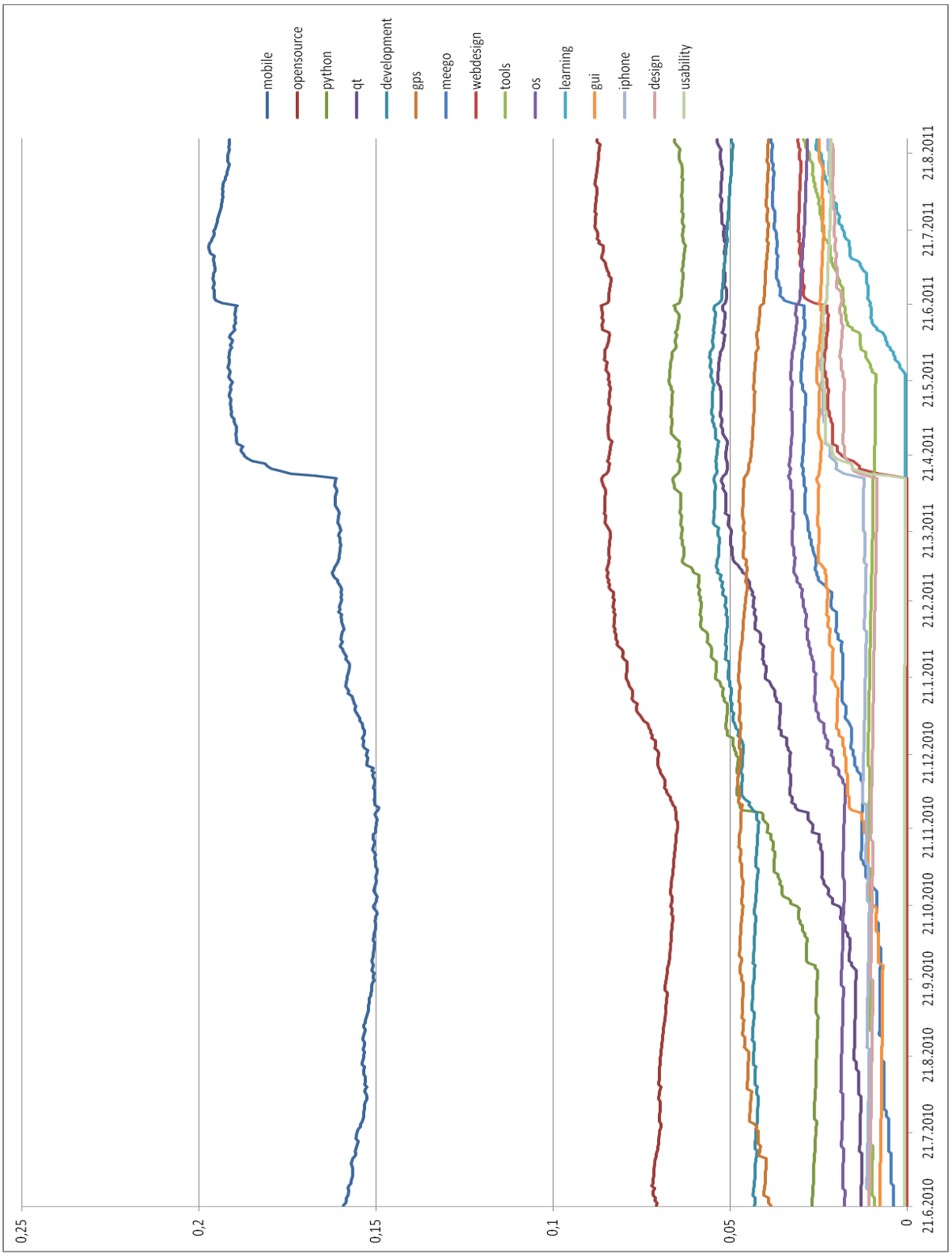
Kuva B.4: Tägihistoria tägiin `cloud` liittyen



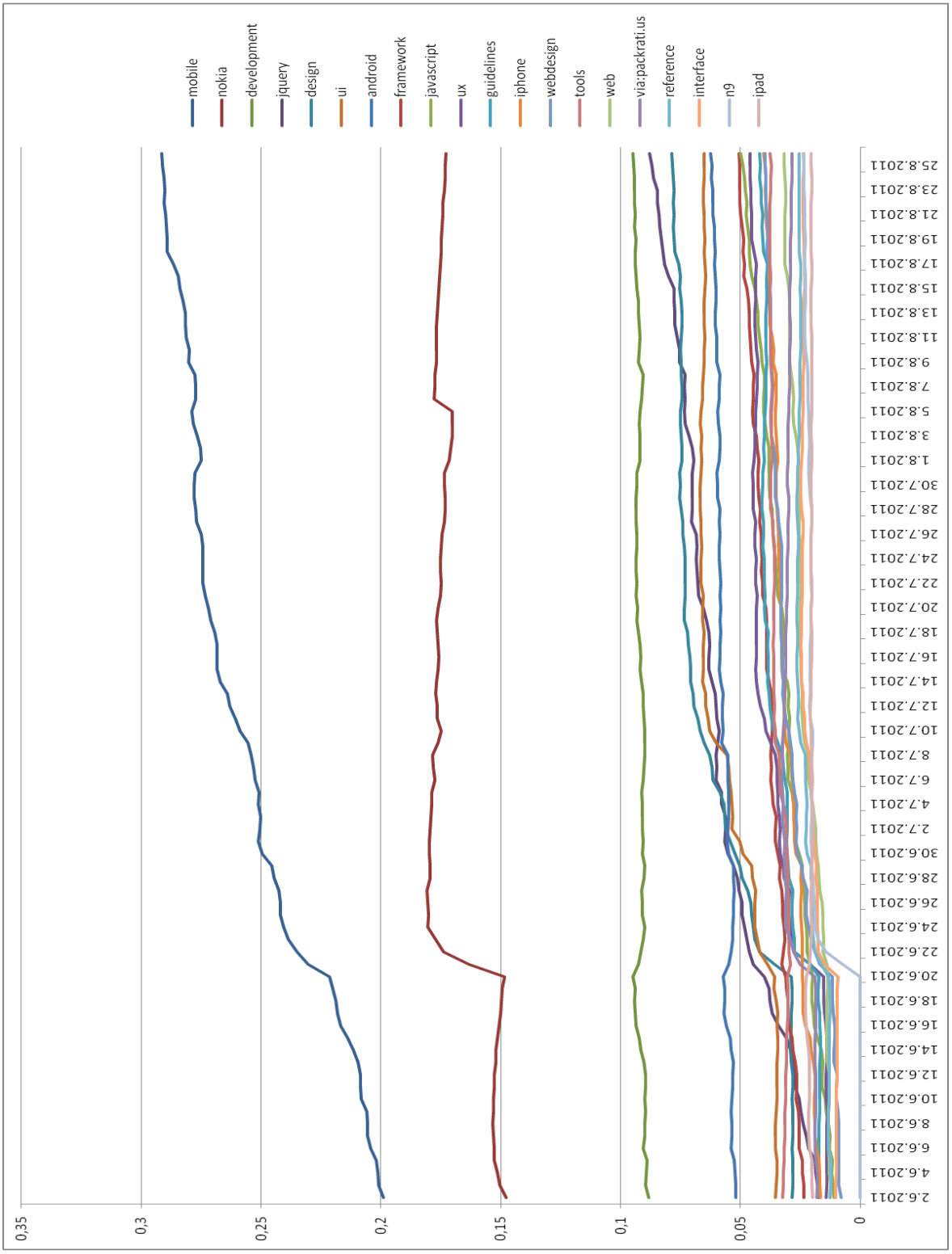
Kuva B.5: Tägihistoria tägien `cloud` ja `computing` yhdistelmään liittyen



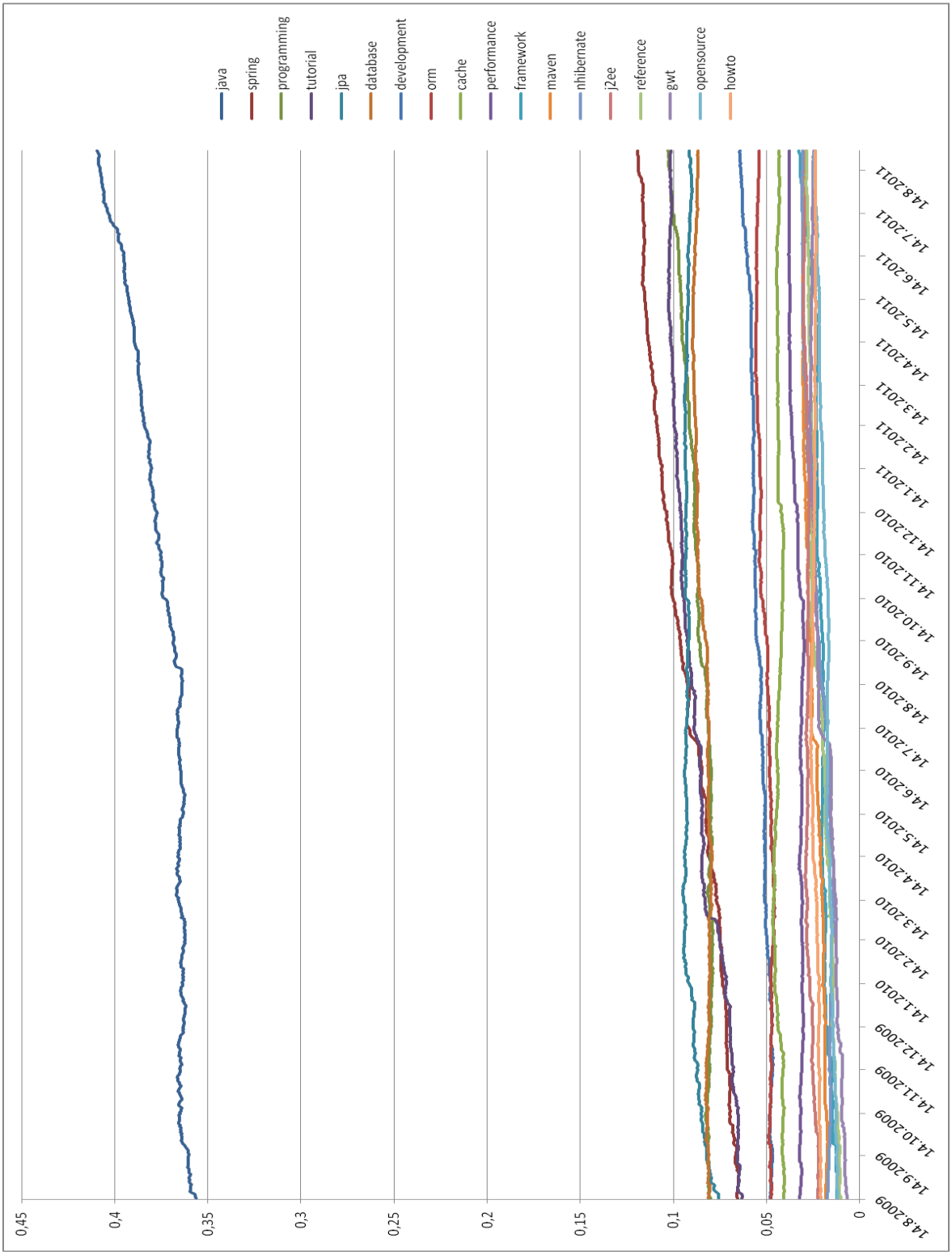
Kuva B.6: Tägihistoria tägiin `iphone` liittyen



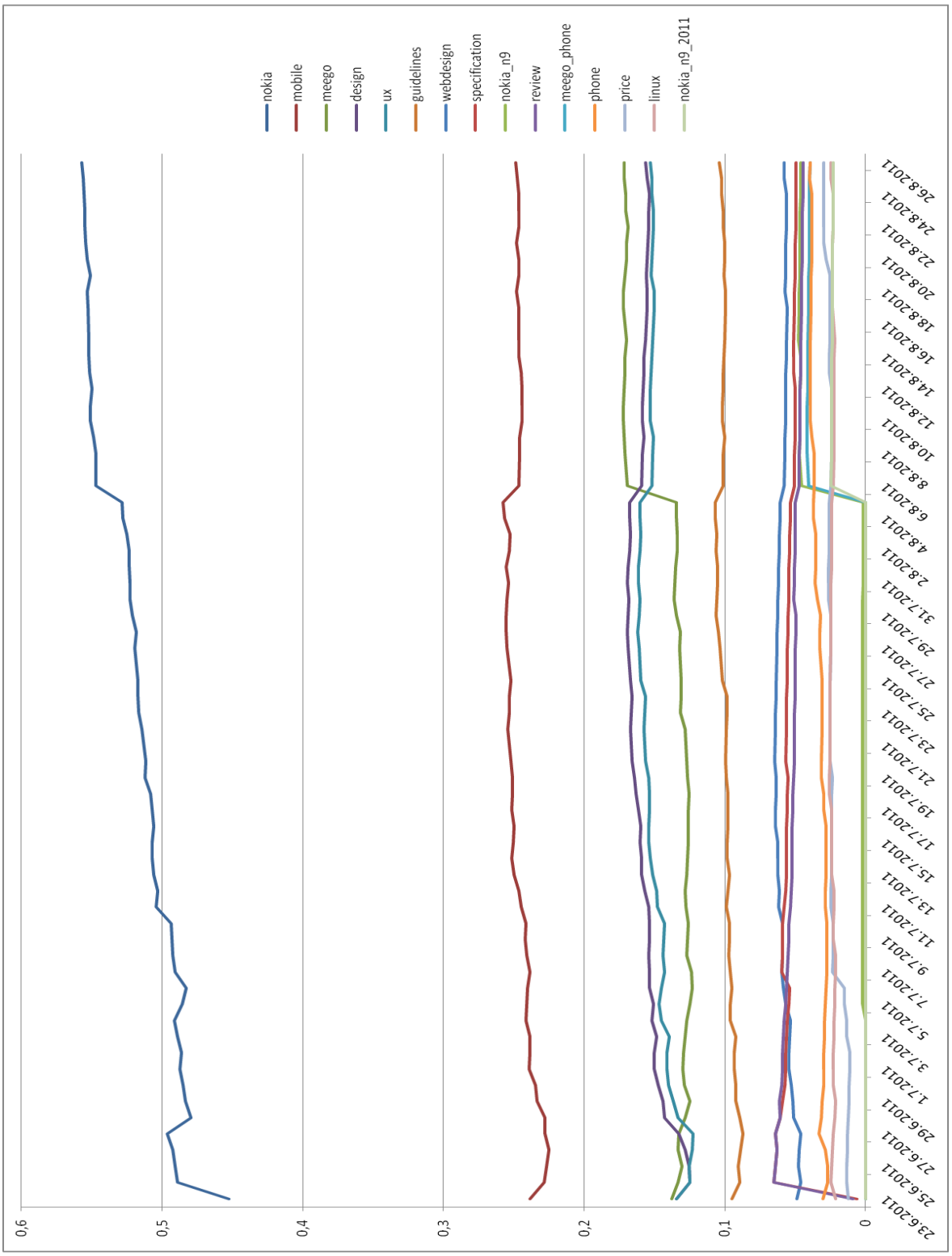
Kuva B.7: Tägihistoria tägiin `maemo` liittyen



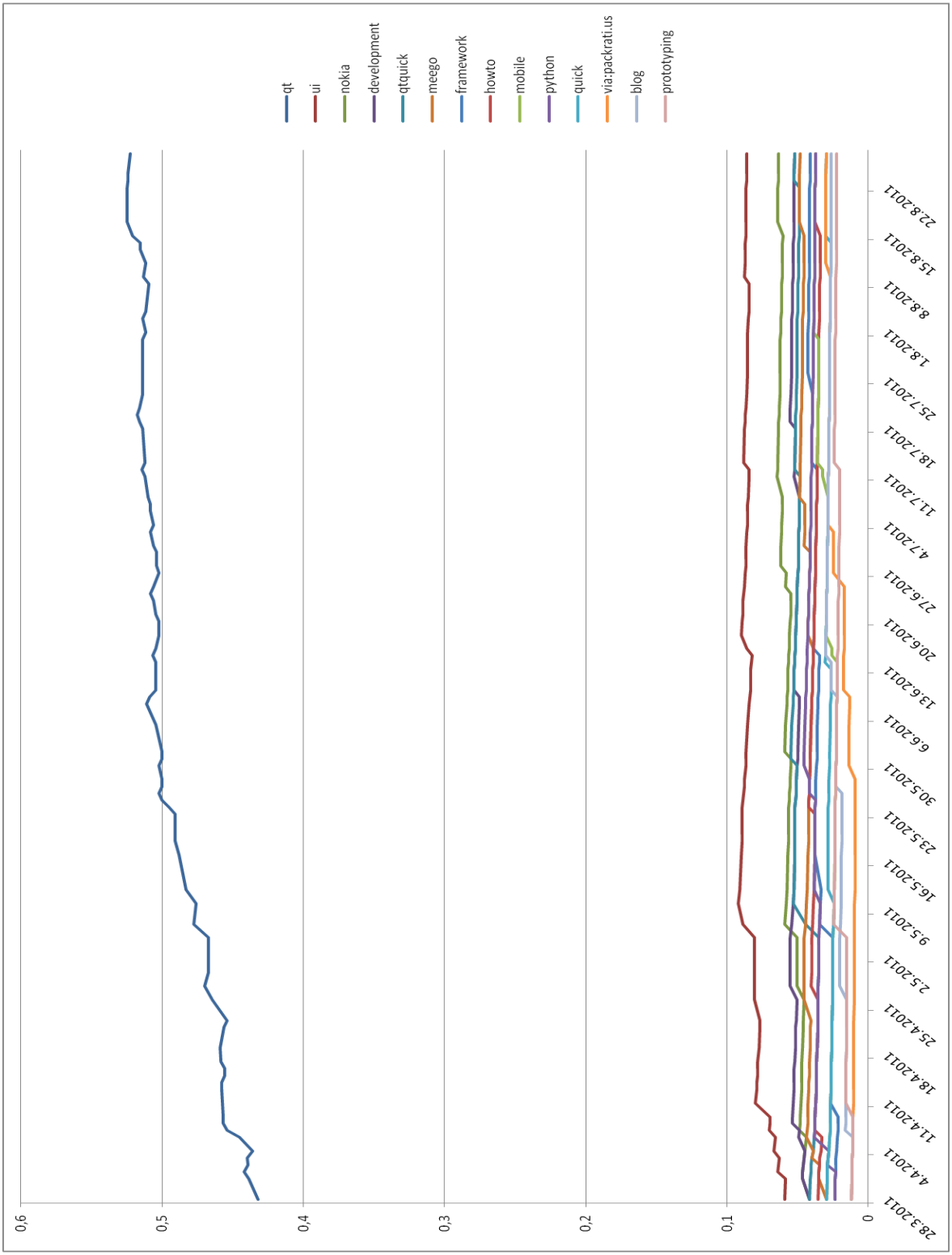
Kuva B.8: Tägihistoria tägiin `meego` liittyen



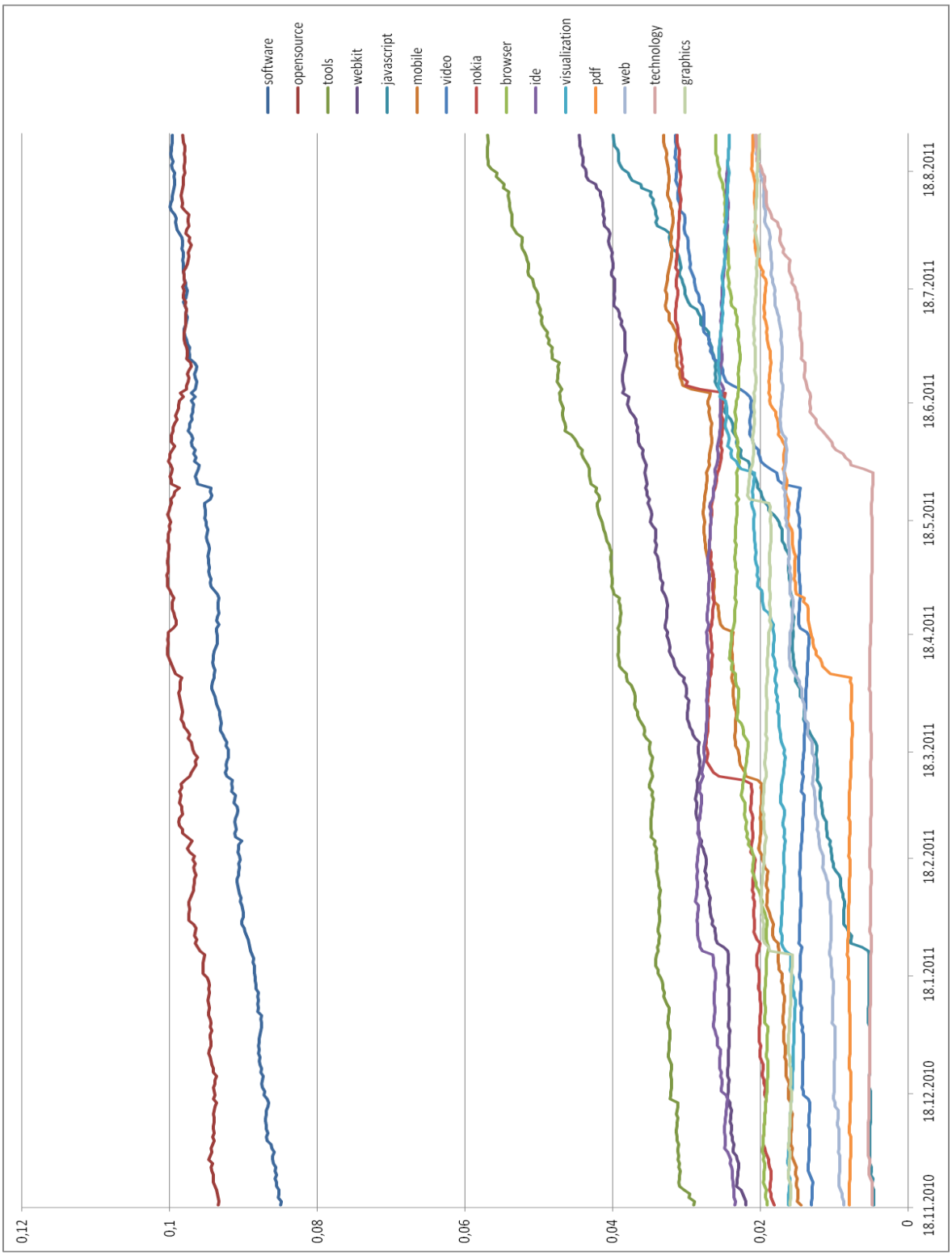
Kuva B.9: Tägihistoria tägiin `hibernate` liittyen



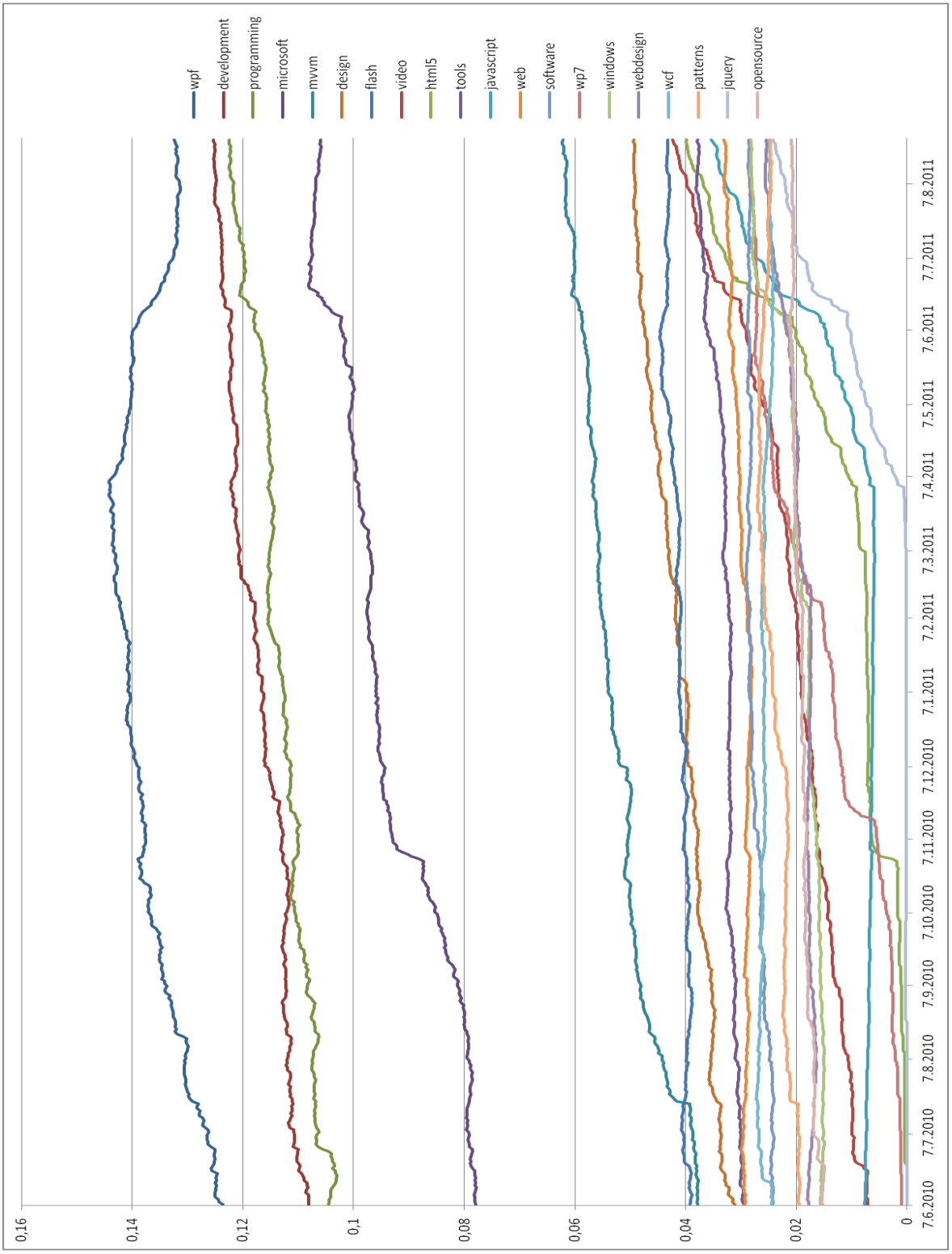
Kuva B.10: Tägihistoria tägiin `n9` liittyen



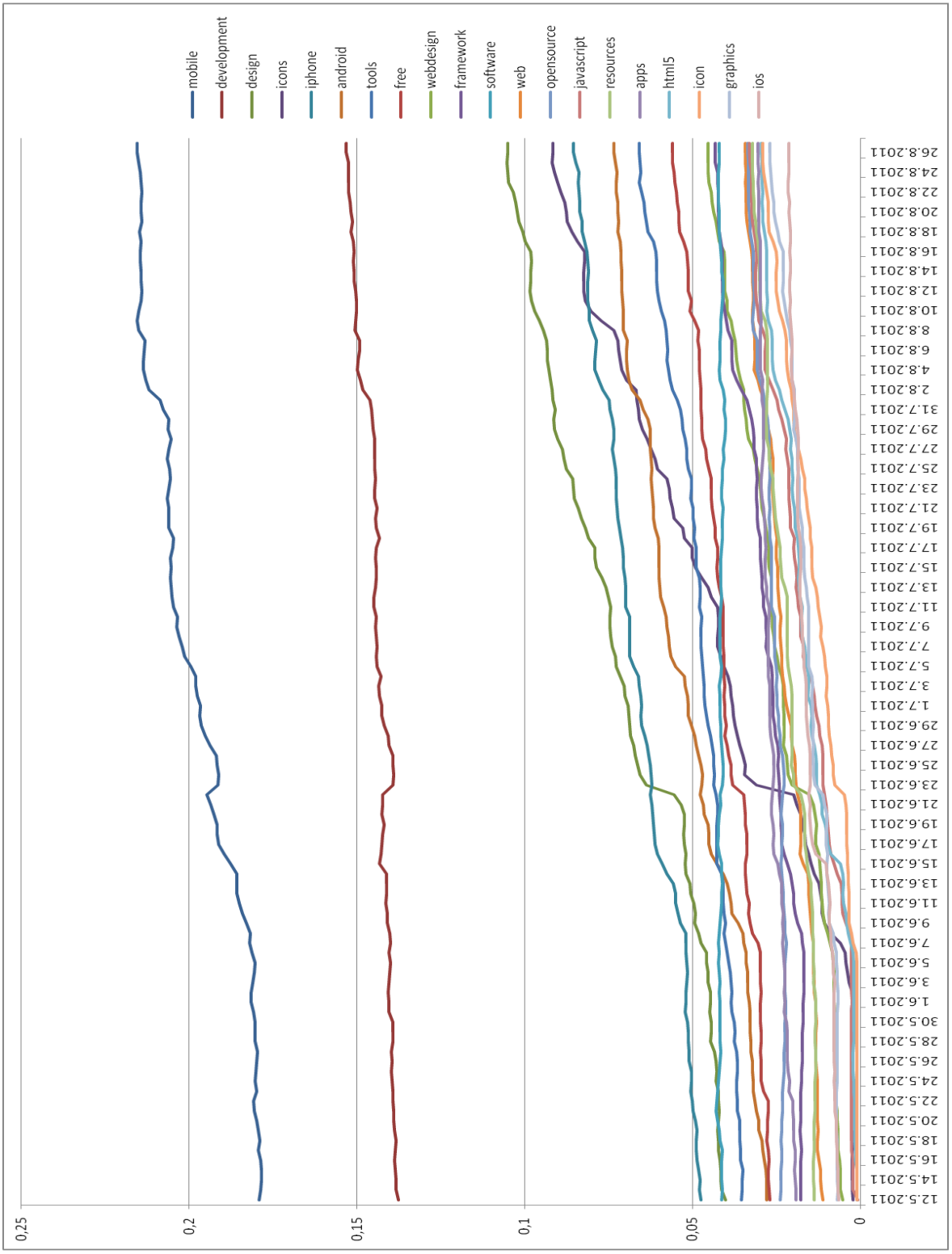
Kuva B.11: Tägihistoria tägiin `qml` liittyen



Kuva B.12: Tägihistoria tägiin `qt` liittyen



Kuva B.13: Tägihistoria tägiin `silverlight` liittyen



Kuva B.14: Tägihistoria tägiin `wp7` liittyen

C Todistus SMM-mallista

Oletetaan että $\langle R_{r\alpha} \rangle^{(t)} = x, x \in \mathbb{R}, x \neq 0$ eli nollasta eroava vakio kaikilla r ja α. Tästä seuraa että

$\hat{\pi}_\alpha^{(t)} = \frac{1}{L} \sum_{r=1}^{L} \langle R_{r\alpha} \rangle^{(t)} = \frac{Lx}{L} = x$ kaikilla α,

$\hat{p}_{i|\alpha}^{(t)} = \frac{1}{L\hat{\pi}_\alpha^{(t)}} \sum_{r:i(r)=i} \langle R_{r\alpha} \rangle^{(t)} = \frac{k_i x}{Lx} = \frac{k_i}{L}$ kaikilla α, jossa k_i on niiden havaintopisteiden määrä, joissa $r : i(r) = i$ on tosi, ja

$\hat{q}_{j|\alpha}^{(t)} = \frac{1}{L\hat{\pi}_\alpha^{(t)}} \sum_{r:j(r)=j} \langle R_{r\alpha} \rangle^{(t)} = \frac{k_j x}{Lx} = \frac{k_j}{L}$ kaikilla α, jossa k_j on niiden havaintopisteiden määrä, joissa $r : j(r) = j$ on tosi.

Näistä voidaan laskea $\langle R_{r\alpha} \rangle^{(t+1)} = \frac{\pi_\alpha^{(t)} \hat{p}_{i(r)|\alpha}^{(t)} \hat{q}_{j(r)|\alpha}^{(t)}}{\sum_{v=1}^{K} \pi_v^{(t)} \hat{p}_{i(r)|v}^{(t)} \hat{q}_{j(r)|v}^{(t)}} = \frac{x k_{i(r)} k_{j(r)} / L^2}{K x k_{i(r)} k_{j(r)} / L^2} = \frac{1}{K}$ kaikilla r ja α.

Tätä hyväksi käyttäen voidaan induktiivisesti todistaa, että alkuarvaus $\langle R_{r\alpha} \rangle^{(0)} = x$ kaikilla r ja α johtaa siihen, että yksikään $R_{r\alpha}$: muuttuja ei muutu ensinmäisen kierroksen jälkeen, eikä täten mikään muukaan mallin muuttujista.

Ensin todettakoon, että $\langle R_{r\alpha} \rangle^{(0)} = x$ kaikilla r ja α johtaa siihen, että $\langle R_{r\alpha} \rangle^{(1)} = \frac{1}{K}$ kaikilla r ja α.

Induktion perusaskeleena voidaan tämän jälkeen vastaavasti todeta että $\langle R_{r\alpha} \rangle^{(1)} = \frac{1}{K} = \langle R_{r\alpha} \rangle^{(2)}$ kaikilla r ja α.

Induktio-oletuksena $\langle R_{r\alpha} \rangle^{(k)} = \frac{1}{K}$ kaikilla r ja α.

Tällöin voidaan todettujen kaavojen avulla todistaa, että $\langle R_{r\alpha} \rangle^{(k+1)} = \frac{1}{K}$ kaikilla r ja α.

Näin ollaan todistettu, että $R_{r\alpha}$:n yksikään muuttuja ei muutu ensimmäisen kierroksen jälkeen.