Informaatioteknologian tiedekunnan julkaisuja
No. 69/2018

Riku Nyrhinen, Pekka Neittaanmäki

Tekstin automaattisesta tiivistämisestä

JYVÄSKYLÄN YLIOPISTO

Informaatioteknologian tiedekunnan julkaisuja
No. 69/2018

Editor: Pekka Neittaanmäki
Covers: Petri Vähäkainu ja Matti Savonen

ISBN 978-951-39-7654-5 (verkkoj.)
ISSN 2323-5004

Jyväskylä 2018

Tekstin automaattisesta tiivistämisestä

Riku Nyrhinen
Pekka Neittaanmäki

Tämä julkaisu on toteutettu osana WHC-hanketta, johon Jyväskylän yliopisto on saanut rahoituksen Business-Finlandilta.

Business Finland-hanke: WHC

KUVIOT

SISÄLLYSLUETTELO

1 Johdanto

Tässä raportissa esitellään Watson Health Cloud Finland -hankkeen aikana hyödynnettyjä tekstin automaattisen tiivistelmän metodeja ja käydään läpi niiden soveltamista suomenkielisen tekstimuotoisen sosiaali- ja terveydenhuoltoaineistoon. Suomenkielisissä teksteissä käytetään usein nimeä "tekstin automaattinen tiivistäminen" (eng.: *automatic text summarization*). Tässä tekstissä termit lyhennelmä, tiivistelmä ja yhteenveto ovat keskenään vaihtokelpoisia, ellei toisin mainita.

Saatavilla olevan tekstimuotoisen aineiston määrä kasvaa alati meitä ympäröivän informaatioyhteiskunnan tuotoksien lisääntyessä. Oleellisen kiteyttäminen tekstidatan joukosta tulee tulevaisuudessa olemaan vielä nykyistäkin tärkeämpää. Tekstin automaattisen tiivistämisen pyrkimyksenä on tuottaa lähdeaineistosta helposti luettava ja ymmärrettävä, ytimekäs lyhennelmä tai yhteenveto, joka välittää lukijalleen lähdeaineiston pääkohdat. Lähdeaineisto voi koostua yhdestä tai useammasta tekstimuotoisesta dokumentista. Lähdeaineiston eriteltävät ominaisuudet, kuten sen pituus, kohdeyleisö, aihe tai rakenne voivat antaa arvokasta informaatiota tiivistämisprosessia varten, jos lähdeaineisto tunnetaan etukäteen tai se on tutkittavissa ennen lyhennelmiä tekevän järjestelmän toteuttamista. Täysin yleinen tiivistäjä, joka tuottaa järkeviä tuloksia riippumatta lähdeaineistosta, on hankala, ellei jopa mahdoton toteuttaa (Spärck, 1999; Spärck Jones, 2001).

Tiivistämisprosessin tyypillinen kulku noudattaa seuraavia askeleita:

0. Lähdeaineiston hankkiminen
1. Aineiston esikäsitteleminen
2. Aineiston esittäminen kokoelmana rakenteita
    - esitetään aineisto listana sanoja ja välimerkkejä, lauseita tai virkkeitä
3. Rakenteiden asettaminen tärkeysjärjestykseen pyrkimyksen mukaisesti
    - lyhennelmiä luodessa yleensä lähdeaineiston sisältöä parhaiten kuvaavat virkkeet
4. Rakenteiden valitseminen tiivistelmään
    - lyhennelmiä luodessa pyritään vähentämään redundanssia eli turhaa toistoa ja kuvaamaan lähdeaineiston sisältö mahdollisimman lyhyesti ja ytimekkäästi
5. Tiivistelmän kokoaminen rakenteiden avulla

Tämän raportin pääpaino on kohdilla 2, 3, ja 4. Kohdista 1 ja 5 puhutaan enemmän kappaleessa 5.

Tekstin automaattisen tiivistämisen metodit jaetaan usein dikotomisesti kahtia erittelevi in ja abstrahoiviin (*extractive and abstractive*) metodeihin (Spärck Jones, 2007). Ensimmäinen perustuu puhtaasti lähdeaineistoon eikä vaadi välttämättä ollenkaan tuntemusta lähde- tai kohdekielestä. Erittelevässä tiivistämisessä lopullisen tiivistelmä sisältö poimitaan lähes sanasta sanaan tai virkkeestä virkkeeseen lähdeaineistosta käyttäen menetelmiä, joilla painotetaan lähdeaineiston rakenneosasten merkitys halutun tuloksen kannalta. Jälkimmäinen kertoo lähdeaineiston pääkohdat "omin sanoin" luoden uutta sisältöä annetun pohjalta eli se ei nojaa toteutuksessaan yhtä paljon lähdeaineiston tekstiin kuin erittelevät menetelmät. Sen oletetaan pystyvän tuottamaan kieliopillisesti oikeaa tekstiä, jota on järkevää ja helppoa lukea. Abstrahointi on kuitenkin vaikea toteuttaa, sillä tiivistelmiä tuottavan systeemin tulisi omata korkean tason ymmärrys kohdekielen toiminnasta. Suurin osa tutkimusaineistosta on juuri tämän takia keskittynyt erittelevään menetelmään. Tulevaisuudessa abstrahoiva tiivistämisen uskotaan kuitenkin kasvattavan merkitystään (Mani, 2001). Tässä raportissa esitellään erittelevi menetelmiä ja jätetään abstrahointi kokonaan välistä sen aiheesta poikkeavien haasteiden vuoksi.

1.1 Kontekstuaalisuus ja tiivistelmän tarkoitus

Spärck Jones (2007) listaa tutkimuksissaan kolme kategoriaa, jotka voidaan huomioida tiivistäessä ja jotka täten vaikuttavat tiivistäjän suunnitteluun ohjelmistotasolla. Kategoriat ovat **input factors**, **purpose factors** ja **output factors**. Ne sisältävät ominaisuuksia, jotka kertovat (tässä järjestyksessä), millainen teksti on luonteeltaan, mihin tiivistelmää käytetään ja miten tiivistelmä koostetaan.

Esimerkiksi **input factors** -luokkaan kuuluvat muun muassa tekstin tyyli (uutisteksti, runo vai akateeminen julkaisu), rakenne (otsikot, tilastolaatikot, kuvat), pituus, yksi- tai monidokumenttisuus, metadata (avainsanat, indeksointiperiaate). **Purpose factors** -luokka huomioi esimerkiksi tiivistelmän kohdeyleisön (akateeminen yleisö vai Matti Meikäläinen) tai kohteen (tietokone vai ihminen). **Output factors** -luokkaan kuuluvat muun muassa synnytettävän tiivistelmän tyyli, standardisoitu rakenne, kieli, kielellinen tyyli, rakenne (muotoillaanko lähdeaineiston mukaan vai vapaasti), kattavuus (yleinen vai tietystä aiheesta) ja lyhennelmän maksimipituus.

Tiivistämisen kaksi peruskysymystä kuitenkin ovat: (i) kuinka valita keskeinen sisältö lähdeaineistosta, ja (ii) kuinka ilmaista valittu sisältö tiivistetyssä muodossa? (Spärck Jones, 1993).

1.2 Millainen on hyvä tiivistelmä?

Erilaisilla tiivistelmillä pyritään erilaisiin pyrkimyksiin, mistä voidaan päätellä, että myös tiivistelmän laadun arvioinnissa pyritään vertailtaviin tuloksiin erilaisten määritelmien kautta. Yleisesti on hankala sanoa, millainen on hyvä tiivistelmä. Sen pitäisi pyrkiä kuvaamaan lähdeaineisto pääkohdittain niin, että aiheesta jäisi lukijalle pääpiirteittäinen kuva, mutta toisaalta lyhennelmän täytyisi olla tiivis, lyhyt paketti, joka on helposti sisäistettävissä. Kielen tulisi olla helposti ymmärrettävää ja soljuvaa välttäen rönsyilyä, ylimääräisiä täytesanoja ja saman toistoa.

Konferenssit, kuten SUMMAC, DUC ja TAC, ovat myös kilpailuja, joissa haetaan parasta automaattista tiivistäjää (Mani, Klein, House, Hirschman, Firmin & Sundheim, 2002; Over, Dang & Harman, 2007; Dang, 2008). Tällöin on oltava arviointiperuste. Tulokset on usein tarkistettu käsin ammattilaisten toimesta, mutta esimerkiksi DUC- ja TAC-kilpailuissa ammattilaisia pyydettiin arvioimaan numeroin referenssitiivistelmiä, joihin osallistujien tiivistelmiä verrattiin, seuraavin perustein (Dang & Harman, 2007):

1. kieliasu (kielioppi, sanojen oikea käyttäminen, jne.)
2. redundanssi
3. fokus aihepiiriin
4. rakenne ja ymmärrettävyys

Kohdat pisteytettiin skaalalla 0- 10 (huonoimmasta parhaimpaan) ja TAC2009-konferenssissa hyödynnetyt referenssitiivistelmät saivat keskimääräiseksi arvosanakseen 8,8. Tiivistelmäkohtaisia pisteitä ei eritelty tarkemmin (Dang & Owczarzak, 2008).

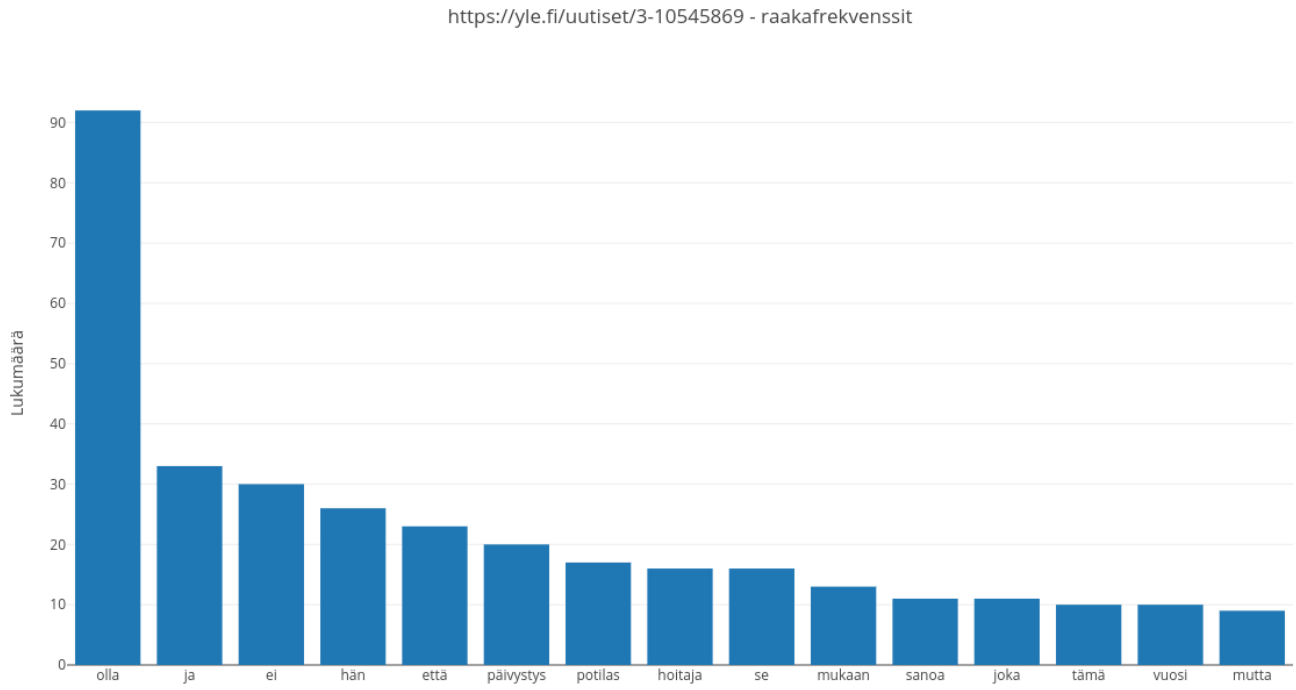
Koska ihmisetkään eivät ole yhtämielisiä siitä, millainen on hyvä tiivistelmä, on vaikea lähteä luomaan automaattista evaluoijaa, joka toimisi riittävän hyvin tilanteesta riippumatta. Nykyisiä koneoppimismalleja ja käsivoimin annotoituja referenssitiivistelmiä hyödyntäen ollaan päästy kelvollisiin tuloksiin, mutta lyhennelmän tarkoituksen ollessa erikoinen, tiivistettävän aiheen määritelmän spesifinen tai lähdeaineisto monidokumenttinen, evaluointitulos jää alle toivotun.

2 Aineiston kuvaaminen osiensa summana

Tässä kappaleessa kerrotaan, kuinka lähdeaineisto voidaan kuvata siinä esiintyvien puheenaiheiden tai merkittävyyttä osoittavien indikaattorien avulla. Käsiteltävät rakenneosaset ovat tyypillisesti virkkeitä, mutta sana- ja kappalekohtaisiakin lähestymistapoja on.

Aihe-esitystavassa pyritään selvittämään lähdeaineistosta löytyvien vihjeiden avulla, mitä aineisto käsittelee. Eräässä tekstin automaattisen tiivistämisen varhaisimmista julkaisuista Luhn esittää ratkaisuksi aihetunnusmerkkien keräämistä lähdeaineistosta (Luhn, 1957). Aihetunnusmerkki tarkoittaa sanaa, jolla on vahva yhteys lähdeaineiston aiheeseen tai joka kuvaa lähdeaineiston aihetta hyvin. Luhn'in lähestymistapa nojaa frekvenssirajoihin ja englannin kielen yleisten sanojen sivuuttamiseen. Lähestysmistapa vaikuttaa nykypäivänä kömpelöltä ja on selvästi kieliriippuvainen. Frekvenssien hyödyntäminen niiden selkeän tulkinnallisuuden vuoksi oli kuitenkin idea, joka sai paljon huomiota myöhemmässä kirjallisuudessa.

Sanafrekvenssit kertovat aineistosta paljon, sillä usein esiintyvien sanojen oletetaan olevan merkittäviä tekstin aiheen kannalta. Raakafrekvenssi eli täysin painottamaton frekvenssi ei välttämättä ole paras tapa tarkastella aineistoa, sillä raakafrekvenssien käyttäminen nostaa kielen yleisimmät sanat kärkeen. Sanat kuten "olla", "ei", "että" tai "ja" ovat merkittäviä yhdistettynä kontekstiin, mutta omillaan ne eivät kerro mitään lähdeaineiston aihesisällöstä.



KUVIO 1. Ylen uutisartikkelin 15 yleisintä lemmaa järjestyksessä frekvenssin mukaan

Frekvenssejä voi painottaa useilla menetelmillä, joista yleisin lienee TF-IDF (*Term Frequency - Inverse Document Frequency*). TF-IDF huomioi sanan dokumenttikohtaisen frekvenssin lisäksi sanan frekvenssin koko aineistoon nähden. Jos lähdeaineisto on monidokumenttinen, useassa dokumentissa esiintyvät sanat, kuten aikaisemmin listatut, ovat merkittävyydeltään harvemmin dokumenteissa esiintyvien sanojen alapuolella. TF-IDF:n laskemiseen käytetään eri tilanteissa eri kaavoja, mutta seuraava on hyvin yleinen:

$$tf(t,d) = \frac{f(t)}{len(d)}$$

$$idf(t,D) = ln\left(\frac{N}{D(t)}\right)$$

$$tfidf = tf \cdot idf$$

jossa **f(t)** on termin **t** esiintymismäärä dokumentissa **d** ja **len(d)** dokumentin kaikkien sanojen määrä; **N** aineiston dokumenttien määrä ja **D(t)** niiden dokumenttien määrä, jotka sisältävät termin **t** (Salton & Buckley, 1988; Spärck Jones, 1972).

Kaavasta huomataan, että sanat, jotka esiintyvät useassa dokumentissa, saavat pienemmän numeroarvon kuin harvemmin esiintyvät, sillä **log_e(1) = 0**. Frekvenssimenetelmät ovat helppoja, sillä niiden käyttämiseen ei tarvita kuin keino erotella lähdeaineiston sanat toisistaan ja koostaa niistä virkkeitä ja lauseita. Tämän voi tehdä jo pelkästään säännöllisten lausekkeiden avulla tulkitsemalla välilyönnein erotellut kirjainryppäät sanoiksi ja pisteeseen päättyvät sanaketjut virkkeiksi.

Sentroidimenetelmä (*centroid*: painopiste) soveltaa TF-IDF-painotusta yhdessä empiirisesti määriteltävän kynnysarvon, **c**, kanssa. Rakenneosasten termit, joiden TF-IDF-arvo on pienempi kuin **c**, mielletään turhaksi kohinaksi, jolla ei ole arvoa dokumentin tiivistämisen kannalta. Tiivistämisen myöhemmässä vaiheessa rakenneyksiköiden arvojärjestystä selvittäessä suositaan osasia, jotka sisältävät mahdollisimman paljon TF-IDF-kynnysarvon ylittäviä termejä, jotka omaavat korkean TF-IDF-arvon muihin kynnysarvon ylittäviin termeihin nähden (Radev, Jing, Sty & Tam, 2004).

Sanojen painottamisen sijaan tai sen kanssa käytetään sanaketjuja, joiden avulla yritään muodostaa aihepiiri siihen liittyvien sanojen pohjalta (Barzilay & Elhadad, 1999; Silber & McCoy, 2002). Sanaketjut voivat yhdistää synonyymisanat tai samassa kontekstissa usein esiintyvät sanat toisiinsa. Tällainen tieto ei kuitenkaan ole helposti pääteltävissä aikaisemmin esitellystä säännöllisiä lausekkeita hyödyntävästä pilkkomistavasta, sillä sanojen erottelu ja kokoaminen virkkeisiin ei kerro sanojen merkityksestä tai niiden välisistä syntaktisista relaatioista mitään. Apuna voidaan käyttää nymiakirjastoja, kuten FinnWordNet'iä. Jatkuva hakujen suorittaminen valtaisalle tietokannalle muodostuu kuitenkin hyvin pian

ongelmalliseksi. Jos valmista tietokantaa ei haluta käyttää, voidaan lingvistisiä piirteitä hahmotella laskennallisesti esimerkiksi LSA:n (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990), word2vec-metodien (Mikolov, Chen, Corrado & Dean, 2013) tai Bayesilaisen (Daumé & Marcu, 2006) kielenmallinnuksen avulla. Nämä lähestymistavat vaativat kuitenkin tarpeeksi kattavan ja monipuolisen lähdeaineiston "oppiakseen".

Indikaattoriesitystapa kuvastaa lähdeaineiston rakenteet listana tärkeyttä osoittavia indikaattoreja, joita voivat olla muun muassa virkkeen pituus, sen sijainti dokumentissa (esimerkiksi uutistekstissä tärkein tulee yleensä ensin käänteisen pyramidin mallisesti), virkkeen yhteys toiseen virkkeeseen tai tiettyjen sanojen ja ilmaisujen esiintyvyys virkkeessä. Verkko-/graafiteorian hyödyntäminen tiivistämisen apuna on poikkeuksellinen indikaattoriesitystavan osa-alue, sillä se käyttää yleisimmässä sovelluksessaan hyväkseen vain yhtä indikaattoria rakenneosasten tärkeyden selvittämiseen. Tämä indikaattori on rakenneosasten keskinäisyys (*centrality*) lähdeaineistosta muodostetussa graafissa. Verkkoteoria yleistyi tekstin automaattisessa tiivistämisessä PageRank-algoritmin (Erkan & Radev, 2004; Mihalcea & Tarau, 2004) myötä, mutta vastaavia ratkaisuja oli kokeiltu jo aiemminkin. Tavallisempaa on kuitenkin käyttää useampaa indikaattoria -- yleensä koneoppimisalgoritmien avustamana.

Verkkoteorian suomassa lähestymistavassa lähdeaineistosta muodostetaan verkko, jonka solmut ovat lähdeaineiston rakenneosasia, yleensä kokonaisia virkkeitä, ja rakenneosasten välisille kaarille määritellään painoarvo kaaren päissä olevien rakenneosasten similaarisuuden perusteella. Eräs tapa on muodostaa virkkeen sanoista TF-IDF-painotettu vektori ja verrata kahden virkevektorin samankaltaisuutta hyödyntäen kosinisimilaarisuutta (Chali & Joty, 2008). Laskennallisesti helpompi tapa on yhdistää vain sellaiset solmut, jotka ovat riittävän vahvasti kytköksissä toisiinsa eli ts. niiden samankaltaisuus ylittää tietyn kynnysarvon. Tällöin toisiinsa kytkeytyneet solmuryppäät ovat todennäköisesti keskeisiä lähdeaineiston sisällön kannalta. Jos kaarille määritellyt painoarvot muutetaan todennäköisyysjakaumiksi, saadaan verkosta Markovin ketju, jolle voidaan soveltaa sille ominaisia ratkaisumenetelmiä optimaalisimman tiivistelmän tuottamiseksi. Verkkoteorian on todettu toimivan hyvin sekä yksi- että monidokumenttisilla lähdeaineistoilla (Erkan & Radev, 2004; Mihalcea & Tarau, 2004). Se on kieliriippumaton (Mihalcea & Tarau, 2005), kunhan sanat ja virkkeet saadaan eroteltua esimerkiksi hyödyntäen säännöllisiä lausekkeita.

3 Aineiston osien järjestäminen tärkeyden mukaan

Kun lähdeaineisto on pilkottu osiin, joilla koetaan olevan merkitystä tiivistämisprosessin kannalta, osaset täytyy järjestellä niiden tärkeyden ja informaatiopitoisuuden mukaan. Esimerkiksi lähdeaineistosta eritellyt virkkeet on voitava järjestää kuvaavuutensa mukaan, jotta niistä voidaan valita parhaat lopulliseen tiivistelmään. Aihe-esitystapaa käyttäen eriteltyjen rakenneosasten merkittävyys päätellään siitä, miten hyvin ne kuvaavat lähdeaineistossa puhuttua asiaa tai asioita. Indikaattoriesitystapa luottaa valittujen indikaattorien osoittamaan järjestykseen, jossa parhaat ominaisuudet omaava rakenneyksikkö sijoitetaan kärkeen. Koneoppimista hyödynnetään usein indikaattorien merkittävyyden päättelemiseen (Hovy & Lin, 1999; Wong, Wu & Li, 2008).

Luhn loi tärkeysjärjestyksen aihetunnusmerkkien avulla: mitä enemmän tunnusmerkkejä rakenneosasessa (tässä tapauksessa virkkeessä) esiintyy, sitä tärkeämpi sen täytyy olla. Lähestymistapa vaatii kuitenkin jonkinasteista normalisointia, sillä helposti huomataan sen suosivan pitkiä virkkeitä huonoin perustein. Frekvenssimenetelmät eivät kärsi samasta ongelmasta. Rakenneosaset voidaan järjestää niiden frekvenssipainotusten kautta (Salton & Buckley, 1988 - 1982; Spärck Jones, 1972).

Koneoppimisen hyödyntäminen sekä aihe- että indikaattorimenetelmissä on hyvin suosittua ja tehokasta. Lähdeaineisto voidaan annotoida ihmisen toimesta niin, että tiivistelmän tekemisen tai lähdeaineiston aiheen kannalta oleellisimmat rakenneosaset tai indikaattorit kirjataan koneen luettaviksi. Annotoidun koulutusdatan pohjalta koulutetaan luokittelija, joka jaottelee osaset joko hyviin ja huonoihin tavoitteiden mukaisesti. Koska indikaattorilla ei ole jäykkää määritelmää, koneoppimisalgoritmit voivat hyödyntää melkein mitä vain oppiakseen parhaan tavan tuottaa tiivistelmiä. Koneoppimisen toistuvana haasteena on kuitenkin sen vaatima ihmislähtöinen esityö, joka voi viedä huomattavan paljon aikaa, ja koska hyvälle tiivistelmälle ei ole tarkkaa määritelmää, koulutusmateriaalia annotoivat ihmiset voivat olla eri mieltä siitä, mikä on tärkeää ja mikä ei. Jopa virketasolla on vaikea päättää, useiden virkkeiden joukosta, mikä virke sopisi parhaiten tiivistelmään ja mikä taas hieman vähemmän paremmin (Rath, Resnick & Savage, 1961).

Eräs keino välttyä esityöltä on käyttää kahta luokittelijaa, jotka päätyvät kouluttamaan toisiaan. Ensin molemmat luokittelijat opetetaan (joko erillisillä tai samoilla luokitteluperusteilla) pienellä kokoelmalla koulutusmateriaaleja (joko erillisillä tai samoilla), minkä jälkeen ensimmäinen niistä päästetään arvioimaan annotoimatonta aineistoa. Parhaimmat annotoimattomasta aineistosta tehdyt havainnot syötetään toiselle luokittelijalle opetusmateriaaleiksi. Tätä prosessia toistetaan (Wong, Wu & Li, 2008; Xie, Lin & Liu, 2010). Koneoppimisalgoritmeilla vaikuttaa olevan tulevaisuus tekstin automaattisen tiivistämisen kentällä, koska valvomattomien syväoppimisalgoritmien on huomattu pärjäävän

verraten hyvin tai jopa paremmin kuin ihmisvoimin tuotetun koulutusmateriaalin kautta koulutettujen luokittelijoiden. Syväoppimisalgoritmit loistavat etenkin vain tietynlaista tekstiaineistoa tiivistäessä. Esimerkkeinä tiedeartikkelien (Teufel & Moens, 2002) ja biografisen tiedon (Zhou, Ticrea & Hovy, 2004; Biadsy, Hirschberg & Filatova, 2008) tiivistäminen.

4 Tiivistelmään sopivien osasten valinta

Koska raportissa käsitellään vain toista metodien dikotomiajaon jäsentä, erottelevia menetelmiä, koostuu lopullinen tiivistelmä pääasiassa lähdeaineistossa esiintyvistä rakenneosasista, joita ei yritetä muuntaa ainakaan lingvistisessä mielessä. Lähdeaineiston voidaan ajatella olevan palapeli, jossa on runsaasti ylimääräisiä paloja. Tiivistelmä on lähdeaineiston palasista koostuva kokoelma, jossa ei ole turhia paloja. Tämän takia on olennaista tietää, mitkä palaset valitaan ja mitkä "jätetään laatikkoon". Kokeiluissa käytettiin pääasiassa seuraavaa kahta menetelmää.

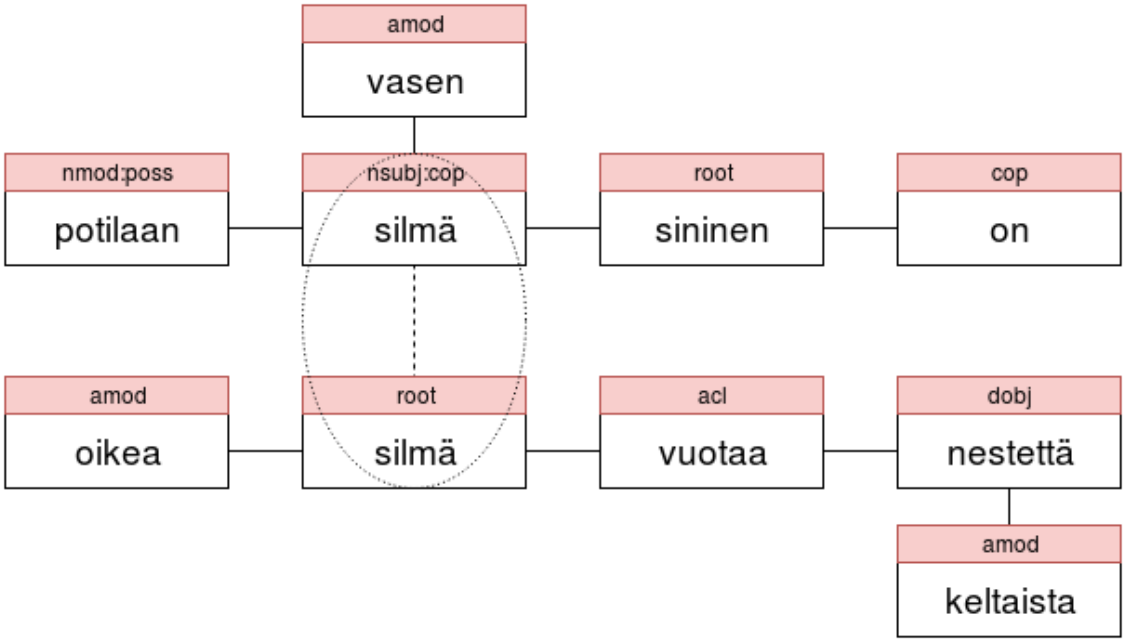
"Parhaat n" -lähestymistapa valitsee rakenneosasten listasta n parhaiten pisteytettyä osaa ja yhdistää ne. MMR (*Maximal Marginal Relevance*) on iteratiivinen lähestymistapa, joka huomioi rakenneosasten merkittävyyden lisäksi redundanssin ja yrittää olla poimimatta osasia, jotka muistuttavat liikaa jo poimittuja, jotta samaa asiaa ei toistettaisi (Carbonell & Goldstein, 1998). Yleisesti rakenneosasia valitessa pyritään minimoimaan redundanssia, maksimoimaan merkittävyys ja koherenssi. Koneoppimista hyödyntäen tämä voidaan saada aikaan tuottamalla erilaisia vaihtoehtoja poimituista rakenneosasista ja arvioimalla niistä paras (McDonald, 2007). Osasia valitessa on kuitenkin muistettava, mihin lyhennelmällä pyritään. Jos tarkoituksena on luoda vain lista lähdeaineistossa esiintyvistä keskeisimmistä aiheista, redundanssin minimoimisen merkitys nousee. Jos kyseessä on proosamaisempi lyhennelmä, sanojen ja aiheiden toistoa ei tarvitse välttää yhtä paljon. Osasten valitseminenkin siis riippuu pyrkimyksestä, mutta osasten valitseminen on aina tiivistysprosessin kahden aikaisemman vaiheen armoilla.

5 Suomenkielisen aineiston esikäsittelystä ja tiivistämisestä

Watson Health Cloud Finland -hankkeessa käsitellyt aineistot ovat kaikki olleet suomenkielisiä. Aineistoja ei ole tiivistetty reaalimaailman tarkoituksiin vaan ainoastaan kokeellisesti. TurkuNLP Group'n työkaluja FDP ja TNP (TurkuNLP Group) hyödynnettiin aineistojen muuntamiseen CoNLL-U-standardin (Universal Dependencies) mukaiseen muotoon, joka erittelee aineistossa esiintyvät osaset sanojen ja välimerkkien tarkkuudella. CoNLL-U-standardi mahdollistaa muun muassa sanojen lemmojen eli sanakirjamuotojen selvittämisen, sanojen taivutusmuotojen erittelemisen ja sanojen lauseensisäiset syntaktiset roolit (esimerkiksi subjekti-predikaatti). TurkuNLP Group’n työkalut pohjaavat toimintansa neuroverkkoon, jota on opetettu rajallisilla koulutusmateriaaleilla. Ne eivät ole täydellisiä, sillä etenkin erisnimien, puhekielisyyden ja slangi-ilmaisujen kanssa tulkinnat menevät toisinaan pieleen. Työkalujen tuottamat tulkinnat ovat kuitenkin enimmäkseen kelvollisia, mutta virheen mahdollisuus on huomioitava järjestelmiä suunnitellessa ja tuloksia tulkitessa.

Maakunta- ja sote-uudistuksen eräs haaste on laajojen, rönsyilevien ja rakenteettomien maakuntatason potilasarkistojen siirtäminen valtakunnalliseen arkistoon, jonka sisällön tulisi olla eheä, tiivis kokonaisuus -- ehkä jopa rakenteeltaan standardisoitu. Kookkaiden aineistojen tärkeimmän sisällön paikantaminen, arviointi ja sisällyttäminen tiivistelmään toimii sovellettuna em. tarkoitukseen. Raportissa esitellyt tekniikat ovat siis käyttökelpoisia ratkaisuja haasteeseen.

Ihminen saattaa käydä elämänsä aikana usean lääkärin puheilla useassa eri laitoksessa. Potilaskertomusten ja käyntilokien muotoa ei ole standardisoitu, joten yhteen ihmiseen liittyvät tekstiaineistot saattavat yhdistettynä olla kaoottisia, itseään toistavia ja sisältää epäoleellista tietoa. Tekstin automaattisen tiivistämisen metodit yhdessä suomenkielisen tekstianalytiikan ja kielentuntemuksen auttavat hahmottamaan, mikä on oleellista ja mikä ei, ja lopulta muodostamaan esimerkiksi seuraavanlaisia datapisteitä potilasta koskevaan tietokantaan (KUVIO 2):



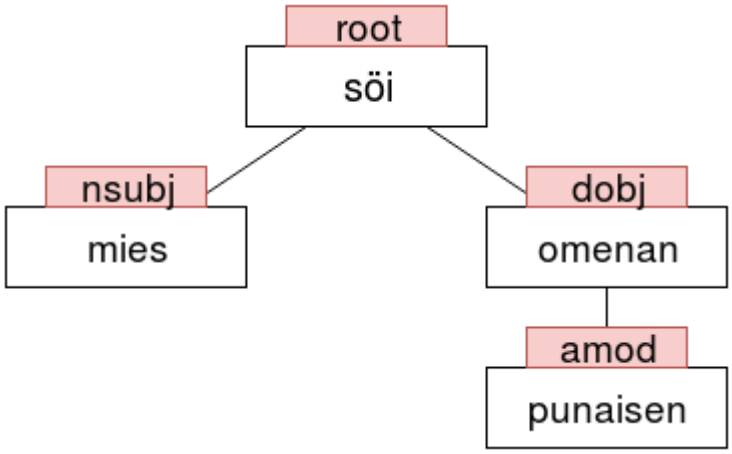
KUVIO 2. Datapiste, jossa sanojen “silmä” dependenssipuut yhdistetään

Kuviossa on kahdesta eri virkkeestä koostettu dependenssipuu, jossa sanat liittyvät toisiinsa niiden syntaktisten relaatioiden kautta. Kuviossa esiintyviä relaatioita ovat muun muassa amod (adjektiiviattribuutti), nsubj:cop (kopulalauseen nominisubjekti) ja dobj (suora objekti). Potilaan rekisteriin on kirjattu kaksi mainintaa, jotka sisältävät sanan "silmä". Ne on kuviossa merkitty erillisellä soikiolla. Sanat voidaan yhdistää toisiinsa, jolloin saadaan laajempi puu, jota pitkin voidaan kulkea etsien haluttua informaatiota sen sisältämien relaatioiden avulla. Kuviossa nähdään selvästi, että "potilaan" vasen silmä on sininen ja oikea silmä vuotaa nestettä, joka on väriltään keltaista. Datapisteen avulla voidaan erinomaisesti eritellä "vasen" ja "oikea" silmä sekä silmistä mainitut seikat. Esimerkkitapauksessa lääkäri löytää nopealla haulla kyseisen potilaan silmään kohdistuvat merkinnät ja pystyy tekemään päätelmänsä helpommin kuin jos lääkärin tarvitsisi lukea vapaamuotoista tekstiä, jossa sana ”silmä” saattaa esiintyvä vaikkapa taivutettuna, tehdäkseen arvionsa. Hyödyntäen apuna lääketieteellistä hakemistoa voidaan tehdä informatiivisempiakin päätöksiä, kuten arvion kahden täysin eri puolilla kehoa olevan maininnan välisestä yhteydestä.

Konstituenttianalyysi tiivistämisen apuna

Konstituentti tarkoittaa kielen rakenneosaa, lausetta, joka koostuu yhdestä tai useammasta yksiköstä, yleensä sanasta. Konstituenttianalyysi tutkii virkkeiden, lauseiden, lausekkeiden ja sanojen välisiä yhteyksiä. Menetelmä on tyypillinen etenkin syntaksille, kun halutaan kuvata lauseiden välisiä tai sisäisiä syntaktisia viittaussuhteita, kuten subjektin suhdetta predikaattiin. Analyysin kautta konstituenteille pyritään löytämään oma tehtävänsä (Chomsky, 1957).

Konstituenttianalyysin hahmottamiseksi käytetään yleisesti apuna puukuvainta (dependenssipuu), jonka avulla pyritään jäsentämään konstituenttien välisiä suhteita. Hierarkia on tämän raportin kannalta kiinnostavin osuus konstituenttianalyysiä. Lauseen *sisällä* hierarkkiset erot tulevat parhaiten esiin kenties attribuuttien viittaussuhteita tulkitessa. Lauseessa "Mies söi punaisen omenan" adjektiivilla "punainen" viitataan sanaan "omena". Tämä ei ole lauseen kannalta olennaista, ellei sana ole erityisen painollinen. Kieliopillisesti yhtä pätevä lause olisi "Mies söi omenan" tai yksinkertaisimmillaan "Mies söi". Tällaista ilmauksien karsintaa kutsutaan *substituutioksi* eli korvaustestiksi. Karsinnan avulla erotettavissa olevasta lauseen ytimestä, jonka subjekti ja predikaatti usein muodostavat, käytetään termiä *neksus*.



KUVIO 3. Konstituenttianalyysiä havainnollistava dependenssipuu

Lauseiden välinen hierarkia on selkeimmin kuvattavissa pää- ja sivulauseiden kohdalla. Sivulauseet määrittävät aina päälausetta ja näin niiden välillä on selkeä arvojärjestys jo merkityksen kannalta, koska päälauseen on mahdollista esiintyä yksin.

Konstituenttianalyysi mahdollistaa tekstin typistämisen esimerkiksi riisumalla attribuutteja em. tavalla tai vielä rajummin, korvaamalla lauseet niiden neksuksilla. Hierarkioiden ymmärtäminen auttaa graafien muodostamisessa tekstistä ja esimerkiksi eri virkkeissä esiintyvän saman lemman ominaisuuksien kasvattamisessa datapistemäisesti ("jalka": "oikea jalka kipeä", "vasen jalka tunnoton").

Lähteet

Barzilay, R. & Elhadad, M. 1999. Text summarization with lexical chains. Teoksessa I. Mani & M. Maybury (toim.) *Advances in Automatic Text Summarization* (111 - 121). MIT Press.

Biadsy, F., Hirschberg, J. & Filatova, E. 2008. An unsupervised approach to biography production using Wikipedia. Teoksessa A. Nenkova, M. Walker & E. Agichtein (toim.), *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (807 - 815), Association for Computational Linguistics.

Carbonell, J. & Goldstein, J. 1998. The use of MMR, diversity-based rerunning for reordering documents and producing summaries. Teoksessa W. Bruce Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson & J. Zobel (toim.), *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 335 - 336 New York: ACM.

Chomsky, N. 1957. *Syntactic Structures*. Saksa: Mouton & Co.

Dang, H.T. 2008. *Proceedings of the Text Analysis Conference*. NIST.

Dang, H.T. & Owczarzak, K. 2008. Overview of the TAC 2008 opinion question answering and summarization tasks. Teoksessa *Proceedings of the TAC 2008 Workshop*. NIST.

Daumé III, H. & Marcu, D. 2006. Bayesian query-focused summarization. *Proceedings of the International Conference on Computational Linguistics and the annual meeting of the Association for Computational Linguistics*, 305- 312.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science,* 391 - 407.

Erkan, G. & Radev, D. 2004. Lexrank: Graph-based centrality as salience in text summarization. Teoksessa *Journal of Artificial Intelligence Research*.

Hovy, E. & Lin, C.Y. 1999. Automated text summarization in summarist. Teoksessa *Advances in Automatic Text Summarization* (82 - 94).

Joty, S. & Chali, Y. 2008. Improving the performance of the random walk model for answering complex questions. Teoksessa *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (9 - 12).

Luhn, H.P. 1957. A Statistical approach to mechanized encoding and searching of literary information. Teoksessa *IBM Journal of Research and Development (1:3),* (309 – 317).

Mani, I., Klein, G., House, D., Hirschman, L., Firmin, T. & Sundheim Beth. 2002. SUMMAC: a text summarization evaluation. Teoksessa *Natural Language Engineering (8:1),* (43 – 68). Cambridge University Press.

Mani, I. 2001. *Automatic Summarization* (129 - 167). The MITRE Corporation.

McDonald, R. 2007. A study of global inference algorithms in multidocument summarization. Teoksessa A. Giambattista, C. Claudio & R. Giovanni (toim.), *29th European Conference on IR Research*, ECIR 2007, (557-564), Berlin: Springer-Verlag.

Mihalcea, R. & Tarau, P. 2004. Textrank: Bringing order into texts. Teoksessa *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* (404 - 411).

Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*.

Over, P., Dang, H.T. & Harman, D. 2007. DUC in Context. Teoksessa *Inf. Process. Manage.* (43:6), (1506 - 1520). Pergamon Press, Inc.

Radev, D., Jing, H., Sty, M., & Tam, D. 2004. Centroid-based summarization of multiple documents. Teoksessa *Information Processing and Management* (40), (919 - 938).

Rath, G., Resnick, A. & Savage, R. 1961. The formation of abstracts by the selection of sentences: Part 1: sentence selection by man and machines. *American Documentation, 2(12),* (139 – 208).

Salton, G. & Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. Teoksessa *Information Processing and Management (24),* (512 – 523).

Silber, H. & McCoy, K. 2002. Efficiently computed lexical chains as intermediate representation for automatic text summarization. *Computational Linguistics, 24(4),* 487- 496.

Spärck Jones. K. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Teoksessa *Journal of Documentation (28*), (11 – 21).

Spärck Jones, K. 1993. What might be in a summary? *Information Retrieval* 93: *Von der Modellierung zur Anwendung* (9 - 26).

Spärck Jones, K. 1999. Automatic summarizing: factors and directions. *Advances in Automatic Text Summarisation*.

Spärck Jones, K. 2001. Automatic language and information processing: rethinking evaluation. *Natural Language Engineering (7:1),* 29 - 46.

Spärck Jones, K. 2007. Automatic summarising: the state of the art. *Information Processing and Management*.

Tarau, P. & Mihalcea, R. 2005. An algorithm for language independent single and multiple document summarization. Teoksessa *Proceedings of the International Joint Conference on Natural Language Processing*, (19 – 24).

Teufel, S. & Moens, M. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. Teoksessa *Computational Linguistics, 28(4),* (409 - 445).

TurkuNLP. TurkuNLP. Saatavilla:14.12.2018 https://turkunlp.github.io/

Universal Dependencies. Universal Dependencies. Saatavilla: 14.12.2018 osoitteesta http://universaldependencies.org/

Wong, K., Wu, M. & Li, W. 2008. Extractive summarization using supervised and semi-supervised learning. Teoksessa *Proceedings of the 22nd International Conference on Computational Linguistics* (985 - 992).

Xie, S., Lin, H. & Liu, Y. 2010. Semi-supervised extractive speech summarization via co-training algorithm. *INTERSPEECH, the 11th Annual Conference of the International Speech Communication Association,* 2522 - 2525.

Zhou, L., Ticrea, M. & Hovy, E. 2004. Multi-document biography summarization. Teoksessa *Conference on Empirical Methods in Natural Language* Processing, *(434 - 441).*

Informaatioteknologian tiedekunnan julkaisuja
No. 69/2018

ISBN 978-951-39-7654-5 (verkkoj.)
ISSN 2323-5004

JYVÄSKYLÄN YLIOPISTO