ICD-koodien automaattinen määritys luonnollisen kielen käsittelyn menetelmillä

Turun yliopisto
Tietotekniikan laitos
LuK-tutkielma
Tietojenkäsittelytiede
Kesäkuu 2024
Henrik Heinonen

TURUN YLIOPISTO
Tietotekniikan laitos

Henrik Heinonen: ICD-koodien automaattinen määritys luonnollisen kielen käsittelyn menetelmillä

LuK-tutkielma, 24 s.
Tietojenkäsittelytiede
Kesäkuu 2024

Terveydenhuollon potilasasiakirjojen sähköistymisen myötä on tullut mahdolliseksi asiakirjojen koneellinen käsittely. Tämän johdosta voidaan myös käyttää luonnollisen kielen käsittelyn menetelmiä näihin tekstiaineistoihin. Maailmanlaajuisesti on käytössä YK:n alaisen Maailman terveysviraston ylläpitämä ICD tautiluokitusjärjestelmä, jossa taudit kuvataan ICD-koodeina. Tutkielmassa keskitytään tautiluokitusjärjestelmän ICD-9 ja ICD-10 versioihin. Tutkielma on toteutettu kirjallisuuskatsauksena etsien tietoa siitä miten koodien automaattista määrittämistä tutkitaan tällä hetkellä, millä luotettavuustasolla järjestelmät ovat ja mitkä ovat keskeisimmät ongelmat tutkimusalueessa. Keskeisessä roolissa ovat nykyaikaiset BERT:iä tekstin käsittelyn menetelmänä käyttävät syväoppivat esikoulutetut kielimallit joiden suorituskykyä verrataan tunnettuihin luokittelumenetelmiin logistisesta regressiosta takaisinkytkettyihin neuroverkkoihin ohjatun koneoppimisen saralla. Järjestelmien vertailemiseen on käytetty MIMIC-tietoaineistoja saavutettavuuden vuoksi. Nykytasolla automaattinen ICD-koodien määrittäminen ei ole yksinkertainen tehtävä koneoppimisluokittelijalle, vaikkakin kehitystä on tapahtunut edellisiin järjestelmiin. Yleisimmät tautiluokat onnistutaan luokittelemaan oikein harvinaisempia useammin. Ongelmaksi on muodostunut tietoaineistojen heikko saatavuus ja koodiluokkien epätasapaino olemassa olevissa aineistoissa, luokkien lukumäärä ja nykyjärjestelmien suorituskyky sekä muistinhallintaongelmat.

Asiasanat: luonnollisen kielen käsittely, NLP, ICD-koodaus, koneoppiminen

Sisällys

1 Johdanto

Digitalisaation myötä on tullut paljon uusia mahdollisuuksia käsitellä suurta määrää aineistoja nopeasti ja automaattisesti. Terveydenhuollon ala ei ole tässä mikään poikkeus. Toisaalta terveydenhuollossa tiedon oikeellisuudella ja tietosuojalla on suuri paino-arvo, kun käsitellään ihmisten henkilökohtaisia tietoja. Potilasasiakirjojen sähköistymisen myötä voitaisiin niistä koneoppimisen menetelmien avulla saada paljon tietoa jalostettua jatkotutkimusta varten. Tästä yksi esimerkki on ICD-koodaukseen perustuvat diagnoosikoodit. Diagnoosikoodeja käytetään Suomessakin moniin eri tarkoituksiin. Terveydenhuollon henkilökunnan kirjaamien potilaskertomusten lisäksi tietoa käytetään niin hoito-ohjeita laatiessa ja vakuutuksiin liittyvissä päätöksissä [1].

Tutkielmassa on tarkoituksena tarkastella nykyhetkeä ICD-koodien automaattisen määrittelyn saralla ja millä luonnollisen kielen käsittelyn menetelmillä nämä on saavutettu. Tarkoitus on löytää kehityksen nykyhetkeä edustavat moniluokkaluokittejat, pohtia niiden rakenneta ja arvioida niiden suorituskykyä.

1. Tutkimuskysymys: Pystytäänkö luonnollisen kielen käsittelyn menetelmien avulla määrittämään ICD-koodeja potilaskertomusten perusteella?

   Tutkimuskysymystä lähestytään tarkastelemalla nykyhetkeä kuvaavien moniluokkaluokittejoiden suorituskysyä herkkyyden, tarkkuuden ja F1-arvon osalta, jos kaikki tiedot ovat ilmoitettu. Tämän lisäksi pohditaan, että miten nämä numerot ovat käyttökelpoisia kyseisessä toimintaympäristössä.

2. Tutkimuskysymys: Minkälaisia menetelmiä on käytetty ja mitä haasteita on kohdattu?

   Toista tutkimuskysymystä lähestytään tarkastelemalla käytettyjä menetelmiä järjestelmät sisältävät ja miksi ne ovat valikoituneet.

Tiedonhaku aloitettu hakemalla englanniksi ”Automatic ICD-coding” ja tarkentamalla AND ”ICD-x”, jossa x $\in 9, 10, 11$ Google Scholarin, IEEE:n ja ACL Anthologyn hakukoneisiin. Haku rajattiin alkaen ensin vuoteen 2018, mutta myöhemmin tarkentui vuoteen 2022. Hakua tarkennettiin sisältämään ICD-koodauksen eri versiot 9, 10 ja 11. Lisäksi suoritettiin haku suomeksi UTU Volter palveluun, mutta sieltä ei aineistoa löytynyt. Lähdeaineiston kasaamiseen on myös käytetty arXiv palvelua, kiinnittäen erityistä huomiota siihen ettei sivuston julkaisut ole vertaisarvioituja. Tiedonhakua varten on tutustuttu noin 50 eri artikkeliin, näistä muodostui päälähteiksi kaksi eri tutkimusta. Määritelmien ja menetelmien avaamiseen on käytetty noin 58 lähdettä. Kaikki tulokset ovat alkuperäisistä lähteistä, luvussa 4 aliluvuissa 4.1.1 ja 4.1.2 löytyvät kuvaajat ovat minun tekemiä. Kaikki muut kuvat ovat alkuperäisistä tutkimuksista.

Johdannon jälkeen tutkielma etenee seuraavasti. Luvussa 2 esitellään mitä luonnollisen kielen käsittely on ja mitä menetelmiä se pitää sisällään. Aliluvussa 2.1 tutustutaan luonnollisen kielen käsittelyn historiaan ja tärkeimpiin kehitysvaiheisiin nykyhetkeen asti. Aliluvussa 2.1 tarkennetaan määritelmä terveydenhuollossa käytetyn kliinisen tiedon käsittelyyn tarkoitettuihin seikkoihin. Luvussa 3 käsitellään ICD-koodaus ja tutkimukset. Ensin avataan ICD-koodauksen määritelmä, käyttötarkoitukset ja koodauksen erot eri versioiden välillä. Aliluvussa 3.1 kerrotaan mitä tarkoittaa sähköinen potilasasiakirja. Aliluvussa 3.3 selitetään käytettävät suorituskykyä mittaavat mittarit ja niiden kaavat. Tuloksista tehdään päätelmiä viimeisessä luvussa 4. Aliluvussa 4.1.1 käydään ICD-9 perustuva tutkimus ja sen tulokset. Aliluvussa 4.1.2 käydään ICD-10 perustuva tutkimus ja sen tulokset. Aliluvussa 4.1.3

keskitytään ICD-11 versioon ja kerrotaan sen kehitysvaiheesta. Tämän jälkeen aliluvussa 4.2 esitän omia pohdintojani kirjallisuuskatsauksen tuloksista. Viimeisessä aliluvussa 4.3 pohditaan mahdollisia suuntia jatkotutkimuksille.

2 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely (Natural Language Processing, NLP) on yksi tekoälyn kehityksen ja kielentutkimuksen yhdistävistä haaroista, joka on erikoistunut ihmiskielen koneelliseen käsittelyyn [2]. Nämä data-aineistot ovat joko kuvamuodossa olevaa tekstiä kuten skannatut dokumentit, digitaalista tekstiä tai puhetta. Näitä aineistoja käsittelemällä tietokoneohjelmat voivat koostaa suuria tekstiarkistoja, joita kutsutaan korpuksiksi. Korpukset toimivat raakana tekstidatana jota voidaan prosessoida monia erilaisia menetelmiä käyttäen.

2.1 Yleisesti

NLP:n saralla tällä hetkellä tunnetuimmat sovellukset liittyvät chat-botteihin, puheesta tekstiksi kääntämiseen, suurten tekstidatamassojen analysointiin, kielenkääntämiseen, asiayhteyksien ja entiteettien tunnistamiseen tekstin sisältä (Named Entity Recognition, NER) sekä tekstiin pohjautuva tunnepohjainen tekstianalyysi (Sentiment Analysis, SA). Luonnollisen kielen käsittelyn kaksi päähaaraa ovat luonnollisen kielen ymmärtäminen ja käsittely (Natural Language Understanding, NLU) ja luonnollisen kielen tuottaminen (Natural Language Generation, NLG) kuten GPT-mallit [2]. Viimeisten vuosien aikana tämä jaottelu kuitenkin on uusien menetelmien myötä tullut häilyvämmäksi.

Työssä tullaan käsittelemään ymmärtämisen ja käsittelyn puolta. NLU:n osa-alueet muodostavat kokonaisuuden, jonka avulla ihminen pystyy tietokoneen ja kie-

lentutkimuksellisten menetelmien avulla analysoimaan suuria tekstimassoja. Teksteistä pystytään prosessoimaan auki niin kokonaisuuksia luokittelulla kuten artikkelin tyyppi, jäsentämään lausetta (parsing) avainsanoja avainsana-analyysin (keyness) avulla kuin tutkimaan miten kieli on kehittynyt. Tässä työssä käytetään seuraavia kielentutkimuksen termejä: syntaksi[1], morfologia[2] ja leksikaalinen[3]. Syntaksi ja morfologia ovat molemmat kieliopillisen järjestelmän peruskomponentteja. Syntaksi eli lauseoppi keskittyy tarkastelemaan sanojen rakentumista lauseiksi tai lausekkeiksi. Morfologia eli muoto-oppi puolestaan tarkastelee sanojen muodostumista ja taivutusta keskittyen kuvaamaan yksittäisten sanojen koostumusta. Leksikaalinen sen sijaan tarkoittaa sanoihin olennaisesti liittyvää asiaa. NLP:n leksikaalisessa analyysissä pilkotaan tekstimassa ensin pelkästään sanoiksi, joille kohdistetaan sanaluokka (Part-Of-Speech tagging, PoS) analyysiä erikseen.

NLP menetelmien kehittämisen on katsottu alkaneen jo viime vuosisadan puolivälissä [3]. Luonnollista kieltä on käsitelty jo 1950-luvulta lähtien, tällöin asia määriteltiin keinoälyn ja kielentutkimuksen risteytymänä. Tietokoneella tehtävän digitaalisen luonnollisen kielen käsittelyn alkuvaiheissa menetelmät perustuivat sääntöpohjaisiin menetelmiin. Teknologioiden kehittyessä sääntöpohjaisista menetelmistä siirryttiin todennäköisyyspohjaisiin menetelmiin kuten tilastopohjaiset sanayhteysanalyysit. Luonnollisen kielen käsittelyn kehitysvaiheet ovat nopeutuneet 2000–luvun alun jälkeen teknologioiden kehittyessä. Vuonna 2003 Bengio ym esittelivät neuro kielimallinnuksen (Neural language modelling), jossa sanaesiintymän seuraavan sanan todennäköisyys määritellään edellisten sanoja perusteella [4]. Yksi koneoppimisen menetelmistä on monitehtävä oppiminen (Multitask Learning, MTL)[5]. Vaikka menetelmänä MTL esitettiin jo 1997 tässä kontekstissa se kuitenkin julkistettiin vasta vuonna 2008. Kahta konvoluutioverkkoa käytetään sanaluokka leimaami-

[1]Tieteen termipankki, Kielitiede:syntaksi, 2024

[2]Tieteen termipankki, Kielitiede:morfologia, 2024

[3]Tieteen termipankki, Kielitiede:leksikaalinen, 2024

seen ja nimettyjen entiteettien tunnistamiseen. Kehitys jatkui tästä muutaman vuoden kuluttua, kun vuonna 2013 sanaupotusmenetelmä (Word Embedding, WE) ja word2vec-kirjasto esiteltiin [6][7]. Menetelmän avulla pystytään tuottamaan nopeasti taulukko niin semanttisella kuin syntakstisella sanayhteydellä, joista muodostuu vektoriavaruusmallit. Vektoriavaruusmallit ovat tärkeitä sanayhteyksien rakentamiseen ja syöttämiseen neuroverkoille.

Seuraava suuri kehitysaskel saavutettiin syvien neuroverkkojen (Deep Neural Net, DNN) käyttöönotolla luonnollisen kielen käsittelyn tehtävissä [8]. Syvillä neuroverkoilla tarkoitetaan koneelle mallinnettua biologisen oppimisen mallia. Neuroverkko sisältää laskennallisia neuroneja, jotka muistuttavat käsitteellisellä tasolla biologista neuronia. Yhdelle tälläiselle neuronille annetaan yksi tai useampi syötesignaali, joka tässä yhteydessä on lukuarvo ja se lähettää siitä koostetun yhden signaalin vrt. impulssi ihmisellä. Neuroverkkojen ensimmäisessä kerroksessa on raaka tekstidata, jonka jälkeen yksi tai useampi piilokerros muokkaavat aina vaiheittain tekstidataa ja lopulta päädytään ulostulokerrokseen, jossa on ennustettavat luokat. Näitä syvien neuroverkkojen käyttämistä NLP:ssä tutki esimerkiksi Vu ym. yhdistämällä konvoluutioneuroverkkoja (Convolutional Neural Network, CNN) ja takaisinkytkettyviä neuroverkkoja (Recurrent Neural Network, RNN) tekstiluokittelussa [9].

Kehitysaskelina eteenpäin edetessä mukaan astuivat vuonna 2015–2017 muuntimet (Transformer, Encoder - Decoder) jotka huomiomekansimin (Attention Mechanism) mallintavat sanojen välisiä yhteyksiä lauseissa [10]. Huomiomekanismin avulla voidaan poimia pitkästä tekstistä kuten lauseesta sitä tärkeimmin kuvaavat sanat. Yksi tämän menetelmän tärkeimpiä tuotteita on Googlen BERT (Bidirectional Encoder Representations from Transformers) vuodelta 2018 [11]. Turun yliopiston TurkuNLP tutkimusryhmä on julkaissut vuonna 2019 FinBERT joka on suomen kielelle koulutettu kielimalli [12][13]. Virtanen ym. toteaa artikkelissaan, että monikielinen kielimalli ei ole hyvä suomen kielen tarpeisiin, jonka vuoksi on vaadit-

tu täysin oma [14]. Järjestelmien kehittyessä, ei tämä kielispesifinen mallintaminen enää ole tarpeellista. Uusimpana kehitysaskeleena pidetään zero-shot ja few-shot menetelmiä, jossa neurokonekäännöstä (Neural machine translation, NMT) tehdään siirtokoulutetuilla kielimalleilla lähtökielestä toiseen [15][16].

2.2 Kliininen NLP

Nopean kehityksen ja digilisaation myötä myös lääkinnälliset tekstit ovat kaikki muuttuneet datamuotoon sähköisiksi potilaskertomuksiksi. Yksi luonnollisen kielen käsittelyn osa-alueista on kliininen luonnollisen kielen käsittely (Clinical NLP). Tämä osa-alue on keskeisessä roolissa, kun käännetään saneltuja ja kirjoitettuja lääkinnällisiä tietoja standardoituun esitystapaan, joka on vertailtavissa ja tietokoneiden käsiteltävissä. Tällöin laaditaan tyypitettyjä data-aineistoa, jotta voidaan toimia useiden eri toimijoiden kesken saman tekstiaineiston parissa. Tekoälyn ja koneoppimisen yleistyessä ja kehittyessä, myös kliininen NLP on siirtynyt käyttämään näitä malleja. Yksi esimerkki tästä on tutkimus, joka on tehty lääkearvosteluista laaditusta asenneanalyysistä. Tunnepohjaisena tekstianalyysinä tehtynä pyrittiin päättelemään automaattisesti kirjoituksen sävyä binäärisesti luokittelemalla negatiiviseksi tai positiiviseksi tekstiaineistosta. Esimerkissä on toteutettu sanavarastopohjainen (Bag of Words, BoW) mielipideanalyysi yleisesti käytössä oleville lääkkeille kuten Viagra ja Oseltamivir [17].

Yksi kliinisen luonnollisen kielen käsittelyn tutkimushankkeista on jo kauan ennen moderneja tietokoneita alkanut LSP-MLP (The Linguistic String Project - Medical Language Processor) [18] vuodelta 1973 [19]. Tästä järjestelmästä pohjautuu myös monia muita tuotteita kuten New Yorkissa sijaitsevan Columbian yliopiston kehittämä lääkinnällisten tietojen tekstistä louhiva MEDLEE (MEDical Language Extraction and Encoding System)[20]. Wu ym. toteaa kliinisen luonnollisen kielen käsittelyn ongelmana olevan puutteellinen tai riittämätön data. Tämä ongelma il-

menee sekä ohjatulle opppimiselle jossa koneelle annetaan jo ihmisen etukäteen luokittelemia esimerkkejä, että ohjaamattomalle oppimiselle jossa konetta koulutetaan pelkkien esimerkkien avulla ja yritetään ryhmitellä samankaltaisia sanoja klustereihin. Nämä ongelmat voidaan jakaa kolmeen kategoriaan: Ensinnäkin luokiteltua dataa ohjatuluille malleille ei ole saatavilla laajasti ja ovat siis vaikeasti skaalattavissa. Toiseksi kliiniset NLP sovellukset joutuvat käsittelemään epätasapainoista dataa, kuten lukumäärällisesti arvoituja diagnooseja. Kolmanneksi NLP järjestelmät vaativat käyttöön suuren määrän laskentatehoa. Järjestelmien saaminen lääkinnälliset vaatimukset täyttävään turvalliseen tutkimusympäristöön kuten sairaalaan on vaikeaa [21].

Kliinisen NLP:n todetaan olevan tietotaidollisesti hankalaa, koska tarvitaan standardoidussa muodossa olevaa tekstiä, jota tietokone pystyy prosessoimaan ja tulkitsemaan [21]. Normaalien NLP tehtävien lisäksi kliinisen NLP järjestelmän tulee myös pystyä tekemään potilastasoista päättelyä kuten nimettyjen entiteettien tunnistamista tai dokumenttien luokittelua. Alityyppien päättelystä esitellään Rannikmäe ym. tutkimuksessa jossa käsiteltiin Scottish UK Biobank:in tietoja aivohalvaustapauksissa. Tutkimustuloksissa todetaan automaattisen menetelmän tarjoavan käyttökelpoisen, skaalattavan ja tarkan ratkaisun tautien alityypitykseen radiologian saralla [22]. Menetelmänä Rannikmäe ym. käyttivät viisi vaiheista prosessia jonka apuna käytettiin valmista SemEHR [23] järjestelmää. Ensin SemEHR annotoi entiteettien tunnistamisella tutkimusraportit liittyviin termeihin SemEHR tietokannastaa. Tämän jälkeen terveydenhuollon opiskelijat tarkistavat tulokset ja tekivät lisäannotointeja jos tarpeen. Tällä saatiin koulutettua SemEHR tarkemmin kyseessä olevaan tehtävään. Seuraavaksi SemEHR kävi tiedostot uudelleen läpi uudella koulutusdatalla annotoiden. Viimeisessä vaiheessa annotoituihin tiedostoihin kohdistettiin terveydenhoidon erikoisalaan erikoistuneita sääntöpohjaisia menetelmiä joiden avulla saatiin jokainen tutkimusraportti tiivistettyä yhteen diagnoosikoodiin. Tutkimuk-

sen tuloksia esitetään kuvassa 2.1 ja englanninkieliset lyhenteet avattu taulukossa 2.1.

Stroke subtype	Precision (i.e. positive predictive value) (95% CI)		Recall (i.e. sensitivity) (95% CI)	
	From codes (based on previous work [2])	From automated method	From codes (based on previous work [2])	From automated method
ICH	42% (31–54%) (11/26)	89% (52–100%) (8/9)	100% (72–100%) (11/11)	89% (52–100%) (8/9)
SAH	71% (54–83%) (17/24)	82% (57–96%) (14/17)	100% (80–100%) (17/17)	82% (57–96%) (14/17)
IS	83% (75–89%) (73/88)	73% (65–81%) (91/124)	49% (41–57%) (73/149)	64% (56–72%) (91/142)
IS (including cases with an unspecified subtype assigned as IS)	80% (76–83%) (147/184)	77% (71–83%) (141/182)	99% (95–100%) (147/149)	99% (96–100%) (141/142)

Kuva 2.1: Tutkimustulokset [22]

Sarake	Selitys
ICH	Intraserebraalivuoto
SAH	Lukinkalvonalainen verenvuotoa
IS	Ohimenevä aivoverenkiertohäiriö
IS + subtype	Ohimenevä aivoverenkiertohäiriö ja määrittelemätön alatyyppi

Taulukko 2.1: Sarakkeiden suomennokset

3 Automaattisen ICD-koodauksen tutkimus

Tässä luvussa käsitellään ICD-koodausta (International Classification of Diseases, ICD) [24], [25]. Luokittelujärjestelmän on laatinut Maailman Terveysvirasto (WHO). ICD-järjestelmä on WHO:n perustuslaillinen terveysstandardi, joka ei ole kieli- tai kulttuurisidonnainen. Standardoidun järjestelmän edut ovat vertailtavat tilastot ja semanttinen yhteentoimivuus. WHO on julkaissut ja toimeenpannut standardista ICD-11 version 1. tammikuuta 2022. Kirjallisuuskatsauksen tekoaikana Suomessa on käytössä ICD-10, mutta Terveyden ja Hyvinvoinninlaitos (THL) on ilmoittanut ICD-11 version käyttöönotosta vuosien 2023–2026 välisenä aikana [26]. Suomessa THL:n mukaan järjestelmällä yhtenäistetään terveystietojen järjestelmiä. ICD-koodausjärjestelmää käytetään kuolinsyy- ja sairastavuustilastotietoja kerättäessä, kliinisessä työssä potilasasiakirjan diagnoosimerkintöjä tehdessä, sosiaalivakuutuksen lääkärilausuntojen diagnoosikirjauksissa josta esimerkkinä A- ja B-todistukset, tutkimus- ja kehittämistyössä tautien nimeämiseen, tiedonhakusanastona sekä sosiaalihuollon asiakasasiakirjoissa sairauden kuvaamiseen [1]. Terveyden ja hyvinvoinninlaitos vastaa Suomessa järjestelmän käytöstä sosiaali- ja terveysministeriön määräysten mukaisesti.

Työssä tullaan tarkastelemaan sekä ICD-9 sekä ICD-10 järjestelmälle tehtyjä tutkimuksia koska elämme muutosvaiheessa eri versioiden välillä. Koska tutkimusala on

suhteellisen uusi, ei myöskään uudemmille versiolle löydy kovinkaan laajasti materiaalia. Tästä syystä kirjallisuuskatsaus keskittyy syvällisemmin ICD-9 versioon, sivuten myös uudempia. Tutkimukset eri aineistoilla eivät ole kaikiltaosin vertailtavissa koska koodiston rakenne on muuttunut versiosta toiseen. ICD-9 koodistossa diagnoosikoodin pituus on 3-5 merkkiä, kun taas ICD-10 diagnoosikoodin pituus on 3-7 merkkiä. Tämä tarkoittaa sitä, että ICD-9 järjestelmässä on noin 13 000 erilaista koodia, kun taas ICD-10 koodistossa on laajimmillaan Yhdysvalloissa käytössä olevassa ICD-10-CM versiossa noin 68 000 eri vaihtoehtoa. Tämä harppaus merkkien lukumäärässä tarkoittaa myös sitä, että ICD-10 järjestelmään pystyään lisäämään helposti uusia diagnoosikoodeja. Toisaalta lisätty merkkimäärä tarkoittaa suurempaa erittelykykyä diagnoosien määrittämisessä ja mahdollistaa erottelun esimerkiksi oikean ja vasemman puolen kehon osan välillä [27].

3.1 Sähköinen potilasasiakirja

Sähköinen potilasasiakirjalla (Electronic Patient Record) on pyritty korvaamaan mahdolliset fyysiset tietovarastot joissa potilaiden sairaustietoja lisätään, päivitetään tai poistetaan. Asiakirjalle itsessään ei ole mitään globaalia formaattia, joten myöskään tiettyyn muotoon tehdystä sähköisestä potilasasiakirjasta tietoja louhiva koneoppimisluokittelija ei välttämättä ole yhteensopiva toisen alueellisen toimijan asiakirjapohjan kanssa.

3.2 Automatisointi

ICD-koodauksen automatisointi on ollut pitkään vaikeaa kuten aliluvussa 2.2 kliinisen aineiston käsittelyn haasteista kerrotaan, kuitenkin tietoaineistoja on olemassa joilla sitä voidaan tehdä kuten MIMIC-tietokannat [28]. Osa tietoaineistoista ovat yksityisiä jotka on koostettu tietyn sairaalan potilastietokannasta eikä näihin ei ole

yleisesti pääsyä. Tämä heikentääkin tutkimusten tulosten toistettavuutta joka on akateemisen tutkimustyön yksi kulmakivistä. Terveydenhuollon palvelut tuottavat valtavat määrät aineistoa ja tämän aineiston käsittelyyn kone-oppiminen on todistetusti hyvä työkalu päätöksentekoon [29].

3.3 Luokittelijan tarkkuus ja käytetyt mittarit

Kirjallisuudessa esiintyvien tutkimusten tulosten kuvaamiseen ja eri mallien suorituskykyä vertaamiseen tutkimusryhmät esittelevät työnsä tuloksia ensinnäkin kahteen keskiarvoon perustuen, jotka ovat *mikro* ja *makro*. Keskiarvoja lasketaan parametreille jotka ovat sisäinen tarkkuus (Precision, Pr), sisäinen herkkyys (Recall, Re) ja $F1$-arvo. Parametrien laskemiseen käytetään havaittuja suureita jotka ovat: Oikea positiivinen (True positive, TP), jolloin luokittelija on tunnistanut todelliselle luokkaan kuuluvalle tapaukselle oikean luokan. Väärä positiivinen (False positive, FP), missä tapauksessa luokittelija on määritellyt tapaukselle kyseisen luokan, eikä tämä pidä paikkaansa. Tätä tapausta kutsutaan myös tyypin 1 virheeksi. Kolmas havaittu suure on väärä negatiivinen (False negative, FN), jolloin tapaus kuuluu oikeasti kyseiseen luokkaan, mutta luokittelija ei onnistu oikein luokittelussa. Tätä kutsutaan myös tyypin 2 virheeksi. Sisäinen herkkyys on suhdeluku sille montako oikeaa positiivista saadaan tunnistettua koko aineistosta. Sisäinen tarkkuus on suhdeluku sille montako järjestelmän oikeaksi tunnistetuista oli todellisuudessa oikein.

Näiden lisäksi todennäköisyyksiin perustuvat luokittelijat tuottavat lopputuloksenaan posterioritodennäköisyyksiä jolloin päätöksentekoa voidaan rajata asettamalla leikkauspiste todennäköisyydelle. Tällöin säätöä voidaan tehdä tyypin 1 ja tyypin 2 virheiden kautta ja kuvaamalla tapahtumaa erottelukykykäyrällä (receiver operating characteristic, ROC) [30]. ROC-käyrää voidaan kuvata pelkästään yhdellä käyränalaisen pinta-alan (Area Under the Curve, AUC) lukuarvolla. Lukuarvoa voidaan tulkita niin että 0.5 on täysin sattumanvarainen luokittelija ja 1.0 kuvaa

täydellistä [31]. Tuloksissa on kuvattu myös AUC niiltä osin kuin ne lähdetiedoissa on ilmoitettu.

Kolmanneksi tutkimukset raportoi $P@K$ $K = 1, 2, \ldots, n$ arvon, jolla kuvataan todennäköisyyttä tapaukseen liittyvien luokkien sisältymistä K-arvon ilmoittaman lukumäärän luokkia sisälle.

Mikro-keskiarvon laskukaavat:

$$Pr_{mikro} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)} \tag{3.1}$$

$$Re_{mikro} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)} \tag{3.2}$$

Makro-keskiarvon laskukaavat:

Makro-keskiarvon laskennassa lasketaan ensin luokittain herkkyys ja tarkkuus kaavoilla:

$$Pr = \frac{TP}{TP + FP} \tag{3.3}$$

$$Re = \frac{TP}{TP + FN} \tag{3.4}$$

Tämän jälkeen lasketaan näiden tulosten summat ja jaetaan luokkien lukumäärällä ($N = 1, 2, \ldots, n$)

$$Pr_{makro} = \frac{\sum_{i=1}^{n} Pr_i}{N} \tag{3.5}$$

$$Re_{makro} = \frac{\sum_{i=1}^{n} Re_i}{N} \tag{3.6}$$

Riippumatta tavasta laskea herkkyys ja tarkkuus, lasketaan F1-suure samalla kaavalla:

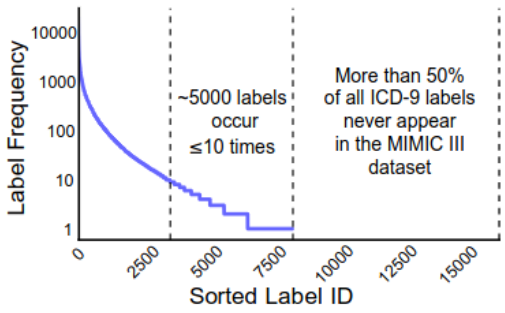
$$F1 = 2 \cdot \frac{Pr \cdot Re}{Pr + Re} \tag{3.7}$$

3.4 ICD-9

Vaikka ICD-9 järjestelmänä on otettu käyttöön jo 1. tammikuuta 1979 [32], on sille edelleen tehty tutkimuksia kuten Duan ym. toukokuussa 2023 esittelemä MHLAT (Multi-Hop Label-Wise Attention Model for Automatic ICD Coding) [33] joka vielä tammikuussa 2024 edusti parasta menetelmää lähes kaikilla mittareilla automatisoinnin osalta. Toisaalta pienillä muutoksilla tutkimuskohteen parametreihin löytyy lisääkin tutkimuksia, mutta tämä valittu tutkimus-skenaario on eniten tutkittu. Parhaimman yleisen suorituskyvyn vuoksi tässä kirjallisuuskatsauksessa keskitytään esittelemään kyseistä tutkimusta tarkemmin. Tuloksissa on esitelty sekä alkuperäisessä tutkimuspaperissa ollut paras tieto vertailukohdista, sekä muualta kirjallisuudesta löytyneitä järjestelmiä jotka ovat samalla esikäsittelyllä ja aineistoilla testattuja. Suurin osa automaattista ICD-koodien määrittämistä tutkivista järjestelmistä tekevät sitä ICD-9 versiolla, koska suorituskykyä pystytään vertaamaan eri järjestelmien välillä. Uudemmilla versioilla tehtävälle tutkimukselle ei tutkimustietoa / -kirjallisuutta ole niin paljon saatavilla.

Wu ym. totetaa omassa tutkimuksessaan automaattisen ICD-koodauksen olevan tyypiltään erittäin laaja-alainen moniluokkaluokittelu tehtävä (Extreme Multi-Label Classification, XMLC) [34] [35]. Aineistona tutkimuksissa on yleisesti käytetty sekä MIMIC-II että MIMIC-III tietoaineistoja vertailtavuuden vuoksi. Näistä MIMIC-III voidaan nähdä edellisen jatkona olleen laajempi. Suurimmat ongelmat automatisoinnissa liittyvät eri ICD-koodien jakaumiin. Tästä esimerkkinä MIMIC-III tietoaineiston luokkajakauma kuvassa 3.1. Kuvasta voidaan todeta, että yli puolet ICD-9 koodiluokista ei esiinny kertaakaan MIMIC-III tietoaineistossa, ja noin 5000 koodiluokkaa esiintyy $n \leq 10$ kertaa. Tämä on toisaalta luonnollista, kun esimerkiksi flunssaa esiintyy populaatiossa enemmän kuin syöpää.

Toinen ongelmakohta liittyy Duan ym. mukaan luokkien leimojen (label), eli tässä kontekstissa eri diagnoosikoodien lukumäärään [33]. Taulukosta 3.1 huomataan,



Kuva 3.1: ICD-koodien jakauma MIMIC-III tietokannassa [36]

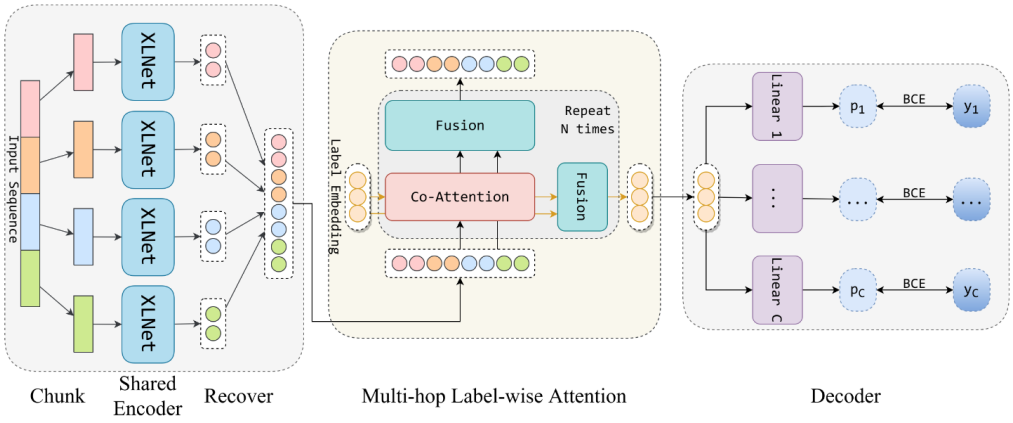
että laajimassa aineistossa on 8922 eri luokkaa.

Taulukko 3.1: Tietoaineistojen tietoja [33]

	MIMIC-III Full	MIMIC-III TOP50	MIMIC-II
Tokenien lkm per dokumentti (ka.)	1483	1527	1106
Tokenien maksimi lkm per dokumentti	8772	7567	6231
Luokkien lkm.	8922	50	5031

Kolmas ongelma liittyy potilasasiakirjan tekstien pitkään muotoon, laajimassa MIMIC-III Full aineistossa tokenien lukumäärä maksimissaan on 8772 joka aiheuttaa esikoulutetuille kielimalleille ongelmia aikavaativuuden kanssa suorituksessa ja muistinhallinnassa [33]. Koska aikavaativuus on neliöllinen pahimmillaan, kehitystyö suurilla kielimalleilla (Large Language Model, LLM) ei ole ollut ongelmatonta [37]. Tästä syystä nyt kuvassa 3.2 esitelty järjestelmä käyttää esikäsittelyä syötteelle välttääkseen suorituskykyongelmat [33]. Duan ym. kertoo käyttäneensä järjestelmäänsä jo olemassa olevia rakenteita kuten XLNet moduuleita [38] jotka on peräisin HiLAT [39] järjestelmästä ja jo esikoulutettu MIMIC-korpuksella [33]. Tuotetut vektorit käsitellään luokkien piirrevektoreiksi jonka jälkeen luokittelukerroksessa luokkakohtainen moniluokittelija päättää mihin luokkaan lähtödokumentti kuuluu [33].

Tietoaineistojen jakoon koneoppimisluokittelijan kouluttamista ja testaamista varten aineistot jaettiin Duan ym. mukaan samalla periaatteella kuin Mullenbach ym. [40]. Tällöin pysyy vertailtavuus järjestelmien välillä. Aineistot esikäsiteltiin



Kuva 3.2: MHLAT järjestelmän arkkitehtuurikuvaus [33]

NLP suoritusketjulla poistamalla pelkät numeeriset tokenit esim ”500” mutta säilyttäen kuitenkin ”250mg”. Teksteistä poistettiin kaikki isot kirjaimet, korvattiin harvinaiset tokenit ”UNK” tokenilla ja tuotettiin sanavektorit Word2Vec-menetelmällä. Tämän lisäksi kaikki dokumentit typistettiin maksimimittaan 2500 tokenia. [41] [40]

Taulukko 3.2: Tietoaineistojen jaot [41]

Aineisto	Koul	Test	Val	Koul (%).	Test (%)	Val (%)
MIMIC-III Full	47719	1631	3372	100	3.4	7.0
MIMIC-III TOP50	8067	1730	1574	70.9	15.2	13.8
MIMIC-II	19392	2282	1141	94.4	5.6	11.1

MHLAT järjestelmää verrattiin samoilla aineistoilla käytettyihin moniluokkaluokittelijoihin: CAML [40] ja sen muunnos DR-CAML [40], MultiResCNN [42], HyperCore [43], LAAT ja sen lisäosa JointLAAT [44], ISD [41], MSATT-KG [45], EffectiveCAN [46], MARN [47], RAC [48], MDBERT [49] ja Longformer-DLAC [50]. Järjestelmistä kuusi viimeksi mainittua on mukana vain MIMIC-III aineistolla tehdyssä tutkimuksessa ja Longformer-DLAC vain MIMIC-III TOP50. Tuloksista tässä kirjallisuuskatsauksessa esitellään vain kahden parhaan järjestelmän tulokset, täydelliset tulossivut löytyvät tutkimuksesta [35]. Tuloksia käydään läpi tarkemmin luvussa 4 aliluvussa 4.1.1.

Taulukko 3.3: MIMIC-III koko tietoaineisto: Järjestelmien vertailu

Mikro (%)	Paras vertailukohta	MHLATT [33]
$F1$	58.9 (EffectiveCAN [46])	59.1
Makro (%)		
$F1$	12.7 (RAC [48])	11.2
$P@K$		
P@8	75.8 (LAAT [46])	75.9

Taulukko 3.4: MIMIC-III TOP50: Järjestelmien vertailu

Mikro (%)	Paras vertailukohta	MHLAT [33]
$F1$	73.5 (HiLAT [39])	73.9
AUC	95.0 (HiLAT [39])	95.1
Makro (%)		
$F1$	69.0 (HiLAT [39])	69.2
AUC	93.5 (ISD [41])	93.1
$P@K$		
P@8	68.2 (ISD [41])	68.7

Taulukko 3.5: MIMIC-II: Järjestelmien vertailu

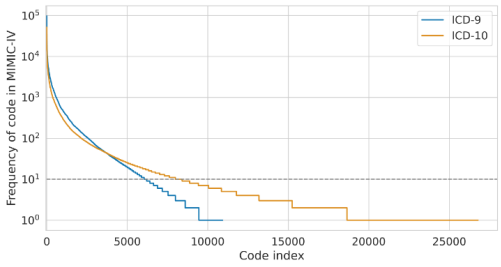
Mikro (%)	Paras vertailukohta	MHLAT [41]
$F1$	49.8 (ISD [41])	51.0
Makro (%)		
$F1$	10.1 (ISD [41])	8.9
$P@K$		
P@8	56.4 (ISD [41])	57.1

3.5 ICD-10

6. päivä tammikuuta 2023 julkaistiin uusi MIMIC-IV tietokanta, jota ei ole aikaisemmin käytetty automaattisen ICD-koodin määrittämiseen. Tässä aineistossa potilasasiakirjat ovat myös koodattu ICD-10 versiolla. Osana kriittistä tutkimusta Edin ym. julkaisivat uudet suositellut parametrit ja suorittivat uudet ajot jo olemassa olevilla malleilla. Tässä osiossa työtä keskitytään MIMIC-IV aineistolla ICD-10 koodien tuloksiin Edin ym. suorittamana jo muiden tutkimusryhmien aikaisemmin esitetyillä malleilla. MIMIC-IV ICD-10 vahvistus- ja testausaineisto käsiteltiin Edin ym. toimesta niin, ettei sellaisia koodeja sisällytetty testaus- tai validointiaineistoon joilla oli alle 100 esimerkkiä koulutusaineistossa. Tämän lisäksi aineistossa keskityttiin puumaisen hierarkian ylimpiin koodeihin, eikä tarkennuksiin [51].

Taulukko 3.6: MIMIC-IV ICD-10 tietoaineisto [51]

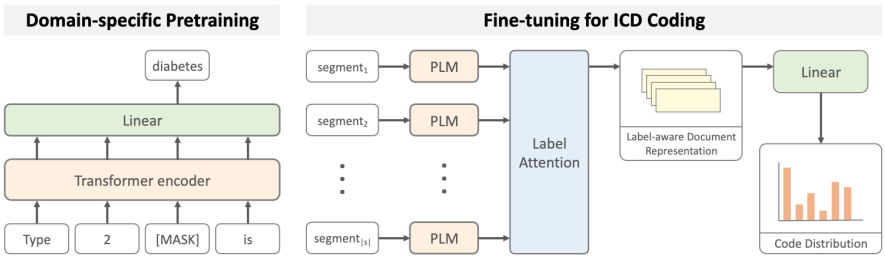
	MIMIC-IV
Potilasasiakirjoja	122279
Luokkien lkm.	7942
Luokkaa per dokumentti (ka.)	14
Sanaa per dokumentti (ka.)	1492



Kuva 3.3: Koodien jakauma MIMIC-IV [51]

Kuten aliluvussa 3.4 myös tälle koodiversiolle parhaat tulokset tuottaa esikoulutettuihin kielimalleihin perustuva järjestelmä PLM-ICD [25]. Huang ym. kertoo tutkimuksessaan, että koska yleiset kielimallit ovat koulutettuja suurella määrällä tekstiaineistoja, ei niissä käytetyt korpukset sisällä lääkinnällisiä termejä tai poti-

lasasiakirjoja [25]. Edellä mainitusta syystä PLM-ICD on esikoulutettu käyttämällä seuraavia kielimalleja: BioBERT [52], PubMedBERT [53] ja RoBERTa-PM [54]. Myös PLM-ICD käyttää samankaltaista lähestymistapaa pitkien lähdeaineistojen ongelmien kanssa kuin luvussa 3.4 esitelty MHLAT. Lähdedokumentti käsitellään ja ositellaan ja näille osajoukoille lasketaan erikseen kielimallien avulla piirteet. Nämä piirteet yhdistetään edustamaan koko lähdedokumenttiä. PLM-ICD käyttää samaa lähestymistapaa luokkien määritelmien piirteiden koostamiseen kuin Vu ym. LAAT [44] järjestelmä. Näillä kahdella kuvauksella lasketaan lopulliset lähdedokumenttikohtaiset luokka-ennusteet [25].



Kuva 3.4: PLM-ICD järjestelmäkuvaus [25]

Edin ym. tekemän tutkimuksen järjestelmien vertailu on alla. Tuloksista tässä kirjallisuuskatsauksessa esitellään vain kahden parhaan järjestelmän tulokset, täydelliset tulossivut löytyvät tutkimuksesta [51]. Tuloksia käydään läpi tarkemmin luvussa 4 aliluvussa 4.1.2.

Taulukko 3.7: MIMIC-IV: Järjestelmien vertailu

Mikro (%)	Paras vertailukohta	PLM-ICD [25]
$F1$	57.9 (LAAT [44])	58.5
AUC	99.0 (LAAT [44])	99.2
Makro (%)		
$F1$	21.1 (MultiResCNN [42])	21.1
AUC	95.4 (LAAT [44])	96.6
$P@K$		
P@8	68.9 (LAAT [44])	69.9

4 Yhteenveto

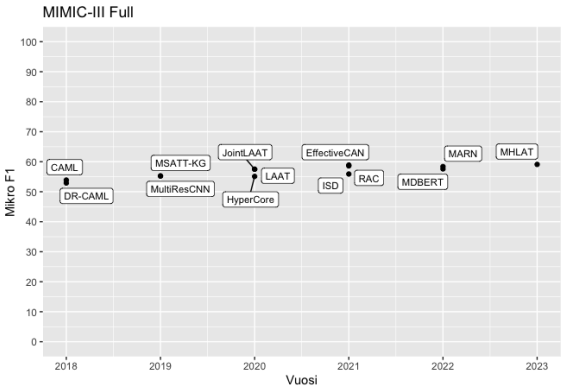
4.1 Päätelmät

Yhteenvetona ICD-koodin automaattinen määrittämänen luonnollisen kielen käsittelyn menetelmillä ei kirjallisuuskatsauksen perusteella ole mitenkään triviaali moniluokittelutehtävä. Tutkimusryhmillä vaikuttaa olevan yhteisymmärrys siitä mitkä järjestelmien suorituskykyä tällä hetkellä rajoittaa. Ongelmat ovat luokkien suuri lukumäärä, koulutusdatan puutteellisuus harvinaisimmille sairauksille ja käsiteltävien tokenien lukumäärää rajoittavat suorituskykyongelmat.

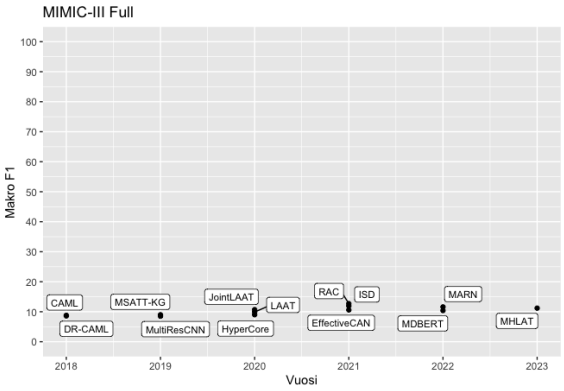
4.1.1 ICD-9

Tutkimuksen tuloksista 3.3, 3.4, 3.5 voidaan todeta kehitystä edellisiin järjestelmiin. Toisaalta kehitystä ei osassa aineistoja ole kuten taulukossa 3.3 oleva makro-F1 arvo jonka nojalla RAC on pystynyt 12.7% ja MHLAT 11.2% suoritukseen. Tätä Duan ym. perustelee mallin sopimattomuudella erityisen pienellä todennäköisyydellä esiintyvien koodien tunnistamiseen [33]. Kuvissa 4.1 ja 4.2 sekä 4.5 ja 4.6 käy selkeästi ilmi mitä tapahtuu suorituskyvylle mikro F1 arvosta makro F1 arvoon.

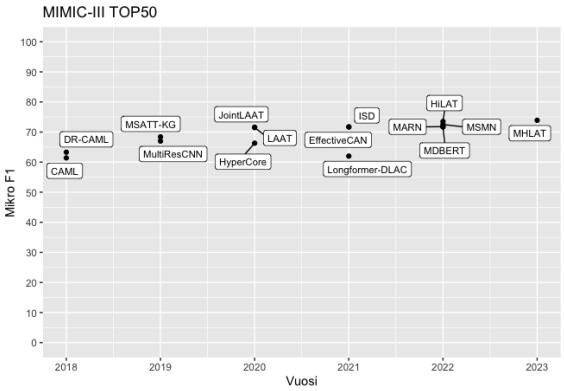
Edin ym. kirjoitti yleistä tutkimusasetelmaa koskevista haasteista, tämän vuoksi lähes kaikissa julkaistuissa makro-F1 tuloksissa on eroavaisuuksia heidän suorittamien uusintatestien kanssa [51]. Erot pohjautuvat MIMIC-III datasetin testiosuuteen, jossa testiaineistoon kuulumattomat mutta lähtöaineistossa olevat luokat



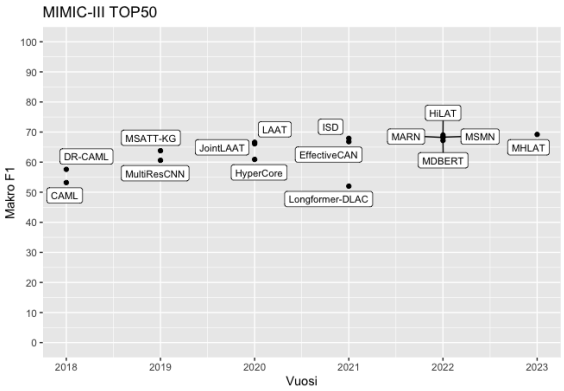
Kuva 4.1: MIMIC-III Full Mikro F1 [33]



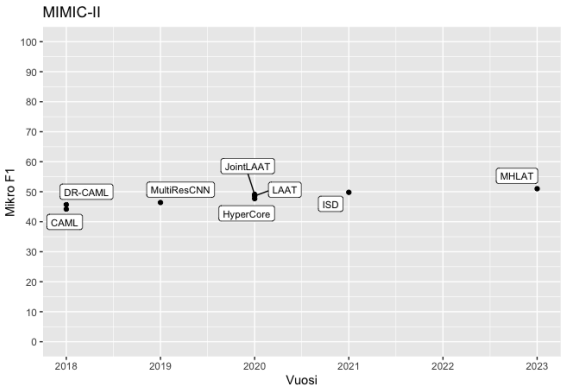
Kuva 4.2: MIMIC-III Full Makro F1 [33]



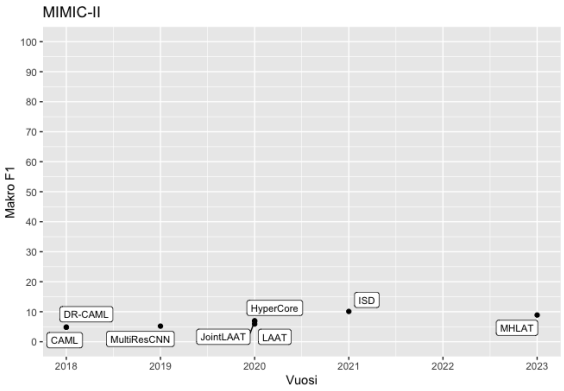
Kuva 4.3: MIMIC-III TOP50 Mikro F1 [33]



Kuva 4.4: MIMIC-III TOP50 Makro F1 [33]



Kuva 4.5: MIMIC-II Mikro F1 [33]

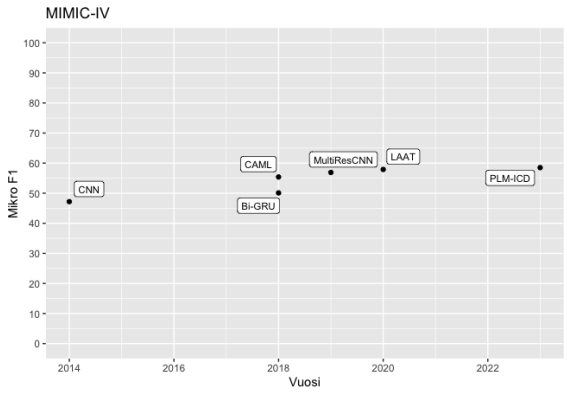


Kuva 4.6: MIMIC-II Makro F1 [33]

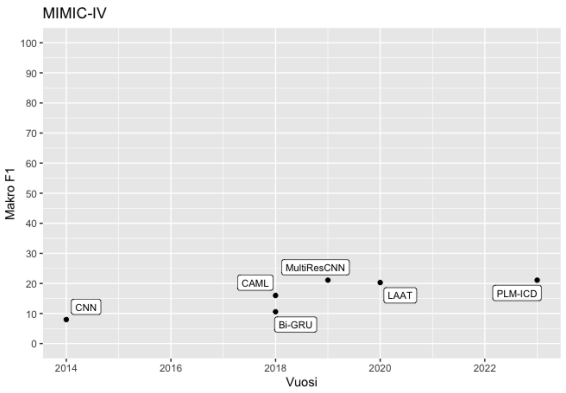
asetetettiin 0:ksi. Koska testiaineistosta puuttui 54% luokista, on makro-F1 arvon maksimi vain 46% joka Edin ym. mukaan johti liian pieneen makro-F1 arvoon [51].

4.1.2 ICD-10

Esikoulutettuja kielimalleja hyväksikäyttävä PLM-ICD suoriutui Edin ym. tekemä tukimuksen mukaan parhaiten. Tuloksista joista löytyvät kaksi parasta järjestelmää taulukossa 3.7 ja jonka graafinen esitys löytyy kuvista 4.1.2 sekä 4.8. Näistä on nähtävissä, että myöskään nämä järjestelmät eivät kovin korkeaan suorituskykyyn yllä. Mikro F1 arvon yltäessä 58.5% ja Makro F1 21.1%. Mikro F1 arvon ollessa marginaalisesti korkeampi kuin LAAT [44], ja makro F1 arvon ollessa sama kuin MultiResCNN [42]. Toisaalta tämän tutkimuksen lähdeaineiston muuttuneen käsittelyn ja tulosten laskennan myötä on havaittavissa makro F1 arvoissa nousua kuvassa 4.8 verraten edeltävien datasettien kanssa kuvissa 4.6 ja 4.2.



Kuva 4.7: MIMIC-IV Mikro F1 [51]



Kuva 4.8: MIMIC-IV Makro F1 [51]

4.1.3 ICD-11

ICD-koodausjärjestelmä on päivittymässä versioon 11. Kirjallisuuskatsausta tehdessä löytyi tälle koodausjärjestelmälle hyvin vähäisiä määriä tietoa, eikä tieteellisesti uskottavista lähteistä löytynyt yhtään vertaisarvioitua tutkimusta järjestelmän tälle

versiolle. ICD-11 versio tuo tullessaan koodien klusterointi järjestelmän. Nykyisessä ICD-10 versiossa koodausjärjestelmä sisälsi esimääriteltyjä termejä diagnooseille, kun taas ICD-11 on pohjautuu päädiagnoosiin ja koodiklustereihin, jonka perusteella pystytään esittämään riippuvuussuhteita diagnoosikoodien välillä. Tämän ryhmittelyn lisäksi voidaan antaa vielä lisätietoja potilaan tilasta. Eastwood ym. esittelevät tarkemmin koodijärjestelmien eroja artikkelissaan [55]. Kuitenkin uudessa järjestelmässä voidaan kuvata yhdellä koodiryhmittymällä samaa asiaa, jolle vaadittiin edellisessä koodausjärjestelmässä useampi diagnoosikoodi [55].

4.2 Pohdinta

Tutkimusaineistoissa julkaistuissa mittareissa ei eksplisiittisesti tuoda julki järjestelmien sisäisen herkkyyden ja tarkkuuden suorituskykyjen tasoja. Kliinisessä NLP:ssä käsiteltävät lähde-aineistot yleisesti koskevat ihmisiä ja heidän elämää. Tällöin myös oikealla positiivisella, oikealla negatiivisella, väärällä positiivisella ja väärällä negatiivisella luokalla voi olla suuri vaikutus yksilön elämään. Tällöin myös herkkyys ja tarkkuus saavat suuren painoarvon jolloin myös näiden arvojen ilmoittaminen tutkimusten tulosten yhteydessä olisi perusteltua. F1-arvon toisaalta voidaan nähdä yleisenä mallin suorituskykyä mittaavana suureena jolla malleja voidaan vertailla keskenään. Tässä kontekstissa järjestelmän sisäinen herkkyys tarkoittaa rajaa jolla luokka eli ICD-koodi asetetaan kuvaamaan kyseistä potilasasiakirjaa. Korkean herkkyyden tavoittelu voi johtaa siihen, että toisaalta tunnistetaan enemmän tapauksia, mutta virheellisten positiivisten diagnoosien lukumäärä kasvaa. Virheelliset positiiviset diagnoosit voivat johtaa turhiin lisätutkimuksiin ja täten lisäkuluihin sairaanhoidossa. Vastapainona korkean tarkkuuden tavoittelu voi johtaa siihen, että järjestelmä jättää luokittelematta positiiviseksi tapauksia jotka oikeasti olisivat positiivisia. Tämä voisi johtaa hoitamatta jääneisiin tapauksiin ja pahimmissa tapauksissa vaikuttaa potilaan elämään suuresti.

Automaattisen ICD-koodin määrittämiseen julkaistujen tutkimusten ja järjestelmien vertailu keskenään ei ollut mitenkään helppo tehtävä. Osa tutkimuksista oli tehty tietyn sairaalan omalla aineistolla, jolloin on mahdotonta päästä varmentamaan. Myöskin julkaistuissa tutkimuksissa vertailuun valitut järjestelmät ja vertailuasetelmat kyseisiin järjestelmiin ei aina avautuneet lukijalle. Tutkimuksissa pystyttiin osoittamaan suorituskyvyn kasvua muuttamalla hieman vertailutasoina käytettyjen järjestelmien lähestymistä. Lähteitä etsiessä löytyi myös tutkimuksia joiden todenperäisyyttä ei pystynyt ristiinvarmentamaan mistään muusta lähteestä ja nämä lähteet on jätetty huomioimatta kirjallisuuskatsausta tehdessä. Toisaalta tämä kilpailuasetelma vääjäämättä johtaa kuitenkin parempien järjestelmien kehittämiseen ja myös tuottaa lopputuloksenaan myös potilastyöhön implementoitavia järjestelmiä.

4.3 Jatkotutkimus

Kirjallisuuskatsausta aloittaessa oli olettamus, että myös Suomessa tehtäisiin tämänkaltaista tutkimustyötä. Tälle hypoteesille ei kuitenkaan löytynyt avoimista tietoarkistoista vahvistusta, voi olla että terveydenhuollon toimijoiden omissa tutkimusosastoissa tälläistä tehdään. Kuitenkaan tästä ei löytynyt akateemista tutkimusmateriaalia. Koska julkaistujen tutkimusten perusteella parhaiten luokittelutehtävästä suorituu BERT-pohjaiset järjestelmät, olisi mielenkiintoista nähdä miten tämä toimisi suomenkielisen aineiston kanssa. Laskentakyvyn ja muistikapasiteettien kasvaessa käsiteltävien tokenien lukumäärää pystytään tulevaisuudessa kasvattamaan joka avaa mahdollisuuksia parempaan suorituskykyyn.

Lähdeluettelo

[1] Terveyden ja hyvinvoinnin laitos, *Tautiluokitus ICD-10*, 2011. (viitattu 20. 04. 2024).

[2] D. Khurana, A. Koli, K. Khatter ja S. Singh, "Natural language processing: state of the art, current trends and challenges", *Multimedia Tools and Applications*, vol. 82, nro 3, s. 3713–3744, 2023. DOI: 10.1007/s11042-022-13428-4. (viitattu 02. 02. 2024).

[3] P. M. Nadkami, L. Ohno-Machado ja W. W. Chapman, "Natural language processing: an introduction", *Journal of the American Medical Informatics Association*, vol. 18, nro 5, s. 544–551, 2011. DOI: 10.1136/amiajnl-2011-000464. (viitattu 03. 02. 2024).

[4] Y. Bengio, R. Ducharme, P. Vincent ja C. Jauvin, "A Neural Probabilistic Language Model", *Journal of Machine Learning Research*, vol. 3, s. 1137–1155, 2003. (viitattu 03. 02. 2024).

[5] R. Caruana, "Multitask Learning", *Machine Learning*, vol. 28, s. 41–75, 1997. DOI: 10.1023/A:1007379606734. (viitattu 23. 02. 2024).

[6] T. Mikolov, K. Chen, G. Corrado ja J. Dean, "Efficient Estimation of Word Representations in Vector Space", 2013. arXiv: 1301.3781 [cs.CL]. (viitattu 28. 01. 2024).

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado ja J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", 2013. DOI: 10.48550/arXiv.1310.4546. (viitattu 28. 01. 2024).

[8] I. Sutskever, O. Vinyals ja Q. V. Le, "Sequence to Sequence Learning with Neural Networks", *Advances in Neural Information Processing Systems*, vol. 27, 2014. (viitattu 12. 02. 2024).

[9] N. T. Vu, H. Adel, P. Gupta ja H. Schütze, "Combining Recurrent and Convolutional Neural Networks for Relation Classification", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, s. 534–539, 2016. DOI: 10.18653/v1/N16-1065. (viitattu 09. 05. 2024).

[10] V. Ashish, S. Noam, P. Niki et al., "Attention Is All You Need", 2017. DOI: 10.48550/arXiv.1706.03762. (viitattu 28. 01. 2024).

[11] J. Devlin, M.-W. Chang, K. Lee ja K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018. DOI: 10.48550/arXiv.1810.04805. (viitattu 24. 01. 2024).

[12] TurkuNLP, *TurkuNLP kotisivu*. (viitattu 21. 01. 2024).

[13] TurkuNLP, *TurkuNLP FinBERT github*. (viitattu 21. 01. 2024).

[14] A. Virtanen, J. Kanerva, R. Ilo et al., "Multilingual is not enough: BERT for Finnish", 2019. arXiv: 1912.07076 [cs.CL]. (viitattu 23. 02. 2024).

[15] G. Chen, S. Ma, Y. Chen et al., "Zero-Shot Cross-Lingual Transfer of Neural Machine Translation with Multilingual Pretrained Encoders", *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, s. 15–26, 2021. DOI: 10.18653/v1/2021.emnlp-main.2. (viitattu 24. 02. 2024).

[16] A. Parnami ja M. Lee, "Learning from Few Examples: A Summary of Approaches to Few-Shot Learning", 2022. DOI: 10.48550/arXiv.2203.04291. (viitattu 23. 02. 2024).

[17] C. Harrison ja C. Sidey-Gibbons, "Machine learning in medicine: a practical introduction to natural language processing", *BMC Medical Research Methodology*, vol. 21, nro 1, s. 158, 2021. DOI: 10.1186/s12874-021-01347-1. (viitattu 30. 04. 2024).

[18] R. Grishman, N. Sager, C. Raze ja B. Bookchin, "The linguistic string parser", *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, s. 427–434, 1973. DOI: 10.1145/1499586.1499693. (viitattu 24. 01. 2024).

[19] N. Sager, M. Lyman, C. Bucknall, N. Nhan ja L. J. Tick, "Natural language processing and the representation of clinical data", *Journal of the American Medical Informatics Association*, vol. 1, nro 2, 1994. (viitattu 23. 02. 2024).

[20] C. Friedman, J. J. Cimino ja S. B. Johnson, "A conceptual model for clinical radiology reports", *Proceedings / the ... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, s. 829–33, 1993. (viitattu 28. 01. 2024).

[21] H. Wu, M. Wang, J. Wu et al., "A survey on clinical natural language processing in the United Kingdom from 2007 to 2022", *npj Digital Medicine*, vol. 5, nro 1, s. 186, 2022. DOI: 10.1038/s41746-022-00730-6. (viitattu 28. 01. 2024).

[22] K. Rannikmäe, H. Wu, S. Tominey, W. Whiteley, N. Allen ja C. Sudlow, "Developing automated methods for disease subtyping in UK Biobank: an exemplar study on stroke", *BMC Medical Informatics and Decision Making*, vol. 21, nro 1, s. 191, 201. (viitattu 01. 02. 2024).

[23] H. Wu, G. Toti, K. I. Morley et al., "SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research", *Journal of the American Medical Informatics Association*, vol. 25, nro 5, s. 530–537, 2018. DOI: 10.1093/jamia/ocx160. (viitattu 09. 05. 2024).

[24] C. Yan, X. Fu, X. Liu et al., "A survey of automated International Classification of Diseases coding: development, challenges, and applications", *Intelligent Medicine*, vol. 2, nro 3, s. 161–173, 2022. DOI: 10.1016/j.imed.2022.03.003. (viitattu 30. 04. 2024).

[25] C.-W. Huang, S.-C. Tsai ja Y.-N. Chen, "PLM-ICD: Automatic ICD Coding with Pretrained Language Models", *Proceedings of the 4th Clinical Natural Language Processing Workshop*, s. 10–20, 2022. DOI: 10.18653/v1/2022.clinicalnlp-1.2. (viitattu 01. 03. 2024).

[26] Terveyden ja hyvinvoinnin laitos, *ICD-11 -diagnoosiluokituksen käyttöönotto*, 2023. (viitattu 21. 04. 2024).

[27] D. Cartwright, "ICD-9-CM to ICD-10-CM Codes: What? Why? How?", *Adv Wound Care (New Rochelle).*, vol. 10, nro 2, 2013. DOI: 10.1089/wound.2013.0478. (viitattu 30. 04. 2024).

[28] MIT Laboratory for Computational Physiology, *Medical Information Mart for Intensive Care*, 2024. (viitattu 21. 04. 2024).

[29] J. Feng, R. Zhang, D. Chen, L. Shi ja Z. Li, "Automated generation of ICD-11 cluster codes for Precision Medical Record Classification", *International Journal of Computers Communications & Control*, vol. 19, nro 1, 2024. (viitattu 28. 03. 2024).

[30] T. Fawcett, ”An introduction to ROC analysis”, *Pattern Recognition Letters*, vol. 27, nro 8, s. 861–874, 2006. DOI: 10.1016/j.patrec.2005.10.010. (viitattu 05. 05. 2024).

[31] C. M. Bishop ja H. Bishop, *Deep Learning: Foundations and Concepts*. Springer, 2024. (viitattu 05. 05. 2024).

[32] World Health Organization, *International Statistical Classification of Diseases and Related Health Problems (ICD)*, 2024. (viitattu 28. 03. 2024).

[33] J. Duan, H. Jiang ja Y. Yu, ”MHLAT: Multi-Hop Label-Wise Attention Model for Automatic ICD Coding”, *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. DOI: 10.1109/ICASSP49357.2023.10096122. (viitattu 09. 05. 2024).

[34] I. Meta Platforms, *Extreme Multi-Label Classification*, 2024. (viitattu 09. 05. 2024).

[35] Y. Wu, Z. Chen, X. Yao, X. Chen, Z. Zhou ja J. Xue, ”JAN: Joint attention networks for automatic ICD coding”, *IEEE journal of biomedical and health informatics*, vol. 26, nro 10, s. 5235–5246, 2022. (viitattu 11. 02. 2024).

[36] A. Rios ja R. Kavuluru, ”Few-shot and zero-shot multi-label learning for structured label spaces”, *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, vol. 2018, s. 3132, 2018. (viitattu 30. 04. 2024).

[37] M. Zaheer, G. Guruganesh, K. A. Dubey et al., ”Big bird: Transformers for longer sequences”, *Advances in neural information processing systems*, vol. 33, s. 17 283–17 297, 2020. (viitattu 09. 05. 2024).

[38] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov ja Q. V. Le, ”Xlnet: Generalized autoregressive pretraining for language understanding”, *Advances in neural information processing systems*, vol. 32, 2019. (viitattu 09. 05. 2024).

[39] L. Liu, O. Perez-Concha, A. Nguyen, V. Bennett ja L. Jorm, "Hierarchical label-wise attention transformer model for explainable ICD coding", *Journal of Biomedical Informatics*, vol. 133, s. 104 161, 2022. DOI: 10.1016/j.jbi.2022.104161. (viitattu 09. 05. 2024).

[40] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun ja J. Eisenstein, "Explainable Prediction of Medical Codes from Clinical Text", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. DOI: 10.18653/v1/N18-1100. (viitattu 11. 02. 2024).

[41] T. Zhou, P. Cao, Y. Chen et al., "Automatic ICD Coding via Interactive Shared Representation Networks with Self-distillation Mechanism", *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, s. 5948–5957, 2021. DOI: 10.18653/v1/2021.acl-long.463. (viitattu 08. 05. 2024).

[42] F. Li ja H. Yu, *ICD Coding from Clinical Text Using Multi-Filter Residual Convolutional Neural Network*, 2019. arXiv: 1912.00862 [cs.CL]. (viitattu 09. 05. 2024).

[43] P. Cao, Y. Chen, K. Liu, J. Zhao, S. Liu ja W. Chong, "HyperCore: Hyperbolic and Co-graph Representation for Automatic ICD Coding", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, s. 3105–3114, 2020. DOI: 10.18653/v1/2020.acl-main.282. (viitattu 05. 05. 2024).

[44] T. Vu, D. Q. Nguyen ja A. Nguyen, "A Label Attention Model for ICD Coding from Clinical Text", *Proceedings of the Twenty-Ninth International Joint*

Conference on Artificial Intelligence, 2020. DOI: 10.24963/ijcai.2020/461. (viitattu 08. 05. 2024).

[45] X. Xie, Y. Xiong, P. S. Yu ja Y. Zhu, "EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation", *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, s. 649–658, 2019. DOI: 10.1145/3357384.3357897. (viitattu 08. 05. 2024).

[46] Y. Liu, H. Cheng, R. Klopfer, M. R. Gormley ja T. Schaaf, "Effective Convolutional Attention Network for Multi-label Clinical Document Classification", *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, s. 5941–5953, 2021. DOI: 10.18653/v1/2021.emnlp-main.481. (viitattu 09. 05. 2024).

[47] W. Sun, S. Ji, E. Cambria ja P. Marttinen, "Multitask Balanced and Recalibrated Network for Medical Code Prediction", *ACM Transactions on Intelligent Systems and Technology*, vol. 14, nro 1, 2022. DOI: 10.1145/3563041. (viitattu 09. 05. 2024).

[48] B.-H. Kim ja V. Ganapathi, *Read, Attend, and Code: Pushing the Limits of Medical Codes Prediction from Clinical Notes by Machines*, 2021. arXiv: 2107.10650 [cs.CL]. (viitattu 09. 05. 2024).

[49] N. Zhang ja M. Jankowski, *Hierarchical BERT for Medical Document Understanding*, 2022. arXiv: 2204.09600 [cs.CL]. (viitattu 09. 05. 2024).

[50] M. Feucht, Z. Wu, S. Althammer ja V. Tresp, "Description-based Label Attention Classifier for Explainable ICD-9 Classification", *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, s. 62–66, 2021. DOI: 10.18653/v1/2021.wnut-1.8.

[51] J. Edin, A. Junge, J. D. Havtorn et al., "Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study", *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023. DOI: 10.1145/3539618.3591918. (viitattu 09. 05. 2024).

[52] J. Lee, W. Yoon, S. Kim et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining", *Bioinformatics*, vol. 36, nro 4, s. 1234–1240, 2019. DOI: 10.1093/bioinformatics/btz682. (viitattu 09. 05. 2024).

[53] Y. Gu, R. Tinn, H. Cheng et al., "Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing", *ACM Transactions on Computing for Healthcare*, vol. 3, nro 1, s. 1–23, 2021. DOI: 10.1145/3458754. (viitattu 09. 05. 2024).

[54] P. Lewis, M. Ott, J. Du ja V. Stoyanov, "Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art", *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, 2020. DOI: 10.18653/v1/2020.clinicalnlp-1.17. (viitattu 09. 05. 2024).

[55] C. A. Eastwood, D. A. Southern, S. Khair et al., "Field testing a new ICD coding system: methods and early experiences with ICD-11 Beta Version 2018", *BMC Research Notes*, vol. 15, nro 343, 1 2022. DOI: 10.1186/s13104-022-06238-2. (viitattu 28. 03. 2024).