Tampereen yliopisto

Daniel Saarimäki

TYÖNTEKIJÖIDEN OSAAMISEN ARVIOIMINEN SISÄISEN VIESTINNÄN PERUSTEELLA BAYES-VERKKOA KÄYTTÄMÄLLÄ

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Syyskuu 2020

TIIVISTELMÄ

Daniel Saarimäki: Työntekijöiden osaamisen arvioiminen sisäisen viestinnän perusteella Bayes-verkkoa käyttämällä
Diplomityö
Tampereen yliopisto
Teknis-luonnontieteellinen DI-tutkinto-ohjelma
Syyskuu 2020

Monilla moderneilla ja erityisesti pienempiä projekteja paljon tekevillä yrityksillä on jatkuva tarve sovittaa työntekijöiden osaamisalueet tarjolla oleviin projekteihin tehokkuuden maksimoimiseksi. Tämä vaatii kattavaa tietoa työntekijöiden hallitsemista taidoista. Yritykset voivat kerätä työntekijöiltä tietoa heidän osaamisestaan erilaisin menetelmin, mutta riskinä aina on, ettei kriittisellä hetkellä satu esimerkiksi olemaan saatavilla tietoa juuri käsillä olevaan projektiin suunnitelluista tekniikoista sekä työntekijöiden suhteesta näihin tekniikoihin.

Tässä diplomityössä tutkitaan voiko tarvittavaa tietoa työntekijöiden osaamisalueista johtaa yrityksen sisäisissä kommunikaatiokanavissa käydyistä sähköisistä keskusteluista. Lisäksi tutkitaan Bayes-verkon eli erilaisia tapahtumia ja niiden välisiä riippuvuussuhteita esittävän graafirakenteen soveltuvuutta kerätyn tietotaitoa kuvaavan tiedon säilömiseen sekä käsittelyyn. Bayes-verkot ovat saavuttaneet suosiota oppilaan taitoihin mukautuvissa opetussovelluksissa, joissa oppilaan hallitsemia ja hallitsemattomia asioita pyritään selvittämään erilaisten koekysymysten vastausten perusteella.

Vaikka Bayes-verkko todetaan työssä hyväksi tavaksi kuvata työntekijän osaamista, joudutaan referenssinä käytetyt opetuskäyttöön rakennetut mallit toteamaan soveltumattomiksi työssä todettuihin tarpeisiin. Tämä yhdessä tavallisesta poikkeavan tekstianalyysitarpeen sekä työn rajallisen skaalan kanssa johtaa siihen, ettei työssä sovelleta edistyneempiä luonnollisen kielen prosessoinnin menetelmiä tai oppivia algoritmeja, kuten esimerkiksi lauserakenneanalyysiä tai neuroverkkoja. Lopputuloksena on algoritmi, joka pystyy tuottamaan jonkin verran tietoa yrityksen käytössä olevasta kokonaistietomäärästä, mutta ei tuota riittävän luotettavaa tietoa yksittäisten työntekijöiden osaamisesta.

Luonnollisen kaoottinen keskusteludata todetaan tutkimuksen pohjalta huonoksi tiedon lähteeksi, ellei analyysissä käytetä edistyneempiä luonnollisen kielen analysointikeinoja yksittäisten viestien tarkoituksen tarkempaan selvittämiseen. Tekstin analysoinnin hyödyllisyyttä voitaisiin myös lisätä merkittävästi ottamalla mukaan muualta saatua ennakkotietoa jo algoritmin toiminnan aikana sekä parantamalla Bayes-verkon rakennetta. Tässä työssä käytettävä Bayes-verkon rakenne johdetaan toisen järjestelmän taitokategorioista, mikä osoittautuu epäoptimaaliseksi ratkaisuksi.

Avainsanat: Bayes-verkko, osaaminen, tekstianalyysi, tietotaito, Flowdock

ABSTRACT

Daniel Saarimäki: Estimating employee knowledge and skills based on internal communications using a Bayesian network
Master of Science Thesis
Tampere University
Science and Engineering, MSc
September 2020

Many modern enterprises, especially those that have a lot of smaller projects, have a continuous need to fit the expertise of their employees to the available projects in order to maximize productivity. This requires good knowledge of the skills possessed by the employees. Enterprises can collect information of the skills of their employees using various methods, but there's always the risk that at the critical moment there's no knowledge of the employee skills regarding the exact technology or technique required for the project.

In this masters' thesis I research if the required information regarding employee expertise could be inferred from the discussions that happen in the internal communication channels within an organisation. I also research if Bayesian networks, graph structures used for presenting various events and their probability relations, are suitable for containing and operating on the collected skill information. Bayesian networks have acquired popularity in adaptive educational applications where the application tries to infer the learning state of a student in various subjects based on a series of exam questions.

Even though Bayesian networks are noted to be a good way to represent the knowledge of an employee, the models developed for educational use were found to be ill-suited for the needs outlined in this research. This in combination with the unusual text analysis needs and the limited scale of the thesis means that advanced natural language processing methods and intelligent algorithms such as sentence structure processing and neural networks are not implemented. The result is an algorithm that can produce some amount of information regarding the knowledge available to an organisation, but fails to produce reliable knowledge regarding the skills of a single employee.

The chaotic and natural conversation data was found to be a poor source of information, unless the analysis uses more advanced natural language processing methods to infer the purpose of individual messages. The usefulness of text analysis could also be increased significantly by taking into account prior information received from other sources during the algorithm and by improving the structure of the Bayesian network. In this thesis the structure of the Bayesian network was based on the skill categories of another system, which turned out to be a suboptimal solution.

Keywords: Bayesian network, skill, text analysis, knowledge, Flowdock

ALKUSANAT

Diplomityö on aloitettu Futurice Oy:n sisäisen koneoppimisen tutkimusryhmän Exponentialin alaisuudessa. Kyseessä ei siis ole asiakasprojekti, vaan tuottoon pyrkimätön yrityksen sisäinen tutkimus. Olen ollut yrityksen osa-aikaisena työntekijänä sovelluskehittäjänä yhtäjaksoisesti vuodesta 2017. Työn aihealue ei suoraan vastaa työnkuvaani, vaan edustaa pyrkimystäni siirtää erikoistumistani perinteisestä sovelluskehityksestä koneoppimisen suuntaan.

Diplomityöhön liittyvien rajoitteiden vuoksi työ joutui hyvin kiusalliseen tilanteeseen, jossa luonnollisen kielen analyysiä ja isoa datamassaa käsittelevää analytiikkaa päädyttiin tekemään ilman kunnollisia luonnollisen kielen analyysin tai data-analytiikan metodeja. Työstä tuli lopulta harjoitus monimutkaisen ongelman ratkaisemisesta aivan liian yksinkertaisin keinoin ja lopputuloksena syntyneistä kaoottisista tuloksista kävi selväksi, että prosessi olisi pitänyt melkein jakaa kahteen eri diplomityöhön, joista toinen keskittyy lähteenä olleen tekstin analysointiin ja toinen tämän analyysin pohjalta rakennettaviin tietorakenteisiin sekä johtopäätöksiin. Tällöin algoritmien laadun ja tarkoituksellisuuden varmistamiseen oltaisiin voitu investoida enemmän resursseja.

Työn vastuuohjaajana toimi yliopistolehtori Henri Hansen ja Futuricen puolelta ohjaajana toimi tietotekniikan maisteri ja vanhempi konsultti Mikko Viikari. Suuri kiitos kummallekin, sekä Azeem Akhterille, joka auttoi keskusteludatan löytämisessä ja sen kanssa alkuun pääsemisessä. Iso kiitos myös Jussi Vaihialle omien data-arkistojensa raottamisesta, taitokyselyn vastauksensa työhön tarjonneille Futuricen työntekijöille ja koko Futuricen väelle loistavasta kokeilemiseen, yrittämiseen ja erehtymiseen kannustavasta työympäristöstä!

Tampereella, 7. syyskuuta 2020

Daniel Saarimäki

SISÄLLYSLUETTELO

KUVALUETTELO

TAULUKKOLUETTELO

LYHENTEET JA MERKINNÄT

Σ^*	kaikkien sellaisten merkkijonojen joukko, jotka saadaan muodostettua aakkostosta Σ
a_j	osaamiskategoriaa j edustava aihetermi
ASCII	amerikkalainen standardimerkistö informaation jakoon (eng. American Standard Code for Information Interchange)
B	verkon solmuja edustavien termien sekä tekstissä esiintyvien tokeneiden välistä suhdetta kuvaava matriisi
$\beta_{E,ij}$	solmun v_{ij} ja sen vanhemman $v_{E,ij}$ välisen suhteen voimakkuutta kuvaava painoarvo
e_{ij}	osaamisarvon t_{ij} epävarmuutta kuvaava epävarmuusarvo
$e_{0,ij}$	epävarmuusarvon e_{ij} laskemiseen käytettävä pohja-arvo
HTTP	hypertekstin siirtoprotokolla (eng. Hypertext Transfer Protocol)
ID	yksilöllinen identifioija (eng. Identifier)
ISO	Kansainvälinen standardointiorganisaatio
JSON	avoin tiedon välitykseen käytetty tiedostoformaatti (eng. JavaScript Object Notation)
M	analysointiin käytetty matemaattinen malli
$\max$	joukon suurin arvo
M_B	työssä käytetyn Bayes-verkon rakennetta kuvaava matemaattinen malli
M_F	lähtödatan kuvaamiseen käytetty matemaattinen malli
μ	normaalijakauman odotusarvo
NLP	luonnollisen kielen prosessointi (eng. Natural Language Processing)
NLTK	Python-ohjelmointikielen luonnollisen kielen prossointiin tarkoitettuja työkaluja sisältävä sovelluskirjasto (eng. Natural Language Toolkit)
PHP	Pääasiassa verkko-ohjelmoinnissa käytettävä ohjelmointikieli
$\mathbb{R}$	reaaliluvut
r_i	työntekijälle i rakennettu graafi
σ^2	normaalijakauman varianssi

SQL	relaatiotietokantojen kanssa kommunikointiin käytetty ohjelmakieli (eng. Structured Query Language)
T	tekstistä kerättyjen tokenien joukko
t_{ij}	työntekijän i osaamiskategorian j hallintaa kuvaava osaamisarvo
$t_{0,ij}$	osaamisarvon t_{ij} laskemiseen käytettävä pohja-arvo
TAU	Tampereen yliopisto (engl. Tampere University)
TF-IDF	termien painottamiseen käytetty kerroin (engl. term frequency-inverse document frequency)
token	luonnollisen kielen prosessoinnissa käytettävä termi sanalle, sanojen yhdistelmälle tai muulle merkkijonolle, jolla on jokin yksittäinen merkitys
TUNI	Tampereen korkeakouluyhteisö (engl. Tampere Universities)
U	työntekijöiden eri aihetermeille altistumista kuvaava matriisi
URL	verkkosivun osoite (engl. Uniform Resource Locator)
UUID	universaali uniikki identifioija (engl. Universally unique identifier)
v_{ij}	työntekijän i osaamisverkon termiä j vastaava solmu

1 JOHDANTO

Nykyisessä informaatioyhteiskunnassa on yrityksillä enenevissä määrin kasvavat intressit erilaisen tiedon keräämiseen, säilömiseen ja analysointiin. Futuricen kaltaisten jatkuvasti mukautuvien ja ketteriä menetelmiä käyttävien yritysten on erityisen tärkeää varmistaa, että eri työntekijöiden tietotaito ja eri projekteista saatu kokemus saadaan jaettua työntekijöille mahdollisimman tehokkaasti ja myös hyödynnettyä tulevien projektien suunnittelussa ja toteutuksessa. Futuricen tietohallintajärjestelyjä on kartoitettu aikaisemminkin, esimerkiksi Mathias Timosen Aalto-yliopistolle toteuttamassa opinnäytetyössä [1].

Timonen korostaa työssään tiedon hakijoiden ja tiedon omaajien yhdistämisen tärkeyttä, sekä mainitsee myös tiedonhallinnan tärkeyden tarjouksia tehdessä. Tehtäessä tarjousta uudesta sovellusprojektista jollekin asiakkaalle on tärkeää tietää kuinka paljon osaamista yrityksen sisällä on niin projektiin liittyvistä teknisistä aiheista kuin asiakkaastakin. [1, s. 51, 127]

Työssään Timonen on tarkastellut erilaisia Futuricella olevia tiedonjakokanavia ja toteaa kahdeksi merkittävimmäksi Flowdock-keskustelusovelluksen, sekä joka perjantai järjestettävät tiedonjakosessiot. Hän myös mainitsee erikseen, että vaikka Flowdock on loistava esimerkki yhteisöllisestä tiedonjaosta, se ei aina ole luotettavin mahdollinen skaalautuvuusongelmien vuoksi. Suuri osa tiedosta jaetaan kysymällä Flowdockissa kysymyksiä ja toivomalla jonkun vastaavan, mutta tarvetta on myös enemmän hajallaan olevan informaation kokoamiseen ja tulkitsemiseen. [1, ss. 130-133]

1.1 Futurice

Futurice Oy on moderni kansainvälinen yritys, jonka päätoimipaikka sijaitsee Helsingissä. Sen päätoimiala on Yritys- ja yhteisötietojärjestelmässä “62010 Ohjelmistojen suunnittelu ja valmistus” [2]. Yrityksen verkkosivuilla mainitaan, että Futurice on “saumaton yhdistelmä strategiaa, designia ja insinöörityötä” ja että yritys “auttaa asiakkaitaan vapauttamaan innovaation digitaalisten tuotteiden suunnittelun sekä rakentamisen, nousevien teknologioiden, agiilin sovelluskehityksen sekä sulavan organisaatiomuutoksen kautta” [3].

Yksinkertaisesti sanottuna Futurice on yritys, joka tarjoaa sovelluskehitys-, design- sekä konsultointipalveluita. Yrityksellä on yhteensä 600+ työntekijää [3] ja näiden työntekijöiden omaamien taitojen voidaan olettaa liittyvän pääasiassa joko johonkin yrityksen tarjoamista palveluista tai yrityksen sisäiseen johtotoimintaan.

1.1.1 Futurice Exponential

Futurice Exponentialin tavoitteena on rakentaa erilaisia kokeellisia tekoälyratkaisuja Futuricen sisällä, parantaen yrityksen osaamista ja projektien onnistuessa myös yrityksen omaa toimintaa. Exponential on osa Futuricen jatkuvaa tavoitetta pysyä muuttuvan IT-teknologioiden ympäristön aallonharjalla ja tarjota asiakkaille moderneja ja laadukkaita data- ja tekoälyratkaisuja.

Tämä työ on osa Exponentialin kokeellista sisäistä kehitystä, joka ei suoraan tähtää valmiiseen tuotteeseen. Työ tutkii monimutkaista, mutta tärkeäksi todettua ongelmaa, jonka ei odoteta ratkeavan täysin tämän yksittäisen tutkimuksen tuloksena.

1.2 Tutkittava ongelma

Yrityksen työntekijöistä valtaosa on osallistunut ainakin kerran keskustelusovellus Flowdockin jaetuilla kanavilla käytyihin keskusteluihin. Mitä enemmän henkilöt ovat osallistuneet keskusteluihin, sen enemmän ja sen varmemmin heidän osaamisestaan voidaan päätellä asioita. Lopulta tietoon saadaan kuitenkin vain pieni osa yrityksen hallussa olevasta hypoteettisesta kokonaistietotaidosta ja arvio tämän osan koosta sekä käyttökelpoisuudesta on kriittinen menetelmän hyödyllisyyden pohtimiselle.

Työntekijöiden tietotaidon analysoinnin lisäksi tärkeää on tietotaidon mallintaminen sekä päivittäminen muualta saadun täydentävän tiedon pohjalta. Yksittäisiltä työntekijöiltä voidaan kysyä heidän omaa mielipidettään omasta osaamisestaan ja yrityksen sisäinen Power-järjestelmä sisältää työntekijöiden itse merkitsemiä arvioita heidän työkokemuksestaan eri ohjelmointikielten, työkalujen, yms. parissa. Tässä työssä yksittäisiltä työntekijöiltä kerättyä tietoa voidaan käyttää mallin tuottamien arvioiden tarkentamiseen ja Power-järjestelmän sisältämää tietoa käytetään pohjatotuutena mallin tulosten arvioinnissa.

Power-järjestelmään merkattavien kokemustietojen kategoriat ovat ennalta määriteltyjä, hierarkkisesti jäsenneltyjä ja saatavilla vertailua varten. Kategoriat ovat listattuna liitteessä A. Mallin tulosten sekä Power-järjestelmän tietojen suoran vertailun mahdollistamiseksi mallin pitää myös tuottaa arvio työntekijän osaamisesta näissä samoissa kategorioissa ja samalla hierarkiarakenteella.

Tutkimuksen pääkysymykset voidaan määritellä seuraavasti:

- Kuinka paljon työntekijöiden osaamisesta ennalta määritellyissä taitokategorioissa voidaan saada selville valittua menetelmää käyttäen suhteutettuna työntekijöiden kokonaismäärään sekä hypoteettiseen yrityksen hallussa olevaan kokonaistietotaitoon?
- Soveltuuko työssä valittu malli työntekijän tietotaidon mallintamiseen?

Tutkimuksen tukikysymykset ovat seuraavat:

- Miten valittua mallia voisi parantaa?
- Kuinka luotettavaa yrityksen sisäinen keskusteludata on työntekijöiden osaamisen arvioimisessa?

1.3 Menetelmän valinta

Käytettävä prosessi jakaantuu pääasiassa kolmeen osaan. Ensin on käytettävissä olevan datan esiprosessointi analysoitavaan muotoon, sitten on työntekijöiden osaamismallin rakentaminen ja lopuksi on mallin päivittäminen saadun lisätiedon pohjalta.

1.3.1 Datan esiprosessointi

Kirjoitetun luonnollisen kielen prosessointiin (natural language processing, lyh. NLP) on olemassa useita eri keinoja eri monimutkaisuusasteilla ja eri vaatimuksilla. Teollisissa ratkaisuissa yleisesti käytetty tiedonhakukirjasto Lucene-kirjasto pilkkoo tekstin tokeneiksi kutsutuiksi palasiksi ja se pyrkii sitten etsimään dokumentteja, joiden sisältämät tokenit vastaavat hakulauseesta poimittuja tokeneita halutuilla painotuksilla sekä reunaehdoilla. Yksittäisten tokenien tärkeyttä tarkastellaan niiden esiintymismäärien sekä eri dokumentteihin hajautumisen kautta. [4]

Monet NLP-sovellukset menevät vielä syvemmälle ja hyödyntävät monimutkaisia algoritmeja tekstin rakenteen sekä yksittäisten sanojen merkityksen analysointiin. Esimerkiksi German Research Centre for Artificial Intelligence (DFKI) ja Saarland University tutkivat yhdessä eritasoisia tekstianalysointimetodeja kysymyksiin vastaamisessa. [5]

Tässä työssä keskitytään pääasiassa erilaisten avainsanojen tunnistamiseen sekä keskustelupalvelusta saadun datan tarjoaman kontekstitiedon analysointiin. Kun yksittäiset tokenit yhdistetään keskusteluihin sekä keskustelukanaviin, voidaan päätellä asioita keskusteluun osallistuneista henkilöistä. Valitaan siis tekstin prosessointitavaksi yksinkertainen tokenien poimiminen. Hieman lisähaastetta tuottaa se, että pääasiassa englanninkielisen tekstin seassa on myös suomeksi lähetettyjä viestejä.

1.3.2 Osaamismalli ja mallin päivittäminen

Osaamismallin pitää työn vaatimuksien mukaiseksi pystyä tarjoamaan tieto työntekijän osaamisesta muodossa, jota on mielekästä vertailla Power-järjestelmästä saatuihin tietoihin. Mallin pitää ilmaista työntekijän osaamistason lisäksi tiedon varmuus ja sitä pitää pystyä päivittämään saadun varman tiedon pohjalta. Power-järjestelmässä kategoriat on määritelty hierarkkisesti puumaiseen rakenteeseen, joten mallin on suositeltavaa kyetä ilmaisemaan samanlaista rakennetta.

Yrityksien kohtaama ongelma yrityksen sisäisen ja erityisesti yksittäisten työntekijöiden osaamisen ja tietotaidon arvioimisessa on monin tavoin rinnastettavissa pedagogisessa käytössä oleviin opetussovelluksiin, joissa oppilaan tietotaitoa ja sen kehittymistä pyritään

seuraamaan opetettavan sisällön yksilöimiseksi. Pedagogian alalla on jo pitkään ollut tiedossa, että yksilöllinen ja oppilaalle kohdistettu opetus voi parantaa oppilaan tuloksia merkittävästi perinteiseen ryhmäopetukseen verrattuna [6, s. 10].

Suuri osa opetussovelluksien käyttämien modernien oppimismallien menestyksestä juontaa juurensa niiden kykyyn päätellä oppilaan tietotaitoa myös niiden konseptien osalta, jotka eivät ole suoraan kytköksissä testattaviin konsepteihin. Tämä on pääasiassa erilaisten graafirakenteiden ansiota, joissa konsepteja edustavat solmut yhdistyvät toisiinsa erilaisilla linkeillä. Kun oppilas vastaa annettuun kysymykseen oikein tai väärin, saadaan lisätietoa oppilaan osaamisesta kysymykseen liittyneiden konseptien osalta, sekä edelleen näihin konsepteihin liittyvien alakonseptien osalta. Erilaisia konsepteja testaavia kysymyksiä esittämällä saadaan näin rakennettua vähitellen tarkentuva kuva oppilaan osaamisesta ja opetusta voidaan painottaa oppilaan huonommin hallitsemiin konsepteihin. [6, ss. 14-16]

Bayes-verkot, jotka ovat erilaisia tapahtumia ja niiden välisiä riippuvuussuhteita kuvaavia todennäköisyyslaskennan graafirakenteita, ovat menestyneet hyvin opetustarkoitukseen luoduissa verkoissa, joissa oppilaalta jatkuvasti kerättävät vastaukset antavat lisätietoa verkon solmujen tiloista. Opetuksessa käytetyissä Bayes-verkoissa yksittäiset verkon solmut esittävät usein jotain isomman kokonaisuuden palasta, jonka oppilas on joko oppinut tai ei. Yksittäiset solmut voivat siis vain olla tiloiltaan joko tosia tai epätosia. Mallin rajallisuutta täydennetään joskus piilotetuilla solmuilla, jotka mallintavat aiheen jakautumista vielä pienempiin tuntemattomiin osasiin. Bayes-verkon avulla rakennettu malli mukautuu helposti saatuun informaatioon ja on lisäksi laskennallisesti verrattain yksinkertainen prosessoida myös tietokoneella, minkä vuoksi se soveltuu hyvin käytännön prosessointitehtäviin. [6, ss. 16-25]

Tässä työssä käytetyt aihekategoriat johdetaan tuloksien vertailun helpottamiseksi Power-järjestelmän taitokategorioista, mikä tekee puun yksittäisten solmujen vastaamista asioista määritelmältään paljon laveampia. Yksittäiselle solmulle pitää olla tästä syystä mahdollista asettaa useampia eri tiloja. Piilotetut solmut ovat mahdollisia, mutta ne hidastavat laskentaa, monimutkaistavat algoritmin toimintaperusteita ja tekevät tuotetun tiedon vertailemisesta referenssiarvoihin hankalaa, joten niiden käytöstä luovutaan. Piilotettujen solmujen edustamien asioiden tarkempi määrittely on myös tässä kontekstissa hankalaa ja epämääräistä, mikä ei ole algoritmin toiminnan ja tuloksien tarkastelun kannalta mielekästä.

Opetukseen kehitettyjen menetelmien yksi ongelma on, että ne ottavat uutta tietoa vastaan vain verkon lehtisolmujen kautta. Ne vaativat joukon kysymyksiä analyysin pohjaksi ja oppilaiden voidaan olettaa vastaavan riittävän moneen kysymykseen tiedon keräämiseksi (vastaamatta jättäminen lasketaan usein vääräksi vastaukseksi). [6, ss. 20-24] Keskusteludataa analysoitaessa on edullista pyrkiä hyödyntämään jokainen mahdollinen tiedon jyvänen, vaikkei se kohdistuisikaan suoraan lehtisolmuihin. Tämä vaatimus johtuu osittain käytettävissä olevan Power-järjestelmän kategorioista johdetun puun haasteellisesta rakenteesta sekä tekstin prosessointiin käytettävien monimutkaisempien NLP-

menetelmien puutteesta.

Toinen ongelma opetukseen kehitetyissä menetelmissä on, että ne ovat huonoja ilmaisemaan epävarmuutta. Suosittu "Bayesian Knowledge-Tracing”-menetelmä esimerkiksi sisällyttää algoritmiin epävarmuuden oppilaan antamista vastauksista “guess”- ja “slip”-parametrien muodossa, mutta tämä tieto tulee algoritmin ulkopuolelta ja lopputuloksena annettu tieto sisältää vain todennäköisyyden siitä, onko taito hallittu vai ei. [6, s. 22] Tämän työn vaatimuksiin kuuluu arvion muodostaminen työntekijän osaamisesta saadun tiedon varmuudesta ja keskusteludataa analysoidessa on hyvin mahdollista saada puutteellisen tai jopa lähes olemattoman määrän todisteita jostain henkilöstä. Siinä missä opetuksessa tietty määrä selkeästi tulkittavaa ja jäsenneltävää lähtödataa on käytännössä taattu, tekstidataa analysoidessa täysi tietämättömyys on validi lähtökohta ja täysi epävarmuus on yksi sen mahdollinen johtopäätös. Tämä vaatii vähintäänkin useamman eri osaamistason ja jonkinlaisen jakauman niiden todennäköisyyksille, jolloin jakauman varianssista voidaan arvioida algoritmin tuottaman arvion validius.

Yleisesti voidaan siis mainita, että oppilaan taidon testaamiseksi tehtyjen kokeiden ja keskusteludatan analysointi on luonteeltaan niin erilaista, etteivät opetukseen käytetyt mallit sovellu sellaisinaan tässä työssä käytettäviksi. Pitkäaikaisen tutkimuksen pohjalle rakennettujen mallien sijaan työn algoritmissa pääasiallisesti käytettävä malli joudutaan siis rakentamaan käytännössä alusta, huomattavasti huterammalle pohjalle. Mallin rakentamisessa mukaillaan kuitenkin Bayes-verkkojen rakennetta ja se rakennetaan syklittömäksi suunnatuksi graafiksi.

2 TEORIA

2.1 Bayesin teoreema

Ehdollinen todennäköisyys on kahden toisistaan riippuvan satunnaisilmiön vuorovaikutusta, jossa tapahtuman A todennäköisyys kun tapahtuman B tiedetään tapahtuneen, saadaan laskettua kaavasta

$$P(A \mid B) = \frac{P(A, B)}{P(B)}, \quad P(B) \neq 0, \tag{2.1}$$

missä $P(B)$ on tapahtuman B todennäköisyys, $P(A, B)$ on todennäköisyys kummankin tapahtuman samaan aikaan tapahtumiselle ja $P(A \mid B)$ on ehdollinen todennäköisyys tapahtuman A tapahtumiselle, kun B on tapahtunut. Todennäköisyysfunktio P kuvaa tilanteesta tiedettyä taustatietoa. [7, s. 32] [8]

Bayesin teoreeman mukaan tämä todennäköisyys voidaan ilmaista myös kaavalla

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \quad P(B) \neq 0, \tag{2.2}$$

missä $P(B \mid A)$ on ehdollinen todennäköisyys tapahtuman B tapahtumiselle kun A on tapahtunut ja $P(A)$ sekä $P(B)$ ovat tapahtumien A ja B havaitut todennäköisyydet [7, s. 33]. Eli kun tapahtumien kontekstista tiedetään tarpeeksi eri tapahtumien riippuvuussuhteiden päättelemiseksi, voidaan yhden tapahtuman varmistuttua todeksi tai epätodeksi laskea suoraan myös toisen siitä riippuvan tapahtuman uusi todennäköisyys.

Bayesin teoreemassa (2.2) oikean käden puolella oleva $P(A)$ edustaa alussa olevaa uskomusta tapahtuman A tapahtumisesta, eli sitä kutsutaan prioriksi (eng. “prior belief”). Jakajassa oleva $P(B)$ edustaa uskomusta siitä, että tapahtuman B todetaan tapahtuneen yleisissä olosuhteissa. Tällöin $P(A \mid B)$ edustaa uskomusta siitä, että tapahtuma A tapahtuu tai on tapahtunut, kun tapahtuman B tila eli sitä edustavan satunnaismuuttujan saama arvo tiedetään. Tätä todennäköisyyttä $P(A \mid B)$ kutsutaan posterioriksi B:ssä (eng. posterior belief in B). [7, s. 33]

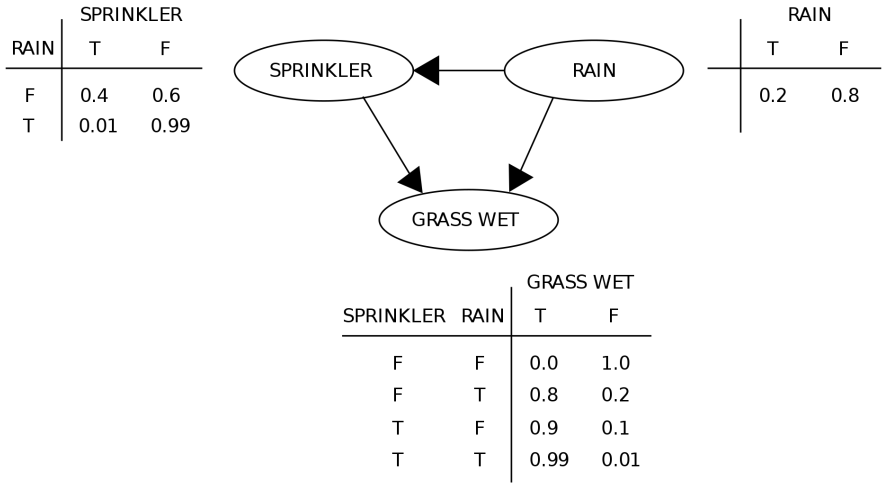
Bayesin teoreema voidaan myös esittää muodossa

$$P(A_1 \mid B) = \frac{P(B \mid A_1)P(A_1)}{P(B \mid A_1)P(A_1) + P(B \mid A_2)P(A_2) + \cdots + P(B \mid A_n)P(A_n)} = \frac{P(B \mid A_1)P(A_1)}{\sum_A P(B \mid A)P(A)}, \tag{2.3}$$

missä $A_1, A_2, \ldots A_n$ ovat satunnaismuuttujan A mahdollisia tiloja, eli $A = A_1 \cup A_2 \cup \cdots \cup A_n$ ja joukot $A_1, A_2, \ldots A_n$ ovat pistevieraat. Summa $\sum_A$ tarkoittaa summausta kaikkien tapahtuman A mahdollisten tilojen yli [8]. Kaava on todennäköisyyslaskennassa hyvin intuitiivinen, sillä siinä jaetaan tapahtuman todennäköisyys kaikkien mahdollisten tapahtumien yhteistodennäköisyydellä.

2.2 Bayes-verkot

Bayes-verkot ovat todennäköisyyslaskennassa käytetty työkalu, jossa satunnaismuuttujien väliset ehdolliset riippuvuudet mallinnetaan käyttäen syklitöntä suunnattua graafia (Directed Acyclic Graph, DAG) ja graafissa olevien toisiinsa yhdistettyjen satunnaismuuttujien vaikutussuhteita kuvaavia todennäköisyystaulukoita [9, ss. 8-9]. Esimerkki Bayes-verkosta on kuvassa 2.1.



Kuva 2.1. *Esimerkki Bayes-verkosta. [10]*

Kuvassa 2.1 satunnaismuuttuja “sprinkler” kuvaa sprinklerin tilaa (T (true) = päällä, F (false) = pois päältä), “rain” kuvaa sateista säätä (T = sataa, F = ei sada) ja “grass wet” kuvaa nurmikon märkyyttä (T = märkä, F = kuiva). Nuolet kuvaavat satunnaismuuttujien vaikutussuhteita ja niiden suuntia, esimerkiksi sää voi vaikuttaa nurmikon märkyyteen, mutta nurmikon märkyys ei vaikuta säähän. Todennäköisyystaulukoista nähdään satunnaismuuttujien kaikkien mahdollisten tilojen todennäköisyydet hierarkiassa ylempänä olevien vanhempien solmujen satunnaismuuttujien eri tiloille. Jos esimerkiksi tiedetään ulkona satavan ja sprinklerin olevan päällä, nähdään satunnaismuuttujan “grass wet” todennä-

köisyystaulukosta märän ruohon todennäköisyyden olevan 0,99.

Bayes-verkkojen avulla voidaan rakentaa probabilistinen malli, jossa satunnaismuuttujia esittävät solmut ovat kytkeytyneet toisiinsa vaikutussuhteita esittävillä kytköksillä, jolloin jokaisella satunnaismuuttujalla on selkeät vaikuttavat tekijät sekä selkeät vaikutuksen kohteet. Tämä auttaa käsittelemään suuria määriä satunnaismuuttujia sisältäviä tilanteita, joissa muuten jouduttaisiin laskemaan kaikkien arvojen keskinäiset vaikutussuhteet. [7, s. 3] Vaikutussuhteita mallinnetaan Bayesin teoreemalla (2.2), joka kuvaa koko verkon todennäköisyyksien muuttumista lisätiedon pohjalta.

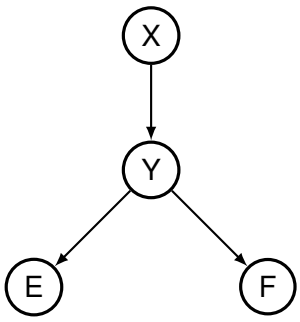
Todennäköisyystaulukoiden sisältämät arvot ovat useimmissa tapauksissa ammattilaisen asettamia arvoja. Tämä on yksi Bayes-verkkojen heikkous ja haittaa erityisesti isojen Bayes-verkkojen rakentamista. [6, s. 19]

2.2.1 Bayes-verkon päivittäminen todisteiden pohjalta

Bayes-verkon selkeimpiä vahvuuksia on sen helppo päivittäminen lisätiedon pohjalta. Kun käytettäessä kuvan 2.1 mukaista Bayes-verkkoa tiedetään ulkona satavan, voidaan sprinklerin todennäköisyystaulukosta katsoa suoraan sprinklerin päällä olon todennäköisyyden olevan 1%. Jos taas tiedetään sprinklerin S olevan tosi, voidaan Bayesin teoreemaa (2.2) käyttämällä laskea sateen R todennäköisyyden olevan

$$P(R \mid S) = \frac{P(S \mid R)P(R)}{P(S \mid R)P(R) + P(S \mid \neg R)P(\neg R)} = \frac{0.01 \cdot 0.2}{0.01 \cdot 0.2 + 0.4 \cdot 0.8} = 0.000644 \approx 0.00.$$

Tarkastellaan kuvan 2.2 mukaista Bayes-verkkoa. Verkko sisältää diskreetit satunnaismuuttujat X, Y, E ja F ja niiden saamia arvoja merkitään vastaavilla pienillä kirjaimilla. Esimerkiksi muuttuja X voi saada arvot $x_1, x_2, \ldots, x_i$.



Kuva 2.2. *Esimerkki Bayes-verkosta verkon päivittymisen kuvaamiseksi. [7, s. 42]*

Jos saadaan tietää satunnaismuuttujan X olevan arvossa x_1, voidaan muuttujan Y tilojen eli mahdollisten arvojen todennäköisyydet $P(y \mid x_1)$ katsoa suoraan muuttujan Y todennäköisyystaulukosta. Vastaavasti voidaan laskea esimerkiksi $P(e \mid x_1) = \sum_y P(e \mid y)P(y \mid x_1)$. [7, s. 42]

Jos satunnaismuuttujien E ja F tilasta saadaan tietoon niiden olevan arvoissa e_1 ja f_1,

päivittyy satunnaismuuttujan X todennäköisyysjakauma Bayesin säännön (2.3) mukaan

$$P(x \mid e_1, f_1) = \frac{\sum_y P(f_1 \mid y)P(e_1 \mid y)P(y \mid x)P(x)}{\sum_{yx} P(f_1 \mid y)P(e_1 \mid y)P(y \mid x)P(x)},$$

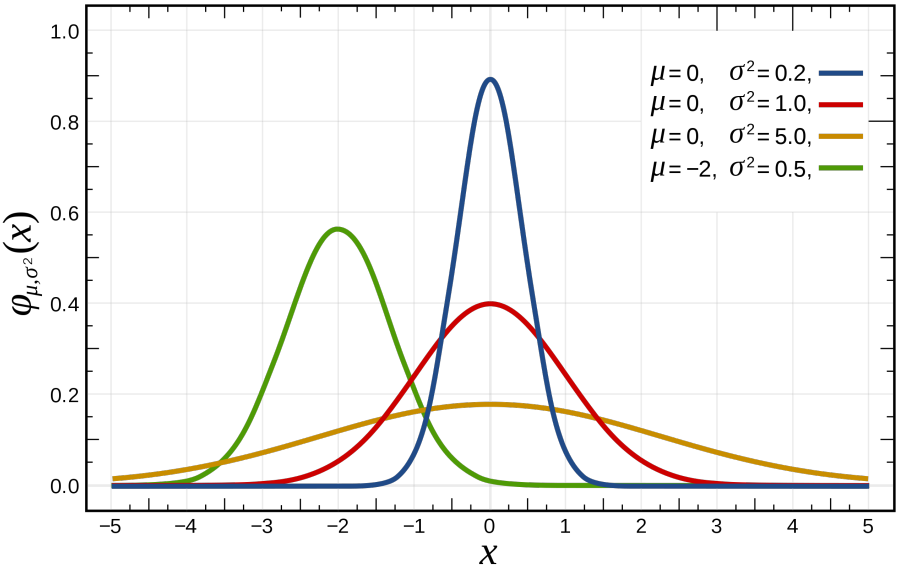
missä $P(x \mid e_1, f_1)$ sisältää posterioritodennäköisyydet ja oikealla puolella esiintyvä $P(x)$ prioritodennäköisyydet. [7, ss. 42 - 43]

2.3 Gaussinen jakauma

Gaussinen jakauma eli normaalitodennäköisyysjakauma on jatkuva reaalisen satunnaismuuttujan todennäköisyysjakauma, jonka tiheysfunktio määritellään

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2.4}$$

missä μ on normaalijakauman odotusarvo ja σ^2 on varianssi. Jakauma antaa sen edustamalle satunnaismuuttujalle suurimman todennäköisyyden odotusarvon kohdalla ja todennäköisyyden vähenemisen jyrkkyys odotusarvosta etääntyessä kasvaa varianssin pienentyessä. Kuvaaja lähestyy x-akselia äärettömyydessä kummassakin suunnassa. [11]



Kuva 2.3. *Esimerkki normaalitodennäköisyysjakaumanfunktion tuottamasta jakaumasta muutamalla eri odotusarvon ja varianssin arvoilla. [12]*

Kuvassa 2.3 on esitetty muutama eri normaalitodennäköisyysjakauma odotusarvon ja varianssin eri arvoilla. Varianssin kasvaessa kuvaajan muodostama kumpu muuttuu matalammaksi ja leveämmäksi. Odotusarvon muutos vain siirtää huippua.

2.4 TF-IDF

TF-IDF on kerroin, jota käytetään luonnollisen kielen prosessoinnissa (NLP) tekstissä olevien termien painottamiseen niiden tärkeyden mukaan. Tavoitteena on antaa dokumentissa usein esiintyville termeille enemmän painoa (term frequency TF), mutta samalla vähentää useissa eri dokumenteissa esiintyvien termien painoa (inverse document frequency IDF). Mitä useammassa dokumentissa termi esiintyy, sen pienemmäksi sen informaatiosisältö voidaan olettaa. [13]

Kertoimen laskemiseen on useampia erilaisia tapoja, joilla on marginaalisia eroja lopputuloksen kannalta. Kaavojen erojen vaikutuksen todetaan olevan työn kokonaisuuden kannalta merkityksetön, joten valitaan suhteellisen yksinkertainen ja yleisesti käytetty versio

$$\text{tfidf}(t,d,D) = \text{tf}(t,d) \cdot \text{idf}(t,D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \cdot \log_e \frac{|D|}{|\{d \in D : t \in d\}|}, \tag{2.5}$$

missä $f_{t,d}$ on termin t frekvenssi dokumentissa d ja D on kaikkien dokumenttien joukko. [13] [14] Tässä työssä tarkastellaan vain dokumenteista löytyneitä termejä, jolloin jakaja $|\{d \in D : t \in d\}|$ ei ole missään tilanteessa nolla.

2.5 Matemaattisten mallien määrittelyt

Työssä käytettävää keskusteludataa mallinnetaan $M_F = (F, K, S, W, T)$, missä

- F on keskusteluhuoneiden (eng. flow) joukko,
- K on keskusteluhuoneissa käytyjen keskustelujen (eng. thread) joukko,
- S on keskusteluihin lähetettyjen viestien (eng. message) joukko,
- W on keskusteluihin osallistuneiden työntekijöiden joukko,
- T on keskusteluista kerättävien käyttökelpoisten eli suodatettujen tokenien joukko.

Kaikkien työntekijöiden joukkoa voidaan merkitä W_{tot} ja kaikkien tekstistä löydettävien tokenien joukkoa T_{tot}. Nämä eivät tosin ole varsinaisen algoritmin kannalta oleellisia määrittelyjä.

Työssä käytettävän matemaattisen mallin tulee pystyä esittämään jokaiselle yksittäiselle työntekijälle W_i (merkitään myöhemmin vain i) hänen osaamistasonsa sekä tämän tiedon epävarmuuden yksittäisissä toisiinsa erivahvuisilla suhteilla kytkeytyneissä kategorioissa. Matemaattinen malli määritellään suunnattuna syklittömänä graafisena mallina $M = (R, V, E, A, \beta, t, e)$, missä

- $R : i \to (V_i, E_i)$ on jokaiselle työntekijälle i erikseen luotavien työntekijän tietotaidon jakautumista esittävien osaamisgraafien r_i joukko,
- $V = \bigcup V_i$ on graafeissa r_i olevien solmujen v_{ij} joukko,

- $E = \bigcup E_i$ on graafeissa r_i olevien kytkösten e_{ij} joukko,
- $A : V \to \Sigma^*$ ovat graafin solmuja vastaavat luonnollisen kielen aihekategoriatermit,
- $\beta : E \to [0, 1]$ ovat graafissa olevien kytkösten painotukset,
- $t : V \to [0, 1]$ ovat osaamisarvot, eli graafin omaavien henkilöiden osaamistasot graafien solmujen edustamiin termeihin liittyen,
- $e : V \to [0, \infty)$ ovat epävarmuusarvot, eli graafin omaavien henkilöiden osaamistasojen tietojen epävarmuudet graafien solmujen edustamiin termeihin liittyen.

Osaamistason arvoille t asetetaan rajoitteeksi, että niiden pitää olla reaalilukuja välillä $[0, 1]$. Tällöin odotusarvo 0 tarkoittaa täyttä tietämättömyyttä asiasta ja odotusarvo 1 tarkoittaa vastaavasti asian täydellistä hallintaa. Osaamistason virheen e ainoa rajoite on, että sen pitää olla positiivinen reaaliluku.

Tulosten tarkastelua varten malli muutetaan Bayes-verkkomalliksi $M_B = (R, V, E, H, X)$. Tässä työssä käytetty Bayes-verkko on diskreetti, eli jokainen satunnaismuuttuja X_i voi saada $N \in \mathbb{Z}_+$ kappaletta erilaisia arvoja eli tiloja. Malliin M_B kuuluvat joukot määritetään mallin M kohdalla määritettyjen joukkojen lisäksi

- $H : V \to [0, 1]^{N^{||V_E||} \times N}$ ovat Bayes-verkkojen solmuja vastaavat totuustaulukot,
- $X : V \to \{x_1, x_2, \ldots, x_N\}$ ovat Bayes-verkkojen solmuja vastaavat satunnaismuuttujat, joilla on N kappaletta mahdollisia tiloja x_i,

missä $V_E \subset V$ on suunnatussa graafissa solmun vanhempien eli hierarkiassa ylempänä olevien solmujen joukko, joka muodostetaan jokaiselle solmulle erikseen.

3 MALLIN LUOMINEN

3.1 Flowdock-data

Flowdock-data koostuu kaikista Flowdock-viestintäpalveluun Futuricen organisaation keskusteluhuoneisiin lähetetyistä viesteistä aikavälillä 7.1.2013 – 21.1.2019, pois lukien suoraan toiselle henkilölle lähetetyt henkilökohtaiset viestit. Yrityksen virallinen kieli on englanti ja valtaosa keskustelusta on tällä kielellä, mutta joukossa on myös muutamia suomenkielisiä keskusteluja. Datan karsittu skeema on taulukossa 3.1.

Taulukko 3.1. *Flowdock-viestidatan skeema, josta karsittu pois epäolennaisia kenttiä.*

Kentän nimi	Tyyppi	Kuvaus	Esimerkkiarvo
id	integer	Viestin ID	12345678
uuid	string	Viestin UUID	'XswOxxxx-qXKcg4n'
app	string	Sovelluksen tyyppi	'chat'
event	string	Tapahtuman tyyppi	'message'
flow	string	Kanavan UUID	'1c52d26f-a35d-4e10-bcc3-03e7ff0237d8'
user	integer	Käyttäjän ID	123456
content	string	Viestin sisältö	'wait, this thread is now super confusing'
created_at	datetime	Viestin lähetysajankohta	'2019-01-15T12:09:26.357Z'
tags	string[]	Lista tageja	[':url', ':user:234566']
tread.created_at	datetime	Keskustelun luontiajankohta	'2019-01-14T14:34:27.000Z'
tread_id	string	Keskustelun ID	'bliBEMdgiSEGi1AAAAqb4Ga7VrO'
thread.internal_comments	float	Keskustelun viestien lukumäärä	21.0

Kenttien `app` ja `event` avulla varmistetaan datan koostuvan vain viesteistä, sillä Flowdock tukee myös muunlaista sisältöä. Tarkastuksen suorituksen jälkeen näitä kenttiä ei enää tarvita. Käytetyn datan joukosta ei löytynyt muita datatyyppejä.

Viestejä on yhteensä 361948. Näistä 484 sisältää tyhjän `uuid`-arvon, vaikka tietue on muuten validi. Datasta löytyy myös kaksi muuten validia viestiä jaetulla `id`-arvolla ja eroavalla `uuid`- sekä `content`-arvoilla. Arvo `id` ei siis ole täysin uniikki jokaiselle viestille. Luodaan tästä syystä jokaiselle viestille uusi uniikki `id_tmp`-arvo ja asetetaan se viestin järjestysnumeroksi. Työssä käytetty data sisältää kuitenkin myös kahdentuneita tietueita, joilla on samat arvot kentissä `id` ja `uuid`, joten käytettävään viestijoukkoon otetaan mukaan näistä vain ensimmäiset. Viesteistä 1860 on sisällöltään tyhjiä, joten ne voidaan myös tiputtaa datasta pois. Suodatuksen jälkeen viestien määräksi jää 359968.

3.2 Raakadatan esikäsittely

Ennen viestien paloittelemista algoritmin käyttöön, niiden sisältö muokataan helpommin koneellisesti parsittavaan muotoon. Korkeatasoisessa luonnollisen kielen prosessoinnissa voitaisiin ehkä hyödyntää tekstipohjaisessa viestinnässä esiintyviä erikoismerkkejä, tunnetiloja ilmaisevia hymiöitä sekä muita viestirakenteen indikaattoreita, mutta tässä työssä suoritettavassa yksinkertaisessa kontekstianalyysissä hyödynnetään vain yksittäiset merkitykselliset sanat. Datan esikäsittelyprosessi on siis poikkeuksellisen raju.

3.2.1 Kommenttien esikäsittely ja jako tokeneihin

Kaikki teksti muutetaan ensin pieniksi kirjaimiksi. Sähköpostiosoitteet, HTTP-linkit, rivinvaihdot, Flowdockin hymiöt, muutamat yleiset hymiöt ja lauserakennetta ilmaisevat erikoismerkit poistetaan Python-ohjelmointikielen säännöllisiä lausekkeita hyödyntävien tekstinkorvaustoimintojen avulla. Tämän lisäksi poistetaan kaikki tähtimerkit, joita käytetään Flowdock-viestinnässä pääasiassa tekstin lihavointiin ja jäsentelyyn. Erikoismerkin ‘`’ kaikki esiintymät poistetaan myös. Sitä käytetään viestien koodiblokkien formatointiin, mutta se ei esiinny suomen tai englannin kielen sanoissa.

Suodatettu teksti jaetaan tokeneiksi käyttäen Pythonin NLTK-kirjaston tarjoamaa funktiota `word_tokenize` oletusasetuksilla, jolloin erilaiset lauserakenteen ilmaisemiseen käytetyt merkit jäävät pois. Saatujen tokenien joukosta poistetaan sellaiset, jotka ovat joko kahta merkkiä lyhyempiä tai kuuluvat NLTK:n suomen- tai englanninkielisten erotinsanojen listaan. Erotinsanat (eng. stop word) ovat luonnollisen kielen sanoja, joita käytetään lauseen rakenteen määrittämisessä, mutta jotka eivät sisällä mitään omaa merkitystä.

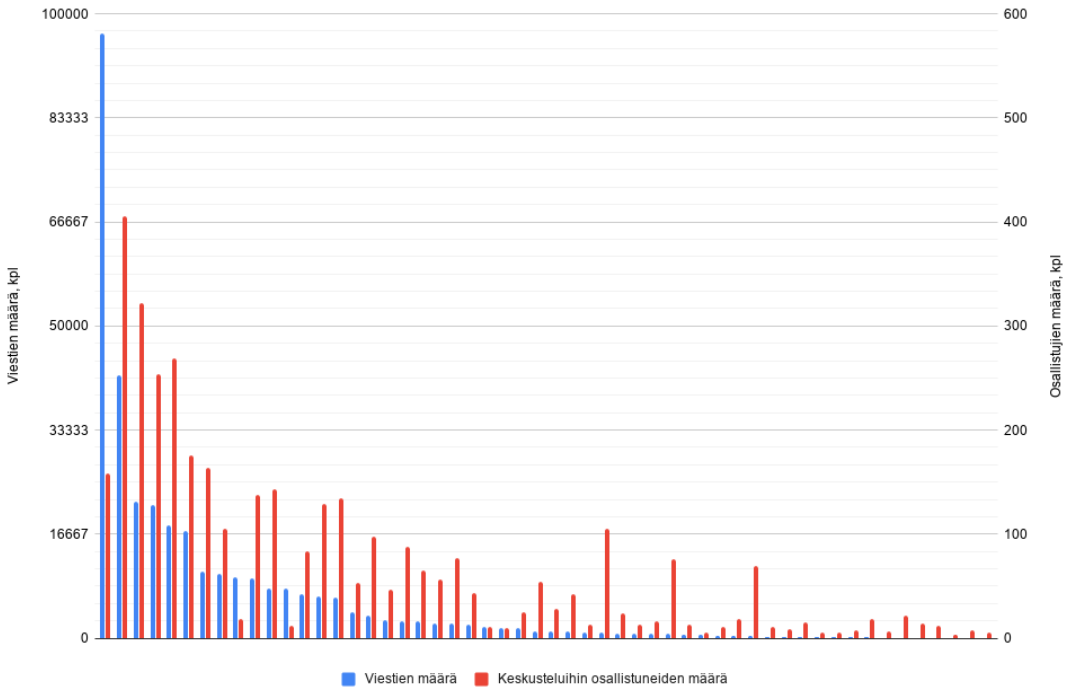
NLTK tuottaa yhden sanan tokeneita, mutta lauseissa vierekkäin esiintyviä tokeneita yhdistetään ns. yhdistelmätokeneiksi. Nämä yhdistelmätokenit sisältävät enintään kolme sanaa. Esimerkiksi lauseesta “Minä tulen huomenna.” saadaan tämän prosessoinnin tuloksena tokenit “minä”, “tulen”, “huomenna”, “minä tulen”, “tulen huomenna” ja “minä tulen huomenna”. Prosessoinnin lopputuloksena uniikkeja tokeneita syntyy yhteensä 4155073 kappaletta.

Osa viesteistä ei esikäsittelyn tuloksena tuottanut lainkaan tokeneita. Nämä viestit eivät tuo lopputulokseen lainkaan lisäinformaatiota ja niiden poistamisen jälkeen viestimäärä vähenee 359968:sta 334504:ään.

Lopuksi luodaan UUID-tunnus kaikkien niiden viestien keskustelutunnuksiksi, joilla sellaista ei vielä ole. Jos kukaan ei ole aloittanut ensimmäiseen viestiin vastaamalla viestiketjua, ei Flowdock generoi sille keskustelutunnusta. UUID-tunnuksen luominen tehdään käyttämällä UUID-tunnusformaatin neljättä versiota.

3.2.2 Viestien jakautuminen keskusteluhuoneisiin

Esikäsittelyn jälkeen jäljelle jääneet viestit jakautuvat keskusteluhuoneisiin kuvan 3.1 mukaisesti. Kaikkien keskusteluhuoneiden nimiä ei ole jaettu tässä työssä tietoturvasyistä, mutta kahdeksan suosituimman kanavan tiedot on listattu taulukossa 3.2. Kuvasta näkyy, että viestien määrä vähenee eksponentiaalisesti vilkkaimmasta keskusteluhuoneesta hiljaisimpaan. Taulukosta näkyy lisäksi, että suosituimpien keskusteluhuoneiden joukossa on Futuricen yleinen tiedotus- ja keskusteluhuone “futurice”, toimistokohtaisia ylisiä kanavia “tampere” ja “helsinki”, sovelluskehitykseen liittyviä keskusteluhuoneita “development”, “frontend-futurice” ja “backend-futurice” sekä harrastuskanava “gamers”. Keskusteluhuone “futunaut-it-shoutbox” on Futuricen sisällä toimistolta toiselle matkaavien IT-tukikanava.



Kuva 3.1. *Keskusteluhuoneisiin lähetettyjen viestien sekä osallistuneiden henkilöiden määrä suurimman viestimäärän keskustelusta pienimpään.*

Futurice-yrityksen toimialueisiin nähden on huomioitavaa, ettei käytössä olevaan dataan ole päätynyt yhtään yleistä myynnin, konsultoinnin tai suunnittelun kanavaa. Tämä tekee näiden osaamisalueiden työntekijöiden havainnoinnista huomattavasti hankalampaa ja keskusteluhuoneiden pohjalta tehtävästä kontekstianalyysistä yleisesti paljon epäluotettavampaa.

3.2.3 Tokenien karsiminen

Tokeneilla on luonnollisen kielen prosessoinnissa hyvin tärkeä rooli lauserakenteiden analysoinnissa, mutta tässä työssä ollaan kiinnostuneita pelkästään yksittäisten tokenien

Keskusteluhuoneen nimi	Viestien määrä	Osallistujien määrä
tampere	96830	158
futurice	42166	405
helsinki	21814	322
development	21262	254
futunaut-it-shoutbox	18146	269
frontend-futurice	17206	176
backend-futurice	10655	164
gamers	10348	105

Taulukko 3.2. *Kahdeksan eniten viestejä sisältävää keskusteluhuonetta sekä niiden viesti- ja osallistujamäärät.*

informaatiosisällöstä. Tästä syystä liian harvoin tai liian usein esiintyviä tokeneita voidaan tiputtaa käsittelystä suhteellisen paljon.

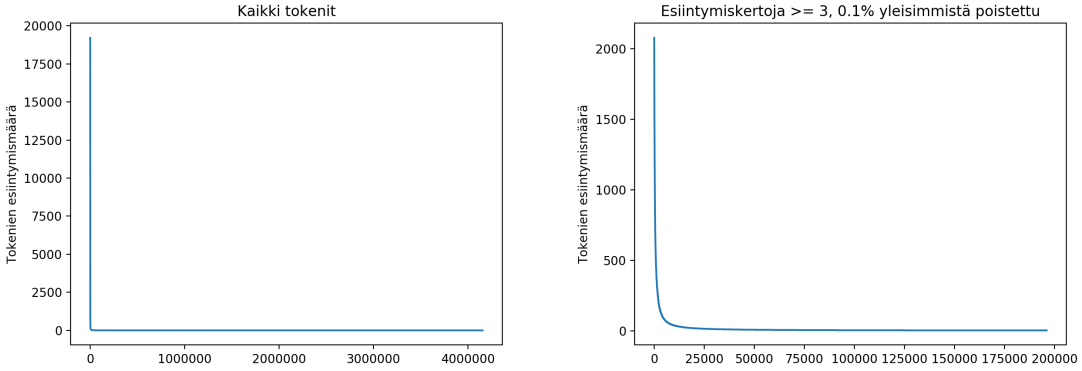
Osa tokeneista esiintyy liian harvoin, että niiden avulla voitaisiin päätellä mitään. Lisäksi osa tokeneista esiintyy tekstissä niin monta kertaa, että niiden voidaan olettaa olevan geneerisiä mihinkään aiheeseen erityisesti liittymättömiä englannin tai suomen kielen termejä. Eniten käytettyjen tokenien joukossa ovat mm. sanat ’would’, ’one’ ja ’like’. Normaalissa NLP-prosessissa näistäkin tokeneista voitaisiin olla kiinnostuneita, sillä esim. termistä ’like’ voitaisiin päätellä paljon henkilön suhteesta viestin käsittelemään asiaan. Tässä diplomityössä käytän kuitenkin yksinkertaisempaa kontekstianalyysiä, minkä vuoksi eniten käytetyt tokenit voidaan myös karsia pois.

Kokeilemisen jälkeen päädyttiin valitsemaan ensin ne tokenit, joilla oli kolme tai enemmän esiintymiskertoja. Tämän jälkeen suodatetaan vielä pois jäljelle jääneistä tokeneista 0.1 % verran eniten käytettyjä tokeneita. Suodatuksen vaikutus käytettävissä oleviin tokeneihin näkyy kuvassa 3.2. Suodatuksessa tippuu ensin pois 3958820 vain yksi tai kaksi kertaa esiintynyttä tokenia ja sen jälkeen 196 eniten käytettyä tokenia, jolloin jäljelle jää 196057 uniikkia tokenia. Tämä on suodatettujen tokenien joukko T.

3.3 Suomenkielisten viestien huomioiminen

Englanninkielisten viestien seassa olevat suomenkieliset viestit voivat olla sekä analysointia haittaava että edistävä tekijä. Kahden eri kielen käsittely kasvattaa kontekstianalyysissä käytettävää tietomäärää ja tekee siitä sanastoltaan hajanaisempaa. Suomen kieli sisältää myös paljon taivutusmuotoja, joiden käsittelemättä jättäminen pahentaa tilannetta entisestään. Suomenkieliset viestit sisältävät kuitenkin myös paljon tuotteiden ja palveluiden nimiä sekä lainasanoja, joiden avulla niistä saatava tieto voidaan yhdistää englanninkielisistä viesteistä saatuun tietoon.

Työssä kokeiltiin aluksi suomenkielisten viestien suodattamista hyvin yksinkertaisella menetelmällä, jossa merkkijonot ‘ä’, ‘ö’, ‘å’ ja ‘ ei ’ sisältävät viestit poistettiin suomenkielisinä.



Kuva 3.2. *Tokenien esiintymismäärät järjestettynä suurimmasta pienimpään. Vasemmalla kaikki tokenit, kun taas oikealla on ensin suodatettu pois kaikki alle 3 kertaa käytetyt ja sitten 0.1 % listan alkupäästä.*

Englanninkielisen tekstin ei pitäisi sisältää skandinaavisia aakkosia, eikä erillistä sanaa "ei". Tällä suodatuksella 359968:sta viestistä jäi jäljelle 314700 ja myöhemmin määriteltävien tokeneiden suodatusvaiheiden jälkeen tokeneita jäi jäljelle 166002. Ilman suomenkielisten viestien suodatusta tokeneita jää suodatuksien jälkeen 196057. Tämä muutos on iso, muttei näkynyt lopputuloksissa ratkaisevasti. Merkittävin havaittu muutos suomenkielisiä viestejä suodatettaessa oli, että yksittäisiä tuotteiden tai palveluiden nimiä sisältävät termit kuten “tableau” (data-analytiikkatyökalu) ja “solidity” (blockchain-kehityksessä käytetty ohjelmointikieli) saivat lopullisissa tuloksissa verrattain vähän pisteitä. Ilman suomenkielisten viestien suodatusta englanninkieliset termit kuten “ar design” ja “information architecture” pärjäävät huonommin.

Suomenkielisten viestien suodatus päätetään jättää pois lopullisista tuloksista. Algoritmi ei ole riippuvainen kielen rakenteista tavanomaisen luonnollisen kielen prosessointiin käytetyn algoritmin tavoin ja suuri osa prosessoimattomista suomenkielisistä termeistä tippuu harvinaisuutensa vuoksi pois tokeneita suodatettaessa.

3.4 Työntekijän tietoverkko

Jokaiselle keskusteluihin osallistuneelle työntekijälle i rakennetaan malliin M osaamisgraafi r_i, jonka solmuihin liitetyt arvot edustavat hänen osaamistaan solmun edustamaan termiin liittyen. Verkon yleinen rakenne on jaettu työntekijöiden kesken ja se on hierarkinen, sillä aiheet luokitellaan kategorioihin, joita edustavat solmut toimivat tarkempia aiheita edustavien solmujen vanhempina. Verkko voitaisiin teoriassa toteuttaa suuntaamattomana verkkona, mutta tällöin rinnakkaiset solmut vaikuttaisivat toisiinsa enemmän. Työntekijöiden voidaan olettaa olevan hyvinkin erikoistuneita tiettyihin työkaluihin, jolloin kategorian sisällä tapahtuvaa vaikuttamista on edullista pyrkiä vähentämään.

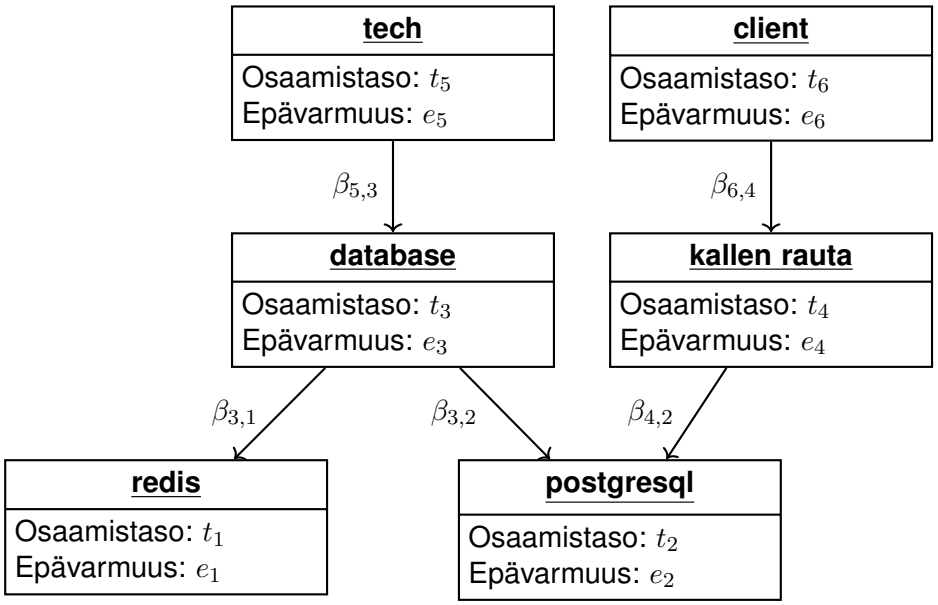
Henkilön i todettuun tietotaitoon aihetermiin a_j liittyen vaikuttavat suorien havaintojen lisäksi sitä vastaavaa verkon solmua v_{ij} hierarkiassa ylempänä olevien vanhempien solmujen $v_{E1}, v_{E2}, \ldots, v_{Em}$ aihetermeistä tehdyt havainnot. Jos työntekijä on esimerkiksi

vahva osaaja tietokantoihin liittyen, voidaan hänen olettaa tietävän jonkin verran myös PostgreSQL-tietokannoista, vaikkei hän täysin asiantuntija kyseisessä aiheessa olisikaan. Solmut $v_{E1}, v_{E2}, \ldots, v_{Em}$ voivat liittyä vahvemmin tai heikommin aihetermiä a_j vastaavaan solmuun v_{ij} ja niiden painotuksien pitää siksi erota toisistaan.

Bayes-verkot on kehitetty nimenomaan tämänlaisten vuorovaikutusten mallintamiseen, mutta toimiakseen tehokkaasti jokaiselle verkon solmulle pitää määrittää erikseen todennäköisyystaulukko. Tämä tehdään usein käsin eksperttien toimesta ja on hyvin aikaa vievä sekä riskialtis prosessi, mistä syystä useimmat Bayes-verkkoja käyttävät ratkaisut yksinkertaistavat oletuksia ja siten verkon toimintaa [6, s. 18]. Tässä työssä koko verkon rakenne on johdettu Power-järjestelmän kategorioista ja sisältää yhteensä 270 solmua, mikä tekee tarkasta Bayes-verkon määrittelemisestä sekä erittäin työlästä että laskennallisesti kallista. Verkon rakenteen tarkempaa käsin määrittelyä ja täyden Bayes-verkon käyttöä koko prosessin ajan voidaan tutkia erikseen, mutta tässä työssä on päädytty myös muodostamaan Bayes-verkosta yksinkertaisempi naiivi versio.

3.4.1 Verkon rakenne

Algoritmin suorittamisen aikana työntekijän tietotaitoa kuvataan mallin M mukaisella verkolla, jossa jokaiselle aihetermille kirjataan työntekijän osaamistaso aiheessa ja tämän tiedon virhearvio. Esimerkki tämänlaisesta verkosta on kuvassa 3.3.



Kuva 3.3. *Esimerkki työntekijän tietoverkosta. PostgreSQL-tietokanta on käytössä kuvitteelliselle Kallen Rauta -yritykselle tehtävässä projektissa.*

Power-järjestelmän taitokategorioiden pohjalta rakennettu Bayes-verkko sisältää muutamia solmuja, joissa termit ovat samat. Esimerkiksi termiä “tech” vastaa kaksi solmua, joista toinen on solmun “business director” bisnesjohtamista kuvaava lapsi ja toinen on solmun “lean service creation (lsc)” projektisuunnittelun teknologioita kuvaava vanhempi. Näitä solmuja käsitellään siten, että termille “tech” tulevat todisteet lisätään kumpaankin

solmuun ja termin lopputulosta laskettaessa otetaan keskiarvo näistä solmuista. Ratkaisu on hyvin puutteellinen, sillä kopiotermit voivat tarkoittaa hyvinkin eri asioita. Ideaalissa tilanteessa puu rakennettaisiin kokonaan ilman tämänlaisia termejä, mutta tässä työssä lähtövaatimus helposta vertailtavuudesta Power-järjestelmän tietojen kanssa pakotti valittuun ratkaisuun.

Vielä ikävämpi piirre Power-kategorioissa on kohta, jossa termiä “accessibility” vastaavan solmun lapsella on sama termi “accessibility”. Tässä tapauksessa lapsisolmu poistetaan rakenteesta kokonaan.

Kuvasta 3.3 poiketen työssä käytetty verkko ei sisällä lainkaan sellaisia solmuja, joilla olisi ollut enemmän kuin yksi vanhempi solmu. Verkko on siis rakenteeltaan puumainen ja tämä johtuu Power-järjestelmän kategorioiden puumaisesta rakenteesta. Verkkoon oltaisiin voitu lisätä kuvan 3.3 mukaisia asiakkaita edustavia useamman vanhemman solmuja käsin ja duplikaattitermejä oltaisiin voitu myös yhdistellä, mutta tämä olisi hankaloittanut sekä algoritmin muodostamista että tulosten vertailemista. Suunnattujen verkkojen päivittämisen kohdalla useamman vanhemman solmut aiheuttavat myös ns. “riesatermien” (eng. nuisance variable) yli summausta, mikä saattaa monimutkaistaa ja hidastaa tulosten päivittämiseen käytettäviä kaavoja [15, s. 2].

3.5 Verkon termien suhde tekstiin

Tietoverkko sisältää vain muutaman tarkkaan määritellyn kategorioita edustavan aihetermin a. Jotta verkon aihetermit saadaan suhteutettua Flowdock-datasta poimitussa tekstissä esiintyviin lukuisiin erilaisiin tokeneihin t, pitää verkon solmuja vastaavien aihetermien sekä tekstissä esiintyvien tokenien välille rakentaa jonkinlainen suhde. Yksinkertaisin tapa ilmaista tätä suhdetta on rakentaa matriisi $B \in [0,1]^{||A||\times||T||}$, joka edustaa kolumnissa olevan tokenin suhdetta rivillä olevaan aihetermiin. Esimerkki matriisin asettelusta ja arvoista on taulukossa 3.3.

	random	sentence	about	postgres	databases
tech	0.00	0.00	0.00	0.23	0.76
client	0.00	0.00	0.00	0.05	0.00
kallen rauta	0.12	0.00	0.00	0.35	0.42
redis	0.00	0.00	0.00	0.12	0.56
postgresql	0.00	0.00	0.00	1.00	0.43
⋮	⋮	⋮	⋮	⋮	⋮

Taulukko 3.3. *Esimerkki aihesanojen välisestä suhdematriisista. Riveillä ovat tietoverkon solmuja vastaavat aihetermit ja sarakkeilla ovat Flowdock-datasta löytyvät tokenit*

Kun suhdematriisia rakennetaan, alustetaan sen kaikki arvot aluksi nolliksi. Tämän jälkeen etsitään solut, joissa sarakkeilla olevat tokenit vastaavat täysin riveillä olevia tietoverkon termejä ja asetetaan ne ykkösiksi kuvaamaan vahvinta mahdollisinta suhdetta termien välillä. Tavoitteena on, että jokaisella solmulla on joko suora suhde johonkin teks-

tissä esiintyvään tokeniin suhdematriisin kautta tai sitten se on hierarkiassa ylempänä tämänlaisen suhteen omaavaa solmua. Matriisiin voidaan lisätä alkuarvoja kunnes tämä ehto toteutuu riittävän hyvin.

Keskusteluihin osallistuvat työntekijät käyttävät samasta asiasta useita eri termejä, esimerkiksi tietokantasovelluksesta PostgreSQL puhutaan usein käyttämällä lyhennettyä nimeä Postgres tai vielä lyhyempää myös komentorivikomentona toimivaa nimeä “psql”. Nämä synonyymit sekä muut suhdematriisiin ennen algoritmin ajamista tehdyt manuaaliset lisäykset on listattu liitteessä B.

3.5.1 Bayes-verkon satunnaismuuttujien tilat ja niiden tulkinta

Tulosten tarkastelua varten algoritmissa käytettävä osaamisgraafeihin pohjautuva malli M muutetaan Bayes-verkkomalliksi M_B. Bayes-verkko toteutetaan diskreeteillä satunnaismuuttujilla, joiden väliset suhteet jodetaan mallissa M solmuille ja solmujen välisille yhteyksille saaduista sekä asetetuista arvoista β, t ja e.

Kukin Bayes-verkon solmua v_{ij} vastaava satunnaismuuttuja X_{ij} voi ottaa määrätyn määrän erilaisia arvoja eli tiloja. Kun tiloja on $N \in \mathbb{Z}_+$ kappaletta, määritellään satunnaismuuttuja $X_{ij} \in \{x_1, x_2, \ldots x_N\}$. Tilojen lukumäärä määrittää työntekijän taitotason mahdollisen esittämistarkkuuden. Liian suuri N tekee eri tilojen välisestä erosta hankalan tulkita ja liian pieni N vähentää Bayes-verkosta saatavan informaation määrää myöhemmissä vaiheissa. Tässä työssä valitaan $N = 10$, jolloin satunnaismuuttujan arvo x_1 tarkoittaa solmua vastaavan termin edustaman aihekategorian hallinnan puuttumista kokonaan ja arvo x_{10} tarkoittaa vastaavasti solmua vastaavan termin edustaman aihekategorian parasta mahdollista hallintaa.

Koska mallin M solmujen osaamisarvot on rajattu lukuvälille $[0, 1]$, voidaan myös satunnaismuuttujan X_{ij} tiloille x_n antaa niitä vastaavat lukuarvot siten, että $x_n = (n - \frac{1}{2})\frac{1}{N}$. Tällöin $x_n \in [0, 1]$, kun $n \in \mathbb{Z}_+,\ n \leq N$.

Tilojen x_n todennäköisyydet johdetaan funktiosta $P_{node}(x_n)$. Tämän funktion pitää tuottaa tiloille x_n validi todennäköisyysjakauma, eli todennäköisyyksien summan $\sum_{i=1}^{N} P_{node}(x_n)$ pitää olla 1. Lisäksi funktion pitää muistuttaa normaalijakaumaa ja laskettaessa sen arvoa satunnaismuuttujalle X_{ij} sen tulee riippua solmulle v_{ij} asetetusta osaamisarvosta sekä sen virheestä.

Funktion $P_{node}(x_n)$ muodostamiseksi on yksinkertaisinta tulkita mallin M solmulle v_{ij} asetettua osaamisarvoa t_{ij} odotusarvona μ_{ij} ja vastaavaa virhettä e_{ij} varianssina σ_{ij}^2. Eli

$$t = \mu, \quad e = \sigma^2.$$

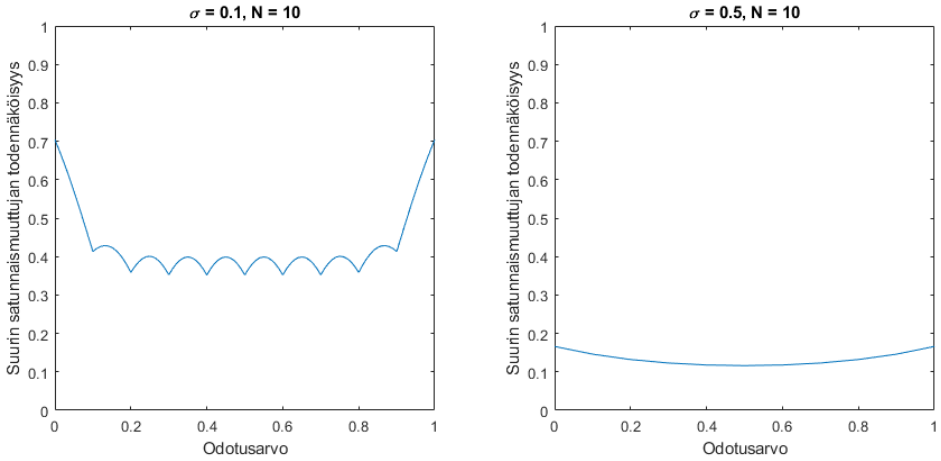
Tällöin funktion $P_{node}(x_n)$ lähtökohdaksi voidaan ottaa suoraan normaalijakauma. Ongelmaksi tässä muodostuu normaalijakauman ääretön määrittelyalue, jolloin osa jakaumas-

ta menee aina arvoalueen $[0, 1]$ ulkopuolelle. Tämä aiheuttaa ongelmia erityisesti silloin, kun odotusarvo on lähellä sallitun arvoalueen laitoja. Ongelmia aiheuttavat myös pienet varianssin arvot, jolloin suurin osa jakaumasta voi osua kahden tilaa x_n vastaavan numeerisen arvon väliin.

Tässä työssä valitaan funktioksi $P_{node}(x_n)$ kaava

$$P_{node}(x_n) = \frac{\varphi_{\mu,\sigma^2}(x_n)}{\sum_{i=1}^{N} \varphi_{\mu,\sigma^2}(x_i)}, \tag{3.1}$$

missä φ_{μ,σ^2} on kaavan (2.4) mukainen normaalijakauman tiheysfunktio solmua vastaavista arvoista johdetulla odotusarvolla $\mu \in [0, 1]$ ja varianssilla σ^2. Funktio (3.1) on käytännössä yksi Riemannin summan osista. Tarkasteltava lukualue $[0, 1]$ jaetaan N:ään yhtä pitkään osaväliin ja funktio $P_{node}(x_n)$ kertoo yhdelle osavälille sijoittuvan suorakulman pinta-alan. Mainitut ongelmat arvoalueen laidoilla olevista odotusarvoista sekä pienistä variansseista vaikuttavat funktion tuottamiin arvoihin. Todennäköisyysjakauman muoto vääristyy siis hieman odotusarvon lähestyessä lukualueen $[0, 1]$ laitoja sekä varianssin lähestyessä nollaa. Tätä on havainnoitu kuvassa 3.4.



Kuva 3.4. *Suurimman satunnaismuuttujan saaman todennäköisyyden muuttuminen odotusarvon funktiona kahdella eri varianssin arvolla.*

Vääristymän olemassaolo tunnustetaan, mutta sen vaikutus todetaan riittävän pieneksi. Tilojen todennäköisyyksien johtamista normaalijakauman osittaisesta integraatiosta palkki-integraation sijaan kokeiltiin myös, mutta erot lopputuloksessa olivat minimaalisia.

3.6 Osaamisgraafin solmujen välinen suhde

Jokaisen mallin M solmun v_{ij} ja sen vanhemman $v_{E,ij}$ välisellä kytköksellä on painoarvo $\beta_{E,ij} \in [0, 1]$, joka kuvaa solmun $v_{E,ij}$ edustaman aihekategorian osaamisen vaikutusta solmun v_{ij} edustaman aihekategorian osaamiseen. Arvo 1 kuvaa täydellistä vaikutusta ja arvo 0 kuvaa olematonta vaikutusta. Jos kaikilla solmun v_{ij} vanhemmilla on paino 1, voidaan työntekijän tieto aiheesta v_{ij} päätellä täysin solmun vanhemmista havaitun tiedon

pohjalta. Jos taas painoarvo jollekin vanhempi-lapsi-suhteelle on 0, voidaan koko suhde poistaa merkityksettömänä. Painoarvot voivat teoriassa saada ääriarvot 0 ja 1, mutta käytännössä näitä arvoja tulee pyrkiä välttämään.

Solmun v_{ij} osaamisarvo t_{ij} pitää siis pystyä johtamaan jollakin kaavalla, joka huomioi sen vanhempien osaamisarvot, pitää osaamisarvon reaalilukuna arvovälillä $[0, 1]$ ja noudattaa kuvaillun kaltaista käyttäytymistä painoarvojen ääriarvoilla 0 ja 1. Tässä työssä valitaan kaava

$$t_{ij} = (1 - \beta_{E,ij})t_{0,ij} + \beta_{E,ij}t_{E,ij} \tag{3.2}$$

missä $t_{0,ij} \in [0, 1]$ on solmun v_{ij} osaamisarvon pohja-arvo, $\beta_{E,ij} \in [0, 1]$ on vanhemman solmun $v_{E,ij}$ ja solmun v_{ij} välisen suhteen voimakkuus ja $t_{E,ij} \in [0, 1]$ on solmun $v_{E,ij}$ osaamisarvo. Tällöin myös solmun v_{ij} osaamisarvo t_{ij} on lukuvälillä $[0, 1]$. Jos solmulla v_{ij} ei ole lainkaan vanhempia, asetetaan $\beta_{E,ij} = 0$ ja solmun v_{ij} osaamisarvoksi tulee sama kuin osaamisarvon pohja-arvo. Jos solmun $v_{E,ij}$ satunnaismuuttuja saa varmuudella arvon x, asetetaan $t_{E,ij} = x$.

Vastaavasti solmun v_{ij} osaamisarvon epävarmuutta kuvaava epävarmuusarvo e_{ij} lasketaan

$$e^2_{ij} = (1 - \beta_{E,ij})e^2_{0,ij} + \beta_{E,ij}e^2_{E,ij} \tag{3.3}$$

missä $e^2_{0,ij}$ on solmun v_{ij} epävarmuusarvon pohja-arvo ja $e^2_{E,ij}$ on vanhemman solmun $v_{E,ij}$ epävarmuusarvo. Jos solmulla v_{ij} ei ole lainkaan vanhempia, asetetaan $\beta_{E,ij} = 0$ ja solmun v_{ij} epävarmuusarvoksi tulee sama kuin osaamisarvon pohja-arvo. Jos solmun $v_{E,ij}$ satunnaismuuttuja saa varmuudella arvon x, asetetaan kyseiselle solmulle $e^2_{E,ij} = 0$.

Painoarvot $\beta_{E,ij}$ asetetaan arvoon 0,4 tietyillä poikkeuksilla, jotka näkyvät liitteessä C. Useimmille verkon kytköksille arvo 0,5 on riittävän hyvä arvio ja sen tarkentaminen voi olla myös joidenkin solmujen vastaamien aihekategorioiden laajuuden takia hyvin hankalaa.

4 ALGORITMI

4.1 Työntekijän aihetermeille altistumisen laskeminen

Muodostetaan matriisit O_K ja O_F kuvaamaan työntekijöiden osallistumista keskusteluihin sekä keskusteluhuoneisiin. Tieto on binäärinen, sillä vaikka keskusteluun osallistuminen useammalla viestillä voi ilmaista aktiivisempaa osallistumista ja siten suurempaa altistumista keskustelun käsittelemille aiheille, tätä yhteyttä on vaikea perustella verkossa käytävän keskustelun kaoottisuuden vuoksi. Jotkin henkilöt kirjoittavat pitkiä viestejä ja toiset suosivat viestien jakamista useampiin eri paloihin. Tarkastelun yksinkertaistamiseksi oletetaan kaikkien keskusteluun osallistuneiden altistuvan keskustelulle yhtä paljon. Matriisit O_K ja O_F määritellään tällöin

$$\begin{aligned} O_K &\in \{0,1\}^{||K||\times||W||} \\ [O_K]_{i,j} &= \begin{cases} 1 & \text{jos henkilö } w_j \text{ on osallistunut keskusteluun } k_i \\ 0 & \text{jos henkilö } w_j \text{ ei ole osallistunut keskusteluun } k_i \end{cases} \\ O_F &\in \{0,1\}^{||F||\times||W||} \\ [O_F]_{i,j} &= \begin{cases} 1 & \text{jos henkilö } w_j \text{ on osallistunut keskusteluhuoneeseen } f_i \\ 0 & \text{jos henkilö } w_j \text{ ei ole osallistunut keskusteluhuoneeseen } f_i \end{cases} \end{aligned} \quad . \tag{4.1}$$

Määritellään myös matriisit Y_K ja Y_F kuvaamaan tokenien t_i esiintymistä keskusteluissa k_j ja keskusteluhuoneissa f_j. Hyödynnetään kaavaa (2.5) ja määritellään matriisit

$$\begin{aligned} Y_K &\in [0,\infty)^{||T||\times||K||} \\ [Y_K]_{i,j} &= \text{tfidf}(t_i, k_j, k) \\ Y_F &\in [0,\infty)^{||T||\times||F||} \\ [Y_F]_{i,j} &= \text{tfidf}(t_i, f_j, f) \end{aligned} \quad . \tag{4.2}$$

Näiden matriisien avulla voidaan laskea työntekijöiden suhde tokeneihin keskusteluihin

ja keskusteluhuoneisiin osallistumisen kautta

$$\begin{aligned} C_K &\in [0,\infty)^{||T||\times||W||}, \quad C_K = Y_K O_K \\ C_F &\in [0,\infty)^{||T||\times||W||}, \quad C_F = Y_F O_F \end{aligned}. \tag{4.3}$$

Käyttämällä kerrointa $c_f \in [0,1]$ keskusteluhuoneisiin osallistumisen vaikutuksen painottamiseen sekä matriisia B ilmaisemaan Bayes-verkon termien suhdetta tokeneihin

$$\begin{aligned} B &\in [0,1]^{||A||\times||T||} \\ [B]_{i,j} &= \text{aihetermin } a_i \text{ suhde tokeniin } t_j \text{ asteikolla } [0,1] \end{aligned} \tag{4.4}$$

voidaan laskea työtekijöiden W suhde aihetermeihin A

$$U \in [0,\infty)^{||A||\times||W||}, \quad C = C_K + c_f C_F, \quad U = BC. \tag{4.5}$$

Matriisille B annetaan alussa arvot siten, että saman tekstin sisältävien tokenien ja aihetermien keskinäinen suhdearvo 1 ja lisäksi matriisiin asetetaan liitteessä B määritellyt esimääritetyt suhdearvot.

Keskusteluhuoneisiin osallistumista kuvaava matriisi C_F on sitä tärkeämpi, mitä enemmän keskusteluhuoneita on ja mitä erilaisempi niiden välinen sisältö on. Kuten alaluvussa 3.2.2 on mainittu, keskusteluhuoneiden viestimäärät ovat eksponentiaalisesti jakautuneita ja joillekin osaamisalueille ei ole ollenkaan dedikoitua keskusteluhuonetta. Tästä syystä kerroin c_f asetetaan hyvin pieneksi, arvoon $c_f = 0.05$. Isolla muuttujan c_f arvolla algoritmi painottaisi enemmän laajempaa keskusteluhuonekontekstia, mikä vaatisi tasaisemmin eri huoneisiin jakautuneen viestimassan sekä työn kannalta edullisemat huoneiden käsittelemät aihealueet.

Kerroin c_f voisi olla myös vektori, jolloin voitaisiin paremmin säätää yksittäisten keskusteluhuoneiden vaikutusta tulokseen. Yleisille tiedotuskanaville voisi antaa pienemmän kertoimen ja pienemmän aihealueen kattaville keskusteluhuoneille isomman. Jos kerrointa ei haluta asettaa käsin eri keskustelukanaville voisi sen arvon määrittää käänteisesti verrannolliseksi keskusteluhuoneiden viestimäärään nähden. Tässä työssä c_f pidetään vakiokertoimena yksinkertaisemman toteutuksen testaamiseksi sekä siksi, että TF-IDF-kaava (2.5) sisältää jo käänteisesti keskustelukanavan viestimäärälle verrannollisen kertoimen.

4.1.1 Altistumisarvojen tulkitseminen

4.1.2 Osaamisarvon laskeminen

Koska matriisin U arvot kuvaavat suoraan työntekijöiden arvioitua suhdetta aihetermeihin, voidaan sen arviot u_{ji} rinnastaa työntekijän i osaamisgraafin r_i termiä a_j edustavan solmun v_{ij} osaamisarvon t_{ij} pohja-arvoon $t_{0,ij}$, joka myös kuvaa samaa suhdetta. Arvot u_{ji} eivät kuitenkaan täytä suoraan osaamisarvolle asetettua vaatimusta, jonka mukaan arvon täytyy asettua välille $[0, 1]$.

Arvon u_{ji} muuttaminen arvoksi $t_{0,ij}$ on hankalaa tehdä oikein, sillä u_{ji} kuvaa työntekijän alistumista termille, kun taas $t_{0,ij}$ kuvaa työntekijän suoraan lähtödatasta johdettavaa osaamista termiin liittyvissä tietotaidoissa. Vaikka oletettaisiin datan pohjalta havainnoidun altistumisen vastaavan täysin henkilön kokemusta termiin liittyen, kokemuksen ja osaamisen suhde on monimutkainen ja vahvasti henkilökohtaisista ominaisuuksista riippuva yhtälö [16]. Tehdään aluksi yksinkertainen oletus siitä, että korkeampi altistuminen ilmaisee suoraan korkeampaa osaamista. Käytettävissä olevan datan rajallisuuden vuoksi tämä tuntuu olevan ainoa järkevä oletus.

Arvojen numeerisen suhteen tarkempi määrittelemiseen on hieman enemmän vaihtoehtoja. Yksinkertaisin mahdollinen ratkaisu olisi jakaa jokainen arvo u_{ji} matriisin U suurimmalla arvolla, mutta tämä korostaa poikkeuksellisen suurien arvojen merkitystä. Suurin arvo on testeissä teknologiataitojen juurisolmuna toimivalla solmulla “tech”, jolla on kertoimella $c_f = 0.05$ suurin muuttujan u_{ji} arvo 83817,11 ja kertoimella $c_f = 0.0$ suurin arvo 81950,77. Tätä voi verrata esimerkiksi kertoimella $c_f = 0.05$ suurimman arvon 913,04 saavaan termiin “design” tai samalla kertoimella suurimman arvon 256.07 saavaan termiin “video production”. Lineaarinen skaalaus tarkoittaisi sitä, että paras teknologian parissa työskentelevä työntekijä olisi noin kahdeksan kertaa parasta suunnittelijaa osaavampi aihealueessaan. Työssä käytettävän datan tarjonneen yrityksen kontekstissa tämä ero on liian iso.

Toinen vaihtoehto numeerisen suhteen määrittämiseen olisi jakaa jokainen matriisin U rivi sen rivin maksimiarvolla, jos rivillä on ainakin yksi nollasta poikkeava arvo. Tällöin eri tokenien painotuksessa tapahtuvat virheet tasoittuisivat hyvin, mutta sivuvaikutuksena jokaiselle taidolle tulisi myös ainakin yksi näennäinen ekspertti. Tämä ei sovi yhteen tutkimuskysymyksien kanssa, sillä tarkoituksena on selvittää datasta saatavaa tietoa yrityksen kokonaistietomäärästä. On paljon mielekkäämpää olettaa, ettei harvoin mainituista asioista ole yrityksen sisällä juurikaan tietoa.

Valitaan muuttujien u_{ji} sekä $t_{0,ij}$ suhteeksi näennäisesti hyvin käyttäytyvä yhtälö

$$t_{0,ij} = \frac{u_{ji}}{u_{ji} + a_c} = 1 - \frac{a_c}{u_{ji} + a_c}, \tag{4.6}$$

missä $a_c > 0$ on mielivaltainen funktion jyrkkyyttä säätävä reaaliparametri. Arvon u_{ji}

kasvamisen merkitys pienenee lähestyttäessä äärettömyyttä ja arvo $t_{0,ij}$ rajautuu välille $[0, 1]$ kun $u_{ji} \geq 0$.

Parametrin a_c arvo riippuu käytettävästä datasta sekä datan pohjalta laskettujen matriisin U arvojen suuruudesta. Sen tehtävä on säätää eri osaamistasojen välistä etäisyyttä tulosten tarkastelun helpottamiseksi. Huomioitavaa on se, että algoritmin myöhemmässä vaiheessa matriisia B muutetaan ja altistumisarvot lasketaan uudestaan. Tämän seurauksena altistumisarvot kasvavat ensimmäisen iteraation jälkeen merkittävästi ja arvon a_c pitää olla myös merkittävästi suurempi. Työssä kokeiltiin aluksi arvoa $a_c = 50$, mikä tuotti ensimmäiselle iteraatiolle näennäisesti järkevästi jakautuneita arvoja, mutta myöhemmillä iteraatioilla lasketut osaamisarvojen pohja-arvot $t_{0,ij}$ menivät kaikki epärealistisen lähelle lukua 1. Arvo $a_c = 500$ tuotti näennäisesti parempia tuloksia. Termille “tech” tulee suurimmaksi muuttujan $t_{0,ij}$ arvoksi 0,99, termille “design” 0,65 ja termille “video production” 0,34.

4.1.3 Epävarmuusarvon laskeminen

Työntekijän i osaamisgraafin r_i termiä a_j vastaavan solmun v_{ij} epävarmuusarvon e_{ij} päivittäminen on valitun algoritmin ja datan pohjalta hyvin hankalaa siksi, ettei käytössä olevasta datasta ole ilman monimutkaisempaa luonnollisen kielen tunnistusta saatavilla mitään konkreettisia vastatodisteita. Termille altistumisen puute voi käytännössä ilmaista joko työntekijän täydellistä tietämättömyyttä asiasta tai täydellistä epävarmuutta työntekijän osaamisen tiedosta. Aikaisemmin tehdyn oletuksen mukaan korkeampi altistuminen ilmaisee suoraan korkeampaa osaamista, mikä tarkoittaa vastaavasti todisteiden täydellisen puuttumisen tulkitsemista henkilön varmaksi osaamattomuudeksi.

Oletusta voidaan kuitenkin myös täydentää toteamalla, että henkilöstä saadun tiedon varmuus on verrattavissa henkilön kokonaisosallistumiseen. Tämä oletus tarkoittaa sitä, että enemmän keskusteluihin osallistuvan henkilön osallistumisesta voidaan varmemmin päätellä henkilön osaamisesta. Käytännössä on myös tapauksia, joissa henkilö haluaa osallistua keskusteluihin ainoastaan jakaessaan tärkeää tietoa, jolloin kyseisen henkilön osaaminen voitaisiin päätellä suoraan muutaman viestin perusteella. Futurice-yrityksen kulttuuri kannustaa kuitenkin runsaaseen sekä avoimeen kommunikaatioon, joka sisältää myös keskustelua työhön liittymättömistä asioista. Tässä työssä keskimääräinen osallistumisaste tarkasteltujen työntekijöiden kesken on 270 keskustelua.

Algoritmin edetessä matriisi U ja sen sisältämät altistumisarvot u_{ji} lasketaan uudestaan useampaan kertaan eri matriisin B arvoilla. Tämän prosessin aiheuttama arvon u_{ji} vaihtelu voidaan ottaa lähtökohdaksi varianssin laskemiselle, sillä termit joiden satunnaismuuttujien varianssit vaihtelevat paljon saavat enemmän vaikutteita graafissa niihin kytketyistä termeistä ja osaamisarvo pohjautuu täten enemmän termien väliseen yhteyteen kuin tekstianalyysiin. Poikkeavan arvoalueensa vuoksi altistumisarvo on kuitenkin huono lähtöarvo epävarmuusarvon laskemiselle, joten käytetään kaavan (4.6) avulla laskettuja osaamisarvon pohja-arvoja $t_{0,ij}$.

Epävarmuusarvon pohja-arvo $e_{0,ij}$ lasketaan sitä vastaavan osaamisarvon pohja-arvon $t_{0,ij}$ vaihtelemisen sekä henkilön i kaikkien osaamisarvojen pohja-arvojen summan $t_{tot,i}$ perusteella. Tarkastellaan yhden työntekijän i yhtä termiä a_j ja merkitään algoritmin nykyisellä sekä sitä aikaisemmilla iteraatioilla b saatuja arvoja $t_{0,ij,b}$ sekä näiden iteraatioiden kokonaismäärää $N_{iter} \in \mathbb{Z}_+$.

Huolimatta oletuksesta todisteiden puutteen ja työntekijän matalan osaamistason välillä, monet tekijät voivat aiheuttaa tulosten vääristymistä erityisesti matalan osaamistason suuntaan. Kaikki asiasta puhuvat henkilöt eivät ole siinä eksperttejä ja jotkin tokenit voidaan vahingossa yhdistää niihin liittymättömiin termeihin. Epävarmuusarvoa laskettaessa epävarmuutta painotetaan lisäämällä osaamisarvohistoriaan ääriarvot 0 ja 1. Tällöin arvojen kokonaismäärä varianssia laskettaessa on $N_{iter} + 2$.

Epävarmuuden laskemiseen käytetään yleisiä keskiarvon ja varianssin kaavoja [17, s. 1] pienillä lisäyksillä. Varianssin kaavaan on tehty Besselin korjaus [18], sillä ilman sitä varianssin arvot jäivät jatkuvasti näennäisesti liian pieniksi. Varianssia kasvatetaan myös erityisellä kertoimella työntekijän i osaamisarvon pohja-arvojen summan $t_{tot,i}$ ollessa suurinta vastaavaa arvoa t_{max} pienempi. Työssä kokeiltiin kerrointa $\frac{t_{tot,i}+t_{max}/2}{t_{tot,i}}$, mutta se tuotti pienille osaamisarvojen summille todella isoja epävarmuusarvoja. Suurimmat epävarmuusarvot menivät yli sadassa, mikä on osaamisarvon arvoalueeseen nähden huono asia. Tätä arvojen valtavaa kasvamista tasoitetaan logaritmilla, mutta logaritmifunktion tuottamia pieniä arvoja pitää kasvattaa lisäämällä tulokseen vakio. Kokeilemisen jälkeen valitaan kertoimeksi $1+\log_e \frac{t_{max}}{t_{tot,i}}$ tasoittamaan kasvua ja varianssin pohja-arvon kaavaksi

$$\begin{aligned} t_{max} &= \max\{t_{tot,i} \mid j = 1..||W||\} \\ \overline{t_{0,ij}} &= \frac{1 + \sum_{b=1}^{N_{iter}} t_{0,ij,b}}{N_{iter} + 2} \\ e_{0,ij} &= \left[1 + \log_e \frac{t_{max}}{t_{tot,i}}\right] \frac{(0 - \overline{t_{0,ij}})^2 + (1 - \overline{t_{0,ij}})^2 + \sum_{b=1}^{n_{iter}} (t_{0,ij,b} - \overline{t_{0,ij}})^2}{N_{iter} + 1} \end{aligned}, \tag{4.7}$$

missä $t_{tot,i} > 0$. Jos $t_{tot,i} = 0$, ei työntekijä ole altistunut millekään termille lainkaan ja asetetaan epävarmuusarvon pohja-arvoksi $e_{0,ij} = 1$ ilmaisemaan osaamisarvon arvoalueeseen $[0, 1]$ nähden merkittävää epävarmuutta.

Valittu kaava ei toimi hyvin sellaisille solmuille, joiden kaikkien iteraatioiden osaamisarvojen pohja-arvot ovat nollia, mutta $t_{tot,i} \neq 0$. Tällöin epävarmuusarvo painuu myös arvoon 0, vaikka todisteiden puute ei välttämättä tarkoita täyttä varmuutta tiedon puutteesta. Pyrkimyksenä on kuitenkin matriisin B alkuarvoa muuttamalla vähentää tämänlaisia solmuja lisäämällä käsin heikkojen kytkentöjen määrää ja siten aktivoimalla graafin eri solmuja edes hieman.

4.2 Termimatriisin parantaminen kontekstin pohjalta

Termimatriisi B sisältää aluksi vain vähän tietoa Bayes-verkkojen termien sekä tekstissä esiintyvien tokenien suhteesta. Tätä tietoa voidaan lisätä työntekijöiden altistumistiedon sekä Bayes-verkon suhteiden avulla.

Jos henkilö on altistunut termille a_y arvolla u_y ja termi a_x on termin a_y vanhempi suhteella $\beta_{E,y}$, voidaan henkilön katsoa altistuneen myös termille a_x arvolla $u_x = u_y\beta_{E,y}$. Termin a_x lapsien määrän n kasvaessa niiden nettovaikutuksen vanhempaan termiin ei pitäisi kasvaa merkittävästi, joten jaetaan vaikutusarvo arvolla n. Kun otetaan altistumismatriisista U termille a_x altistumista kuvaava rivivektori $\overline{\boldsymbol{u}}_x$ ja sen lapsitermeille $a_{y,1}, a_{y,2}, \ldots, a_{y,n}$ altistumista kuvaavat rivivektorit $\overline{\boldsymbol{u}}_{y,1}, \overline{\boldsymbol{u}}_{y,2}, \ldots, \overline{\boldsymbol{u}}_{y,n}$, voidaan kaavaa (3.2) mukaillen laskea rivivektorille $\overline{\boldsymbol{u}}_x$ uusi arvo

$$\overline{\boldsymbol{u}}_{x,new} = \overline{\boldsymbol{u}}_x + \frac{1}{n}\sum_{i=1}^{n} \beta_{E,y,i}\overline{\boldsymbol{u}}_{y,i,new}, \tag{4.8}$$

missä $\beta_{E,y,i}$ on termin a_x suhteen voimakkuus sen lapsitermiin $a_{y,i}$ ja $\overline{\boldsymbol{u}}_{y,i,new}$ on lapsitermille $a_{y,i}$ laskettu vastaava uusi rivivektorin arvo. Jos termiä edustavalla Bayes-verkon solmulla ei ole lainkaan lapsia, asetetaan vain $\overline{\boldsymbol{u}}_{x,new} = \overline{\boldsymbol{u}}_x$.

Uusista rivivektoreista kootun matriisi U_{new} kuvaa altistumisarvoja silloin, kun Bayes-verkon suhteet on otettu huomioon. Tämän matriisin sisältämän tiedon pohjalta voidaan lähteä parantamaan tietoa tokenien ja termien suhteesta, eli matriisista B.

Tehdään oletus, jonka mukaan työntekijä osallistuu pääasiassa sellaisiin keskusteluihin ja keskusteluhuoneisiin, jotka sisältävät kyseistä henkilöä kiinnostavia aiheita. Jos työntekijä ei vielä hallitse asiaa, hän on vähintäänkin kiinnostunut asiasta. Tämä tarkoittaa sitä, että kun aihetermin a_j hyvin tunteva työntekijä i osallistuu tokenin t_k sisältävään keskusteluun tai keskusteluhuoneeseen, voidaan tokenin ja termin välille päätellä yhteys. Tätä yhteyttä voidaan kuvata matriisien U_{new} ja C alkioiden u_{ji} ja c_{ik} laskulla $b_{jk} = u_{ji}c_{ki}$. Tästä nähdään, että kaikkien työntekijöiden kautta luotujen token-termi-yhteyksien summat saadaan matriisitulolla $B'_{new} = U_{new}C^T$.

Saatu matriisi B'_{new} ei kuitenkaan noudata matriisille B asetettua arvoaluevaatimusta $B \in [0,1]^{||R||\times||T||}$. Korjauksen tekemiseen on useita eri keinoja, mutta tässä oletetaan matriisin B'_{new} edustavan termien suhdetta tokeneihin karkeasti lineaarisella skaalalla, jolloin uusi matriisi B saadaan laskettua yksinkertaisesti

$$B_{new} = \frac{B'_{new}}{\max\{b \in B'_{new}\}} = \frac{U_{new}C^T}{\max\{b \in U_{new}C^T\}}. \tag{4.9}$$

Matriisin B_{new} avulla lasketaan uudelleen työntekijöiden suhteet aihetermeihin käyttäen kaavaa (4.5) ja prosessia toistetaan $N_{iter} \in \mathbb{Z}_+$ kertaa. Tässä työssä on valittu kokeilemalla arvo $N_{iter} = 3$. Tuloksia tarkastellessa tullaan huomaamaan, että valitulla algorit-

milla tulokset eivät muutu merkittävästi toisen iteraation jälkeen. Algoritmi pysäytetään kolmannen iteraation jälkeen lähinnä varianssin pienenemisen pysäyttämiseksi.

4.3 Todennäköisyystaulukoiden muodostaminen

Algoritmin lopuksi kerätty tieto muutetaan Bayes-verkon solmujen todennäköisyystaulukoiksi. Tämä tehdään jokaisen työntekijän i verkon r_i jokaiselle solmulle v_{ij} erikseen. Lisäksi jokaisen solmun satunnaismuuttujan X_{ij} mahdollisten tilojen x_n todennäköisyydet pitää laskea erikseen kaikille mahdollisille solmun vanhempia $v_{E,ij}$ vastaavien satunnaismuuttujien tilayhdistelmille.

Satunnaismuuttujien tilojen todennäköisyydet lasketaan kaavalla (3.1), joka riippuu solmun v_{ij} osaamisarvosta $\mu_{ij} = t_{ij}$ sekä sen epävarmuudesta $\sigma_{ij}^2 = e_{ij}^2$. Nämä arvot taas riippuvat solmun mahdollisen vanhemman $v_{E,ij}$ vastaavista arvoista, joten johdetut arvot pitää laskea erikseen jokaiselle mahdollisen vanhemman satunnaismuuttujan $X_{E,ij}$ mahdolliselle tilalle. Kun mahdollisen vanhemman solmun satunnaismuuttuja on saanut jonkin arvon x_n, tehdään korvaus $t_{E,ij} = x_n$ ja käytetään kaavaa (3.2) solmun v_{ij} osaamisarvon laskemiseksi. Tällöin mahdollisen vanhemman solmun epävarmuusarvo asetetaan myös nollaksi $\sigma_{E,ij} = 0$ ja solmun v_{ij} epävarmuusarvo lasketaan käyttäen kaavaa (3.3).

Kaikilla tässä työssä Bayes-verkoissa esiintyvillä solmuilla on vain enintään yksi vanhempi solmu. Yhden vanhemman omaavan solmun todennäköisyystaulukko muodostetaan kuvaillulla tavalla siten, että solmun v_{ij} todennäköisyys satunnaismuuttujan arvolle x_n solmun vanhemman saadessa tilan x_m on

$$P_{ij}(x_n \mid x_m) = P_{node}(x_n), \quad \mu_{ij} = (1 - \beta_{E,ij})t_{0,ij} + \beta_{E,ij}x_m, \quad \sigma_{ij}^2 = (1 - \beta_{E,ij})e_{0,ij}^2,$$

missä $\beta_{E,ij}$ on solmun v_{ij} ja sen vanhemman $v_{E,ij}$ välisen kytköksen painoarvo.

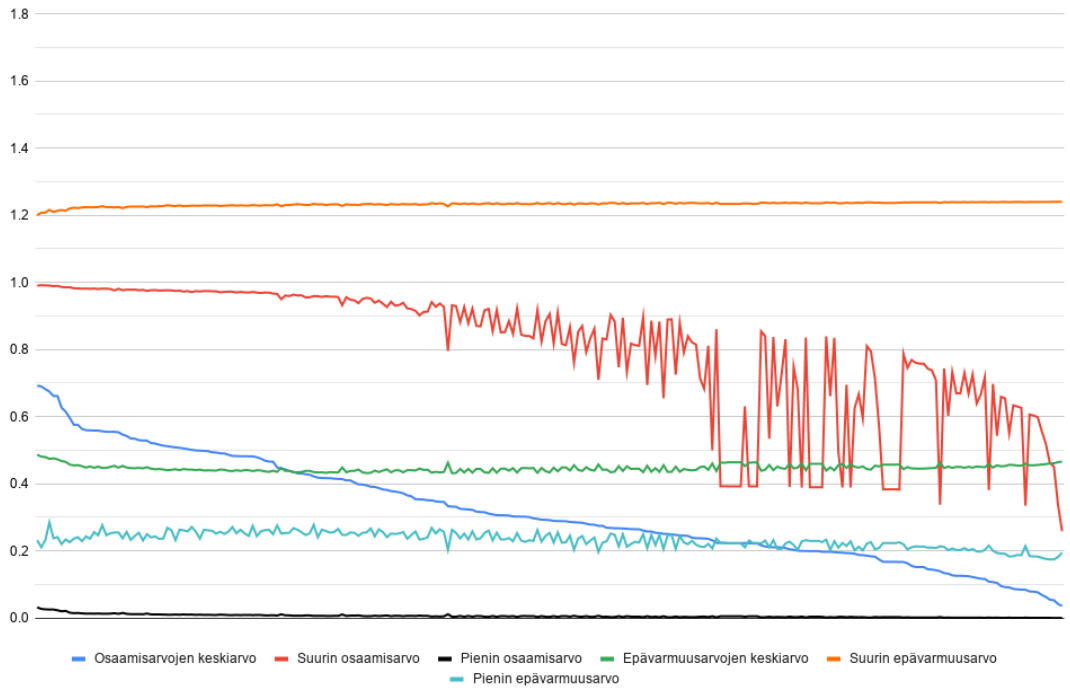
4.4 Työntekijöiltä saadun lisätiedon huomioiminen

Työntekijöitä pyydetään arvioimaan omaa osaamistaan yksittäisiin termeihin liittyen asteikolla 1-10, jonka jälkeen työntekijän Bayes-verkon vastaava satunnaismuuttuja asetetaan työntekijän valitsemaan tilaan. Tämän jälkeen työntekijän Bayes-verkon muut satunnaismuuttujat voidaan päivittää alaluvussa 2.2.1 kuvatulla tavalla.

5 TULOKSET

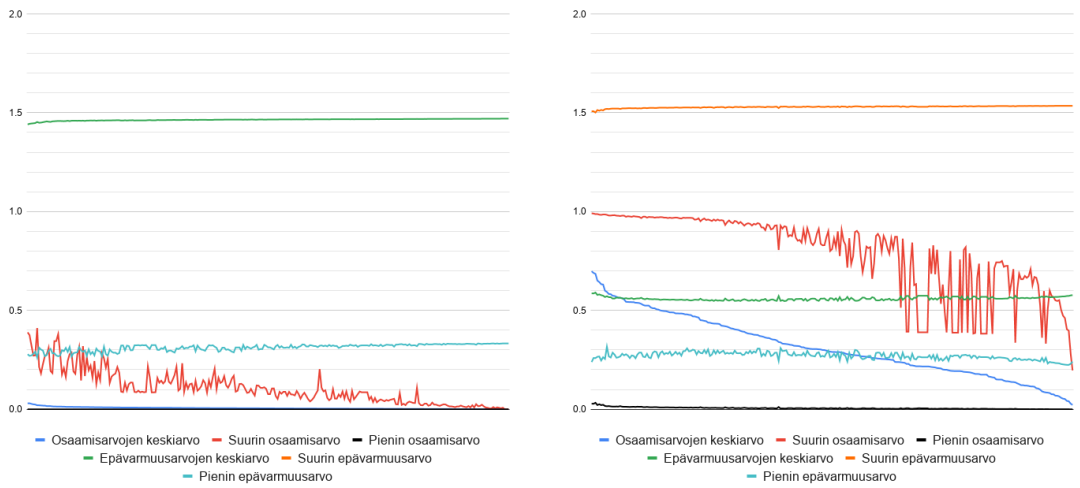
5.1 Yleisen tietomäärän arviointi

Algoritmin tuottamien mallin M solmujen osaamis- ja epävarmuusarvojen ominaisuuksia ennen todennäköisyystaulukoiden generointia on tarkasteltu kuvassa 5.1. Kuvasta näkyy, että termien jakauma hyvin ja huonosti hallittuihin on sopivan tasainen. Lisäksi lähes kaikille termeille löytyy ne hyvin hallitsevia työntekijöitä ja niistä täysin tietämättömiä työntekijöitä. Osaamisarvojen jakauma vastaa intuitiivisesti odotettua tilannetta, sillä termien joukossa on sekä todella yleisiä yrityksen pääasiallista toimintaa vastaavia termejä että harvinaisempia erityistietoja vastaavia termejä.



Kuva 5.1. *Osaamisgraafin termien osaamisarvojen ja epävarmuusarvojen jakautuminen työntekijöiden välillä kolmannen iteraation jälkeen. Termit on lajiteltu suurimmasta osaamisarvon keskiarvosta pienimpään.*

Epävarmuusarvo riippuu kaavassa (4.7) olevan logaritmisen kertoimen vuoksi paljon työntekijöiden aktiivisuudesta keskusteluissa, mistä syystä jotkin työntekijät saavat tasaisesti korkeita arvoja ja jotkin työntekijät tasaisesti matalia arvoja. Tämä selittää epävarmuusar-



Kuva 5.2. *Osaamisgraafin termien osaamisarvojen ja epävarmuusarvojen jakautuminen työntekijöiden välillä ensimmäisen ja toisen iteraation jälkeen. Termit on lajiteltu kuvaajakohtaisesti suurimmasta osaamisarvon keskiarvosta pienimpään. Ensimmäisen iteraation suurimmat epävarmuusarvot ovat kaikki 3,92.*

von kuvaajien tasaisuutta, mutta epävarmuusarvon keskiarvossa pitäisi silti olla paljon enemmän kasvua vähemmän hallittuihin termeihin päin edetessä. Osaamisarvojen vaihteluiden ottaminen epävarmuusarvon laskemisen pohjaksi osoittautuu odotetusti melko huonoksi ratkaisuksi, mutta valittu kaava tuottaa silti arvoja suhteellisen järkevältä arvoalueelta.

Kuvassa 5.2 on sama kuva ensimmäiselle ja toiselle algoritmin iteraatiolle. Tästä huomataan, että ensimmäisen iteraation tuottamat arvot sisältävät myöhempiin iteraatioihin verrattuna hyvin korkeita epävarmuusarvoja ja hyvin matalia osaamisarvoja, mutta toisen iteraation jälkeen tuloksissa tapahtuu lähinnä epävarmuusarvojen pienenemistä. Jos algoritmin iterointia jatkettaisiin, pienenisivät epävarmuusarvot mielivaltaisen pieniksi. Tämä ei ole kovin intuitiivista, sillä käytettävissä olevasta datasta ei ole oletettavasti mahdollista saada mielivaltaisen tarkkoja tuloksia. Algoritmin toiminta kokonaisuudessaan todetaan tässä suhteessa tarpeeksi hyväksi valituilla pienillä iteraatiomäärillä, mutta iteraatioiden kasvaessa tuloksissa tapahtuva muutos ei ole enää näennäisesti järkevää epävarmuusarvojen lähestyessä nollaa.

5.1.1 Vertailu Power-dataan

Työntekijät ovat raportoineet erilaisten asioiden parissa viettämäänsä aikaa Power-järjestelmään. Tämä on oikein hyvä tieto työntekijän osaamisesta eri asioihin liittyen, mutta manuaalisen päivittämisen vaatimuksen vuoksi tiedot voivat olla puutteellisia tai vanhentuneita. Työntekijät merkkaavat lisäksi usein vain niitä taitoja, jotka kokevat relevanteiksi omaan senhetkiseen rooliinsa. Suomalainen sovelluskehittäjä ei tästä syystä usein merkitse suomen kielen taitoaan, mahdollisia graafisen suunnittelun kokemuksiaan tai vapaa-ajalla satunnaisesti kokeilemiaan teknologioita järjestelmään, vaikka tähän olisikin

Termi	$avg(\mu)$	$max(\mu)$	$min(\mu)$	$avg(\sigma^2)$	$max(\sigma^2)$	$min(\sigma^2)$
tech	0.6920	0.9902	0.0316	0.4865	1.2009	0.2318
recruitment	0.6899	0.9914	0.0270	0.4807	1.2071	0.2108
finnish	0.6820	0.9908	0.0259	0.4796	1.2079	0.232
account management	0.6743	0.9903	0.0251	0.4741	1.2157	0.2847
sales	0.6615	0.9886	0.0253	0.4757	1.2099	0.2375
language	0.6614	0.9895	0.0228	0.4725	1.2133	0.2409
photography	0.0688	0.5570	0.0003	0.4571	1.2396	0.1792
user insights	0.0628	0.5194	0.0003	0.4584	1.2396	0.1760
video production	0.0545	0.4617	0.0003	0.4600	1.2390	0.1745
game design	0.0530	0.4510	0.0002	0.4610	1.2397	0.1745
motion design	0.0418	0.3382	0.0002	0.4640	1.2397	0.1815
ar design	0.0358	0.2585	0.0002	0.4658	1.2398	0.1945

Taulukko 5.1. *Suurimman ja pienimmän osaamisarvon keskiarvon omaavat termit ja niiden osaamisarvon sekä epävarmuusarvon statistiikat.*

mahdollisuus.

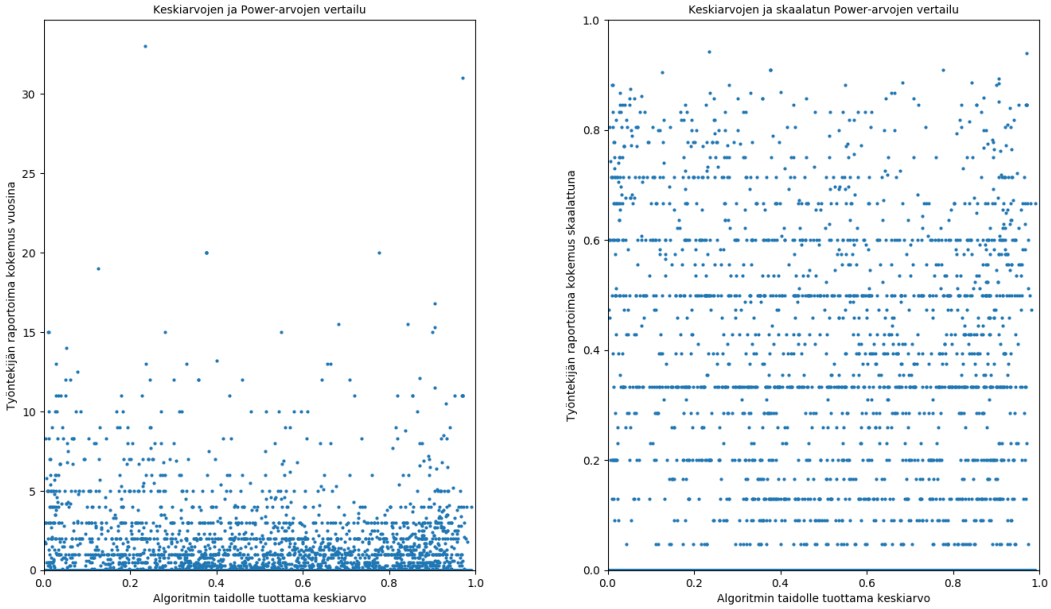
Vertailemalla Power-järjestelmän sisältämää ja algoritmin tuottamaa tietoa voidaan tutkia algoritmin onnistumista. Ennen tarkastelua pitää kuitenkin luoda yhteys Flowdock-käyttäjien ja Power-käyttäjien välille. Tämä voidaan tehdä työntekijöiden nimien perusteella, mutta käytössä olleet käyttäjälistat olivat eri ajoilta eivätkä yrityksen jatkuvan työvoiman muutoksen vuoksi sisältäneet täysin samoja työntekijöitä. Tästä syystä yhteys saatiin luotua vain 239:n työntekijän Flowdock- ja Power-tietojen välille, vaikka Flowdock-keskusteluista oli tunnistettavissa 484 uniikkia käyttäjää.

Kuvassa 5.3 on vertailtu algoritmin mallin M solmuille tuottamien osaamisarvojen suhdetta työntekijöiden itse Power-järjestelmään raportoimiin arvoihin, jotka kuvaavat työntekijän kokemuksen määrää aiheen parissa vuosina mitattuna. Kuvaajista nähdään, että algoritmin tuottamien tulosten ja Power-järjestelmässä olevan tiedon välille on hyvin hankala vetää suoraa yhteyttä. Optimaalisti pisteet asettuisivat jollekin nousevalle suoralle tai logaritmiselle käyrälle, mutta näin ei silmämääräisesti tapahdu.

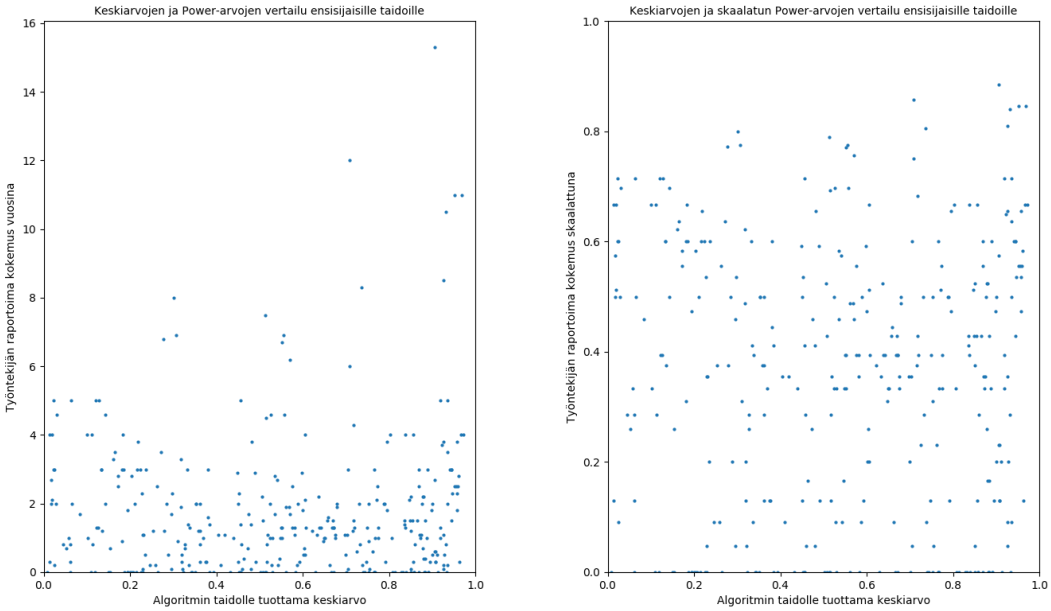
Työntekijät pystyvät Power-järjestelmässä merkkaamaan valitsemiaan taitoja ensisijaisiksi (eng. preferred), jolloin työnantaja tietää työntekijän haluavan osallistua näitä taitoja vaativiin tai näitä teknologioita sisältäviin projekteihin. Kuvassa 5.4 on tehty sama tarkastelu kuin kuvassa 5.3, mutta tarkasteluun on otettu vain työntekijöiden ensisijaiseksi merkkaamat taidot. Pistejoukko on vielä hyvin satunnaisen oloinen alle kuusi vuotta kokemusta vastaavien taitojen osalta, mutta kuuden kokemusvuoden yläpuolella korrelaatio on paljon vahvempi.

Algoritmin tuottamien tulosten sekä työntekijöiden Power-järjestelmään merkkaamien arvojen väliset näytejoukkojen statistiset korrelaatiokertoimet (eng. “sample correlation coefficient”) ovat taulussa 5.2. Tulokset vahvistavat kuvista 5.3 sekä 5.4 havaittuja tuloksia, eli algoritmin tuottamat tulokset vastaavat pääosin erittäin huonosti työntekijöiden merkkaamia arvoja. Valittu algoritmi ei siis toimi työntekijöiden osaamisalueiden tunnistamisessa.

Otin tarkempaan tarkasteluun yksittäisiä pisteitä kuvasta 5.4. Yhdellä työntekijällä oli 33



Kuva 5.3. *Algoritmin tuottamat osaamisarvot suhteutettuna työntekijän samalle taidolle merkitsemään kokemukseen. Ensimmäisessä kuvassa on työntekijän ilmoittama kokemus vuosina, kun taas toisessa on työntekijän ilmoittamaa kokemusta skaalattu kaavalla* $x/(x+2)$.



Kuva 5.4. *Algoritmin tuottamat osaamisarvot työntekijöiden ensisijaisiksi merkkaamille taidoille suhteutettuna työntekijän samalle taidolle merkitsemään kokemukseen. Ensimmäisessä kuvassa on työntekijän ilmoittama kokemus vuosina, kun taas toisessa on työntekijän ilmoittamaa kokemusta skaalattu kaavalla* $x/(x+2)$.

r_{xy}	Kaikki arvoparit	Ensisijaisten taitojen arvoparit
Korjaamaton	0.036	0.042
Skaalattu	0.063	-0.001

Taulukko 5.2. *Algoritmin tuottamien osaamisarvojen sekä työntekijöiden Power-järjestelmään merkkaamien kokemusarvojen välinen statistinen näytekorrelaatiokerroin.*

kokemusvuotta taidossa “german”, mutta algoritmi antoi sille ensimmäisen kierroksen jälkeen hyvin matalan osaamisarvon 0,0020 ja kolmannen kierroksenkin jälkeen osaamisarvo pysyi matalassa arvossa 0,2352. Tämä ei ole yllättävää, sillä ihmiset harvoin keskustelevat suoraan omasta kielitaidostaan ja analyysi keskittyi pääasiassa englanninkieliseen tekstiin.

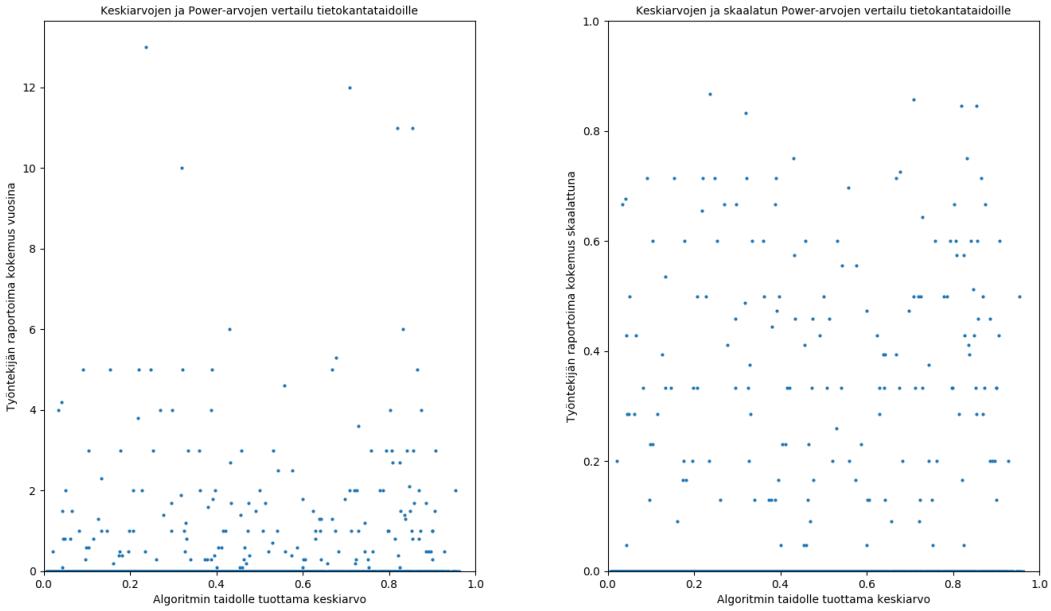
Toiselle työntekijälle algoritmi antoi hyvin korkean osaamisarvon 0,9218 termille “recruitment”, mutta kyseinen henkilö ei ole merkannut siitä yhtään kokemusta Power-järjestelmään. Kyseinen henkilö on projektijohtaja ja myynnin asiantuntija, minkä vuoksi hän on ottanut useammassakin viestissä puheeksi rekrytoinnin ja haastattelut. Haastattelut ovat osaamisgraafin hierarkiassa rekrytoinnin alla, joten haastatteluihin liittyvät viestit kasvattavat myös rekrytoinnin keskiarvoa. Yksi rekrytointia koskeva viesti on osa Futuricen sisäistä erityistiimeihin rekrytointia koskevaa keskustelua ja sen sisältö on:

> Here is my 2 cents: I have started 5 months ago at Futurice, and tribes only started make sense, say, only after the third month. It is not a familiar concept so takes time get into the tribe game. Hands-on experience required to understand how tribes work in Futurice and it also take sometime to understand your own role/significance inside the tribe. Long story short, I dont see the value in branding the tribes when it comes to recruitment. The applicants will apply for the brand Futurice not for the tribe four, south, or avalon anyways. IMO, this could have been done under the Futurice brand, with Tribe Four working together with HC to reach their recruitment goals.

Algoritmin tuottamat tulokset ovat oletettavasti hyvin kaoottisia yleiseen sanastoon kuuluvien termien kuten “recruitment” kohdalla. NLP:n keinoilla pitäisi paremmin tunnistaa keskustelun aihe ja puhujan rooli, sillä satunnaiset maininnat johonkin aiheeseen liittyen toimivat paremmin erilaisista tuotteista tai nimetyistä tekniikoista puhuttaessa.

Tulokset eivät kuitenkaan näytä parantuvan ollenkaan, vaikka rajoitettaisiin tarkastelu vain osaamisgraafin solmun “databases” alla oleviin solmuihin. Nämä solmut sisältävät erilaisten tietokantajärjestelmien nimiä, eli ne eivät kuulu normaaliin keskustelusanastoon. Tietokantajärjestelmiin liittyvien taitojen tulokset ovat kuvassa 5.5.

Otin jälleen tietokantatulosten joukosta tarkasteluun yksittäisiä pisteitä. Yhdellä työntekijällä on merkittynä 13 vuotta kokemusta MySQL-tietokannoista Power-järjestelmässä, mutta hän on koko työskentelyaikanaan lähettänyt vain 13 viestiä Flowdockin julkisille kanaville, eikä yksikään niistä liity tietokantoihin. Algoritmi on tästä huolimatta antanut vastaavalle taidolle nollasta reilusti poikkeavan osaamisarvon 0,2370.



Kuva 5.5. *Algoritmin tuottamat osaamisarvot tietokantajärjestelmiin liittyville taidoille suhteutettuna työntekijän samalle taidolle merkitsemään kokemukseen. Ensimmäisessä kuvassa on työntekijän ilmoittama kokemus vuosina, kun taas toisessa on työntekijän ilmoittamaa kokemusta skaalattu kaavalla* $x/(x+2)$*.*

Toinen henkilö sai termille “postgresql” algoritmilta korkean osaamisarvon 0,9274, mutta on merkannut Power-järjestelmään vain puoli vuotta kokemusta asian parissa. Hän on käynyt Flowdock-alustalla ainakin kaksi keskustelua tietokannasta, josta toisessa hän pyytää apua Travis-testiympäristössä tapahtuvaan PostgreSQL-tietokannan lokalisaatio-ongelmaan ja toisessa apua SQL-kyselyjen palauttamien outojen tulosten selvittämiseen. Hän on siis työskennellyt PostgreSQL-järjestelmän kanssa ja on keskustellut asiasta, mutta on ollut keskusteluissa asiantuntemusta hakevan henkilön eikä asiantuntijan roolissa. Lainaus yhdestä henkilön lähettämästä viestistä:

> I have an interesting problem. Postgres sorts Ä and Ö as A and O on my machine and correctly on ——’s machine. We both have tried with postgresql 9.5, with LC_COLLATE and LC_CTYPE set to UTF-8.

Erityishuomiona tarkastelussa selvisi, että samat termit mainitaan usein keskusteluissa moneen kertaan, mikä TF-IDF-algoritmin normalisoinnista huolimatta vääristää tuloksia merkittävästi. Yksikin vääränlainen keskustelu voi tehdä henkilöstä algoritmin silmissä ammattilaisen, minkä ei pitäisi puutteellisen luonnollisen kielen prosessoinnin huomioon ottaen olla mahdollista.

Vaikka algoritmin tuottama tieto yksittäisten termien osaamisesta ei olisikaan korrekti, pitäisi algoritmin silti tuottaa hyvä yleiskuva työntekijän kompetenssin jakautumisesta tietotekniikan, konsultoinnin, johtamisen ja suunnittelun välillä. Kuvasta 5.6 kuitenkin nähdään, että näin ei käy. Työntekijät on jaoteltu pääasiallisiin kompetenssialueisiin hei-

dän Power-järjestelmään merkitsemien kokemusvuosiensa perusteella. Tämän jälkeen mallista M on poimittu kutakin kompetenssialuetta edustavan juurisolmun osaamisarvo. Tuloksista nähdään, että algoritmi tuottaa suunnilleen samanlaisen jakauman jokaiselle työntekijälle, mutta vain eri voimakkuudella. Algoritmin tuloksista on siis mahdotonta päätellä suoraan, onko kyseessä esimerkiksi ohjelmoija vai suunnittelija.

Kuvasta 5.7 nähdään, että kompetenssit ovat helpommin hahmotettavissa algoritmin ensimmäisen iteraation jälkeen. Ongelma tulosten tasoittumisesta johtuu siis matriisin B päivittämiseen käytetystä kaavasta, joka lisää epäolennaisten termien vaikutusta lopputuloksiin merkittävästi. Ensimmäisenkin iteraation jälkeen tulokset sisältävät kuitenkin liikaa kohinaa ollakseen suoraan käyttökelpoisia.

5.2 Bayes-verkot

Käytössä olleet Bayes-verkot sisälsivät 270 solmua. Valitaan tarkastelun helpottamiseksi verkosta muutama pienempi osa. Valitut osat näkyvät kuvassa 5.8.

Futurice yrityksenä tekee pääasiassa sovelluskehitystä, suunnittelua ja konsultointia, joten valitut Bayes-verkon osat edustavat näitä kolmea taitokategoriaa. Sovelluskehitystä edustamaan olen valinnut kaksi verkon osaa, joista “web backend” on Futuricella hyvin edustettuna, kun taas “desktop apps”-kategoriaan pitäisi tulla paljon pienempiä arvoja yksittäisiä poikkeuksia lukuun ottamatta.

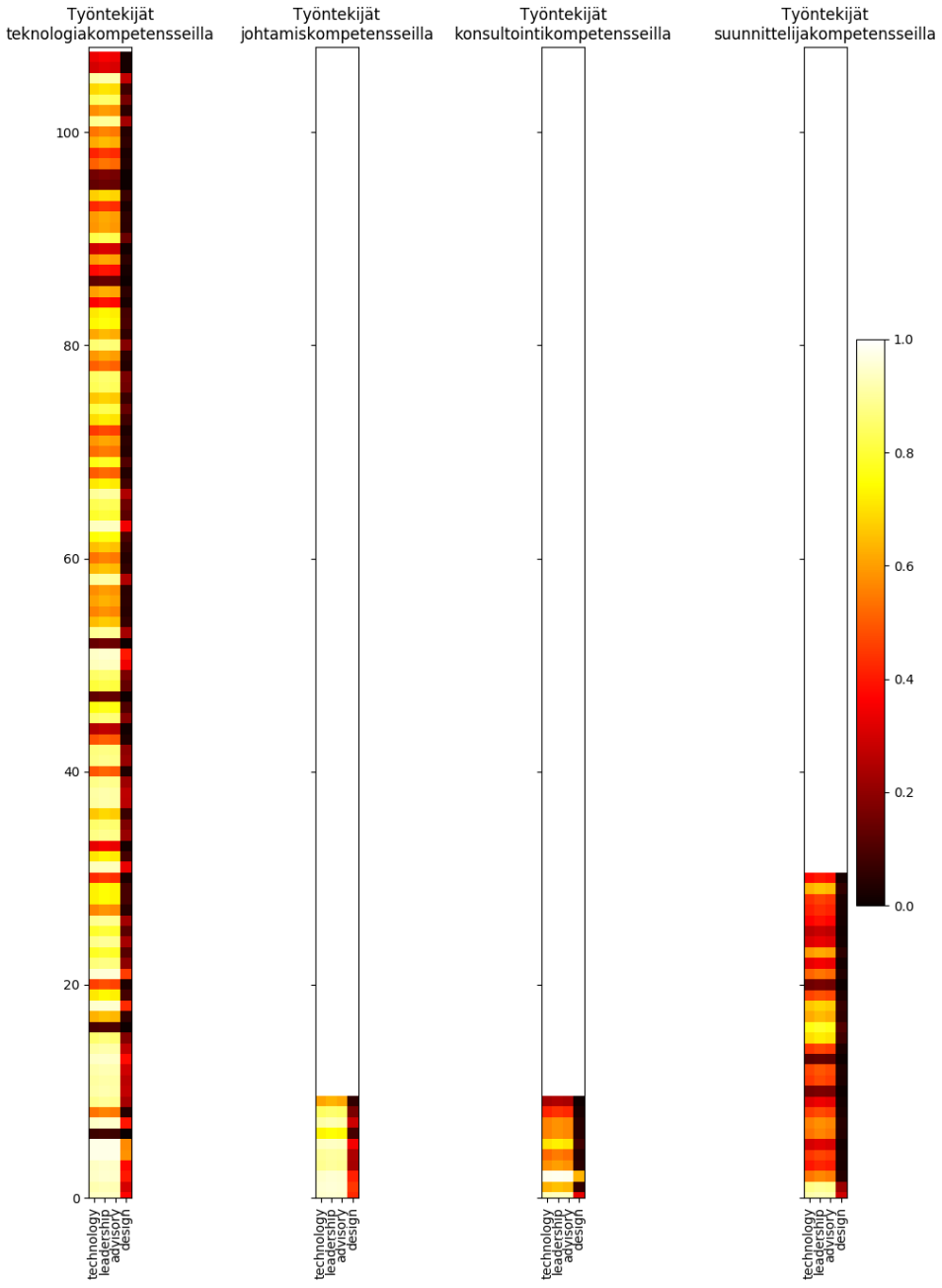
Konsultointia edustavassa verkon osassa oleva termi “lean service creation” on Futuricella sisäisesti kehitetty avoin työkalu palveluiden suunnitteluun ja projektien hallintaan. Koska se on Futuricen oma työkalu, löytyvät myös sen parhaat osaajat Futuricelta.

Työtä varten valittiin satunnaisesti kolme työntekijää, joiden Bayes-verkon arvot on otettu tarkempaan tarkasteluun. Jokaiselle henkilölle on tehty kysely, jossa heitä pyydetään valitsemaan asteikolla yhdestä kymmeneen heidän taitotasonsa valituissa aiheissa. Nämä vastaukset on listattu Bayes-verkon arvojen yhteydessä henkilön ilmoittamina arvoina ja niitä Bayes-verkkoon syöttämällä voidaan tarkastella mallin mukautumista työntekijöiltä saatuun tietoon. Työntekijän antaman arvon pitäisi antaa hyvä kuva työntekijän oikeasta osaamistasosta, mutta erilaiset näkemyserot ja muut tekijät voivat vaikuttaa arvon todenperäisyyteen sekä vertailukelpoisuuteen.

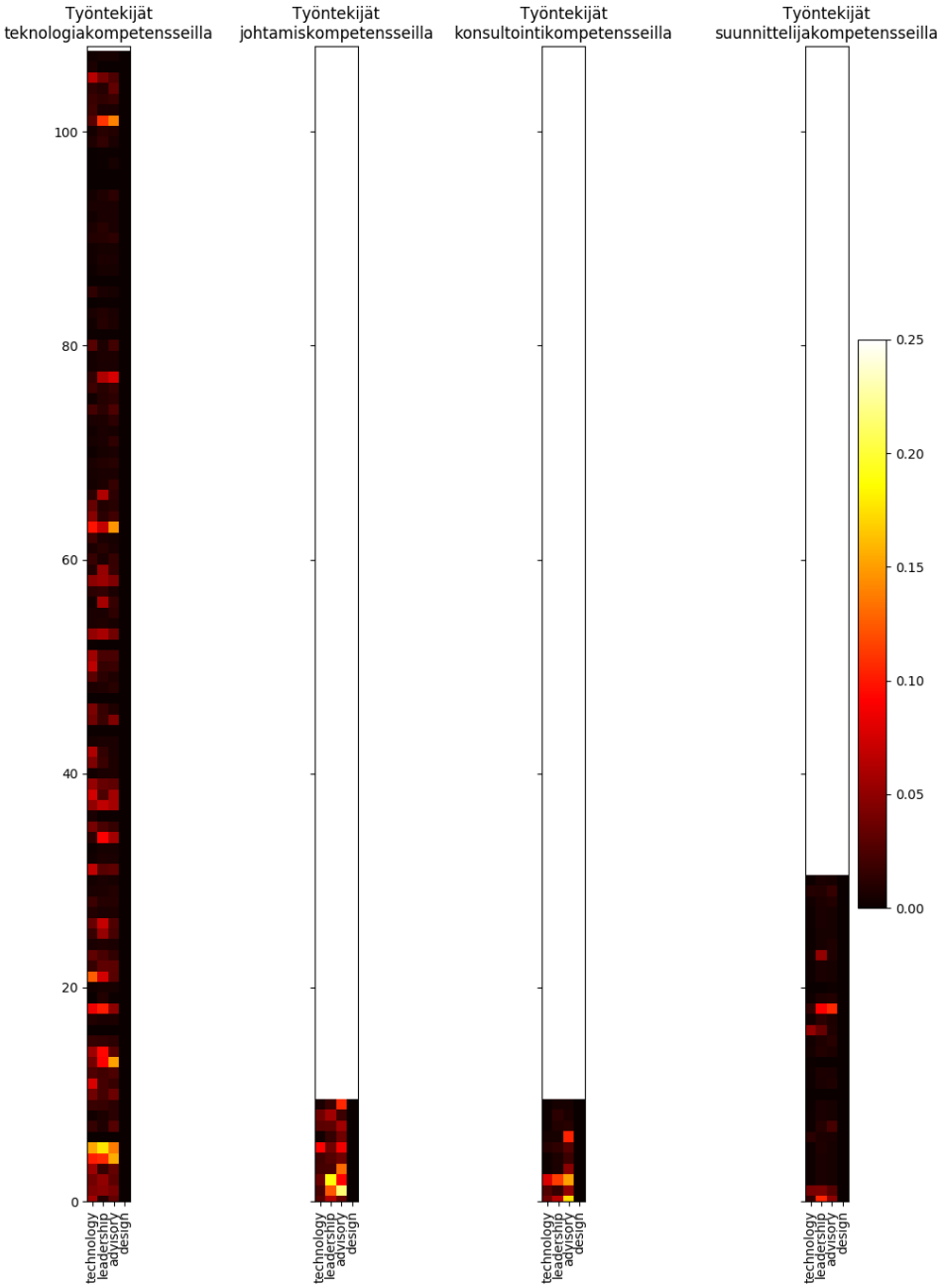
5.2.1 Henkilö A: backend-sovelluskehittäjä

Ensimmäinen henkilö on backend-verkkojärjestelmiä sekä internetiin kytkettyjen laitteita ohjelmoiva sovelluskehittäjä Futuricen Tampereen toimistolta ja hän on Power-järjestelmässä merkannut Bayes-verkosta valittuihin osiin liittyen olevansa asiantuntija Node.js-sovelluskehityksessä. Hänellä on myös merkittynä keskiverto-osaamisella kaksi muuta “web backend”-kategorian alla olevaa taitoa.

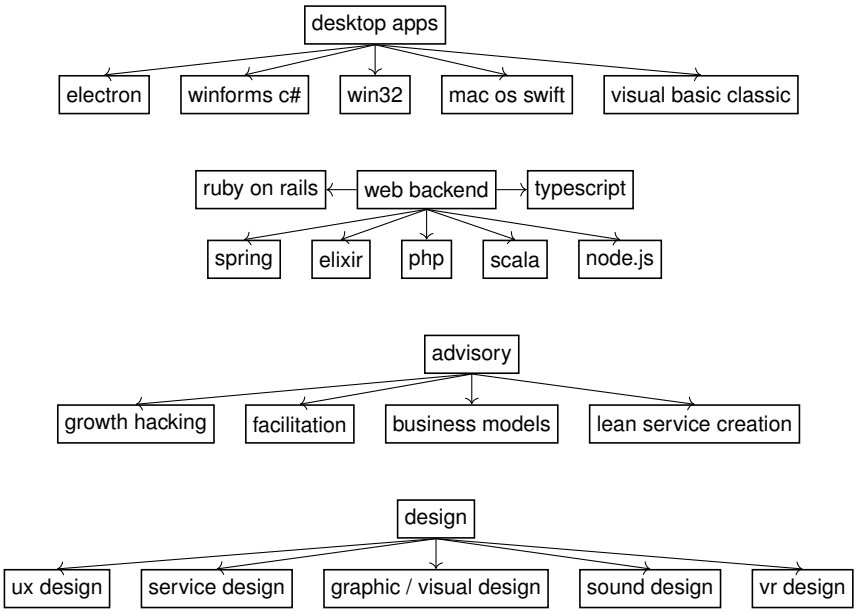
Taulukosta 5.3 nähdään, että algoritmin tuottamat tulokset ovat selkeästi painottuneita



Kuva 5.6. *Algoritmin mallin M juurisolmuille tuottama osaamisarvo eri kompetenssin työntekijöille. Työntekijät on jaoteltu kompetensseihin sen mukaan, missä heillä on yhteensä eniten merkittyjä kokemusvuosia Power-järjestelmässä.*



Kuva 5.7. *Algoritmin ensimmäisen iteraation jälkeen mallin M juurisolmuille tuottama osaamisarvo eri kompetenssin työntekijöille. Työntekijät on jaoteltu kompetensseihin sen mukaan, missä heillä on yhteensä eniten merkittyjä kokemusvuosia Power-järjestelmässä.*



Kuva 5.8. *Tarkempaan tarkasteluun valitut osat Bayes-verkosta. Solmuilla "desktop apps" ja "web backend" on kummallakin vanhempana solmu "technology"*

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	ilmoitettu
desktop apps	0.00	0.01	0.02	0.05	0.10	0.16	0.21	0.20	0.16	0.10	x_2
electron	0.00	0.00	0.00	0.01	0.05	0.11	0.19	0.25	0.23	0.15	x_5
winforms c#	0.00	0.01	0.06	0.16	0.28	0.28	0.16	0.05	0.01	0.00	x_1
win32 (c/c++)	0.00	0.00	0.02	0.08	0.20	0.29	0.25	0.12	0.03	0.01	x_1
mac os swift	0.00	0.00	0.02	0.06	0.17	0.27	0.26	0.16	0.05	0.01	x_1
visual basic classic	0.00	0.01	0.05	0.16	0.28	0.28	0.16	0.05	0.01	0.00	x_1
web backend	0.00	0.00	0.01	0.02	0.04	0.08	0.14	0.20	0.25	0.26	x_{10}
elixir	0.00	0.00	0.00	0.01	0.03	0.08	0.16	0.24	0.27	0.22	x_3
spring	0.00	0.00	0.00	0.01	0.02	0.06	0.13	0.22	0.28	0.27	x_5
ruby on rails	0.00	0.00	0.00	0.01	0.03	0.09	0.17	0.25	0.26	0.19	x_3
php	0.00	0.00	0.00	0.01	0.02	0.06	0.14	0.23	0.28	0.26	x_2
scala	0.00	0.00	0.00	0.01	0.03	0.07	0.14	0.22	0.28	0.26	x_2
node.js	0.00	0.00	0.00	0.01	0.03	0.07	0.15	0.24	0.27	0.23	x_{10}
typescript	0.00	0.00	0.00	0.01	0.03	0.07	0.14	0.22	0.27	0.26	x_{10}
advisory	0.00	0.01	0.02	0.03	0.06	0.10	0.14	0.18	0.22	0.23	x_6
business models	0.00	0.00	0.01	0.03	0.10	0.20	0.27	0.23	0.12	0.04	x_5
facilitation	0.00	0.00	0.01	0.03	0.09	0.19	0.27	0.24	0.13	0.04	x_7
growth hacking	0.00	0.00	0.01	0.04	0.11	0.21	0.27	0.22	0.11	0.03	x_5
lean service creation	0.00	0.00	0.00	0.01	0.03	0.07	0.14	0.22	0.27	0.26	x_7
design	0.02	0.05	0.11	0.17	0.21	0.19	0.14	0.07	0.03	0.01	x_3
graphic / visual design	0.00	0.02	0.09	0.21	0.29	0.23	0.11	0.03	0.00	0.00	x_3
service design	0.00	0.00	0.01	0.03	0.08	0.15	0.21	0.22	0.18	0.11	x_5
ux design	0.00	0.00	0.01	0.03	0.07	0.14	0.20	0.23	0.20	0.13	x_4
sound design	0.00	0.00	0.03	0.09	0.19	0.26	0.23	0.14	0.05	0.01	x_1
vr design	0.00	0.00	0.01	0.02	0.07	0.13	0.20	0.23	0.20	0.13	x_3

Taulukko 5.3. *Algoritmin henkilölle A tuottamat Bayes-verkon satunnaismuuttujien tilojen todennäköisyydet, sekä henkilön ilmoittamat satunnaismuuttujan tilat.*

verkkokehityksen suuntaan. Algoritmi antaa oikein termeille “web backend”, “node.js” ja “typescript” korkeat todennäköisyydet asiantuntijatason taitotiloille, mutta antaa samalla liian korkeat arviot termeille “elixir” ja “php”, joiden työntekijän ilmoittama arvo saa erittäin pienen todennäköisyyden. Kategorioiden “advisory” ja “design” alla olevien termien todennäköisyydet eivät osu ihan kohdilleen, mutta eri termien tulokset eroavat toisistaan tavalla, joka vaikuttaa heikosti myötäilevän työntekijän ilmoittamia arvoja. Merkittävänä poikkeuksena tähän on termi “sound design”, jolle algoritmi antaa poikkeuksellisen korkean ennusteen. Kaikkein pahimmat erehdykset tapahtuvat termin “dekstop apps” määrittämässä kategoriassa, jossa algoritmi on ilmeisesti tunnistanut henkilön teknologiaorientoituneeksi henkilöksi ja painottaa todennäköisyyksissä liian korkeita taitotasoja.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	ilmoitettu
desktop apps	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_2
electron	0.00	0.01	0.04	0.11	0.19	0.25	0.21	0.13	0.05	0.01	x_5
winforms c#	0.04	0.17	0.34	0.30	0.13	0.02	0.00	0.00	0.00	0.00	x_1
win32 (c/c++)	0.01	0.07	0.22	0.34	0.25	0.09	0.02	0.00	0.00	0.00	x_1
mac os swift	0.01	0.05	0.18	0.32	0.29	0.13	0.03	0.00	0.00	0.00	x_1
visual basic classic	0.04	0.17	0.33	0.31	0.13	0.03	0.00	0.00	0.00	0.00	x_1
web backend	0.00	0.01	0.04	0.09	0.14	0.19	0.20	0.16	0.11	0.06	x_{10}
elixir	0.00	0.00	0.00	0.02	0.05	0.12	0.20	0.24	0.22	0.14	x_3
spring	0.00	0.00	0.00	0.01	0.04	0.10	0.17	0.23	0.24	0.20	x_5
ruby on rails	0.00	0.00	0.01	0.02	0.06	0.14	0.21	0.24	0.20	0.12	x_3
php	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_2
scala	0.00	0.00	0.00	0.02	0.05	0.10	0.18	0.23	0.24	0.19	x_2
node.js	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	x_{10}
typescript	0.00	0.00	0.00	0.01	0.03	0.07	0.14	0.22	0.27	0.25	x_{10}
advisory	0.00	0.00	0.01	0.02	0.05	0.10	0.16	0.21	0.23	0.21	x_6
business models	0.00	0.00	0.01	0.03	0.09	0.20	0.28	0.24	0.13	0.04	x_5
facilitation	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	x_7
growth hacking	0.00	0.00	0.01	0.03	0.10	0.21	0.28	0.23	0.11	0.03	x_5
lean service creation	0.00	0.00	0.00	0.01	0.02	0.07	0.14	0.22	0.28	0.27	x_7
design	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_3
graphic / visual design	0.01	0.06	0.20	0.34	0.27	0.10	0.02	0.00	0.00	0.00	x_3
service design	0.00	0.00	0.02	0.06	0.14	0.21	0.23	0.18	0.10	0.04	x_5
ux design	0.00	0.00	0.02	0.05	0.12	0.20	0.23	0.20	0.12	0.05	x_4
sound design	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_1
vr design	0.00	0.00	0.01	0.05	0.12	0.20	0.24	0.20	0.12	0.05	x_3

Taulukko 5.4. *Algoritmin henkilölle A tuottamat Bayes-verkon satunnaismuuttujien tilojen todennäköisyydet Bayes-verkon satunnaismuuttujiin tehtyjen päivityksien jälkeen, sekä henkilön ilmoittamat satunnaismuuttujan tilat.*

Taulukkoa 5.4 varten valittiin muutama Bayes-verkon solmu, joihin asetettiin työntekijältä saatu tarkka arvo. Tällöin voidaan tarkastella näiden muutoksien vaikutusta tuloksiin. Heikosti arvioidun kategorian “desktop apps” kohdalla koko kategorian arvon asettaminen kohdilleen ajoi kaikkia sen lapsisolmujen todennäköisyyksiä huomattavasti pienempiä osaamisarvoja kohti toivotulla tavalla. Termiä “electron” vastaava tulos on melkein kohdillaan, mutta muut tarkasteltavat termit samasta kategoriasta saavat yhä algoritmilta liian korkeita arvioita.

Kategoriassa “web backend” nähdään, että todisteiden asettaminen satunnaisesti kategorian sisään voi pahimmassa tapauksessa pahentaa tuloksia. Kyseisessä kategoriassa on iso määrä lapsitermejä ja jokainen niistä vaatii hyvin spesifiä tietoa. Vaikka henkilö kokee itsensä ekspertiksi kyseisessä aiheessa, hän tietää tarkemmin vain muutamasta tarkasta aiheesta. Yksi todiste huonosti hallitusta teknologiasta saa henkilön koko osaamisen näyttämään huomattavasti heikommalta verkkokehitykseen liittyen, vaikka toinen todiste saatiinkin henkilön parhaiten hallitsemasta aiheesta “node.js”. Henkilön osaamisen hahmottaminen puumaisen Bayes-rakenteen lehdistä lähtevällä haarukoinnilla vaikuttaa tämän mukaan hyvin huonolta ratkaisulta.

5.2.2 Henkilö B: suunnittelija

Toinen henkilö on Futuricen Helsingin toimistolla työskentelevä suunnittelija. Hänen ainoat suunnitteluun liittyvät merkintänsä Power-järjestelmässä ovat, että hänellä on noin kuusi vuotta kokemusta graafisesta ja visuaalisesta suunnittelusta sekä tuntematon määrä kokemusta käyttökokemussuunnittelusta.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	ilmoitettu
desktop apps	0.13	0.13	0.13	0.12	0.11	0.10	0.09	0.08	0.06	0.05	x_1 (x_7)
electron	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
winforms c#	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
win32 (c/c++)	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
mac os swift	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
visual basic classic	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
web backend	0.12	0.13	0.13	0.13	0.12	0.11	0.09	0.08	0.06	0.05	x_1
elixir	0.14	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
spring	0.13	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1
ruby on rails	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
php	0.13	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1
scala	0.13	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1
node.js	0.14	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1
typescript	0.13	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1
advisory	0.12	0.12	0.12	0.12	0.11	0.10	0.09	0.08	0.07	0.06	x_3
business models	0.14	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.05	0.03	x_1
facilitation	0.14	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.05	0.03	x_5
growth hacking	0.14	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.05	0.03	x_3
lean service creation	0.12	0.14	0.14	0.14	0.13	0.11	0.09	0.07	0.05	0.03	x_7
design	0.13	0.13	0.12	0.12	0.11	0.10	0.09	0.08	0.07	0.06	x_7
graphic / visual design	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.05	0.03	x_6
service design	0.14	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_5
ux design	0.13	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_8
sound design	0.14	0.15	0.14	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
vr design	0.14	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1

Taulukko 5.5. *Algoritmin henkilölle B tuottamat Bayes-verkon satunnaismuuttujien tilojen todennäköisyydet, sekä henkilön ilmoittamat satunnaismuuttujan tilat.*

Algoritmin henkilölle B tuottamat todennäköisyydet ovat hyvä esimerkki tilanteesta, jos-

sa algoritmi ei ole saanu käyttöönsä tarpeeksi informaatiota kyseisestä henkilöstä. Kuten taulukosta 5.5 nähdään, todennäköisyydet ovat sekä pienempiin osaamisarvoihin painottuneita, että jakaumaltaan laveita. Joissain taidoissa näkyy hieman muutosta paremman kokemustason suuntaan, mutta tuloksista olisi vaikea sanoa mitään varmuudella ilman työntekijän antamia oikeita tiloja. Lisäksi algoritmi on antanut termille “web backend” taidon puolesta muihin verrattuna suhteellisen hyvän ennusteen, vaikka henkilö kokee oman taitotasonsa hyvin matalaksi aiheeseen liittyen. Henkilö vastasi aluksi termin “desktop apps” olevan arvossa x_7, mutta varmisti myöhemmin kyseessä olleen väärinymmärryksen.

Taulukossa 5.5 termiä “desktop apps” vastaavalla rivillä oleva poikkeuksellinen ilmoitetun arvon merkintä johtuu siitä, että x_7 vastaa työntekijän kokemusta työpöytäsovelluksien käytöstä, mutta kyseinen kategoria edustaa nimenomaan työpöytäsovelluksien ohjelmointia. Henkilö B korjasi väärinymmärryksestä johtuneen merkinnän tilaan x_1. Tämä luonnollisen kielen epätarkkuus aiheuttaa myös ongelmia tekstin analysoinnin aikana.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	ilmoitettu
desktop apps	0.21	0.18	0.15	0.13	0.10	0.08	0.06	0.04	0.03	0.02	x_1 (x_7)
electron	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_1
winforms c#	0.15	0.15	0.15	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
win32 (c/c++)	0.15	0.15	0.15	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
mac os swift	0.15	0.15	0.15	0.13	0.12	0.10	0.08	0.06	0.04	0.03	x_1
visual basic classic	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_1
web backend	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_1
elixir	0.18	0.17	0.16	0.14	0.11	0.09	0.06	0.04	0.03	0.02	x_1
spring	0.17	0.17	0.16	0.14	0.11	0.09	0.06	0.04	0.03	0.02	x_1
ruby on rails	0.18	0.17	0.16	0.14	0.11	0.09	0.06	0.04	0.03	0.02	x_1
php	0.18	0.17	0.16	0.14	0.11	0.09	0.06	0.04	0.03	0.02	x_1
scala	0.18	0.17	0.16	0.14	0.11	0.09	0.06	0.04	0.03	0.02	x_1
node.js	0.18	0.17	0.16	0.14	0.11	0.09	0.06	0.04	0.03	0.02	x_1
typescript	0.13	0.14	0.14	0.14	0.12	0.10	0.08	0.06	0.04	0.03	x_1
advisory	0.08	0.10	0.10	0.11	0.11	0.11	0.11	0.10	0.09	0.08	x_3
business models	0.13	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.05	0.03	x_1
facilitation	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	x_5
growth hacking	0.13	0.14	0.14	0.13	0.12	0.10	0.08	0.06	0.05	0.03	x_3
lean service creation	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	x_7
design	0.08	0.09	0.10	0.10	0.11	0.11	0.11	0.11	0.10	0.09	x_7
graphic / visual design	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	x_6
service design	0.13	0.14	0.14	0.13	0.12	0.11	0.09	0.07	0.05	0.03	x_5
ux design	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	x_8
sound design	0.13	0.14	0.14	0.13	0.12	0.10	0.08	0.07	0.05	0.03	x_1
vr design	0.13	0.14	0.14	0.13	0.12	0.11	0.09	0.07	0.05	0.03	x_1

Taulukko 5.6. *Algoritmin henkilölle B tuottamat Bayes-verkon satunnaismuuttujien tilojen todennäköisyydet Bayes-verkon satunnaismuuttujiin tehtyjen päivityksien jälkeen, sekä henkilön ilmoittamat satunnaismuuttujan tilat.*

Todennäköisyysjakauma pysyy hyvin huterana henkilön B kohdalla varman tiedon lisäämisenkin jälkeen. Kategorioiden “desktop apps” ja “web backend” todennäköisyydet siir-

tyvät heikompaa tietotasoa kohti, mutta liian hitaasti. Todennäköisyyksien jakauma pysyy aivan liian laveana, sillä jos vaikka henkilön PHP-ohjelmointikielen osaamisesta ei ole todisteita, on kokemattomuus koko verkkosovelluksien kehityksen kategoriassa riittävä todiste olettamaan myös lähes täyttä tietämättömyyttä PHP:n käytöstä. Algoritmin pitäisi siis osata siirtyä täydestä tietämättömyydestä paljon nopeammin lähes varmaan tietoon, kun solmun vanhempiin tulee lisätietoa. Tässä mielessä Bayes-verkon todennäköisyystaulukoiden muodostamisessa ei pitäisi käyttää lainkaan solmua vastaavaa epävarmuustietoa, vaan jotain sille käänteistä arvoa. Mitä epävarmempaa tieto on, sen vahvemmin solmun vanhemmista saadun tiedon pitäisi vaikuttaa lopputulokseen.

Kategorioihin “advisory” ja “design” on kumpaankin asetettu kahden työntekijän parhaiten hallitseman taidon tarkka tila. Tämä parantaa algoritmin antamaa ennustetta henkilön yleisestä kompetenssista ihan hyvin, eivätkä muut kyseisten kategorioiden taidot saa heti merkittävästi korkeampiin taitotasoihin painottuneempia todennäköisyyksiä. Tässä tapauksessa algoritmi toimii suunnilleen odotetusti, joskin varsinkin kategorioita vastaavat todennäköisyysjakaumat pysyvät yhä aivan liian laveina.

5.2.3 Henkilö C: konsultti

Kolmas henkilö on konsultti Futuricen Helsingin toimistolta. Hän on Power-järjestelmässä merkannut noin vuoden verran kokemusta aiheista “Business models” ja “Facilitation” sekä hieman kokemusta aiheesta “Lean Service Creation (LSC)” ja yhdestä muusta samaan kategoriaan kuuluvasta aiheesta. Tämän lisäksi hänellä on suunnilleen saman verran kokemusta myös suunnittelutyöstä, josta eniten kokemusta hänellä on merkattuna aiheeseen “Service Design”.

Kolmannen henkilön kohdalla algoritmin tuottamat todennäköisyysarvot ovat lähellä käyttökelvotonta. Kuten taulukosta 5.7 nähdään, todennäköisyyksien jakauma on suunnilleen samanlainen kaikille taidoille, vaikka henkilön antamat arvot ovat vahvasti painottuneita. Algoritmi tuottaa henkilön suhteellisen hyvin hallitsemalle kategorialle “design” jopa muihinkin kategorioihin verrattuna pienempiä arvoja, mikä ei kuvasta 5.6 vedettyjen johtopäätöksien varjolla ole kovin yllättävää. Ainoa jotenkin henkilön ilmoittamiin arvoihin hyvin sopiva tulos on kategorian “desktop apps” matalien osaamistasojen hyvät ennusteet, mutta matalaa osaamistasoa on helpompi ennustaa kuin korkeaa. Algoritmi antaa termin “electron” osaamiselle poikkeuksellisen positiivisia tuloksia, vaikka henkilö ei ole missään viestissään maininnut termiä suoraan.

Asettamalla kaikkien kategorioiden tarkat arvot Bayes-verkkoon saadaan yksittäisten kategorioiden tulokset painotettua oikeisiin suuntiin ja tulokset näyttävät heti pintapuolisesti paremmalta, mikä nähdään taulukossa 5.8. Tuloksien mukautuminen ei kuitenkaan ole toivotun voimakasta, sillä kategoriassa “web backend” yksittäisten tekniikoiden tulokset jäävät liian myönteisiksi ja kategoriassa “advisory” yleisesti liian negatiivisiksi. Algoritmilla on myös vaikeuksia mukautua kategorian “design” hyvin vaihteleviin arvoihin.

Asettamalla työntekijän antamia tuloksia ylempiin kategorioihin voidaan Bayes-verkon

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	ilmoitettu
desktop apps	0.15	0.17	0.17	0.16	0.13	0.10	0.06	0.04	0.02	0.01	x_2
electron	0.09	0.14	0.19	0.20	0.17	0.11	0.06	0.03	0.01	0.00	x_1
winforms c#	0.21	0.23	0.21	0.16	0.10	0.05	0.02	0.01	0.00	0.00	x_1
win32 (c/c++)	0.19	0.22	0.21	0.17	0.11	0.06	0.03	0.01	0.00	0.00	x_1
mac os swift	0.19	0.22	0.21	0.17	0.11	0.06	0.03	0.01	0.00	0.00	x_1
visual basic classic	0.21	0.23	0.21	0.16	0.10	0.05	0.02	0.01	0.00	0.00	x_1
web backend	0.02	0.04	0.06	0.09	0.11	0.14	0.15	0.15	0.13	0.11	x_2
elixir	0.03	0.07	0.12	0.17	0.19	0.17	0.13	0.08	0.04	0.01	x_1
spring	0.01	0.03	0.06	0.10	0.14	0.17	0.17	0.15	0.10	0.06	x_2
ruby on rails	0.05	0.09	0.14	0.18	0.19	0.16	0.11	0.06	0.02	0.01	x_1
php	0.01	0.03	0.07	0.11	0.15	0.18	0.17	0.14	0.09	0.05	x_1
scala	0.01	0.03	0.07	0.11	0.15	0.18	0.17	0.14	0.09	0.05	x_2
node.js	0.02	0.05	0.10	0.15	0.18	0.18	0.15	0.09	0.05	0.02	x_2
typescript	0.01	0.03	0.07	0.11	0.15	0.18	0.17	0.14	0.09	0.05	x_2
advisory	0.03	0.05	0.07	0.09	0.11	0.13	0.14	0.14	0.13	0.11	x_8
business models	0.10	0.15	0.18	0.19	0.16	0.11	0.06	0.03	0.01	0.00	x_9
facilitation	0.10	0.15	0.18	0.19	0.16	0.11	0.06	0.03	0.01	0.00	x_{10}
growth hacking	0.11	0.15	0.19	0.19	0.16	0.11	0.06	0.03	0.01	0.00	x_6
lean service creation	0.01	0.02	0.05	0.08	0.12	0.16	0.17	0.16	0.13	0.09	x_8
design	0.19	0.19	0.17	0.14	0.11	0.08	0.05	0.03	0.02	0.01	x_7
graphic / visual design	0.21	0.23	0.21	0.16	0.10	0.06	0.03	0.01	0.00	0.00	x_4
service design	0.06	0.12	0.17	0.19	0.18	0.13	0.08	0.04	0.02	0.01	x_9
ux design	0.04	0.08	0.14	0.18	0.18	0.16	0.11	0.06	0.03	0.01	x_6
sound design	0.18	0.22	0.21	0.17	0.12	0.06	0.03	0.01	0.00	0.00	x_3
vr design	0.04	0.09	0.14	0.18	0.18	0.16	0.11	0.06	0.03	0.01	x_1

Taulukko 5.7. *Algoritmin henkilölle C tuottamat Bayes-verkon satunnaismuuttujien tilojen todennäköisyydet, sekä henkilön ilmoittamat satunnaismuuttujan tilat.*

tuottamia todennäköisyysarvoja painottaa nopeasti haluttuihin suuntiin. Tämä ei kuitenkaan auta yhtään, jos verkko ei pysty tämänkään jälkeen tarjoamaan mitään lisätietoa. Kun henkilö merkataan kokemattomaksi jossain kategoriassa, saadaan kategorian alatuloksia tarkastelemalla tietoon vain, että henkilö on keskimääräisesti kokematon kyseisessä kategoriassa. Silloinkin kun yksittäiset arvot erottuvat selkeästi joukosta, on vaikea sanoa ilman käyttäjän antamaa tarkkaa tietoa, onko kyseessä termin “lean service creation” kaltainen tässä suhteellisen hyvin onnistunut ennustus vai termin “electron” kaltainen huonommin onnistunut ennustus.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	ilmoitettu
desktop apps	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_2
electron	0.13	0.20	0.23	0.20	0.14	0.07	0.03	0.01	0.00	0.00	x_1
winforms c#	0.27	0.27	0.21	0.14	0.07	0.03	0.01	0.00	0.00	0.00	x_1
win32 (c/c++)	0.26	0.26	0.22	0.14	0.07	0.03	0.01	0.00	0.00	0.00	x_1
mac os swift	0.25	0.26	0.22	0.15	0.08	0.03	0.01	0.00	0.00	0.00	x_1
visual basic classic	0.27	0.27	0.21	0.14	0.07	0.03	0.01	0.00	0.00	0.00	x_1
web backend	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	x_2
elixir	0.11	0.18	0.22	0.21	0.15	0.08	0.03	0.01	0.00	0.00	x_1
spring	0.04	0.09	0.14	0.18	0.19	0.16	0.11	0.06	0.03	0.01	x_2
ruby on rails	0.15	0.21	0.23	0.19	0.12	0.06	0.02	0.01	0.00	0.00	x_1
php	0.05	0.10	0.16	0.19	0.19	0.15	0.09	0.04	0.02	0.01	x_1
scala	0.05	0.10	0.16	0.19	0.19	0.15	0.09	0.04	0.02	0.01	x_2
node.js	0.09	0.15	0.21	0.21	0.17	0.10	0.05	0.02	0.00	0.00	x_2
typescript	0.01	0.03	0.07	0.11	0.16	0.18	0.17	0.13	0.09	0.05	x_2
advisory	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	x_8
business models	0.05	0.11	0.17	0.20	0.19	0.14	0.08	0.04	0.01	0.00	x_9
facilitation	0.06	0.11	0.17	0.20	0.19	0.14	0.08	0.04	0.01	0.00	x_{10}
growth hacking	0.06	0.11	0.17	0.20	0.19	0.14	0.08	0.04	0.01	0.00	x_6
lean service creation	0.00	0.01	0.02	0.05	0.10	0.15	0.18	0.19	0.17	0.12	x_8
design	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	x_7
graphic / visual design	0.10	0.16	0.19	0.19	0.16	0.10	0.06	0.02	0.01	0.00	x_4
service design	0.01	0.04	0.09	0.15	0.20	0.20	0.16	0.10	0.04	0.02	x_9
ux design	0.01	0.02	0.06	0.11	0.17	0.20	0.18	0.13	0.07	0.03	x_6
sound design	0.07	0.13	0.18	0.21	0.18	0.12	0.07	0.03	0.01	0.00	x_3
vr design	0.01	0.03	0.06	0.12	0.17	0.20	0.18	0.13	0.07	0.03	x_1

Taulukko 5.8. *Algoritmin henkilölle C tuottamat Bayes-verkon satunnaismuuttujien tilojen todennäköisyydet Bayes-verkon satunnaismuuttujiin tehtyjen päivityksien jälkeen, sekä henkilön ilmoittamat satunnaismuuttujan tilat.*

6 MUITA HAVAINTOJA

6.1 Käytetyt tekniikat

Algoritmin toteuttamiseen käytettiin Python-ohjelmointikieltä ja SQL-tietokantaa. Data saatiin JSON- ja CSV-formaateissa. CSV-tiedostojen lataamiseen ja esikäsittelyyn käytettiin Pythonin pandas-kirjastoa ja matriisilaskentaa varten data muutettiin NumPy-matriiseiksi ja SciPy-kirjaston harvoiksi matriiseiksi. Tämä osoittautui toteuttamisen puolesta tehokkaaksi ja nopeasti toteutettavaksi ratkaisuksi, joskin pandas-kirjaston yleinen käytettävyys osoittautui paljon SQL-kantaa heikommaksi. Kirjaston monimutkaisempia taulumanipulaatioon tarkoitettuja ominaisuuksia käytettiin lähinnä ylimääräisten taulukon kolumnien poistamiseen ja SQL-kyselyiden palauttamien taulukoiden muuttamiseen Python-ohjelmointikielen rakenteiksi. Bayes-verkkojen käsittelyyn käytettiin Python-ohjelmointikielen pomegranate-kirjastoa.

Pandas osoittautui yksinkertaiseen laskentaan ja tarkasteluun todella hitaaksi. Algoritmi suorittaa paljon satunnaishakuja mm. käyttäjien ID:n perusteella, jolloin Pythonin omat sanakirjat (eng. dictionary) osoittautuivat paljon nopeammiksi. Listamaista käyttöä varten data muutettiin Pythonin natiivilistoiksi. SciPy-kirjaston tarjoamat harvat matriisit osoittautuivat erityisen tärkeäksi matriisien O_K, O_F, Y_K ja Y_F toteuttamisessa.

6.2 Luonnollisen kielen analysointi

Hyvin toteutettuun luonnollisen kielen analyysiin liittyy aina paljon kontekstin huomioimista. Monet yksittäiset termit voivat esiintyä eri yhteyksissä ja eri tarkoituksilla. Esimerkiksi ohjelmointikielen nimeä voi käyttää joko aiheesta osaava henkilö asiasta keskustellessaan tai vasta-alkaja apua kysyessään. Termi “language” taas voi tarkoittaa joko ohjelmoinnissa tai puheessa käytettävää kieltä.

Luonnollinen kieli on sanastonsa puolesta hyvin kaoottista. Tässä työssä ammatiaiheisiin liittymättömiä sanoja yritettiin karsia pois erilaisin perustein, mutta lopputulos oli silti hyvin kaoottinen ja henkilöiden taitotasosta tehdyt tulkinnat liian geneerisiä ja epävarmoja. Iteroinnin yhteydessä tehtävä matriisin B päivittäminen lisäsi käyttökelpoisen informaation määrää, mutta toi myös mukaan erilaiset hajanaiset roskatokenit, jotka tasoittivat lopputulosta liikaa. Mitä enemmän lähtötietoa algoritmille voidaan antaa synonyymeistä, keskusteluhuoneista tai yksittäisistä henkilöistä, sitä paremmin algoritmi voi suorittaa tarvitsemaansa analyysiä erilaisten tokenien merkityksestä ja viestien kontekstista.

Tarkempi luonnollisen kielen analysoinnin puute on merkittävä puute algoritmin toiminnassa, mutta myös pieni etu käsitellessä suomen ja englannin kieliä sekoittavaa materiaalia. Suomen kielen taivutusmuodot vaikeuttavat suomenkielisten viestien analysointia, mutta käytetyssä token-pohjaisessa systeemissä tämä johti useimpien normaalissa keskustelussa käytettävien sanojen karsiutumiseen tokeneita suodatettaessa. Lopullisten tokenien joukkoon tuli lisää roskatokeneita, kuten esimerkiksi sanat “kyllä” (2007 esiintymää) ja “joku” (2016 esiintymää), mutta merkittävä osa tarkempia aiheita koskevista muutaman henkilön tietoteknisistä keskusteluista tapahtuu myös suomen kielellä Futuricen merkittävän suomenkielisen enemmistön vuoksi. Lisäksi monet verkon solmujen termeinä käytetyt englanninkieliset termit, kuten “growth hacking” ovat käytössä myös suomenkielisissä keskusteluissa yleisesti hyväksyttyjen käännösten puutteen vuoksi.

6.3 Algoritmin toiminta

Yksi algoritmissa vahvasti toimiva idea on tekstistä kerätyn lähes varman tiedon (asioiden suorat maininnat) sekä henkilöiden osallistumistiedon käyttö tokenien kontekstitiedon parantamiseen. Jos jokin token esiintyy toistuvasti samassa kontekstissa kuin joku aikaisemmin tunnistettu token, voidaan joitain aikaisemmin tunnistetun tokenin ominaisuuksia siirtää toiselle tokenille. Näin saadaan hajanaisesta tekstidatasta lisää tietoa irti ja tietoa tokeneista voidaan taas parantaa. Tässä tavoitteessa kuitenkin epäonnistuttiin, sillä algoritmin nykyinen muoto on tässä liian aggressiivinen ja tarkempi tieto hukkuu roskatermien vaikutuksen alle.

Merkittävin ongelma tokenien kontekstitiedon iteratiivisessa parantamisessa on, että sillä voidaan parantaa vain yhtä osaa algoritmista. Tässä valittiin matriisi B parantamisen kohteeksi, mutta tällöin Bayes-verkon ominaisuudet pitää joko määritellä käsin tai asettaa johonkin oletusarvoon. Iteraatioiden tuottamilla tuloksilla voitaisiin myös teoriassa päivittää verkon ominaisuuksia suoraan, mutta tällöin matriisin B tarkempaan määrittelyyn pitäisi nähdä enemmän vaivaa. Bayes-verkkoa voitaisiin myös laajentaa erilaisilla piilotetuilla solmuilla tai muilla vastaavilla rakenteilla ja luoda vahvempi kytkös Bayes-verkon ja tokenien välille ilman matriisia B.

Kuvassa 3.3 olevia asiakassuhteita edustavia solmuja ei tässä työssä käytetty, mutta niiden mukaan ottaminen olisi erityisen tärkeää yrityksen liiketoiminnan kannalta. Niiden kohdalla Bayes-verkon kytköksien tulkinta muuttuu kuitenkin paljon monimutkaisemmaksi. Jos henkilöllä on paljon tietoa asiakasprojektiin liittyen, hän ei välttämättä tiedä mitään asiakasprojektissa hyödynnetyistä tekniikoista. Kyseinen työntekijä voi olla vaikka projektin alkuvaiheessa tukea antanut konsultti, joka tietää projektin tavoitteet hyvin, mutta ei osallistu lainkaan varsinaiseen toteutukseen. Tämä tarkoittaa siis sitä, että asiakasprojektista paljon tietävä henkilö voi olla joko asiantuntija projektissa käytetyissä tekniikoissa tai täysin tietämätön niistä, vain välissä olevien tilojen todennäköisyys pienenee. Työssä käytetty algoritmi ei tue tämänlaisten totuustaulukoiden muodostamista.

7 YHTEENVETO

7.1 Tutkimuskysymyksiin vastaaminen

> Kuinka paljon työntekijöiden osaamisesta ennalta määritellyissä taitokategorioissavoidaan saada selville valittua menetelmää käyttäen suhteutettuna työntekijöidenkokonaismäärään sekä hypoteettiseen yrityksen hallussa olevaan kokonaistietotaitoon?

Opetuskäytössä hyödynnetyt opiskelijan tiedon analysointiin käytetyt menetelmät eivät sovellu hyvin tekstianalysointitehtäviin, ellei tekstin käsittelyssä pystytä ilmaisemaan luonnollisen kielen analysoinnin keinoin henkilön tarkempaa suhdetta käsiteltäviin asioihin. Kaoottisen keskusteluhistoriadatan pohjalta pelkästään kontekstianalyysin keinoin rakennetut tulokset ovat epävarmoja ja laveita. Algoritmi pystyy rakentamaan melko hyvän päätelmän yrityksen käytössä olevasta kokonaistietomäärästä keskustelutrendien pohjalta, mutta ei pysty antamaan yksittäisille työntekijöille suoraan luotettavia tai käyttökelpoisia arvioita heidän osaamisestaan.

> Soveltuuko työssä valittu malli työntekijän tietotaidon mallintamiseen?

Bayes-verkot vaikuttavat olevan hyvä valinta yrityksen työntekijöiden osaamisen arviointiin. Oikein määriteltynä Bayes-verkko pystyy ilmaisemaan erilaisia osaamisen tasoja ja tiedon varmuutta. Yrityksen käytössä olevia eri tietolähteitä voidaan myös helposti käyttää saadun mallin tarkentamiseen ilman, että kaikkia määriteltyjä taitoja joudutaan erikseen kartoittamaan. Verkon määrittelyllä on kuitenkin suuri merkitys lopputuloksen käyttökelpoisuuteen. Aiheiden karkea kategorioihin pohjautunut organisointi osoittautui liian matalaksi rakenteeksi, sillä hierarkiassa ylempänä olevilla solmuilla oli hyvinkin erilaisia ja paljon erityistä asiantuntemusta vaativia teknologioita lapsinaan. Kunnollisessa Bayes-verkossa nämä yksittäiset asiantuntemuksen palaset olisivat mukana verkossa tarjoamassa lisätietoa henkilön osaamisen tarkemmasta painottumisesta.

> Miten valittua mallia voisi parantaa?

Algoritmi tärkeimmät osat jouduttiin rakentamaan työtä varten lähes alusta, mikä heikensi tuloksia diplomityön aika- ja resurssirajoitteiden vuoksi. Algoritmin testaaminen isolla ja kaoottisella tekstidatalla oli myös huono ratkaisu, sillä useat tulokset joudutaan esittämään osittaisina tai hankalasti tulkittavissa muodoissa. Kaventamalla tarkastelua yksittäisiin osiin prosessia ja parantamalla niiden toimintaa voitaisiin koko algoritmin tuottamia tuloksia parantaa huomattavasti. Erityisesti vastatodisteiden kerääminen eli henkilön tie-

tämättömyyden päätteleminen kysyvien viestien pohjalta voisi myös parantaa algoritmin toimintavarmuutta.

> Kuinka luotettavaa yrityksen sisäinen keskusteludata on työntekijöiden osaamisenarvioimisessa?

Yrityksen sisäinen keskusteludata on itsessään melko epäluotettava lähde työntekijöiden osaamisen arvioimiseen. Silloinkin kun työntekijöiden osallistuminen keskusteluihin on korkea, varsinaisten kiinnostavien keskusteluiden määrä on hyvin pieni ja haluttua informaatiota sisältävät viestit pitää suodattaa tarkasti joukosta. Yhdistettynä muuhun tietoon keskusteludata voi kuitenkin tarjota tärkeää lisätietoa, sillä riittävä esitieto joko tarkasteltavista aiheista tai henkilöistä voi auttaa puuttuvien tietojen täydentämiseen silloinkin kun keskustelusta voidaan päätellä vain niihin osallistuneiden henkilöiden interaktion yleinen määrä ja laatu.

7.2 Henkilökohtainen arviointi tehtyjen valintojen toimivuudesta

Monia työhön tehtyjä valintoja on vaikea arvioida kriittisesti sopivalla tarkkuudella. Tähän kappaleeseen on kerätty henkilökohtaisen intuition pohjalta tehtyjä arvioita siitä, mitkä työssä tehdyt valinnat onnistuivat ja mitkä eivät. Seuraavat valinnat vaikuttivat osoittautuneen hyviksi:

- Bayes-verkon mallinnus diskreettinä. Aloitin työn tekemisen normaalijakautunutta Bayes-puuta käyttämällä, mutta tämä tekisi kuvan 3.3 kaltaisten puiden tarkasta mallintamisesta mahdotonta, kuten kappaleessa 6.3 on mainittu. Tässä työssä tämä valinta ei kuitenkaan auttanut verkon hieman erilaisen rakenteen vuoksi, millään verkon solmulla ei ollut kahta tai useampaa vanhempaa solmua.
- Matriisin U rakennukseen käytettävä kaava (4.5) ottaa hyvin huomioon kaiken työssä tarkastellun lähtötiedon ja sisällyttää yksinkertaisen tekstianalyysin TFIDF-kaavaa käyttämällä. Parempien tulosten saamiseksi lähtödatalle pitäisi suorittaa esiprosessointi NLP:n keinoin, jolloin saataisiin käyttöön enemmän lähtötietoa ja matriisin U rakentamiseen käytettävä algoritmi pitäisi myös rakentaa uudestaan.

Seuraavat valinnat vaikuttivat osoittautuneen huteriksi tai jopa haitallisiksi:

- Graafin rakenne. Power-järjestelmästä poimitut kategoriat ovat liian laajoja ja niiden väliset kytkennät eri tarkoitukseen rakennettuja. Verkon pitäisi olla paljon isompi ja siinä pitäisi olla enemmän haarojen välisiä ristisuhteita. Kytkösten vahvuudet pitäisi myös asettaa tarkemmin.
- Todisteiden kerääminen kaikille graafin solmuille toteutettiin verkon suppean rakenteen vuoksi, mutta tämä aiheutti useita ongelmia todisteiden propagoinnin käsittelyssä. Informaatiota olisi pitänyt ottaa sisään graafiin vain lehtisolmujen kautta, kuten opetukseen käytetyissä malleissa.

- Viestien lähetysajan huomiotta jättäminen. Viestien lähetysajoista voitaisiin päätellä enemmän henkilöiden työskentelyajasta tiettyjen aiheiden parissa. Tämä jätettiin pois lähinnä aikaresurssien puutteen vuoksi, mutta tieto oli helposti saatavilla, eikä sen hyödyntämiseen olisi tarvittu monimutkaisempia NLP:n keinoja.
- Mallista M rakennetun graafin solmujen välisien suhteiden mallintamiseen käytetyt kaavat (3.2) ja (3.3) ovat suhteellisen hyvin vaatimusten mukaan rakennettuja, mutta ovat ongelman käsittelyn kannalta hankalia, sillä ne kertovat graafin arvojen propagoinnista vain yhteen suuntaan ja ovat hankalasti invertoitavia.
- Matriisin B päivittämiseen käytettävä kaava (4.9) on aivan liian naiivi. Vastaavanlainen menetelmä voisi toimia monipuolisemmalla lähtödatalla, mutta nykyisellä kaoottisella datalla kaava tuottaa liian voimakkaita muutoksia matriisiin B.
- Osaamistiedon epävarmuutta ei saatu lähtödatan pohjalta johdettua kunnolla, mutta siihen liittyvissä kaavoissa olisi myös paljon parannettavaa. Osaamisarvot T eivät muutu matriisin B päivityksen ongelmien vuoksi riittävästi, jotta kaava (4.7) tuottaisi järkeviä tuloksia.

7.3 Jatkotutkimus

Ehkä merkittävin parannuksen kohde käytetyssä menetelmässä on tekstin analysointi sekä viestin lähettäjän roolin tunnistaminen. Keskustelut noudattavat yleensä jonkinlaista selkeää rakennetta, joka on tunnistettavissa kielen analysoinnin keinoilla. Tätä rakennetta voitaisiin mallintaa esimerkiksi samaan tapan kuin Memphisin yliopistolle tehdyssä tutkimuksessa peliympäristössä käytyjen keskustelujen rakenneanalyysistä, jossa käytettiin Naiivi Bayes -klassifikaatiota sanojen ominaisuuksien tunnistamiseen [19, ss. 162-167]. Ilman keskustelun rakenteen tunnistamistakin yksittäisten viestien tarkoituksen tulkinta voisi auttaa ilmaisemaan kuka on keskustelussa kysyjän ja kuka asiantuntijan roolissa. Tämä auttaisi aikaisemmin mainittujen vastatodisteiden muodostamisessa.

Algoritmi voisi myös tarkastella tarkemmin henkilön altistumisaikaa termille pelkästään frekvenssin sijaan. Tarkastelemalla aikaeroa ensimmäisen ja viimeisen tokenin esiintymisen välillä kullekin henkilölle erikseen voitaisiin päätellä enemmän henkilön toiminnasta kyseisen aiheen parissa. Vallitsevan käsityksen mukaan tämä aika voisi korreloida henkilön taitotasoon, joskin ajatusta on myös kritisoitu paljon ja henkilön taitotason kehittymistä on yhdistetty vahvemmin henkilökohtaisiin ominaisuuksiin [16].

Yksittäisissä funktioissa ja vakioissa on myös paljon parannettavaa saatujen tulosten pohjalta. Useaa kohtaa prosessista voidaan parantaa asiantuntijatietoa lisäämällä, esimerkiksi määrittelemällä keskusteluhuoneet tarkemmin ja parantamalla tietoa synonyymeistä. Algoritmille voisi myös asettaa enemmän rajoitteita, ettei se esimerkiksi pystyisi merkitsemään yhä henkilöä jokaisen alan ekspertiksi.

LÄHTEET

[1] Timonen, M. Analyzing Knowledge Management in a Software Consulting Company. Tutkielma. Aalto University, syyskuu 2018.

[2] *Futurice Oy*. 3. lokakuuta 2015. URL: https://tietopalvelu.ytj.fi/yritystiedot.aspx?yavain=1395003&tarkiste=7C19C5245A7222E245A6C1DC2C1D9EA07A31DC4A (viitattu 12. 03. 2020).

[3] *Futurice about*. URL: https://www.futurice.com/about (viitattu 12. 03. 2020).

[4] McCandless, M. *Lucene in Action: Covers Apache Lucene V. 3. 0*. eng. 2nd ed. Place of publication not identified: Manning Publications Company. ISBN: 1-933988-17-7.

[5] Uszkoreit, H. Shallow Language Processing, Deep Language Processing and Domain Ontologies. *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering : (IEEE NLP-KE'05) : Oct. 30-Nov. 1, 2005, Wuhan, China*. IEEE. ISBN: 1-5386-0245-8.

[6] Desmarais, M. C. ja Baker, R. S. J. User Model User-Adap Inter (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. Springer Science+Business Media, 9. URL: https://doi.org/10.1007/s11257-011-9106-8.

[7] Das, B. *Representing Uncertainties Using Bayesian Networks*. DSTO-TR-0918. DSTO Electronics ja Surveillance Research Laboratory, 1999.

[8] Joyce, J. Bayes' Theorem. *The Stanford Encyclopedia of Philosophy*. Toim. E. N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019.

[9] Darwiche, A. *Modeling and Reasoning with Bayesian Networks*. 1. painos. Cambridge University Press, 2009. ISBN: 9780511501524.

[10] *A simple Bayesian network with conditional probability tables. A simple Bayesian network*. 16. syyskuuta 2006. URL: https://en.wikipedia.org/wiki/Bayesian_network#/media/File:SimpleBayesNet.svg (viitattu 07. 11. 2019).

[11] *Normal Distribution*. 17. helmikuuta 2020. URL: https://mathworld.wolfram.com/NormalDistribution.html (viitattu 09. 03. 2020).

[12] *Probability density function for the normal distribution*. 2. huhtikuuta 2008. URL: https://en.wikipedia.org/wiki/Normal_distribution#/media/File:Normal_Distribution_PDF.svg (viitattu 09. 03. 2020).

[13] *What does tf-idf mean?* URL: http://www.tfidf.com (viitattu 08. 03. 2020).

[14] *TF IDF | TFIDF Python Example*. 5. toukokuuta 2019. URL: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76 (viitattu 08. 03. 2020).

[15] Nasini, S. *Notes on Bayesian Networks*. URL: http://www-eio.upc.es/~nasini/Blog/BayesianNetworks.pdf (viitattu 09. 05. 2020).

[16] Hambrick, D. Z., Altmann, E. M., Oswald, F. L., Meinz, E. J., Gobet, F. ja Campitelli, G. Accounting for expert performance: The devil is in the details. eng. *Intelligence* 45.1 (2014), 112–114. ISSN: 0160-2896.

[17] Zhang, Y., Wu, H. ja Cheng, L. Some new deformation formulas about variance and covariance. eng. *2012 Proceedings of International Conference on Modelling, Identification and Control*. IEEE, 2012, 987–992. ISBN: 9781467315241.

[18] Upton, G. ja Cook, I. *Bessel correction*. eng. 2014. URL: http://www.oxfordreference.com/view/10.1093/acref/9780199679188.001.0001/acref-9780199679188-e-1824.

[19] *Intelligent Tutoring Systems 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings*. eng. 1st ed. 2012. Lecture notes in computer science, 7315. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 3-642-30950-X.

A POWER-JÄRJESTELMÄN KATEGORIAT

Power-järjestelmän kategoriat raakadatasta poimittuna. Kategoriat muodostavat hierarkian siten, että jokaisella kategorialla voi olla yksi vanhempi kategoria tai ei yhtään vanhempaa kategoriaa.

ID	Nimi	Vanhemman kategorian ID
1	Desktop Apps	
2	Electron	1
3	WPF C#	1
4	WinForms C#	1
5	Win32 (C/C++)	1
6	Delphi	1
7	Lazarus FPC	1
8	Mac OS Objective-C	1
9	Mac OS Swift	1
10	Visual Basic Classic	1
11	WPF VB.Net	1
12	WinForms VB.Net	1
13	Windows Phone Native	
14	C#	13
266	Blockchain dev	
16	Strategy	15
17	Operating Model	15
18	Architecture	15
19	Language	
22	System integration	
23	Salesforce	22
24	Sap	22
25	Dynamics	22
26	Project management	
27	PM	26
28	Web backend	
29	Elixir	28
30	Spring a	28
31	Spring Boot	28
32	Grunt	28
33	C#	28
34	ASP.Net	28
35	Clojure	28

36	Ruby on Rails	28
37	Golang	28
38	Haskell	28
39	PHP	28
40	Gulp	28
41	C++	28
42	C	28
43	Django	28
44	Scala	28
45	Java	28
46	Play Framework	28
47	Node.js	28
48	Python	28
49	Redis	28
50	TypeScript	28
51	GraphQL	28
52	Rust	28
53	Server-side Swift	28
54	Web frontend	
55	ClojureScript	54
56	TypeScript	54
57	PureScript	54
58	MobX	54
59	Angular	54
60	Mocha	54
61	Karma	54
62	Selenium	54
63	Grunt	54
64	Elm	54
65	Redux	54
66	React	54
67	Webpack	54
68	JavaScript	54
69	CSS	54
70	Vue.js	54
71	Vuex	54
72	Leaflet	54
73	Angular 2	54
74	d3.js	54
75	Cycle	54
76	Jest	54
77	Mentoring	
78	Career path coaching	77
79	Sales	77
80	Ways of working	77
81	Software development	77

82	Frontend web development	77
83	Backend web development	77
84	iOS	77
85	Android	77
86	Leadership	77
87	IoT	
88	Arduino	87
89	Embedded software	87
90	Embedded Linux	87
91	Platforms	87
92	Business development	87
93	React Native	
94	iOS	93
95	Android	93
96	Android Native	
97	Java	96
98	Kotlin	96
99	Databases	
100	MS SQL	99
101	Oracle	99
102	MySQL	99
103	Redis	99
104	Cassandra	99
105	PostgreSQL	99
106	MongoDB	99
107	NoSQL	99
108	Elasticsearch	99
109	Analytics	
110	Service Analytics	109
111	Advisory	
112	Current state analysis	111
113	Service vision sprint	111
114	Business models	111
115	Facilitation	111
116	Change management	111
117	Opportunity roadmapping	111
118	Operating models	111
119	Culture change	111
120	Growth hacking	111
121	Ecosystems	111
122	Capability development	111
123	Lean Service Creation	111
124	Solution Architect	111
125	Data science	
126	Machine learning	125
127	Computer vision	125

128	Data visualisation	125
129	Data analysis	125
130	Biometrics	125
131	Recruitment	
132	Interviews	131
133	Technical Interviews	131
134	Hybrid Mobile	
135	Cordova	134
136	Unity	134
137	Business Director	
138	Tech	137
139	Design	137
140	Sales	137
141	Senior Chief	137
142	Operative	137
143	Agile	
144	Scrum Master	143
145	Product Owner	143
146	Agile at Scale Consulting	143
21	German	19
20	Finnish	19
147	User Research	
148	DevOps	
149	Kubernetes	148
150	Docker	148
151	Kontena	148
152	Jenkins	148
153	Travis	148
154	Bamboo	148
155	TeamCity	148
156	AWS	148
157	Azure	148
158	Ansible	148
159	API management	148
160	Heroku	148
161	CircleCI	148
162	Visual Studio Team Services	148
163	Datadog	148
164	Sumologic	148
165	Prometheus	148
166	Google Cloud Platform	148
167	Terraform	148
168	QA	
169	Exploratory Testing	168
170	Load Testing	168
171	Performance Testing	168

172	Security Testing	168
173	Test Automation - Robot FW	168
174	Test Automation - Selenium	168
175	Quality Center	168
176	Test Management	168
177	Test Lead	168
178	Test Planning	168
179	User Testing	168
180	Error Management	168
181	Test Automation - Appium	168
182	Design	
183	Graphic / Visual Design	182
184	Service Design	182
185	User Insights	182
186	Branding Design	182
187	Motion Design	182
188	UX Design	182
189	Strategy Design	182
190	Interaction Design	182
191	Industrial services	182
192	Consumer IoT services	182
193	Sound Design	182
194	Game Design	182
195	3D Modeling	182
196	Physical UX Design	182
197	AR Design	182
198	VR Design	182
199	3D CAD	182
200	Xamarin	
201	Forms	200
202	Android	200
203	iOS	200
204	iOS Native	
205	Objective-C	204
206	Swift	204
207	Account Management	
208	Sales	207
209	CMS	
210	Drupal	209
211	WordPress	209
212	Contentful	209
213	EPIServer	209
214	Prismic.io	209
215	Tech	
216	Lean Service Creation (LSC)	215
267	Ethereum	266

217	Accessibility	
15	Tech Advisory	
219	Kafka	218
218	Data Engineering	
220	Hadoop	218
221	Spark	218
222	Splunk	218
223	Business Intelligence	
224	Tableau	223
225	Power BI	223
226	Natural Language Processing (NLP)	125
228	Big Data	99
227	Time series	99
229	Extract, Transform, Load (ETL)	99
230	Accessibility	217
231	F#	81
232	Next.js	54
233	Kotlin	28
234	Swedish	19
235	Synthetic data	125
268	Solidity	266
269	LDAP	254
238	Generic	
237	Presenting	238
236	Professional Writing	238
239	bacon.js	54
240	Flutter	
241	Android	240
242	iOS	240
245	Design sprint	111
246	Print Design	182
244	Photography	182
247	Illustration	183
248	Dhall	148
243	Video production	182
251	UX Animation	182
252	French	19
253	Information Architecture	182
254	Authentication	
255	Auth0	254
256	DB2	99
257	RxJava	28
258	BigQuery	228
259	Emerging Tech	
260	Augmented Reality	259
261	Virtual Reality	259

262	Facial Recognition	259
263	Voice Recognition	259
264	Immersive Visual Tech	259
265	Raspberry Pi	87

B SYNONYYMIT

Graafisen verkon termien a_i ja tekstidatassa esiintyvien tokeneiden t_j välille algoritmin alussa luodut ylimääräiset yhteydet matriisissa B. Suhdearvo mittaa yhteyden voimakkuutta, jolloin arvo 1 on täydellinen synonyymi ja arvo 0 tarkoittaa täysin toisiinsa liittymättömiä konsepteja.

Termi	Tokeni	Suhdearvo
3d cad	cad	0.60
3d cad	3d	0.05
3d modeling	3d	0.05
3d modeling	modeling	0.02
agile at scale consulting	agile at scale	0.50
postgresql	psql	1.00
postgresql	postgres	1.00
postgresql	sql	0.50
ms sql	sql	0.50
oracle	sql	0.20
mysql	sql	0.50
nosql	sql	0.20
elasticsearch	es	0.30
blockchain dev	blockchain	0.40
test automation - robot fw	robot fw	0.90
test automation - appium	appium	0.90
test automation - selenium	selenium	0.90
test automation - robot fw	test automation	0.20
test automation - appium	test automation	0.20
test automation - selenium	test automation	0.20
natural language processing (nlp)	natural language processing	1.00
natural language processing (nlp)	nlp	0.90
extract, transform, load (etl)	extract transform load	1.00
extract, transform, load (etl)	etl	0.90
ruby on rails	rails	0.90
ruby on rails	ruby rails	1.00
mac os objective-c	objective-c	0.40
mac os swift	swift	0.40
server-side swift	swift	0.40
visual basic classic	visual basic	0.80
visual basic classic	vbc	0.60
visual basic classic	vb.net	0.20

windows phone native	windows phone	0.20
windows phone native	native	0.01
electron	electron.js	1.00
web frontend	frontend	0.70
frontend web development	frontend	0.60
web backend	backend	0.70
backend web development	backend	0.60
node.js	node	0.40
vue.js	vue	0.70
winforms c#	winforms	0.60
winforms c#	c#	0.10
winforms vb.net	winforms	0.60
winforms vb.net	vb.net	0.10
wpf c#	wpf	0.60
wpf c#	c#	0.10
wpf vb.net	wpf	0.60
wpf vb.net	vb.net	0.10
win32 (c/c++)	win32	0.60
win32 (c/c++)	c	0.10
win32 (c/c++)	c++	0.10
machine learning	ml	0.20
embedded software	embedded	0.10
embedded linux	embedded	0.10
embedded linux	linux	0.01
facial recognition	recognition	0.10
voice recognition	recognition	0.10
augmented reality	ar	0.40
virtual reality	vr	0.20
recruitment	interview	0.10
technical interviews	interview	0.05
physical ux design	ux design	0.70
physical ux design	ux	0.20
service design	service	0.05
graphic / visual design	graphic design	0.90
graphic / visual design	visual design	0.90
graphic / visual design	graphic	0.05
graphic / visual design	visual	0.10
sound design	sound	0.05
ux design	ux	0.60
vr design	vr	0.30
print design	print	0.02
motion design	motion	0.01
lean service creation	lsc	0.90
lean service creation (lsc)	lean service creation	0.90
lean service creation (lsc)	lsc	0.90
career path coaching	career path	0.05

opportunity roadmapping	roadmapping	0.20
immersive visual tech	immersive	0.02
immersive visual tech	visual tech	0.10
lazarus fpc	lazarus	0.60
lazarus fpc	fpc	0.20
ux animation	animation	0.02
ux animation	ux	0.05
industrial services	industrial	0.02
industrial services	services	0.01
industrial services	industry	0.01
consumer iot services	consumer	0.01
consumer iot services	iot services	0.20
consumer iot services	iot	0.05
branding design	branding	0.30

C TYÖSSÄ KÄYTETYN VERKON RAKENNE

Työssä käytetyn verkon rakenne. Verkon solmut muodostavat hierarkian siten, että jokaisella termiä vastaavalla solmulla voi olla yksi vanhempi solmu tai ei yhtään vanhempaa solmua. Jokaisella vanhempisuhteella on painoarvo β, joka kuvaa suhteen voimakkuutta.

ID	Termi	Vanhemman kategorian ID	Vanhempisuhteen painoarvo
1	desktop apps	272	0.1
2	electron	1	0.4
3	wpf c#	1	0.4
4	winforms c#	1	0.4
5	win32 (c/c++)	1	0.4
6	delphi	1	0.4
7	lazarus fpc	1	0.4
8	mac os objective-c	1	0.4
9	mac os swift	1	0.4
10	visual basic classic	1	0.4
11	wpf vb.net	1	0.4
12	winforms vb.net	1	0.4
13	windows phone native	270	0.2
14	c#	13	0.4
266	blockchain dev	272	0.1
16	strategy	15	0.4
17	operating model	15	0.4
18	architecture	15	0.4
19	language		
22	system integration	272	0.1
23	salesforce	22	0.4
24	sap	22	0.4
25	dynamics	22	0.4
26	project management	273	0.2
27	pm	26	0.4
28	web backend	271	0.2
29	elixir	28	0.4
30	spring	28	0.4
31	spring boot	28	0.4
32	grunt	28	0.4
33	c#	28	0.4
34	asp.net	28	0.4
35	clojure	28	0.4

36	ruby on rails	28	0.4
37	golang	28	0.4
38	haskell	28	0.4
39	php	28	0.4
40	gulp	28	0.4
41	c++	28	0.4
42	c	28	0.4
43	django	28	0.4
44	scala	28	0.4
45	java	28	0.4
46	play framework	28	0.4
47	node.js	28	0.4
48	python	28	0.4
49	redis	28	0.4
50	typescript	28	0.4
51	graphql	28	0.4
52	rust	28	0.4
53	server-side swift	28	0.4
54	web frontend	271	0.2
55	clojurescript	54	0.4
56	typescript	54	0.4
57	purescript	54	0.4
58	mobx	54	0.4
59	angular	54	0.4
60	mocha	54	0.4
61	karma	54	0.4
62	selenium	54	0.4
63	grunt	54	0.4
64	elm	54	0.4
65	redux	54	0.4
66	react	54	0.4
67	webpack	54	0.4
68	javascript	54	0.4
69	css	54	0.4
70	vue.js	54	0.4
71	vuex	54	0.4
72	leaflet	54	0.4
73	angular 2	54	0.4
74	d3.js	54	0.4
75	cycle	54	0.4
76	jest	54	0.4
77	mentoring		
78	career path coaching	77	0.4
79	sales	77	0.4
80	ways of working	77	0.4
81	software development	77	0.4

82	frontend web development	77	0.4
83	backend web development	77	0.4
84	ios	77	0.4
85	android	77	0.4
86	leadership	77	0.4
87	iot	272	0.1
88	arduino	87	0.4
89	embedded software	87	0.4
90	embedded linux	87	0.4
91	platforms	87	0.4
92	business development	87	0.4
93	react native	270	0.2
94	ios	93	0.4
95	android	93	0.4
96	android native	270	0.2
97	java	96	0.4
98	kotlin	96	0.4
99	databases	272	0.1
100	ms sql	99	0.4
101	oracle	99	0.4
102	mysql	99	0.4
103	redis	99	0.4
104	cassandra	99	0.4
105	postgresql	99	0.4
106	mongodb	99	0.4
107	nosql	99	0.4
108	elasticsearch	99	0.4
109	analytics	272	0.05
110	service analytics	109	0.4
111	advisory		
112	current state analysis	111	0.4
113	service vision sprint	111	0.4
114	business models	111	0.4
115	facilitation	111	0.4
116	change management	111	0.4
117	opportunity roadmapping	111	0.4
118	operating models	111	0.4
119	culture change	111	0.4
120	growth hacking	111	0.4
121	ecosystems	111	0.4
122	capability development	111	0.4
123	lean service creation	111	0.4
124	solution architect	111	0.4
125	data science	272	0.1
126	machine learning	125	0.4
127	computer vision	125	0.4

128	data visualisation	125	0.4
129	data analysis	125	0.4
130	biometrics	125	0.4
131	recruitment		
132	interviews	131	0.4
133	technical interviews	131	0.4
134	hybrid mobile	270	0.2
135	cordova	134	0.4
136	unity	134	0.4
137	business director	273	0.2
138	tech	137	0.4
139	design	137	0.4
140	sales	137	0.4
141	senior chief	137	0.4
142	operative	137	0.4
143	agile	273	0.1
144	scrum master	143	0.4
145	product owner	143	0.4
146	agile at scale consulting	143	0.4
21	german	19	0.4
20	finnish	19	0.4
147	user research		
148	devops	272	0.1
149	kubernetes	148	0.4
150	docker	148	0.4
151	kontena	148	0.4
152	jenkins	148	0.4
153	travis	148	0.4
154	bamboo	148	0.4
155	teamcity	148	0.4
156	aws	148	0.4
157	azure	148	0.4
158	ansible	148	0.4
159	api management	148	0.4
160	heroku	148	0.4
161	circleci	148	0.4
162	visual studio team services	148	0.4
163	datadog	148	0.4
164	sumologic	148	0.4
165	prometheus	148	0.4
166	google cloud platform	148	0.4
167	terraform	148	0.4
168	qa	272	0.1
169	exploratory testing	168	0.4
170	load testing	168	0.4
171	performance testing	168	0.4

172	security testing	168	0.4
173	test automation - robot fw	168	0.4
174	test automation - selenium	168	0.4
175	quality center	168	0.4
176	test management	168	0.4
177	test lead	168	0.4
178	test planning	168	0.4
179	user testing	168	0.4
180	error management	168	0.4
181	test automation - appium	168	0.4
182	design		
183	graphic / visual design	182	0.4
184	service design	182	0.4
185	user insights	182	0.4
186	branding design	182	0.4
187	motion design	182	0.4
188	ux design	182	0.4
189	strategy design	182	0.4
190	interaction design	182	0.4
191	industrial services	182	0.4
192	consumer iot services	182	0.4
193	sound design	182	0.4
194	game design	182	0.4
195	3d modeling	182	0.4
196	physical ux design	182	0.4
197	ar design	182	0.4
198	vr design	182	0.4
199	3d cad	182	0.4
200	xamarin	270	0.2
201	forms	200	0.4
202	android	200	0.4
203	ios	200	0.4
204	ios native	270	0.2
205	objective-c	204	0.4
206	swift	204	0.4
207	account management	273	0.1
208	sales	207	0.4
209	cms	272	0.1
210	drupal	209	0.4
211	wordpress	209	0.4
212	contentful	209	0.4
213	episerver	209	0.4
214	prismic.io	209	0.4
215	tech	272	0.2
216	lean service creation (lsc)	215	0.4
267	ethereum	266	0.4

217	accessibility	272	0.1
15	tech advisory	111	0.2
219	kafka	218	0.4
218	data engineering	272	0.1
220	hadoop	218	0.4
221	spark	218	0.4
222	splunk	218	0.4
223	business intelligence	273	0.1
224	tableau	223	0.4
225	power bi	223	0.4
226	natural language processing (nlp)	125	0.4
228	big data	99	0.4
227	time series	99	0.4
229	extract, transform, load (etl)	99	0.4
231	f#	81	0.4
232	next.js	54	0.4
233	kotlin	28	0.4
234	swedish	19	0.4
235	synthetic data	125	0.4
268	solidity	266	0.4
269	ldap	254	0.4
238	generic		
237	presenting	238	0.4
236	professional writing	238	0.4
239	bacon.js	54	0.4
240	flutter	270	0.2
241	android	240	0.4
242	ios	240	0.4
245	design sprint	111	0.4
246	print design	182	0.4
244	photography	182	0.4
247	illustration	183	0.4
248	dhall	148	0.4
243	video production	182	0.4
251	ux animation	182	0.4
252	french	19	0.4
253	information architecture	182	0.4
254	authentication	272	0.1
255	auth0	254	0.4
256	db2	99	0.4
257	rxjava	28	0.4
258	bigquery	228	0.4
259	emerging tech	272	0.1
260	augmented reality	259	0.4
261	virtual reality	259	0.4
262	facial recognition	259	0.4

263	voice recognition	259	0.4
264	immersive visual tech	259	0.4
265	raspberry pi	87	0.4
270	mobile development	272	0.1
271	web development	272	0.1
272	technology		
273	leadership		