Tampereen yliopisto

Tuomo Salonen

SENTIMENTTIANALYYSI R-YMPÄRISTÖSSÄ

Informaatioteknologian ja viestinnän tiedekunta
Tilastollisen data-analyysin kandidaattitutkielma
Huhtikuu 2021

Tiivistelmä

Tuomo Salonen: Sentimenttianalyysi R-ympäristössä
Kandidaattitutkielma
Tampereen yliopisto
Matematiikan ja tilastollisen data-analyysin tutkinto-ohjelma
Huhtikuu 2021

Internet on täynnä tekstidataa, kuten sosiaalisen median viestejä, verkkokauppojen tuotearvosteluja ja eri tahojen tuottamia artikkeleja. Tämä tekstidata sisältää esimerkiksi hyödyllistä palautetta yritysten tuotteista ja palveluista. Ongelmana on se, että tekstidataa on internetissä massiivisia määriä ja sitä ilmestyy koko ajan lisää, mikä tekee kaiken tämän datan tutkimisen ihmisvoimin mahdottomaksi. Täten on kehitetty erilaisia koneellisia menetelmiä tekstidatan analysointiin. Yksi näistä on sentimenttianalyysi, jolla yritetään arvioida tekstissä esiintyviä tunteita, ja jonka toteuttamiseen R-ohjelmointiympäristössä tässä tutkielmassa keskitytään.

Jotta tekstidataa voidaan analysoida, pitää se muokata helpommin käsiteltävään muotoon. Yksi vaihtoehto tähän on tidy text -formaatti, jossa tekstidata pilkotaan tiettyihin merkittäviin osiin, kuten esimerkiksi sanoihin, virkkeisiin tai kappaleisiin.

Kun tekstidata on esikäsitelty, voidaan siinä esiintyviä tunteita arvioida sentimenttianalyysin avulla. Tidy text -formaatin kanssa tunnetusti hyvin toimiva ratkaisu on ajatella tekstiä yhdistelmänä yksittäisiä sanoja. Tällöin yksittäisten sanojen sentimentit arvioidaan erikseen, ja näiden sentimenttien summa edustaisi koko tekstin sentimenttiä. Yksittäiset sanat arvioidaan käyttämällä eri tarkoituksiin suunniteltuja sanastoja. Yksi varhaisimmista sanastoista on Bing-sanasto, joka arvioi sanat binäärisesti joko positiivisiksi tai negatiivisiksi.

Tutkielman lopussa tarkastellaan, voidaanko sentimenttianalyysin tuloksia käyttää tekstin kirjoittajan tunnistamiseen. Tähän käytettiin Gutenberg-projektista löytyviä tekijänoikeudettomia kirjoja. Aineistoon otettiin satunnaisesti 30 kirjaa, joista 6 olivat Fjodor Dostojevskin kirjoittamia ja 24 olivat muiden kirjoittajien kirjoja, jotka oltiin Gutenberg-projektissa luokiteltu fiktioksi. Sentimenttianalyysien tuloksista rakennettiin logistinen malli, jonka avulla arvioitiin, onko tekstipätkä Dostojevskin vai muun kirjailijan kirjoittama. Luotu malli osasi noin 75 %:n tarkkuudella arvioida tekstin alkuperän, ja täten myös osoitettiin, että sentimenttianalyysin tuloksia voidaan käyttää jatkoanalyyseissäkin.

Avainsanat: tekstidata, tekstinlouhinta, tekstianalytiikka.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

Sisällys

1 Johdanto **4**

2 Tekstimuotoinen aineisto **5**

3 Tekstin sentimenttianalyysi **6**
3.1 Sanojen sentimentit 6
3.1.1 Bing-sanasto 7
3.1.2 AFINN-sanasto 8
3.1.3 NRC-sanasto 8
3.2 Suomenkielisen tekstin sentimenttianalyysi 8

4 Esimerkki sentimenttianalyysin käytöstä **10**
4.1 Aineiston kuvailu 10
4.2 Käytetyt menetelmät 10
4.2.1 Sentimenttianalyysi 10
4.2.2 Logistinen regressio 11
4.3 Tutkimuksen tulokset 11
4.4 Johtopäätökset 13

Lähteet **14**

Liite: R-koodi **16**

1 Johdanto

Internet on nykyään osa ihmisen arkielämää, ja esimerkiksi sosiaalista mediaa käyttää jopa 79 % länsi- ja pohjoiseurooppalaisista (Datareportal, We Are Social ja Hootsuit, 2021). Myös verkkokaupat ja niiden käyttäjäarvostelut ovat yleistyneet ja internet on täynnä sivuja, jotka julkaisevat artikkeleita ja uutisia. Kaikki nämä tuottavat internetiin suuria määriä tekstidataa, jota hyödyntäen yritykset ja muut organisaatiot pystyvät saamaan hyödyllistä ilmaista palautetta heidän tuotteistaan ja palveluistaan. Datan paljous on nostanut kysyntää tekstinlouhinnalle, eli tekstin analysoimiselle tietokoneen avustuksella. Yksi tekstinlouhinnan tyyppi on sentimenttianalyysi, jolla tekstissä esiintyviä tunteita yritetään koneellisesti arvioida.

Tässä kandidaattitutkielmassa perehdytään tarkemmin sentimenttianalyysiin ja siihen, miten sen voi toteuttaa R-ohjelmointiympäristössä (R Core Team, 2021). R on erityisesti tilastolliseen laskentaan ja grafiikoihin suuntautuva ohjelmointikieli, ja täten se soveltuu hyvin muun muassa tiedonlouhintaan ja data-analyysiin. R myös tukee käyttäjien tekemiä paketteja, jotka lisäävät ominaisuuksia ja helpottavat tiettyjen asioiden tekemistä. Yksi asioista, joita R:n paketeilla saa helposti tehtyä, on juuri sentimenttianalyysi.

Tutkielman toisessa luvussa katsotaan, miten tekstin saa sopivaan muotoon jatkoanalyysejä varten. Sen jälkeen perehdytään sentimenttianalyysiin ja erilaisiin tapoihin toteuttaa se. Viimeisessä osassa käytetään edellisissä osissa opittuja asioita ja tutkitaan sentimenttianalyysin avulla, voidaanko tietyn tekstin kirjoittaja ennustaa tekstissä esiintyvien sentimenttien perusteella.

2 Tekstimuotoinen aineisto

Tekstidatalla tarkoitetaan yksinkertaisesti kaikkea dataa, joka on tekstimuodossa. Tekstidata voi siis olla esimerkiksi viestejä sosiaalisessa mediassa, tuotearvosteluita tai kokonainen romaani. Koska tekstidataa voidaan saada monesta lähteestä ja monessa muodossa, on se ja siitä tekstinlouhinnalla saadut tiedot myös relevantteja monella tieteenalalla.

Kuten muukin data, tekstidata on saatava sopivaan muotoon, että analyysejä voidaan suorittaa. Tähän on monenlaisia erilaisia tapoja, mutta tässä tutkielmassa keskitytään tidy text -formaattiin (Silge ja Robinson, 2017). Tidy text -formaatissa teksti jaetaan taulukkoon siten, että jokaisella rivillä on yksi merkittävä tekstin osa. Nämä osat voivat olla esimerkiksi sanoja, lauseita tai rivejä. Kuvassa 2.1 on esimerkki virkkeen muuntamisesta tidy text -formaattiin, jossa merkittävänä osana on sana.

”Lause, joka muutetaan, ja on tässä esimerkkinä.” ⟶

1	lause
2	joka
3	muutetaan
4	ja
5	on
6	tässä
7	esimerkkinä

käsittelemätön teksti *tidy text*

Kuva 2.1. Esimerkki tekstin muuntamisesta tidy text -formaattiin

R:llä tidy text -formaattia voidaan käyttää osuvasti nimetyn tidytext-paketin avulla (Silge ja Robinson, 2016). Kyseisen paketin sisältämä unnest_tokens-funktio käsittelee sille annetun tekstin ja palauttaa siitä tidy text -formaatin mukaisen taulukon, jossa joka rivillä on yksi tekstin merkitsevä osa. Funktio käyttää oletuksena sanaa merkitsevänä osana, mutta parametrinä sille voi määrittää myös muita sellaisia, kuten virkkeen tai kappaleen.

Tidy text -formaattiin saatua tekstidataa voidaan käsitellä kuten muitakin taulukkomuotoisia datoja. Täten myös monet yleisesti käytetyt R-paketit, kuten datatieteen tarpeisiin suunnitellun tidyverse-kokoelman (Wickham et al., 2019) sisältämät dplyr ja ggplot2, toimivat tidy text -formaatin kanssa. Tämän seurauksena voidaan esimerkiksi etsiä sanoja, jotka ovat tietyn pituisia, laskea sanojen esiintymiä ja piirtää niistä sanapilviä, tai vertailla kirjan eri lukujen sentimenttejä toisiinsa.

3 Tekstin sentimenttianalyysi

Sentimenttianalyysi on tekstinlouhinnan tyyppi, jolla yritetään koneellisin menetelmin arvioida, millaisia sentimenttejä tekstissä esiintyy. Menetelmänä se ei ole laskennallisesti monimutkainen, vaan laskutoimitukset ovat enimmäkseen summaamista, ja kielen osaavat ihmiset myös pystyvät helposti ymmärtämään tekstin sentimentin pelkästään lukemalla sen. Nämä asiat voivat herättää kysymyksiä sentimenttianalyysin tarpeellisuudesta, sillä millään menetelmällä ei pystytä arvioimaan sentimenttejä ihmisen tarkkuudella. Toisaalta aikuinen ihminen lukee englanninkielistä tekstiä keskimäärin 238 sanan minuuttinopeudella, jos teksti on tietokirjallisuutta, ja 260 sanan minuuttinopeudella, jos teksti on fiktiota (Brysbaert, 2019). Parhaassa tapauksessa keskivertolukijalta täten menee noin 64 tuntia miljoonan sanan lukemiseen, kun taas koneellisesti miljoonan sanan sentimenttianalyysi voidaan toteuttaa alle sekunnissa. Täten menetetty tarkkuus ei välttämättä ole ongelma sentimenttianalyysissä, sillä analysoitavaa tekstidataa tuotetaan kuitenkin internettiin käsittämättömiä määriä joka sekunti, ja tämän kaiken lukeminen ihmisvoimin olisi mahdotonta.

Yksi tapa analysoida tekstin sentimenttiä on ajatella tekstiä yhdistelmänä yksittäisiä sanoja. Tällöin arvioidaan näiden yksittäisten sanojen sentimentit erikseen, ja näiden erillisten sanojen sentimenttien summa edustaisi koko tekstin sentimenttiä. Tällä menetelmällä on omat heikkoutensa ja se ei ota huomioon ympäröiviä sanoja sentimenttiä havaitessaan, joten esimerkiksi negaatiot, kuten ”not good”, voivat vaikuttaa arvioitujen sentimenttien tarkkuuteen. Tekstit eivät kokonaisuuksina kuitenkaan suuria määriä negaatioita sisällä, joten tämän vaikutus suuremmassa mittakaavassa ei ole merkittävä, ja menetelmä on paljon käytetty. Yksittäisten sanojen sentimentteihin perustuvat sanastopohjaiset menetelmät toimivat myös hyvin luvussa 2 mainitun tidy text -formaatin kanssa, joka sopivasti sisältää yhden sanan jokaisella rivillä, ja täten tutkielmassakin keskitytään niihin.

3.1 Sanojen sentimentit

Yksittäisten sanojen sentimentit pystytään säilyttämään hakemistossa, jonka englanninkielinen termi, dictionary, myös osuvasti kääntyy sanakirjaksi. Hakemisto on tietorakenne, joka koostuu avain-arvo-pareista. Antamalla hakemistoon avaimen, saadaan sieltä tähän avaimeen yhdistetty arvo. Nämä avaimet ja arvot voivat olla mitä tahansa tietotyyppiä, ja täten esimerkiksi voidaan käyttää sanoja avaimina ja näihin sanoihin yhdistettyinä arvoina voi olla listoja sentimenteistä. Kuvassa 3.1 havainnollistetaan tämän rakenteen toimintaa pienellä merkkijonoja sisältävällä hakemistolla.

Hakemistoihin pohjautuvat rakenteet ovat sen ominaisuuksien seurauksena myös sentimenttianalyysin yhteydessä yleisiä. Sentimenttianalyysiin tarkoitetuista hakemistoista käytetään termiä sentiment lexicon, ja ne koostuvat juuri sana-sentimentit-pareista. Sentiment lexiconeja on monenlaisiin eri tarkoituksiin, esimerkiksi yksi voi erikoistua sosiaaliseen mediaan kun taas toinen erikoistuu finanssialan asiakirjoihin, ja täten sama sana voi eri sentiment lexiconeissa yhdistyä täysin erilaisiin ja jopa vastakkaisiin sentimentteihin.

Avain		*Arvo*
katumus	⟶	negatiivinen, suru
sekunda	⟶	negatiivinen
tyytyväinen	⟶	positiivinen, ilo

Kuva 3.1. Esimerkki hakemistosta, jossa avaimina on sanoja ja arvoina listoja sentimenteistä

Koska kielissä on satojatuhansia sanoja, on sentiment lexiconien luominen haastavaa. Tämän seurauksena niiden luomisessa yleensä käytetään apuna joukkoistamista tai koneoppimista. Näiden toimenpiteiden seurauksena tuotetut sanastot kuitenkin pitää validoida. Validointiprosessissa asiantuntevat henkilöt arvioivat käsin sanastoissa olevien sanojen sentimentit, ja näihin ihmisten arvioimiin sentimentteihin vertaamalla tarkistetaan, ovatko sanaston luokittelut oikein. Eri arvioijat eivät välttämättä arvioi sanojen sentimenttejä samoin, joten prosessissa tarvitsee myös asettaa sopivia rajoja tämän suhteen ja hylätä myös sanoja, joista arvioitsijat eivät olleet tarpeeksi yksimielisiä (Mozetič, Grčar ja Smailović, 2016). Täten objektiivisesti täydellistä sentiment lexiconia ei siis ole mahdollista tehdä.

Seuraavaksi perehdytään tarkemmin yleiskäyttöön tarkoitettuihin Bing, AFINN ja NRC sentiment lexiconeihin, jotka ovat myös saatavilla tidytext-paketin tarjoaman get_sentiments-funktion kautta.

3.1.1 Bing-sanasto

Bing sentiment lexicon (Hu ja Bing, 2004) on yksi varhaisimmista sanastopohjaisista sentimenttimenetelmistä, ja sen työstäminen aloitettiin jo vuonna 2004. Bing sentiment lexicon on menetelmänä tässä tutkielmassa käsitellyistä yksinkertaisin. Se luokittelee sanat binäärisesti joko positiiviseksi tai negatiiviseksi, ja koko tekstikokonaisuuden sentimentti on yksinkertaisesti vain näiden erotus. Tämä suhteellinen yksinkertaisuus tosin johtaa myös siihen, että uusien sanojen lisääminen on helpompaa, ja validoinnissa esiintyy vähemmän erimielisyyksiä. Täten Bing sentiment lexicon on myös tutkielmassa käsitellyistä kattavin, sisältäen 6 786 sanaa sentimentteineen.

Bing sentiment lexiconin luomiseen käytettiin WordNet-palvelua (Princeton University, 2010). WordNet sisältää englanninkielisiä sanoja, ja niiden semanttisia suhteita, kuten sanan synonyymejä ja antonyymejä. Aluksi Hu ja Bing loivat niin sanotun siemenlistan yleisistä ja selkeästi joko positiivisista tai negatiivisista sanoista, kuten ”great” tai ”bad”. Tämän jälkeen he keräsivät listan sekalaisista adjektiiveista ja loivat algoritmin, joka automaattisesti katsoo WordNet-palvelusta, onko uudesta listalla tulleella adjektiivilla synonyymejä tai antonyymejä siemenlistassa. Jos sanalla on synonyymi siemenlistalla, lisätään uusi sana myös siemenlistaan samalla sentimentillä kun sen synonyymi, ja jos siemenlistalla on antonyymi, lisätään se listaan sen vastakkaisella sentimentillä. Sanat, joille ensimmäisellä kierroksella ei löytynyt synonyymiä tai antonyymiä, käydään läpi uudestaan adjektiivilistan loputtua, sillä siemenlistalle lisätyt uudet sanat voivat olla sen synonyymejä tai antonyymejä.

3.1.2 AFINN-sanasto

AFINN sentiment lexicon (Nielsen, 2011) muistuttaa Bing sentiment lexiconia (Hu ja Bing, 2004), sillä sekin arvioi sanat joko positiiviseksi tai negatiiviseksi. Se ei kuitenkaan luokittele sanoja pelkästään binäärisesti, kuten Bing, vaan asteikolla -5:stä +5:een, jossa -5 on erittäin negatiivinen ja +5 erittäin positiivinen. Tämä merkitsee, että esimerkiksi virke ”While the characters were fine, the prose was horrible”, jonka Bing sentiment lexicon luokittelee neutraaliksi, on AFINN sentiment lexiconin mukaan negatiivinen, koska sana ”fine” saa arvon +2 ja ”horrible” -3. Täten AFINN sentiment lexicon voi joissain tilanteissa arvioida tekstin sentimenttiä tarkemmin kuin yksinkertaisempi Bing. Toisaalta AFINN sentiment lexicon sisältää vain 2 477 sanaa sentimentteineen, joka on huomattavasti pienempi määrä kuin muissa tutkielmassa käsitellyissä sentiment lexiconeissa.

AFINN sentiment lexicon poikkeaa myös siten, että se on ainoa käsitellyistä sentiment lexiconeista, joka on täysin manuaalisesti tehty. Nielsen kokosi sanaston avoimesti internetissä saatavilla olevista lähteistä, kuten sanakirjoista ja sosiaalisen median viesteistä, ja se sisältää myös internet-slangia. Saatu sanasto validoitiin aineistolla, joka koostui tuhannesta twiitistä, joiden sentimentit oli etukäteen jo arvioitu.

3.1.3 NRC-sanasto

NRC sentiment lexicon (Mohammad ja Turney, 2013), kuten kaksi edellistäkin, sisältää sanoja, joihin on yhdistetty joko positiivinen tai negatiivinen sentimentti. Muista tutkielman sentiment lexiconeista poiketen se luokittelee sanoja myös kymmeneen tunteeseen, sisältäen tunteet anger, anticipation, disgust, fear, joy, sadness, surprise ja trust. NRC oli ensimmäinen sentiment lexicon, joka arvioi myös tunteita. NRC sentiment lexiconin luokittelemat sanat, joita se sisältää 6 468, voivat yhdistyä moneen tunteeseen, esimerkiksi sanaan ”abandoned” yhdistyy negatiivisuuden lisäksi tunteet anger, fear ja sadness. Kuten voisi olettaa, sisällyttämällä tunteet pystytään avaamaan uusia käyttökohteita sentimenttianalyysille, ja NRC sentiment lexiconia on käytetty muun muassa havaitsemaan internet-kiusaamista ja muuntamaan koneellisesti kirjallisuutta musiikiksi siinä eri kohdissa esiintyvien tunteiden mukaan.

NRC sentiment lexicon luotiin joukkoistamisen avulla. Aluksi Mohammad ja Turney keräsivät listan sanoja, joita he halusivat arvioida, ja tämän listan pohjalta he loivat kyselyitä jotka jaettiin Amazonin Mechanical Turk -palvelun avulla eteenpäin. Kyselyissä kysyttiin yhdistääkö vastaaja sanaa valittuihin sentimentteihin, mutta sen lisäksi siihen oli lisätty kysymyksiä, joilla tarkistettiin, voidaanko kyseiseen kysymyserän vastauksiin luottaa, ja yleisemmällä tasolla voidaanko vastaajan muihinkaan vastauksiin luottaa. Jäljelle jääneiden vastausten perusteella koottiin lopullinen sanasto.

3.2 Suomenkielisen tekstin sentimenttianalyysi

Vaikka tutkielma on kirjoitettu suomeksi, ovat kaikki tutkielmassa tähän mennessä mainitut sanastot englanninkielisiä. Tämä johtuu siitä, että suomenkielisiä sentiment

lexiconeja ei ole tähän mennessä avoimeen käyttöön julkaistu. Eräät suomen kielen piirteet todennäköisesti ovat vaikeuttaneet toimivan sanaston luomista.

Suomen kieli on agglutinatiivinen kieli, eli siinä sanan vartaloon liittyy jopa useitakin päätteitä ja muita sanaliitteitä. Esimerkiksi sana ”kauppa” voi taipua tuhansiin muotoihin, kuten ”kauppaan”, ”kaupassa”, ”kauppoihinko” tai ”kaupastanikohan”. Kuten edellä mainituista sanoista voidaan huomata, suomen kielessä sanan vartalo ei välttämättä pysy samana päätteitä ja sanaliitteitä siihen liitettäessä. Jos nyt halutaan kääntää suoraan englanninkielistä sentiment lexiconia suomeksi, kohdataan ongelmia. Esimerkiksi negatiivisiin sentimentteihin yhdistetyt sanat ”bully” ja ”bullied” kattavat suurimman osan kiusaamiseen liittyvistä termeistä englannin kielellä. Suomen kielellä sanastossa tarvitsisi olla ”kiusata”, ”kiusaaja”, ”kiusattu”, ”kiusannut”, ”kiusasi” ja niin edelleen, jotta kaikki mahdolliset eri sanamuodot saataisiin katettua. Nämä taivutusmuodot paisuttaisivat sanaston kokoa, jolloin se sekä vaatisi enemmän muistia että käyttäisi enemmän aikaa sentimenttianalyysin toteuttamiseen, sillä sanastojen läpikäyminen jokaisen sanan kohdalla veisi niiden koon seurauksena enemmän aikaa. Täten puhtaan sanastopohjaisen menetelmän sijaan parempi ratkaisu voisikin olla se, että sanoista ensin etsitään niiden vartalo koneellisesti käyttäen esimerkiksi omorfia (Pirinen, 2015), joka sisältää suomen kielen käsittelyyn tarkoitettuja työkaluja, ja tätä vartaloa sitten haetaan sentiment lexiconista, joka koostuu juuri sanojen vartaloista. Tässäkin on omat ongelmansa, esimerkiksi sana ”rahatta” voitaisiin nähdä negatiivisena, mutta sen vartaloa ”raha” taas ei, joten analyysin tarkkuus voisi kärsiä.

Lähiaikoina suomenkielistä sentimenttianalyysiä on alettu tarkemmin tutkia. Esimerkiksi Nukarinen (2018) osoitti diplomityössään, että koneoppimisen, omorfin (Pirinen, 2015) ja verkkokauppojen arvosteluiden avulla voidaan luoda suomenkieliselle tekstille luokittelija, joka osaa tekstissä esiintyvien sentimenttien perusteella ennustaa minkä arvosanan käyttäjä tuotteelle antaisi. Myös Öhman (2021) on luonut suomenkielisen sanaston kääntämällä tässäkin tutkielmassa käsitellyn NRC sentiment lexiconin (Mohammad ja Turney, 2013) aluksi Google Translaten avulla suomen kielelle, jonka jälkeen hän käsin muokkasi sanoja, joissa automaattinen käännös ei täysin onnistunut. Näiden lisäksi Kielipankin työkaludemoissa (Helsingin yliopisto, Digitaalisten ihmistieteiden osasto, 2019) on suomenkielisen tekstin sentimenttianalyysiä varten luotu finsentiment-demo, jota toivottavasti jalostetaan jatkossa.

4 Esimerkki sentimenttianalyysin käytöstä

Tässä esimerkissä sentimenttianalyysin käytöstä katsotaan, voidaanko jonkin tietyn tekstipätkän kirjoittaja päätellä kyseisessä tekstissä esiintyvien sentimenttien perusteella. Esimerkissä päädyttiin pohtimaan, voiko Fjodor Dostojevskin englanniksi käännettyä tekstiä erottaa muusta englanninkielisestä fiktiosta. Dostojevski valittiin, koska hänen teoksensa ovat tunnettuja niiden filosofisista ja psykologisista teemoistaan, joka todennäköisesti merkitsee, että niissä esiintyvät sentimentit myös eroavat muusta yleisestä fiktiokirjallisuudesta. Tämän esimerkkianalyysin tekemiseen käytetyt R-koodit löytyvät liitteenä tutkielman lopusta.

4.1 Aineiston kuvailu

Aineistona tässä esimerkissä käytetään Gutenberg-projektin tarjoamia public domain, eli tekijänoikeudettomia, kirjoja (Project Gutenberg, n.d.). Gutenberg-projekti käynnistettiin vuonna 1971 ja sen tavoitteena on kääntää tekijänoikeudettomia kirjoja mahdollisimman avoimeen ja helposti jaettavaan sähköiseen muotoon. Tämä prosessi tapahtuu vapaaehtoisvoimin. Tällä hetkellä Gutenberg-projekti tarjoaa yli kuusikymmentätuhatta sähköistä kirjaa.

Aineiston käsittelyssä apuna toimii gutenbergr-paketti (Robinson, 2020). Tämän avulla pystytään esimerkiksi listaamaan kirjoja erilaisin ehdoin. Tässä esimerkissä kyseistä ominaisuutta käyttäen haettiin listaus sekä Dostojevskin teoksista että teoksista, joiden Gutenberg-projektin kirjahyllyluokittelussa on termi ”fiction”. Näistä listauksista otettiin sen jälkeen otos, jossa oli 6 Dostojevskin teosta ja 24 muuta fiktioteosta. gutenbergr-paketin avulla nämä kirjat pystyttiin myös lataamaan suoraan R:n ympäristöön, jolloin niitä ei tarvinnut erikseen ladata ja sen jälkeen lukea levyltä. Tämä latausominaisuus halutessa poistaa kirjoista samalla Gutenberg-projektin lisäämät lisenssitiedot, ja tässä esimerkissä näin tehtiin, koska analyysin oli tarkoitus kohdistua alkuperäisiin teksteihin.

Gutenberg-projektista ladatut raakatekstitiedostot muutettiin luvussa 2 käsiteltyyn tidy text -formaattiin käyttäen tidytext-pakettia (Silge ja Robinson, 2016). Aineisto jakaantui täten 2 409 673:een eri riviin, joista jokainen sisälsi tidy text -formaatin mukaisesti yhden sanan, sekä tämän lisäksi sanan sisältäneen teoksen tunnuksen ja rivinumeron jolla sana kyseisessä teoksessa esiintyi. Samalla aineiston eri osiin liitettiin koodaus 1 tai 0, jossa arvo 1 tarkoitti Dostojevskin tekstiä ja 0 muuta tekstilähdettä.

4.2 Käytetyt menetelmät

4.2.1 Sentimenttianalyysi

Sentimenttianalyysiä varten teksti jaettiin 500 sanan osiin. Tämä vaikutti tarpeeksi isolta määrältä tekstiä, jotta tekstipätkien välillä voitaisiin löytää systemaattisia eroja sentimenteissä. Aineiston sentimenttianalyysi toteutettiin luvussa 3.1.3 mainitulla

NRC sentiment lexiconilla, joka ei luokittele sanoja pelkästään positiivisiksi tai negatiivisiksi, vaan luokittelee sanoja myös tunteisiin anger, anticipation, disgust, fear, joy, sadness, surprise ja trust (Mohammad ja Turney, 2013). Tekstipätkässä jokaisen sentimentin esiintymät laskettiin, ja näistä muodostui tekstipätkälle täten kymmenen eri muuttujaa.

NRC-sanasto ei välttämättä täydellisesti sovi vanhan kirjallisuuden analysointiin, sillä käytetyt sanat ja niiden merkitykset ovat voineet hieman ajan myötä muuttua. Oletettavasti kieli ei ole kuitenkaan niin paljoa muuttunut, ettei menetelmällä mitään hyödyllistä informaatiota saataisi.

4.2.2 Logistinen regressio

Logistinen regressio on tavanomaisen regressiomallin erityistyyppi. Samoin kuin muissa regressiomalleissa, myös logistisessa regressiossa yritetään löytää malli, joka parhaiten selittää selitettävän muuttujan ja yhden tai useamman selittävän muuttujan väliset suhteet. Toisin kuin tavanomaisessa regressiossa, jossa selitettävä muuttuja on jatkuva, logistisessa regressiossa selitettävä muuttuja on binäärinen. Täten selitettävä muuttuja voi siis saada joko arvon 0 tai 1, jossa arvo 1 on se luokka, jota analyysissä halutaan ennustaa.

Merkitään odotusarvoa $E(Y|\boldsymbol{x}) = \pi(\boldsymbol{x})$. Tällöin logistinen regressiomalli on

$$\log\left(\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}\right) = \boldsymbol{\beta}'\boldsymbol{x},$$

jossa $\boldsymbol{\beta}$ on vektori, joka koostuu mallin regressiokertoimista, ja $\boldsymbol{x}$ on vektori, joka koostuu selittävien muuttujien arvoista. Täten logistisessa regressiomallissa vastemuuttujana toimii siis tapahtuman vedon logaritmi. Tästä ratkaisemalla $\pi(\boldsymbol{x})$ saadaan

$$\pi(\boldsymbol{x}) = \frac{e^{\boldsymbol{\beta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}},$$

jolla voidaan arvioida todennäköisyys sille, että vastemuuttuja saa arvokseen 1. (Hosmer, Lemeshow ja Sturdivant, 2013.)

Tässä tutkielmassa logistista regressiota käyttäen luotiin malli, jolla yritettiin arvioida, onko teksti Dostojevskin kirjoittamaa vai ei. Selitettävänä muuttujana malleissa oli siis tekijä, jossa 1 tarkoitti Dostojevskiä ja 0 jotain muuta, ja selittävinä muuttujina oli 500 sanan tekstipätkistä sentimenttianalyysin kautta saadut tunnemäärät.

4.3 Tutkimuksen tulokset

Tekstipätkistä tehtyjen sentimenttianalyysien tuloksista luotiin logistinen regressiomalli, jota työstettiin käyttämällä askeltavaa regressiota. Aluksi logistiseen regressiomalliin siis lisättiin selittäviksi muuttujiksi kaikki mahdolliset sentimentit, jonka jälkeen katsottiin, että ovatko kaikki selittävät muuttujat tilastollisesti merkitseviä. Jos eivät, luotiin uusi malli, jossa vähiten tilastollisesti merkitsevä muuttuja poistettiin, ja tätä jatkettiin, kunnes kaikki selittävät muuttujat olivat tilastollisesti merkitseviä. Malli 1 sisälsi täten kaikki kymmenen sentimenttiä, eli anger, anticipation,

disgust, fear, joy, negative, positive, sadness, surprise ja trust. Tämän jälkeen tulevista malleista jokaisesta poistettiin yksi sentimentti edellisen mallin sisältämistä sentimenteistä. Malli 2:sta poistettiin anger, malli 3:sta joy, malli 4:stä positive, malli 5:stä sadness, malli 6:sta anticipation ja malli 7:stä negative.

Taulukko 4.1. Askeltavassa regressiossa luodut regressiomallit ja niiden informaatiokriteerit

Malli	BIC	AIC
Malli 1	587.24	540.08
Malli 2	581.00	538.13
Malli 3	575.62	537.03
Malli 4	570.16	535.86
Malli 5	564.89	534.88
Malli 6	560.42	534.69
Malli 7	555.74	534.30

Bayes-informaatiokriteeri, englanniksi lyhennettynä BIC, ja Akaiken informaatiokriteeri, englanniksi lyhennettynä AIC, ovat keinoja arvioida mallin hyvyyttä. Nämä molemmat toimivat samalla periaatteella, eli mitä pienempi arvo, sitä parempi malli. Kuten taulukosta 4.1 nähdään, jokaisessa askeltavan regression vaiheessa mallin sopivuus aineistoon kasvoi. Alkutilanteessa, kun kaikki sentimentit olivat mukana, oli mallin BIC 587.24 ja AIC 540.08, kun taas lopullisessa mallissa, jossa mukana olivat vain sentimentit disgust, fear, surprise ja trust, oli BIC 555.74 ja AIC 534.30.

Taulukko 4.2. Logistinen regressioanalyysi sentimenttien yhteydestä siihen, että teksti on Dostojevskin kirjoittamaa

Muuttuja	Regressiokerroin	Vetosuhde	Keskivirhe	p-arvo
Vakiotermi	-1.884	0.152	0.464	< 0.001
disgust	0.101	1.106	0.01	< 0.001
fear	-0.044	0.957	0.006	< 0.001
surprise	0.042	1.043	0.008	< 0.001
trust	-0.014	0.987	0.003	< 0.001

Taulukosta 4.2 nähdään, että lopullisessa mallissa kaikki jäljellä olevat muuttujat ovat tilastollisesti merkitseviä, koska niiden p-arvot ovat erittäin lähellä nollaa. Siitä myös nähdään, että miten jokainen muuttuja vaikuttaa mallin luokitteluun. Sentimenttien disgust ja surprise määrien kasvaessa todennäköisyys sille, että teksti on Dostojevskin kirjoittamaa, myös nousee, kun taas sentimentit fear ja trust vähentävät tätä todennäköisyyttä. Muuttujien vetosuhteista nähdään, että esimerkiksi disgust-sentimentin esiintymien kasvaminen yhdellä tarkoittaa sitä, että tekstin kirjoittajana Dostojevskin olemisen veto kasvaa 1.106-kertaiseksi.

Lopullisen logistisen regressiomallin tarkkuutta tarkasteltiin käyttämällä sitä ennustamaan kaikkien mallin luomiseen käytettyjen sentimenttianalyysien kohdalla,

kuinka todennäköisesti kyseinen sentimenttianalyysi oli tehty Dostojevskin tekstistä. Tämän jälkeen sentimenttianalyysit luokiteltiin mallin antamien todennäköisyyksien perusteella siten, että jos todennäköisyys oli yli 0.5, luokiteltiin kyseinen tekstipätkä Dostojevskin kirjoittamaksi, eli annettiin sille arvo 1, ja muussa tapauksessa se luokiteltiin jonkun muun kirjoittamaksi, eli annettiin arvo 0. Näitä verrattiin oikeisiin aineistoon aiemmin liitettyihin kirjoittajakoodauksiin. Malli onnistui luokittelemaan 74.7 % sentimenttianalyyseistä oikein, joka oli odotettua parempi tulos.

Se, että malli osaa luokitella mallin luomiseen käytettyä aineistoa melko hyvin, ei kuitenkaan kerro toimiiko malli tämän aineiston ulkopuolella. Täten otettiin uusi otos kirjoja Gutenberg-projektista, jossa ei mukana ollut mitään aiemmin mallin luomiseen käytettyjä kirjoja. Uuteen otokseen otettiin 2 Dostojevskin kirjaa ja 6 muuta fiktiokirjaa, ja näille tehtiin myös samalla tavalla sentimenttianalyysit. Kun logistista regressiomallia käytettiin ennustamaan ja luokittelemaan nämä uudet sentimenttianalyysit samalla tavalla kun viime kappaleessa, onnistui regressiomalli luokittelemaan 75.6 % näistä oikein.

4.4 Johtopäätökset

Tämän esimerkkianalyysin perusteella logistisella regressiomallilla voidaan suhteellisen hyvin ennustaa Gutenberg-projektin tarjoamien Dostojevskin teosten ja muiden yleisen fiktiokirjallisuuden alkuperää. Toisaalta, vaikka malli osasikin tässä tapauksessa arvioida noin 75 % tekstipätkistä oikein, se ei tarkoita, että myös muiden kirjailijoiden teoksia voitaisiin samoin menetelmin arvioida. Dostojevski on kuitenkin klassikkokirjailija, joka tunnetaan psykologisista ja filosofisista teemoistaan. Voitaisiinko esimerkiksi jonkun keskivertokirjailijan tekstit erottaa muista aikansa keskivertokirjailijoista pelkästään niissä esiintyvien sentimenttien avulla? Erilaiset tunteet ovat kuitenkin melko tärkeä ja normaali osa perinteistä tarinankerrontaa.

Jatkossa voitaisiin tutkia, kuinka hienojakoisesti sentimenttianalyysien tuloksilla kirjailijoita voidaan luokitella. Tässä tutkielmassa käytettiin tavallista logistista regressiomallia, jossa selitettävä muuttuja pystyy saamaan vain kaksi mahdollista arvoa. Multinomiaalisessa logistisessa regressiossa näitä luokkia voisi olla enemmän, ja sitä käyttäen voitaisiin esimerkiksi tutkia, onko teksti Dostojevskin, Edgar Allan Poen vai H.G. Wellsin kirjoittamaa. Tässä todennäköisesti kohdattaisiin pelkän sentimenttianalyysin tuottaman tiedon raja, sillä kuten edellisessä kappaleessa mainittiin, ovat tunteet kuitenkin suuri osa tarinankerrontaa. Täten voitaisiin tarvita myös käytettyjen sanojen lukumääriä, sillä näissäkin voisi löytyä eroja eri kirjailijoiden välillä. Sanojen lukumäärien laskeminen onnistuisi onneksi erittäin helposti käyttämällä tidy text -formaattia.

Lopuksi todettakoon, että vaikka sentimenttianalyysi onkin melko yksinkertainen menetelmä, antaa se hyödyllisiä tietoja aineistosta, ja sen tuloksia käyttämällä voidaan myös tehdä jatkoanalyysejä. Tulevaisuudessakin tarvitaan nopeasti ja helposti tietoa ihmisten mielipiteistä esimerkiksi tuotekehittämisen ja politiikan alueilla, joten sentimenttianalyysiä tullaan jatkossakin käyttämään.

Lähteet

Brysbaert, M. (2019). ”How many words do we read per minute? A review and meta-analysis of reading rate”. *Journal of Memory and Language* 109. DOI: 10.1016/j.jml.2019.104047.

Datareportal, We Are Social ja Hootsuit (2021). *Global social network penetration rate as of January 2021, by region*. Statista. URL: https://www.statista.com/statistics/269615/social-network-penetration-by-region/ (viitattu 03. 04. 2021).

Helsingin yliopisto, Digitaalisten ihmistieteiden osasto (2019). *Kielipankin työkaludemot*. Kielipankki. URL: https://www.kielipankki.fi/tools/demo/ (viitattu 27. 03. 2021).

Hosmer, D. W., Lemeshow, S. ja Sturdivant, R. X. (2013). *Applied Logistic Regression*. 3. painos. Hoboken: Wiley.

Hu, M. ja Bing, L. (2004). ”Mining and Summarizing Customer Reviews”. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’04, s. 168–177. DOI: 10.1145/1014052.1014073.

Mohammad, S. M. ja Turney, P. D. (2013). ”Crowdsourcing a word–emotion association lexicon”. *Computational intelligence* 29, s. 436–465. DOI: 10.1111/j.1467-8640.2012.00460.x.

Mozetič, I., Grčar, M. ja Smailović, J. (2016). ”Multilingual Twitter Sentiment Classification: The Role of Human Annotators”. *PLOS ONE* 11(5). DOI: 10.1371/journal.pone.0155036.

Nielsen, F. Å. (2011). ”A new ANEW: Evaluation of a word list for sentiment analysis in microblogs”. *Proceedings of the ESWC2011 Workshop on ’Making Sense of Microposts’: Big things come in small packages*, s. 93–98. URL: https://arxiv.org/abs/1103.2903.

Nukarinen, V. (2018). ”Automated text sentiment analysis for Finnish language using deep learning”. Diplomityö. Tampereen teknillinen yliopisto. URL: http://urn.fi/URN:NBN:fi:tty-201805221790.

Pirinen, T. A. (2015). ”Development and Use of Computational Morphology of Finnish in the Open Source and Open Science Era: Notes on Experiences with Omorfi Development”. *SKY Journal of Linguistics* 28, s. 381–393.

Princeton University (2010). *About WordNet*. WordNet. URL: https://wordnet.princeton.edu/ (viitattu 05. 04. 2021).

Project Gutenberg (n.d.). URL: https://www.gutenberg.org/ (viitattu 18. 03. 2021).

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/ (viitattu 27. 03. 2021).

Robinson, D. (2020). *gutenbergr: Download and Process Public Domain Works from Project Gutenberg*. Versio 0.2.0. URL: https://CRAN.R-project.org/package=gutenbergr (viitattu 18. 03. 2021).

Silge, J. ja Robinson, D. (2016). ”tidytext: Text Mining and Analysis Using Tidy Data Principles in R”. *Journal of Open Source Software* 1(3), s. 37. DOI: 10.21105/joss.00037.

Silge, J. ja Robinson, D. (2017). *Text Mining with R*. Sebastopol: O'Reilly Media.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K. ja Yutani, H. (2019). ”Welcome to the tidyverse”. *Journal of Open Source Software* 4(43), s. 1686. DOI: 10.21105/joss.01686.

Öhman, E. (2021). ”The Language of Emotions : Building and Applying Computational Methods for Emotion Detection for English and Beyond”. Tohtorinväitöskirja. Helsingin yliopisto. URL: http://urn.fi/URN:ISBN:978-951-51-7106-1.

Liite: R-koodi

Tuodaan paketit ja asetetaan työkansio

```
library(gutenbergr)
library(dplyr)
library(tidytext)
library(tidyr)
library(stringr)

setwd("~/UTA/Kandi")
```

Haetaan listaukset Fjodor Dostojevskin teoksista ja muusta fiktioista

```
works.d <- gutenberg_works(author == "Dostoyevsky, Fyodor")
works.o <- gutenberg_works(str_detect(gutenberg_bookshelf, "Fiction"))
```

Otetaan otos teoksista

```
set.seed(1984)
d.sample <- sample_n(works.d, 6)
o.sample <- sample_n(works.o, 24)
```

Gutenberg muutti latauspolkuja, eikä gutenbergr ole vielä päivittynyt, joten haetaan tiedostot paikallisesti

```
d.paths <- paste0(".\\aineisto\\", d.sample$gutenberg_id, ".zip")
o.paths <- paste0(".\\aineisto\\", o.sample$gutenberg_id, ".zip")

d.books <- gutenberg_download(d.sample$gutenberg_id, files=d.paths)
o.books <- gutenberg_download(o.sample$gutenberg_id, files=o.paths)
```

Muunnetaan kirjat tidy text -formaattiin

```
d.tidy <- d.books %>%
  group_by(gutenberg_id) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)

o.tidy <- o.books %>%
  group_by(gutenberg_id) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)
```

Lasketaan sentimentit kirjan osista ja lisätään kirjoittaja koodattuna (1 = Dostojevski)

```
d.sent <- d.tidy %>%
  inner_join(get_sentiments("nrc")) %>%
  count(gutenberg_id, index = linenumber %/% 500, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(author = 1)

o.sent <- o.tidy %>%
  inner_join(get_sentiments("nrc")) %>%
  count(gutenberg_id, index = linenumber %/% 500, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(author = 0)
```

Yhdistetään datan osat

```r
sent <- rbind(d.sent,o.sent)
```

Luodaan logistinen regressiomalli ja työstetään sitä askeltavasti

```r
logit1 <- glm(author ~ . - gutenberg_id - index, data = sent, family="binomial")
summary(logit1)
```

```
##
## Call:
## glm(formula = author ~ . - gutenberg_id - index, family = "binomial",
##     data = sent)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5684  -0.7492  -0.2260   0.8155   2.3676
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.884927   0.540932  -3.485 0.000493 ***
## anger         0.001838   0.008405   0.219 0.826922
## anticipation -0.010421   0.007556  -1.379 0.167834
## disgust       0.090806   0.011945   7.602 2.91e-14 ***
## fear         -0.047831   0.007457  -6.414 1.42e-10 ***
## joy           0.007806   0.008122   0.961 0.336537
## negative      0.013560   0.007149   1.897 0.057865 .
## positive     -0.005912   0.004595  -1.286 0.198289
## sadness      -0.011132   0.008586  -1.297 0.194772
## surprise      0.046298   0.010363   4.468 7.90e-06 ***
## trust        -0.006388   0.006154  -1.038 0.299301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 518.08  on 527  degrees of freedom
## AIC: 540.08
##
## Number of Fisher Scoring iterations: 5
```

```r
BIC(logit1)
```

```
## [1] 587.2443
```

```r
logit2 <- update(logit1, . ~ . - anger)
summary(logit2)
```

```
##
## Call:
## glm(formula = author ~ anticipation + disgust + fear + joy +
##     negative + positive + sadness + surprise + trust, family = "binomial",
##     data = sent)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5814  -0.7540  -0.2253   0.8182   2.3631
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.913148   0.526712  -3.632 0.000281 ***
## anticipation -0.010379   0.007556  -1.374 0.169573
## disgust       0.091340   0.011710   7.800 6.18e-15 ***
## fear         -0.047281   0.007016  -6.739 1.59e-11 ***
## joy           0.007656   0.008092   0.946 0.344070
## negative      0.013890   0.006987   1.988 0.046831 *
## positive     -0.005810   0.004573  -1.271 0.203883
## sadness      -0.010985   0.008567  -1.282 0.199745
## surprise      0.046560   0.010303   4.519 6.21e-06 ***
## trust        -0.006462   0.006146  -1.051 0.293061
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 518.13  on 528  degrees of freedom
## AIC: 538.13
##
## Number of Fisher Scoring iterations: 5
```

```
BIC(logit2)
```

```
## [1] 581.0043
```

```
logit3 <- update(logit2, . ~ . - joy)
summary(logit3)
```

```
##
## Call:
## glm(formula = author ~ anticipation + disgust + fear + negative +
##     positive + sadness + surprise + trust, family = "binomial",
##     data = sent)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5762  -0.7479  -0.2339   0.8101   2.4940
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.036364   0.514586  -3.957 7.58e-05 ***
## anticipation -0.006892   0.006528  -1.056   0.2910
## disgust       0.092956   0.011592   8.019 1.07e-15 ***
## fear         -0.047712   0.007024  -6.793 1.10e-11 ***
## negative      0.011914   0.006661   1.789   0.0737 .
## positive     -0.003548   0.003892  -0.912   0.3620
## sadness      -0.009056   0.008290  -1.092   0.2747
## surprise      0.048846   0.010022   4.874 1.10e-06 ***
## trust        -0.006993   0.006105  -1.145   0.2521
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 519.03  on 529  degrees of freedom
## AIC: 537.03
##
## Number of Fisher Scoring iterations: 5
```

```
BIC(logit3)
```

```
## [1] 575.6161
```

```
logit4 <- update(logit3, . ~ . - positive)
summary(logit4)
```

```
##
## Call:
## glm(formula = author ~ anticipation + disgust + fear + negative +
##     sadness + surprise + trust, family = "binomial", data = sent)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6024  -0.7382  -0.2327   0.8071   2.4451
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.061650   0.513959  -4.011 6.04e-05 ***
## anticipation -0.007737   0.006456  -1.199   0.2307
## disgust       0.094331   0.011527   8.184 2.76e-16 ***
## fear         -0.047564   0.007005  -6.790 1.13e-11 ***
## negative      0.011016   0.006587   1.672   0.0944 .
## sadness      -0.008319   0.008249  -1.008   0.3132
## surprise      0.047657   0.009904   4.812 1.49e-06 ***
## trust        -0.011241   0.003967  -2.834   0.0046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 519.86  on 530  degrees of freedom
## AIC: 535.86
##
## Number of Fisher Scoring iterations: 5
```

```
BIC(logit4)
```

```
## [1] 570.1614
```

```
logit5 <- update(logit4, . ~ . - sadness)
summary(logit5)
```

```
##
## Call:
## glm(formula = author ~ anticipation + disgust + fear + negative +
##     surprise + trust, family = "binomial", data = sent)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6426  -0.7537  -0.2294   0.8080   2.4141
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.998288   0.507187  -3.940 8.15e-05 ***
## anticipation -0.008574   0.006408  -1.338  0.18087
## disgust       0.092727   0.011360   8.163 3.28e-16 ***
## fear         -0.048709   0.006949  -7.009 2.40e-12 ***
## negative      0.007962   0.005843   1.363  0.17296
## surprise      0.047515   0.009892   4.804 1.56e-06 ***
## trust        -0.011035   0.003959  -2.787  0.00532 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 520.88  on 531  degrees of freedom
## AIC: 534.88
##
## Number of Fisher Scoring iterations: 5
```

```r
BIC(logit5)
```

```
## [1] 564.8942
```

```r
logit6 <- update(logit5, . ~ . - anticipation)
summary(logit6)
```

```
##
## Call:
## glm(formula = author ~ disgust + fear + negative + surprise +
##     trust, family = "binomial", data = sent)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5799  -0.7477  -0.2216   0.8003   2.2775
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.104807   0.503794  -4.178 2.94e-05 ***
## disgust      0.093812   0.011318   8.289  < 2e-16 ***
## fear        -0.048592   0.006905  -7.037 1.96e-12 ***
## negative     0.007353   0.005808   1.266    0.205
## surprise     0.040764   0.008421   4.841 1.29e-06 ***
## trust       -0.014121   0.003272  -4.316 1.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 522.69  on 532  degrees of freedom
```

```
## AIC: 534.69
##
## Number of Fisher Scoring iterations: 5
```

```
BIC(logit6)
```

```
## [1] 560.4183
```

```
logit7 <- update(logit6, . ~ . - negative)
summary(logit7)
```

```
##
## Call:
## glm(formula = author ~ disgust + fear + surprise + trust, family = "binomial",
##     data = sent)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5975  -0.7584  -0.2379   0.8160   2.2154
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.883704   0.463628  -4.063 4.85e-05 ***
## disgust      0.101013   0.009972  10.130  < 2e-16 ***
## fear        -0.043632   0.005587  -7.809 5.75e-15 ***
## surprise     0.042443   0.008357   5.079 3.80e-07 ***
## trust       -0.013578   0.003251  -4.177 2.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.56  on 537  degrees of freedom
## Residual deviance: 524.30  on 533  degrees of freedom
## AIC: 534.3
##
## Number of Fisher Scoring iterations: 5
```

```
BIC(logit7)
```

```
## [1] 555.7392
```

Lopullisessa mallissa siis disgust, fear, surprise ja trust, otetaan ylös myös vetosuhteet

```
exp(coef(logit7))
```

```
## (Intercept)     disgust        fear    surprise       trust
##   0.1520260   1.1062915   0.9573063   1.0433566   0.9865133
```

Katsotaan kuinka hyvin malli luokittelee sen luomiseen käytetyn datan

```
author.pred <- ifelse(predict(logit7, sent, type="response") > 0.5, 1, 0)
mean(author.pred == sent$author)
```

```
## [1] 0.7472119
```

Otetaan uusi otos, josta on poistettu mallin luomiseen käytetyt kirjat, ja tehdään näille samanlaiset sentimenttianalyysit

```r
outofsample.d <- works.d %>%
  anti_join(d.sample)

outofsample.o <- works.o %>%
  anti_join(o.sample)

newsample.d <- sample_n(outofsample.d, 2)
newsample.o <- sample_n(outofsample.o, 12)

nd.paths <- paste0(".\\aineisto\\", newsample.d$gutenberg_id, ".zip")
no.paths <- paste0(".\\aineisto\\", newsample.o$gutenberg_id, ".zip")

nd.books <- gutenberg_download(newsample.d$gutenberg_id, files=nd.paths)
no.books <- gutenberg_download(newsample.o$gutenberg_id, files=no.paths)

nd.tidy <- nd.books %>%
  group_by(gutenberg_id) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)

no.tidy <- no.books %>%
  group_by(gutenberg_id) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)

nd.sent <- nd.tidy %>%
  inner_join(get_sentiments("nrc")) %>%
  count(gutenberg_id, index = linenumber %/% 500, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(author = 1)

no.sent <- no.tidy %>%
  inner_join(get_sentiments("nrc")) %>%
  count(gutenberg_id, index = linenumber %/% 500, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(author = 0)

nsent <- rbind(nd.sent,no.sent)
```

Katsotaan, miten hyvin malli luokittelee ulkopuoliset kirjat

```r
n.pred <- ifelse(predict(logit7, nsent, type="response") > 0.5, 1, 0)
mean(n.pred == nsent$author)
```

```
## [1] 0.7564103
```