Tampereen yliopisto

Teemu Mikkonen

HAKEMUSTEN AUTOMAATTINEN LUOKITTELU LUONNOLLISEN KIELEN KÄSITTELYN KEINOIN

Automatic application classification utilizing Natural Language Processing

Diplomityö
Tekniikan ja luonnontieteiden tiedekunta
Professori Samuli Pekkola
TkT Jukka Huhtamäki
Toukokuu 2020

TIIVISTELMÄ

Teemu Mikkonen: Hakemusten automaattinen luokittelu luonnollisen käsittelyn keinoin
Diplomityö
Tampereen yliopisto
Tietojohtaminen
05/2020

Luonnollisen kielen käsittely eli Natural Language Processing (NLP) tarkoittaa tekstimuotoisen datan koneellista tulkitsemista, käsittelyä ja tuottamista esimerkiksi koneoppimisen keinoin. Tässä diplomityössä tutkitaan, miten NLP-tekniikoiden avulla voidaan luokitella tekstimuotoista dataa. Tavoitteena on selvittää, mitkä valituista NLP-tekniikoista soveltuvat parhaiten luokittelemaan Business Finlandin rahoitushakemuksia luokkiin 'cleantech' ja 'ei cleantech'. Tutkimus jakautuu teoreettiseen osioon ja empiiriseen tutkimusosioon. Diplomityön tutkimusaineistona käytetään rahoitushakemuksia ja niille annettuja luokituksia.

Teoreettisessa osuudessa tutkitaan, minkälaista dataa luonnollinen kieli on, ja määritellään, mitä NLP-tekniikoilla tarkoitetaan. Teoriaosuudessa tutkitaan myös, miten tekstimuotoista dataa esikäsitellään ja miten sitä voidaan käyttää koneoppimistarkoituksessa. Esikäsittelyn ja koneoppimisen lisäksi tutkitaan empiirisessä tutkimusosiossa käytettävien luokittelumallien taustaa. Lisäksi taustoitetaan vertailuun käytettäviä metriikoita sekä niiden soveltuvuutta mittaamaan valittujen luokittelumallien toimivuutta.

Työn empiirisessä osuudessa rakennetaan vertailtavat luokittelumallit Python-ohjelmointikielellä. Tutkimusaineisto luetaan ja esikäsitellään siihen muotoon, jota kukin luokittelumalli voi käyttää. Tutkimuksessa käytetään säännöllisiin lausekkeisiin perustuvaa luokittelumallia, ohjattuun koneoppimiseen perustuvaa luokittelumallia sekä puoliohjattuun koneoppimiseen perustuvaa luokittelumallia. Jokaisen mallin toimintaa arvioidaan sekaannusmatriisin avulla, joka vertaa luokittelumallin antamaa luokittelutulosta testijoukon oikeaan luokitukseen. Sekaannusmatriisin arvojen avulla lasketaan jokaiselle mallille niiden toimintaa kuvaavat metriikat. Metriikoiden avulla arvioidaan mallin soveltuvuutta cleantech-hankkeiden luokitteluun.

Perusteellinen esikäsittely havaitaan tutkimuksessa hyvin tärkeäksi osaksi NLP-prosessia, sillä se mahdollistaa tekstin muuntamisen vektorimuotoon piirteiden erottamisen ja ulottuvuuksien vähentämisen avulla. Vektorimuotoisten dokumenttien esittäminen semanttisessa vektoriavaruudessa mahdollistaa mm. tekstien vertailun niiden merkitykseen, eli semanttiseen informaatioon perustuen. Kirjallisuudesta havaitaan, että luokitteluongelman arviointia on harhaanjohtavaa tehdä vain yhden mittarin avulla. Tämän takia tarvitaan kokonaisvaltaisempia mittaristoja. Tarve korostuu, mikäli eri luokkien alkioita on opetusdatassa epätasainen määrä.

Tutkimuksen tuloksena on, että useimmilla mittareilla koneoppimiseen pohjautuvat mallit tuottavat parempia luokittelutuloksia kuin säännöllisiin lausekkeisiin perustuva malli. Ohjatun koneoppimismallin tunnusluvut viittaavat sen soveltuvan tilanteeseen, jossa on tärkeää tunnistaa kaikista luokitelluista alkioista mahdollisimman monta relevanttia tulosta ja virheellisen 'ei cleantech' luokituksen haitta on pieni. Puoliohjattu koneoppimismalli sopii tilanteeseen, jossa virheellisen positiivisen cleantech-luokituksen haitta on pieni ja halutaan löytää mahdollisimman luotettavasti todelliset cleantech-luokitukset. Säännöllinen lauseke on useimmilla mittareilla arvioituna heikko, eikä sitä voida pitää soveltuvana luokittelijaksi.

Tutkimuksesta ilmenee, että Business Finlandin tapauksessa cleantech-hakemusten luokitteluun parhaiten soveltuu ohjattu koneoppimismalli. Sen lisäksi, että se suoriutuu luokittelusta useimpien metriikoiden valossa parhaiten, se ei tuota herkästi virheellisiä positiivisia 'cleantech' luokituksia. Virheelliset positiiviset luokitukset suosittelevat cleantech-luokitusta hakemuksille, jotka eivät sisälly cleantech-hankkeiden määritelmään.

Avainsanat: Luonnollisen kielen käsittely, NLP, fastText, Latent Semantic Indexing, luokittelu, koneoppiminen

ABSTRACT

Teemu Mikkonen: Automatic application classification utilizing Natural Language Processing
Master's Thesis
Tampere University
Information and Knowledge Management
05/2020

Natural language processing means the computational interpretation, processing and producing of data in a text format. This Master's thesis researches how NLP can be used for text classification. The aim of this research is to find which ones of the selected NLP technologies are the best suited for classification of Business Finland's funding applications to classes 'cleantech' or 'not cleantech'. The research consists of literature section and empirical research. The research material used in this research are funding applications and their corresponding classes.

In the literature section, natural language's attributes are researched, and NLP-technologies are defined in the context of this thesis. Literary section also includes research on how text data is preprocessed and how it can be used in machine learning. In addition to preprocessing and machine learning, the literature section also studies the background of classifiers that are used in the empirical research section. Also, the theory and suitability behind the metrics used to compare the chosen classifiers is studied.

In the empirical section, the classifier models chosen for comparison are built using Python programming language. The data is read and preprocessed to a format that can be used in classification by each classifier. The chosen classifier models in this thesis are a regular expression-based model, a supervised machine learning model and a semi-supervised machine learning model. The performance of the models is evaluated by using a confusion matrix, that compares the classification given by a classifier with the correct classification of test documents. Confusion matrix enables the calculation of metrics that are used in the evaluation of the models, based on which the models' suitability for classifying cleantech-application are assessed.

Thorough preprocessing was found to be imperative in the NLP-process because it enables altering text into vector format by feature extraction and dimension reduction. Presenting documents in vectorized form in a dense vector space enables e.g. comparing vectors based on their semantic information. Literature shows that it is misleading to measure a classifying problem by only using single metric and more comprehensive set of metrics is required. This requirement is increased if there are is an imbalanced number of items in the classes in question.

The research shows that on most metrics, the machine learning-based models outperform regular expression-based model. The score of the supervised model indicate that it is best suited for a scenario, where finding maximum number of items belonging to the class 'cleantech' is important and the cost of false negative 'not cleantech' is low. Semi-supervised model however appears to be best suited for a scenario where the aim is to find the items belonging to class 'cleantech' as reliably as possible and the cost of false positive 'cleantech' classifications is low. Regular expression-based model's performance is ranked very low by most metrics and thus it is not applicable as a classifier in this context.

The research shows that in Business Finland's case the supervised machine learning model is the best suited for classification of cleantech applications. This is because it does not produce a high number of false positives cleantech classifications. False positive classifications can recommend cleantech classification to projects that are not cleantech related.

Keywords: Natural Language Processing, NLP, fastText, Latent Semantic Indexing, classification, machine learning

ALKUSANAT

Tämä diplomityö toteutettiin Business Finlandin antamaan tutkimusaiheeseen. Kokemus on ollut erittäin monipuolinen ja opettavainen. Haluan kiittää sekä Business Finlandia että työnantajaani Solitaa tämän diplomityön mahdollistamisesta. Kiitos kuuluu myös tämän työn erinomaisille ohjaajille Timo Lehtoselle, Jukka Huhtamäelle ja Samuli Pekkolalle, joiden tukeen ja ohjaukseen pystyi aina luottamaan työn edetessä.

Diplomityöni aihe ja opintojeni suuntautuminen eivät olleet sitä, mitä opiskelun alkuaikoina luulin päätyväni tekemään. Fuksivuoden rientojen vuoksi hieman alavireisesti menneiden vektorilaskennan, todennäköisyyslaskennan ja Johdatus ohjelmointiin -kurssin jälkeen on jokseenkin ironista, että löysin ammatillisen kiinnostuksen kohteeni juuri näitä taitoja hyödyntävien koneoppimisen ja datatieteiden parista. Tästä kuuluu kiitos myös tätä diplomityötä ohjanneelle Jukka Huhtamäelle, jonka kursseista kiinnostukseni datatieteisiin heräsi.

Suurin osa tästä diplomityöstä kirjoitettiin kevään 2020 koronakaranteenin aikana. Eristäytyminen, tilanteen epäselvyys ja viimeisen (opiskeluaikaisen) Teekkariwappuni siirtyminen syksylle varjostivat tunnelmaa. Diplomityön suoritusprosessissa tulin usein kuitenkin pohtineeksi ja muistelleeksi opiskeluaikaani. Kuuteen vuoteen mahtuu uskomaton määrä muistoja tapahtumista, pippaloista ja kiltahuoneella istumisesta mutta päällimmäisenä mieleen jäävät opiskelutoverini, joiden kanssa sain tämän opiskelutaipaleeni jakaa.

Haluan kiittää rakasta kiltaani, Tietojohtajakilta Man@geria, jonka toiminnassa mukana oleminen sekä hallituksessa että jäsenistössä oli opintoaikojeni ehdoton kohokohta. Erityiskiitos Hallitus 2017 sekä rakkaat Nottikset! Kiitos myös Milla Väänäselle kärsivällisyydestä, avusta ja oikoluvusta. Kiitos tuesta myös perheelleni ja etenkin äidilleni Satu Herralalle, joka myös toimi tämän työn kieliopillisena oikolukijana. Lopuksi haluan kiittää vielä Tampereen teknillistä yliopistoa, jonne opiskelemaan hakeminen oli elämäni parhaita päätöksiä.

Tampereen Hervannassa 15.5.2020

Teemu Mikkonen

SISÄLLYSLUETTELO

KUVALUETTELO

TAULUKKOLUETTELO

LYHENTEET JA MERKINNÄT

AUC	Area Under (ROC) -curve, ROC-käyrän alle jäävä pinta-ala
BOW	Bag of Words
CBOW	Continuous bag of words, jatkuva bag of words.
FN	False Negative, virheellinen negatiivinen
FP	False Positive, virheellinen positiivinen
kNN	k-Nearest Neighbor -algoritmi
NLP	Natural language processing, luonnollisen kielen käsittely
PR	Tarkkuus-saanti-käyrän alle jäävä pinta-ala
SMOTE	Synthetic minority over-sampling technique
ROC	Receiver Operating Characteristic
TF-IDF	Term Frequency – Inverse Document Frequency, termifrekvenssi – käänteinen dokumenttifrekvenssi
TN	True negative, todellinen negatiivinen
TP	True positive, todellinen positiivinen

1. JOHDANTO

Koneoppimisen tai tekoälyn hyödyntämisestä viranomaistoiminnassa voidaan tunnistaa eettisiä haasteita, kuten esimerkiksi inhimillisen näkökulman huomioiminen, päätöksenteon perusteiden läpinäkymättömyys ja luottamuksen heikkeneminen viranomaiseen. Vastavuoroisesti voidaan kuitenkin nähdä tekoälyratkaisujen hyödyttävän julkisia organisaatioita mm. resurssitehokkuuden, aikariippumattomuuden ja yhdenmukaisen kohtelun takaamisen avulla. (Koivisto et al. 2019) Coglianese & Lehr (2016) mukaan koneoppiminen päätöksenteon tukena voi auttaa julkisia organisaatioita tekemään tarkempia ja parempia päätöksiä, joka puolestaan hyödyttävät koko yhteiskuntaa. Koneoppimisen ja tekoälyn hyödyntäminen julkisten organisaatioiden päätöksenteossa edellyttää siis eettisten näkökulmien huomioon ottamista, mutta onnistuessaan se voi luoda arvoa sekä organisaatiolle että yhteiskunnalle.

Koneoppimisavusteisen päätöksenteon kohteena tässä tutkimuksessa on rahoitushakemuksen kuuluminen luokkaan cleantech. Cleantech valikoitui tähän tutkimukseen muista Business Finlandin luokituksista merkityksellisyytensä ja raportointivelvoitteensa vuoksi. Business Finland raportoi Työ- ja elinkeinoministeriölle toimintaansa Työ- ja elinkeinoministeriön asettaman tulossopimuksen mittarien mukaisesti. Mittareihin kuuluu ”*Biotalous- ja cleantech-ratkaisuja kehittävien pk-yritysten vienti / myönnetty rahoitus (innovaatiorahoitus, milj.e*)”, joka kuvaa, kuinka paljon rahoitusta biotalous- ja cleantech-hakemuksille on myönnetty. (Työ- ja elinkeinoministeriö 2018) Tämän raportointivelvoitteen vuoksi on tärkeää, että cleantech-hankkeet saadaan luokiteltua mahdollisimman luotettavasti. Cleantech-hankkeille ei kuitenkaan ole olemassa olevaa ratkaisua automaattiseen luokitukseen viranomaisen tueksi, jonka vuoksi luokittelijoiden tutkiminen juuri tähän luokitukseen on tarpeellinen.

Tulevaisuustalo Sitran määritelmän mukaan cleantech on: ”*Teknologia, tuote, palvelu, prosessi tai suljettu systeemi, joka edistää luonnonvarojen kestävää käyttöä. Maksimoi materiaali-, vesi- ja energiatehokkuuden sekä taloudellisesti että teknologisesti ja pienentää samalla päästöjä veteen, ilmaan ja maahan*” (SITRA 2020). Cleantech on siis hyvin laaja kattokäsite ympäristöystävällisille ratkaisuille, jolloin jonkin teknologian luokittelu cleantechiksi ei ole yksiselitteinen. Hankerahoitusta myönnettäessä kriteereihin perustuva ja yhdenmukainen linja on hyvin tärkeä näkökulma, jotta luokittelua voidaan toteuttaa yhtenevin perustein. Koneoppimisen avulla voidaan mahdollistaa yhdenmukainen, säännönmukainen ja reilu kohtelu (Koivisto et al. 2019). Voidaan päätellä, että päätöksenteon tukena voidaan käyttää koneoppimiseen

perustuvia ratkaisuja, jotka perustuvat aikaisempiin päätöksiin hankkeiden cleantech-kohdistumisesta.

Oikein luokiteltu cleantech-hakemus mahdollistaa sen, että tuki saadaan myönnettyä määrittelyä vastaavalle, relevantille hankkeelle. Tästä voidaan arvioida olevan sekä välitöntä taloudellista hyötyä uusien innovaatioiden kautta että välillistä ilmastolle ja ympäristölle kestävän kehityksen ansiosta.

Tässä diplomityössä tutkitaan vaihtoehtoja kehittää Business Finlandin rahoitushakemusjärjestelmän automaattista hakemusluokittelua cleantech-hankkeisiin. Automaattista, koneoppimiseen pohjautuvaa hakemusluokittelua käytetään avustamaan Business Finlandin viranomaisen päätöksentekoa rahoitushakemuksen luokittelussa.

1.1 Tutkimuskohteen esittely

Diplomityö suoritetaan Business Finlandin esittämään tutkimusongelmaan. Business Finland on julkishallinnollinen organisaatio, jonka tehtäviin kuuluu tukea innovaatioita, edistää suomalaisyritysten kansainvälistymistä ja Suomeen kohdistuvaa matkailua ja innovaatioita (Business Finland 2020e). Organisaatio on muodostunut Finpron ja Tekesin yhdistymisen seurauksena vuonna 2018. Yhdistyminen toteutettiin mm. kokonaisvaltaisemman asiakkuuselinkaaren ja eri kansainvälistymistoimintojen välisen yhteistyön kehittämiseksi. Työ- ja elinkeinoministeriö asettaa organisaatiolle strategiset tavoitteet sekä tulosohjaa sitä. (Työ- ja elinkeinoministeriö 2020) Business Finlandin visio ja missio ovat seuraavat:

Visio:

- ”Tehdään Suomesta yhdessä houkutteleva ja kilpailukykyinen innovaatioympäristö, jossa luodaan maailmanluokan menestystarinoita”
- ” Olemme asiakkaittemme halutuin innovoinnin ja globaalin kasvun kumppani.”

Missio:

- ”Luomme uutta kasvua auttamalla yrityksiä kansainvälistymään sekä rahoittamalla tutkimusta ja innovaatioita yrityksissä että tutkimusorganisaatioissa.”

(Business Finland 2020d)

Business Finlandin innovaatiorahoitushankkeet painottuvat ennalta määritettyihin teemoihin. Näitä teemoja ovat consumer business, digitalisaatio, matkailu, terveys ja

hyvinvointi, sekä biotalous ja cleantech (Business Finland 2020d). Tämän diplomityön tutkimus keskittyy biotalouden ja cleantechin teemakokonaisuuteen. Business Finland tukee yrityksiä cleantech-hankkeissa mm. innovaatiorahoituksella, luomalla kansainvälisiä ja kotimaisia verkostoja sekä tarjoamalla asiantuntijoita ja sidosryhmiä strategisen suunnittelun tueksi (Business Finland 2020a).

Business Finlandin asiakassegmentti koostuu kolmesta pääkomponentista. Näitä ovat kansainvälistä kasvua etsivät yritykset, tutkimusorganisaatiot ja julkishallinnon organisaatiot, jotka tähtäävät innovaatiokehitykseen (Business Finland 2020b). Rahoitushankkeet jakautuvat siis taustaperustaisesti, mutta myös erilaisiin rahoitustarpeisiin on luotu niihin erikoistuneet rahoituspalvelut. Esimerkkejä Business Finlandin rahoituspalveluista ovat esimerkiksi Tutkimus-, kehitys- ja innovaatiorahoitus (T&K&I), TEMPO-kansainvälistymisrahoitus sekä Research to Business -rahoitus. T&K&I-rahoitus tähtää mahdollistamaan esimerkiksi tuotteiden, palvelujen tai jopa liiketoimintamallien kehityksen tai luomiseen. Research to Business -rahoitus puolestaan tähtää tutkimustuloksista heräävien innovaatioiden kaupallistamisen tehostamiseen. TEMPO-rahoitus on suunnattu kansainvälistä kasvua etsiville yrityksille (Business Finland 2020c)

1.2 Tutkimuksen tavoitteet ja rajaukset

Diplomityön tavoitteena on muodostaa vertaileva tutkimus eri NLP-tekniikoista, joita voidaan käyttää rahoitushakemusten automaattiseen luokitteluun cleantech-hakemusten tunnistamiseksi kaikista muista rahoitushakemuksista. Tutkimus toteutetaan vertaamalla valittuja NLP-tekniikoita ihmisen luokittelemaan aineistoon, jonka perusteella voidaan arvioida teknologioiden ennusteen oikeellisuutta. Luokitteluun hyödynnettäviä NLP-tekniikoita verrataan lisäksi toisiinsa sekä kirjallisuuden että empiirisen tutkimuksen avulla. Empiirisessä tutkimuksessa verrataan Python-ohjelmointikielen avulla luotuja luokittelumalleja kirjallisuuden avulla muodostettujen arviointimetriikoiden perusteella.

Tutkimusongelmaksi on tunnistettu ”Rahoitushakemusten luokittelu luokkaan ’cleantech’”. Tutkimusongelma on jaoteltu tutkimusta varten päätutkimuskysymykseen ja sitä selventäviin apututkimuskysymyksiin. Tutkimusongelma ja tutkimuskysymykset on esitetty taulukossa 1.

Taulukko 1: Tutkimusongelma ja tutkimuskysymykset

Tutkimusongelma	Rahoitushakemusten luokittelu luokkaan ’cleantech’
Päätutkimuskysymys	Miten hakemustekstiä voidaan luokitella luonnollisen kielen käsittelyn (NLP) avulla?
Apututkimuskysymys 1	Mitä on NLP tämän tutkimuksen kontekstissa?
Apututkimuskysymys 2	Miten valittujen NLP-tekniikoiden ominaisuuksia voidaan verrata toisiinsa?
Apututkimuskysymys 3	Mitä esikäsittelytoimenpiteitä hakemustekstille on tehtävä NLP-prosessia varten?

Tutkimuksessa on rajattu teknologiat yksinomaan NLP-teknologioiden alle, joista on valittu 3 kappaletta diplomityön tutkimuksen laajuuden huomioon ottaen. Koska tutkimuksessa tutkitaan luonnollisen kielen prosessointia, myös tutkimuksessa käytettävä aineisto on rajattu tekstimuotoiseen dataan ja luokitustietoihin.

1.3 Tutkimusmetodologia

Tutkimuksen metodologia rakentuu Saunders et al. (2009 s.108) esittelemän kerroksittaisen mallin mukaisesti. Tässä luvussa on kuvattu tutkimuksen metodologia jokaisen kerroksen osalta aina tieteenfilosofisista valinnoista tutkimusmenetelmän valintaan.

1.3.1 Vaikuttavat tieteenfilosofiat

Tutkimuksen taustalla vallitsevat tieteenfilosofiat vaikuttavat tutkimuksen asetteluun määrittelemällä olettamukset, joiden pohjalta tutkittavaa ilmiötä tarkastellaan. Tieteenfilosofian ymmärtäminen vaikuttaa tutkimuksen käytännön valintojen lisäksi tapaan, jolla tutkimus tunnistaa tutkimuksen laadullisia ominaisuuksia sekä haastaa olemassa olevia ennakkokäsityksiä. (Saunders et al. 2009 ss. 107–109.; Park et al. 2020) Tieteenfilosofiset oletukset muodostavat siis tutkimukselle perustan, joka vaikuttaa sekä tutkimuksen suoritukseen että kontekstiin, jossa tutkimusta tarkastellaan. Tämän diplomityön tieteenfilosofiset taustaoletukset ovat positivismi ja pragmatismi.

Positivistinen tieteenfilosofia (positivismi) nojaa periaatteeseen, jossa on olemassa yksi, tunnistettavissa ja mitattavissa oleva todellisuus. Positivistisessa todellisuuskuvassa voidaan tehdä yksiselitteisiä olettamuksia ilmiöiden keskinäisistä suhteista. Ilmiöiltä

voidaan tunnistaa kausaliteetti, korrelaatio, sekä ilmiön ulkopuolisten vaikutteiden puute. (Park et al. 2020) Positivistisessa tutkimuksessa on myös erittäin tärkeää pyrkimys puolueettomuuteen ja objektiivisuuteen, jonka toteutuessa tutkimusta voidaan pitää luotettavana ja tarkkana. (Saunders et al. 2009 s. 114; Park et al. 2020).

Positivistisia vaikutteita tässä tutkimuksessa ovat etenkin sen perusolettamukset, kuten esimerkiksi tulosten yleistettävyys, ilmiöiden rajatut suhteet toisiinsa sekä tutkimuksen objektiivisuus. Näistä syistä positivismi on nykyisin hyvin yleinen tutkimusfilosofia tietotekniikan tutkimuksessa (Siponen & Tsohou 2018). Positivistinen filosofia on usein taustalla kvantitatiivista tutkimusta tehtäessä (Saunders et al. 2009 s.114) Myös tässä tutkimuksessa on valittu kvantitatiivinen tutkimusmenetelmänä luokittelumallien vertailuun.

Pragmatismi tieteenfilosofiana korostaa näkökulmaa, jossa ei rajoituta jyrkästi tieteenfilosofioiden ominaisuuksiin. Pragmaattisen tutkimuksen lähtökohta on käytettävyys ja sovellettavuus, ja sen pyrkimys on vastata spesifeihin kysymyksiin (Glasgow 2013). Asiakkaan tutkimusongelma ja ratkaisun laajempi konteksti vaikuttavat siis osaltaan siihen, millä parametreilla ongelmaa lähdetään tutkimaan ja tutkimuksen tuloksista saadaan vastauksia asiakkaan esittämään kysymykseen. NLP-aiheisessa tutkimuksessa on objektiivisen matemaattisen metriikan lisäksi otettava huomioon asiakkaan näkökulma tiedon hyödynnettävyydestä. Tästä syystä tutkimuksessa on myös pragmaattisia piirteitä.

Pragmaattisen tieteenfilosofian yhdistämistä empiiriseen positivistiseen kontekstiin tukee näkökulma, jossa pragmaattinen tutkimus pyrkii tutkimaan sidosryhmille tärkeitä asioita (Glasgow 2013). Pragmaattisessa tutkimuksessa nimenomaan tutkimuskysymys ohjaa tieteenfilosofisia valintoja todellisuuden ja tiedon luonteesta (Saunders et al. 2009 s.109). Koska työ toteutetaan Business Finlandille oikeaan liiketoimintaongelmaan, on syytä tarkastella absoluuttisten matemaattisten metriikoiden lisäksi myös kohdeyrityksen näkökulmaa siitä, vastaako matemaattinen malli ihmisen ymmärrystä. Tässä tutkimuksessa asetettu tutkimuskysymys ohjaa tutkimusta, joskin empiirisen tutkimusvaiheen toteutuksessa ja tulosten vertailussa nojataan vahvasti positivistiseen tieteenfilosofiaan.

1.3.2 Tutkimusstrategia

Saunders et al. (2009 s.138) mallin mukaan tieteenfilosofiset valinnat ohjaavat tutkimuksen ja teorian suhdetta. Deduktiivisen tutkimusstrategian voidaan nähdä juontuvan positivistisesta tieteenfilosofiasta, jonka vuoksi myös tässä tutkimuksessa

käytettävä teorian ja empirian suhde on deduktiivinen (Saunders et al. 2009 s.124). Deduktiivisessa päättelyssä muodostetaan havainnot ja päätelmät testaamalla olemassa olevaan teoriaan pohjautuvaa hypoteesia (Mantere & Ketokivi 2013). Deduktiivinen päättely noudattaa seuraavaa kaavaa:

1. Muodostetaan teorian pohjalta hypoteesi
2. Tutkimuksen muodostaminen tutkimukseksi (mitä mitataan, miten mitataan)
3. Tutkimuksen toteuttaminen hypoteesin pohjalta
4. Tutkimuksen lopputuloksen arviointi
5. Teorian muokkaaminen tutkimustulosten pohjalta

(Saunders et al. 2009)

Deduktiivisen päättelyyn perustuen on valittu käytettäväksi tutkimusstrategiaksi kokeellinen tutkimus perustuen kvantitatiiviseen aineistoon. Kokeellisessa tutkimusasetelmassa toteutetaan teoriasta johdetun hypoteesin mukainen tutkimus koeotokselle, jota verrataan kontrolliotokseen. Tutkimuksen toteuttamisen jälkeen koeotoksen ja kontrolliotoksen välisiä eroja voidaan mitata perustuen valittuun metriikkaan. (Saunders et al. 2009) Sovellettua koeasettelua voidaan hyödyntää tutkimuksessa, jossa arvioidaan ratkaisun kykyä vastata spesifiin tutkimusongelmaan (Edgar & Manz 2017). Tämän tutkimuksen osalta voidaan siis todeta sopivaksi tutkimusmetodologiaksi sovellettu kokeellinen tutkimus.

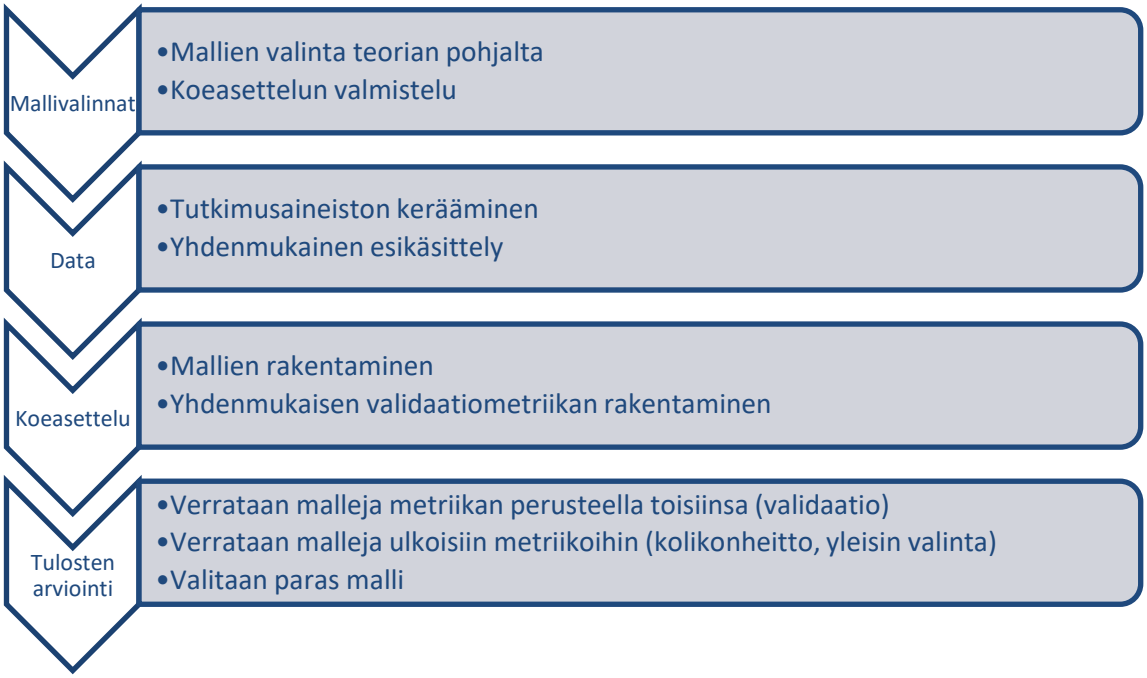
Sovelletussa kokeellisessa tutkimuksessa noudatetaan deduktiivisen päättelyn kaavaa, jossa tutustutaan ensin vallitsevaan teoriaan ja tutkittavaan järjestelmään, jonka pohjalta muodostetaan ongelman ratkaisuun tähtäävä hypoteesi. Tämän kaltaisen tutkimuksen tärkeitä osa-alueita ovat *vertailututkimus* (benchmarking) ja *validaatio*. Vertailututkimuksessa hypoteesin pohjalta kehitettyä ratkaisua verrataan esimerkiksi reaalimaailman dataan tai käyttötapauksiin. Validaatiolla testataan ratkaisun toimivuutta spesifimmässä ympäristössä, esimerkiksi tiettyyn metriikkaan perustuen. (Edgar & Manz 2017)

Tutkimukseen käytettävissä oleva aineisto koostuu Business Finlandin rahoitushakemuksista vuosilta 1995–2018. Kuitenkin tutkimusongelman kannalta relevanttia dataa on saatavilla vuosilta 2014–2018. Tutkimuksessa käytetään siis *tarkoituksenmukaista otantaa* (purposive sampling), jossa otannan perusteena on aineiston soveltuvuus vastaamaan tutkimuskysymykseen (Saunders et al. 2009 s.237). Koska tutkimusaineisto on otannan perusteella rajattu tiettyyn aikaväliin, tutkimuksen aikahorisontti on *poikittaistutkimus* (cross-sectional research) (Edgar & Manz 2017).

Tutkimuksessa käytettävä aineisto on kerättyä historiadataa, joten kyseessä on sekundäärinen aineisto (Saunders et al. 2009 s. 256). Aineiston muodostaminen on kuvattu tarkemmin luvussa 1.5.

1.4 Tutkimuksen rakenne

Tutkimuksen rakenne noudattaa luvussa 1.3.2. kuvattua tutkimusstrategiaa ja deduktiivisen päättelyn ja kokeellisen tutkimuksen kaavaa, jossa lähdetään liikkeelle teoriasta ja päädytään joko validoimaan tai kumoamaan sen perustalle rakennettu hypoteesi. Tutkimuksen rakenne on esitetty kuvassa 1.



Kuva 1: Vertailututkimuksen rakenne

Tutkimuksessa tutustutaan NLP:n yleiseen teoriaan kirjallisuutta hyödyntäen ja valitaan kolme soveltuvaa teknologiaa, joiden teoriaan syvennytään tarkemmin. Teorian pohjalta muodostetaan hypoteesit, joiden pohjalta rakennetaan tutkittavia malleja. Myös mallien vertaamiseen käytetyt metriikat valitaan kirjallisuuden pohjalta.

Tutkimusaineisto saadaan kohdeyritykseltä käsittelemättömässä muodossa. Tutkimusaineisto tehdään yhdenmukainen esikäsittely, jotta voidaan varmistua mallien vertailukelpoisuudesta. Koeasettelussa rakennetaan mallit teorian pohjalla hyödyntäen

Python-ohjelmointikieltä ja Jupyter Notebook-alustaa. Pythonia hyödynnetään siihen rakennettujen koneoppimis-, statistiikka- ja NLP-kirjastojen vuoksi.

Koeasettelussa rakennetaan soveltuvien teknologioiden pohjalta luokittelumallit, joiden avulla luokitellaan rahoitushakemuksia ”cleantech”-sisällön perusteella. Tämän jälkeen malleja verrataan toisiinsa käyttämällä kirjallisuuden avulla valittuja metriikoita. Yhdenmukainen metriikka toimii sovelletun koeasetelman validaatiomenetelmänä. Kokeessa on tärkeää myös verrata luokittelumalleja koeasetelman ulkopuolisiin luokittelijoihin, jotta luokittelijoiden tunnusluvut voidaan sitoa käytännön kontekstiin. Tutkimuksessa verrataan malleja esimerkiksi 50%-tarkkuudella toimivaan satunnaismuuttujaa, jota kuvataan tutkimuksessa kolikonheittomallina. Ulkoisiin ja käytännönläheisiin luokittelijoihin vertaaminen tuo tutkimukseen pragmaattisen näkökulman ja havainnollistaa luokittelijan toimintaa käytännön kontekstissa.

1.5 Tutkimusaineiston muodostaminen

Tutkimuksen aineisto koostuu vuosina 2014–2018 luoduista suomenkielisistä rahoitushakemuksista ja niiden luokituksista. Tutkimusaineisto on siis kerätty yhdistämällä hakemuksen laatijan kirjoittamat hakemustekstit Business Finlandin työntekijän asettamaan luokitukseen. Hakemuksen laatija on myös valinnut hakemukselle jonkin Business Finlandin tarjoaman rahoituspalvelun. Koska kaikki rahoituspalvelut eivät sisällä cleantech-luokitusta, rajataan tutkimus niihin rahoituspalveluihin, jotka soveltuvat rahoituksen hakemiseen cleantech-hankkeelle. Tämä varmistaa, että tutkimusaineistossa on vain rahoitushakemuksia, jotka joko ovat cleantech-luokiteltuja tai niille on ollut ylipäätään mahdollista antaa cleantech-luokitus.

Business Finlandin rahoitushakemusta tehdessä tulee hakijan vastata palvelussa esitettyihin kysymyksiin. Rahoitushakemus koostuu hakijan vastauksista, ja pitää sisällään kuvauksen hakijan liiketoiminnan nykytilasta, henkilöstöstä ja resursseista, yrityksen kasvuvisiosta sekä rahoituksen kohdeprojektin tavoitteista, suunnitelmasta ja kustannusarviosta. Tämän lisäksi hakemus sisältää hakijayrityksen taloustietoja, esimerkiksi tuloslaskelman ja taseen. (Business Finland 2020f) Tämän tutkimuksen empiirisessä osuudessa hyödynnetään vain hakemukseen liittyvät tekstimuotoisten kysymysten vastaukset, sillä tutkimus on rajattu tekstimuotoisen datan luokitteluun.

2. LUONNOLLISEN KIELEN KÄSITTELY

2.1 Mitä NLP on?

Luonnollisen kielen käsittely tai Natural Language Processing (NLP) voidaan hahmottaa monitieteellisenä pyrkimystä prosessoida, ymmärtää tai tuottaa luonnollista kieltä (esim. suomi, englanti) koneellisesti (Deng & Liu 2018 s.1). NLP:tä voidaan kuvata myös tietotekniikan ja koneellisen lingvistiikan tutkimusalueena, jossa rakennetaan luonnollisen kielen rakenteita (sana, lause, dokumentti) hyödyntäviä sovelluksia (Cohen & Demner-Fushman 2014 ss.1-2). NLP-nimityksellä tarkoitetaan usein myös koneoppimisen ja tekoälyn alakategoriaa, jonka avulla voidaan sekä käsitellä kirjoitettua kieltä että jäsentää puhuttua kieltä tekstiformaattiin (Martinez 2010). Tämän diplomityön kontekstissa NLP käsitetään koneoppimisen teknologioina, joita voidaan käyttää tekstidokumentteina esitetyn kielen prosessointiin.

NLP:n avulla pyritään siis hyödyntämään luonnollista kieltä datalähteenä. NLP:llä on paljon kaupallisia ja arkisia sovelluskohteita. NLP-teknologioiden yleisiä käyttökohteita ovat mm. hakukoneet, puheentunnistus, automaattiset käännössovellukset, ihmisen ja koneen rajapinnat (esim. chatbotit) sekä informaation keruu ja koneelliset tiivistelmät päätöksenteon tueksi (Laippala et al. 2014; Hirschberg & Manning 2015; Aggarwal 2018 s.2; Deng & Liu 2018 s.1). NLP:tä voidaan siis hyödyntää lähestulkoon kaikilla osa-alueilla, joissa on olemassa tekstimuotoista dataa. Tämän työn kontekstissa NLP:n käytännön osa-alueista pureudutaan informaation keruuseen tekstimassasta.

NLP:n kehitykseen ovat vaikuttaneet samat tekijät, jotka ovat laajemminkin vaikuttaneet koneoppimisen kehitykseen: laskentatehon kasvu, koneoppimisteknologioiden kehitys, kehittynyt ymmärrys luonnollisen kielen rakenteesta ja käytöstä eri konteksteissa ja luonnollisen kielen datamäärän kasvu (Hirschberg & Manning 2015) Digitaaliset datalähteet, kuten digikirjastot, verkkouutiset, verkkosivustot ja sosiaalinen media ovat syy luonnollisen kielen datamäärän kasvuun (Aggarwal 2018 ss. 1–3). Digitaalisen, tekstimuotoisen datan määrä ja kehittynyt kyky muokata teksti käsiteltävään ja laskettavaan muotoon ovat siis olleet NLP:n suurimpia muutosajureita.

NLP:n tutkimukseen vaikuttaa laaja kirjo eri tutkimusaloja, esimerkiksi koneoppiminen kognitiotieteet, lingvistiikka ja tietotekniikka (Deng & Liu 2018 s.1). Tämän diplomityön kontekstissa NLP:tä lähestytään insinööritieteiden, tietotekniikan ja koneoppimisen lähtökohdista.

2.2 Luonnollinen kieli datana

Data voidaan jakaa karkeasti kolmeen luokkaan. *Rakenteellinen data* (structured data), kuvaa esimerkiksi relaatiotietokantaan tallennettua määrämuotoista dataa. *Puolirakenteellinen data* (semi-structured data) ei ole taulukkomuotoista, mutta se on jaoteltu loogisiin kokonaisuuksiin esimerkiksi tunnisteiden perusteella (esimerkiksi HTML-tunnisteet). *Rakenteeton data* (unstructured data) ei ole määrämuotoista, ja näin ollen sitä on myös vaikea analysoida. (Sagiroglu & Sinanc 2013) Luonnollinen kieli datana luokitellaan rakenteettomaksi dataksi, sillä ei ole eksplisiittistä tai määrämuotoista.

Vaikka luonnollinen kieli datalähteenä luokitellaan rakenteettomaksi, se kuitenkin perustuu sääntöihin ja sisäisiin rakenteisiin. Kunkin luonnollisen kielen säännöt määritellään kieliopin ja oikeinkirjoituksen kautta. Nämä rakenteet muodostavat kielen *syntaksin.* (Martinez 2010) Syntaktisten rakenteiden välisiä suhteita ja niiden merkitystä tulkitsijalle kutsutaan *semantiikaksi* (Martinez 2010; Altınel & Ganiz 2018)**.** Semantiikka tarkoittaa siis jonkin asian merkitystä, jonka viestimiseen syntaksi muodostaa tarvittavat rakenteet.

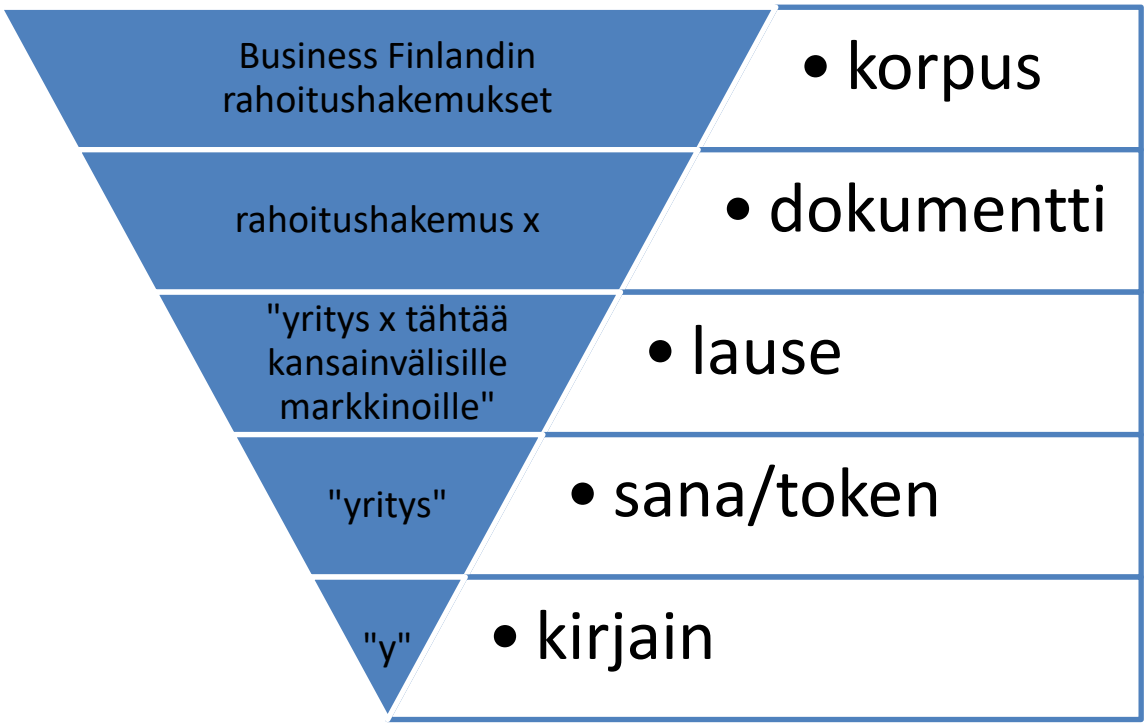
Luonnollinen kieli on datana ei ole yksiselitteistä. Haasteita voi syntyä mm. synonyymeista tai homonyymeistä. Myös usealla eri ilmaisutavalla voidaan tarkoittaa tismalleen samaa asiaa. (Goldberg 2017 ss. 1–2) Esimerkiksi suomen kielen lausahduksen, ”kuusi palaa”, voidaan ymmärtää monella tavalla, eikä sitä voi yksiselitteisesti ymmärtää ilman kontekstia.

Myös subjektiivisuus vaikuttaa luonnollisen kielen ymmärtämiseen: kaksi eri ihmistä voi ymmärtää saman tekstin eri tavoin omaan kokemukseensa perustuen ja esimerkiksi luokitella sen eri tavalla. (Deng et al. 2019) Voidaan päätellä, ettei sanan tai lauseen semantiikka ole absoluuttinen, vaan siihen voi vaikuttaa mm. ympäröivä konteksti ja subjektiivinen havainnoija.

Tekstin analysointiin voidaan käyttää eri kokoisia luonnollisen kielen rakenteita. Matalin näistä rakenteista on *kirjain*. Eri aakkoset sisältävät erilaisia joukkoja kirjaimia, numeroita tai jopa sanoja. Rajattu määrä kirjaimia muodostaa sanan eli *merkkijonon* (string). (Clark et al. 2013 s. xxxix) Tekstijono ei ole kuitenkaan rajoitettu pelkkiin kirjaimiin, vaan siihen voidaan lukea myös välimerkit. Tätä tekstijonosta voidaan käytetään ilmaisua *sana* tai *token* (Cohen & Demner-Fushman 2014 s.4). Tiettyyn luonnolliseen kieleen kuuluvia

sanoja kutsutaan *sanastoksi* (lexicon tai dictionary) (Martinez 2010). Seuraava rakenteellinen taso on lause, joka voidaan määritellä tokeneista koostuviksi joukoksi, joka voidaan erottaa toisista lauseista välimerkeillä, kuten pisteellä, pilkulla, huutomerkillä tai kysymysmerkillä. Tämän tutkimuksen kontekstissa ei tehdä eroa lauseen ja virkkeen välillä, sillä teksti irrotetaan välimerkkien avulla pienempiin kokonaisuuksiin esikäsittelyssä ja kaikki välimerkit pistettä lukuun ottamatta poistetaan tekstistä. Tämä prosessi on kuvattu luvussa 2.3.1.

Luonnollisen kielen käsittelyssä dokumentilla on laajempi merkitys kuin arkikielessä. Dokumentti voidaan määritellä löyhästi tekstiksi, jota käsitellään itsenäisenä kokonaisuutena (Struhl 2015). Dokumentin pituus voi vaihdella aina muutaman lauseen kokonaisuudesta esimerkiksi kirjan kokoiseen rakenteeseen. (Struhl 2015). Tässä diplomityössä dokumentilla tarkoitetaan yksittäistä rahoitushakemusta ja siihen liittyvää hakemustekstiä. Dokumenttien muodostamia tekstikokoelmia kutsutaan NLP:n kontekstissa korpukseksi (Martinez 2010). Tässä tutkimuksessa korpus kattaa tutkimuskontekstin ja tutkimusongelman avulla rajatun joukon rahoitushakemusdokumentteja. Luonnollisen kielen rakenteiden tasot on laajimmasta pienimpään esitetty kuvassa 2.



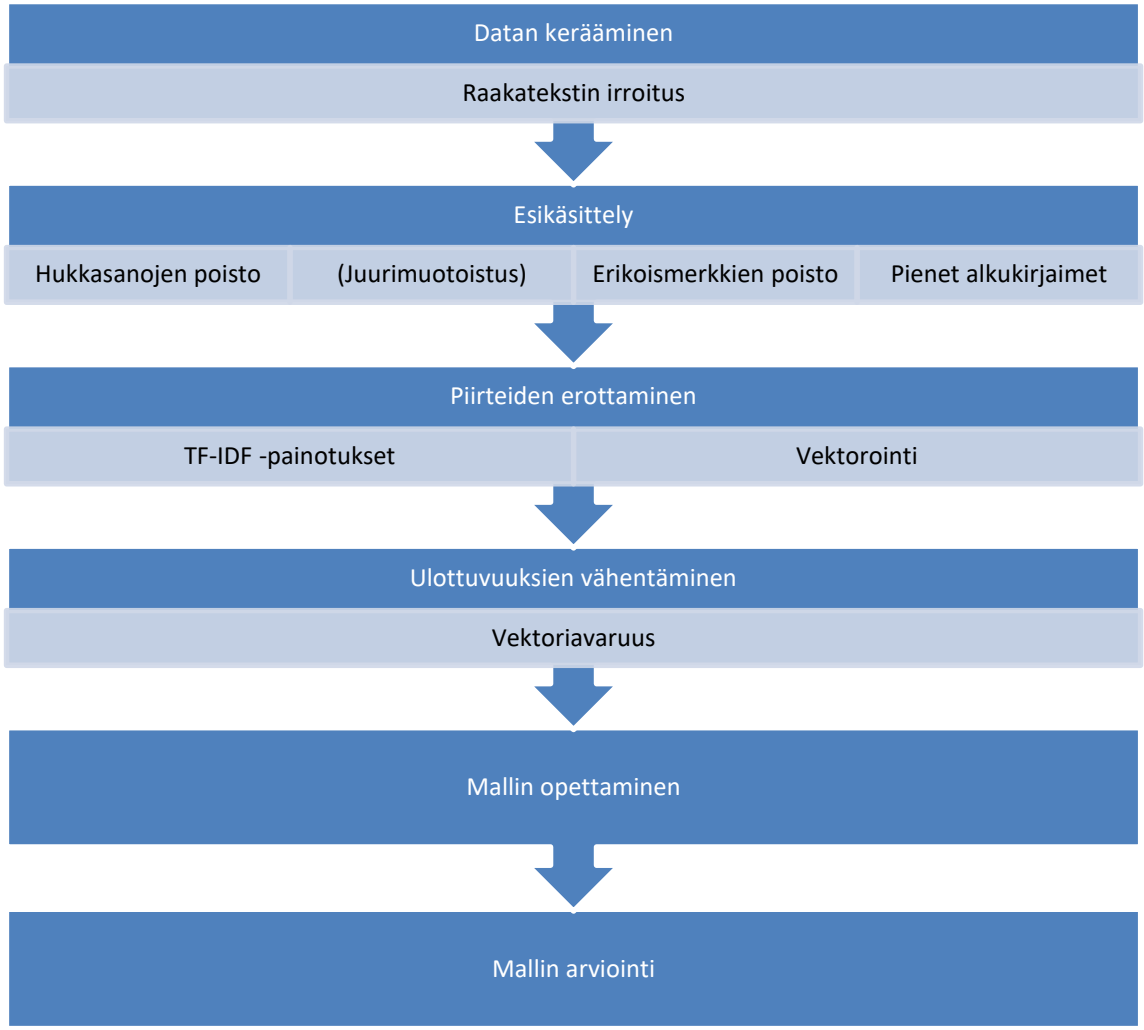
Kuva 2: Luonnollisen kielen rakenteiden tasot

Tässä diplomityössä muodostetaan ratkaisuvaihtoehtoja suomenkieliseen ongelmaan. Suomen kieli kuuluu fenno-ugrilaiseen kieliperheeseen ja ominaisuuksiltaan se sisältyy agglutinatiiviseen kieliluokkaan. Agglutinatiivisten kieliin kuuluu oleellisesti affiksien käyttö, jossa sanan merkitys muuttuu tai täsmentyy sanaan liitettyjen etu- tai loppuliitteiden käytön perusteella. (Martín et al. 2004) Tämä tarkoittaa NLP:n kannalta sitä, että yksittäinen suomenkielinen sana voi pitää sisällään yhtä paljon informaatiota kuin esimerkiksi kokonainen englanninkielinen lause, esimerkiksi: ”juoksentelisinkohan” – ”I wonder if I should run around (aimlessly)”.

Suomen kieli on siis hyvin rikas ja kompleksinen kieli. Muita suomen kielen erityispiirteitä ovat monimuotoinen yhdyssanojen käyttö, jossa sanan *juuri* (stem) voi kattaa tuhansia erilaisia yhdyssanakombinaatioita ja taivutuksia. (Martín et al. 2004) Esimerkki sanan ”autokaistoillakin” juureksi voidaan päätellä sana ”auto” (tai vaihtoehtoisesti ”kaista”). Voidaan päätellä, että yleispätevien sääntöjen muodostaminen suomenkielisen tekstin käsittelystä on haastavaa monipuolisen rakenteen vuoksi.

2.3 NLP ja koneoppimisprosessi

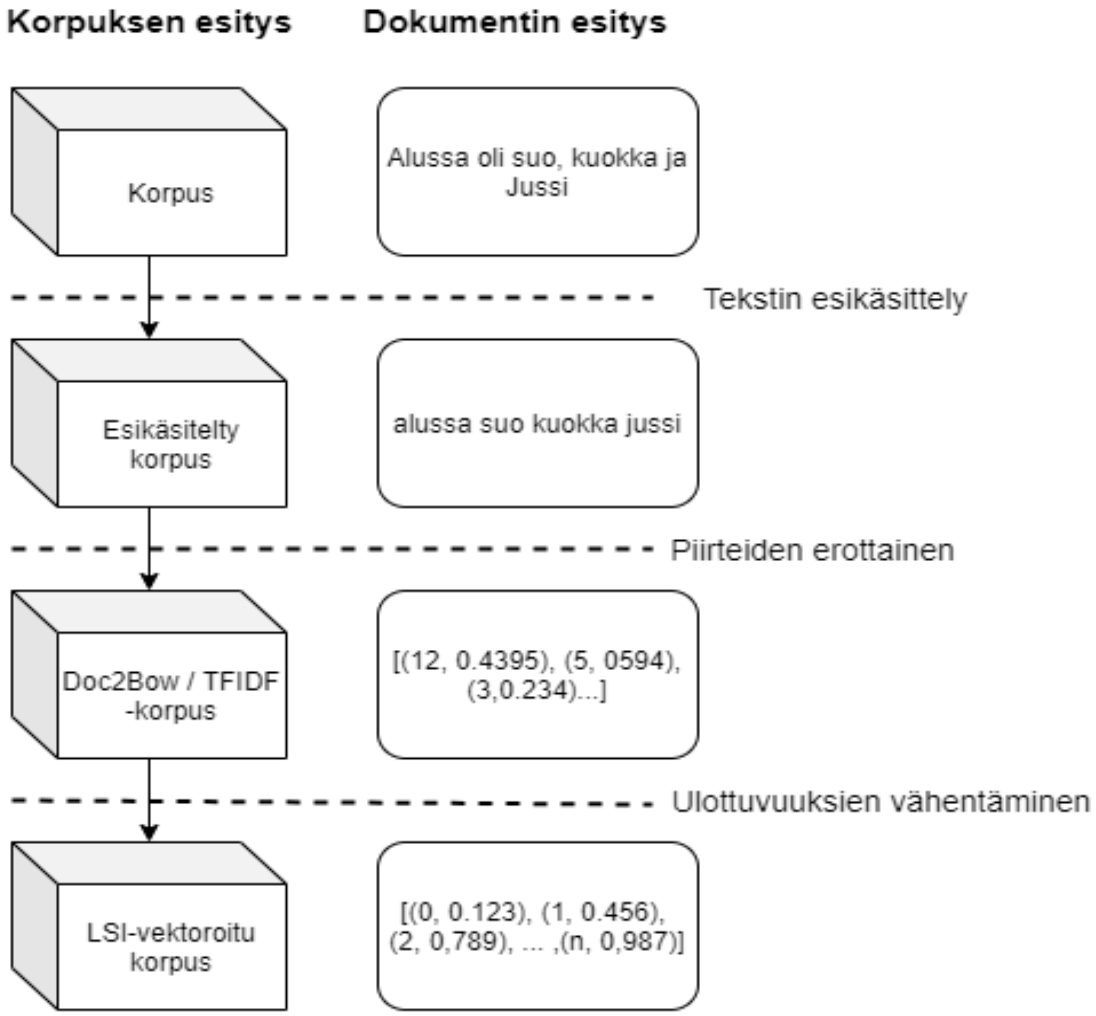
Tässä luvussa on kuvattu koneoppimisprosessin yleiset piirteet NLP:n näkökulmasta. Tutkimuksen empiirinen osuus on rakennettu tässä luvussa kuvatun luonnollisen kielen käsittelyn prosessin mukaisesti. NLP-koneoppimisprosessi on kuvattu yksinkertaistetusti kuvassa 3.



Kuva 3: NLP koneoppimisprosessi

Tekstidata on hyvin moniulotteista ja kompleksista dataa, jota on raskasta käsitellä koneellisesti. Datan *esikäsittely* (preprocessing) on edellytys sille, että tekstimuotoista dataa voidaan hyödyntää esimerkiksi koneoppimistarkoituksessa. Esikäsittelyssä pyritään yksinkertaistamaan ja muuntamaan dataa rakenteellisempaan muotoon, jossa sitä voidaan koneellisesti käsitellä ja analysoida (Hu & Liu 2012 ss.388-389). On siis löydettävä keino tuoda esiin tekstin rakennetta ohjaavat tekijät: syntaksi ja semantiikka.

Esikäsittely toteutetaan tutkimusaineiston sisältävälle tekstikorpukselle. Kuvassa 4 on esitetty korpuksen esikäsittelyvaiheet ja muoto, jossa yksittäisen dokumentin teksti on esitetty esikäsittelyvaiheen jälkeen. Tässä tutkimuksessa tehdään ero *korpuksen esikäsittelyn* ja *tekstin esikäsittelyn* välillä. Korpuksen esikäsittelyllä tarkoitetaan laajempaa kokonaisuutta, joka sisältää tekstin esikäsittelyn, piirteiden erottamisen ja ulottuvuuksien vähentämisen.



Kuva 4: Korpuksen esikäsittely (mukailtu Hu & Liu 2012; Mirończuk & Protasiewicz 2018)

Luonnollisen kielen esikäsittely ei toimi samalla tavalla jokaiselle kielelle. Esikäsittelyssä tuleekin ottaa huomioon käsiteltävän kielen erityispiirteet. (Aggarwal 2018 s.24) Kielikohtainen esikäsittely vaatisi jokaiselle hakemuskielelle oman esikäsittelynsä, minkä takia tämän tutkimuksen kontekstissa rajoitetaan aineisto suomenkielisin rahoitushakemuksiin.

2.3.1 Tekstin esikäsittely

Tekstin esikäsittelyssä poistetaan datasta epäolennaisia piirteitä, joiden merkitys datassa on vähäinen (Hu & Liu 2012 s. 389). Epäolennaisia piirteitä voi tekstin tyypin mukaan olla esimerkiksi html-elementit tai tarpeettomat erikoismerkit. *Raakatekstin irrottaminen* (text extraction) on esikäsittelyn ensimmäinen vaihe. Tässä vaiheessa poistetaan usein myös erikoismerkit ja isot alkukirjaimet. Seuraava askel tekstin

esikäsittelyssä on erittäin yleisten sanojen, *hukkasanojen* (stopword), poisto. Hukkasanoilla ei ole tekstin semanttisen merkityksen kannalta juurikaan arvoa. Näitä ovat esimerkiksi englanninkieliset artikkelit ”a” tai ”the”. (Aggarwal 2018 s. 5–6, 22) Suomen kielessä vastaavia hukkasanoja voivat olla esimerkiksi yleisimpiä pronominit ja partikkelit. Yleisen määritelmän mukaan hukkasanoja ovat informaatioköyhät, hyvin usein esiintyvät sanat. Yleisten hukkasanojen lisäksi voidaan kuitenkin määritellä toimialaspesifejä hukkasanoja, jotka eivät sisällä juurikaan lisäarvoa tutkittavan toimialan korpukselle. (Makrehchi & Kamel 2017) Esimerkiksi jalkapalloa käsittelevässä korpuksessa sana ”pallo” olisi mahdollista luokitella toimialakohtaiseksi hukkasanaksi, vaikka se olisi tärkeä sana esimerkiksi geometriaa tutkivassa korpuksessa.

Seuraava askel esikäsittelyprosessissa on sanojen *juurimuotoistaminen* (stemming) tai *perusmuotoistaminen* (lemmatization) (Hu & Liu 2012 s.389). Tämä tarkoittaa mm. etuliitteiden tai päätteiden poistamista sanasta. Esimerkiksi sanan ”koulussa” juurimuoto on ”koulu”. Juurimuotoistamisen taustalla on tarve saada sanan eri muodot, kuten monikot muunnettua samaan muotoon. Perusmuotoistaminen on juurimuotoistuksen kehittyneempi muoto, jossa päätteen poistamisen sijaan sanalle haetaan kieliopillinen perusmuoto (Aggarwal & Zhai 2012; Aggarwal 2018 ss.23-24). Juurimuotoistaminen ei sovellu hyvin esikäsittelymenetelmänä suomenkieliseen aineistoon (Korenius et al. 2012). Kuten luvussa 2.2. todetaan, suomi perustuu yhdyssanoihin ja sanojen päätteet ja etuliitteet sisältävät paljon informaatiota. Juurimuotoistaminen siis voi muuntaa sanojen ja lauseiden semanttista merkitystä, sekä vähentää niiden sisältämää informaatiota.

2.3.2 Piirteiden erottaminen ja ulottuvuuksien vähentäminen

Koska luonnollinen kieli on luonteeltaan hyvin kompleksista ja moniulotteista, sen käsittely edellyttää kielen muuntamista laskettavaan muotoon (Wajeed & Adilakshmi 2011). Laskettavaan muotoon muuntaminen koostuu esikäsittelyn lisäksi piirteiden erottamisesta ja ulottuvuuksien vähentämisestä (Mirończuk & Protasiewicz 2018). Jung (2018) määrittelee ulottuvuuksien vähentämisen prosessina, jossa data voidaan esittää tiivistetyssä muodossa ja sen jälkeen rekonstruoida data takaisin lähelle alkuperäistä muotoaan. Piirteiden erottaminen ja ulottuvuuksien vähentäminen ja voidaan toteuttaa esimerkiksi *vektoroimalla* tekstikokonaisuuksia, kuten esimerkiksi dokumentteja (Novotný & Ircing 2017; Mirończuk & Protasiewicz 2018). Vektorien muodostamiseen tarvitaan tietoa dokumenteissa esiintyvistä sanoista ja niiden esiintymistiheydestä.

Piirteiden erottaminen voidaan aloittaa laskemalla yksittäisten sanojen esiintyvyyttä korpuksen dokumenteissa *termifrekvenssin* (term frequency, TF), sekä laskemalla dokumenttien määrä, jossa kukin sana esiintyy. Tätä kuvataan nimellä *käänteinen dokumenttifrekvenssi* (inverse document frequency, IDF). Sekä termifrekvenssissä että käänteisessä dokumenttifrekvenssissä esitetään sanat *bag of words-muodossa* (BOW). BOW-muoto tarkoittaa sanojen esittämistä muodossa, jossa tallennetaan sana ja sanan esiintymisfrekvenssi muodossa: {sana: frekvenssi} (Müller 2016). BOW-muodon kannalta sanojen järjestyksellä ei ole merkitystä, vaan esimerkiksi lauseet ”Alussa oli suo kuokka ja jussi” ja ”Suo ja kuokka oli alussa Jussi” muodostavat saman BOW-muodon. (Aggarwal 2018 ss.305–306) Hukkasanojen poiston jälkeen esimerkkilauseiden BOW-muoto voidaan esittää seuraavassa muodossa:

{’alussa’:1, ’suo’:1, ’kuokka’:1, ’jussi’:1}

Termifrekvenssiä ja käänteistä dokumenttifrekvenssiä hyödynnetään TF-IDF-vektorien muodostamisessa. TF-IDF-vektorien muodostamisella toteutetaan piirteiden erottaminen (Underhill et al. 2007; Mirończuk & Protasiewicz 2018). Piirteiden erottamisella muodostetaan luvussa 2.3. kuvattu koneoppimisen opetusdata jalostamalla ja muokkaamalla tekstimuotoinen raakadata numeraaliseksi, koneluettavaksi dataksi.

TF-IDF-vektorointia käytetään tunnistamaan tekstin kannalta merkityksellisimmät sanat (Müller 2016). Jotta korkean frekvenssin sanat eivät vähentäisi vähän esiintyvien sanojen merkitystä liikaa, IDF-laskelmaan hyödynnetään ns. vaiennusfunktiota (damping function), joka voi tekniikasta riippuen olla joko juurifunktio tai logaritmi. Vaiennusfunktion tarkoitus on tasapainottaa sanojen painoarvoja ja tehdä niistä vertailukelpoisempia. (Aggarwal 2018 ss. 5–7, 21–25) TF-IDF-arvo kullekin sanalle saadaan esimerkiksi seuraavalla kaavalla:

$$TFIDF_{i,j} = \frac{t/T}{\lg(D/d)} \quad (1)$$

(Underhill et al.2007)

jossa:

- t = sanan *j* frekvenssi dokumentissa *i*
- T = dokumentissa esiintyvien sanojen määrä
- lg = luonnollinen logaritmi
- D = korpuksen dokumenttimäärä

- d = sanan *j* sisältävien dokumenttien frekvenssi

Vektoriesitys mahdollistaa dokumenttien vertailun vektorilaskennan menetelmin. Esimerkkinä tästä on sanojen tai dokumenttien samankaltaisuuden arviointi niiden perusteella muodostettujen vektorien välisen kulman kosinin avulla. Samankaltaisuuden arviointiin esitetty laskennallinen metriikka on siis kulman kosiniin perustuva kosinisimilariteetti (cosine similarity). (Aggarwal 2018 s.27) Kosinisimilariteetti vektoreille v_1 ja v_2 lasketaan seuraavasti:

$$\cos\theta = \frac{v_1 \cdot v_2}{\|v_1\| \cdot \|v_2\|}, \quad (2)$$

(Deng et al.2019)

Koska kosini saa arvoja välillä {-1, 1}, on myös vektorien samankaltaisuus esitetty vastaavalla asteikolla. Kahden tismalleen saman vektorin välinen kulma on 0°, jolloin kosini cos = 1. Kun vektorit ovat 90° kulmassa, kosinin arvo cos = 0, ja kun vektorien välinen kulma on 180°, kosinin arvo on -1. (Kalhori et al. 2018) Tästä seuraa, että kaksi sanaa ovat vähiten saman kaltaisia, kun niille määritettyjen vektorien välinen kulma on 180°.

Vaikka vektoroitu tekstidata mahdollistaa laskutoimitukset ja esimerkiksi dokumenttien samankaltaisuuden arvioinnin, on se sellaisenaan puutteellinen tekstien semanttisen samankaltaisuuden arviointiin. TF-IDF-matriisin muodostama vektoriavaruus on hyvin laaja ja riippuvainen samojen sanojen käytöstä, jotta dokumentit tunnistetaan samankaltaisiksi. (Zelikovitz & Marquez 2005; Novotný & Ircing 2017) Tämän voidaan nähdä muodostavan korostetun ongelman etenkin suomenkielisten tekstien käsittelyssä, sillä pelkkä sanan etuliite tai pääte muodostaa matemaattiseen malliin uniikin sanan, jolloin lauseen semantiikka TF-IDF-matriisissa on riippuvainen syntaksista.

Ulottuvuuksien vähentämiseen voidaan hyödyntää tekniikkaa, jolla voidaan tiivistää TF-IDF-matriisi matriisilaskutoimituksin tiivistetyksi vektoriavaruudeksi, joka säilyttää tekstin semanttisen informaation (Mitra et al. 2007; Novotný & Ircing 2017). Tämä prosessi on kuvattu tarkemmin luvussa 3.2.2. Kuten TF-IDF-vektoroinnin toteuttamassa piirteiden erotuksessa, myös vektoriavaruuden kompressoinnissa on olennaista esittää datan vähäulotteisemmassa formaatissa kuitenkaan menettämättä datan alkuperäisiä tärkeitä piirteitä ja etenkin tekstin semantiikkaa (Underhill et al. 2007). Muita saavutettuja etuja ulottuvuuksien vähentämisestä on mallin ylisovittamisen todennäköisyyden laskeminen,

sillä ulottuvuuksien vähentyessä myös mallin kompleksisuus vähenee (Jung 2018 s.107). Ylisovittamisen käsite on kuvattu luvussa 2.3.3.

On kuitenkin huomattava, että luvun aloituskappaleessa esitetty Jungin (2018 s.106) kuvaamaa käänteinen toimenpide ulottuvuuksien vähentämisen jälkeen (rekonstruktio) ei päde TF-IDF-vektoroidulle tekstidatalle. Toisin sanoen, tekstidataa ei pystytä enää vain TF-IDF vektoreiden perusteella muuntamaan alkuperäiseen muotoonsa. Tämä johtuu siitä, että TF-IDF-arvojen muodostamisessa sanojen esiintymisfrekvenssiä lasketaan BOW-muodossa (Altınel et al. 2015). BOW-lähestymistapa vähentää vektorien sisältämää informaatiota, mutta Aggarwal (2018 ss. 305–306) arvioi sanojen BOW-muodon olevan riittävällä tarkkuustasolla esimerkiksi binääriluokitteluongelmassa käytettäväksi.

2.3.3 Koneoppiminen

Koneoppiminen voidaan määritellä tekoälyn tutkimusalueeksi, jolla voidaan koneellisesti simuloida oppimista jatkuvalla iteratiivisella prosessilla (Haoyong Lv & Hengyao Tang 2011). Koneoppimiseen liittyy myös tekoälyn periaate, jossa tarkoituksena on opitun perusteella muodostaa (koneellisesti) ongelmaan laskennalliset optimiratkaisut, jotka maksimoivat pitkän aikavälin hyödyn (Jung 2018). Toisaalta koneoppiminen voidaan määritellä tilastotieteen, tekoälytutkimuksen ja tietotekniikan yhdistelmänä, jonka avulla voidaan algoritmia datan avulla opettamalla muodostaa dataan perustuvia ennusteita, joita ei opetusdatassa esiinny (Müller 2016). Koneoppiminen on siis dataan perustuvaa iteratiivinen prosessi, jossa pyritään opetetun algoritmin avulla muodostamaan ennusteita tai havaintoja, jotka perustuvat opetusdataan.

Koneoppiminen voidaan jakaa tyypillisesti kolmeen luokkaan: *ohjattu oppiminen* (supervised learning), *ohjaamaton oppiminen* (unsupervised learning) ja *puoliohjattu oppiminen* (semi-supervised learning). Ohjatussa oppimisessa pyritään opettamaan koneoppimismallia itsenäisten ominaisuuksien eli *piirteiden* (feature) ja niitä selittävien *luokkien* (class, label) avulla. (Wajeed & Adilakshmi 2011) Luokalla tarkoitetaan kuvausta, joka kuvailee sitä vastaavaa datapistettä. (Jung 2018 s. 4). Esimerkiksi tässä tutkimuksessa piirteitä ovat esikäsittelyt rahoitushakemustekstit ja niitä kuvaavat luokat ovat joko ’cleantech’ tai ’ei cleantech’.

Opetuksen jälkeen koneoppimismallin avulla voidaan luoda ennusteita datalle, jota malli ei ole käsitellyt (Wajeed & Adilakshmi 2011). Opetusdatan piirteet voidaan siis nähdä ohjeina siitä, miten mallin tulee käyttäytyä koulutuksen jälkeen (Aggarwal 2018 s. 11). Tämän tutkimuksen empiirisessä osuudessa ohjatun oppimisen piirteinä käytetään

rahoitushakemuksen esikäsiteltyä tekstiä ja luokkana sitä kuvaavaa luokitusta, eli ’cleantech’.

Ohjaamattomassa oppimisessa puolestaan algoritmille ei anneta luokitusta tai kuvausta ”ohjeistukseksi” siitä, miten mallin tulee toimia, vaan algoritmi pyrkii esimerkiksi etsimään datasta yhdenmukaisuuksia ja ryhmittelemään havaintoja. Tämä kuvaa *klusteroivaa* ohjaamatonta oppimista. (Jung 2018) Tekstianalyysin kontekstissa tärkeä esimerkki ohjaamattomasta oppimisesta ja klusteroinnista on *aihemallinnus* (topic modeling). Aihemallinnuksessa generoidaan tekstikorpuksesta siinä esiintyvien teemojen perusteella *aiheklustereita* (topic cluster) ja päätellään laskennallisen todennäköisyyden perusteella, mihin aiheeseen kukin korpuksen dokumentit kuuluvat. (Aggarwal & Zhai 2012) Aihemallinnusta käytetään tämän tutkimuksen empiriaosuudessa dokumenttien semanttisten vektoriesitysten luomisessa, joka on esitetty luvussa 3.2.2. Klusterointi ei ole ainoa ohjaamattoman oppimisen muoto, mutta muita menetelmiä ei käsitellä tässä tutkimuksessa.

Puoliohjattu oppiminen yhdistää ohjatun ja ohjaamattoman oppimisen piirteitä, jolloin opetukseen käytetään sekä luokiteltua että luokittelematonta dataa. (Mirończuk & Protasiewicz 2018) Puoliohjattu oppiminen on potentiaalinen vaihtoehto, mikäli luokitellun datan osuus kokonaisdatasta on rajallinen (Aggarwal 2018 s. 131). Puoliohjatussa oppimisessa oletetaan, että toisiaan lähellä olevat datapisteet (vektorit) ovat myös ominaisuuksiltaan samankaltaisia. Näin ollen yhdistämällä luokiteltua dataa ohjaamattoman oppimisen malliin (esim. klusterointimalliin), voidaan luotuja klustereita käyttää hyödyksi luokittelemattoman datan luokittelussa. (Jung 2018 s.94) Esimerkiksi NLP-kontekstissa ohjaamattoman aihemallinnukseen teemoihin yhdistetty luokiteltu data-aineisto voi auttaa yhdistämään koneellisesti luotuja teemakokonaisuuksia ennalta määrättyihin luokkiin.

Koneoppimisen prosessiin kuuluu olennaisesti oppimisen validointi. Mallin kykyä tehdä luotettavia ennusteita opetukseen kuulumattomalle datalle kutsutaan mallin *yleistettävyydeksi* (generalization). (Jung 2018 s.80) Yleistettävyydessä on tärkeää löytää malli, joka ei noudata opetusdatan piirteitä liian yksityiskohtaisesti, mutta kuitenkin siten, että malli tunnistaa datasta hyödyllisiä rakenteita. Mikäli malli noudattaa opetusdataa liian tarkasti, se muodostaa erittäin kompleksisen mallin. Tätä kutsutaan *ylisovittamiseksi* (overfitting). Ylisovittamisen vastakohta on liian yksityiskohtainen malli, joka *alisovittaa* (underfitting) datapisteitä. Alisovitteinen malli on yleistettävissä mutta ei kuitenkaan opi datasta hyödyllisiä piirteitä. (Müller 2016) Koneoppimismalli on siis yleistettävissä, kun se ei ole ylisovitteinen tai alisovitteinen, eli silloin, kun se oppii datan rakenteita kuitenkaan kopioimatta niitä.

2.3.4 Tekstin luokittelu

Luokitteluongelma on koneoppimisen osa-alue, jossa muodostetaan olemassa olevan datan perusteella *luokittelumalli* (classifier). Luokittelumallin avulla pyritään ennustaman, mihin ennalta määriteltyyn luokkaan havainnot kuuluvat. (Lessmann & Voß 2008 ss. 231–232) Tekstin automaattisella luokittelulla tarkoitetaan esimerkiksi koneoppimiseen perustuvaa tekstien jaottelua jo olemassa oleviin luokkiin, esimerkiksi teemakokonaisuuksiin. (Weng et al. 2017; Mirończuk & Protasiewicz 2018). Luokat sekä niiden määrät ja ominaisuudet voidaan määritellä tutkimuskohteen mukaisesti. Tyypillisiä luokitteluperusteita, joilla dokumentteja voidaan jaotella ovat esimerkiksi määrätyt aiheet (urheilu, musiikki, luonto) ja tekstin tunnesävy (sentimenttianalyysi). (Wu et al. 2018) Voidaankin päätellä, että luokiteltavan kohteen määrittämisessä on oleellista, että data vastaa tutkimuskohdetta, jotta luokittelua voidaan aiheeseen soveltaa.

Luokitteluongelma koostuu kahdesta osasta: mallin opettamisesta opetusaineistolla ja mallin testaaminen testiaineistolla (Aggarwal 2015). Prosessi aloitetaan jakamalla luokitteluun käytettävä datajoukko *opetusdataan* (training data) ja *testidataan* (test data). Luokittelu tapahtuu opettamalla valittua luokittelualgoritmia datalla, jolle on osoitettu dataa kuvaava luokka. Opetusprosessi tuottaa koneoppimismallin, jota voidaan arvioida testidatalla, jota ei ole käytetty mallin opetukseen. Koneoppimismalli pyrkii ennustamaan testidatan alkioille niitä kuvaavan luokan opetusdatan perusteella. (Aggarwal & Zhai 2012) Tekstin (ohjatussa) luokittelussa opetusdatana käytetään usein manuaalisesti ihmisen työn tuloksena luokiteltua tekstidataa (Xu et al. 2014). Koska testidatan oikea luokka tiedetään, voidaan koneen ennusteen oikeellisuutta arvioida.

Luokittelijat voidaan toimintaperiaatteensa perusteella jakaa *moniluokkaluokitteluun* (multi-class classificaiton) ja *binääriluokitteluun* (binary classification). Binääriluokittelija luokittelee tutkimusaineistoa kahteen ennalta määriteltyyn luokkaan ja näin ollen soveltuu vastaamaan kyllä/ei-tyyppisiin kysymyksiin. Moniluokkaluokittelija vastaa kysymykseen, mihin ennalta määrättyyn luokkaan havainto kuuluu. (Müller 2016) Tämän tutkimuksen empiriaosuudessa tutkitaan binääriluokittelijaa, joka arvioi, kuuluuko rahoitushakemus luokkaan ’cleantech’, vai ’ei cleantech’. Binääriluokittelijan toimintaa arvioitaessa luokittelijan tulos on joko positiivinen tai negatiivinen ja positiivinen luokka määräytyy tutkimuskohteen perusteella (Aggarwal 2018 s.227). Tämän tutkimuksen kontekstissa tämä tarkoittaa sitä, että mikäli luokittelija tunnistaa oikein luokan ’cleantech’, kyseinen luokitus saa positiivisen tuloksen.

Luokittelu on tärkeä keino rakenteettoman tekstidatan organisointiin (Altınel & Ganiz 2018). Luokittelulla voidaan vastata tekstidatan räjähdysmäiseen kasvuun

mahdollistamalla dokumenttien indeksointi, hakutoiminnot, suodatus (mm. roskaposti) ja lopulta tekstianalyysi (Mitra et al. 2007; Deng et al. 2019) Luokittelu tekstin osalta on siis tehokas menetelmä muuttamaan rakenteetonta dataa helpommin hyödynnettävään muotoon.

Tekstin luokitteluun on useita keinoja, joista tämän tutkimuksen kannalta olennaisin on semanttisiin tekijöihin, eli tekstin merkitykseen perustuva luokittelu. Semanttiset luokittelualgoritmit voidaan jakaa toimialatietämyspohjaisiin (domain knowledge-based), korpuspohjaisiin, syväoppimispohjaisiin, sana/kirjain-yhdistelmä-pohjaisiin ja lingvistiikan avulla rikastettuihin menetelmiin (Altınel & Ganiz 2018). toimialatietämyspohjaiset, eli sanastopohjaiset menetelmät käyttävät sanojen semanttisen merkityksen taltioimiseen ulkoisen järjestelmän sisältämää sanastoa, johon on kerätty tieto sanojen semanttisesta merkityksestä ja esimerkiksi niiden synonyymeistä (Aggarwal 2018). Toimialatietämyspohjaiset järjestelmät ovat kieliriippuvaisia ja Altinel & Ganiz (2018) listaavat tärkeimmiksi sanastojärjestelmiksi mm. WordNetin, Wiktionaryn ja Wikipedian. Toimialatietämykseen perustuvat mallit vaativat siis toimiakseen valtavat määrät taustadataa ja myös määrämuotoisen kielirakenteen, jota suomi ei sijapäätteineen edusta.

Korpuspohjaiset luokittelumenetelmät sen sijaan ovat kieliriippumattomia, eivätkä ne ole myöskään riippuvaisia ulkoisesta tietojärjestelmästä. Näissä menetelmissä tieto semanttisesta rakenteesta luodaan käytössä olevasta datasta. Tämä varmistaa, että luokittelussa käytetystä sanastosta ei puutu esimerkiksi harvinaisia, liiketoimintaspesifejä aihesanoja, joita ei välttämättä ole tallennettu ulkoiseen järjestelmään. (Altınel & Ganiz 2018) Luvussa 4.3. esitetty koneoppimisprosessi kuvaa nimenomaan korpuspohjaisen NLP-prosessin työnkulkua. Korpuspohjaisessa luokittelussa on tärkeää löytää semanttiset yhteydet sanojen ja dokumenttien välille syntaktisen samankaltaisuuden sijaan. Tämä mahdollistuu hyödyntämällä kompressoituja vektoriavaruuksia, esimerkiksi Latent Semantic Indexing -menetelmällä. (Altınel & Ganiz 2018) Korpuspohjaiset menetelmät soveltuvat siis tilanteisiin, jossa on käytössä riittävä pohja-aineisto, josta voidaan muotoilla riittävän kattava korpus. Tässä diplomityössä hyödynnetään pääasiallisesti korpuspohjaisia menetelmiä semanttisessa luokittelussa.

Syväoppimismenetelmissä muodostetaan monikerroksinen neuroverkko, jossa manuaalisen piirteiden erottamisen sijaan piirteet tunnistetaan neuroverkon alkukerroksissa. Neuroverkon kerroksissa tehdään datalle transformaatio, jonka jälkeen transformoitu esitys syötetään hierarkisesti seuraavalle neuroverkon kerrokselle, jolloin semanttinen informaatio välittyy. Neuroverkkojen käyttäminen esikäsittelyssä perustuu

esimerkiksi *jatkuvan bag of words* (continuous bag of words, CBOW) -menetelmän hyödyntämiseen, jossa sanan piirteet saadaan erotettua vektoriksi arvioimalla kutakin sanaa ympäröivään kontekstinsa eli ympäröivien sanojen perusteella. (Altınel & Ganiz 2018) Neuroverkolla voidaan siis toteuttaa semanttisen informaation säilyttäminen piirteiden erottamisessa.

3. NLP-MALLIVALINNAT JA VERTAILUMENETELMÄT

Tässä luvussa esitellään diplomityön empiiriseen osaan valittujen luokittelumallien ja vertailumetriikoiden menetelmät ja periaatteet. Tämän lisäksi luvussa käsitellään tutkimusaineiston ominaisuuksien vaikutusta luokittelumalleihin. Empiirisessä osiossa muodostetaan binääriluokittelija, eli kahteen eri luokkaan aineistoa jakava luokittelumalli. Tässä tutkimuksessa tutkittava luokka on ’cleantech’ ja muu aineisto kuuluu luokkaan ’Ei cleantech’.

Tutkimusasetelman mukaisesti käytössä on myös *kontrollimalleja* (benchmarking), joiden avulla arvioidaan mallien toimivuutta. Tutkimuksessa käytettävät kontrollimallit ovat *säännöllisiin lausekkeisiin* (regular expression) perustuva malli, tilastollinen todennäköisyys ja kolikonheiton todennäköisyys.

3.1 Aineiston epätasapaino

Luokiteltavassa data-aineistoissa esiintyy usein epätasaisuutta luokiteltavien alkioiden luokkakohtaisissa määrissä (Johnson & Khoshgoftaar 2019). Tätä kutsutaan *luokkaepätasapainoksi* (class imbalance) tai pahimmillaan *luokkaharvinaisuudeksi* (class rarity). (Hasanin et al. 2020) Esimerkiksi syöpiä tutkivan luokittelijan kohdalla on usein kyse tilanteesta, jossa negatiivisia tuloksia on huomattavasti enemmän kuin positiivisia, jolloin positiivisessa luokassa vallitsee luokkaepätasapaino.

Aineiston luokkaepätasapainosta seuraa usein opetetun luokittelijan taipumus suosia luokkaa, jonka määrää esiintyy aineistossa eniten, mikä johtaa virheellisesti luokiteltuihin alkioihin vähemmistöluokassa (Ma et al. 2018). Äärimmäisessä tilanteessa tämä voi johtaa siihen, että malli ennustaa joka kerralla yleisempää luokkaa (Johnson & Khoshgoftaar 2019). Tästä voidaan päätellä, että epätasapainoisella aineistolla opeteltu luokittelija voi ylikorostaa yleisemmän luokan ennustamista. Yleisempää luokkaa ylikorostava luokittelija ei toimi tällöin halutulla tavalla, jolloin tilannetta on syytä välttää ehkäisemällä aineiston epätasapainoa.

Aineiston epätasapainoon voidaan vaikuttaa joko vaikuttamalla luokittelualgoritmiin tai aineiston otantaan (Ma et al. 2018; Johnson & Khoshgoftaar 2019). Luokittelualgoritmin toimintaan vaikuttavat, algoritmipohjaiset menetelmät perustuvat epätasapainon kompensoimiseen luokittelualgoritmin ominaisuuksien avulla. Otantamenetelmät eli

datapohjaiset menetelmät perustuvat aineiston muokkaamiseen epätasapainon vähentämiseksi. (Castellanos et al. 2018)

Yliotannassa (oversampling) kasvatetaan vähemmistöluokan alkioiden määrää esimerkiksi luomalla keinotekoisia datapisteitä (Castellanos et al. 2018). Tähän perustuu tässä tutkimuksessa käytetty *SMOTE* (synthetic minority over-sampling technique) -menetelmä, joka luo uusia alkioita opetusdataan olemassa olevan vähemmistöluokan datapisteiden perusteella. (Johnson & Khoshgoftaar 2019) Yliotannassa siis luodaan opetusdataan perustuvia keinotekoisia datapisteitä, jotka simuloivat opetusdatan alkioita

Aineistoa voidaan muokata laskemalla enemmistöluokkaisen aineiston määrää *aliotannan* (undersampling) avulla. Aliotanta toteutetaan poistamalla enemmistöluokan alkioita. On kuitenkin huomattava, että aliotantaa käytettäessä vähennetään myös opetusdatan määrää. (Castellanos et al. 2018) Tästä syystä tässä tutkimuksessa vaikutetaan epätasapainoon vain yliotannan menetelmin.

3.2 NLP-tekniikat

Tässä aliluvussa on kuvattu valittujen NLP-tekniikoiden tausta ja toimintaperiaate. Tutkittaviksi malleiksi valikoitui kaksi koneoppimismallia: ohjattuun oppimiseen perustuva luokittelumalli fastText ja puoliohjattuun oppimiseen perustuva luokittelumalli LSI-kNN. Näiden lisäksi tutkitaan myös säännöllisiin lausekkeisiin perustuvan luokittelumallin toimintaa.

3.2.1 fastText

Ohjatun oppimisen luokittelumallina toteutetaan tässä tutkimuksessa fastText-kirjaston avulla. FastText on Facebook AI Researchin lanseeraama avoimen lähdekoodin kirjasto, jonka tarkoituksena on tarjota skaalautuvia ratkaisuja tekstin vektoriesityksiin ja luokitteluun (Bojanowski et al. 2016). FastTextin käyttöä puoltaa mallin kouluttamisen nopeus, kieliriippumattomuus ja sanansisäisen informaation hyödyntäminen vektoriesitysten muodostamisessa (Bojanowski et al. 2016).

FastText-luokittelumalli perustuu sekä syväoppimispohjaiseen että korpuspohjaiseen tekstiluokittelun toimintaperiaatteeseen, jotka ovat esitetty luvussa 2.3.4. FastText käyttää piirteiden erottamiseen ja ulottuvuuksien vähentämiseen CBOW-menetelmää ja *ngrameja*, Ngramit ovat tapa esittää merkkijonoja (esimerkiksi sanoja) koodatussa muodossa. (Joulin et al. 2016a). FastTextin CBOW-menetelmä tarkoittaa

neuroverkkoihin pohjautuva mallia, joka koostuu *syötteestä* (input layer), *piilokerroksesta* (hidden layer) ja *tuotteesta* (output layer). (Joulin et al. 2016b) Syötteenä CBOW-menetelmä käyttää kutakin aineiston sanaa ympäröivää kontekstia halutulla etäisyydellä sanasta. Etäisyydellä tarkoitetaan käyttäjän asettamaa sanamäärää, jonka kone lukee ennen sanaa ja sanan jälkeen Sanamäärän perusteella muodostunut joukko on sanan konteksti. Kontekstin sanojen vektoriesitysten pohjalta CBOW-malli ennustaa kyseessä olevan sanan. (Luo et al. 2014)

FastTextin käytöstä on hyötyä etenkin muotorikkaissa rikkaissa kielissä, kuten suomen kielessä, sillä se ottaa huomioon sanan kontekstin lisäksi myös sanan sisäisen rakenteen (Bojanowski et al. 2017). Tämä tarkoittaa, että fastText-kirjasto pilkkoo sanan osiin ja tutkii asetetun ngram-parametrin mukaisesti sanan rakennetta. Esimerkiksi sana ”suomi” käsitellään seuraavalla tavalla, kun ngram-parametrin arvo n = 3 (*trigram*):

<su, suo, uom, omi, mi>

Oleellista on myös se, että sanan osana oleva trigram ”suo” eroaa kontekstinsa vuoksi irrallisesta sanasta ”suo”. (Bojanowski et al. 2017) Sanavektorien muodostaminen sanansisäisestä informaatiosta sopii siis tutkimuskontekstiin, sillä suomenkieliset sanat sijapäätteineen sisältävät merkittävän määrän informaatiota.

3.2.2 Latent Semantic Indexing

Luvussa 2.3.2 esitellään esikäsittelyn ja ulottuvuuksien vähentämisen yhteydessä kompressoitu vektoriavaruus, jossa tekstipohjaisia semanttisia vektoreita voidaan verrata. Tähän käyttötarkoitukseen hyödynnetään tässä diplomityössä *Latent Semantic Indexing* (LSI) -pohjaista menetelmää. LSI on jo alan standardiksi muodostunut menetelmä dokumentin semanttisen merkityksen löytämiseen ja esittämiseen (Řehůřek 2011).

Tekstien semanttisten merkitysten samankaltaisuutta on vaikea arvioida vain yhteneväisten sanojen kautta, sillä syntaksi ja ihmisten sattumanvaraiset ilmaisutavat piilottavat tekstin semanttisen rakenteen (Deerwester et al. 1990). Latent Semantic Indexing perustuu olettamukseen, että tekstidatalla on taustallaan semanttinen rakenne, joka voidaan esittää vektorimuodossa (Zelikovitz & Marquez 2005). Voidaan siis päätellä, että dokumenttien semanttista sisältöä (dokumentin merkitystä) voidaan LSI-menetelmän avulla verrata toisiinsa matemaattisin keinoin.

LSI-mallin rakentamisessa hyödynnetään *termi-dokumenttimatriisia* (term-document matrix), jossa on esitetty kaikki korpuksen dokumentit ja dokumentit BOW-muodossa.

Matriisin arvoille annetaan painotus, esimerkiksi TF-IDF-vektoroinnin avulla. (Bradford 2008) Tässä tilanteessa doc2bow -termi-dokumenttimatriisi muuttuu TF-IDF termi-dokumenttimatriisiksi.

Kuten luvussa 2.3.2. todetaan, TF-IDF-vektorit ovat kompleksisia ja moniulotteisia. LSI-menetelmä käyttää hyväkseen Singular Value Decomposition (SVD) -menetelmää moniulotteisten termi-dokumenttimatriisien tiivistämiseen (Deerwester et al. 1990). SVD laskee matriisitulon termi-dokumenttimatriisin eri komponenteista. Matriisitulo tuottaa sellaisen approksimaation termi-dokumenttimatriisista, jossa semanttinen informaatio on irrotettu syntaktisesta muodosta. (Zelikovitz & Marquez 2005). Termi-dokumenttimatriisi siis muutetaan matriisitulojen avulla muotoon, joka esittää sanojen frekvenssien sijaan dokumentin semanttisen informaation LSI-vektorina.

Tuloksena saatuna matriisia voidaan hyödyntää tekstien luokittelussa kääntämällä sen pohjalta muita TF-IDF-mallinnettuja tekstijoukkoja samaan vertailukelpoiseen LSI-avaruuteen.

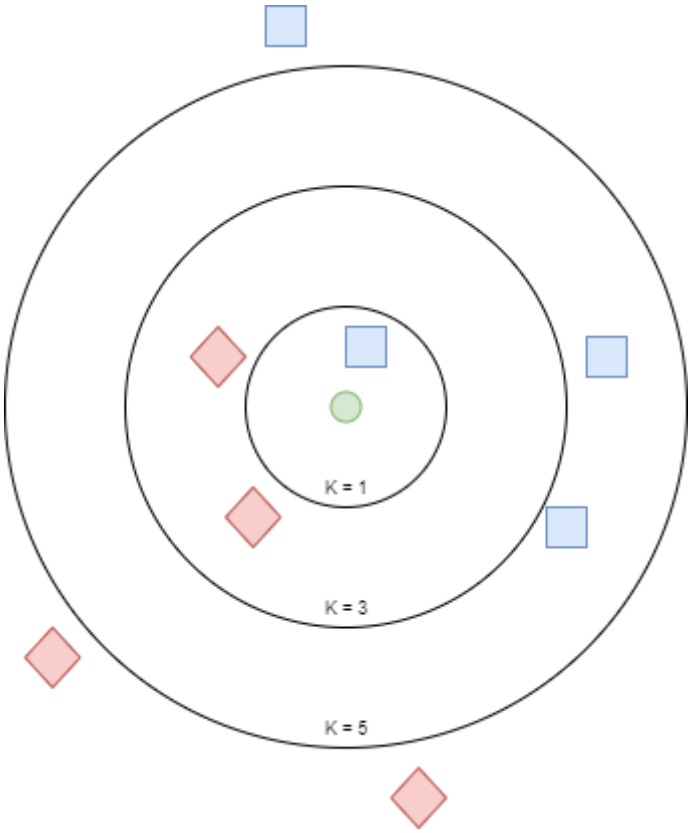
3.2.3 K-Nearest Neighbor -algoritmi

Luokittelussa käytettäväksi algoritmiksi valittiin k-Nearest Neighbor (kNN) -luokittelualgoritmi. K-Nearest Neighbor algoritmin toiminta perustuu testiaineiston alkioiden ja opetetun algoritmin sisältämien alkioiden välisen etäisyyden mittaamiseen. KNN algoritmi luokittelee alkion kuuluvan tiettyyn luokkaan lähimmän (tai lähimpien) alkioiden perusteella. (Aggarwal 2015). Testialkion ja opetetun mallin alkioiden välistä etäisyyttä *d* mitataan pisteiden *x* ja *y* välisenä etäisyytenä euklidisen etäisyyden avulla, joka on esitetty kaavassa 3:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}, \quad (3)$$

(Peterson 2009)

Joukon suuruutta kuvataan kokonaisluvulla *k*, jonka avulla määritetään, kuinka moneen lähimpään alkioon testialkiota verrataan. K:n arvo määritetään luokittelumallin rakennusvaiheessa. Mikäli k-määrä alkioita sisältää eri luokkien alkioita, luokaksi valitaan se, jonka alkioita on suurin määrä (Aggarwal 2015). KNN algoritmin luokittelun toiminta on havainnollistettu kuvassa 5, jossa luokkia havainnollistetaan sinisellä ja punaisella luokalla ja testialkiota vihreällä ympyrällä.



Kuva 5: kNN-luokittelijan toiminta binääriluokittelussa[1]

Kuvasta 5 havaitaan, että mikäli on määritetty k = 1, luokittelija ottaa huomioon vain testialkioita lähinnä olevan alkion, ja luokaksi määräytyy sininen. Mikäli k = 3, lähimpiä alkioita ovat sininen alkio ja kaksi punaista alkiota, jolloin luokaksi määräytyy punainen. Mikäli k = 5 lähimmät viisi alkiota koostuvat kolmesta sinisestä ja kahdesta punaisesta alkiosta, jolloin luokaksi määräytyy sininen.

3.2.4 Säännöllinen lauseke

Säännölliset lausekkeet (regular expression) ovat aakkosista ja erikoismerkeistä koostuva merkintätapa, jolla voidaan kuvata luonnollista kieltä (Clark et al. 2013. p. xliii). Säännöllinen lauseke ei ole koneoppimista hyödyntävä työkalu, vaan kokoelma menetelmiä kielen rakenteiden kuvaamiseen. Säännöllinen lauseke voidaan nähdä myös matemaattisena menetelmänä kuvata kieltä, sillä se sisältää matemaattisia

[1] Mukailtu MachineLearning — KNN using scikit-learn https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75. (viitattu 20.4.2020)

operaatioita tekstin osille (Good 2005. s.xxv). Säännöllisessä lausekkeessa voidaan esittää tekstijonoja kirjaimien, kirjainjoukkojen ja joukko-opin operaatioiden keinoin. Näitä ovat esimerkiksi unioni, ketjutus (concatenation) ja villikorttioperaattori *****, jolla voidaan esittää jotain ennalta määrittelemätöntä kirjainta tai kirjainjoukkoa. (Butterfield et al. 2016) Tässä tutkimuksessa Säännöllisten lausekkeiden käyttö perustuu Pythonin sisäänrakennetun regex-kirjaston hakutoimintoon.

Säännöllisiä lausekkeita voidaan myös käyttää haluttujen tekstijonojen etsimiseen laajemmasta tekstijoukosta. Säännöllisen lausekkeen käyttö valikoitui käytettäväksi tässä tutkimuksessa edellä mainittujen joukko-opillisten ominaisuuksiensa vuoksi. Haluttua hakusanaa voidaan etsiä riippumatta esimerkiksi isoista alkukirjaimista tai suomen kielen sijapäätteistä.

3.3 Mallien vertailumenetelmät

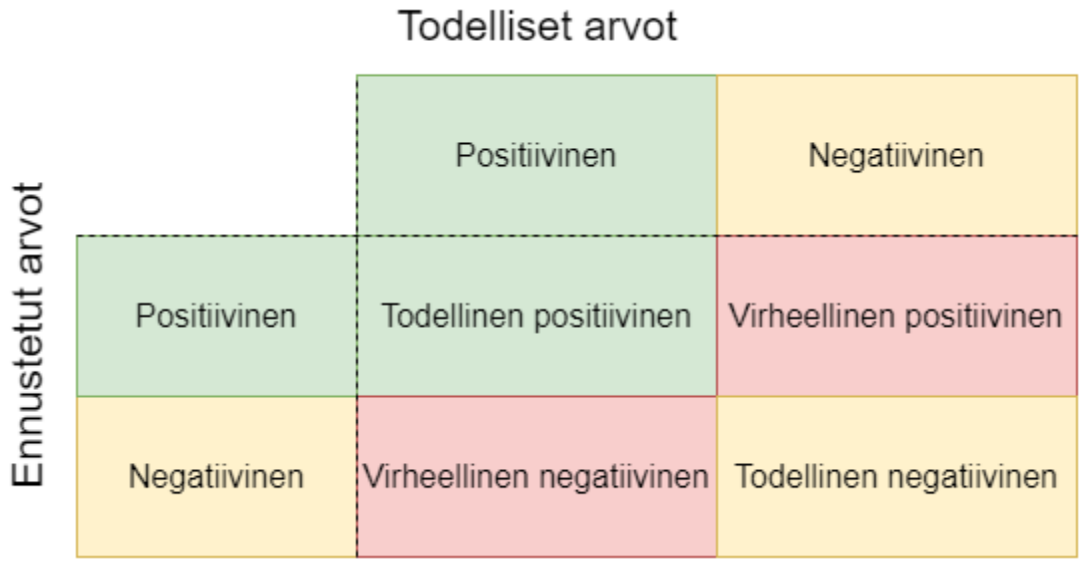
Tässä aliluvussa kuvataan luokittelumallien vertailussa käytettävät menetelmät ja metriikat. Ensin esitetään, miten luokittelutulokset jaetaan totuusarvoihin ja miten luokittelijalle asetettu kynnysarvo vaikuttaa luokittelutulokseen. Tämän jälkeen esitetään luokittelun totuusarvojen perusteella lasketut yleisimmät metriikat. Lopuksi mallien toimintaa eri kynnysarvoilla esitetään kattavampien kuvaajien avulla.

3.3.1 Luokittelun totuus- ja kynnysarvot

Koska yksi tutkimuksen tavoitteista on arvioida luokittelussa käytettyjä malleja, on tärkeä löytää yhteneväiset vertailumenetelmät ja mittarit, joiden perusteella mallien toimivuutta voidaan arvioida. Ohjatussa koneoppimisessa luokittelumallien arviointi tapahtuu syöttämällä mallille testausdataa, jonka luokka on ennalta tiedossa (Aggarwal 2015). Mallin luokittelemaa tulosta voidaan verrata oikeaan luokkaan, jolloin voidaan asettaa luokitukselle totuusarvo. Kuten luvussa 2.3.4. todetaan, binääriluokittelijan kohdalla luokat ovat positiivinen ja negatiivinen, joista positiivinen luokka kuvaa tutkimuksen kohteena olevaa luokkaa.

Mallien arvioinnissa jaetaan luokiteltavat testiaineiston alkiot mallin luokittelutuloksen perusteella *totuusarvoihin*. Testiaineiston todellista luokitusta ja luokittelijan antamaa luokitusta vertaamalla voidaan jakaa luokitukset *todellisiin positiivisiin* (true positive, TP), *todellisiin negatiivisiin* (true negative, TN), *virheellisiin positiivisiin* (false positive FP) ja

virheellisiin negatiivisiin (false negative, FN) totuusarvoihin. Näiden avulla voidaan muodostaa *sekaannusmatriisi* (confusion matrix), joka kuvaa binääriluokittelijan luokittelutuloksen tyyppiä. Positiivisena luokkana sekaannusmatriisissa kuvataan tutkimuskohteena olevaa luokkaa. (Tharwat 2018; Hasanin et al. 2020) Sekaannusmatriisi on esitetty kuvassa 6.

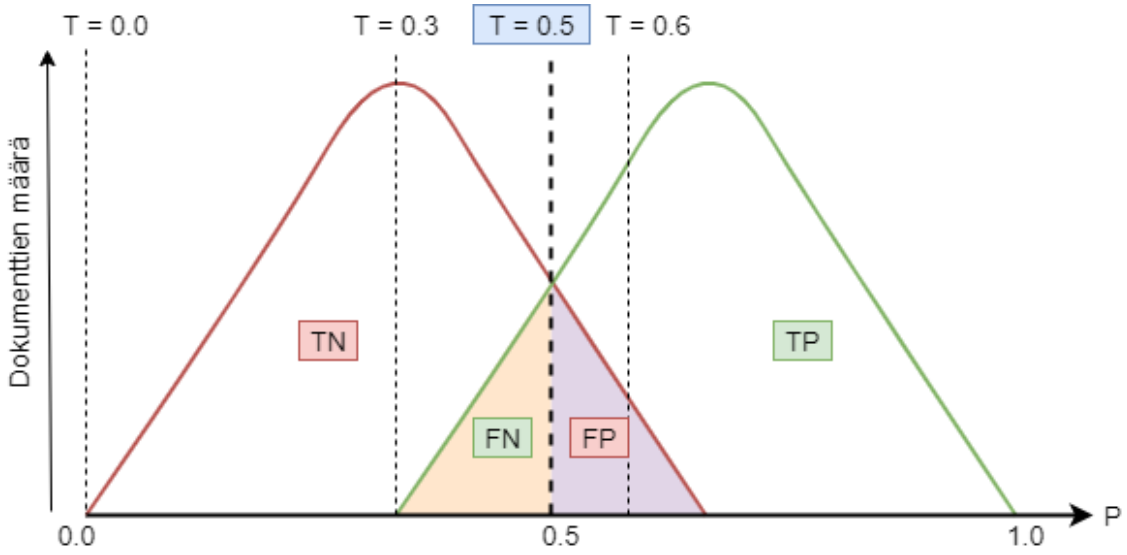


Kuva 6: Sekaannusmatriisi (mukaillen Tharwat 2018; Hasanin et al. 2020))

Sekaannusmatriisissa on vaaka-akselilla alkioiden todelliset arvot ja pystyakselilla luokittelijan niille antamat arvot. Totuusarvo muodostuu vertaamalla todellisia arvoja ja luokittelijan arvoja. Esimerkiksi mikäli alkion todellinen luokka on positiivinen ja ennustettu luokka on negatiivinen, saa alkio totuusarvokseen virheellisen negatiivisen. Mikäli alkion todellinen luokka on positiivinen ja ennustettu luokka on positiivinen, saa alkio totuusarvokseen todellisen positiivisen.

Luokittelijan toimintaan vaikuttaa merkittävästi *kynnysarvon* (threshold) käsite. Kynnysarvo kuvaa rajaa, johon luokittelijan antamaa luokituksen todennäköisyyttä verrataan. Mikäli luokiteltavan alkion luokituksen todennäköisyys alittaa asetetun kynnysarvon, luokitellaan alkio negatiiviseksi. Mikäli todennäköisyys ylittää kynnysarvon, alkio saa positiivisen luokituksen. (Krzanowski 2009 s.7) Jos esimerkiksi luokittelijan kynnysarvoksi on asetettu 0.5, ja luokittelija ennustaa alkion positiivisen luokan todennäköisyydeksi 0.45, se saa luokituksen negatiivinen. Jos kuitenkin kynnysarvoa muutetaan esimerkiksi arvoon 0.3, sama alkio luokiteltaisiinkin positiiviseen luokkaan. Useimmissa luokittelijoissa on oletuskynnysarvona 0.5, mutta tämä ei välttämättä ole

ideaalinen valinta epätasapainoiselle aineistolle (Zou et al. 2016). Kuvassa 7 on havainnollistettu kynnysarvojen vaikutus sekaannusmatriisin arvojen avulla.



Kuva 7: Kynnysarvon vaikutus sekaannusmatriisin arvoihin (mukaillen Tharwat 2018)

Kuvassa y-akseli kuvaa teoreettisten jakaumien avulla luokiteltavan aineiston määrää ja x-akseli positiivisen luokituksen todennäköisyyttä P. Mitä suurempi P:n arvo, sen todennäköisempää on, että kuuluu positiiviseen luokkaan. Havainnollistetulla kynnysarvolla T = 0.5 havaitaan, että malli saa yhtä suuren osan sekä todellisia. positiivisia ja todellisia negatiivisia, että yhtä suuren osan virheellisiä positiivisia ja virheellisiä negatiivisia. Mikäli kynnysarvoa lasketaan arvoon T = 0.3, havaitaan, että tässä tilanteessa malli ei tuota lainkaan virheellisiä negatiivisia arvoja, mutta virheellisten positiivisten määrä kasvaa merkittävästi. Kynnysarvon asettamisessa onkin siis usein kyse valintatilanteesta, johon vaikuttaa luokittelijan sovelluskohde. Luokittelussa tulee valita, parannetaanko esimerkiksi positiivisten arvojen luotettavaa tunnistamista (laskemalla kynnysarvoa) sillä kustannuksella, että se johtaa samalla virheellisten positiivisten määrän kasvuun.

3.3.2 Luokittelun yleiset tunnusluvut

Sekaannusmatriisin perusteella voidaan laskea mallin toimintaa kuvaavat tunnusluvut. Tässä tutkimuksessa luokittelumalleja arvioidaan moniulotteisen mittariston avulla, jotta mallien ominaisuudet ja mahdolliset syy-seuraussuhteet käyvät ilmi. *Tarkkuus* (precision) kuvaa todellisten positiivisten osuutta kaikista positiiviseksi luokitelluista

alkioista (Jiang 2013 s. 33). *Saanti* (recall) puolestaan mittaa todellisten positiivisten alkioiden osuutta kaikista aineiston positiivisista alkioista (Johnson & Khoshgoftaar 2019). Toisin sanoen, tarkkuus mittaa miten suuri osuus positiiviseksi luokitelluista alkioista ovat oikeasti positiivisia ja näin ollen relevantteja. Saanti puolestaan mittaa sitä, miten suuren osuuden aineiston positiivisista alkioista malli onnistuu luokittelemaan positiiviseksi. *F1-arvo* (F1-score) on tarkkuuden ja saannin geometrinen keskiarvo (Jiang 2013 s. 33). Mittarien laskennalliset kaavat on esitelty kaavoissa 4–8.

$$Kokonaistarkkuus = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}, \quad (4)$$

$$Tarkkuus = \frac{True\ Positive}{True\ Positive + False\ Positive}, \quad (5)$$

$$Saanti = Tod.positiivisten\ aste = \frac{True\ Positive}{True\ Positive + False\ Negative}, \quad (6)$$

$$Virh.\ positiiviste\ aste = \frac{False\ Positive}{False\ Positive + True\ Negative}, \quad (7)$$

$$F1 = \frac{2 \cdot Tarkkuus \cdot Saanti}{Tarkkuus + Saanti}, \quad (8)$$

(Aggarwal 2018 s.228; Tharwat 2018; Johnson & Khoshgoftaar 2019)

Mallin *kokonaistarkkuus* (accuracy), kuvaa mallin oikein luokittelemien alkioiden suhdetta kaikkiin luokituksiin. Kokonaistarkkuus on kuitenkin mittarina harhaanjohtava, eikä sen perusteella pystytä tekemään päätelmiä mallin toiminnasta. (Davis & Goadrich 2006) Esimerkkinä tästä ovat luvussa 3.1. mainitut aineistot, joilla on äärimmäisen suuri luokkaepätasaisuus. Tämä johtaa siihen, että malli ennustaa vain enemmistöluokkaa. Tässä tapauksessa tarkkuus voi tuottaa näennäisesti hyviä tuloksia pelkästään tilastolliseen todennäköisyyteen perustuen, mikäli myös testiaineisto jakautuu epätasaisesti. (Johnson & Khoshgoftaar 2019) On siis olennaista, että tarkkuuden lisäksi mallin toimintaa arvioidaan monipuolisemmilla mittareilla tarkemman kokonaiskuvan ymmärtämiseksi ja varsinkin epätasaisen aineiston vaikutuksen huomioon ottaen.

Tarkkuuden ja saannin välillä joudutaan tekemään valinta riippuen luokittelun kohteesta (Johnson & Khoshgoftaar 2019). Tarkkuuteen painottuvaa mallia tulee suosia tilanteessa, jossa halutaan välttää virheellisten positiivisten tulosten määrää ja virheellisen negatiivisen tuloksen haitta on pieni. Saantiin painottuva malli puolestaan sopii tilainteisiin, joissa halutaan minimoida virheelliset negatiiviset tulokset sillä kustannuksella, että virheellisten positiivisten tulosten määrä kasvaa. Vaihtokaupan lisäksi metriikoihin sisältyy rajoitteita. Tarkkuus kärsii epätasaisesta aineistosta eikä sovellu käytettäväksi ainoana mittarina, sillä se ei ota kantaa virheellisiin negatiivisiin tuloksiin. Tämän vuoksi on yleistä käyttää mittarina sekä tarkkuutta että saantia. Saanti ei myöskään kärsi epätasaisesta opetusaineistosta yhtä merkittävästi kuin tarkkuus. (Johnson & Khoshgoftaar 2019) Voidaan siis todeta, että tarkkuutta ja saantia tulee käyttää yhdessä toisiaan tukien, antaen painoarvon sille, kumman tarkempi tulos on kriittisempi tutkimuskohteen kannalta.

3.3.3 ROC-käyrät ja tarkkuus-saanti-käyrät

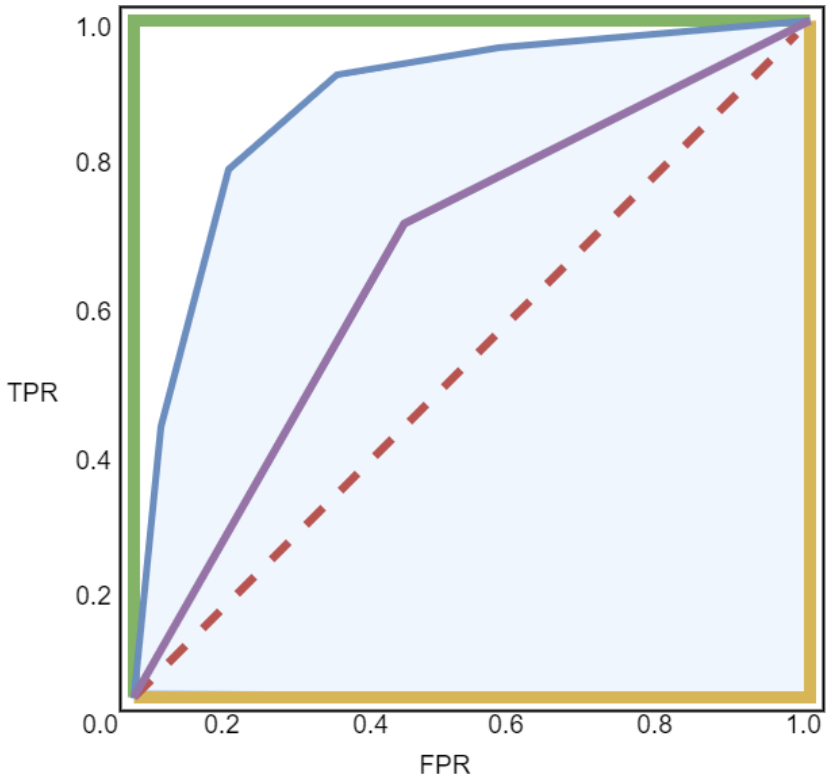
Kuten kuvasta 7 voidaan havaita, kynnysarvon pienentäminen nostaa todellisten positiivisten luokitusten määrää. Luokittelija ei kuitenkaan yleensä voi kasvattaa todellisten positiivisten määrää ilman, että virheellisten positiivisten määrä kasvaa samalla. (Zou et al. 2016) Tällöin muodostuu valintatilanne, jota voidaan kuvata. *ROC-käyrän* (receiver operator characteristic) avulla. (Aggarwal 2018 s. 227)

Binääriluokitteluongelmassa käytettyä luokittelumallia voidaan siis yksittäisten arvojen lisäksi arvioida kattavammin ROC-käyrän avulla. ROC-käyrä vertaa luokittelijan *todellisten positiivisten asteen* (true positive rate, TPR) ja *virheellisten positiivisten asteen* (false positive rate, FPR) suhdetta eri kynnysarvoilla. ROC-käyrän tarkoitus on kuvata ja arvioida luokittelijan toimintaa kaikilla mahdollisilla kynnysarvoilla, sen sijaan, että luokittelijalle asetettaisiin yksi vakioitu kynnysarvo. (Krzanowski 2009 s. 19) Koska ROC-käyrä kuvaa luokittelijan toimintaa kaikilla eri kynnysarvoilla, voidaan sitä käyttää mm. optimaalisen kynnysarvon löytämiseen.

ROC-käyrä käyttäytyy eri tavoin eri luokittelijoilla. Esimerkiksi tässä tutkimuksessa käytetty säännöllisiin lausekkeisiin perustuva malli antaa vain päätöksen siitä, kumpaan luokkaan kukin alkio kuuluu. Monet muut luokittelijat antavat luokituspäätöksen lisäksi myös todennäköisyyden, jolla päätös on tehty (Tharwat 2018). Tämä tarkoittaa, että yksinkertaisimmillaan päätös on binäärinen {0,1}, jolloin ROC-kuvaajassa on vain yksi kynnysarvo. Kuitenkin esimerkiksi tämän tutkimuksen empiirisessä osiossa käytettävät

koneoppimisen perustuvat luokittelijat antavat myös luokituksen todennäköisyyden, jolloin jokaisen luokituksen kohdalta voidaan piirtää piste ROC-käyrään.

ROC-käyrän avulla voidaan kuvata luokittelumallin toimintaa myös yksittäisen metriikan avulla. Luokittelijan luokitusten oikeellisuutta voidaan arvioida k*äyrän alle jäävän pinta-alan* (AUC) avulla. Koska TPR ja FPR saavat arvoja välillä {0,1}, myös pinta-ala AUC saa arvoa samalta väliltä (Krzanowski 2009 s. 26). AUC on yleisesti käytetty mittari tilanteessa, joissa luokkien välillä vallitsee luokkaepätasapaino. On kuitenkin huomattava, että äärimmäinen luokkaepätasapaino voi aiheuttaa AUC:ta käytettäessä epäluotettavia tuloksia (Hasanin et al. 2020). Kuvassa 8 on esitetty esimerkkejä eri luokittelijoiden ROC-käyristä, sekä AUC:n muodostuminen.

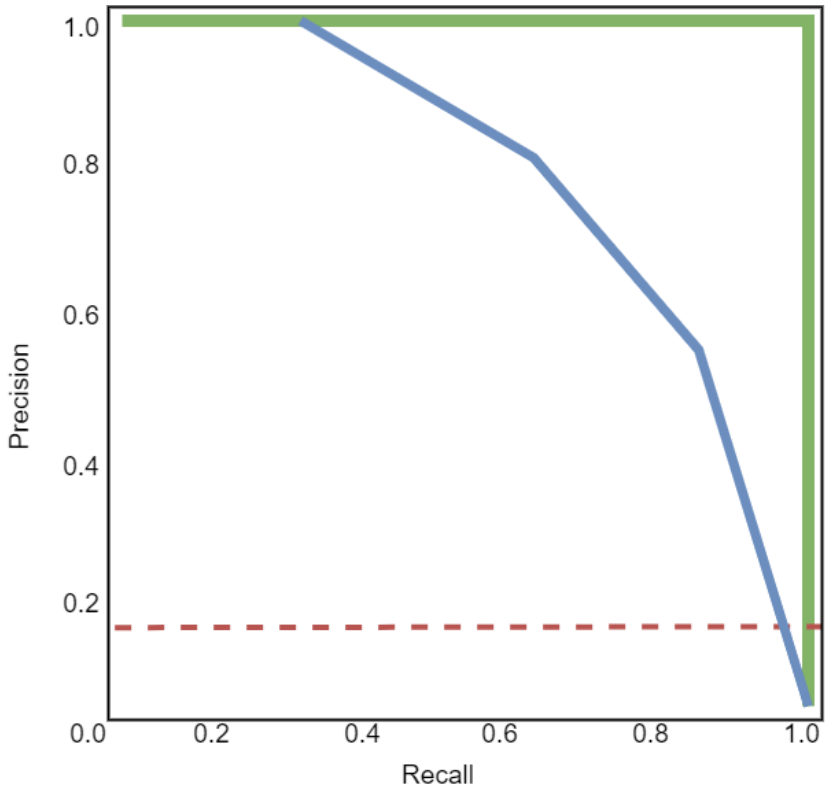


Kuva 8: Esimerkki ROC-käyristä ja käyrän alle jäävästä pinta-alasta (AUC) (mukaillen Tharwat 2018)

Kuvan 8 kuvaajassa akseleita ovat todellisten positiivisten aste (TPR) ja virheellisten positiivisten aste (FPR). ROC-käyrä kuvaa TPR:n ja FPR:n suhdetta siten eri kynnysarvoilla. Kuvaajan vasemmassa yläkulmasta löytyvä vihreä käyrä kuvaa täydellistä luokittelijaa, joka luokittelee jokaisen alkion oikeaan luokkaan. Tässä tilanteessa käyrän alle jäävä pinta-ala eli AUC saa maksimiarvon 1. Keltainen viiva puolestaan luokittelijaa, jossa malli luokittelee jokaisen alkion päinvastaiseen luokkaan

kuin mihin ne oikeasti kuuluvat, jolloin AUC saa minimiarvon 0. Punainen katkoviiva kuvaa kolikonheittoa vastaavaa luokittelijaa, jossa luokittelijalla ei ole kykyä erottaa mallista kumpaankaan luokkaan viittaavia piirteitä ja näin ollen luokittelee kumpaan tahansa luokkaan samalla todennäköisyydellä. Tässä tilanteessa AUC saa arvon 0.5. Kuvaaja luokittelee siis kolikonheittoa paremmin siinä tilanteessa, kun käyrä saa arvoja punaisen katkoviivan vasemmalta puolelta. Sininen käyrä simuloi hyvin suoriutuvaa luokittelumallia, jonka avulla demonstroidaan myös käyrän alle jäävä pinta-alaa. Violetti käyrä kuvaa mallia, jonka todennäköisyysjakaumaa ei voida arvioida, vaan ROC-käyrä piirretään binäärisen luokittelupäätöksen mukaisesti. Sininen käyrä on siis ROC-mittarilla kuvattuna parempi käyrä kuin violetti, sillä käyrä kulkee lähempänä täydellistä luokittelijaa vasemmassa yläkulmassa.

Tarkkuuden ja saannin avulla voidaan myös piirtää kuvaaja: tarkkuus-saanti-käyrä, jonka alle jäävän pinta-alan avulla voidaan arvioida mallin toimivuutta. Tarkkuus-saanti-käyrän alle jäävä pinta-ala (PR) saa arvoja väliltä {0,1}, jossa 1 kuvaa täydellistä luokittelijaa. Tarkkuus-saanti-käyrä toimii epätasapainoisella opetusaineistolla ROC-käyrää paremmin, mikä puoltaa sen käyttöä myös tässä tutkimuksessa, jossa negatiivisten alkioiden luokkamäärä on positiivista suurempi. (Liu & Bondell 2019) Myös Johnson ja Khosgoftaar (2019) arvioivat, että ROC-käyrä on ylioptimistinen epätasapainoisella aineistolla, joskin ROC ja tarkkuus-saanti-käyrä voivat tukea toisiaan, jolloin hyvä tarkkuus-saanti-käyrän tulos voi validoida ROC-käyrän alle jäävän pinta-alan tulosta. (Johnson & Khoshgoftaar 2019) On siis perusteltua sisällyttää tarkkuus-saanti-käyrän alle jäävä pinta-ala tutkimuksen metriikoihin, jolloin saadaan kattavampi kokonaisymmärrys mallin toiminnasta, sekä saadaan vähennettyä luokkien epätasaisuudesta koituvaa vääristymää. Esimerkki tarkkuus-saanti-käyrästä on esitetty kuvassa 9.



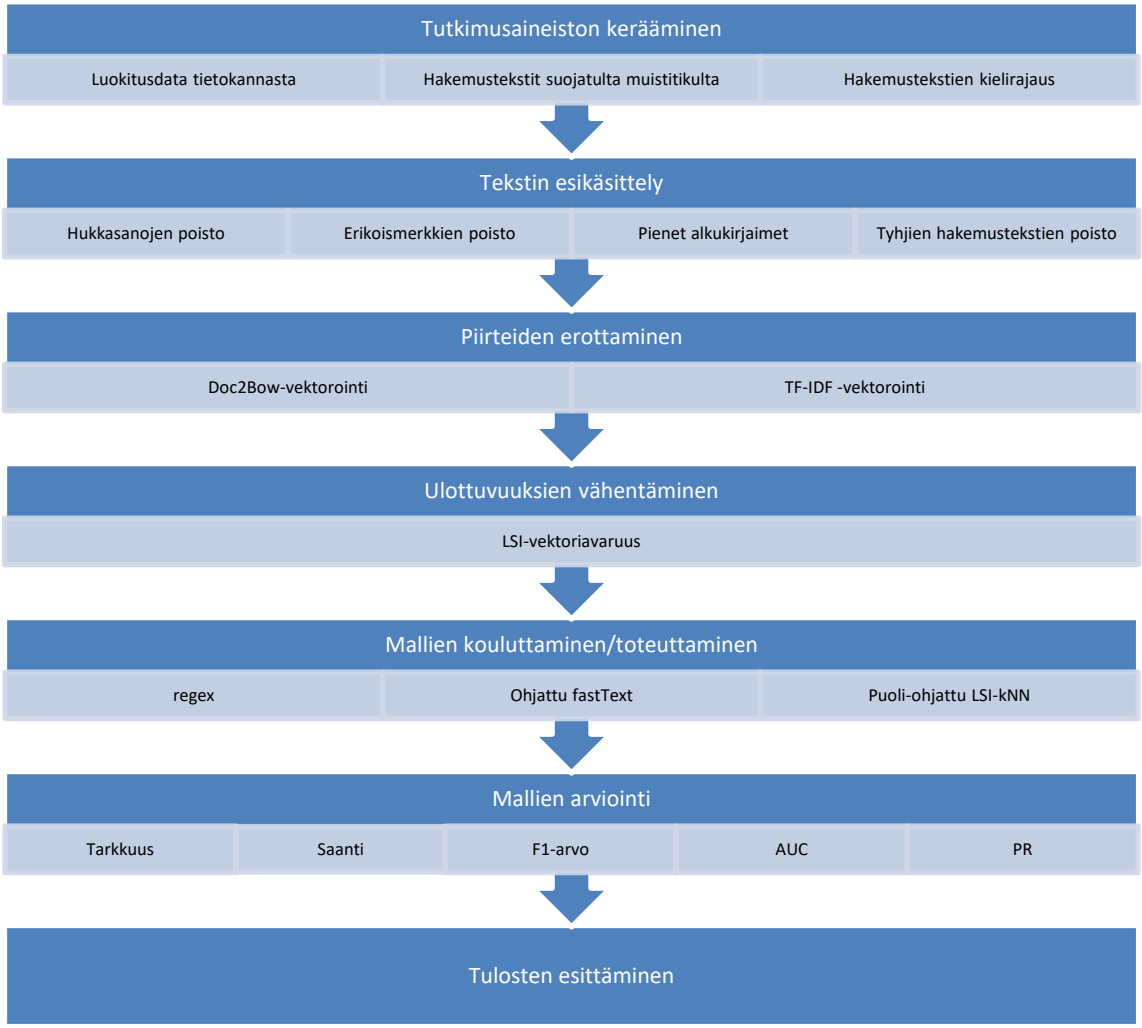
Kuva 9: Esimerkki PR-käyrästä (mukaillen Tharwat 2018; Liu & Bondell 2019)

Kuvassa 9 esitetyn tarkkuus-saanti-käyrän kuvaaja piirretään vertailemalla luokittelijan tarkkuuden ja saannin arvoja pareittain eri kynnysarvoilla. Vaikka PR-käyrän kuvaaja kuvaa tarkkuuden ja saannin suhdetta eri pisteissä, voidaan siitä laskea myös yksittäinen metriikka käyrän alle jäävän pinta-alan avulla. (Boyd et al. 2013) Täydellinen luokittelija PR-kuvaajalla mitattuna on tässä esimerkkitapauksessa kuvattu vihreällä käyrällä (Tharwat 2018). Tässä tapauksessa kuvaaja muodostuu oikeaan yläkulmaan, jolloin joka pisteessä mitattuna joko tarkkuus tai saanti (tai molemmat) saavat aina arvon 1, jolloin myös käyrän alle jäävän pinta-alan suuruus PR = 1. Tilastollinen todennäköisyys, eli yleisin valinta on kuvattu punaisella katkoviivalla ja sen arvo lasketaan luokiteltavan luokan ja koko aineiston suhteen perusteella.

4. EMPIIRINEN TUTKIMUSOSIO

4.1 Empiirisen osion tutkimusasetelma

Diplomityön empiirinen vaihe toteutettiin tutkimussuunnitelman mukaisessa koeasettelussa. Tutkimusta varten rakennettiin Python-ohjelmointikielellä interaktiivinen Jupyter Notebook, johon empiirisen tutkimuksen vaiheet kirjoitettiin. Notebook toteuttaa datan keräämisen lähdejärjestelmistä, datan yhdistämisen, esikäsittelyn, mallien kouluttamisen, pisteyttämisen, arvioinnin ja visualisoinnin. Kuvan 3 mukaisesta yksinkertaistetusta NLP-koneoppimismallista on mukailtu tässä tutkimuksessa hyödynnettävä empiirisen tutkimuksen koeasetelma. Asetelma on esitetty kuvassa 10.



Kuva 10: Tutkimuksessa käytetty NLP-koneoppimisprosessi

Kuvasta nähdään empiirisessä tutkimuksessa käytetyt datan lähteet, esikäsittelytoimenpiteet, mallivalinnat sekä malleja kuvaava mittaristo. Tutkimus alkoi aineiston keräämisellä, josta korpuksen esikäsittelyn jälkeen opetettiin valitut mallit aineistoa käyttäen. Lopuksi mallien toimintaa arvoitiin valitun mittariston perusteella ja tulokset esitettiin taulukkojen ja kuvaajien avulla.

4.2 Tutkimusaineiston muodostus

Tutkimusaineistona käytettiin Business Finlandin rahoitushakemusten dataa. Aineisto koostui kahdesta osasta, Business Finlandin asettama luokitus sekä hakemuksen tekstimuotoinen osa. Tutkimusaineisto rajattiin seuraavien rajoitusten mukaisesti:

- Aikarajaus vuosille 2014–2018
- Aineistorajaus, jossa valitaan vain rahoitushakemuksia, joiden rahoituspalveluun kuuluu cleantech-luokitus
- Kielirajaus suomen kielelle

Jokaisen mallin kohdalla käytettiin samaa aineistoa, joka muodostettiin luvussa 3.4. kuvaillun tavan mukaisesti. Taulukossa 2 on kuvattu aineiston sisältö lukuina

Taulukko 2: Tutkimusaineiston koko ja jakauma

Dokumenttien kokonaismäärä (kpl)	6746
Cleantech-luokiteltuja dokumentteja (kpl	1291
Ei cleantech-luokiteltuja dokumentteja (kpl)	5455
Cleantech-luokiteltujen dokumenttien (%)	19.1

4.3 Empiirisen tutkimuksen toteutus

Tutkimus toteutettiin noudattaen luvussa 1.4. esitettyä tutkimuksen asettelua. Empiirisen tutkimusosion suorituksessa käytettiin apuna Řehůřekin (2019) esittämän *Gensim*-Python-kirjaston dokumentaatiota[2]. Gensim on NLP-kirjasto, jonka taustalla vaikuttavat tavoitteet ovat korpuksen koosta riippumaton mallinnus, intuitiivinen rajapinta, helppo käyttöönotto ja suosituimpien NLP-algoritmien sisällyttäminen. Näihin algoritmeihin sisältyy tässä tutkimuksessa hyödynnetyt TF-IDF ja LSI. (Řehůřek & Sojka 2010) Empiirisen tutkimusosion tarkempi toteutus on kuvattu tässä aliluvussa. Empiirisen osion rakentamiseen käytetty Python-koodi tulee rajoitetusti saataville GitHub-repositorioon[3]. Julkaistavasta koodista on poistettu kaikki kohdeyritystä koskeva materiaali ja datalähteeksi on vaihdettu avoin tekstikorpus.

4.3.1 Tutkimusaineiston lukeminen

Tutkimusaineisto kerättiin kahdesta lähteestä. Hakemusten luokitukset ja metatiedot haettiin tietokantakyselyllä kehitystietokannasta. Hakemusten teksti haettiin suojatulta muistitikulta tietosuojan varmistamiseksi. Tekstimuotoinen data sisältää hakemustekstin ja diaarinumeron, joka on hakemustekstin yksilöllinen tunniste. Tekstidata yhdistettiin siten, että jokaista diaarinumeroa vastaa tekstimuotoinen kenttä, johon on yhdistetty kaikki diaaria vastaavat uniikit tekstikentät. Muistitikulta luetusta datasta poistettiin muut kuin suomenkieliset hakemukset Pythonin kielentunnistuskirjaston avulla. Tässä vaiheessa poistettiin myös rahoituspalvelut hakemuksineen, jotka eivät sisällä cleantech-luokitusta.

Hakemustekstien luokitusdata, diaari- ja metatiedot luettiin tietokannasta, joka sisältää historiadataa vuoteen 2018 asti. Data ei myöskään sisällä turvaluokiteltua aineistoa. Kantaan tehtiin SQL-kysely, joka noudattaa luvussa 4.2. kuvattua tutkimusaineiston muodostamistapaa. Diaaritietokenttiä yhdistämällä saatiin tuotettua diaarinumero, joka vastaa hakemustekstien diaarinumeroita. Datasta poistettiin tutkimusaineiston rajausten mukaisesti kaikki diaarit, joille ei löydy cleantech-luokituksia sisältävää rahoituspalvelua.

Hakemustekstit ja luokitusdata yhdistettiin yhteisen diaarinumeron perusteella. Tämän tutkimuksen kannalta tarpeettomat kentät poistettiin, jonka jälkeen suoritettiin

---

[2] (Rehurek 2019) https://radimrehurek.com/gensim/auto_examples/index.html (viitattu 25.4.2020)

[3] https://github.com/MikkonenTS/NLP_Classification

esikäsittelytoimenpiteet. Esikäsittelyssä poistetiin tyhjät hakemustekstit, hukkasanat ja erikoismerkit hakemusteksteistä. Tekstin luokitus muokattiin tutkimusasettelun mukaiseen binäärimuotoon, jossa cleantech-luokan tunniste ”cleantech” jätettiin ennalleen, mutta muihin luokkiin asetettiin tunniste alkuperäisen luokituksensa tilalle ”ei cleantech”. Tämä toteutettiin datan yksinkertaistamiseksi, sillä muilla luokituksilla ei ole binääriluokitteluongelman kannalta merkitystä.

4.3.2 Mallien rakentaminen

Säännöllisiin lausekkeisiin perustuvassa mallissa luokiteltiin hakemukselle luokka ’cleantech’, mikäli hakemustekstistä löytyy positiivinen tulos regex-kyselylle: ”r'cleantech'”. Kysely ottaa huomioon isot ja pienet alkukirjaimet ja etsii hakusanaa myös yhdyssanoista tai merkkijoukkojen sisältä. Säännöllisiin lausekkeisiin perustuva malli rakennettiin käyttämällä Pythonin regex-kirjastoa[4].

FastText-luokitus rakennettiin Pythonin fastText-kirjaston avulla hyödyntäen kirjaston ohjatun oppimisen moduulia[5]. Opetuksessa jaettiin tutkimusaineisto opetus- ja testijoukkoon suhteella 80%-20%. Opetusdata kirjoitettiin tekstitiedostoihin, joissa on toisessa sarakkeessa hakemuksen luokitus ja toisessa esikäsitelty rahoitushakemusteksti. FastText- malli opetettiin tiedostoon kirjoitetun ja luokitellun opetusdatan avulla. Mallin opetusparametrit selitteineen on esitetty taulukossa 3.

Taulukko 3: fastText-mallin opetusparametrit

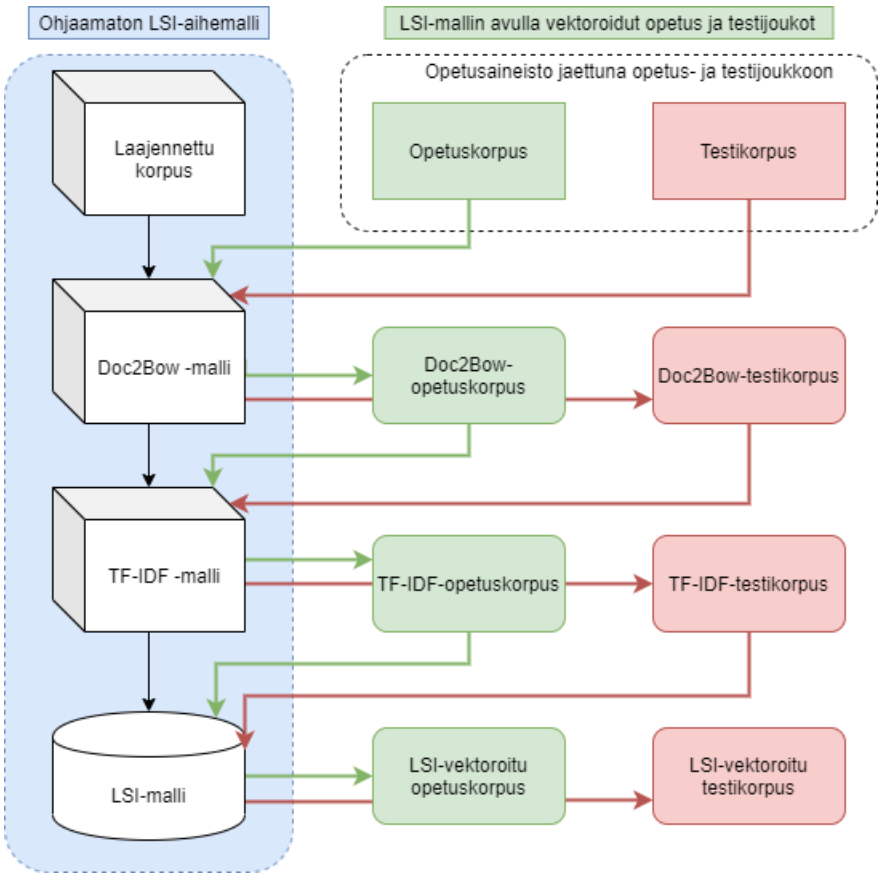
Parametri	Arvo	Selite
epoch	50	Montako iteraatiota malli käy läpi
dim	100	Sanavektorin ulottuvuudet
word_ngrams	2	Minkä mittaisia sanayhdistelmiä tutkitaan
ws	3	Kuinka monta sanaa tarkastellaan kunkin sanan molemmin puolin
lr	0.01	Kuinka paljon malli oppii yhdestä iteraatiosta
minn	3	Merkki ngramien minimipituus
maxn	8	Merkki ngramien maksimipituus

[4] Python regex-kirjaston dokumentaatio https://docs.python.org/2/library/re.html (viitattu 25.4.2020)

[5] fastText-kirjaston tekstiluokittelumoduulin dokumentaatio https://fasttext.cc/docs/en/python-module.html#text-classification-model (viitattu 2.5.2020)

FastText-luokittelijan opettamisen jälkeen sitä käytettiin luokittelemaan testidatajoukko. Testaaminen toteutettiin fasttextin *predict*-funktiolla, joka arvioi testidatan alkiolle luokan, sekä todennäköisyyden, jolla luokka on valittu. Dataa muokattiin niin, että jokainen luokituksen todennäköisyys arvioi luokan 'cleantech' todennäköisyyttä asteikolla {0,1}.

LSI-kNN -mallissa opetettiin puoliohjatun oppimisen malli, jossa on sekä ohjaamattoman oppimisen prosessi sekä ohjatun oppimisen prosessi. LSI--kNN -mallin ohjaamattomassa vaiheessa mallin rakentaminen aloitettiin keräämällä uniikkien sanojen esiintymistiheys sanastoon. Tämän avulla voitiin esimerkiksi poistaa joukosta sanat, jotka esiintyvät vain kerran. Tekstit muokattiin opetusta varten muotoon, jossa ne voidaan syöttää lause kerrallaan funktioihin, jotka toteuttavat piirteiden erottamisen, ulottuvuuksien vähentämisen ja LSI-mallin koulutuksen kuvassa 11 esitellyllä tavalla.



Kuva 11: Piirteiden erottaminen ja ulottuvuuksien vähentäminen Pythonilla Gensim-kirjaston avulla [6]

[6] Mukailtu Gensim: Topics and Transformations https://radimrehurek.com/gensim/auto_examples/core/run_topics_and_transformations.html (viitattu 30.4.2020)

Kuva havainnollistaa LSI-prosessin kulkua, jossa korpuksesta muodostettiin Doc2Bow-vektoreita, jotka kuvaavat kunkin dokumentin sanoja ja niiden esiintymistiheyksiä BOW-muodossa luvussa 2.3.2. esitellyllä tavalla. Doc2Bow-vektoreista muodostettiin TF-IDF-vektoreita aineiston normalisoinnin vuoksi. Vektorit syötettiin lause kerrallaan ohjaamattoman LSI-mallin opetukseen. Tämä prosessi muodosti kuvassa 11 sinisellä pohjalla merkatun ohjaamattoman aihemallinnusmallin, jota käytettiin opetus- ja testimateriaalin piirteiden erotukseen ja ulottuvuuksien vähentämiseen. Piirteiden erottaminen ja ulottuvuuksien vähentäminen on esitetty liitteessä 1 julkisten dokumenttien tiivistelmiä käyttäen.

Vertailukelpoisuuden varmistamiseksi LSI-kNN-mallin ohjatun oppimisen vaiheessa aineistojako opetus- ja testijoukkoon noudatti samaa jakoa, joka toteutettiin fastText-mallille. Mallin opetuksessa muunnettiin molemmat joukot ensin Doc2Bow-muotoon yhteisen sanaston avulla (piirteiden erottaminen), jonka jälkeen tulokset normalisoitiin koko aineistoon pohjautuvan TF-IDF-mallin avulla. TF-IDF-vektorit muunnetaan vertailtavaan muotoon kääntämällä ne LSI-vektoreiksi (ulottuvuuksien vähentäminen).

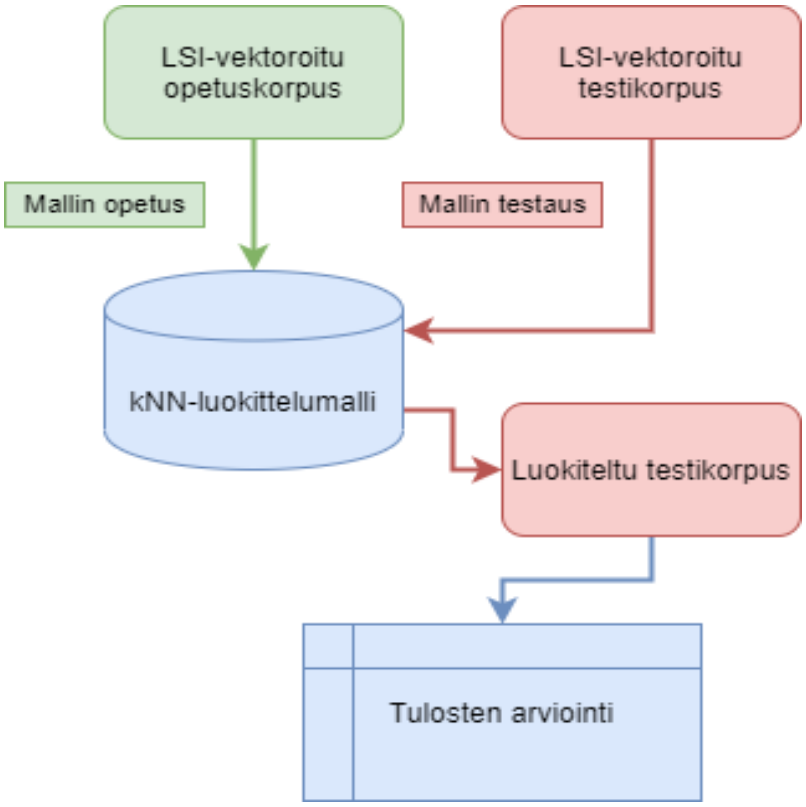
Ennen luokittelua opetusaineistolle muodostetaan yliotanta aineiston luokkaepätasapainon vaikutuksen pienentämiseksi. Yliotanta toteutetaan käyttämällä SMOTE-menetelmää[7], joka luo opetusdatan perusteella keinotekoisia datapisteitä luvun 3.1 mukaisesti. Opetusdataan muodostettiin uusia positiivisia datapisteitä suhteella 0.5 joka tarkoittaa, että positiivisesta aineistosta muodostetaan yliotanta siten, että positiivisten alkioiden määrä on puolet negatiivisten alkioiden määrästä. FastText-malliin yliotantaa ei ole sovellettu, sillä mallin piirteiden erottaminen ja ulottuvuuksien vähentäminen tapahtuu Python-paketin sisäisissä toiminnoissa. LSI-malin opetusaineiston suhteet on esitetty taulukossa 4.

[7] Imbalanced Learn SMOTE dokumentaatio https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html (viitattu 4.5.2020)

Taulukko 4: Opetusaineiston jakauma SMOTE-yliotannan jälkeen

Cleantech	2190
Ei cleantech	4381
Yhteensä	6571
Cleantech-osuus aineistosta	33.3%

LSI-vektoriavaruuteen muunnettujen hakemustekstien perusteella opetettiin luvussa 3.2.3. esitetty ohjatun oppimisen k-Nearest Neighbors (kNN) -luokittelumalli. Parametrina kNN-algoritmille annettiin haluttu k:n arvo k = 3. Opetuksen jälkeen luokittelumallin avulla luokiteltiin LSI-vektoroidun testikorpuksen dokumentit kuvan 12 mukaisesti.

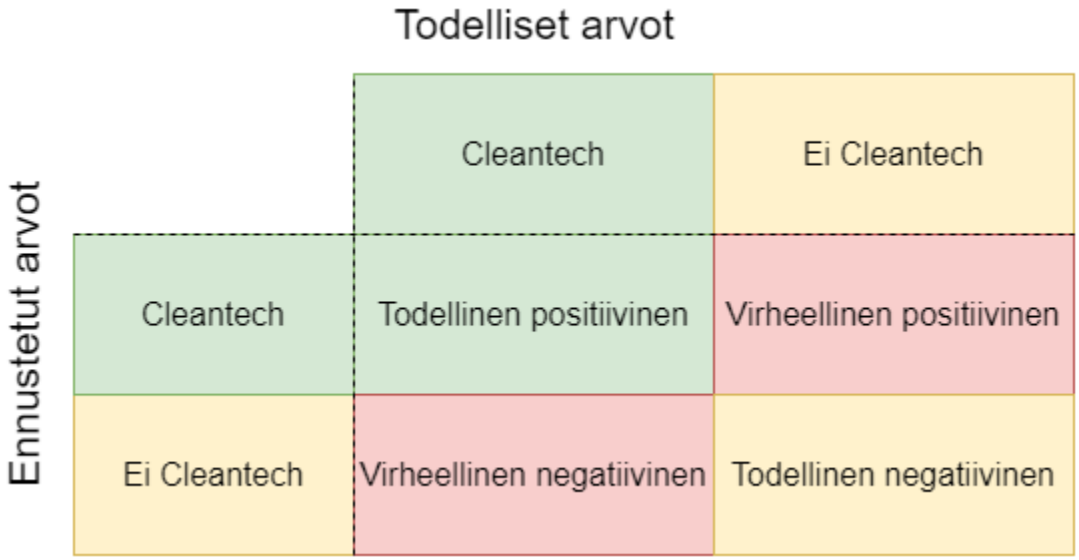


Kuva 12: LSI-vektoreita hyödyntävän kNN-luokittelumallin toiminta

Kuva havainnollistaa LSI-vektoreilla toteutettua ohjatun oppimisen prosessia, jossa luokittelumallia opetettiin opetusaineistolla. Opetuksen jälkeen luokittelumallin avulla luokiteltiin LSI-vektoroitu testiaineisto, jonka perusteella luokittelijan toimintaa arvioitiin.

4.3.3 Mallien arviointi

Mallien vertailu toteutettiin hyödyntämällä luvussa 3.2.3. esiteltyjä metriikoita. Tätä varten kirjattiin kunkin testiaineiston alkion ja luokittelumallin sille luoman ennusteen perusteella totuusarvo. Näitä arvoja ovat todellinen positiivinen, todellinen negatiivinen, virheellinen positiivinen ja virheellinen negatiivinen arvo. Näiden arvojen muodostaminen cleantech-luokasta on esitetty sekaannusmatriisissa kuvassa 13.



Kuva 13: Tutkimuksessa käytetty sekaannusmatriisi

Jokaiselle mallille laskettiin sekaannusmatriisin mukaiset totuusarvot. Näiden avulla voidaan laskea luvussa 3.2.3. esitettyjen kaavojen tunnusluvut kokonaistarkkuus, tarkkuus, saanti ja F1-arvo. Näiden lisäksi jokaiselle mallille laskettiin ROC-käyrän alle jäävä pinta-ala (AUC), sekä tarkkuus-saanti-käyrän alle jäävä pinta-ala (PR). Julkisiin kuvauksiin perustuva sekaannusmatriisi on esitetty liitteessä 2.

5. TULOKSET

Tässä luvussa on esitetty luokittelumallien tunnusluvut sekä arvioitu niiden eri ominaisuuksia tulosten valossa. Taulukossa 5 on kuvattu jokaisen mallin testattavien alkioiden määrä N, todellisten positiivisten arvojen määrä (TP), todellisten negatiivisten arvojen määrä (TN), virheellisten positiivisten määrä (FP) ja virheellisten negatiivisten määrä. Tämän lisäksi edellä mainittujen luokitusten perusteella on laskettu kaavojen 4–8 mukaiset arvot: kokonaistarkkuus (ACC), tarkkuus (precision) P, saanti (recall) R, F1-arvo (F1) ja ROC-käyrän alle jäävän pinta-ala AUC sekä tarkkuus-saanti-käyrän alle jäävä pinta-ala, PR.

Taulukko 5: Luokittelumallien totuusarvot ja tunnusluvut

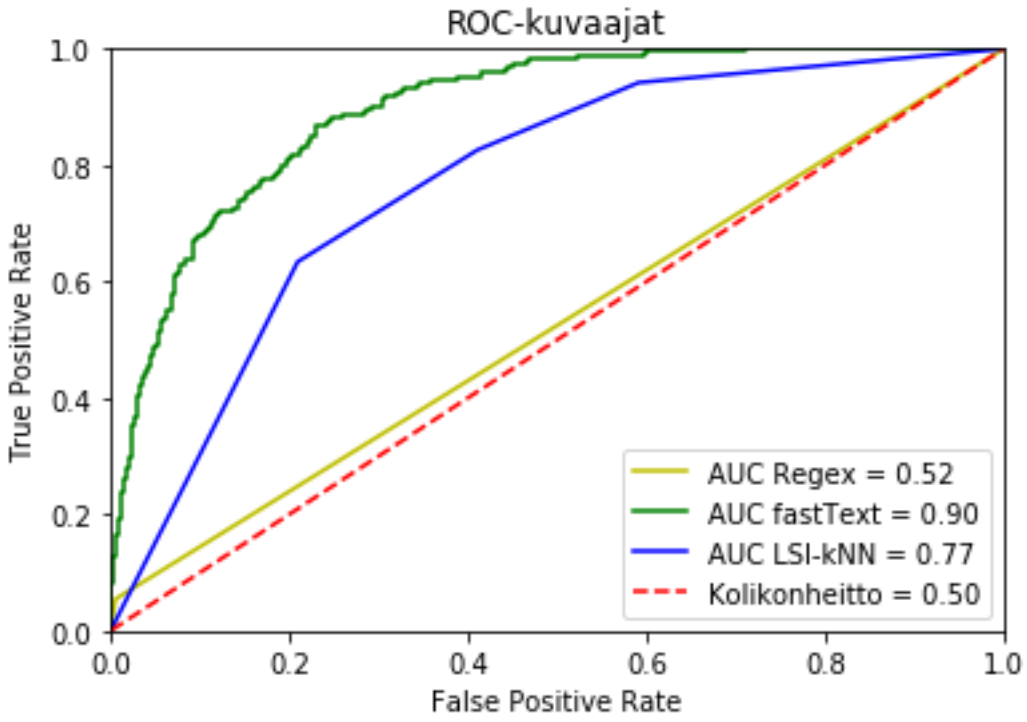
Luokittelija	N	TP	TN	FP	FN	P	R	F1	ACC	AUC	PR
Regex	6746	70	5426	29	1221	0,707	0,054	0,101	0,815	0,524	0,471
Regex (suhteutettu)	1349	14	1085	6	244	0,700	0,054	0,101	0,815	0,524	0,471
fastText	1350	131	1024	50	145	0,724	0,475	0,573	0,856	0,896	0,692
LSI-kNN	1350	175	849	225	101	0,438	0,634	0,518	0,759	0,770	0,581

Taulukosta havaitaan, että säännöllisiin lausekkeisiin (regex) perustuvan luokittelumallin kokonaistarkkuus on n. 81 %. Koska kokonaistarkkuus on epätasaisella aineistolla harhaanjohtava mittari, tutkitaan sen lisäksi myös tarkkuutta ja saantia. Tarkkuus itsessään näyttää vertailukelpoiselta, mutta on tärkeä huomata, että mallin saanti on vastaavasti todella heikko. Tämä tarkoittaa, että kaikista mallin ennustamista positiivisista tuloksista n. 70 % on relevantteja, joskin se luokittelee onnistuneesti vain n. 5 % kaikista positiivisista alkioista. Tämän takia myös tarkkuuden ja saannin geometrinen keskiarvo on erittäin heikko 0.1.

FastText-luokittelumallin kokonaistarkkuus on n. 86 %, joka on hieman säännöllisten lausekkeiden kokonaistarkkuutta parempi. Malli löytää saannin perusteella hieman alle 50 % kaikista positiivisista tuloksista ja positiivisiksi luokitelluista tuloksista on tarkkuuden perusteella relevantteja n. 72 %. Tarkkuuden ja saannin geometrinen keskiarvo F1 on n. 57 %.

LSI-kNN-mallin kokonaistarkkuus on hyvin lähellä säännöllisten lausekkeiden mallia ja fastText-mallia, n. 76%. Konkreettisin eroavaisuus tulee tarkkuuden ja saannin painotuksesta, sillä muista malleista poiketen, LSI-kNN-malli painottaa saantia tarkkuuden sijaan. Tämä tarkoittaa sitä, että malli löytää 63% kaikista positiivisista tuloksista, joskin vain n. 44 % kaikista positiivisista luokituksista ovat relevatteja. Keskiarvo F1 on hieman fastText-luokittelijaa heikompi, joskin merkittävästi parempi kuin säännöllisiin lausekkeisiin perustuvalla luokittelumallilla.

Koska tutkimusaineistossa luokissa on epätasapainoinen määrä alkioita, on tarpeen kuvata luokittelumalleja myös kattavampien mittarien avulla. Kuvassa 14 on esitetty luokittelumallien ROC-käyrä. ROC-käyrät kuvaavat kunkin mallin todellisten positiivisten asteen ja virheellisten positiivisten asteen suhdetta, kun luokittelijalle asetetaan eri kynnysarvoja (ks. kuva 7) Kuvan 14 selitteessä on esitetty käyrän alle jäävä pinta-ala (AUC) jokaisesta luokittelumallista.



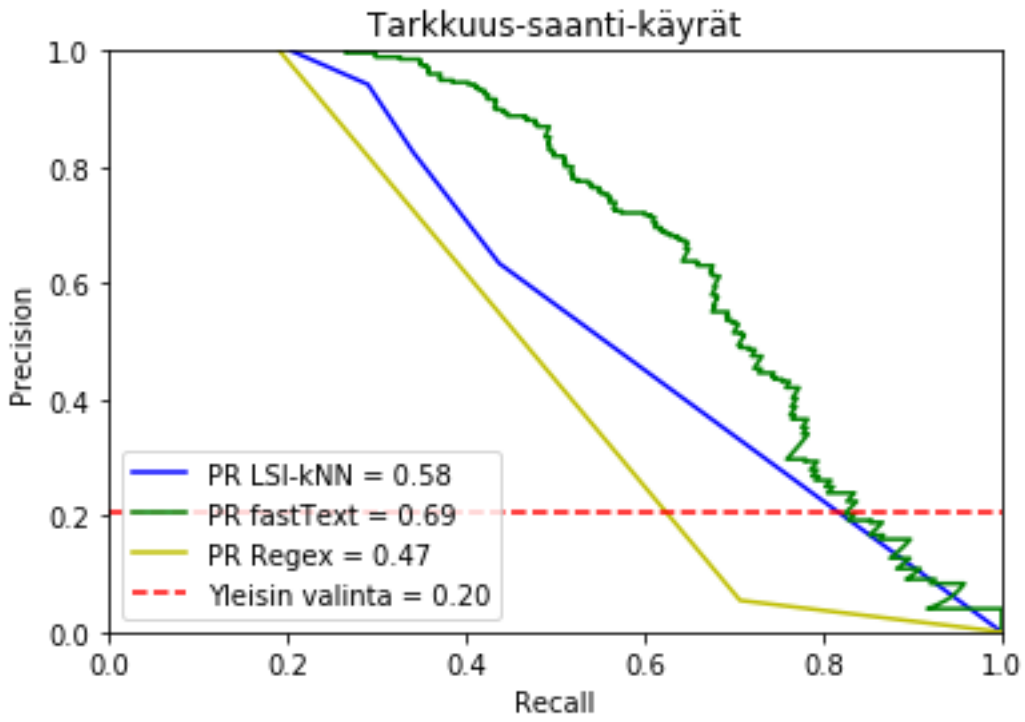
Kuva 14: Luokittelumallien ROC-käyrät ja käyrien alle jäävät pinta-alat (AUC) [8]

[8] Mukailtu https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/ (viitattu 2.5.2020)

Kuvaajasta on havaittavissa, että kaikkien mallien osalta todellisten positiivisten ja virheellisten positiivisten arvojen suhde on parempi kuin satunnaismuuttujalla. Säännölliseen lausekkeeseen perustuva luokittelija (keltainen) on hyvin lähellä sattumanvaraista luokittelijaa, jota tämän tutkimuksen kontekstissa on havainnollistettu kolikonheittomallilla.

FastText-malli (vihreä) tuottaa malleista parhaan tuloksen, sillä se saavuttaa arvoja, jotka ovat lähimpänä vasenta yläkulmaa, joka kuvaa täydellistä luokittelijaa (ks. kuva 8). Käyrän alle jäävän pinta-ala on n. 0.9. LSI-kNN-malli (sininen) saa merkittävästi parempia tuloksia, kuin satunnaismuuttuja, joskaan ei yhtä hyviä kuin fastText-malli. LSI-kNN-mallin käyrän alle jäävä pinta-ala AUC on n. 0.77.

Kuten luvussa 3.3. todetaan, ROC-käyrä voi epätasaisella opetusaineistolla tuottaa optimistisia tuloksia. Kuvassa 15 on piirretty luokittelumallien tarkkuus-saanti-käyrä ja selitteessä jokaisen mallin käyrän alle jäävä pinta-ala. Tarkkuus-saanti-käyrää voidaan käyttää validoimaan ROC-käyrän tuloksia.



Kuva 15: Tarkkuus-saanti-käyrät ja käyrien alle jäävät pinta-alat (PR) [9]

[9] Mukailtu https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/ (viitattu 2.5.2020)

Yleisin valinta kuvaa tarkkuus-saanti-käyrällä esitettynä positiivisen luokan valinnan tilastollista todennäköisyyttä, eli yleisintä valintaa. Pinta-alaan PR verraten voidaan huomata yleiseen valintaan perustuvan luokittelijan olevan muita vaihtoehtoja merkittävästi heikompi. ROC-käyrän tuloksia mukaillen, fastText-luokittelumallin alle jäävä pinta-ala on suurempi kuin LSI-kNN -mallin, joskin molemmat saavat merkittävästi parempia tuloksia, kuin säännölliseen lausekkeeseen perustuva luokittelumalli.

6. YHTEENVETO JA PÄÄTELMÄT

Tutkimuksen tavoitteena oli löytää ja vertailla soveltavan kokeellisen tutkimuksen keinoin NLP-tekniikoita ja menetelmiä, joiden avulla voidaan mahdollisimman hyvin tunnistaa ja luokitella cleantech-hankkeisiin kohdistuvia rahoitushakemuksia kaikista rahoitushakemuksista, joille cleantech-luokitus on mahdollinen. Tätä varten rakennettiin kolme päämallia, joiden lisäksi malleja verrattiin suhteessa ulkoisiin luokittelijoihin, kuten satunnaisluokittelijana toimivaan ”kolikonheittoon” ja tilastolliseen todennäköisyyteen perustuvaan yleisimpään valintaan. Tässä luvussa on esitelty tutkimuksen yhteenveto, joka sisältää kirjallisuusosion ja empiirisen osion, Tämän lisäksi luku sisältää tutkimuskysymyksiin vastaamisen, sekä kirjallisuuden että empiiristen tulosten avulla, tulosten käytännön vaikutusten kuvaamisen, tutkimuksen arvioinnin ja jatkotutkimuskohteet.

6.1 Empiirisen tutkimusosion yhteenveto

Tutkimuksen valikoitui säännönmukaiseen lauseeseen perustuva malli, ohjattu fastText-luokittelumalli, sekä puoliohjattu LSI-vektoreihin perustuva kNN-luokittelumalli. NLP-prosessin mukaisesti aineistolle toteutettiin perusteellinen esikäsittely ennen luokittelua. Tutkimusaineisto muokattiin pisteillä erotettuun raakatekstiin fastText-opetusta varten. LSI-aihemallinnuksen esikäsittelyssä toteutettiin piirteiden erottaminen muuntamalla tekstimuotoinen data vektorimuotoon, josta edelleen sen avulla muodostettiin ohjaamattoman oppimisen aihemalli. Aihemallin avulla toteutettiin ulottuvuuksien vähentäminen opetusaineistolle, jonka avulla dokumentteja kuvaavat vektorit saatiin vertailukelpoiseen muotoon samalla säilyttäen niiden sisältämä semanttinen informaatio.

Esikäsittelyn jälkeen aineisto luokiteltiin valikoituja luokittelijoita käyttäen. Säännönmukaisella lausekkeella luokittelu toteutettiin hakemalla tekstijonoa ’cleantech’. FastText ja LSI-kNN-luokittelijoiden tapauksessa aineisto jaettiin opetus- ja testiaineistoon, joiden avulla opetettiin kNN-luokittelualgoritmia. Luokittelun tulosta arvioitiin jälkimmäisten mallien tapauksessa testiaineiston perusteella.

Luokittelun tulosten perusteella muodostettiin sekaannusmatriisi, joka jakaa luokittelun tulokset totuusarvonsa mukaisesti todelliseen positiiviseen, todelliseen negatiiviseen, virheelliseen negatiiviseen ja virheelliseen positiiviseen tulokseen. Näiden perusteella

laskettiin malleja kuvaavat tunnusluvut, joiden perusteella arvioitiin luokittelijoiden soveltuvuutta cleantech-hankkeiden luokitteluun.

6.2 Tulosten yhteenveto

Lähes kaikilla mittareilla arvioituna malleista parhaiten toimi fastText-malli. LSI-kNN-malli tuotti hieman heikompia tuloksia kaikilla muilla osa-alueilla paitsi saannin suhteen. Huomionarvoista onkin mallien painotus tarkkuuden ja saannin suhteen: fastText-mallin tarkkuus on huomattavasti vahvempi. LSI-kNN-malli puolestaan suosii saantia tarkkuuden kustannuksella. Myöskään fastText- ja LSI-kNN-mallien välillä ei ole suurta eroa tarkkuuden ja saannin geometrisessa keskiarvossa, eli F1-arvossa. Säännölliseen lausekkeeseen perustuvan mallin tarkkuus on korkea, joskin muiden mittareiden perusteella voidaan päätellä, ettei se sovellu luotettavaksi luokittelijaksi.

Tarkkuuden ja saannin arvoja tulkitessa on olennaista tuntea tutkimuskohteen viitekehys, jolloin voidaan päättää, kumpaa mittaria on suosittava toisen kustannuksella. Mikäli cleantech-hankkeiden luotettava ja varma tunnistaminen on kriittinen tekijä, eivätkä virheellisten positiivisten kustannukset ole haitalliset, LSI-kNN-malli sopii tarkoitukseen. FastText soveltuu tarkoitukseen paremmin, mikäli on tärkeää tunnistaa mahdollisimman suuri osa relevantteja tuloksia positiiviseksi luokitelluista alkioista, ja virheellisten negatiivisten tulosten määrä tutkimuskohteelle on pieni.

ROC-käyrän alle jäävän pinta-ala (AUC) osoittaa, että säännönmukaiseen lausekkeeseen perustuvan mallin suorituskyky on hyvin lähellä satunnaismuuttujaa eli kolikonheittoa. FastText-mallin ja LSI-kNN suhteen on nähtävissä, että molemmat onnistuvat luokittelemaan testiaineistoa merkittävästi paremmin kuin satunnaismuuttuja, joista FastText-malli on kuitenkin selkeimmin lähimpänä täydellistä luokittelijaa.

On kuitenkin todettava, että koska ROC-käyrän ominaisuuksiin liittyy optimismi epätasaisella tutkimusaineistolla, on tuloksia validoitava tarkkuus-saanti-käyrän alle jäävän pinta-alan (PR) avulla. Myös PR-arvon perusteella fastText-malli tuottaa parhaita tuloksia valituista malleista, joka vahvistaa ROC-käyrältä havaittuja tuloksia.

6.3 Tutkimuskysymyksiin vastaaminen

Diplomityön tutkimusongelma ”*Rahoitushakemusten luokittelu luokkaan cleantech*” on lähtöisin Business Finlandin tarpeesta luokitella rahoitushakemuksia luotettavasti etukäteen määriteltyihin luokkiin. Vastaavasti tutkimuksen päätutkimuskysymys on ”*Miten hakemustekstiä voidaan luokitella luonnollisen kielen prosessoinnin (NLP) avulla?”*. Tämä on edelleen jaettu apututkimuskysymyksiin, joihin on vastattu tässä luvussa.

Ensimmäinen apututkimuskysymys on ”*Mitä on NLP tämän tutkimuksen kontekstissa?*”. Tutkimuksessa tunnistetaan NLP:n olevan hyvin laaja kattokäsite teknologioita ja menetelmiä, joilla voidaan lukea, käsitellä ja tuottaa luonnollista kieltä. Tässä tutkimuksessa tutkitaan NLP:a koneoppimisen teknologioina, joiden avulla voidaan käsitellä luonnollista kieltä. Tässä tutkimuksessa NLP-tekniikoita käytetään sekä esikäsittelyprosessissa että rahoitushakemusten luokittelussa. Esikäsittelyvaiheessa toteutetaan piirteiden erottaminen ja ulottuvuuksien vähentäminen ohjaamattoman LSI-mallin avulla, jonka muodostamien semanttisten dokumenttivektorien avulla opetetaan k-Nearest-Neighbor-luokittelija. Luokittelussa tutkitaan lisäksi neuroverkkopohjaista fastText-mallia luokittelemaan rahoitushakemuksia sanansisäistä informaatiota hyödyntäen.

Toinen apututkimuskysymys on ”*Miten valittujen NLP-tekniikoiden ominaisuuksia voidaan verrata toisiinsa?*” Tutkimuksessa havaitaan useita vertailukelpoisia menetelmiä ja metriikoita, joiden avulla mallien toimintaa voidaan arvioida ja verrata toisiinsa. Jokainen malli pisteytetään sekaannusmatriisin avulla todellisiin positiivisiin, todellisiin negatiivisiin, virheellisiin positiivisiin ja virheellisiin negatiivisiin tuloksiin, joiden perusteella voidaan laskea mallin toimintaa kuvaavia tunnuslukuja. Mallit arvioidaan ensin kokonaistarkkuuden, tarkkuuden, saannin ja näiden geometrisen keskiarvon, F1-arvon avulla.

Koska tutkimusaineistossa on epätasapainoa, on myös syytä valita mittarit, jotka ottavat tämän huomioon. Tähän tarkoitukseen hyödynnetään ROC-käyrää ja sen alle jäävää pinta-alaa (AUC). ROC- ja AUC-metriikoita pyritään validoimaan käyttämällä lisäksi tarkkuus-saanti-käyrää ja tämän alle jäävää pinta-alaa (PR). Useaa metriikkaa käyttämällä saadaan muodostettua kattava, monipuolinen ja vertailukelpoinen joukko mittareita, joiden avulla NLP-tekniikoita voidaan verrata toisiinsa. Tämän lisäksi joitakin metriikoita voidaan hyödyntää luokittelijoiden optimoinnissa. Esimerkiksi ROC-käyrän

kynnysarvon avulla voidaan optimoida todellisten positiivisten asteen ja virheellisten positiivisten asteen suhdetta.

Kolmas ja viimeinen apututkimuskysymys on. *”Mitä esikäsittelytoimenpiteitä hakemustekstille on tehtävä NLP-prosessia varten?”*. Ennen koneoppimista tai analyysiä on tekstimuotoiselle datalle toteutettava korpuksen esikäsittely. Korpuksen esikäsittely pitää sisällään tekstin esikäsittelyn, piirteiden erottamisen ja ulottuvuuksien vähentämisen.

Tekstin esikäsittelyssä poistetaan erikoismerkit, numerot, symbolit ja hukkasanat datasta, jotta vain merkitykselliset sanat jäävät jäljelle. Tämän jälkeen tekstidatalle tehdään piirteiden erottaminen, jossa tekstimuotoinen data muunnetaan numeraaliseen vektorimuotoon sanojen esiintymistiheyksien perusteella ja niille lasketaan TF-IDF-arvo. Näin ollen jokaista tutkimusaineiston sisältämää dokumenttia voidaan kuvata TD-IDF-vektorin avulla.

Ulottuvuuksien vähentämisessä hyödynnetään Latent Semantic Indexing menetelmää, jonka avulla voidaan tiivistää kompleksiset TF-IDF-vektorit yhteiseen, vertailukelpoiseen LSI-vektoriavaruuteen. LSI-vektoriavaruuteen tiivistetyt dokumenttivektorit pyrkivät esittämään dokumentin semanttisen merkityksen riippumatta syntaksista. Näin ollen dokumentteja voidaan vertailla semanttisen samankaltaisuutensa perusteella.

Käytännön kannalta empiirisen tutkimusosion mallit vaativat eri laajuisen esikäsittelyn. Säännöllisiin lausekkeisiin perustuvassa mallissa esikäsittelyä ei tarvita lainkaan. Ohjatun oppimisen fastText-mallille toteutetaan tekstin esikäsittely itse, mutta piirteiden erottaminen ja ulottuvuuksien vähentäminen tapahtuu Python-paketin sisäisissä toiminnoissa CBOW-menetelmän avulla. LSI-kNN-mallia varten toteutetaan kaikki korpuksen esikäsittelyn vaiheet

6.4 Käytännön vaikutukset

Tutkimuksen tavoitteena oli muodostaa vertaileva tutkimus valituista luonnollisen kielen käsittelyn tekniikoista, joita voidaan käyttää cleantech-rahoitushakemusten luokitteluun. Cleantech-hakemusten koneellista luokittelua voidaan käyttää tukemaan Business Finlandin työntekijöiden hakemusten luokitteluprosessia. Valittujen luokittelumallien teoreettisen tutkimuksen ja empiirisen vertailun tuloksien perusteella voidaan muodostaa käytäntöön vaikuttavia johtopäätöksiä.

Luokittelumalleja kuvaavien tunnuslukujen ominaisuudet täytyy ottaa huomioon valittaessa mallia käytännön kontekstiin, kuten esimerkiksi cleantech-hankkeiden luokitteluun. Tarkkuus kuvaa luokittelijan kykyä tunnistaa relevantit cleantech-rahoitushakemukset kaikista cleantech-luokitelluista rahoitushakemuksista (Jiang 2013 s. 33). Mikäli tarkkuus on korkea, voidaan todeta, että virheellisesti cleantech-hakemukseksi luokiteltuja hakemuksia on vain vähän. Tarkkuuden haittapuolena on, että se ei arvioi niitä oikeita cleantech-hakemuksia, jotka luokitellaan virheellisesti luokkaan ’ei cleantech’ (Johnson & Khoshgoftaar 2019). Toisin sanoen riskinä on, että korkeaa tarkkuutta painottava luokittelija voi jättää jonkin relevantin hakemuksen huomioimatta.

Saantiin painottuva malli puolestaan luokittelee luotettavasti suurimman osan oikeista cleantech-hakemuksista luokkaan ’cleantech’. Saantiin painottuva malli kuitenkin korostaa luotettavuutta niin paljon, että se luokittelee ’cleantech’ luokkaan myös rahoitushakemuksia, jotka eivät sinne kuulu. Toisin sanoen, riski että cleantech-hakemus jää tunnistamatta on pieni, mutta samalla ’cleantech’-luokkaan luokitellaan myös sinne kuulumattomia rahoitushakemuksia (Johnson & Khoshgoftaar 2019). Tarkkuuden ja saannin painottamisesta voidaan todeta, että mallia valittaessa tulee arvioida kumpi näistä skenaarioista aiheuttaa vähemmän haittaa siinä kontekstissa, jossa luokituksia tehdään.

Business Finlandin kontekstissa virheelliset positiiviset tulokset voivat potentiaalisesti aiheuttaa kustannuksia, sillä ne suosittelevat cleantech-luokituksen asettamista myös sellaisille hankkeille, jotka eivät vastaa cleantech-luokiteltavan hankkeen määritelmää. Tästä syystä tarkkuutta korostavat luokittelijat soveltuvat tässä kontekstissa luokitteluun paremmin kuin saantia korostavat luokittelijat. Tutkimuksen metriikoilla arvioituna parhaiten suoriutuva fastText-luokittelumalli korostaa myös tarkkuutta saannin yli, joten sitä voidaan suositella käytettäväksi rahoitushankkeiden cleantech-luokitteluun.

Tekstin esikäsittelyllä, piirteiden erottamisella ja ulottuvuuksien vähentämisellä voidaan muokata rahoitushakemustekstejä laskettavaan ja vertailukelpoiseen LSI-vektorimuotoon (Wajeed & Adilakshmi 2011; Novotný & Ircing 2017; Mirończuk & Protasiewicz 2018). Tämä mahdollistaa dokumenttien matemaattisen vertaamisen sisältönsä ja merkityksensä perusteella (Zelikovitz & Marquez 2005). Rahoitusdokumenttien semanttisia esityksiä voidaan hyödyntää esimerkiksi erilaisissa tekstianalyysin tutkimuskohteissa tai dokumenttien semanttisen samankaltaisuuden vertaamisessa. Esikäsittelytoimenpiteiden lisäksi aineiston luokkaepätasapainon pienentäminen ehkäisee luokittelijan taipumusta suosia yleisempää luokkaa ja näin ollen tuottaa luotettavampia luokittelutuloksia (Ma et al. 2018).

Tutkimusten suorituksessa havaittiin myös seikkoja, jotka suoraan eivät osuneet tämän tutkimuksen rajaukseen. Käytännön kontekstissa nämä havainnot kuitenkin auttavat ymmärtämään tutkimusaineistoa. Esimerkiksi semanttisia vektoreita varten muodostetun LSI-aihemallin tunnistamista aiheista havaittiin kohdeyritykselle huomionarvoisia seikkoja. Yksi aihe koostui suurimmaksi osaksi englanninkielisistä sanoista. Tämä tarkoittaa joko sitä, että aineistoa rajaava kielentunnistus ei toimi täydellisesti, tai sitä, että jotkin rahoitushakemukset sisältävät tekstiä useammalla kielellä. Toinen huomionarvoinen aihe sisälsi sanoja kuten ”ks.”, tai ”liite”. Tämä voi tarkoittaa mm. sitä, että rahoitushakemuksen tekstistä osa on erillisessä liitteessä, eikä luokitukseen näin ollen voida hyödyntää kaikkea informaatiota, jota rahoitushakemusteksti sisältää.

Tutkimuksesta käy myös ilmi, että luokittelumalleja voidaan ennestään parantaa hyödyntämällä ROC-käyriä optimaalisen kynnysarvon valinnassa. Kynnysarvon optimoinnilla voidaan vaikuttaa sekä luokittelun tarkkuuden parantamiseen ja mm. siihen painottaako luokittelumalli saantia vai tarkkuutta. (Krzanowski 2009 ss. 18–19) ROC-käyrän hyödyntäminen mahdollistaa myös mallin arvioinnin huomattavasti kattavammin kuin mitä voidaan esittää yksiarvoisilla mittareilla.

6.5 Tutkimuksen arviointi

Tutkimuksen luotettavuutta voidaan arvioida *validiteetin* (validity), *reliabiliteetin* (reliability) sekä *yleistettävyyden* (generalizability) avulla (Saunders et al. 2009 ss.156-158). Guba (1981 ss.80-81) lisää arviointinäkökulmaksi myös *objektiivisuuden* (objectivity). Edellä mainittujen kriteerien lisäksi tutkimusta on arvioitu tietotekniikassa yleisesti käytetyn *toistettavuuden* (reproducibility) perusteella (Peng 2011).

Validiteetti, tarkastelee sitä, onko tutkimuksessa mitattu niitä asioita, joita siinä oli alun perin tarkoitus mitata (Shenton 2004). Toisin sanoen validiteetti tarkastelee, vallitseeko mitattavien ilmiöiden välillä kausaliteetti (Saunders et al. 2009 s.157) Vallitsevan positivistisen tieteenfilosofian nojalla arvioidaan onko tutkimuksen tulos totuus (Guba 1981 s.79). Tässä tutkimuksessa tutkittiin, miten rahoitushakemuksia voidaan luokitella NLP:n keinoin. Tutkimus on siis validi, mikäli tutkimuksen metriikat näyttävät toteen, eli mikäli löytyy luokittelija, joka onnistuu tutkimushankkeiden luokittelussa. Tutkimuksen tuloksena voidaan todeta, että luokitteluun soveltuvia esikäsittelytoimenpiteitä ja luokittelijoita onnistuttiin löytämään. Kuitenkin relevanttien luokittelijoiden löytämisessä oli validiteetin kannalta haasteita. Tutkimuksessa ilmeni, että tutkimusaineiston

luokkaepätasapainon vuoksi esimerkiksi kNN-luokittelijan käyttö ei ollut optimaalinen valinta. Tutkimuksessa kuitenkin päädyttiin aineiston tasapainotukseen algoritmipohjaisten epätasapainon korjaustoimenpiteiden sijaan.

Reliabiliteetti kuvaa, että tuottaako tutkimus johdonmukaisia tuloksia ja onko tutkimus toistettavissa joko alkuperäisen tutkijan, tai ulkopuolisen tutkijan toimesta. (Saunders et al. 2009 s.156) Tämä tarkoittaa, että mikäli tutkimus toistetaan samassa ympäristössä, samoilla menetelmillä ja samoilla osallistujilla, tutkimuksen tulokset ovat alkuperäisen kaltaiset. Tätä voidaan edistää mm. tutkimussuunnitelman, toteutuksen ja aineiston keruun huolellisella raportoinnilla. Tämän lisäksi on oleellista arvioida tutkimusprojektin toimivuutta (Shenton 2004 ss.71-72). Tämän tutkimuksen suunnitelma, toteutus ja aineiston keruu on kuvattu sekä tähän tutkimusdokumenttiin että empiirisen osion osalta kooditiedostoon. Näin ollen tutkimuksen kaikki vaiheet sekä tutkimuksen arviointi ovat eksplisiittisesti saatavilla, mikä noudattaa korkean reliabiliteetin periaatteita.

On kuitenkin huomattava, että empiirisessä tutkimusosiossa on joitain komponentteja, jotka aiheuttavat tuloksiin variaatiota jokaisella suorituskerralla. Esimerkiksi kielentunnistus arvioi tekstien kieltä todennäköisyyden perusteella, mikä voi johtaa siihen, että tekstit, joissa esiintyy useita eri kieliä, voidaan eri suorituskerroilla tulkita sellaiseksi kieleksi, jota tässä tutkimuksessa ei oteta huomioon. Myös mallien opetuksessa tapahtuva jako opetus- ja testiaineistoon toteutetaan satunnaisuuteen perustuvalla menetelmällä, mikä voi johtaa erilaisiin luokitusten osuuksiin opetus- ja testiaineistossa. Vaikka luokkien tasapaino ei merkittävästi muutu opetus- ja testidatassa satunnaisuuden vuoksi, vaikuttaisi siltä, että näin pienellä tutkimusaineistolla jo hyvin vähäisellä määrällä dokumentteja on merkittävä vaikutus. Tämä seikka laskee tutkimuksen reliabiliteettia, ja jatkotutkimuksessa olisikin syytä haarukoida relevantti satunnaisotanta, jossa luokkien suhde pysyy vakiona.

Yleistettävyys arvioi, onko tutkimus sovellettavissa eri tilanteisiin, esimerkiksi uudella aineistolla. (Shenton 2004 s.69) Yleistettävyyttä voidaan arvioida tutkimalla, soveltuuko tutkimus esimerkiksi jonkin toisen organisaation kontekstiin (Saunders et al. 2009 s.158). Positivistisen tieteenfilosofian mukaisesti tutkitaan siis, onko tutkimuksen tulos tosi missä tahansa kontekstissa (Guba 1981 s.80) Tämä tutkimus on kohdistettu Business Finlandin tutkimusongelmaan, jossa rahoitushakemus on selkeästi määrämuotoinen. Esikäsittelyn ja luokituksen teoria ja empiirinen tutkimus ovat kuitenkin yleistettävissä myös mille tahansa tekstimuotoiselle datalle, jolle on olemassa ennalta määrätty luokitus. Tutkimuksen empiirinen osio on siis yleistettävissä, mutta tulokset vaihtelevat aineiston perusteella.

Objektiivisuudella tarkoitetaan, että tutkimukseen ei vaikuta mikään tutkimuksen ulkopuolinen tekijä, kuten esimerkiksi tutkijan subjektiiviset näkemykset (Guba 1981 s.80). Objektiivisuutta voidaan edistää esimerkiksi käyttämällä tutkimusmetodologioita tai välineitä, jotka eivät ole riippuvaisia ihmisen taidosta tai havaintokyvystä (Shenton 2004). Tämän tutkimuksen positivistisiin perusoletuksiin kuuluu, että ulkoiset tekijät eivät vaikuta tutkimuksen suoritukseen, aineistoon tai tuloksiin. Lisäksi esimerkiksi empiirinen tutkimusosio perustuu Python-ohjelmointikielellä rakennettuun kooditiedostoon, jolloin käytössä on tutkimusmenetelmä, jonka tuloksiin tutkijan subjektiivinen näkemys ei voi vaikuttaa. On kuitenkin todettava, että esimerkiksi tutkittavien mallien valintaa ei voida perustella objektiivisilla menetelmillä, vaan valinnat on tehty subjektiivisen näkemyksen perusteella.

Toistettavuus perustuu tutkimuksen empirian taustalla olevan koodin ja datan avoimuuteen ja toistettavuuteen. Tutkimus on parhaalla toistettavuuden tasolla, kun sen yhteydessä on saatavilla koodi ja data, jolla alkuperäisen tutkimuksen tulokseen on päästy. Vastaavasti voidaan sanoa, ettei tutkimus ole toistettava, mikäli tutkimuksesta julkaistaan vain tutkimusraportti. Tutkimuksen voidaan sanoa olevan osittain toistettava, mikäli sen yhteydessä julkaistaan minimissään tutkimuksen empiirisessä osiossa käytetty koodi. (Peng 2011) Tämä tutkimus yltää tietyllä tarkkuudella toistettavuuden määritelmään, sillä tutkimuksen tuottama koodi julkaistaan tutkimuksen julkaisun yhteydessä. Koodin julkaisu ei kuitenkaan tule sisältämään käytetyn datan julkaisua ja koodista myös poistetaan kaikki kohdeyritykseen viittaavat piirteet, jolloin jäljelle jää mille tahansa tekstimuotoiselle datalle yleistettävä tutkimusrakenne. Tämän vuoksi voidaan sanoa tutkimuksen olevan toistettavuuden asteikolla, mutta toistettavuus on hyvin matala.

Positivistinen tieteenfilosofia soveltuu mallien vertaamiseen, kun määritetään totuutta siitä, onko malli soveltuva luokittelijaksi vai ei. Tutkimuksessa kuitenkin havaitaan, ettei mallia voida yksiselitteisesti matemaattisten metriikoiden perusteella arvioida soveltuvaksi luokittelijaksi, sillä esimerkiksi tarkkuuden ja saannin välillä muodostuu vaihtokauppa, jolloin sovellettava ongelma määrittää, mikä malleista soveltuu parhaiten luokittelijaksi. Positivistisen paradigman vastaisesti siis totuuden määrittäminen ei välttämättä onnistu yksiselitteisesti, vaan pragmaattinen, tutkimusongelmakeskeinen ote määrittää, missä tilanteessa kukin malli on sovellettavissa luokitteluun.

6.6 Jatkotutkimuskohteet

Tutkimuksen aikana heräsi runsaasti jatkotutkimuskohteita, jotka eivät mahtuneet tämän diplomityön kontekstiin. Jatkotutkimuskohteet painottuvat empiiriseen tutkimusosioon ja varsinkin tutkimusaineiston ominaisuuksiin.

Tutkimusongelma on erittäin mielenkiintoinen ja on perusteltua tutkia ongelmaa myös laajemmalla aineistolla. Koska tutkimus on toistettavissa millä tahansa tekstiaineistolla, voitaisiin luokittelijoiden tarkkuutta verrata suhteessa aineiston kokoon sekä aineiston luokkien väliseen tasapainoon. On myös mahdollista, että laajempi aineistokoko myös muodostaisi kattavamman sanaston, jolloin suomen kielen monet sijapäätteet tulevat paremmin edustetuiksi yksittäisten sanojen frekvenssiä laskettaessa, ja näin ollen tuottavat rikkaampia vektorimuotoja.

Koska Business Finlandin hakemuksia voi hakea useammalla kielellä, olisi mielekästä yleistää myös luokittelu useammille kielille. Jokainen kieli vaatii kuitenkin erilaisen esikäsittelyn sekä oman luokittelijan ja yksilöllisen esikäsittelyn.

Tutkimuksen koneoppimiseen pohjautuvat luokittelualgoritmit rajoittuvat kahteen tutkittuun malliin. Koska esimerkiksi kNN-malli kärsii luokkien välisestä epätasapainosta, olisi mielekästä toistaa tutkimus käyttämällä vaihtoehtoisia luokittelualgoritmeja, kuten esimerkiksi Naive Bayes tai Support Vector Machine -pohjaisilla algoritmeilla (Xu et al. 2014).

Luokittelijan yksityiskohtainen optimointi esimerkiksi luokittelijoiden parametrien ja kynnysarvon valinnan avulla olisi mielekäs kohde tapaustutkimukselle. ROC-käyrää voidaan hyödyntää optimaalisen kynnysarvon valinnassa (Krzanowski 2009 s.19). Tässä tilanteessa tulisi perehtyä tarkasti luokituksen vaikutuksiin tutkimuskohteelle, jolloin voitaisiin luoda tietoisiin valintoihin perustuva kynnysarvon valinta.

LÄHTEET

Aggarwal, C. C. (2015). Data classification : algorithms and applications, Boca Raton, Florida: CRC Press, Taylor & Francis Group.

Aggarwal, C. C. (2018). Machine learning for text, Cham, Switzerland: Springer. Saatavissa: https://andor.tuni.fi/permalink/358FIN_TAMPO/i4qs1o/alma9910393524205973

Aggarwal, C. C. & Zhai, C. (2012). A survey of text classification algorithms, , Vol. 9781461432234, Springer US, s. 222.

Altınel, B., Can Ganiz, M. & Diri, B. (2015). A corpus-based semantic kernel for text classification by using meaning values of terms, Engineering Applications of Artificial Intelligence, Vol. 43, pp. 54–66.

Altınel, B. & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances, Information Processing and Management, Vol. 54(6), pp. 1129–1153.

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2016, August 18). fastText. Saatavissa: https://research.fb.com/blog/2016/08/fasttext/

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017). Enriching Word Vectors with Subword Information, ArXiv:1607.04606 [Cs]. Saatavissa: http://arxiv.org/abs/1607.04606

Boyd, K., Eng, K. H. & Page, C. D. (2013). Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals, H. Blockeel, K. Kersting, S. Nijssen, & F. Železný, eds., Machine Learning and Knowledge Discovery in Databases, Berlin, Heidelberg: Springer Berlin Heidelberg, s. 451–466.

Bradford, R. (2008). An empirical study of required dimensionality for large-scale latent semantic indexing applications, pp. 153–162.

Business Finland (2020a). Biotalous ja cleantech. Saatavissa (viitattu 30.3.2020): https://www.businessfinland.fi/suomalaisille-asiakkaille/strategia/bio/

Business Finland (2020b). Business Finland Customer Handbook.

Business Finland (2020c). Business Finlandin Rahoituspalvelut. Saatavissa (viitattu 30.3.2020): https://www.businessfinland.fi/suomalaisille-asiakkaille/palvelut/rahoitus/

Business Finland (2020d). Business Finlandin strategia. Saatavissa (viitattu 29.3.2020): https://www.businessfinland.fi/suomalaisille-asiakkaille/strategia/

Business Finland (2020e). Tietoa Business Finlandista lyhyesti. Saatavissa (viitattu 29.3.2020): https://www.businessfinland.fi/suomalaisille-asiakkaille/tietoa-meista/lyhyesti/

Business Finland (2020f, May 2). Hakuohjeet. Saatavissa (viitattu 2.5.2020): https://www.businessfinland.fi/suomalaisille-asiakkaille/palvelut/rahoitus/ohjeet-ehdot-ja-lomakkeet/hakuohjeet/

Butterfield, A., Ngondi, G. E. & Kerr, A. (2016). Regular Expression.

Castellanos, F., Valero-Mas, J., Calvo-Zaragoza, J. & Rico-Juan, J. (2018). Oversampling imbalanced data in the string space, Pattern Recognition Letters, Vol. 103, pp. 32.

Clark, A., Fox, C. & Lappin, S. (2013). The handbook of computational linguistics and natural language processing, West Sussex, England: Wiley-Blackwell. Saatavissa: https://andor.tuni.fi/permalink/358FIN_TAMPO/i4qs1o/alma9910617088205973

Coglianese, C. & Lehr, D. (2016). Regulating by Robot: Administrative Decision Making in the Machine-Learning Era, Georgetown Law Journal, Vol. 105, pp. 1147.

Cohen, K. Bretonnel. & Demner-Fushman, Dina. (2014). Biomedical natural language processing, Amsterdam: J. Benjamins Publishing Company. Saatavissa: https://andor.tuni.fi/permalink/358FIN_TAMPO/i4qs1o/alma9910652565705973

Davis, J. & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves, Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, s. 233–240.

Deerwester, S., Dumais, S., Furnas, G. & Landauer, T. (1990). Indexing by latent semantic analysis., Journal of the American Society for Information Science, Vol. 41(6), pp. 391–407.

Deng, L. & Liu, Y. (2018). Deep learning in natural language processing, Singapore: Springer. Saatavissa: https://andor.tuni.fi/permalink/358FIN_TAMPO/i4qs1o/alma9910307094205973

Deng, X., Li, Y., Weng, J. & Zhang, J. (2019). Feature selection for text classification: A review, Multimedia Tools and Applications, Vol. 78(3), pp. 3797–3816.

Edgar, T. & Manz, D. (2017). Research Methods for Cyber Security.

Glasgow, R. E. (2013). What Does It Mean to Be Pragmatic? Pragmatic Methods, Measures, and Models to Facilitate Research Translation, Health Education & Behavior, Vol. 40(3), pp. 257–265.

Goldberg, Y. (2017). Neural network methods in natural language processing, San Rafael: Morgan & Claypool Publishers. Saatavissa: https://andor.tuni.fi/permalink/358FIN_TAMPO/i4qs1o/alma9910307074205973

Good, Nathan. (2005). Regular Expression Recipes for Windows Developers A Problem-Solution Approach, , 1st ed. 2005., Berkeley, CA: Apress.

Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries, ECTJ, Vol. 29(2), pp. 75.

Haoyong Lv & Hengyao Tang (2011). Machine Learning Methods and Their Application Research, pp. 108–110.

Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L. & Bauder, R. A. (2020). Investigating class rarity in big data, Journal of Big Data, Vol. 7(1), pp. 23.

Hirschberg, J. & Manning, C. (2015). Advances in natural language processing, Science, Vol. 349(6245), pp. 261–266.

Hu, X. & Liu, H. (2012). Text analytics in social media, , Vol. 9781461432234, Springer US, s. 414.

Jiang, J. (2013). Information extraction from text, , Vol. 9781461432234, Springer US, s. 41.

Johnson, J. & Khoshgoftaar, T. (2019). Survey on deep learning with class imbalance, Journal of Big Data, Vol. 6(1), pp. 1–54.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H. & Mikolov, T. (2016a). FastText.zip: Compressing text classification models, ArXiv:1612.03651 [Cs]. Saatavissa: http://arxiv.org/abs/1612.03651

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016b). Bag of Tricks for Efficient Text Classification, ArXiv:1607.01759 [Cs]. Saatavissa: http://arxiv.org/abs/1607.01759

Jung, A. (2018). Machine Learning: Basic Principles.

Kalhori, H., Alamdari, M. M. & Ye, L. (2018). Automated algorithm for impact force identification using cosine similarity searching, Measurement, Vol. 122, pp. 648–657.

Koivisto, R., Leikas, J., Auvinen, H., Vakkuri, V., Saariluoma, P., Hakkarainen, J. & Koulu, R. (2019, February 1). Tekoäly viranomaistoiminnassa - eettiset kysymykset ja yhteiskunnallinen hyväksyttävyys. raportti. Saatavissa (viitattu 11.5.2020): http://julkaisut.valtioneuvosto.fi/handle/10024/161345

Korenius, T., Laurikkala, J., Järvelin, K. & Juhola, M. (2012). Stemming and lemmatization in the clustering of finnish text documents, ACM.

Krzanowski, W. J. (2009). ROC curves for continuous data, Boca Raton: CRC Press.

Laippala, V., Viljanen, T., Airola, A., Kanerva, J., Salanterä, S., Salakoski, T. & Ginter, F. (2014). Statistical parsing of varieties of clinical Finnish, Artificial Intelligence In Medicine, Vol. 61(3), pp. 131–136.

Lessmann, S. & Voß, S. (2008). Supervised Classification for Decision Support in Customer Relationship Management, Wiesbaden: Gabler, s. 253.

Liu, Z. & Bondell, H. (2019). Binormal Precision–Recall Curves for Optimal Classification of Imbalanced Data, Statistics in Biosciences, Vol. 11(1), pp. 141–161.

Luo, Q., Xu, W. & Guo, J. (2014). A Study on the CBOW Model's Overfitting and Stability, Vol. 2014-(November), pp. 9–12.

Ma, Y., Zhu, X., Zhu, S., Wu, K. & Chen, Y. (2018). Combating the class imbalance problemin sparse representation learning, Journal of Intelligent & Fuzzy Systems, Vol. 35(2), pp. 1865–1874.

Makrehchi, M. & Kamel, M. (2017). Extracting domain-specific stopwords for text classifiers, Intelligent Data Analysis : IDA, Vol. 21(1), pp. 39–62.

Mantere, S. & Ketokivi, M. (2013). Reasoning in Organization Science, The Academy of Management Review, Vol. 38(1), pp. 70–89.

Martín, F. M. D. P., Bertram, R., Häikiö, T., Schreuder, R. & Baayen, R. H. (2004). Morphological Family Size in a Morphologically Rich Language: The Case of Finnish Compared With Dutch and Hebrew, Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol. 30(6), pp. 1271–1278.

Martinez, A. R. (2010). Natural language processing, Wiley Interdisciplinary Reviews: Computational Statistics, Vol. 2(3), pp. 352–357.

Mirończuk, M. M. & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification, Expert Systems with Applications, Vol. 106, pp. 36–54.

Mitra, V., Wang, C.-J. & Banerjee, S. (2007). Text classification: A least square support vector machine approach, Applied Soft Computing, Vol. 7(3), pp. 908–914.

Müller, A. C. (2016). Introduction to machine learning with Python : a guide for data scientists, Sebastopol, California: O'Reilly Media.

Novotný, J. & Ircing, P. (2017). Unsupervised document classification and topic detection, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10458, pp. 748–756.

Park, Y. S., Konge, L. & Artino, A. R. J. (2020). The Positivism Paradigm of Research, Academic Medicine, Vol. Publish Ahead of Print.

Peng, R. D. (2011). Reproducible research in computational science, Science (New York, N.Y.), Vol. 334(6060), pp. 1226.

Peterson, L. E. (2009). K-nearest neighbor, Scholarpedia, Vol. 4(2), pp. 1883.

Řehůřek, R. (2011). Subspace tracking for latent semantic analysis, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 6611, pp. 289–300.

Rehurek, R. (2019, January 11). Gensim: Topics and Transformations. Saatavissa (viitattu 28.4.2020): https://radimrehurek.com/gensim/auto_examples/core/run_topics_and_transformations.html

Řehůřek, R. & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora.

Sagiroglu, S. & Sinanc, D. (2013). Big data: A review, pp. 42–47.

Saunders, Mark., Thornhill, A. & Lewis, Philip. (2009). Research methods for business students, , 5th ed., Harlow: Pearson.

Shenton, A. K. (2004). Strategies for Ensuring Trustworthiness in Qualitative Research Projects, Education for Information, Vol. 22(2), pp. 63.

Siponen, M. & Tsohou, A. (2018). Demystifying the influential IS legends of positivism, Journal of the Association of Information Systems, Vol. 19(7), pp. 600–617.

SITRA (2020, April 26). Cleantech. Saatavissa: https://www.sitra.fi/tulevaisuussanasto/cleantech/

Struhl, S. M. (2015). Practical text analytics : interpreting text and unstructured data for business intelligence, London, England ; Kogan Page. Saatavissa: Practical text analytics : interpreting text and unstructured data for business intelligence

Tharwat, A. (2018). Classification assessment methods, Applied Computing and Informatics.

Työ- ja elinkeinoministeriö (2018, December 1). Innovaariorahoituskeskus Business Finlandin tulostavoiteasiakirja vuosille 2018-2021.

Työ- ja elinkeinoministeriö (2020). Kysymyksiä ja vastauksia Business Finlandista. Saatavissa (viitattu 30.3.2020): https://tem.fi/kysymyksia-ja-vastauksia-business-finlandista

Underhill, D. G., McDowell, L. K., Marchette, D. J. & Solka, J. L. (2007). Enhancing Text Analysis via Dimensionality Reduction, 2007 IEEE International Conference on Information Reuse and Integration, s. 348–353.

Wajeed, M. A. & Adilakshmi, T. (2011). Semi-supervised text classification using enhanced KNN algorithm, 2011 World Congress on Information and Communication Technologies, s. 138–142.

Weng, W.-H., Wagholikar, K. B., McCray, A. T., Szolovits, P. & Chueh, H. C. (2017). Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach, BMC Medical Informatics and Decision Making, Vol. 17(1).

Wu, X., Du, Z. & Guo, Y. (2018). A visual attention-based keyword extraction for document classification, Multimedia Tools and Applications, Vol. 77(19), pp. 25355–25367.

Xu, G. X., Li, C. J., Li, Y. J., Ma, Y., Ma, X. L. & Pei, Z. X. Q. (2014). Review on Semantic Text Categorization, Applied Mechanics and Materials; Zurich, Vol. 644–650, pp. 2323–2328.

Zelikovitz, S. & Marquez, F. (2005). Transductive Learning for short-text classification problems using latent semantic indexing, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 19(02), pp. 143–163.

Zou, Q., Xie, S., Lin, Z., Wu, M. & Ju, Y. (2016). Finding the Best Classification Threshold in Imbalanced Classification, Big Data Research, Vol. 5, pp. 2–8.

LIITE 1: ESIKÄSITTELY JULKISTEN KUVAUSTEN PERUSTEELLA

Teksti	esikäsittely	doc2bow	tf_idf	lsi
Nykyisessä energiajärjestelmässä energian tuotantotavat johtavat useasti moniin energiavirtoihin kuten sähköön ja lämpöön...	nykyisessä energiajärjestelmässä energian tuotantotavat johtavat moniin energiavirtoihin sähköön lämpöön.	[(30, 1), (68, 2), (62, 1), (133, 1), (223, 1), (285, 1), ...	[(30, 0.08943389128246597), (68, 0.16159762535584044), (62, 0.04792100849093507), ...	[(0, 0.10239775686301074), (1, 0.0007332942715694459), (2, -0.021618566314944403), (3, 0.015730343183762 81), (4, -0.0015645198172367114)
Hankkeessa rakennetaan Kazakstaniin liiketoimintaekosysteemi, joka toteuttaa cleantech hankkeita Kazakstanissa hyödyntäen suomalaista teknologiaa.	hankkeessa rakennetaan kazakstaniin liiketoimintaekosysteemi toteuttaa cleantech hankkeita kazakstanissa hyödyntäen suomalaista teknologiaa...	[(351, 1), (407, 1), (513, 1), (934, 1), (1054, 1), (114, 1), ...	[(351, 0.22437691954920327), (407, 0.14673696859965152), (513, 0.30013403601405286), ...	[(0, 0.06281324482013 76), (1, -0.009631203850679889), (2, -0.011144637404195815), (3, 0.019831238698251208), (4, 0.013886997995806 83)
Fractuscan TUTL perustuu kahteen tutkimuskeksintöön ja sen tavoitteena on kehittää skannausjärjestelmä kallion rakojen karkeuden määrittämiseksi...	fractuscan tutl perustuu tutkimuskeksintöön kehittää skannausjärjestelmä kallion rakojen karkeuden määrittämiseksi...	[(80, 1), (109, 1), (422, 1), (844, 1), (4191, 1), (5182, 1), ...	[(80, 0.09751745255917518), (109, 0.21262192737251673), ...	[(0, 0.05222121804787884), (1, 0.007519967396572954), (2, -0.002558168018850976), (3, -0.014684911576442 6), (4, 0.0014190224479744757)
Suolistoperäisillä taudeilla ja oireyhtymillä on suuri merkitys ihmisen terveydelle ja lisääntyneisiin sairaanhoitokuluihin, jotka puolestaan kuormittavat valtiontaloutta...	suolistoperäisillä taudeilla oireyhtymillä merkitys ihmisen terveydelle lisääntyneisiin sairaanhoitokuluihin kuormittavat valtiontaloutta...	[(180, 1), (346, 1), (466, 1), (560, 1), (572, 1), (646, 1), ...	[(180, 0.07514345632475533), (346, 0.09652640031976131), (466, 0.06489696209474385), ...	[(0, 0.04947492572007591), (1, -0.004423209928779655), (2, -0.004504732010378078), (3, -0.012475463325852029), (4, 0.006056982929993 86)

LIITE 2: JULKISTEN KUVAUSTEN TOTUUSARVOT

Teksti	Luokitus	Luokittelijan luokitus	Totuusarvo
Nykyisessä energiajärjestelmässä energian tuotantotavat johtavat useasti moniin energiavirtoihin kuten sähköön ja lämpöön…	cleantech	cleantech	Todellinen positiivinen
Hankkeessa rakennetaan Kazakstaniin liiketoimintaekosysteemi, joka toteuttaa cleantech hankkeita Kazakstanissa hyödyntäen suomalaista teknologiaa.	cleantech	ei cleantech	Virheellinen negatiivinen
Fractuscan TUTL perustuu kahteen tutkimuskeksintöön ja sen tavoitteena on kehittää skannausjärjestelmä kallion rakojen karkeuden määrittämiseksi…	ei cleantech	cleantech	Virheellinen positiivinen
Suolistoperäisillä taudeilla ja oireyhtymillä on suuri merkitys ihmisen terveydelle ja lisääntyneisiin sairaanhoitokuluihin, jotka puolestaan kuormittavat valtiontaloutta…	ei cleantech	ei cleantech	Todellinen negatiivinen