Markus Koskela

Automaattinen tekstien lyhentäminen chatboteille

Tietotekniikan pro gradu -tutkielma

14. toukokuuta 2023

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Markus Koskela

Yhteystiedot: markus.j.koskela@student.jyu.fi

Ohjaajat: Joonas Hämäläinen ja Tommi Mikkonen

Työn nimi: Automaattinen tekstien lyhentäminen chatboteille

Title in English: Automatic text shortening for chatbots

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 84+4

Tiivistelmä: Nykyisin ihmiset eivät lyhyiden viestien kulttuurin aikana jaksa usein lukea pitkiä tekstikokonaisuuksia kokonaan. Tekstin lyhentämiseen liittyvät menetelmät auttavat ihmisiä löytämään parhaimmillaan olennaisen tiedon nopeasti. Chatbotit ovat toimineet Covid-19-pandemian aikana tiedonjakajina ja antaneet emotionaalista tukea. Ne voivat toimia myös muun muassa asiakaspalvelijoina. Käyttäjäkokemuksen nimissä chatbotin tulosteen tulisi olla napakka ja informatiivinen. Automaattisten yhteenvetojen arvioinnissa abstrahoiduille yhteenvedoille on suositeltavaa tarkistaa, kuinka automaattisesti mitataan niiden laatua.

Avainsanat: chatbot, keskusteluagentti, transformer, koneoppiminen, NLP, yhteenvedon tekeminen, LSA, aihemallinnus, LDA

Abstract: In today's culture of short messages people don't often read long blocks of text entirely. Text shortening methods help at best people to find the essential information quickly. Chatbots have served as information distributors during the Covid-19 pandemic, providing emotional support, too. They can also serve as customer service agents, among other things. For the better user experience, the utterances of the chatbots should be concise and informative. When evaluating automatic abstractive summaries, it is recommended to revise the way in which the quality of the summaries is automatically measured.

Keywords: chatbot, conversational agent, transformer, machine learning, NLP, text summarization, LSA, topic modeling, LDA

Termiluettelo

Autoregressiivinen	Tulokset regressoidaan saman aikasarjan aiempiin arvoihin.
Sanapussi	NLP:n tekniikka, joka huomioi sanat ja niiden määrän, mutta ei säilytä sanajärjestystä (engl. bag of words).
GPT	Generative Pre-trained Transformer. Ottaa syötteenä dokumentin ja jatkaa sitä parhaalla mahdollisella tavalla.
Koheesio	Tekstin sidoksisuus. Eksplisiittisesti käsittää tekstinosia yhdistävät kieliopilliset sidosteet. Implisiittisesti käsittää merkityssuhteet, teeman- ja informaatiokulun, argumentoinnin ja tekstin kokonaisjäsentelyn.
Konvoluutio	Matemaattisesti ottaen muuttaa kaksi funktiota yhdeksi funktioksi.
Korpus	Tarkkaan määritelty kirjoitetun kielen tekstikokoelma.
Kosinisamankaltaisuus	Mittaa numeerisiksi vektoreiksi muutetun tekstin samankaltaisuutta vertaamalla vektorien pituutta ja niiden välistä kulmaa.
Leksikaalinen	Sanoihin olennaisesti liittyvä.
Longformer	Lineaarisesti skaalautuva transformer-malli pitkien, jopa tuhansien sanojen, sekvenssien käsittelyyn.
N-grammit	Vierekkäisten *n*:n elementtien sarja tekstinäytteessä.
NLP	Luonnollisen kielen prosessointi (engl. Natural Language Processing).
Ohjattu oppiminen	Koneoppimisessa menetelmä, jossa opetusdata koostuu nimetyistä esimerkeistä (engl. supervised learning).
Ohjaamaton oppiminen	Koneoppimisessa menetelmä, jossa opetusdatassa ei ole nimettyä esimerkkidataa (engl. unsupervised learning).
Ristientropia	Käytetään usein mittaamaan, kuinka hyvin estimoitujen luokkien todennäköisyyksien joukko vastaa kohdeluokkia (engl. cross entropy).
Transduktiomalli	Mallissa tekstiä käytetään sitä muuntamalla.
Whitespace-merkit	Merkit, joilla ei ole näkyvää muotoa kuten välilyönti.

Kuviot

Taulukot

Sisällys

1 Johdanto

Chatbotteja on ollut olemassa vuosikymmeniä (Brandtzaeg ja Følstad 2018). Alan Turingia pidetään Caldarini, Jaf ja McGarry 2022 mukaan ensimmäisenä ihmisenä, joka on luonut chatbotin konseptin esittäessään vuonna 1950 kysymyksen ”Voivatko koneet ajatella?”. Turingin esittämässä koeasetelmassa kone on läpäissyt Turingin testin, jos ihminen ei osaa tehdä eroa, keskusteleeko hän ihmisen vai koneen kanssa (Mays 1952). Turingin kuvaus älykkään koneen käyttäytymisestä edustaa yleisesti ymmärretyn chatbotin konseptia (Caldarini, Jaf ja McGarry 2022). Kuitenkin niihin liittyvä teknologia lähti Brandtzaeg ja Følstad 2018 mukaan todella nousuun vasta noin vuonna 2016. Syy suureen uudistuneeseen kiinnostukseen chatbotteja kohtaan on Brandtzaeg ja Følstad 2018 mukaan tekoälyn suuri kehitys ja ihmisten laajamittainen siirtyminen sosiaalisista tietoverkoista sosiaalisen viestinnän sovellusten käyttämiseen kuten Facebook Messenger, Telegram, Slack, Kik ja Viber.

Koronaviruspandemia osoitti, että chatbotit voivat auttaa ihmisiä. Miner, Laranjo ja Kocaballi 2020 mukaan koronaviruspandemian aikana esimerkiksi Maailman terveysjärjestö (WHO) on hyödyntänyt chatbotteja tiedon jakamisessa, käyttäytymisohjeistuksessa ja tarjonnut chatbottien kautta emotinaalista tukea. Amiri ja Karahanna 2022 mukaan lisäksi sosiaalisen etäisyyden vaatimukset viruksen leviämisen hidastamiseksi asettivat lisärajoituksia perinteisten henkilökohtaisten palvelujen käytölle ja vaativat sosiaalisen etäisyyden huomioon ottavia ratkaisuja.

Lisäksi Amiri ja Karahanna 2022 mukaan koronavirusta koskevan tiedon nopeasti muuttuva maailma, johon sisältyi käytäntöjen muutos sekä disinformaation ja väärän tiedon nopea leviäminen aiheuttivat ahdistusta ja hämmennystä. Tämä johti auttavaan puhelimeen soitettujen puheluiden määrän kasvuun. Eräs vastaus näihin ongelmiin käsitti chatbottien käyttöönoton skaalautuvana, helppokäyttöisenä, nopeasti käyttöönotettavana, inhimillisen sosiaalisen etäisyyden mahdollistavana ratkaisuna (Amiri ja Karahanna 2022).

Itseasiassa Nicolescu ja Tudorache 2022 mukaan eräs tärkeä alue chatbottien hyödyntämiseen onkin asiakaspalveluaktiviteetit. Lisäksi Wang, Lin ja Shao 2022 mukaan organisaatiot odottavat, että tekoälyn tukema vähittäiskaupan kulutus kasvaa dramaattisesti. Esimerkiksi

kuluttajien vähittäiskulutuksen chatbottien kautta ennustetaan nousevan Wang, Lin ja Shao 2022 mukaan 142 miljardiin dollariin vuoteen 2024 mennessä, kun se vuonna 2019 oli vain 2.8 miljardia dollaria. Näin ollen tekoäly tarjoaa organisaatioille loistavat mahdollisuudet parantaa suorituskykyään parantamalla työn suunnittelua ja liiketoimintaa (Wang, Lin ja Shao 2022). Chatbottien merkityksen liike-elämässä ennustetaan siis lisäävän huomattavasti jalansijaansa.

Myös koulutuksessa voi hyödyntää chatbotteja. Tsivitanidou ja Ioannou 2021 mukaan chatbot-ratkaisu voi tarjota tukea erilaisiin opetus- ja oppimistehtäviin arkkitehtuuristaan ja käytetystä tekniikasta riippuen. He jatkavat, että toisaalta chatbotit voivat helpottaa opiskelijoiden oppimista ja tarjota opiskelijoille vuorovaikutteisia oppimiskokemuksia ja jopa sujuvoittaa toisen asteen opiskelijoiden siirtymistä yliopistoympäristöön tai lisätä yliopisto-opiskelijoiden määrää. Toisaalta chatbotit voivat keventää ohjaajien työtaakkaa toimimalla opettajan assistentteina ja ottamaan tutorin roolin esimerkiksi vastaamalla opiskelijan kysymyksiin ja usein kysyttyihin kysymyksiin sekä lähettämällä opiskelijoille muistutuksia tulevista määräajoista tai jopa suorittamalla online-arvioinnit (Tsivitanidou ja Ioannou 2021).

ChatGPT:n kaltaiset tekoälypohjaiset chatbotit voivat hyödyntää laajoja tietovarantoja vastatakseen nopeasti kysymyksiin, jotka koskevat parasta mahdollista tutkimusta tietyissä kliinisissä tilanteissa. Ongelmana tulee esiin mahdollisuus käyttää ChatGPT:tä tieteellisten töiden väärentämiseen, mikä on herättänyt huolta tiedeyhteisössä. (Shen ym. 2023).

1.1 Tutkimuskysymykset

Sarkar 2019 hahmottaa nykyaikaa aikakautena, jolloin ei enää odoteta sanomalehtiä, jotta saataisiin tietoa maailman tapahtumista. Hänen mukaan ihmiset nykyään käyttävät sosiaalista mediaa, joka on luonut lyhyiden viestien kulttuurin. Hän jatkaa, että ihmisillä on yleensä lyhyt keskittymiskyky, joka johtaa siihen, että ihmiset eivät usein jaksa lukea laajoja tekstidokumentteja ja artikkeleita. Chatbotit voivat osaltaan toimia olennaisen tiedon jakajina. Toisaalta ne voivat toimia erilaisina keskustelukumppaneina ihmisille tai esimerkiksi toimia asiakaspalvelijoina.

Tämän pro gradu -tutkimuksen tavoitteena on selvittää chatbottien tulosteiden pituuden ja

laadun problematiikkaa ja tarkastella, voivatko chatbotit itsessään hyödyntää tekstin tiivistämistä. Tutkimuskysymyksiksi ovatkin jalostuneet seuraavassa esitetyt.

1. *Millainen on mielekkään pituinen chatbotin tuloste?*
2. *Mitä menetelmiä on lyhentää tekstiä mielekkään pituisiksi chatbottien tulosteiksi?*
3. *Millaisia keinoja on koneellisesti generoitujen yhteenvetojen automaattiseen laadunarviointiin?*
4. *Voidaanko chatboteissa itsessään hyödyntää tekstin tiivistämistä?*

Aiemmassa esitetyn tutkimustiedon perusteella chatbotit tulevat saamaan olennaisen osan tulevaisuudessa. Siksi chatbottien dialogien toteuttamisen on tärkeää pohjautua tutkittuun tietoon, jotta ne palvelisivat tarkoitustaan mahdollisimman hyvin.

1.2 Tutkimusmenetelmä

Tutkimuskysymyksiä on lähestytty suunnittelutieteellisellä periaatteella. March ja Smith 1995 mukaan suunnittelutieteellä pyritään luomaan teknologialähtöisesti asioita, jotka palvelevat ihmisten tarkoituksia. Heidän mukaan suunnittelutieteellisiä tuotteita on neljää tyyppiä, konstruktioita, malleja, metodeja ja toteutuksia. March ja Smith 1995 mukaan suunnittelutieteessä tarvitaan käsitteitä eli konstruktioita, joilla karakterisoidaan ilmiöitä. He jatkavat, että ilmiöt voidaan yhdistää korkeamman tason rakenteiksi, joita usein kutsutaan malleiksi ja joita käytetään kuvaamaan tehtäviä, tilanteita tai artefakteja. Suunnittelutieteilijät kehittävät myös metodeja, joilla toteutetaan tavoitteellista toimintaa (March ja Smith 1995).

Suunnittelutieteellisen tutkimuksen perusperiaate on Hevner ym. 2004 mukaan se, että tieto ja ymmärrys suunnitteluongelmasta ja sen ratkaisusta hankitaan artefaktin rakentamisessa ja käytössä. Gregor ja Hevner 2013 siteeraavat Wilson 2002 artikkelia pohtiessaan suunnittelutieteen kontribuutiota tietoon. ”Onko se totta? Onko se uutta? Onko se mielenkiintoista?” kysyy tunnettu matemaatikko G. H. Hardy Wilson 2002 mukaan. Viimeinen kysymys on Gregor ja Hevner 2013 mukaan ehkä kaikkein tärkein. Jos siihen vastataan kielteisesti, niin kaksi ensimmäistä kysymystä voidaan jättää huomiotta. Suunnittelutieteen tulosten tulisi siis olla mielenkiintoisia.

Peffers, Tuunanen ja Niehaves 2018 mukaan suunnitteluteoreettisessa tutkimuksessa on neljä tärkeää edellytystä tulosten soveltamiseen, mitkä ovat esitetty seuraavassa.

- Abstraktio: Kunkin artefaktin on oltava sovellettavissa tiettyyn ongelmaluokkaan.
- Omaperäisyys: Jokaisen artefaktin on olennaisesti edistettävä tutkimustiedon kehittymistä.
- Perustelu: Kukin artefakti on perusteltava ymmärrettävällä tavalla ja sen hyöty on oltava validoitavissa.
- Hyöty: Jokaisen artefaktin on tuotettava hyötyä välittömästi tai tulevaisuudessa asianomaisille sidosryhmille (Österle ym. 2011).

Seuraavassa on mukailtu Peffers ym. 2007 esittämä suunnittelutieteen prosessi, jota tämän pro gradu -tutkimuksen aikana on sovellettu.

1. **Tutkimusongelman tunnistaminen ja motivointi.**
2. **Ratkaisun päämäärien määrittely.**
3. **Artefaktin suunnittelu ja kehittäminen.**
4. **Artefaktin käyttäminen ongelman ratkaisemiseen.**
5. **Arviointi.** Tarvittaessa aloitetaan uudelleen kohdasta 2 tai 3.
6. **Julkaisu**.

March ja Smith 1995 mukaan lopuksi suunnittelutieteen tuotteita voidaan ilmentää spesifisiin tuotteisiin, fyysisiin toteutuksiin, jotka on tarkoitettu suorittamaan tiettyjä tehtäviä.

1.3 Tutkielman rakenne

Tutkielmassa esitetään tutkimusaiheen motivointia ja teoriataustaa luvussa 2. Luvussa 3 esitellään koneellisesti generoituihin yhteenvetoihin liittyviä automaattisia metriikoita. Luvussa 4 esitellään modernien yhteenvetomenetelmien käyttämien transformerien arkkitehtuuria ja käydään läpi niiden toimintaperiaatteita. Luvussa 5 esitetään teoriataustaa chatbottien dialogien suunnittelusta niiden tulosteen pituus huomioiden. Luvussa 6 käydään läpi saadut tulokset ja analysoidaan ne. Luvussa 7 pohditaan tutkimuskysymyksiä ja tuloksia sekä lopuksi luvussa 8 tehdään johtopäätökset kaikesta esitetystä.

2 Tekstien lyhentäminen

Automaattisessa tekstin yhteenvetoprosessissa lähdetekstiä lyhennetään tehokkaasti säilyttäen pääajatus (Koh ym. 2022). Tekstistä yhteenvedon tekeminen kategorisoidaan ekstraktiiviseen ja abstrahoivaan yhteenvetoon (Nguyen ym. 2020). Luvussa tehdään katsaus olennaisiin tekniikoihin tekstistä yhteenvedon tekemiseen sekä yksittäisten virkkeiden tiivistämiseen.

2.1 Motivaatio chatboteille

Bathija ym. 2020 ehdottavat työkaluna järjestelmää, jonka avulla voidaan tehdä yhteenveto ja analysoida useita koulutuksellisen sisällön tekstikappaleita ja esittää nämä käyttäjälle interaktiivisena ja älykkäänä keskusteluna chatbotin käyttöliittymän kautta.

Bathija ym. 2020 esittävät työkalun tärkeimmiksi toiminnoiksi seuraavassa mainittuja.

1. **Aihemallinnusta**. Useita tiettyä aihetta koskevia tekstidokumentteja otetaan syötteeksi. Kaikissa näissä dokumenteissa oletetaan olevan tietty määrä aiheita. Jotkut sanat edustavat tiettyä aihetta enemmän kuin toiset. Jokainen lause luokitellaan johonkin aihepiiriin lauseen sanojen perusteella. Näin sisältö voidaan jakaa niin, että sisältö voidaan esittää kiinnostavasti ja myös luoda automaattisesti kysymyksiä, joita tulee kysyä, jotta voidaan varmistua, että käyttäjä on sitoutunut ja ymmärtänyt sisällön oikein.
2. **Semanttista suhdemallinnusta**. Sisäisesti aihe käsittää useita virkkeitä sisältönään. Näiden virkkeiden välillä on erilaisia luontaisia suhteita. Esimerkiksi yksi virke voisi esitellä käsitteen, josta toinen virke voi olla esimerkki. Useilla virkkeillä on osittainen suhde, jos ne listaavat tiettyä käsitettä koskevia kohtia.
3. **Kysymysten luomista**. Jokaista aihemallinnuksen määrittämä aihetta vastaava sisältö analysoidaan. Tästä sisällöstä syntyy kysymyksiä käyttäjille. Kunkin kysymyksen tärkeys sekä kysymysten välinen suhde määritellään. Tätä käytetään keskusteluvirran luomiseen, jotta käyttäjät voivat kattaa kaikki aiheet kysymysten kautta. Sisällöstä luodaan myös mallivastaukset kuhunkin kysymykseen. Kysymykset luodaan käyttämällä sisällöstä ydinvirkkeitä ekstraktoivaa yhteenvetoa tärkeiden lauseiden saamiseksi. Nä-

mä virkkeet muunnetaan sitten kysymyksiksi. Bathija ym. 2020 ehdottavat yhteenvedon tekemiseen Devlin ym. 2018 luomaa transformeria.

4. **Keskustelun kulkua**. Tämä moduuli määrittää tavan, jolla tiedot esitetään käyttäjälle ja auttaa käyttäjää oppimaan uutta sisältöä tehokkaasti. Aihemallinnusvaiheessa sisältö on analysoitu, jotta ymmärretään, mitkä aiheet ovat läsnä ja ovat sisällössä relevantteja.
5. **Vastausten analysointia.** Jokaisen käyttäjän tallennetut vastaukset analysoidaan käyttäjien oppimisen seuraamiseksi. Tämä tehdään heidän oppimiskokemuksensa mukauttamiseksi ja auttamaan heitä peittämään oppimisensa puutteet tai väärinkäsitykset tarjoamalla heille asianmukaista sisältöä.

Kyselypohjainen dokumenttien yhteenveto pyrkii tuottamaan tietystä dokumentista kompaktin ja sujuvan yhteenvedon, joka vastaa dokumenttia kuvaavaa hakukyselyä tai on relevantti sen kannalta. Sovelluksissa kyselypohjainen dokumenttien yhteenveto voi toimia myös tärkeänä koneellisen luetunymmärtämisen alkuvaiheen tehtävänä, kun tarkoituksena on tuottaa vastaus kysymykseen tekstin perusteella. Eräs sovelluskohde ovat chatbotit. (Zhao ym. 2021)

GPT-3:n (Generative Pre-trained Transformer 3) tekstingeneroimisominaisuudet ovat herättäneet Sezgin, Sirrianni, Linwood ym. 2022 mukaan paljon huomiota mahdollisena ratkaisuna muun muassa paranneltujen chatbottien luomiseen. GPT-3 tekee erilaisia tehtäviä riippuen käyttäjän pyynnöstä. Tehtävät ovat esimerkiksi kysymyksiin vastaaminen ja tekstistä yhteenvedon tekeminen (Sezgin, Sirrianni, Linwood ym. 2022).

Sezgin, Sirrianni, Linwood ym. 2022 tuovat esiin hypoteettisen esimerkin GPT-3:n soveltamisesta sairaalaympäristön chatbottiin. Siinä chatbot on liitettynä sairaalan tietoverkkoon yhdistettynä kiireellisen tiedon tekstinyhteenvetopalveluun. Kiireellisten toimenpiteiden lopussa hoidollisista keskusteluista tehdään yhteenveto. Lisäksi, jotta voidaan vähentää koko keskustelun lukemisesta aiheutuvaa kliinistä lisätaakkaa, GPT-3 tekisi yhteenvedon tekstistä ja tallentaisi sen potilaan terveystietoihin. Chatbot olisi osa potilastietojärjestelmää.

Else 2023 kuvailee ChatGPT:tä chatbotiksi, joka luo realistista tekstiä vastauksena käyttäjän kehotuksiin. Se perustuu OpenAI:n GPT-3 kielimalliin (Aydın ja Karaarslan 2022) ja hermoverkkoihin, oppii suorittamaan tehtävän prosessoimalla valtavia määriä ihmisen tuottamaa tekstiä ja julkaistiin 30.11.2022 (Else 2023). Tutkijat ovat käyttäneet ChatGPT:tä muun

muassa kirjallisuudesta yhteenvedon tekemiseen (Dis ym. 2023).

ChatGPT:tä on pyydetty esimerkiksi kirjoittamaan 50 lääketieteellisen tutkimuksen tiivistelmää ja tutkijat vertasivat niitä julkaistuun valikoimaan. Sen jälkeen he vertasivat ChatGPT:n luomia tiivistelmiä alkuperäisiin tiivistelmiin ajamalla ne plagiointitunnistimen ja tekoälytulostunnistimen läpi ja pyysivät ryhmää lääketieteen tutkijoita havaitsemaan väärennetyt tiivistelmät. ChatGPT:n tuottamat tiivistelmät läpäisivät plagiointitarkastimen: alkuperäisyyden mediaani oli 100% eli plagiointia ei havaittu. Tekoälytulosten tunnistin havaitsi 66% generoiduista tiivistelmistä. Ihmistarkistajat tunnistivat virheellisesti 32% generoiduista tiivistelmistä aidoiksi ja 14% aidoista tiivistelmistä ChatGPT:n generoimiksi. (Else 2023)

ChatGPT:n avulla lääkärit voivat antaa lyhyesti tiedot sisällytettävistä erityistiedoista, käsitteitä joita on käsiteltävä tarkemmin ja ohjeita, joita on selitettävä sekä tulostaa muutamassa sekunnissa virallisen kotiutusyhteenvedon. Ne ovat ilmeinen valinta tälle teknologialle, koska niiden muoto on pitkälti standardoitu. (Patel ja Lam 2023)

2.2 Aihemallinnus

Yhteenvedon ja tiedon poimimisen tavoitteena on saada käsitys tärkeimmistä aiheista ja teemoista ja tiivistää laajat tietoasiakirjat muutamaksi ymmärrettäväksi ja luettavaksi riviksi. Tavoitteena on pystyä tekemään tietoon perustuvia päätöksiä lyhyemmässä ajassa. Tähän tavoitteeseen voidaan päästä myös aihemallinnuksella (engl. topic modeling). (Sarkar 2019)

Aihemallit (engl. topic models) ovat tilastopohjaisten algoritmien kehys, joita käytetään tunnistamaan ja mittaamaan piileviä aiheita dokumenteissa (Wesslen 2018). Kalepalli ym. 2020 mukaan aihemallinnuksella voidaan automaattisesti klusteroida valtava määrä dokumentteja, ryhmitellä niiden sanat ja löytää samankaltaiset ilmaisut, jotka karakterisoivat yksittäistä dokumenttia. Opitut aiheet voivat olla Wei ja Croft 2006 mukaan hyödyllisiä käyttöliittymissä *ad hoc* -hakuja varten dokumenteille.

Kherwa ja Bansal 2020 mukaan suosituimmat aihemallinnusalgoritmit ovat ne, jotka osallistuivat tekstianalyysin jokaiseen osa-alueeseen useilla aloilla: LSA (engl. latent semantic analysis), NMF (engl. non-negative matrix factorization, käytetään myös lyhennettä NNMF),

LDA (latent Dirichlet allocation) ja PLSA. PLSA on probabilistinen versio LSA:sta. PLSA on sanapussimalliin perustuva tekstinlouhinnan tekniikka ulottuvuuden vähentämiseen termien semanttisen esiintymisen havaitsemiseksi käyttämällä korpuksen todennäköisyyspohjaista kehystä (Kherwa ja Bansal 2020). Seuraavassa tarkastellaan lähemmin LDA:ta ja LSA:ta, joita on käytetty tämän pro gradu -tutkielman koeasetelmassa.

2.2.1 Toimintaperiaatteet

Dokumenttien tai virkkeiden esikäsittely on erittäin tärkeä vaihe missä tahansa luonnollisen kielen prosessointitehtävässä Anantharaman ym. 2019 mukaan. Heidän mukaan se koostuu seuraavista vaiheista:

- Tokenisaatio (engl. tokenization). Teksti pilkotaan yksittäisiksi sanoiksi.
- Typistäminen (engl. stemming). Sanat muutetaan juurimuotoon. Esimerkiksi *amazing*-sanasta tulee *amaze*.
- Lemmaus (engl. lemmatization). Esimerkiksi sanat *better* ja *best* tulkitaan sanaksi *good*.

Amalia ym. 2017 mainitsevat vielä koko tekstin muuttamisen joko pien- tai suuraakkosiksi. Seuraavana vaiheena Anantharaman ym. 2019 esittävät sanapussi- tai TF-IDF-esitystä olemassa olevalle korpukselle. Näin muodostuu dokumentti-termi-matriisi.

Jones 1972 ehdotti ensimmäisenä TF-IDF-algoritmia Havrlant ja Kreinovich 2017 mukaan. TF-IDF on lyhenne englanninkielisistä sanoista term frequency-inverse document frequency. Fan ja Qin 2018 mukaan TF-IDF:lle on useita esitysmuotoja. TF-IDF on Gebre ym. 2013 mukaan tekstin painotusjärjestelmä. He jatkavat, että termifrekvenssi ilmaisee, kuinka usein tietty termi esiintyy tekstissä. Gebre ym. 2013 mukaan termin esiintyminen tekstissä usein ei ole hyvä indikaattori ja usein esiintyville sanoille annetaan vähemmän painoarvoa, sillä kaikki usein esiintyvät termit eivät ole tärkeitä tekstin kannalta. Tämän IDF ilmaisee määrällisesti. Guan, Smetannikov ja Tianxing 2020 esittävät TF-IDF:n formaalisti seuraavasti:

$$\mathrm{TF}(t) = \frac{\text{termin } t \text{ lukumäärä dokumentissa}}{\text{termien kokonaislukumäärä dokumentissa}},$$

$$\text{IDF}(t) = \log \frac{\text{dokumenttien lukumäärä}}{\text{dokumenttien lukumäärä termillä } t} \text{ ja}$$

$$\text{TF-IDF} = \text{TF}(t) \cdot \text{IDF}(t).$$

Logaritmi voi olla esimerkiksi 2- tai *e*-kantainen. Oleellista logaritmissa on lukuarvojen liian suureksi kasvamisen estäminen määrittelyalueellaan aidosti kasvavalla ja jatkuvalla funktiolla, jolloin arvojen järjestys säilyy.

Aihemallinnuksen esikäsittelyn jälkeen alkaa koneoppimisvaihe, joka kullekin aihemallinnusmenetelmälle erilainen (Anantharaman ym. 2019). LDA on tutkijoiden Blei, Ng ja Jordan vuonna 2003 kehittämä menetelmä, jonka he kuvailevat artikkelissaan *Latent Dirichlet Allocation* (Blei, Ng ja Jordan 2003) generatiivisena probabilistisena mallina. Generatiivisessa probabilistisessa mallinnuksessa Blei 2012 mukaan dataa käsitellään siten, että se on generatiivisen prosessin tulos, mihin sisältyy myös piilomuuttujia. Hän jatkaa, että prosessi määrittelee yhteisen todennäköisyysjakauman havaituille muuttujille ja piilomuuttujille. Blei 2012 mukaan data-analyysiä suorittamalla käytetään tätä yhtenäistä jakaumaa laskemalla piilomuuttujien ehdollinen jakauma havaittujen muuttujien perusteella. Saatua jakaumaa kutsutaan myös posterioriseksi jakaumaksi (Blei 2012). Lisäksi termi aihemallinnus on Vayansky ja Kumar 2020 mukaan koneoppimisyhteisössä tutkijoiden Blei, Ng ja Jordan antama vuonna 2003.

Blei, Ng ja Jordan 2003 kuvailevat LDA:ssa *sanaa* indeksoiduksi yksiköksi diskreetistä datasta sanastosta. *Dokumenttia* he puolestaan kuvaavat N:n sanan joukoksi ja *korpusta* dokumenttien joukoksi. LDA:ssa dokumenttia pidetään Cao ym. 2009 mukaan aihejakaumana. Aihe puolestaan on sanajakauma (Cao ym. 2009). Aiheiden generointi tapahtuu Blei 2012 mukaan kahdessa vaiheessa, mitkä ovat esitetty seuraavassa.

1. Satunnaisesti valitaan jakauma aiheiden kesken.
2. Jokaiselle sanalle dokumentissa
   (a) Satunnaisesti valitaan aihe aiheiden jakaumasta.
   (b) Valitaan satunnaisesti sana vastaavasta jakaumasta sanaston yli.

Tämä tilastollinen malli heijastaa intuitiota, että asiakirjoissa on useita aiheita. Jokainen dokumentti esittelee aiheet eri suhteessa (vaihe 1); jokainen sana kussakin dokumentissa saadaan yhdestä aiheesta (vaihe 2 (b)), jossa valittu aihe valitaan dokumenttikohtaisesta jakaumasta aiheiden kesken (vaiheen 2 kohta (a)). (Blei 2012)

LSI (engl. Latent Semantic Indexing) kehitettiin 1970-luvulla tilastolliseksi tekniikaksi, jolla voidaan korreloida semanttisesti linkitettyjä termejä korpuksesta (Sarkar 2019). Cosma ja Joy 2012 mukaan LSI on LSA:n erikoistapaus, ja termiä LSI käytetään tiedon indeksointiin tai hakuun liittyviin tehtäviin, kun taas termiä LSA käytetään kaikenlaiseen muuhun kuten tekstinyhteenvetoon.

LSA on yksi parhaista malleista ohjaamattoman koneoppimisen paradigmassa, jossa yhdistyy tilastollisia ja algebrallisia menetelmiä (Amala, Richasdy ja Purbolaksono 2022). Karl, Wisnowski ja Rushing 2015 mukaan LSA:ssa on sanapussimatriisi tai Amala, Richasdy ja Purbolaksono 2022 mukaan TF-IDF-matriisi, joka jaetaan kolmeen uuteen matriisiin singulaariarvohajotelmalla (engl. singular value decomposition, SVD). Amalia ym. 2017 mukaan SVD:n pääideana on dimensiovähennys sekä kohinan tai ei-toivotun datan poistaminen sanasuhteiden näkemiseksi.

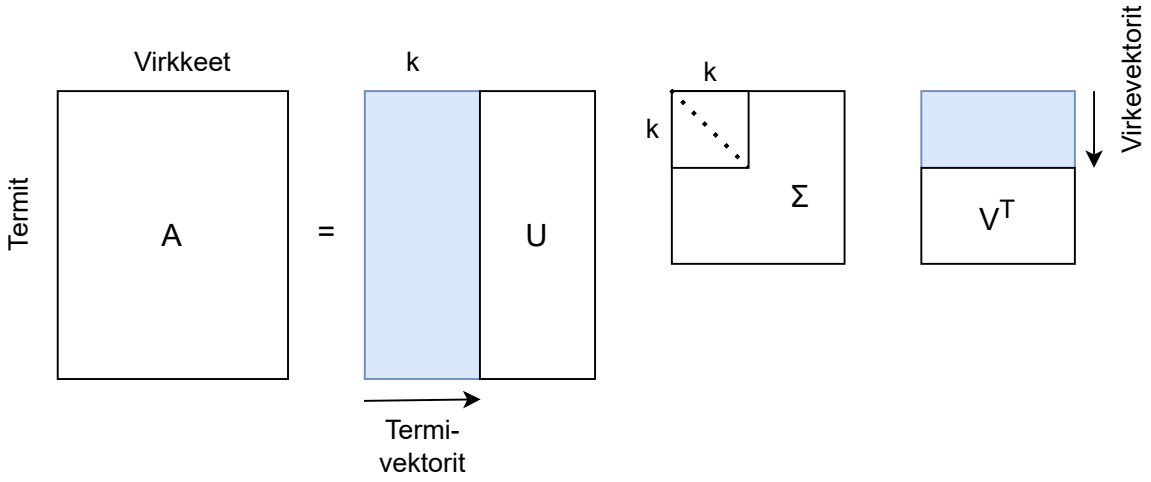
Steinberger, Jezek ym. 2004 esittävät LSA:han perustuvan yhteenvetomenetelmän, kuten seuraavassa esitetään. Olkoon annettuna $m \times n$ matriisi $\mathbf{A}$ jossa aina $n \geq m$. Matriisin $\mathbf{A}$ singulaarihajotelma SVD määritellään seuraavasti:

$$A = U\Sigma V^T,$$

missä $\mathbf{U} = [u_{ij}]$ on $m \times n$ sarakeortonormaali matriisi, jonka sarakkeita kutsutaan vasemmanpuoleisiksi singulaarivektoreiksi; $\Sigma = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_n)$ on $n \times m$ diagonaalimatriisi, jonka diagonaalielementit ovat ei-negatiivisia singulaariarvoja järjestettynä laskevaan järjestykseen ja $\mathbf{V} = [v_{ij}]$ on $n \times n$ ortonormaalimatriisi, jonka sarakkeita kutsutaan oikeanpuoleisiksi singulaarivektoreiksi. Jos $\mathbf{A}$:n aste on r, niin Σ:lle pätee:

$$\sigma_1 \geq \sigma_2 ... \geq \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0.$$

Semanttisesta näkökulmasta SVD johtaa piilevän semanttisen rakenteen dokumentista, jota



Kuvio 1. Singulaariarvohajotelma mukaillen (Steinberger, Jezek ym. 2004).

matriisi **A** edustaa. Operaatio kuvaa alkuperäisen dokumentin jakamista *r*:ään lineaarisesti riippumattomaan kantavektoriin tai käsitteeseen. Kukin dokumentin termi ja virke indeksoidaan yhdessä näiden kantavektoreiden (käsitteiden) avulla. SVD:n ainutlaatuinen ominaisuus on se, että se pystyy kuvaamaan ja mallintamaan termien välisiä suhteita, jolloin se voi klusteroida termejä ja virkkeitä semanttisesti. Jos sanayhdistelmäkuvio on merkittävä ja toistuva dokumentissa, otetaan kuvio talteen ja esitetään yhdellä singulaarivektorilla. Vastaavan singulaarisen arvon suuruus osoittaa tämän kuvion tärkeysasteen dokumentissa. Kaikki tätä sanayhdistelmäkuviota sisältävät lauseet projisoidaan tätä singulaarivektoria pitkin, ja tätä kuviota parhaiten edustavalla virkkeellä on suurin indeksiarvo tällä vektorilla. Koska kukin tietty sanayhdistelmäkuvio kuvaa tiettyä aihetta (käsitettä) dokumentissa, edellä kuvatut seikat johtavat hypoteesiin, että kukin singulaarinen vektori edustaa dokumentin keskeistä aihetta (käsitettä) ja että sitä vastaavan singulaarisen arvon suuruus edustaa keskeisen aiheen (käsitteen) tärkeysastetta. (Steinberger, Jezek ym. 2004)

Matriisi $\mathbf{V}^T$ kuvaa kunkin aiheen tärkeysastetta kussakin virkkeessä. Yhteenvetoprosessissa valitaan informatiivisin virke kustakin aiheesta. Se tarkoittaa, että *k*.:nen virkkeen indeksiarvo on suurin *k*.:nessa oikeanpuoleisessa singulaarivektorissa matriisissa $\mathbf{V}^T$. (Steinberger, Jezek ym. 2004)

2.2.2 Aiheiden määrän määrittäminen

Paras hyöty aihemalleista tulee esiin, kun mallin sanaston määrä on suuri (Arun ym. 2010). Algoritmin valinnasta riippumatta Greene, O'Callaghan ja Cunningham 2014 mukaan keskeistä aihemallinnuksen onnistuneessa soveltamisessa on sopiva aiheiden määrä k tarkasteltavaan korpukseen. He jatkavat, että liian pienen k:n arvon valitseminen luo liian laajoja aiheita ja liian suuren arvon valitsemisesta seuraa aiheiden "yliklusteroituminen".

Newman ym. 2010 tutkimuksen pyrkimyksenä on kehittää tekniikoita laskennallisen lingvistiikan ja koneoppimisen aihearvioinnin laadulliseen arviointiin. He ehdottavat malleja aiheiden koherenssin ennustamiseksi. Heidän mukaan Wikipediaa resurssina käyttävät menetelmät ovat pystyneet saavuttamaan aiheiden koherenssin arvioinnissa erinomaisen yhtäpitävyyden ihmisten kanssa.

Aihekoherenssianalyysi mittaa sanojen semanttisen samankaltaisuuden astetta. Korkea todennäköisyysarvo sekä monimutkaisuus (engl. perplexity) kussakin aiheessa antaa arvion aiheklusteroinnin tehokkuudesta. Näitä mittauksia käytetään usein määrittämään aihemallinnuksen aiheiden lukumäärää. (Principe ym. 2022)

Principe ym. 2022 mukaan Mimno ym. 2011 systemaattisesti ja empiirisesti tutkivat aihemallinnuksen koherenssimittauksia ja heidän tuloksensa johtivat koherenssimetriikkaan *UMass*. Se on epäsymmetrinen vahvistusmitta tärkeimpien sanaparien ympärillä (Principe ym. 2022). Pasquali 2016 mukaan *UMass*-metriikka laskee korrelaation dokumentin sanoihin perustuen ehdolliseen todennäköisyyteen.

Mimno ym. 2011 mukaan heidän kehittämä koherenssimetriikka perustuu ainoastaan sanojen samanaikaisen esiintymisen tilastoihin, jotka on kerätty mallinnettavasta korpuksesta, eikä metriikka riipu ulkoisesta vertailukorpuksesta. He jatkavat, että ihannetapauksessa kaikki tällaiset tilastot olisivat jo otettu huomioon aihemallissa. He uskovat, että eräs työnsä tärkeimmistä saavutuksista on sen osoittaminen, että tavanomaiset aihemallit eivät täysin hyödynnä saatavilla olevaa samanaikaisen esiintymisen tilostoa. Siksi heidän aihemallien arvioinnissa ei tarvita ulkopuolisia vertailukorpuksia.

Cao ym. 2009 ovat validoineet parhaan aiheiden määrän LDA:ssa korreloivan aiheiden väli-

ten etäisyyksien kanssa. He ovat tutkimuksessaan demonstroineet, että LDA suoriutuu parhaiten, kun keskimääräinen kosinietäisyys aiheiden välillä saavuttaa minimiarvon. Metodi perustuu koko korpuksen tilastoihin.

2.2.3 Uusimmat aihemallinnuksen innovaatiot

Eräs tavanomaisten aihemallinnusmenetelmien rajoituksista on se, sanapussimallien avulla ne eivät ota huomioon sanojen välisiä semanttisia suhteita. Koska esitykset eivät huomioi sanojen virkkeiden kontekstia, sanapussisyöte ei välttämättä pysty edustamaan dokumentteja tarkasti. (Grootendorst 2022)

Viimeaikaiset innovaatiot, joissa käytetään transformer-malleja, ratkaisevat myös sanapussioletuksien rajoituksia. Se saavutetaan erityisesti aihemalleissa, joissa yhdistetään BERT-malleja (engl. Bidirectional Encoder Representations from Transformers). (Chen ym. 2023)

BERT:in arkkitehtuuri perustuu Devlin ym. 2018 mukaan olennaisesti Vaswani ym. 2017 esittämään transformer-arkkitehtuuriin, joka on kuvailtu luvussa 4.2. Devlin ym. 2018 mukaan BERT:in arkkitehtuuri onkin lähes identtinen Vaswani ym. 2017 transformer-arkkitehtuurin kanssa ja jättävät BERT:in arkkitehtuurin tarkemman kuvauksen Vaswani ym. 2017 transformer-arkkitehtuurin kuvauksen varaan.

BERT:in kehyksessä on kaksi vaihetta: esiopetus ja hienosäätö. Esiopetuksen aikana malli opetetaan nimeämättömällä (engl. unlabeled) datalla käyttäen erilaisia esiopetustehtäviä. Hienosäätöä varten BERT-malli alustetaan ensin esiopetetuilla parametreilla, ja kaikkia parametreja hienosäädetään käyttämällä myöhempien tehtävien nimettyä dataa. Kullakin jatkotehtävällä on erilliset hienosäädetyt mallit, vaikka ne alustetaan samoilla esiopetetuilla parametreilla. (Devlin ym. 2018)

Grootendorst 2022 esittämä BERTopic tuottaa aiheiden esityksiä kolmessa vaihessa. Ensin kukin dokumentti muunnetaan upotusrepresentaatioksi (engl. embedding representation) käyttäen esiopetettua kielimallia. Grootendorst 2022 käyttää dokumenttien upotuvaiheessa SBERT-kehystä (engl. Sentence-BERT). Hänen mukaan sen ovat kehittäneet Reimers ja Gurevych 2019. Heidän mukaan SBERT hienosäätää BERT:iä triplaverkkoarkkitehtuurissa. Siinä johdetaan semanttisesti mielekkäitä virkkeiden upotuksia, joita voidaan verrata kosinisa-

mankaltaisuuden avulla (Reimers ja Gurevych 2019).

Ennen näiden upotusten klusterointia tuloksena syntyvien upotusten dimensiota pienennetään klusterointiprosessin optimoimiseksi. Lopuksi dokumenttien klustereista poimitaan aiheiden esitykset käyttämällä TF-IDF:n mukautettua luokkapohjaista varianttia. (Grootendorst 2022)

Aiheiden esitykset mallinnetaan kussakin klusterissa olevien dokumenttien perusteella, ja kullekin klusterille annetaan yksi aihe (Grootendorst 2022). BERT:in klusterit ovatkin semanttisesti merkityksellisiä ja uniikkeja (Thompson ja Mimno 2020).

Kunkin aiheen osalta halutaan tietää, mikä erottaa jonkin aiheen sen klusterin sanajakauman perusteella toisesta. Tätä tarkoitusta varten muokataan TF-IDF:ää, joka on mittari sanan tärkeyden esittämiseksi dokumentille siten, että se mahdollistaa termin esittämisen tärkeyden aiheelle. (Grootendorst 2022)

BERT käsittää myös dynaamista aihemallinnusta. Se tarkoittaa, että aiheiden tietyn aikakauden luonteen ei pitäisi vaikuttaa globaalien aiheiden luomiseen. Sama aihe saattaa esiintyä eri aikakausina mahdollisesti eri tavoin esitettynä. Sanat ”auto” ja ”ajoneuvo” voivat vuonna 2020 esiintyä myös esimerkiksi muodossa ”Tesla”. (Grootendorst 2022)

Jos sama aihe on määritetty autoihin liittyville dokumenteille vuosina 1990 ja 2020, sen sisältö saattaa vaihdella. Siksi luodaan ensin globaali esitys aiheista niiden ajallisesta luonteesta riippumatta, ennen kuin kehitetään aiheiden paikallinen esitys. Tätä varten BERTopic sovittaa ensin koko korpukseen ikään kuin aiheissa ei olisi aikakausiin liittyviä näkökohtia, jotta korpuksesta luodaan globaali kokonaiskuva aiheista. (Grootendorst 2022)

2.3 Ekstraktiivinen yhteenveto

Ekstraktiiviset yhteenvetomenetelmät tekevät yhteenvedon lähdetekstistä muuttamatta lähdetekstiä mitenkään (Agrawal 2020). Kågebäck ym. 2014 mukaan ekstraktiivisissa yhteenvedoissa on keskeistä käsitys tekstissä olevien virkkeiden samankaltaisuudesta. He jatkavat, että virkkeitä ekstraktoivat yhteenvetojärjestelmät luovat tiivistelmiä syötteestä valittujen edustavien virkkeiden avulla. Kågebäck ym. 2014 mukaan tyypillisesti ekstraktiiviset

yhteenvetotekniikat voidaan jakaa kahteen osaan, yhteenvetokehykseen ja virkkeiden vertaamiseen käytettyihin samankaltaisuusmittauksiin.

Luhn loi ensimmäisen automaattisen yhteenvetojärjestelmän vuonna 1958 (Suanmali, Salim ja Binwahlan 2009). Cajueiro ym. 2023 mukaan Luhnin menetelmässä sanan tärkeys riippuu seuraavista säännöistä: 1) Se ei huomioi pronomineja, prepositioita ja artikkeleita; 2) se ei huomioi vähiten yleisiä sanoja; 3) sanat, joiden frekvenssi on suurempi kuin vähiten yleisimmillä sanoilla ovat merkityksellisiä. He jatkavat, että poimittavien virkkeiden valitsemiseksi tarkastellaan myös virkkeen merkitsevien sanojen välisen etäisyyden heuristista arviointia, mitä käytetään merkitsevien sanojen ryhmien määrittämiseen. Näillä määritelmillä tärkeät virkkeet ovat niitä, joissa on klusteroituneena suuri määrä tärkeitä sanoja (Cajueiro ym. 2023).

TextRank-yhteenvetomenetelmä puolestaan on Mihalcea ja Tarau 2004 mukaan graafipohjainen ja heidän mukaan se on johdettu Googlen PageRank-algoritmista. Google käyttää PageRank-algoritmia verkkosivustojen tärkeysjärjestykseen asettamiseen (Sarkar 2019).

Mihalcea ja Tarau 2004 ovat kuvailleet TextRankia analogian kautta PageRankiin ”satunnaisen surffausmallin” kautta, missä käyttäjä surffaa verkossa seuraamalla minkä hyvänsä verkkosivun linkkejä. Tekstinmallintamisen yhteydessä puolestaan TextRank toteuttaa Mihalcea ja Tarau 2004 mukaan ”tekstinsurffaamista” tekstin koheesion kontekstissa (Halliday ja Hasan 2014): tietystä käsitteestä K tekstissä ”seurataan” todennäköisyyteen perustuen linkkejä käsitteisiin, joilla on suhde nykyiseen käsitteeseen K, oli suhde leksikaalinen tai semanttinen. Koheesio liittyy Mihalcea ja Tarau 2004 mukaan myös ”neulontailmiöön” (Hobbs 1974): Sanoihin liittyvät tosiasiat jaetaan diskurssin eri osissa ja tällaiset suhteet ”neulovat keskustelun yhteen”.

TextRank huomioi koko tekstistä rekursiivisesti muodostetun informaation (graafit). Se tunnistaa teksteihin rakentamiensa graafien avulla yhteyksiä tekstin eri entiteettien välillä ja suosittelee muita asiaan liittyviä tekstiyksiköitä. Suosituksen vahvuus lasketaan rekursiivisesti suosituksen antavien yksiköiden tärkeyden perusteella. Virkkeet, joita tekstin muut virkkeet suosittelevat, ovat todennäköisesti informatiivisempia annetulle tekstille, ja siksi niille annetaan korkeampi pisteytys. Parhaiten pisteytetyt virkkeet valitaan yhteenvetoon graafille

ajetun pisteytysalgoritmin tuloksista. (Mihalcea ja Tarau 2004)

2.4 Abstrahoitu yhteenveto

Abstrahoitu (engl. abstractive) yhteenveto ei ole vain joistakin valituista virkkeistä koostuva yhteenveto, vaan se on tiivistetty parafraasi asiakirjan pääsisällöstä, jossa mahdollisesti käytetään lähdetekstissä tuntematonta sanastoa (Nallapati ym. 2016). Gupta ja Gupta 2019 jakavat abstrahoivat yhteenvetometodit kolmeen pääluokkaan: rakennepohjaisiin, semanttispohjaisiin ja syväoppimispohjaisiin metodeihin. Heidän mukaan syväoppiminen on osa koneoppimiseen perustuvia menetelmiä. He jatkavat, että syväoppimiseen perustuva yhteenveto käsittää useita epälineaarisia käsittelykerroksia, joiden avulla tekstistä voidaan poimia piirteitä. Oppiminen voi olla sekä ohjattua että ohjaamatonta ja se perustuu neuroverkkoihin.

Guan, Smetannikov ja Tianxing 2020 mukaan Vaswani ym. 2017 ehdottivat vuonna 2017 transformeria korvaamaan aiemmat syväoppimismenetelmät attention-mekanismilla. Wolf ym. 2020 mukaan transformerista on tullut nopeasti hallitseva arkkitehtuuri luonnollisen kielen prosessoinnissa. Heidän mukaan arkkitehtuuri skaalautuu opetusdatan ja mallin koon mukaan, mahdollistaa tehokkaan rinnakkaisopetuksen ja kaappaa pitkän kantaman sekvenssiominaisuudet. He lisäävät, että mallien esiopetus mahdollistaa niiden opetuksen yleisillä korpuksilla johtaen merkittäviin tarkkuuden parannuksiin erilaisissa tehtävissä kuten Lewis ym. 2019 mukaan tekstistä yhteenvedon tekemisessä.

Transformer-malli voi rakentaa käsityksen tekstistä kokoamalla dataa attention-mekanismilla ja luo yhteenvedon. Sisällönvalitsija määrittää ennalta lähdetekstin virkkeet, jotka ovat osa yhteenvetoa. Sitten se rajoittaa hermoverkkomallia valitulla sisällöllä ja tuottaa siitä abstrahoidun yhteenvedon. (Syed, Gaol ja Matsuo 2021)

2.5 Virkkeiden tiivistäjät

Song ym. 2022 mukaan käyttäjien monimutkaiset ilmaisut chatboteille, joista ei havaita käyttäjän tarkoitusta (engl. intent), siirretään manuaaliseen asiakaspalveluun, mikä heikentää chatbotin käyttäjäkokemusta tai lisää manuaalista työmäärää. He ehdottavat kaksivaiheista

käyttäjän tarkoituksen havaitsemismallia chatboteille, joka käsittää virkkeen tiivistämisen ja käyttäjän tarkoituksen luokittelun.

Gholipour Ghalandari, Hokamp ja Ifrim 2022 mukaan virkkeiden tiivistämisen idea on tekstin pituuden lyhentäminen poistamalla epäolennaista sisältöä säilyttäen samalla tärkeät tosiasiat ja kieliopillisuus. He ovat toteuttaneet vahvistumisoppimiseen (engl. reinforcement learning) perustuvan ekstraktoivan virkkeidentiivistäjän[1]. Sen hyödyiksi he mainitsevat seuraavassa mainitut.

- **Ohjaamattoman oppimisen:** nimettyjä (engl. labeled) esimerkkejä ei tarvita.
- **Nopean päättelyn:** testihetkellä malli suorittaa vain yksivaiheisen sekvenssin nimeämisen (engl. sequence labeling).
- **Konfiguroitava:** palkitsemismekanismi voidaan räätälöidä täsmällisiin käyttötapauksiin.

Vahvistumisoppimisessa on tavoitteena ohjata järjestelmää siten, että jokin suorituskykykriteeri maksimoidaan (Wiering ja Van Otterlo 2012). Gholipour Ghalandari, Hokamp ja Ifrim 2022 mukaan vahvistumisoppiminen on tullut suosituksi tekstin tiivistämisen kentällä ekstraktoivien ja abstrahoivien yhteenvetojen osatehtäviin. Gupta ja Gupta 2019 mukaan käytetyn oppimistyypin perusteella virkkeiden tiivistäjät jaetaan ohjattuihin ja ohjaamattomiin malleihin. Ohjatut lähestymistavat sisältävät opetuksen tiivistettävien virkkeiden löytämiseksi käyttämällä todennäköisyyksiä. Niiden löytämiseksi menetelmät käyttävät generatiivista tai diskriminoituvista (engl. discriminative) mallia. Generatiivisissa malleissa kohteen tiivistämisen todennäköisyys löydetään suoraan tai epäsuorasti käyttämällä kohinaista kanavaa. Diskriminaatioisissa malleissa päätavoitteena on vähentää opetusvirheitä. Gupta ja Gupta 2019 mukaan virkkeiden tiivistäminen on NLP-yhteisöissä erittäin tärkeä aihe, jolla on hyvin tärkeä rooli abstrahoivien yhteenvetojen luomisessa.

1. https://paperswithcode.com/paper/efficient-unsupervised-sentence-compression-1. Viitattu 25.12.2022.

3 Yhteenvetojen laadunarviointi

Luvussa tarkastellaan kahta suosituimpaa *n*-grammeihin perustuvaa automaattisen yhteenvedon laadunarviointimetriikkaa. Lisäksi tarkastellaan semanttisen samankaltaisuuden ja kokonaisuuden kontekstin huomioivaa transformer-pohjaista BERTScorea sekä lähdetekstin faktojen sisältymistä yhteenvetoon arvioivaa myös transformeria hyödyntävää CoCo-metriikkaa. Scialom ym. 2021 mukaan tekstien yhteenvedot ovat eräs vaikeimmin automaattisesti arvioitava kohde.

3.1 ROUGE

ROUGE on *de facto* metriikka Khurana ja Bhatnagar 2022 mukaan yhteenvetojen laadunmittaamiseen. Lin 2004 mukaan ROUGE on lyhenne englanninkielisistä sanoista *Recall-Oriented Understudy for Gisting Evaluation*. Hän jatkaa, että ROUGE käsittää automaattisia mittaustapoja yhteenvetojen laadun määrittämiseen vertaamalla yhteenvetoja ideaaleihin ihmisten luomiin yhteenvetoihin.

ROUGE-1 ja ROUGE-2 mittaavat samanaikaisten *n*-grammien esiintymisen tilastoja. ROUGE-1 mittaa unigrammien päällekkäisyyksien määrää malliyhteenvetojen ja automaattisten yhteenvetojen välillä ja vastaavasti ROUGE-2 mittaa bigrammien päällekkäisyyden määrää. ROUGE-L mittaa mukaan pisimmän yhteisen jakson määrää yhteenvedoissa. (Lin 2004)

Hingu, Shah ja Udmale 2015 ovat arvioineet Wikipedia-artikkeleita automaattisilla yhteenvedoilla ROUGE F1-pisteillä. Heidän mukaan F1-pisteet mittaavat tietyn *systeemin tarkkuutta*. F1-pisteiden laskemiseen käytetään yhteenvetoihin liittyvää osumien määrää (engl. recall) ja tarkkuutta (engl. precision). Tarkkuus edustaa yhteenvedon oikeellista osuutta malliyhteenvetoon. Osumien määrä on suhteessa oikeaan malliyhteenvetoon ja mittaa järjestelmän herkkyyttä.

Agrawal 2020 esittävät formaalisti osumien määrän, tarkkuuden ja F1-mittauksen seuraavasti:

$$\text{osumien määrä} = \frac{\text{päällekkäisten sanojen lukumäärä}}{\text{malliyhteenvedon kokonaissanamäärä}},$$
$$\text{tarkkuus} = \frac{\text{päällekkäisten sanojen lukumäärä}}{\text{tuotetun yhteenvedon kokonaissanamäärä}} \text{ ja}$$
$$\text{F1} = \frac{2 * \text{tarkkuus} * \text{osumien määrä}}{\text{tarkkuus} + \text{osumien määrä}}.$$

F1-pisteet ovat harmoninen keskiarvo osumien määrästä ja tarkkuudesta (Agrawal 2020).

3.2 METEOR

Banerjee ja Lavie 2005 ovat ehdottaneet METEOR-metriikkaa alunperin konekäännösten arviointia varten. M. Zhang ym. 2022 mukaan METEOR on suosittu myös yhteenvetojen arvioinnissa käytettävänä metriikkana.

METEOR-pisteet lasketaan Banerjee ja Lavie 2005 mukaan seuraavasti:

$$\text{Score} = \text{FMean} * (1 - \text{Penalty}).$$

FMean viittaa harmoniseen keskiarvoon, joka lasketaan käyttämällä unigrammitarkkuutta (P) laskemalla järjestelmäkäännöksessä olevien unigrammien määrä suhteena järjestelmäkäännöksessä olevien unigrammien kokonaismäärään. Vastaavasti unigrammien osumien määrä (R) lasketaan järjestelmäkäännöksessä olevien yhdistettyjen unigrammien lukumäärän ja mallikäännöksen unigrammien lukumäärän ja mallikäännöksessä olevien unigrammien kokonaismäärän suhteena. Tästä saadaan FMean yhdistämällä tarkkuus ja osumien määrä harmonisen keskiarvon avulla (van Rijsbergen, 1979), joka painottaa eniten osumien määrää. Formaalisti FMean saa muodon

$$\text{FMean} = \frac{10\text{PR}}{\text{R} + 9\text{P}},$$

missä P tarkoittaan tarkkuutta ja R saantia. (Banerjee ja Lavie 2005)

Penalty (suom. ”sakko”) puolestaan saadaan seuraavasti:

$$\text{Penalty} = 0.5 * \frac{\#chuncks}{\#unigrams\_matched}.$$

Tässä kaikki järjestelmäkäännöksen unigrammit, jotka on kartoitettu unigrammeiksi mallikäännöksessä, ryhmitellään mahdollisimman pieneen määrään paloja (engl. chunks) siten, että kunkin osan unigrammit ovat vierekkäisissä paikoissa järjestelmän käännöksessä ja ne kartoitetaan myös unigrammeihin, jotka ovat vierekkäisissä paikoissa mallikäännöksessä. Näin ollen mitä pidemmät n-grammit, sitä vähemmän paloja, ja ääritapauksessa, jossa koko järjestelmän käännösmerkkijono vastaa mallikäännöstä, paloja on vain yksi. Toisessa ääripäässä, jos bigram- tai pidempiä osumia ei ole, paloja on yhtä monta kuin unigramosumia. Sakko lasketaan tällöin edellä esitetyllä kaavalla. (Banerjee ja Lavie 2005)

3.3 BERTScore

BERTScore on saanut inspiraationsa Xie ym. 2021 mukaan esiopetettujen kontekstuaalisten sanojen upotusten menestyksestä ja se hyödyntää esiopetettua BERT-mallia laskemaan luodun yhteenvedon ja referenssin välistä samankaltaisuutta. BERT-mallia on esitelty luvussa 2.2.3. T. Zhang ym. 2019 mukaan BERT voi tuottaa samalle sanalle eri lauseissa erilaisia vektoriesityksiä riippuen ympäröivistä sanoista, jotka muodostavat kohdesanan kontekstin. Näin heidän mukaan saadaan joustava samankaltaisuuden mittaus täsmällisen merkkijonon tai heuristisen yhteensovittamisen sijasta.

Mallisanan x_i ja kandidaattisanan $\hat{x}_j$ kosinisamankaltaisuus on

$$\frac{x_i^{\mathrm{T}} \hat{x}_j}{||x_i|| ||\hat{x}_j||}.$$

T. Zhang ym. 2019 käyttävät esinormalisoituja vektoreita, mikä minimoi tämän laskutoimituksen sisätuloksi $x_i^{\mathrm{T}} \hat{x}_j$. Vaikka heidän metriikassaan tarkastellaan sanoja erillisinä, heidän mukaan kontekstisidonnaiset upotukset sisältävät tietoa muusta lauseesta.

T. Zhang ym. 2019 mukaan pistemäärä (BERTScore) vastaa jokaista sanaa x:stä $\hat{x}$:ään osumien määrän laskemiseksi ja jokaista sanaa $\hat{x}$:stä x:ään tarkkuuden laskemiseksi. He käyttävät ahnetta vastaavuutta maksimoidakseen vastaavuuden samankaltaisuuspistemäärän, jossa kukin merkki sovitetaan toisen lauseen samankaltaisimpaan sanaan. Sitten he yhdistävät tarkkuuden ja osumien määrän F1-pisteiden laskemiseksi. Mallivirkkeen x ja kandidaatti-

virkkeen $\hat{x}$ osumien määrän, tarkkuus ja F1-pisteet ovat järjestyksessä seuraavat:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^{\mathrm{T}} \hat{x}_j,$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^{\mathrm{T}} \hat{x}_j \text{ ja}$$

$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.$$

T. Zhang ym. 2019 mukaan BERTScore käyttää TF-IDF:ää harvinaisten tärkeiden sanojen painotukseen yleisten sanojen joukossa. Heidän käyttämä TF-IDF-menetelmä on kuvailtu seuraavassa.

Olkoon annettu M referenssilausetta $\{x^{(i)}\}_{i=1}^{M}$. Tällöin sanapalan w IDF-pisteet ovat

$$\text{IDF}(w) = -\log \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}[w \in x^{(i)}],$$

jossa $\mathbb{I}[\cdot]$ on indikaattorifunktio painotukseen. T. Zhang ym. 2019 eivät käytä koko TF-IDF-mittausta, koska BERTScore prosessoi yksittäisiä virkkeitä, joissa termifrekvenssi (TF) on todennäköisesti 1.

3.4 CoCo

Hallitsevat automaattiset metriikat yhteenvetojen arviointiin, esimerkiksi ROUGE ja METEOR, perustuvat pääasiassa n-grammien leksikaaliseen päällekkäisyyteen ja niiden on todettu korreloivan huonosti ihmisten tekemien arvioiden kanssa asiasisällön johdonmukaisuuden osalta (Wang, Cho ja Lewis 2020). BERTScoren kaltaiset metriikatkaan eivät johda tyydyttävään korrelaatioon tosiseikkojen johdonmukaisuutta koskevien inhimillisten arvioiden kanssa, koska ne kuvaavat vain sanatasolla päällekkäisyyttä tai samankaltaisuutta (Xie ym. 2021). Sanatasolla määriteltyjen metriikoiden sijaan Goodrich ym. 2019 ehdottavat, että asiasisällön johdonmukaisuutta mitataan laskemalla luodusta tiivistelmästä ja lähdedokumentista poimittujen faktojen päällekkäisyys. Xie ym. 2021 ovat ehdottaneet metriikkaa, jolla mitataan tosiasioiden johdonmukaisuutta tekstin tiivistämisessä. Heidän metriikkansa on nimeltään **Co**unterfactual **Co**nsistency, lyhyesti CoCo.

Chen ja Zhang 2023 mukaan CoCo suorittaa arvioinnin tärkeiden sanojen sijaintipisteiden perusteella. He jatkavat, että sijaintipisteet antavat logaritmisen todennäköisyyden sille, että kukin sana tiivistelmässä esiintyy tietyssä paikassa, kun otetaan huomioon alkuperäinen teksti X. CoCo:n perusajatuksena on ensin havaita avainsanat tiivistelmässä, peittää ne alkuperäisessä tekstissä, jotta saadaan luotua peitetty asiakirja M, ja sen jälkeen laskea M:n ja alkuperäisen tekstin X avulla tärkeiden sanojen sijaintipisteiden erotus luoduissa yhteenvedoissa (Chen ja Zhang 2023). Intuitio on, että kun tekstit luodaan enemmän lähdeasiakirjan kuin kielellisen lähtökohdan perusteella, niin niiden pitäisi olla todennäköisemmin faktojen osalta johdonmukaisia lähdeasiakirjojen kanssa (Xie ym. 2021).

Xie ym. 2021 mukaan CoCo-metriikassa on neljälle tasolle peittämismekanismi, sanaesiintymistasolle (engl. token), virketasolle (engl. sentence), tekstialuevälille (engl. span) ja koko dokumentin tasolle (engl. document). Xie ym. 2021 tulkiten peittämismekanismi antaa kontekstin avainsanoille arvioitavasta yhteenvedosta valitun peittämismekanismin laajuudella alkuperäisestä dokumentista. Liian pieni peittoalue voi aiheuttaa sen, että dekooderi pystyy yhä päättelemään peitetyt sanat kontekstista, kun taas liian suuri peittoalue saattaa heikentää aiemman kielen vaikutusta ja johtaa siihen, että kaikki arvioitavan yhteenvedon sanat saavat lähes nollapisteet (Xie ym. 2021).

Xie ym. 2021 esittävät seuraavassa esitetyn esimerkin tekstialuevälille.

Lähdedokumentti X: ”People with a DNA variation in a **gene** called PDSS2 tend to drink fewer cups of coffee, a study carried out at the University of Edinburgh has found. It suggests the **gene** reduces cell ability to break down caffeine...”

Yhteenveto Y: ”Researcers have indentified a **gene** that appears to curb coffee consumption.”

Keltaisella korostettu on relevanttia sisältöä arvioitavan yhteenvedon käsitteelle ”gene”.

4 Transformerit

Luvussa esitellään transformerin perusidea ja tehdään katsaus sen yleiseen arkkitehtuuriin. Lisäksi esitellään transformer-arkkitehtuureista BART-arkkitehtuuri ja tehdään tarkempi katsaus `philschmid`-transformerin toimintaan.

4.1 Transformerin perusidea

Ghojogh ja Ghodsi 2020 kuvailevat Vaswani ym. 2017 luomaa transformeria automaattisena kooderina, joka ottaa syötteen, upottaa datan kontekstivektoriin ja generoi ulostulon. Syöte ja ulostulo ovat suhteessa toisiinsa siten, että automaattinen kooderi muuntaa syötteen suhteessa ulostuloon.

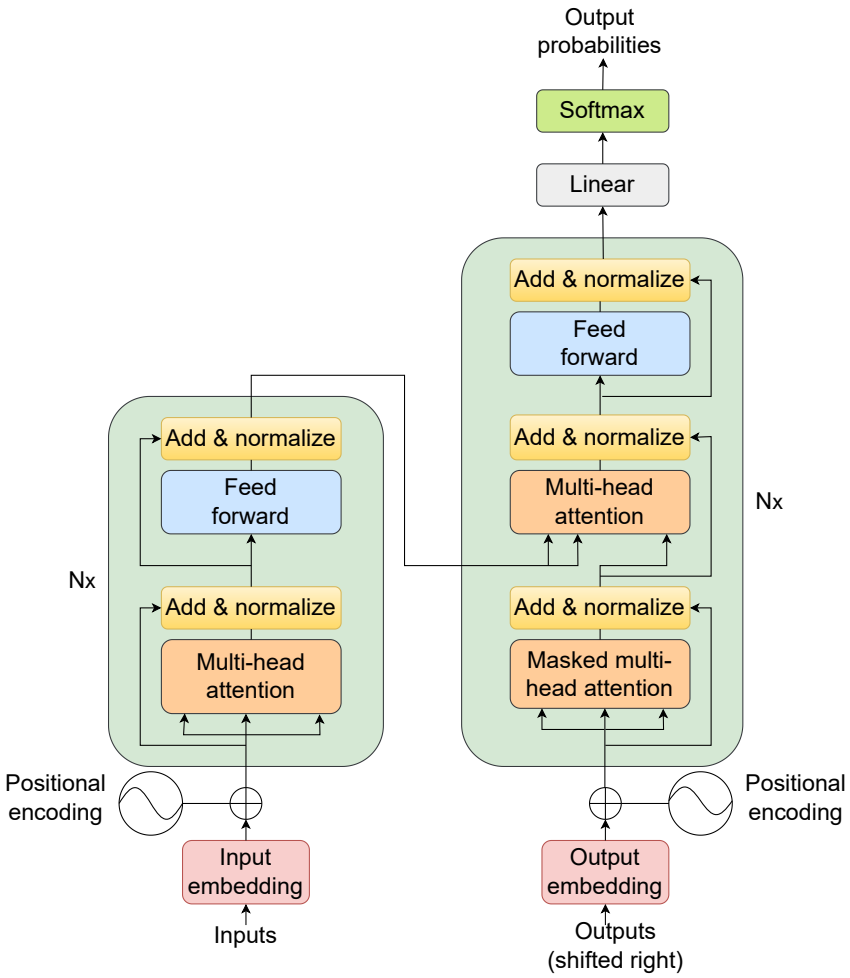
Vaswani ym. 2017 mukaan heidän ehdottaman transformerin arkkitehtuuri välttää toistamista ja nojaa sen sijaan täysin attention-mekanismiin luodakseen globaaleja riippuvuuksia syötteen ja ulostulon välille. He jatkavat, että transformer mahdollistaa huomattavasti enemmän rinnakkaisuutta laskennassa kuin aiemmat syväoppimismallit. Engel, Belagiannis ja Dietmayer 2021 mukaan attention-mekanismin idea on asettaa tärkeysperusteinen fokus syötesekvenssin eri osiin. Tästä seuraa, että syötteiden väliset suhteet korostuvat, joita voidaan käyttää kontekstin ja korkeamman tason riippuvuuksien poimimiseen (Engel, Belagiannis ja Dietmayer 2021).

Vaswani ym. 2017 mukaan self-attention-mekanismi yhdistääkin yksittäisen sekvenssin eri kohdat sekvenssin esityksen laskemiseksi. Heidän tietämyksensä mukaan transformer on ensimmäinen transduktiomalli, joka nojaa täysin self-attention-mekanismiin ja laskee syötteensä ja ulostulonsa esitykset ilman konvoluutiota tai sekvenssikohdistettuja takaisinkytkeytyviä neuroverkkoja, joissa syötteiden informaatio kulkee aika-askeleittain eteenpäin.

4.2 Transformerin arkkitehtuuri

Transformer on automaattinen kooderi, joka käsittää kooderin ja dekooderin (Ghojogh ja Ghodsi 2020). Vaswani ym. 2017 esittämän transformerin kooderi-dekooderi-rakenne on esi-

tetty kuviossa 2. Kukin kooderilohko koostuu pääasiassa monipäisestä self-attention-moduulista ja positionaalisesti toimivasta eteenpäin syöttävästä neuroverkosta (Lin ym. 2022).



Kuvio 2. Transformerin arkkitehtuuri mukaillen (Vaswani ym. 2017). Vasemmalla on kooderi ja oikealla on dekooderi. Kooderin puolelta nuolet dekooderin puolelle edustavat risti-attentiota.

Singh ja Mahmood 2021 mukaan eri sekvensseissä olevilla samankaltaisilla sanoilla voi olla erilaisia tulkintoja, jotka ratkaistaan *positionaalisen kooderin* avulla, joka tuottaa kontekstiin perustuvaa sanan sijaintia koskevaa tietoa. Tämän jälkeen Singh ja Mahmood 2021 mukaan parannettu kontekstuaalinen esitys syötetään attention-kerrokselle, joka jatkaa kontekstualisoinnin kehittämistä tuottamalla attention-vektoreita, jotka määrittävät sekvenssin *i*.:nen sanan merkityksen muiden sanojen kannalta. He jatkavat, että nämä attention-vektorit syötetään sitten eteenpäin syöttävään neuroverkkoon, jossa ne muunnetaan helpommin lä-

hestyttävään muotoon seuraavaa kooderi- tai dekooderilohkoa varten. Kuviossa 2 lohko on vaaleanvihreä alue, joita on N kappaletta kerroksittain.

Monipäinen attention lisää mallin kykyä korostaa sekvenssin eri sanojen sijainteja toteuttamalla attention rinnakkain useita kertoja. Tuloksena syntyvät yksittäiset attention-ulostulot tai päät yhdistetään ja muunnetaan lineaarisen kerroksen kautta odotettuihin ulottuvuuksiin. Kukin useista päistä mahdollistaa sekvenssin osien huomioimisen eri näkökulmasta, jolloin jokaiselle sanalle saadaan samanlaiset esitysmuodot. Tuloksena olevat monipäiset attention-vektorit (engl. multi-headed attention vectors) lasketaan rinnakkain, ja ne syötetään eteenpäin syöttävään kerrokseen. (Singh ja Mahmood 2021)

Kukin self-attention-moduulin koodauslohko onkin Lin ym. 2022 mukaan monipäinen sijainnin mukaan etenevä syöttöverkko (engl. Feed-Forward Network, FFN). Se toimii erikseen ja identtisesti jokaisessa positiossa (Lin ym. 2022).

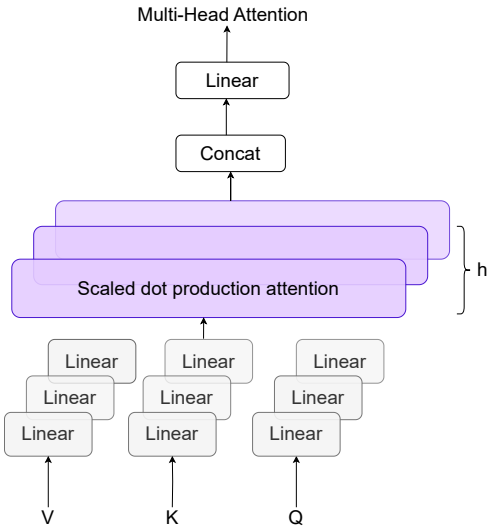
Lin ym. 2022 mukaan Vaswani ym. 2017 transformerin käsittämä attention-mekanismi käsittää kysely-avain-arvo (engl. Query-Key-Value, QKV) -mallin. Kun on annettuna matriisirepresentaatio kyselyistä $Q \in \mathbb{R}^{NxD_k}$, avaimet $K \in \mathbb{R}^{MxD_k}$ ja arvot $V \in \mathbb{R}^{MxD_v}$, niin skaalattu attentionin käyttämä sisätulo

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{D_k}})V = AV, \tag{4.1}$$

missä N ja M ovat järjestyksessä kyselyiden ja avainten (tai arvojen) pituudet. D_k ja D_v ovat järjestyksessä avainten (tai kyselyjen) ja arvojen dimensiot. Lin ym. 2022 jatkavat, että matriisia $\text{softmax}(\frac{QK^T}{\sqrt{D_k}})$ kutsutaan usein attention-matriisiksi, jota sovelletaan riveittäin. Kysely-avain-arvo -mallia monipäisessä attentiossa on havainnollistettu kuviossa 3.

Transformerin käyttämässä monipäisessä attentiossa D_m-ulotteiset alkuperäiset kyselyt, avaimet ja arvot projisoidaan D_k-ulotteisiksi. D_k ja D_v ovat ulottuvuuksia h:n erilaisten opittujen projektioiden avulla. Kullekin ennustetulle kyselylle avaimet ja arvot sekä ulostulo lasketaan attentiolla yhtälön 4.1 mukaisesti. Sitten malli ketjuttaa kaikki ulostulot ja projisoi ne takaisin D_m-ulotteiseksi esitykseksi. (Lin ym. 2022)

Vaswani ym. 2017 mukaan self-attention-funktio kuvaa kyselyn sekä avain- ja arvoparien joukon ulostuloon, jossa kysely, avaimet, arvot ja ulostulo ovat kaikki vektoreita. Syöte las-



Kuvio 3. Monipäinen attention mukaillen (Vaswani ym. 2017). Kuviossa näkyy usea lineaarinen kerros rinnakkain, joiden arvot projisoidaan rinnakkain ja ketjutetaan lopuksi uudella projektiolla D_m-ulotteisiksi esityksiksi.

ketaan arvojen painotettuna summana, jossa kullekin arvolle annettu paino lasketaan kyselyn ja vastaavan avaimen yhteensopivuusfunktion avulla (Vaswani ym. 2017). Toisin sanoen kuvailtu operaatio antaa kontekstin kanssa semanttisen tason painotuksen sanoja edustaville vektoreille.

Maskeeratun monipäisen attention-moduulin ulostulo syötetään monipäiseen attention-moduuliin, jossa on risti-attentio. Tämä ei ole self-attention-moduuli, koska kaikki sen arvot, avaimet ja kyselyt eivät ole samasta sekvenssistä, vaan sen arvot ja avaimet ovat kooderin ulostulosta ja kyselyt dekooderin maskeeratun monipäisen attention-moduulin ulostulosta. Toisin sanoen arvot ja avaimet ovat peräisin käsitellyistä syötteen upotuksista ja kyselyt käsitellyistä ulostulon upotuksista. Laskettu monipäinen attention määrittää, kuinka paljon kukin ulostulo käsittelee ulostulon upotuksia, kuinka paljon kukin ulostulon upotuspari käsittelee ulostulon upotuspareja, kuinka paljon kukin ulostulon upotusparien pari käsittelee ulostulon upotusparien pareja ja niin edelleen. Tämä osoittaa syötesarjan ja tuotetun ulostulosarjan välisen yhteyden. (Ghojogh ja Ghodsi 2020)

Syvässä transformerin kooderi- ja -dekooderimallissa dekooderin risti-attentio-moduulit hyödyntävät vain kooderin lopullisia ulostuloja, joten virhesignaalin on kuljettava kooderin sy-

vyyttä pitkin. Tämä tekee transformerista alttiimman optimointiongelmille. (Lin ym. 2022)

Monipäisen risti-attention-moduulin ulostulo normalisoidaan ja lisätään sen syötteeseen. Sen jälkeen se syötetään eteenpäin syöttävään neuroverkkoon (engl. feed-forward neural network), jossa kerrokset normalisoidaan ja lisätään sen syötteeseen jälkeenpäin. Syöttöverkon ulostulo kulkee lineaarisen kerroksen läpi lineaarisella projektiolla, ja lopuksi käytetään softmax-aktivointifunktiota. Softmax-aktivointifunktioilla varustettujen ulostuloneuronien määrä on yhtä suuri kuin sanaston kaikkien sanojen määrä. Dekooderin ulostulojen summa on yksi ja ne ovat jokaisen sanastossa olevien sanojen todennäköisyydet olla seuraava generoitu sana. Sekvenssin muodostamisessa merkki tai sana jolla on suurin todennäköisyys, on seuraava sana. (Ghojogh ja Ghodsi 2020)

4.3 BART ja DistilBART

Usein käytettyjä BART-arkkitehtuuriin liittyviä transformer-malleja ovat seuraavat:

- `facebook/bart-large-cnn`[1],
- `lidiya/bart-large-xsum-samsum`[2],
- `linydub/bart-large-samsum`[3],
- `philschmid/bart-large-cnn-samsum`[4] ja
- `sshleifer/distilbart-cnn-12-6`[5].

Yllä mainituista transformereista muut perustuvat Lewis ym. 2020 julkisesti julkaisemaan BART-arkkitehtuurimalliin, paitsi viimeksi mainittu perustuu DistilBART-malliin.

Lewis ym. 2020 mukaan BART (engl. Bidirectional and Auto-Regressive Transformers) käyttää standardia Vaswani ym. 2017 kehittämää transformer-arkkitehtuuria, paitsi GPT:n mukaisesti aktivointifunktiot ovat muokattu ReLU:sta (engl. Rectified Linear Unit) GELU:ksi (engl. Gaussian Error Linear Units).

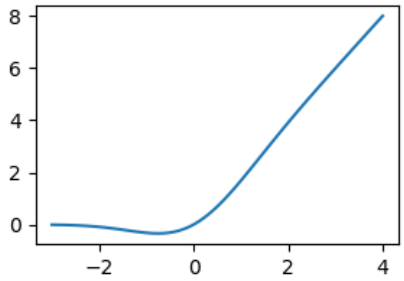
1. https://huggingface.co/facebook/bart-large-cnn
2. https://huggingface.co/lidiya/bart-large-xsum-samsum
3. https://huggingface.co/linydub/bart-large-samsum
4. https://huggingface.co/philschmid/bart-large-cnn-samsum
5. https://huggingface.co/sshleifer/distilbart-cnn-12-6

ReLU on luonteeltaan lineaarinen, jossa negatiiviset arvot kuvautuvat nollaksi ja aidosti positiiviset arvot tulevat ulostulona samansuuruisina, kuin syötearvot ovat. Saaduista arvoista valitaan suurin. (Clevert, Unterthiner ja Hochreiter 2015)

Hendrycks ja Gimpel 2016 mukaan GELU on epälineaarinen, jossa syöte x skaalataan sen mukaan, kuinka paljon suurempi se on kuin muut syötteet. Gaussin kumulatiivinen jakaumafunktio lasketaan usein Hendrycks ja Gimpel 2016 mukaan virhefunktion kanssa. Hendrycks ja Gimpel 2016 antavat tarkan määritelmän GELU:lle sen kanssa. Heidän mukaan määritelmää voidaan approksimoida seuraavasti:

$$f(x) = 0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)]).$$

Kuviossa 4 approksimaatiota on havainnollistettu graafina.



Kuvio 4. Esimerkki GELU-aktivointifunktion graafista.

Transformerina BART on kohinanpoistoautokooderi sekvenssistä sekvenssiin -mallien esiopetusta varten (Guan, Smetannikov ja Tianxing 2020). Sen sekvenssiraja on 1024 esiintymää sekvenssissä (Hartl ja Kruschwitz 2021). BART:in rakenne käsittää kaksisuuntaisen kooderin ja yksisuuntaisen (vasemmalta oikealle) dekooderin ja se on esiopetettu vääristämällä tekstiä satunnaisesti mielivaltaisella kohinafunktiolla ja oppimalla rekonstruoimaan alkuperäisen tekstin (Guan, Smetannikov ja Tianxing 2020). Alkuperäisen tekstin kokoaminen uudelleen vääristyneestä tekstistä auttaa mallin oppimisprosessia (Rahali ja Akhloufi 2023).

Koska BART:ssa on autoregressiivinen dekooderi, se voidaan suoraan hienosäätää sekvenssin luomistehtäviä varten kuten yhteenvedon tekemiseen. Se tapahtuu kopioimalla syötteinformaatio manipuloituna, joka on läheisesti yhteydessä kohinanpoisto esiopetustavoittee-

seen. Kooderin syöte on syötesekvenssi ja dekooderi generoi tulosteen autoregressiivisesti. (Lewis ym. 2020)

Autoregressiivinen malli ennustaa seuraavan sanan sen edellisen kontekstin perusteella (Naseem ym. 2021). Rahali ja Akhloufi 2023 mukaan autoregressiiviset transformer-mallit perustuvat opetusvaiheessa käytettävään peittämismekanismiin. Singh ja Mahmood 2021 mukaan dekooderissa vain aiemmat aika-askeleen kohdemerkit otetaan huomioon tulevan kohteen ennustamisessa, jota kutsutaan kausaaliseksi peittämiseksi. Näin mahdollistetaan kielenkääntämistehtävissä myöhemmin käännettävien kohdemerkkien maksimaalinen oppiminen. Tämän vuoksi rinnakkaistamisen aikana matriisioperaatioiden avulla varmistetaan, että seuraavat kohdesanat peitetään nolliksi, jotta attention-verkko ei voi nähdä tulevaisuuteen (Singh ja Mahmood 2021).

Alkuperäisen transformer-mallin dekooderi on analoginen Rahali ja Akhloufi 2023 mukaan autoregressiivisen transformerin kanssa, joka peittää koko lauseen maskilla. Tekstingeneroiminen on ilmeisin käyttötapaus näille malleille, vaikka niitä voidaan muuttamalla käyttää erinomaisesti myös muunlaisissa tehtävissä (Rahali ja Akhloufi 2023).

Hartl ja Kruschwitz 2021 mukaan `sshleifer/distilbart-cnn-12-6` on pienempi BART-malli ja se on opetettu uutisyhteenvetodatasetillä, jonka ovat luoneet Hermann ym. 2015. Hermann ym. 2015 mukaan he ovat käyttäneet CNN:n ja DailyMailin artikkeleita ja niiden yhteenvetoja. DistilBART on suunniteltu tarkkaan ja nopeaan päättelyyn yhteenvedon tekemistä varten (Wolf ym. 2020).

Tietämyksen purkaminen (engl. knowledge distillation) tarkoittaa menetelmiä, joiden avulla voidaan opettaa uusi pienempi oppilasverkko oppimalla opettajaverkosta opetusdatan lisäksi. Yleensä oletetaan, että opettaja on koulutettu aiemmin, ja opiskelijan parametrit estimoidaan sovittamalla opiskelijan ennusteet opettajan ennusteisiin. (Liu, Shen ja Lapata 2020)

BART:in osalta Shleifer ja Rush 2020 ovat käyttäneet tietämyksen purkamista vastaamaan opettajan suorituskykyä. He esittävät tietämyksen purkamiseen kolme päävaihetta. Ne ovat esitetty seuraavassa.

1) Kutistamisessa ja hienosäädössä kutistetaan opettajamalli oppilaan kokoon ja hieno-

säätämällä oppilasmalli uudelleen. Alustuksen jälkeen opiskelijamalli jatkaa hienosäätöä yhteenvetodatasetin perusteella, ja tavoitteena on minimoida standardi ristientropian häviö (engl. cross entropy loss).
2) Pseudomerkintäasetelmassa korvataan kohdedokumenttien pohjatotuudet (engl. ground truth) opettajan tuottamilla lähdedokumenteilla, jotka on laskettu sädehaun (engl. beam search) avulla. Toimenpiteen jälkeen opiskelijamallia hienosäädetään vain uuden pseudomerkityn datan perusteella.
3) Suorassa tiedon poistamisessa opettajalta oppilaalle siirretään vielä enemmän tietoa kannustamalla oppilasta noudattamaan opettajan täydellistä todennäköisyysjakaumaa mahdollisista seuraavista sanoista kussakin kohdassa minimoimalla KL-divergenssi (Kullback ja Leibler 1951).

4.4 Philschmid-transformer

Pro gradu -tutkimuksen koeasetelmassa käytetty `philschmid`-transformer on Yamaguchi ym. 2021 luoma. Toimintalogiikka hyödyntää viittausvapaasti Ghosal ym. 2021 ehdottamaa automaattista pöytäkirjajärjestelmää. Siinä on automaattinen viittausvapaa pöytäkirjakirjaamisliukuhihna (engl. minuting pipeline), joka perustuu aihekohtaisen yhteenvedon sijaan argumenttirakenteisiin (Yamaguchi ym. 2021).

Kirjaaminen NLP-tehtävänä liittyy läheisesti yhteenvetoon. Tehtävissä on silti eroa. Vaikka tekstin tiivistäminen on motivoitunut luomaan johdonmukaisen ja tarkan yhteenvedon annetusta tekstisisällöstä, niin kirjaaminen on tarkoitettu vain kokouksiin. (Ghosal ym. 2021)

Automaattinen kirjausjärjestelmä on motivoitu Ghosal ym. 2021 mukaan NLP-yhteisöjen yhteisistä tehtävistä ja haasteista, jotka ovat nykyisen kukoistavan tekstin tiivistämisen yhteisön kehittymisessä olleet merkittävässä osassa. NLP-yhteisön kamppanjoissa on hyödynnetty ponnisteluja monien ongelmien ratkaisemiseksi (Ghosal ym. 2021). Ghosal ym. 2021 kehittämä kokousten kirjaamiseen tarkoitettu järjestelmä on eräs näistä.

Automaattisen pöytäkirjan kirjausjärjestelmään perustuvan `philschmid`-transformerin jaetut tehtävät käsittävät pääasiallisen tehtävän A ja kaksi alatehtävää B ja C. Tehtävä A tähtää generoimaan pöytäkirjan kokouspöytäkirjoista. Tehtävä B tarkastelee, kuuluuko annettu

pöytäkirja kokouspöytäkirjoihin ja C tarkastelee, kuuluuko kaksi annettua pöytäkirjaa samaan (samoihin) kokoukseen (kokouksiin). (Yamaguchi ym. 2021)

Tehtävää A varten Yamaguchi ym. 2021 käyttämä liukuhihna (engl. pipeline) koostuu pääasiassa segmentoinnin, yhteenvedon ja argumenttien louhintamoduuleista. Segmentointimoduuli on suunniteltu poimimaan ilmaisuja lohkoittain. Lohkojen ilmaisujen pitäisi mainita sama aihe kokouspöytäkirjoissa. Samalla tämä moduuli voi suodattaa epäolennaiset ilmaisut sulkemalla ne pois lohkosta. Yamaguchi ym. 2021 ovat rakentaneet Longformer-pohjaisen segmentointimoduulin, joka on opetettu manuaalisesti merkityllä aineistolla, joka on poimittu englanninkielisestä opetusaineistosta. Yhteenvetomoduuli luo yhteenvedon ilmaisulohkosta. Sen saavuttamiseksi he ovat käyttäneet SAMSum-korpuksen[6] pohjalta hienosäädettyä valmiiksi koulutettua BART-mallia. Koska malli pystyy osittain ratkaisemaan viiteongelman segmentointi- ja yhteenvetomoduulien avulla, saadaan useimmiten kattava yhteenveto syötetystä kokouspöytäkirjasta. Lopuksi strukturoidaan tiivistetyt lohkot ja muotoillaan tuloksena syntynyt pöytäkirja valmiilla argumenttien louhinnan jäsentimellä.

Tämän moduulin Yamaguchi ym. 2021 ovat johtaneet havainnostaan, jonka mukaan suurin osa referenssipöytäkirjoista on jäsennelty erittelyn avulla. Vaikka mainituissa rakenteiden muodoissa, näkökohdissa ja näkökulmissa on paljon vaihtelevuutta, he olettavat, että kaikki johdonmukaiset rakenteet parantaisivat luettavuutta ja antaisivat lukijoille mahdollisuuden tunnistaa tiettyjä näkökohtia ja näkökulmia pöytäkirjoihin. Tällaisten rakenteiden muodostamiseen pöytäkirjaan Yamaguchi ym. 2021 käyttävät argumenttien louhinnan jäsentäjää, joka voi ennustaa argumenttimerkinnät ja perustelut kullekin virkkeelle. Tämän saavuttamiseksi Yamaguchi ym. 2021 käyttävät SAMSum-korpuksen pohjalta hienosäädettyä, valmiiksi opetettua BART-mallia. Koska malli pystyy osittain ratkaisemaan viiteongelman segmentointi- ja yhteenvetomoduulien avulla, saadaan useimmiten kattava yhteenveto syötetystä kokouspöytäkirjasta. Lopuksi strukturoidaan tiivistetyt lohkot ja muotoillaan tuloksena syntyvä kokouspöytäkirja valmiilla argumenttien louhinnan jäsentimellä.

6. *SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization* (Gliwa ym. 2019).

5 Chatbottien dialogit

Perinteisiä parhaita käytänteitä, joita tavallisesti sovelletaan käyttäjäkokemuksen suunnitteluun, ei voida helposti soveltaa chatbotteihin (Holmes ym. 2019). Luvussa tehdään katsaus olennaisimpiin chatbottien tulosteisiin liittyviin ohjenuoriin ja tarkastellaan käytännön tuloksia tulosteiden pituuksista.

5.1 Chatbottien ilmaisujen pituuden merkitys chatbotin laatuun

Cañizares ym. 2022 tutkimuksen mukaan chatbottien käyttäjäkokemusta huonontaa muun muassa vaikealukuiset vastaukset. Luettavuuden mittaamista he arvioivat tarkastelemalla chatbotin tulosteen ilmaisuissa

1. sanojen lukumäärää,
2. verbien lukumäärää,
3. merkkien määrää ja
4. lukemisaikaa.

Kohdat 1, 2 ja 3 vaikuttavat chatbotin tehokkuuteen. Kohdat 1, 3 ja 4 vaikuttavat käyttäjätyytyväisyyteen. Suuri merkkimäärä chatbotin ilmaisussa vähentää käyttäjien tyytyväisyyttä ja chatbotin tehokkuutta. Myös Ruane, Farrell ja Ventresque 2020 tutkimuksen mukaan leksikaaliset ominaisuudet, kuten keskusteluvuorossa käytettyjen sanojen tai merkkien määrä, voivat olla yhteydessä chatbotin informatiivisuuteen ja tehokkuuteen. Cañizares ym. 2022 mukaan chatbottien tulosteiden pituudessa ohjenuorana voidaan pitää Moore ym. 2018 esiin tuomaa minimoimisen periaatetta: Keskustelukeskeisissä käyttöliittymissä suunnittelijoiden tulee pyrkiä pitämään keskusteluagentin ilmaisut mahdollisimman lyhyinä. Sacks ja Schegloff 1979 mukaan inhimillisessä vuorovaikutuksessa nimenomaan minimointi on etusijalla. Moore ym. 2018 mukaan digitaalisista kokemuksista pyritäänkin tekemään mahdollisimman inhimillisiä ja siten keskustelun hallitsemia.

Langevin ym. 2021 ovat mukailleet Nielsenin heuristiikkoja arvioidessaan chatbottien käyttöliittymiä. Niiden tulisi olla esteettisiä, minimalistisia ja mukaansatempaavia. Dialogit eivät

saisi sisältää epäolennaista tai harvoin tarpeellista tietoa. Langevin ym. 2021 tutkimuksessa kerrotaan vastaavan Nielsenin heuristiikan sanovan seuraavaa: ”Dialogit eivät saa sisältää epäolennaista tai harvoin tarpeellista tietoa. Jokainen ylimääräinen informaatioyksikkö dialogissa kilpailee asiaankuuluvien tietoyksiköiden kanssa ja heikentää niiden suhteellista näkyvyyttä.”

Cañizares ym. 2022 mukaan sosiaaliseen mediaan pitkät ilmaisut eivät ole sopivia tai aina edes mahdollisia. Esimerkkinä he mainitsevat Twitterin, jossa on viestille tiukka pituusrajoitus. Liian pitkä chatbotin ilmaisu jäisi siten esimerkiksi Twitterissä keskeneräiseksi. Cañizares ym. 2022 tuovat lisäksi esiin, että pitkät chatbottien tulosteet voivat tuottaa käytännön ongelmia, kuten vaatia näytön skrollaamista mobiililaitteilla. Esimerkkinä tästä he mainitsevat Telegrammissa toimivan chatbotin, joka antaa tietoa koronaviruksesta. Chatbotin tuottama merkkimäärä oli 285. Tapaus osoitti Cañizares ym. 2022 mukaan chatbotin kannalta heikkoa käytettävyyttä ja edustavan tulosteen heikkoa luettavuutta. Höhn ja Bongard-Blanchy 2020 ovat evaluoineet 24 koronaviruksesta (COVID-19) tietoa jakavaa chatbottia. Heidän tuloksissaan tulee esiin, että näistä 65% olivat dialogien osalta ytimekkäitä ja täsmällisiä. Ne toteuttivat minimalistisuuden ja esteettisyyden suunnittelun heuristisen ohjenuoran.

Moore ym. 2018 mukaan keskustelukeskeisen vuorovaikutuksen vastaukset ovat suhteellisen lyhyitä tai napakoita. Tämä mahdollistaa Moore ym. 2018 mukaan tehokkuuden ja nopeuden, kun keskustelut tapahtuvat reaaliajassa joko puhe- tai teksti-ilmaisuina. Erityisen ongelmallisia pitkät chatbottien ilmaisut ovat Cañizares ym. 2022 mukaan äänipohjaisissa chatboteissa, koska puhuminen kestää kauemmin kuin lukeminen.

Myös Yang ja Aurisicchio 2021 ehdottavat tutkimuksessaan, että chatbottien vastakset tulisi esittää ytimekkäästi ja informatiivisesti. He jatkavat, että tapauksissa, joissa vastaukset ovat verkon hakutulosten muodossa, nykyisillä chatboteilla on erilaisia esitysmuotoja. Joskus ne antavat lyhyen yhteenvedon hakulinkin kera.

5.2 Chatbottien ilmaisujen pituuden merkitys käyttäjälle

Moore ym. 2018 mukaan äänirajapintojen avulla selviää nopeasti, ovatko keskusteluagentin puheet liian pitkiä. He jatkavat, että koska puheen tuottaminen kestää suhteellisen kau-

an, suunnittelijoiden tulee myös vastustaa kiusausta olla tekemättä keskusteluagentin tekstiilmaisuista liian monisanaisia. Vaikka lukeminen on nopeampaa kuin kuunteleminen, käyttäjät eivät välttämättä lue agentin kaikkia vastauksia, jos ne ovat tarpeettoman pitkiä.

Cañizares ym. 2022 mukaan pitkät chatbottien ilmaisut ovat vaikeammin ymmärrettävissä sosiaalisessa mediassa. Kognitiivista kuormaa käyttäjälle lisää myös muutoin kuin tekstimuodossa chatbotilta tuleva rikas sisältö. Cañizares ym. 2022 jatkavat, että suuri merkkimäärä chatbotin tulosteessa lisää vaaraa, että chatbotin ilmaisua ei lueta kokonaan. Lisäksi Sugisaki ja Bleiker 2020 mukaan eräs ohjenuora keskustelukeskeisten käyttöliittymien suunnittelussa on ihmisen lyhytkestoisen muistin kuormittamisen vähentäminen. Heidän tekemästään kyselytutkimuksesta kävi myös ilmi, että keskustelukeskeisten käyttöliittymien suunnittelussa ilmaisujen sopivaa pituutta piti 80% arvioijista erittäin merkittävänä seikkana sekä 20% piti tätä jokseenkin merkittävänä. Arvioijista siis 100% piti chatbottien ilmaisujen sopivaa pituutta vähintään jokseenkin merkittävänä. Arvioija oli 15 erilaisilla työkuvataustoilla käsittäen esimerkiksi keskusteluagenttien lingvistiikkaan ja NLP:hen erikoistuneita ihmisiä.

Rooein ym. 2022 ovat ehdottaneet käyttäjän toimintaan adaptoituvaa chatbot-kehystä. Heidän mukaan adaptiivisuutta lisäävät keskusteluominaisuudet kuten tyyli ja sanamuodot, voidaan säätää käyttäjän profiilia ja käyttökontekstia koskevien erityistarpeiden mukaan. Rooein ym. 2022 jatkavat, että näin voidaan myös ohittaa dialogista ei-toivotut yksityiskohdat ja ehdottaa käyttäjille ja heidän toimintakontekstilleen sopivampia dialogeja. Lisäksi adaptiivinen chatbot voi ohjata keskusteluvuorojen määrää ja lausumien pituutta. Esimerkiksi, jos käyttäjä pitää ystävällisistä keskusteluista asiakeskustelujen sijaan, chatbot merkitsee tämän käyttäjän profiiliin ja muuttaa sitä myöhemmin käyttäjän pyynnöstä. Sitä laajennetaan myös muihin chatbotin asetuksiin, kuten virkkeiden pituuteen tai keskusteluvuorojen määrään.

Ruane, Farrell ja Ventresque 2020 mukaan käyttäjän osalta sanojen ja merkkien määrä voi olla merkki käyttäjän sitoutumisesta, kun taas chatbotin osalta käytetty sanojen määrä voi heijastaa sen persoonallisuuden piirteitä. Völkel ym. 2022 mukaan ekstrovertti chatbot tuottaa pitkiä ilmaisuja, joissa on suurempi määrä sanoja kuin keskiverto introvertti chatbot. Lisäksi ekstrovertti chatbotin kieli on epävirallisempaa.

5.3 Chatbottien ilmaisujen pituuksista

Jain ym. 2018 ovat arvioineet usean eri tahon chatbottien informointiominaisuuksia. Heidän tutkimustuloksistaan ilmenee, että korkein keskimääräinen merkkimäärä chatbotin viestiä kohden oli *CNN*:n chatbotilla, 84.3 $\pm$ 60.2 merkkiä. Tätä lähimpänä olivat *Hi Poncho*n ja *Call of Duty*n chatbotit, joiden keskimääräiset merkkimäärät olivat järjestyksessä 79.4 $\pm$ 55.1 ja 73.8 $\pm$ 38.1. Suurin keskimääräinen merkkimäärä on ollut *CNN*:n chatbotilla noin 144.5 merkillä. Toisaalta *CNN*:n chatbotin keskimääräinen merkkimäärä on ollut myös vertailussa pienimmillään pienin, noin 24.1 merkkiä. Tutkimuksessa tuli esiin lisäksi, että chatbottien tekemää kokonaisten uutisartikkelien avaamista toisiin selaimen välilehtiin pidettiin joskus turhauttavana, koska tuolloin joutui jättämään senhetkisen selaimen välilehden. Aiemmat tutkimukset tukevat Jain ym. 2018 mukaan kuitenkin ulkoisten linkkien laittamista chatbottien tulosteisiin.

Hill, Ford ja Farreras 2015 tutkimuksessa on tullut esiin, että ihmiset kirjoittavat chatboteille lyhyemmin kuin ihmiskumppaneilleen verkkokeskusteluissa. Heidän tutkimuksessaan tuli esiin, että sanamäärä vaihteli välillä 2–13 ihmiseltä chatbotille. Keskimäärin ihmiset kirjoittivat chatbotille 7.95 sanaa viestiä kohden. Vastaavasti chatbot vastasi ihmiselle keskimäärin sanamäärällä 4.29 viestiä kohden. Tutkimuksen mukaan kuitenkin keskustelujen kesto oli ihmiseltä chatbotille suurempi kuin ihmisten kesken. Tutkimuksessa käytetty chatbot oli Cleverbot. Hill, Ford ja Farreras 2015 mukaan Cleverbot läpäisi vuonna 2011 Turingin testin.

Kung ym. 2023 ovat mitanneet ChatGPT:n tuottamien selitysten sisältämän tiedon tiiviyttä lääketieteellisissä monivalintakysymyksissä, joiden vastauksille vaadittiin sanallinen perustelu. Perusteluiden tiheysindeksi määritettiin normalisoimalla uniikkien oivallusten määrä suhteessa mahdollisten vastausvaihtoehtojen määrään. Korkealaatuisille perusteluille oli yleensä ominaista oivalluksen tiiviys.

6 Tulokset ja niiden analysointi

Tulosten luomisessa on käytetty englanninkielistä lähdemateriaalia, koska tulosten tuottamisen hetkellä NLP-tekniikoilla ja -työkaluilla oli parempi tuki englannin kielelle kuin suomen kielelle. Lisäksi tuloksia varten käytettyjä datasettejä ei välttämättä löydy lainkaan suomen kielelle.

6.1 Yhteenvedot

Tuloksia varten Wikidatasta generoitiin SPARQL-kielellä 25000 satunnaista tieteentekijän nimeä. Generoimisjärjestyksessä pyrittiin löytämään vastaava englanninkielinen Wikipedia-sivu ja tuottamaan yhteensä 1000 sivusta yhteenveto. Ne pyrittiin tuottamaan sektioittain Wikipedia-sivuista neljällä transformerilla, joiden sekvenssinpituus on 1024 esiintymää sekvenssissä. Lopuksi sektioiden yhteenvedot tuotettiin vielä yhdeksi yhteenvedoksi. Transformerit olivat koko nimeltään seuraavat:

- `facebook/bart-large-cnn`,
- `lidiya/bart-large-xsum-samsum`,
- `philschmid/bart-large-cnn-samsum` ja
- `sshleifer/distilbart-cnn-12-6`.

Jos edellä kuvailtu onnistui kaikilla neljällä transformerilla, niin vielä viidennellä transformerilla yritettiin tehdä kerralla koko Wikipedia-sivusta yhteenveto. Transformerin sekvenssinpituus on 4096 ja sen koko nimi on

- `linydub/bart-large-samsum`.

Wikipedia-sivujen yhteenvetojen tuloksiin otettiin ne sivut, joille edellä kuvattu operaatio onnistui kokonaisuudessaan. Transformereiden tulokset tuotettiin 16.10.2022–31.10.2022.

Ekstraktiiviset yhteenvedot Wikipedia-sivuista tuotettiin 5.11.2022 samoille Wikipedia-sivuille kuin transformereilla. Ekstraktiivisille menetelmille Wikipedia-sivut syötettiin samoin kuin transformereille eli mitään sektioita ei karsittu pois. Tämä heikensi ekstraktiivisten yh-

teenvetotulosten tasoa, koska ne ottivat joskus yhteenvetoon mukaan erilaisia Wikipedia-sivun lopussa olevia luetteloiden osia. Näin saatiin kuitenkin käsitys perinteisten ekstraktiivisten menetelmien ja abstrahoivien transformereiden kyvystä käsitellä lähdetekstiä.

Lin 2004 mukaan ROUGE-1 ja ROUGE-L F1-pisteet ovat sopivia lyhyiden yhteenvetojen arviointiin. Tuotetut yhteenvedot ovatkin suhteellisen lyhyitä. Yhteenvetojen ROUGE-tulokset ovat määritetty käyttäen Pythonin `rouge`-kirjastoa. Kaikki yhteenvetojen tulokset ovat ilmaistu neljän desimaalin tarkkuudella.

6.1.1 Ekstraktiivisten yhteenvetojen tulokset

Ekstraktiivisia yhteenvetoja arvioitiin käyttämällä kolmea eri menetelmää: Luhn, LSA ja TextRank. Ne ovat sisäänrakennettuna Pythonin `sumy`-kirjastossa, jota käytettiin ekstraktiivisille menetelmille. Menetelmät valittiin aikaisempien tutkimusten valossa perinteisistä ekstraktiivisista historiallisesti tärkeinä ja keskenään eritavoin toimivina menetelminä.

Ekstraktiiviset yhteenvedot tuotettiin 1000 eri Wikipedia-sivulle. Virkkeiden määräksi asetettiin neljä yhteenvetoja tehdessä kullekin menetelmälle. Tuloksien keskiarvot ovat esitetty taulukossa 1.

Ekstraktiivisten yhteenvetojen ROUGE F1-pisteiden keskiarvot			
	Luhn	**LSA**	**TextRank**
ROUGE-1	**0.1761**	0.1707	0.1754
ROUGE-2	0.0454	0.038	**0.0462**
ROUGE-L	**0.1585**	0.1526	0.1574

Taulukko 1. Kolmen ekstraktiivisen yhteenvetomenetelmän ROUGE F1-pisteiden keskiarvot 1000 Wikipedia-sivusta.

Huomionarvoista on, että tehdyssä koejärjestelyssä vanhin menetelmä, joka on myös ensimmäinen koskaan esitelty menetelmä, on menestynyt keskimäärin parhaiten. Kuitenkin on huomattava, että keskiarvo on herkkä poikkeaville arvoille. Luvussa 6.2.1 tarkastellaan, kuinka F1-pisteet jakautuivat eri menetelmillä.

6.1.2 Abstrahoitu yhteenveto Wikipedialla

Abstrahoidun yhteenvedon tuottamista arvioitiin käyttäen viittä eri transformeria samoille 1000 eri Wikipedia-sivulle kuin luvussa 6.1.1 esitettyjen tulosten kanssa. Transformerit ovat esitelty luvun 6.1 alussa. Transformereilla generoitujen yhteenvetojen suurimmaksi merkkimääräksi asetettiin 450. Malliyhteenvetoina käytettiin verkossa olevia Wikipedian yhteenvetoja. Ne sopivat joskus hyvin malliyhteenvedoiksi, koska Wikipedian sisältö perustuu periaatteessa usean ihmisen konsensukseen. Yhteenvetojen ROUGE F1-pisteet ovat esitetty taulukossa 2.

Transformereiden ROUGE F1-pisteiden keskiarvot					
	facebook	**lidiya**	**linydub**	**philschmid**	**sshleifer**
ROUGE-1	0.2219	0.2351	0.2105	**0.2375**	0.2192
ROUGE-2	0.0586	0.0644	0.059	**0.0667**	0.0527
ROUGE-L	0.2073	0.2187	0.1951	**0.2219**	0.2025

Taulukko 2. Viiden transformerin ROUGE F1-pisteiden keskiarvot 1000 Wikipedia-sivusta.

Tuloksista käy ilmi, että `philschmid`-transformerin ROUGE F1-pisteet ovat korkeimmat. Siten tulosten perusteella suhteessa Wikipedian ihmisten tekemiin malliyhteenvetoihin nähden sen yhteenvedot ovat keskimäärin lähimpänä malliyhteenvetoja.

6.1.3 Abstrahoitu yhteenveto CiteSum-datasetilla

Wikipedia-koeasetelmassa parhaiten menestynyttä `philschmid`-transformeria testattiin myös käyttämällä CiteSum-nimistä datasettiä. Mao, Zhong ja Han 2022 mukaan CiteSum on opetusdatasetti tieteellisten artikkelien hyvin tiiviiseen tiivistämiseen. Se on haasteellinen datasetti transformerin arviointiin, jota ei ole opetettu sillä. Mao, Zhong ja Han 2022 mukaan datasetilla opetuksessa lähdevittaukset ovat asetetaan muotoon ”REF” esimerkiksi ”author name et al.” tyyppisen ilmaisun sijaan. ”REF”-kirjainyhdistelmällä on myös korvattu ilmaisut kuten ”this paper”. `Kaggle.com`:ssa CiteSum-datasetin[1] käytettävyysarvo on 10.00.

CiteSum-datasetissä on opetukseen tarkoitetussa testijoukossa duplikaattien poistamisen jäl-

1. https://www.kaggle.com/datasets/nbroad/cite-sum Viitattu 23.1.2023.

keen 4805 lähdetekstiä malliyhteenvedon kanssa. Näistä 3600 ensimmäistä otettiin koeasetelmaan. Yhteenvetoja tuotettaessa pienimmäksi ja suurimmaksi sanamääräksi asetettiin Babar, Tech-Cse ja Rit 2013 mukaan informatiivisen yhteenvedon mukainen sanamäärä järjestyksessä 20 ja 30% lähdetekstistä. Taulukossa 3 ovat ROUGE F1-pisteet ja METEOR-pisteet näin saaduille yhteenvedoille.

CiteSum-datasetti	
	Pisteet
ROUGE-1	0.2474
ROUGE-2	0.0881
ROUGE-L	0.1944
METEOR	**0.2909**

Taulukko 3. ROUGE F1-pisteiden keskiarvot ja METEOR-pisteet CiteSum-datasetille.

ROUGE-1:n ja ROUGE-2:n osalta tulokset ovat vain hieman paremmat kuin Wikipedia-koeasetelmassa. ROUGE-L:n osalta tulos on huonompi kuin Wikipedian osalta vastaavassa koeasetelmassa. METEOR-metriikan pisteiden keskiarvoksi saatiin hieman parempi tulos kuin ROUGE F1-psiteille. METEOR-pisteiden mediaani on hieman alle niiden keskiarvon, 0.2635.

N-grammien päällekkäisyyteen perustuvilla metriikoilla keskimääräiset pisteet jäivät jokseenkin alhaisiksi. Siksi koeasetelmaa päätettiin vielä tarkistaa käyttämällä uudempia laadunarviointimenetelmiä. Menetelmiksi valittiin BERTScore ja CoCo[2].

BERTScoren kanssa käytettiin sen kehittäjien Zhang* ym. 2020 suosittelemaa[3] mallityyppiä `microsoft/deberta-xlarge-mnli`. CoCo:n kanssa käytettiin XSum-datasetillä hienosäädettyä Lewis ym. 2019 luomaa transformer-mallia `bart.large.xsum`[4]. Sen sekvenssin pituusrajoituksen vuoksi 4805:stä uniikeista CiteSum-datasetin lähdeteksteistä ja malliyhteenvedoista rajoituttiin käyttämään alusta lukien 3600 perättäistä, koska kokeilut osoittivat CoCo:n suoriutuvan niistä varmuudella. Siksi myös muilla metriikoilla tehtiin vastaava rajaus, vaikka generoituja yhteenvetoja oli alun perin enemmän.

2. https://github.com/xieyxclack/factual_coco Viitattu 11.4.2023.
3. Suositus on annettu ainakin verkko-osoitteessa https://pypi.org/project/bert-score/. Viitattu 3.5.2023.
4. https://github.com/facebookresearch/fairseq/tree/main/examples/bart Viitattu 11.4.2023.

CoCo-metriikan käyttöönotto edellytti hienoisia muutoksia CoCo:n lähdekoodiin, jotta sen käyttäminen oli ylipäätään mahdollista. Olennaisimmat muutokset käsittivät CoCo:n sisäisten datatiedostojen lukemisen UTF-8-muodossa. Muutokset eivät vaikuta CoCo:n laadunarvioinnin tuloksiin, koska toimintalogiikkaa ei muokattu, ja datatiedostot luetaan muutoksen jälkeen oikein.

Taulukossa 4 ovat esitettynä BERTScore- ja CoCo-metriikalla saadut tulokset. BERTScore viittaa sen F1-pisteisiin.

CiteSum-datasetti	
	Pisteet
BERTScore	**0.6215**
CoCo (token)	0.4411
CoCo (sent)	0.4802
CoCo (span)	0.4555
CoCo (doc)	0.4849

Taulukko 4. BERTScore- ja CoCo-metriikoiden keskiarvot CiteSum-datasetille.

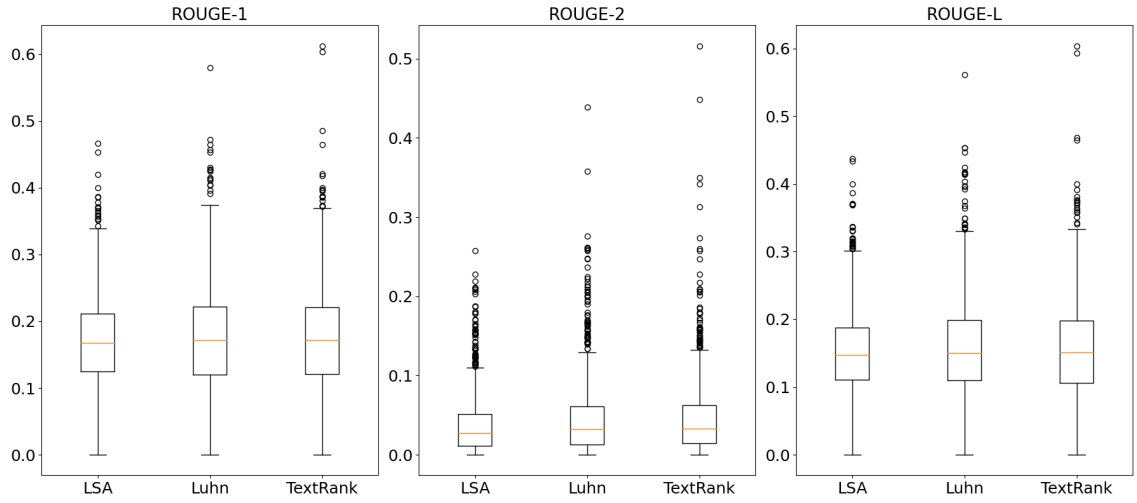
BERTScoren F1-pisteiden keskiarvoksi puolestaan saatiin 0.6215 ja sen mediaaniksi saatiin 0.6163. Saadut tulokset ovat jokseenkin hyvät. CoCo-metriikalla koko dokumenttipohjainen peittämismekanismi tuotti korkeimmat CoCo-metriikan pisteet. CoCo:lla saatiin kaikilla neljällä peittämismekanismilla vähäinen määrä negatiivisia arvoja, mikä on laskenut CoCo-pisteiden keskiarvoa. Mediaani on kaikilla peittämistyypeillä hieman korkeampi. Suurimmillaan mediaanin ja keskiarvon erotus oli kuitenkin vain 0.0074.

6.2 Yhteenvetotulosten analysointi

Tuloksia analysoitaessa on käytetty laatikkokuvioita (engl. boxplot). Laatikkokuvion laatikko käsittää 50% saaduista arvoista ja sen sisällä oleva oranssi viiva edustaa mediaania. Mediaanin ala- ja yläpuolelle jäävät alueet laatikossa kuvaavat tulosten määrää mediaanin suhteen siten, että mediaanin yläpuolella on 25% mediaania suuremmista arvoista, vastaavasti mediaanin alapuolella on 25% sitä pienemmistä arvoista. Lisäksi on nähtävissä 50%:in arvo-osuuden ulkopuolisten arvojen jakaumaa sekä pienin ja suurin arvo.

6.2.1 Wikipedia

Kuviosta 5 nähdään ekstraktiivisten yhteenvetomenetelmien ROUGE F1-pisteiden jakauma. Vaikka Luhnin menetelmä menestyi keskimäärin parhaiten, niin TextRank sai korkeimpia



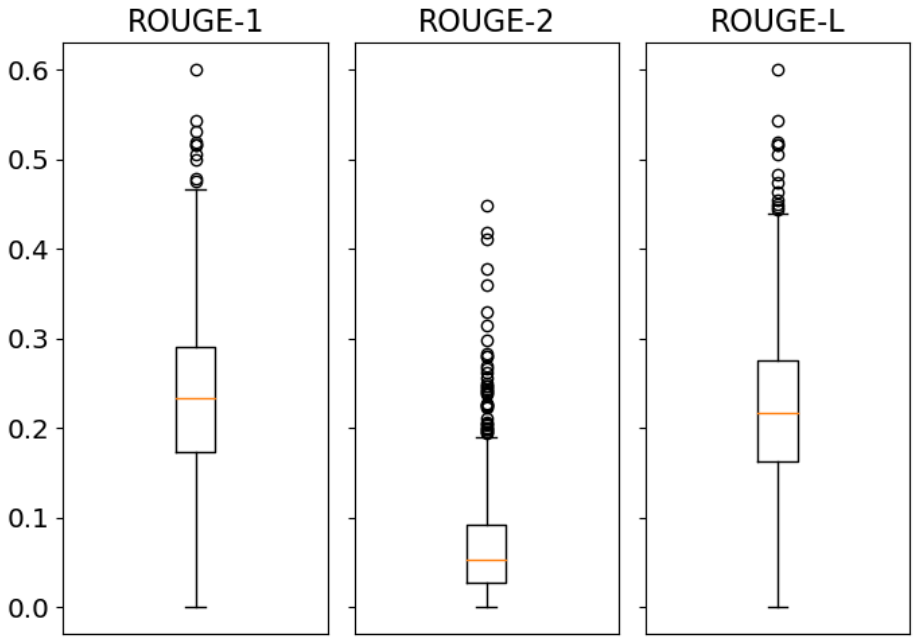
Kuvio 5. Ekstraktiivisten menetelmien ROUGE F1-pisteiden jakaumat.

ROUGE-1 ja ROUGE-L F1-pisteitä. Tarkastelu näyttää, että TextRankin ROUGE F1-pisteet ovat ROUGE-1:n kohdalla 0.2% (2 yhteenvetoa) osalta vähintään 0.6 ja ROUGE-L:n kohdalla 0.1% (1 yhteenveto) osalta vähintään 0.6. Muut ekstraktiiviset menetelmät eivät saaneet koejärjestelyssä näin korkeita pisteitä. Lisäksi TextRankin mediaanit kaikista 1000 yhteenvedoista olivat ROUGE-2 ja ROUGE-L F1-pisteissä korkeimmat. Ero oli kuitenkin hyvin pieni. Suurimmillaan ero TextRankin ja myös Luhnin menetelmää paremmin ROUGE-2:ssa menestyneen LSA:n ROUGE-2:n F1-pisteissä oli vain noin viisi promillea. Muiden arvojen osalta ero TextRankin eduksi parhaimmillaan on vielä pienempi LSA:han nähden.

Ekstraktiiviset menetelmät poimivat joskus yhteenvetoon myös luetteloiden osia, vaikka Wikipedia-sivussa itsessään oli asiatekstiä olennaisesti enemmän kuin neljä virkettä. Näin oli laita esimerkiksi Luhnin menetelmällä keskimääräisesti korkeat ROUGE-1 F1-pisteet 0.3388 saaneen Høgni Reistrupin Wikipedia-sivun yhteenvedon tuottamisen osalta. Yhteenvedon kolme ensimmäistä virkettä olivat olennaista asiasisältöä, mutta neljännen virkkeen osalta yhteenveto jatkui luetteloina. Samoin TextRank-menetelmän osalta tuotetulle Andrea Grimes Parkerin Wikipedia-sivun yhteenvedolle saatiin ROUGE-1 F1-pisteiksi 0.4. Yhteenve-

don virkkeistä kolme ensimmäistä sisälsi olennaista asiasisältöä, mutta neljäs virke koostui luetteloista.

Myös abstrahoivilla yhteenvetomenetelmillä ROUGE F1-pisteet 1000 Wikipedia-sivusta vaikuttavat keskimääräisesti jokseenkin alhaisilta. Tarkasteltaessa keskimäärin parhaimpia ROUGE F1-pisteiden keskiarvoja saaneen `philschmid`-transformerin 1000 Wikipedia-sivun F1-pisteiden jakaumaa, saadaan kuvion 6 mukaiset tulokset. Tarkasteltaessa kuviosta 6 hei-



Kuvio 6. Transformerin `philschmid` ROUGE F1-pisteiden jakauma.

koimpia pisteitä nähdään, kaikista 1000:sta yhteenvedosta kaksi yhteenvetoa ovat saaneet ROUGE-1:n F1-pisteiksi 0.0. Näissä tapauksissa Wikipedian yhteenveto oli yhden virkkeen mittainen selväsanainen yhteenveto, mutta Wikipedia-sivun sektiot, joista automaattinen yhteenveto muodostettiin, koostui vain julkaisuluetteloista, palkintoluetteloista tai ulkoisista linkeistä. Kyseisten henkilöiden elämää itseään, jota Wikipedian yhteenveto koski, ei käsitelty lainkaan.

Vastaavasti korkeimmat ROUGE-1 F1-pisteiden saaneiden henkilöiden Wikipedia-sivuilla sektioissa käsiteltiin kyseisen henkilön elämän eri osa-alueita, jolloin on ollut ylipäätään mahdollista muodostaa jonkinlainen yhteenveto kyseisestä Wikipedia-sivusta. Esimerkiksi parhaat ROUGE-1 F1-pisteet, 0.6, saanut yhteenveto oli `philschmid`-transformerilla seu-

raava:

Huitfeldt was elected as a full representative to Parliament for the first time in 2005. She served as the deputy leader of the Standing Committee on Education, Research and Church Affairs from 2005 to 2008 and from 2013 to 2021. She was Minister of Children and Equality from 2008 to 2009, Minister of Culture from 2009 to 2012 and Minister of Labour and Social Inclusion from 2012 to 2013. She is married to Ola Petter Flem and has three children.

Wikipedian malliyhteenveto[5] puolestaan oli:

Anniken Scharning Huitfeldt (born 29 November 1969) is a Norwegian historian and politician for the Labour Party. She has served as Minister of Foreign Affairs since 2021. She previously served as Minister of Children and Equality from 2008 to 2009, Minister of Culture from 2009 to 2012 and Minister of Labour and Social Inclusion from 2012 to 2013.

Tarkasteltaessa yhteenvetojen eroja, huomio kiinnittyy ensimmäiseen virkkeeseen. Wikipedian malliyhteenvedossa olevaa Anniken Huitfeldtin syntymäaikaa ja etunimeä ei ole mainittu lainkaan sektioissa, joista transformer on rakentanut yhteenvedon. Siksi transformer ei ole voinut sisällyttää omaan yhteenvetoonsa Huitfeldtin etunimeä eikä syntymäaikaa. Wikipedian sektioissa on useita viitteitä norjalaisuuteen. Kuitenkaan sektioista ei eksplisiittisesti tule selväksi, että Huitfeldt on norjalainen. Ihmisen lukiessa hänestä kertovia sektioita, ei myöskään ihminen voisi ehdottoman varmasti vakuuttua, että Huitfeldt olisi ainakaan varmuudella syntynyt Norjassa tai olisi tällä hetkellä norjalainen, vaikkakin selväksi tulee, että hän on toiminut Norjassa politiikassa. Siten transformeria ei voi moittia myöskään sektioista tuotetun yhteenvedon osalta ehdottomasti siitä, että siinä ei mainita, että Huitfeldt on norjalainen. Muutoin transformerin yhteenvedon sisältö on hieman seikkaperäisempi kuin Wikipedian oma yhteenveto, joka johtuu luultavasti asetetusta suurimman sanamäärän (450) arvosta yhteenvedolle.

Korkeimmat ROUGE-1 F1-pisteet saaneen transformerin edellä esitetyn yhteenvedon kaikki ROUGE F1-pisteet ovat esitetty taulukossa 5. Bhandari ym. 2020 mukaan ROUGE-2-arvot korreloivat parhaiten abstrahoitujen yhteenvetojen kanssa, kun taas ROUGE-1-arvot korreloivat parhaiten ekstraktiivisten yhteenvetomenetelmien kanssa. Tätä pro gradu -tutkimusta

5. https://en.wikipedia.org/wiki/Anniken_Huitfeldt Viitattu marraskuussa 2022.

varten tehdyistä 1000 Wikipedia-sivun yhteenvedoista `philschmid`-transformerin parhaat ROUGE-2 F1-pisteet saaneen yhteenvedon kaikki ROUGE F1-pisteet ovat esitetty taulukossa 6.

Philschmid: paras ROUGE-1	
	F1-pisteet
ROUGE-1	0.6
ROUGE-2	0.4186
ROUGE-L	0.6

Taulukko 5. 1000 Wikipedia-sivun keskimäärin parhaimmat ROUGE-1 F1-pisteet saaneen transformerin kaikki ROUGE F1-pisteet.

Philschmid: paras ROUGE-2	
	F1-pisteet
ROUGE-1	0.5185
ROUGE-2	0.449
ROUGE-L	0.5185

Taulukko 6. 1000 Wikipedia-sivun keskimäärin parhaimmat ROUGE-2 F1-pisteet saaneen transformerin kaikki ROUGE F1-pisteet.

Kyseinen yhteenveto oli seuraava:

Joseph S.B. Mitchell is Distinguished Professor of Applied Mathematics and Statistics and Research Professor of Computer Science at Stony Brook University. He is an editor-in-chief of the International Journal of Computational Geometry and Applications and co-chair of the PC for the 21st ACM Symposium on ComputationalGeometry in 2005. He shared the 2010 Gödel Prize with Sanjeev Arora for devising a polynomial-time approximation scheme for the Euclidean travelling salesman problem.

Wikipedian oma yhteenveto[6] oli seuraava:

Joseph S. B. Mitchell is an American computer scientist and mathematician. He is Distinguished Professor and Department Chair of Applied Mathematics and Statistics and Research

6. https://en.wikipedia.org/wiki/Joseph_S._B._Mitchell Viitattu marraskuussa 2022.

Professor of Computer Science at Stony Brook University.

Poiketen tyypillisestä Wikipedian henkilöä koskevasta yhteenvedosta, oheisessa yhteenvedossa ei mainita syntymäaikaa. Transformerin tekemä yhteenveto on jälleen hieman seikkaperäisempi johtuen luultavasti asetetusta sanarajasta. Kuitenkaan transformer ei ole generoinut koko sanarajan mittaista yhteenvetoa.

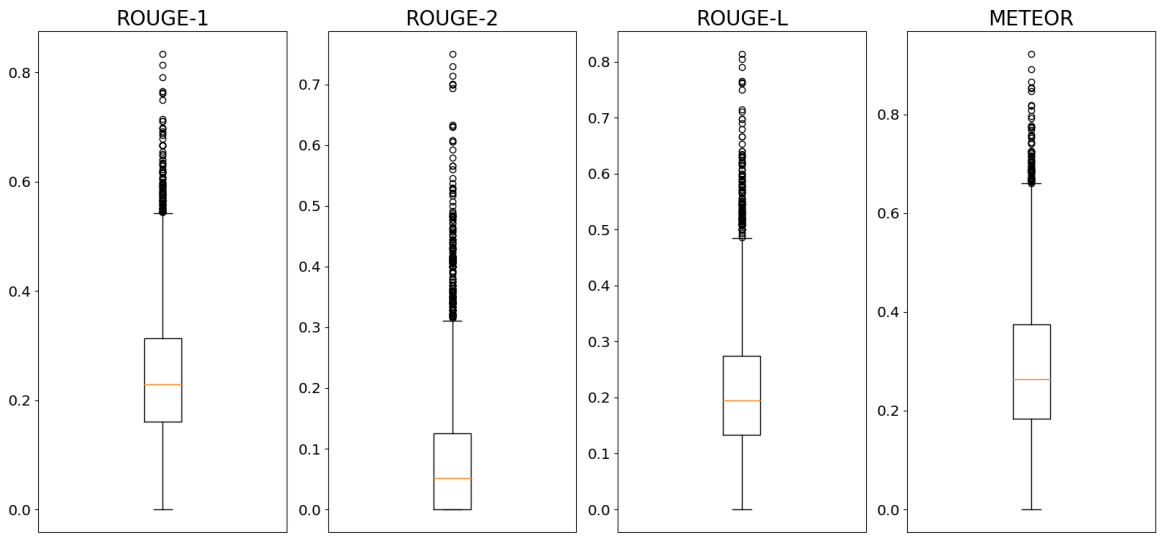
Verratessa muutoin näitä yhteenvetoja huomataan, että transformerin ensimmäinen virke kattaa melko hyvin koko Wikipedian yhteenvedon. Seuraavat transformerin virkkeet käsittävät asiaa, jota Wikipedian yhteenvedossa ei mainita. Välilyönnin puuttuminen sanojen ”Computational” ja ”Geometry” välissä keskimäärin hyvin suoriutuneella transformerilla generoimassaan yhteenvedossa on virhe.

Tarkasteltaessa noin 0.2:n ROUGE-1 F1-pisteet saaneita Wikipedia-sivujen yhteenvetoja, havaittiin, että Wikipedian oma yhteenveto käsitti erilaisia yksityiskohtia kyseisen henkilön henkilökohtaisesta elämästä. Kuitenkin sektiot, jotka käsittelivät kyseistä henkilöä, saattoivat käsitellä esimerkiksi vain henkilön tutkimustyötä eikä syntymäaikaa tai muutoin henkilökohtaista elämää. Loput sektioista saattoivat käsittää lähinnä ulkoisia linkkejä.

6.2.2 CiteSum

CiteSum-opetusdatasetin testijoukon osajoukon 3600 ensimmäistä lähdetekstiä malliyhteenvetoineen osoittautuivat haasteelliseksi *n*-grammeihin perustuvien metriikoiden tulosten perusteella. Haasteellisuudesta huolimatta `philschmid`-transformerille kuviosta 7 voidaan nähdä korkeitakin ROUGE F1-pisteitä. ROUGE-1 F1-pisteistä vähintään 0.5 saavuttaneita yhteenvetoja saatiin 150 (noin 4.17%). Vastaavasti ROUGE-2 F1-pisteiden osalta saatiin 27 (0.75%) ja ROUGE-L F1-psiteiden osalta 121 (3.36%) yhteenvetoa.

Parhaiksi ROUGE-1 F1-pisteet saatiin 0.8333. Sen osalta saatiin seuraava yhteenveto: *”Taking the Artwork Home” is a mobile app that allows people to digitally curate their own augmented reality art exhibitions in their own homes by digitally replacing the pictures they have on their walls with content from the Peter Scott Gallery in Lancaster.* Vastaava malliyhteenveto oli seuraava: *In REF , a work is presented that allows people to digitally curate their own augmented reality art exhibitions in their own homes by augmenting the pictures they*



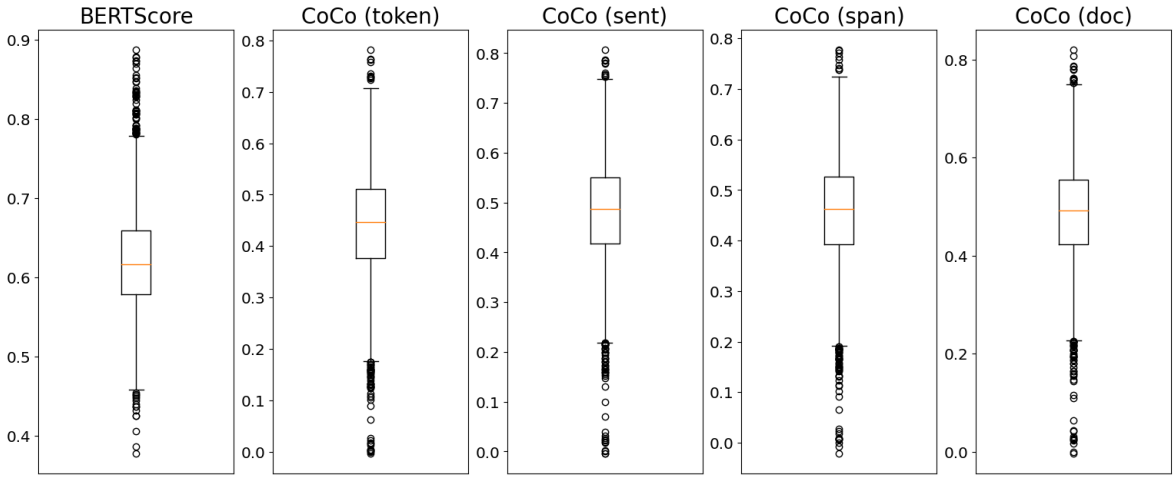
Kuvio 7. CiteSum-datasetin testijoukon osajoukosta tehtyjen yhteenvetojen ROUGE F1-pisteiden ja METEOR-pisteiden jakauma.

have on their walls with content from the Peter Scott Gallery in Lancaster. Malliyhteenveto on hieman tiiviimpi ja ilmaistu enemmän lähdetekstin sanaston ulkopuolelta.

ROUGE-2 F1-pisteiksi 0.0 saatiin 1048:lle (noin 28%) yhteenvedolle. ROUGE-1 F1-pisteet ja ROUGE-L F1-pisteet 0.0 saaneita yhteenvetoja saatiin 23 (noin 0.64%). Kyseessä ovat samat yhteenvedot. Osa yhteenvedoista käsittää keskenjääneitä virkkeitä. Heikoin generoitu yhteenveto koostuu vain yhdestä merkistä ”A”. Epäonnistunut yhteenveto selittyy ehkä osin lähdetekstin lyhyydellä. Siinä oli vain 17 sanaa, mistä 30% on noin viisi sanaa. Muutoinkin epäonnistuneet yhteenvedot käsittivät useita keskeneräiseksi jääneitä virkkeitä. Esimerkiksi eräs yhteenveto koostui vain ilmaisusta ”High-through”. Kuitenkin joukossa oli myös ehjiä virkkeitä. METEOR-pisteiksi samoille 23:lle yhteenvedoille saatiin keskimäärin 0.0642 ja mediaaniksi 0.05814. Näille parhaimmillaan METEOR-pisteiksi saatiin 0.2222 ja huonoimmillaan saatiin neljälle yhteenvedolle arvo 0.0. METEOR-pisteet ovat siis myös näille yhteenvedoille keskimäärin lähellä nollaa. METEOR-pisteiden 0.2222 osalta generoitu yhteenveto oli: *Optimal Transport GAN (OT-GAN) is a variant of generative adversarial nets. It minimizes the distance.* Viimeinen virke ei päättynyt pisteeseen, joten se voi olla virke voi olla keskeneräinen. Malliyhteenveto oli kummallisenkin tiivis: *OT-GAN REF* . METEOR-metriikka on selvästi kuitenkin löytänyt olennaisen asiasisällön ”OT-GAN”.

METEOR-metriikan osalta puolestaan parhaiksi pisteiksi saatiin 0.9219. Kyseinen yhteenveto oli seuraava: *The identification of mathematical models describing the behaviour of wave energy devices (WECs) in the ocean is investigated through the use of numerical wave tank experiments. Nonlinear hydrodynamic effects may appear in the*. Generoidussa yhteenvedossa toinen virke on jäänyt kesken. Yhteenvedon ROUGE-1 F1-pisteet ovat 0.766. Malliyhteenveto oli: *Identification of the mathematical models describing the behavior of wave energy devices (WECs) in the ocean is investigated through the use of numerical wave tank experiments REF* . Generoitu yhteenveto sisältää malliyhteenvedon ydinajatuksen. Siltä osin se on laadukas.

Puolet METEOR-pisteistä sijoittuivat välille 0.1822...0.3738. Tulos on keskimääräisesti hieman parempi kuin millään ROUGE-menetelmällä. Koska *n*-grammeihin perustuvilla metriikoilla yhteenvetojen laatupisteet vaikuttavat keskimäärin kuitenkin heikoilta, on ollut syytä arvioida laatua myös uudemmilla metriikoilla. Kuviossa 8 ovat esitettynä BERTScoren F1-pisteiden ja CoCo-pisteiden jakaumat.



Kuvio 8. CiteSum-datasetin testijoukosta tehtyjen yhteenvetojen BERTScoren F1-pisteiden ja CoCo-pisteiden jakauma.

Jakaumista nähdään, että BERTScoren F1-pisteiden osalta saatiin jokseenkin hyvä tulos. 25% arvoista on mediaanin 0.6163 yläpuolella rajana 0.6588 ja 25% arvoista on mediaanin alapuolella rajana 0.5785.

METEOR-metriikan parasta pistemäärää vastaavalle yhteenvedolle saatiin BERTScoren F1-

pisteiksi 0.8777. Samalle yhteenvedolle parhaimmat CoCo-pisteet saatiin virke- ja dokumenttitason peittämismekanismeilla samalla pistemäärällä 0.4868.

Parhaat saadut BERTScoren F1-pisteet olivat 0.8871. Tätä vastaavat ROUGE-1, ROUGE-2 ja ROUGE-L F1-pisteet olivat järjestyksessä 0.7907, 0.7143 ja 0.7907. Vastaavat CoCo-pisteet olivat korkeimmillaan 0.7857, jotka saatiin virke- ja dokumenttitason peittämismekanismeilla. Saatu yhteenveto oli: *Outlier detection is an important research problem in data mining that aims to discover useful abnormal and irregular patterns*. Malliyhteenveto puolestaan on: *REF proclaimed outlier detection is an important research problem in data mining that attempts to discover useful abnormal and irregular patterns hidden in large datasets.* Generoitu yhteenveto ei ole huono. Se vaikuttaa jopa tiiviimmältä kuin malliyhteenveto. Koska generoitu yhteenveto itse ei pääty pisteeseen, se voi kuitenkin olla osin kesken jäänyt yhteenveto, jonka suurin sanamäärä (lähdetekstistä 30%) on pakottanut lyhyeksi.

BERTScoren heikoimpia F1-pisteitä (0.3778) vastaava yhteenveto on jokseenkin pitkä ja se mukailee melko orjallisesti lähdetekstiä. Kuitenkin on outoa, että subjektiivisesti ottaen laadullisesti heikoin yhteenveto käsittää vain kirjaimen ”A” ja sille saatiin BERTScoreksi korkeammat F1-pisteet, 0.3864. Samalle yhteenvedolle CoCo:n sanavälimaskilla (engl. span) saatiin negatiiviset pisteet -0.0211. Muilla CoCo:n peittämismekanismeilla on tälle yhteenvedolle saatu hyvin lähellä nollaa oleva positiivinen pisteytys. Suurimmillaan se oli virke- ja dokumenttitason peittämismekanismeilla 0.0268.

Muista yhteenvedoista itseisarvoltaan suurin negatiivinen pisteytys CoCo-metriikalla saatiin sanavälinpeittämismekanismilla. Pisteet olivat -0.0072. Yhteenvedolle saatiin koko dokumenttitason peittämismekanismilla suurin CoCo-pisteytys arvolla 0.0227. Generoitu yhteenveto oli: ”*A paper proposes a Privacy Butler to monitor a person’s online presence and attempt to make corrections based on policies specified by the owner of the online presence.* Malliyhteenveto oli: *Jannasch et al. have developed a system of statically moored ocean sensors to monitor environmental processes off the coast of California in Monterey Bay REF* . Subjektiivisella tarkastelulla yhteenvedon lähdetekstiin nähden `philschmid`-transformer on tulkinnut väärin. Siksi CoCo-metriikan antamaa heikkoa pisteytystä voi pitää oikeutettuna. BERTScore yhteenvedolle on 0.5929, jota ei voi pitää oikeutetun suuruisena pistemääränä transformerin väärin tulkitsemalle lähdetekstille.

Koska CoCo-metriikka toimintalogiikkansa myötä löytää lähdetekstin faktat yhteenvedoista paremmin kuin BERTScore ja siten tulkitsee lähdetekstin hyvin, voidaan yhteenvedoista arvioida keskimääräisesti hyvät CoCo-pisteet saaneet yhteenvedot myös transformerin osalta hyvin generoiduiksi. Kuitenkin manuaalisella tarkastelulla havaittiin yhteenvedoissa myös keskenjääneitä virkkeitä.

Virkkeenpeittämismekanismilla CoCo-pisteet välille 0.45...0.52 saatujen yhteenvetojen keskimääräiset BERTScoren F1-pisteet olivat 0.6215 ja vastaavasti METEOR ja ROUGE-1 F1-pisteet järjestyksessä 0.2971 ja 0.2474. Vastaavasti CoCo:n sanavälinpeittämismekanismilla saatiin samalle arvovälille näillä metriikoilla järjestyksessä keskiarvoisesti pisteet 0.6219, 0.3026 ja 0.2474.

Arvovälille generoiduista yhteenvedoista nähdään, että niissä on myös lähdetekstin ulkopuolista sanastoa. Edellä käytetylle arvovälille generoitiin esimerkiksi seuraava yhteenveto: *In the study, we consider the dynamics of a linear stochastic approximation algorithm driven by Markovian noise. We also solve the open problem of obtaining finite-time bounds for the performance of temporal difference learning algorithms with linear function approximation and a constant step-size.* Sana ”study” ei esiinny yhteenvetoon liittyvässä lähdetekstissä lainkaan.

6.3 Virkkeiden tiivistäminen

Virkkeiden tiivistämistä arvioitiin käyttämällä luvussa 2.5 kuvailtua Gholipour Ghalandari, Hokamp ja Ifrim 2022 luomaa ekstraktoivaa virkkeentiivistäjää. Datasettinä käytettiin `Huggingface.co`-sivustolta löydettyä opetusdatasettiä[7], jossa kullekin virkkeelle on mallitiivistelmä. Datasetti käsittää kaikkiaan 180 000 virkettä. Virkkeentiivistäjän sekvenssinpituus on 512. Siksi tuloksia laskettaessa suurimmaksi sanamääräksi datasetin virkkeelle asetettiin 512 ajonaikaisten virheiden välttämiseksi. Koejärjestelyllä testattavia virkkeitä kertyi näin 179 990, joka antaa hyvän kokonaiskäsityksen virkkeen tiivistäjän suorituskyvystä sen eri malleilla.

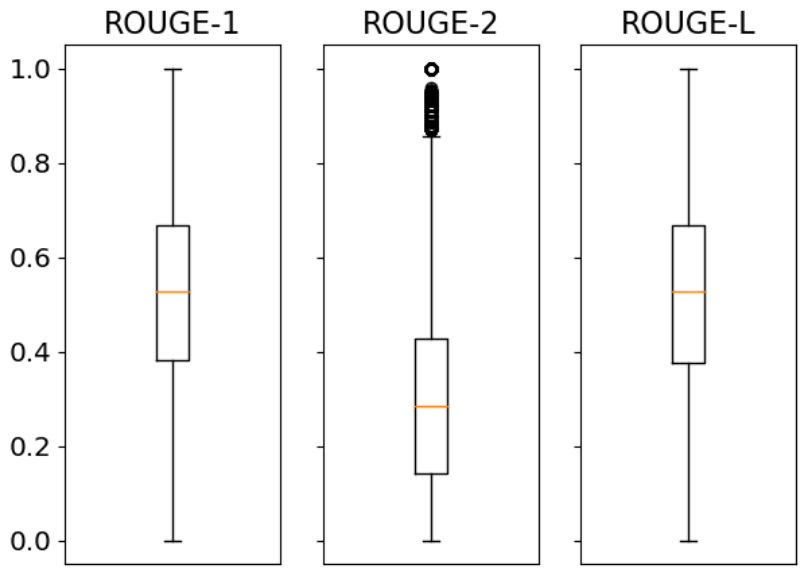
Käytetty virkkeentiivistäjäkokonaisuus sisälsi kolme eri mallia. Malleilla saatujen ROUGE F1-pisteiden keskiarvot ovat esitetty taulukossa 7.

7. https://huggingface.co/datasets/embedding-data/sentence-compression

Virkkeentiivistäjän ROUGE F1-pisteiden keskiarvot			
	newsroom-L11	**newsroom-P75**	**gigaword-L8**
ROUGE-1	0.5143	0.4206	**0.5209**
ROUGE-2	**0.2942**	0.2343	0.2866
ROUGE-L	0.5101	0.4157	**0.5175**

Taulukko 7. Virkkeentiivistäjän mallien ROUGE F1-pisteiden keskiarvot.

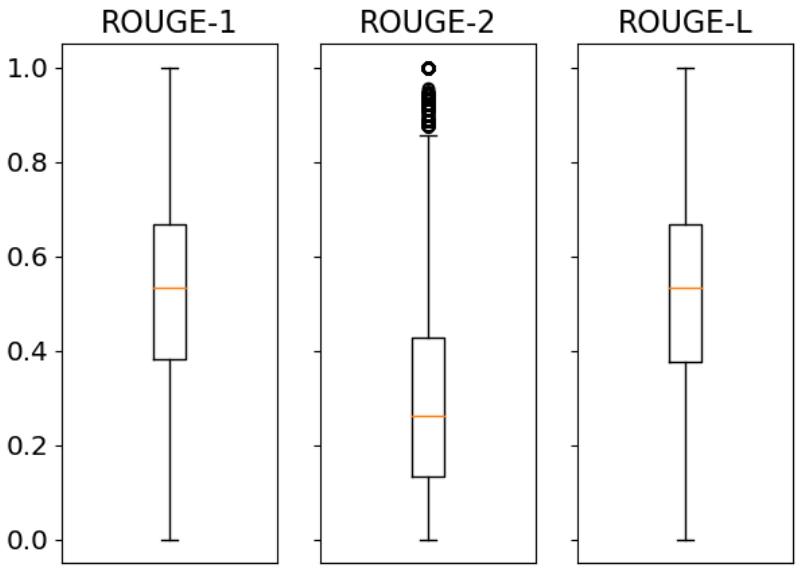
Kuviossa 9 näkyy mallille **newsroom-L11** ROUGE F1-pisteiden jakauma. Se on saanut parhaimmat ROUGE-2 F1-pisteet vertailussa keskimäärin. Mallin ROUGE-2 F1-pisteiden mediaani on 0.2857, joka on hieman keskiarvoa alhaisempi, mistä voidaan päätellä, että mediaanista nähden on suuria poikkeavia arvoja ainakin yksi.



Kuvio 9. Mallin **newsroom-L11** ROUGE-arvot laatikkokuviona.

Kuviossa 9 näkyy pieni rypäs lähellä 1.0:aa olevia ROUGE-2:n F1-pisteitä. Näitä tuloksia tarkastellessa lähemmin käyttämällä vertailuarvoa 1.0, saadaan 374 tulosta. Se on noin 0.2% kaikista tuloksista. Tiivistelmistä 29 on jokaista merkkiä myöten *whitespace*-merkkejä huomioimatta samat mallitiivistelmien kanssa. Samalla tarkkuudella vain hieman alemmat ROUGE F1-pisteet saivat myös ne yhteenvedot, jotka erosivat toisistaan vain pisteellä virkkeen lopussa mukaanlukien *whitespace*-vaihtelu.

Kuviossa 10 on esitettynä keskimäärin parhaan mallin **gigaword-L8** laatikkokuvio ROUGE F1-pisteistä. Arvot ovat suhteellisen lähellä toiseksi parhaan mallin **newsroom-L11** ROUGE F1-pisteitä.



Kuvio 10. Mallin **gigaword-L8** ROUGE-arvot laatikkokuviona.

Mallin **gigaword-L8** ROUGE-L F1-pisteistä on 0.68% (1229) 1.0. Näistä 1090 erosi vain *whitespace*-ominaisuuksien osalta mallitiivistelmästä. Mediaaniksi malli sai ROUGE-L F1-pisteiksi 0.5333. Mediaanin paikkeilla olevista ROUGE-L F1-pisteistä eräs tiivistelmäesimerkki oli *Teen REACH Vermilion County invites youth 8-17 to join after-school program*. Mallitiivistelmä tälle puolestaan oli *Teen REACH invites youth*. Mediaanin paikkeilla ROUGE-L F1-pisteiden osalta tuotetut tiivistykset olivatkin hieman monisanaisempia kuin malliverrokkinsa.

Virkkeentiivistäjän mallien tulosten laatikkokuvioiden sekä lähemmän tarkastelun perusteella kaikki kolme mallia saivat kaikista ROUGE F1-pisteistä muutamalle tiivistelmälle arvon 0.0. Se selittyy siten, että osa mallitiivistelmistä on tehty selvästi abstrahoiville virkkeentiivistäjille tiivistettävän virkkeen sanaston ulkopuolelta. Toisin sanoen tiivistelmä oli ilmaistu ”omin sanoin”. Esimerkiksi malli **newsroom-P75** sai 642 virkkeen osalta ROUGE-1 F1-pisteiksi 0.0. Se on vain noin 0.36% kaikkien testattujen virkkeiden määrästä. Lähempi tarkastelu näyttää, että mallitiivistelmäksi on selvästi haettu osissa datasettiä abstrahoivan mal-

lin tulosta. Eräs ROUGE-1 F1-pisteet 0.0 saanut tiivistelmä itsessään oli *SEVERE depression can shrink the brain by blocking nerve connections, a study shown*. Mallitiivistelmä puolestaan oli *Depression 'shrinks brain'*. Näitä kahta siis verrattiin keskenään.

Kaikkiaan Gholipour Ghalandari, Hokamp ja Ifrim 2022 ekstraktoiva virkkeentiivistäjä on selvinnyt hyvin tehtävästään. Kaikkien mallien keskiarvo oli ROUGE-1 ja ROUGE-L F1-pisteiden osalta hieman yli 0.5. ROUGE-2 F1-pisteet jäivät kaikilla malleilla alle 0.3:n, mutta kuitenkin ne olivat yli 0.2.

6.4 Aihemallinnus LDA:lla

Aihemallinnuksen korpuksena käytettiin Alan Turingin Wikipedia-sivun[8]. suhteellisen pitkää yhteenvetoa, joka käsittää useita eri asioita. Motivaationa mallinnettavalle tekstivalinnalle on nähdä, kuinka tutkimustiedon valossa tulosteina mahdollisesti liian pitkät yhteenvedot voi antaa lyhyemmin osissa tulosteiksi chatbotille.

Aihemallinnusmenetelmänä on käytetty LDA:ta. Aiheiden määrä on määritetty automaattisesti käyttäen kahta eri metriikkaa. Aihemäärämetriikoiden käyttämisen mahdollisti Pythonin `tmtoolkit`-kirjasto. Käytännön aihemallinnuksen generoimien aihepiirikokonaisuuksien mielekkyyttä voi arvioida vain subjektiivisesti. Siksi niiden kuvailu on tehty lyhyesti.

Korpuksen aihemallinnuksessa mallinnuksen dokumenttiyksikkö käsittää kaksi perättäistä virkettä korpuksesta, jotta saataisiin yhtenäisempi konteksti aiheisiin. Virkkeitä korpuksessa on 25. Siksi suurimmaksi aiheiden määräksi metriikoille on asetettu kokonaisluvuksi katkaistuna 25 / 2 + 1 = 13. Koska korpuksessa on pariton määrä virkkeitä, niin myös yksi aiheista käsittää parittoman määrän virkkeitä. Muutoin LDA niputtaa korpuksen virkkeet käyttäen vähintään yhtä dokumenttikooksi määriteltyä määrä virkkeitä.

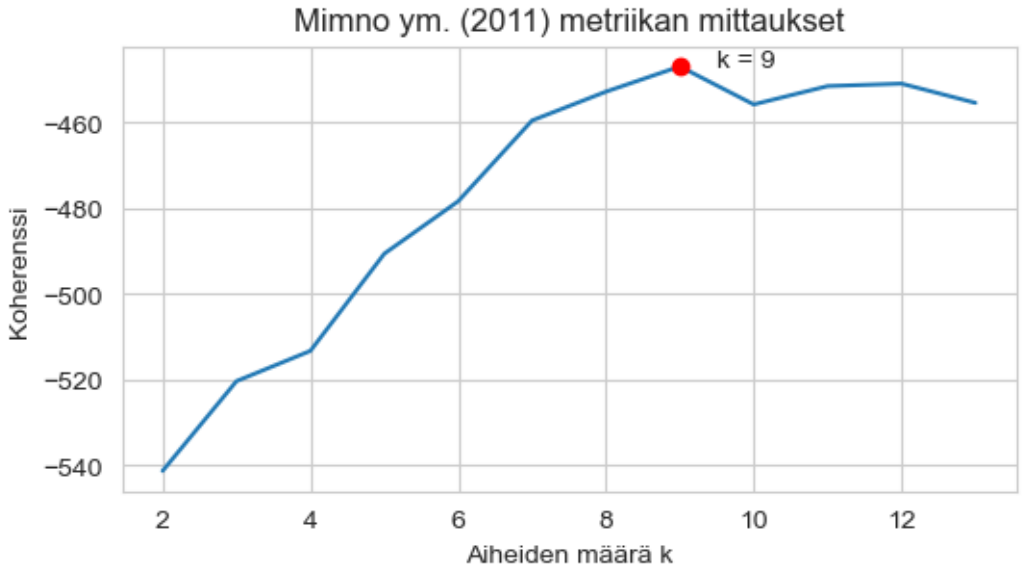
Korpuksen sanastosta on aihemallinnukseen otettu mukaan Pythonin `spaCy`-kirjaston näkökulmasta erisnimet, substantiivit, organisaatioentiteetit, geopoliittiset entiteetit ja persoonat. Näin sanaston kooksi saatiin 147 uniikkia sanaa. Tämä käsittää myös esikäsittelyssä välttämättömän lähdetekstin ”GC&CS”-ilmaisun erottamisen välilyönneillä muotoon ”GC & CS”

8. https://en.wikipedia.org/wiki/Alan_Turing Viitattu 11.3.2023.

ja näin saadut kaksi eri sanaa, ”GC” ja ”CS”. Koska Pythonin `tmtoolkit`-kirjaston evaluointimetodin tuottama LDA-olio itsessään tekee tämän erottelun, niin käsittely oli tehtävä, koska aihesanoja tulkitessa toteutuksessa lemmatuista sanoista haetaan yksittäisen sanan alkuperäinen korpuksen muoto.

Esikäsittelyn jälkeen korpuksen kokonaissanamäärä oli 553 sanaa. Näistä uniikkeja sanoja oli 304, kun kaikki sanat oli muutettu kokonaan pienaakkosiksi. Välimerkkejä ei laskettu sanojen joukkoon.

Mimno ym. 2011 ja Cao ym. 2009 metriikoilla saatiin järjestyksessä 9 ja 5 aihetta. Metriikoiden mittaukset ovat esitetty järjestyksessä kuvioissa 11 ja 12. Saatujen aiheiden aihesanat ovat esitetty vastaavasti taulukoissa 8 ja 9. Aihesanoja on käytetty seitsemän.



Kuvio 11. Alan Turingin Wikipedia-sivun yhteenvedolle aihemallinnuksen aiheiden määrä Mimno ym. 2011 metriikalla.

Mimno ym. 2011 metriikalla saadut aiheet käsittävät keskimäärin vain hieman vähemmän kahden virkkeen dokumenttiyksiköitä, kuin aiheita voisi lähdeaineistosta koejärjestelyllä saavuttaa. Metriikalla on löydetty siis paljon piileviä aiheita korpuksesta. Cao ym. 2009 metriikan aiheet puolestaan ovat selvästi laajempia ja aiheiden määrä on vähäisempi.

Kaikkiaan Mimno ym. 2011 metriikalla mallinnuksen jokainen aihe on johdonmukaisuudel-

Kuvio 12. Alan Turingin Wikipedia-sivun yhteenvedolle aihemallinnuksen aiheiden määrä Cao ym. 2009 metriikalla.

taan ja kontekstiltaan jokseenkin ehjä. Generoidut aiheet ovat esitettynä liitteessä A. Cao ym. 2009 metriikalla puolestaan ei saatu samassa määrin mielekkäitä aihekokonaisuuksia. Kuitenkin saaduista aiheista aiheet 3, 4 ja 5 ovat johdonmukaisia kokonaisuuksia, joissa konteksti ei sanottavasti rikkoudu. Aihe 2 käsittää kolme virkettä, joista kaksi ensimmäistä ovat lähdetekstissä perättäiset. Aiheen viimeinen virke olisi sopinut paremmin esimerkiksi aiheen 3 loppuun. Aihe 1 on hieman sekava, koska asiat ovat liian irrallaan toisistaan. Toisaalta voidaan olettaa, että laajan yhteenvedon sisältö itsessään voi olla epäyhtenäinen. Aiheet ovat esitettynä liitteessä B.

Molempien metriikoiden määrittämien aiheiden osalta on kuitenkin ongelmallista, jos aihe alkaa viittausilmaisulla kuten pronominilla. Vaikka tiedetään, että kyseessä on Alan Turingin Wikipedia-sivun yhteenveto, ei ole välttämättä selvää kehen aiheen alussa oleva pronomini viittaa.

Suurimmillaan Mimno ym. 2011 metriikan aihejakaumalla mallinnettaessa aiheen sanamääräksi tuli 125, joka on noin 22.6% korpuksesta. Kyseessä on aihe 7. Siinä kuvaillaan Turingin elämää ja sen jälkeistä aikaa hänen saavutuksiensa valossa. Lyhyimmässä Mimno ym. 2011 metriikalla saadulla aiheessa sanamäärä on 33, joka on noin 5.97% korpuksesta. Kyseessä

Mimno ym. 2011 metriikkaan perustuva aihemallinnus LDA:lla	
	Aihesanat
Aihe 1	minister, PhD, Department, campaign, Turing, poisoning, Government
Aihe 2	science, legacy, alternative, University, hormone, June, birthday
Aihe 3	biology, School, Cypher, Laboratory, Manchester, Computing, reactions
Aihe 4	questions, Mathison, role, computation, Cambridge, Machine, Turing
Aihe 5	GC, settings, men, engagements, Kingdom, law, Turing
Aihe 6	series, section, program, scientist, Atlantic, century, June
Aihe 7	Bank, Maida, science, England, Machine, Turing, computer
Aihe 8	Britain, suicide, Mathematics, Bletchley, DES, centre, Newman
Aihe 9	Britain, Official, pardon, Physical, lifetime, accomplishments, Act

Taulukko 8. Mimno ym. 2011 metriikkaa käyttäen LDA:lla saadut aihesanat Alan Turingin Wikipedia-sivun yhteenvedolle.

on aihe 9, joka käsittää kahden virkkeen verran Turingin elämän tragiikkaa.

Cao ym. 2009 metriikalla aihejakaumalla suurimmillaan aiheen sanamääräksi tuli 155, joka on noin 28% korpuksesta. Kyseessä on kelpo aihe 5. Aiheessa on asiaa Turingin tiedustelupalvelun aikaisesta työstä ja sen jälkeisestä tieteellisestä urasta. Pienimmillään sanamäärä metriikalla oli 52, joka on vain noin 9.4% korpuksesta. Aihe käsittää kolme virkettä, joista kaksi ensimmäistä käsittelee Turingin elämän loppua ja viimeinen virke hänen vuonna 2019 saamaansa tunnustusta merkittävimmäksi ihmiseksi 1900-luvulla.

Aihemallinnusta kokeiltiin myös laajemmalle korpukselle, joka kattaa historiallisesti tunnetusti työnsä puolesta toisiinsa liittyvien henkilöiden Arthur Eddingtonin, Albert Einsteinin ja Max Planckin Wikipedia-sivut. Mallinnuksessa tehtiin vastaava sanakarsinta kuin Alan Turing yhteenvedon osalta. Ohjelmoinnin aikana havaittiin, että korpuksen luonteesta ja koosta johtuen voi olla mielekästä käyttää mallinnuksessa dokumenttikokona esimerkiksi yhtä kappaletta liittäen sen perään vielä mahdollinen korpuksen sisäinen sitaattikappale tai kappaleet yhtenäisemmän kontekstin saavuttamiseksi.

Näin saatiin laajempaan korpukseen 226 dokumenttiyksikköä, joista Mimno ym. 2011 met-

Cao ym. 2009 metriikkaan perustuva aihemallinnus LDA:lla	
	Aihesanat
Aihe 1	prime, Princeton, prison, procedure, computation, poisoning, Government
Aihe 2	men, Maida, Brown, birthday, Alan, science, June
Aihe 3	London, England, Machine, reactions, Laboratory, computer, Turing
Aihe 4	Kingdom, Machine, Cambridge, Britain, law, Act, Turing
Aihe 5	designs, Elizabeth, Battle, Zhabotinsky, Machine, Computing, Manchester

Taulukko 9. Cao ym. 2009 metriikkaa käyttäen LDA:lla saadut aihesanat Alan Turingin Wikipedia-sivun yhteenvedolle.

riikalla saatiin 17 ja Cao ym. 2009 metriikalla 28 aihetta. Kukin aihe käsitti useita kappaleita. Kokonaisuudeksi saatiin hieman sekava aihekokonaisuus, jossa kuitenkin oli myös lähdedokumenteissa perättäisiä kappaleita yksittäisessä aiheessa perättäin.

7 Pohdinta

Luvussa pohdiskellaan tutkimuskysymyksiä ja saatuja tuloksia. Osaan tutkimuskysymyksistä voidaan johtaa vastaus esitetystä teoriataustasta. Kolmanteen tutkimuskysymykseen johdetaan vastaus teoriataustasta ja tulosten pohjalta.

7.1 Mielekkään pituinen chatbotin tuloste

Aikaisempiin tutkimuksiin pohjaten voidaan sosiaaliselle chatbotille mielekkään pituinen tuloste nähdä minimoimisen periaatetta noudattavana. Tällöin chatbotin harjoittama keskustelu mukailee ihmisten välisen vuorovaikutustutkimuksen mukaan ihmisten keskenään harjoittamaa vuoropuhelua. Vuonna 2011 Turingin testin läpäisseen Cleverbotin tapauksessa keskimääräinen sanamäärä Cleverbotilla oli vain noin 4.29 sanaa viestiä kohden. Sosiaalisen chatbotin tulosteen tulisi siis olla mahdollisimman lyhyt, eikä sen tarvitse välttämättä olla formaalia kieltä.

Tietoa jakavien chatbottien osalta olennaista tietoa on hankala tiivistää vain muutamaan sanaan, ellei kyseessä ole esimerkiksi syntymäaika ja/tai -paikka. Tietoa jakavien chatbottien ilmaisun pituuden suhteen onkin mielekästä noudattaa Nielsenin heuristiikkaa: Dialogissa ei tule olla ylimääräisiä tekstiyksiköitä. Ihmisen näkökulmasta ilmaistaan vain välttämätön ja riittävä määrä tietoa käyttäjän pyyntöön.

Lisäksi on syytä välttää ihmisen kognitiivista kuormittamista liian rikkaalla muulla kuin tekstisisällöllä. Jos teksti hukkuu esimerkiksi chatbotin esittämien kuvien joukkoon, voi käyttäjän huomio keskittyä tekstin sijaan enemmän kuviin, jolloin chatbotin tarjoama tekstimuotoinen informaatio jää toissijaiseksi.

Myös esimerkiksi jonkinlaisissa asiakaspalvelutehtävissä toimivan chatbotin vastuulla on jakaa käyttäjän tarvitsema tieto. Olemassa olevan tutkimustiedon valossa chatbotin vastauksen on syytä olla informatiivinen ja ytimekäs. Liiketaloudellisesti ajatellen chatbotin on syytä jättää miellyttävä käyttäjäkokemus, jotta chatbottia ylipäätään käytettäisiin ja toisaalta tämä voisi houkutella ihmisiä palamaan yrityksen verkkosivuille.

Käyttäjäkokemus on tärkeä näkökulma siihen, että chatbottia käytetään ylipäätään. Olipa siis chatbotin käyttötarkoitus mikä hyvänsä, chatbotin on palveltava tarkoitustaan mahdollisimman hyvin. Siten ei ole mielekästä pitää liian orjallisesti kiinni tietyistä olemassa olevista ohjenuorista. Jos käyttäjä pyytää chatbotilta esseettä ja sen luominen kuuluu chatbotin käyttötarkoituskontekstiin, niin chatbotin tulee täyttää käyttäjän pyyntö ja tuottaa essee. Tuotettavan esseen itsessään pitäisi olla kuitenkin mahdollisimman tiivis ja ytimekäs olemassa olevan tutkimuksen valossa.

ChatGPT:n osalta tutkimusaineistossa tuli esiin, että tarkat vastaukset olivat myös tiiviitä. Hyvä chatbotin tuloste on siis tiedon osalta tarkka ja tiivis. Jos chatbotin käyttäjä haluaa lukea asiasta laajemmin, niin osa tutkimuksista tukee hyperlinkkien sisällyttämistä chatbotin tulosteeseen. Silti joidenkin tutkimusten mukaan osaa ihmisiä voi häiritä chatbotin jättäminen siirtymällä hyperlinkin kautta muualle.

7.2 Menetelmät tekstin lyhentämiseen tulosteen pituus huomioiden

Lähdetekstistä yhteenvedon tekeminen on eräs keino tuottaa chatbotille pitkistä lähdeteksteistä lyhyitä tulosteita, joissa säilytetään lähdetekstien ydinajatukset. Automaattinen yhteenveto voidaan tehdä kahdessa pääkategoriassa: ekstraktoituna tai abstrahoituna.

Tässä pro gradu -tutkielmassa esitellyt ekstraktiiviset menetelmät tekstistä yhteenvedon tekemiseen poikkeavat toimintalogiikkansa osalta olennaisesti toisistaan. Luhnin menetelmässä jätetään huomiotta yhteenvedon kannalta epäoleelliset sanat. Menetelmässä virkkeet jotka sisältävät eniten heuristisesti klusteroituneita sanoja, ovat yhteenvedon kannalta olennaisimmat. LSA:ssa puolestaan sanapussi- tai TF-IDF-matriisille tehdään singulaariarvohajotelma. Muodostuneista matriiseista yksi pitää sisällään tärkeimmät virkkeet vektoreina. TextRank-menetelmä puolestaan mukailee Googlen PageRank-algoritmia, jossa tekstissä olevat virkkeet ”suosittelevat” tärkeitä virkkeitä, jotka otetaan mukaan yhteenvetoon.

Ydinvirkkeitä poimivien ekstraktiivisten menetelmien heikkoutena kuitenkin on se, että niissä ei voi sanamäärän tarkkuudella asettaa yhteenvedon pituutta. Pituus asetetaan virkkeiden määrässä, ja yksittäinen virke voi ydinajatuksen lisäksi sisältää epäolennaista tietoa. Tutkimusaineiston perusteella hyvä yhteenveto on sanamäärältään 20–30% lähdetekstistä. Koea-

setelman CiteSum-datasetin testijoukossa oli lyhyempiäkin malliyhteenvetoja.

Sanamäärää asetettaessa yhteenvedolle on syytä analysoida lähdeaineisto luetteloiden ja mahdollisten ulkoisten linkkien osalta. Niiden sisältämät sanat ovat järkevää karsia pois sana- ja virkemäärästä tekstiaineistosta, josta yhteenveto tehdään. Ekstraktiiviset yhteenvetomenetelmät koeasetelmassa ottivat joskus mukaan lisäksi erilaisia luetteloita Wikipediasta. Koeasetelmassa käytetyt transformerit eivät sisällyttäneet Wikipedia-aineistosta luetteloita muutoin, paitsi jos koko Wikipedia-sivu pois lukien sen yhteenveto sisälsi vain niitä.

Ekstraktiivisia yhteenvetoja tuotettaessa voidaan käyttää yhä perinteisiä yhteenvetomenetelmiä, jos poimitaan vain ydinvirkkeitä. Abstrahoitujen yhteenvetojen osalta transformermallit ovat syrjäyttäneet perinteiset syväoppimismenetelmät. Transformer-mallit mahdollistavat laskennassa enemmän rinnakkaisuutta kuin perinteiset syväoppimismenetelmät. Self-attention-mekanismin kanssa transformer-mallit oppivat pisteyttämään sanoja luoden kontekstin sanoille. Näin voidaan virkekontekstin myötä tehdä ero esimerkiksi englanninkielisten ilmaisujen ”river bank” (suom. joen ranta) ja ”bank” (suom. pankki) osalta. Tekniikka auttaa myös uusimmissa aihemallinnuksen innovaatioissa tunnistamaan tekstin merkityksen paremmin.

Aihemallinnus on toinen keino lyhentää pitkää lähdetekstiä chatboteille sopivan mittaisiksi tulosteiksi. Joskus yhteenveto voi olla laaja käsittäen useita aihepiirejä, jotka voidaan onnistuneella aihemallinnuksella poimia erilleen. Esimerkiksi Wikipediassa oleva Alan Turingin yhteenveto olisi liian pitkä sellaisenaan vaikka Twitter-viestinä jaettavaksi, mutta yhteenvedosta mallinnettu yksittäinen aihe voi olla sopivan mittainen Twitteriin. Koska koeasetelmassa aihemallinnus käsitti kaikki yhteenvedon virkkeet, voitaisiin yhteenveto periaatteessa chatbotin toimesta lähettää aiheittain Twitteriin.

Korpuksista johdettujen aihesanojen määrä ja tyyppi vaikuttaa siihen, saadaanko niiden perusteella edes karkeaa käsitystä siitä, mitä yksittäinen aihe käsittää. Jos käyttäjällä on pohjatietoa mallinnetusta aiheesta, saattavat aihesanat valaista yksittäisen aiheen sisältöä. Laajojen korpusten osalta perinteisillä aihemallinnusmenetelmillä, kuten LDA, voidaan saada suhteellisen mielekäs aihekokonaisuus korpuksesta, joka käsittää erillisiä dokumenttejakin. Kuitenkin konteksti voi paikoin rikkoutua häiritsevästi. Koeasetelmassa käytetty laaja korpus

käsitti historiallisesti työnsä osalta jotenkin toisiinsa liittyvien henkilöiden elämää ja tuloksena oli subjektiivisesti ottaen korpus huomioiden jossain määrin johdonmukainen kuitenkin osin sekava aihekokonaisuus niputtamalla korpus aiheisiin käyttämällä dokumenttiyksikkönä kappaletta tai kahta perättäistä kappaletta.

7.3 Yhteenvedoiksi tiivistetyn tekstin laadunarviointi

Ihmisten tekemät malliyhteenvedot tekstikokonaisuuksille korreloivat olemassa olevien tutkimusten perusteella vaikeasti automaattisten yhteenvetojen kanssa. Kuitenkin abstrahoitu yhteenveto voi parhaimmillaan olla tulosten perusteella hyvinkin lähellä ihmisten ehdotusta sopivasta yhteenvedosta lähdetekstille tulosten perusteella.

Ekstraktoivilla yhteenvetomenetelmillä voi virkkeisiin helposti tulla ylimääräisiä tekstiyksiköitä, koska ne sisällyttävät kokonaisia virkkeitä sellaisenaan lähdetekstistä, eivätkä ne tulosten perusteella pärjänneet yhtä hyvin Wikipedia-aineistolla, kuin abstrahoidut yhteenvedot. Toisaalta Wikipedian tapauksessa lähdetekstistä ei oltu suodatettu mitään luetteloita pois, joiden sisältöä ekstraktiiviset menetelmät joskus sisällyttivät yhteenvetoon yksittäisinä virkkeinä.

Jos malliyhteenveto käsittää sellaisenaan lähdetekstin virkkeet vain mahdollisesti yksittäisiä sanoja niistä poistettuna, niin ekstraktiiviset yhteenvetomenetelmät sopivat hyvin lähdetekstistä yhteenvedon tekemiseen. Tällöin niiden arviointiin sopivat hyvin leksikaaliseen päällekkäisyyteen perustuvat *n*-grammeihin perustuvat automaattiset metriikat. Virkkeentiivistäjän tuloksista ilmenee huippuluokan tuloksia ekstraktiiviselle transformer-pohjaiselle virkkeentiivistäjälle ROUGE F1-pisteiden osalta tapauksissa, joissa mallitiivistelmä edustaa lähdetekstistä vain sanojen karsintaa. Virkkeentiivistäjän tuloksissa on nostettu esiin esimerkki myös ROUGE-1 F1-pisteet 0.0 saaneesta tuotetusta tiivistelmästä. Mallitiivistelmässä oli käytetty lähdetekstiin nähden ulkopuolista sanastoa.

CiteSum-datasetin malliyhteenvedot ovat ilmaistu tyypillisesti lähdetekstin sanaston ulkopuolelta tiiviisti. Koska abstrahoidun yhteenvedon idea on tiivistää lähdeteksti käyttämällä mahdollisesti myös sen ulkopuolista sanastoa, toisin sanoen ilmaista tiivistettävä teksti ”omin sanoin”, niin erilaisia semanttisesti jokseenkin yhteneviä tiivistelmiä voi olla useita.

Tulosten pohjalta havaittiin, että *n*-grammien päällekkäisyyteen perustuvilla automaattisilla arviointimenetelmillä saatiin keskimääräisesti CiteSum-datasetin osalta jokseenkin heikot tulokset.

CiteSum-datasetti onkin hyvin haastava, koska se on tarkoitettu tieteellisten artikkeleiden tiivistämisen opettamiseen sen omalla tyylillä. Siksi se ei välttämättä ole paras mahdollinen datasetti myöskään `philschmid`-transformerilla generoitujen yhteenvetojen arviointiin, joka on hienosäädetty SAMSum-datasetillä ja CNN-uutisdatasetillä. Johtuen `philschmid`-transformerin hienosäädön luonteesta, eivät sillä generoidut yhteenvedot helposti voi saada vähintään 0.85:n pisteytystä CiteSum-datasetistä millään koeasetelmassa käytetyllä metriikalla.

Kontekstin ja sanojen semanttisen samankaltaisuuden huomioivalla BERTScorella saatiin CiteSum-datasetistä automaattisille yhteenvedoille keskimäärin olennaisesti paremmat tulokset kuin pääasiassa *n*-grammeihin perustuvilla menetelmillä. Kuitenkin BERTScoren F1-pisteet saattavat antaa myös harhaanjohtavan korkeat tulokset. Tuloksissa saatiin kolme esimerkkiä epäonnistuneista yhteenvedoista, joissa yhteenveto sisälsi parhaimmillaankin vain muutamia merkkejä. Nämä yhteenvedot vastasivat heikosti niiden malliyhteenvetoja. Silti niiden BERTScoren F1-pisteet olivat noin 0.38:n ja 0.45:n välillä.

CoCo-metriikka arvioi faktojen sisältymistä lähdetekstistä generoituun yhteenvetoon ja tulosten perusteella antaa realistisemman tuloksen yhteenvedon laadusta kuin BERTScore. Lisäksi tulosten perusteella CoCo-metriikka antaa todenmukaisemman pisteytyksen yhteenvedon laadusta kuin pääasiassa *n*-grammeihin perustuvat laadunarviointimenetelmät. Sen tuottamat pisteet olivat kaikilla peittämismekanismeilla myös keskimäärin olennaisesti korkeammat kuin pääasiassa *n*-grammeihin perustuvilla menetelmillä.

Epäonnistuneet yhteenvedot olisi voinut välttää tuloksien tarkastelun perusteella siten, että ei olisi käytetty suurimpana sanamääränä kiinteästi 30% lähdetekstin sanamäärästä. Osa CiteSum-datasetin lähdeteksteistä sisälsi alle 20 sanaa ja malliyhteenveto suurin piirtein saman verran sanoja. Jos siis yhteenvedot generoiva Python-skripti olisi ”lyhyiden” (noin 20 sanaa) lähdetekstien osalta käyttänyt suurimpana sanamääränä lähdetekstin sanamäärää, niin olisi voitu saada heikoimpien yhteenvetojen osalta paremmat yhteenvedot. Vastaavasti näille

olisi luultavasti saatu paremmat pisteet kaikilla metriikoilla. Automaattisten yhteenvetojen laadunarvioinnin tuloksiin vaikuttaa siis myös ohjelmointitason lähestymistapa datasettiin.

7.4 Tulosten heikot ROUGE F1-pisteet

Wikipedian osalta suhteellisen heikkoja keskimääräisiä ROUGE F1-pisteitä selittää ekstraktiivisten ja abstrahoitujen yhteenvetojen osalta se, että osa koeasetelmassa käytetyistä Wikipedia-sivuista sisälsi yhteenvedon lisäksi vain erilaisia luetteloita. Tällöin ei ollut yhteenvedon ulkopuolelta luetteloista sinällään mahdollistakaan luoda Wikipedian yhteenvetoa muistuttavaa yhteenvetoa. Jos yhteenvedon pituus ei olisi käytetyillä menetelmillä ollut vakio, vaan olisi ollut suhteessa Wikipedia-sivun asiatekstin sisältöön, niin yhteenvetojen laatupisteet olisivat olleet luultavasti korkeammat.

CiteSum-datasetin osalta puolestaan heikoimmissa ROUGE-1 F1-pisteet saaneissa generoiduissa yhteenvedoissa malliyhteenveto oli ilmaistu hyvin tiiviisti osin lähdetekstin sanaston ulkopuolelta. Siten *n*-grammien leksikaaliseen päällekkäisyyteen perustuva ROUGE ei voikaan antaa usein erityisen korkeita F1-pisteitä. Generoiduista yhteenvedoista noin 0.64% oli saanut ROUGE-1 ja ROUGE-L F1-pisteiksi 0.0. Kolmen yhteenvedon osalta heikon tuloksen selittää epäonnistunut yhteenveto. Muissa tapauksissa yhteenvedoissa kuitenkin oli asiasisältöäkin, mutta malliyhteenvetojen kanssa leksikaalista päällekkäisyyttä *n*-grammien osalta ei ollut. ROUGE-2 F1-pisteiksi 0.0 saaneita yhteenvetoja oli tuotetuista yhteenvedoista noin 28%. Bigrammien osalta päällekkäisyys malliyhteenvetojen kanssa oli siis jokseenkin heikko.

CiteSum-datasetin tarkoitus on opettaa tekemään tieteellisestä tekstistä erittäin lyhyitä yhteenvetoja. Tällöin 20–30% lähdetekstin sanamäärästä yhteenvedossa voi joskus olla liian paljon. Toisaalta havaittiin muutamia tapauksia, joissa ei olisi ollut malliyhteenvedon pituuteen nähden lainkaan mielekästä laskea sanamäärää. Lisäksi datasetti käyttää osassa lähdeteksteistä matemaattista symboliikkaa, jonka tulkitseminen transformerille ilman asianmukaista opetusta voi olla haastavaa.

7.5 Tekstin tiivistämisen hyödyntäminen chatboteissa

Lähdeaineistosta tulee esiin, että virkkeentiivistäjiä voidaan hyödyntää ihmisen chatbotille esittämän ilmaisun tulkitsemisessa. Jos ihminen on kirjoittanut monisanaisesti chatbotille, voi avainsanojen löytyminen hankaloitua. Virkkeentiivistäjät karsivat epäolennaiset sanat ja chatbot voi tulkita paremmin, mitä ihminen on tarkoittanut. Näin voidaan parantaa myös chatbotin käyttäjäkokemusta: sen ja ihmisen keskustelua ei tarvitse siirtää väliaikaisesti ihmistoimijan varaan. Se heikentäisi chatbotin käyttäjäkokemusta vastausviiveen muodossa.

Yhteenvetoja voidaan käyttää aiempien tutkimusten perusteella myös chatboteille koneellisen luetunymmärtämisen tukena. Lisäksi niiden keskustelutaito voi tutkimusten perusteella perustua osin aihemallinnukseen, jolla saadaan käsitys käydyn keskustelun relevanteista aiheista. Ekstraktoivalla yhteenvedolla voidaan luoda kysymyksiä käyttäjälle aihemallinnuksen aiheista keskustelun aikana. Eräs aiemmissa tutkimuksissa esiin tuleva sovellus yhteenvetojen hyödyntämiseen chatboteille ovat myös kysy-vastaa-tyyppiset järjestelmät sinällään.

Erilaisten verkon hakutulosten yhteydessä chatbot voi tehdä yhteenvedon haetusta aiheesta. Se auttaa käyttäjää saamaan nopeasti käsityksen siitä, onko chatbot löytänyt hänen kannaltaan relevanttia sisältöä. Yhteenvedon tulisi olla tiivis ja informatiivinen.

Lisäksi chatbottien yhteenvetoja voisi tutkimusaineiston perusteella ainakin hypoteettisesti hyödyntää sairaalaympäristössä potilastietojärjestelmässä. Chatbot voisi tehdä yhteenvedon lääkärin ja potilaan välisistä keskusteluista ja saataisiin nopeasti tarvittava kirjaus sairauskertomuksiin.

7.6 Jatkotutkimus

Jatkotutkimusaiheena esitetään hybridiyhteenvedon eli ekstraktiivisen ja abstrahoivan yhteenvedon yhdistämisen tutkimista. Hybridiyhteenvedolla on saatu ROUGE:n näkökulmasta laadullisesti lupaavia tuloksia pienille joukoille CiteSum-datasetistä tämän pro gradu -tutkimuksen aikana. Siten sillä on odotettavissa laadullisesti lupaavia tuloksia myös muilla metriikoilla.

7.7 Tulevaisuus

ChatGPT:n ensimmäinen versio julkaistiin tämän pro gradu -tutkielman tekemisen aikana ja se on luonteeltaan Gozalo-Brizuela ja Garrido-Merchan 2023 mukaan generatiivinen tekoäly ja hyödyntää transformer-arkkitehtuuria. ChatGPT:n julkaisemisen jälkeen on käynyt ilmeiseksi, että tämäntyyppisellä tekoälyteknologialla tulee olemaan valtavia vaikutuksia tutkijoiden työskentelytapaan (Dis ym. 2023).

Aydın ja Karaarslan 2023 mukaan myös Google esitteli keskustelevan tekoälyn nimeltään Bard helmikuussa 2023. Bardin moottorina toimii LaMBDA ja se perustuu transformer-arkkitehtuuriin. Bard-tekoäly on pilvipohjainen keskustelevan tekoälyn alusta, jonka avulla organisaatiot voivat rakentaa ja ottaa käyttöön chatbotteja, jotka voivat kommunikoida kuluttajien kanssa esimerkiksi verkkosivustojen kautta (Aydın ja Karaarslan 2023).

Aiempien tutkimusten mukaan chatbottien kehitys lähti nopeaan kasvuun vuonna 2016. Tässä pro gradu -tutkielmassa käytettyjen tuoreimpien tutkimusten pohjalta vaikuttaa siltä, että vuoden 2022 loppu on ollut käänteentekevä chatbottien toteutuksen ja toiminnallisuuden osalta ja niiden tulevaisuuden kehityksen kannalta. Tällä tulee olemaan luultavasti myös kauaskantoiset vaikutukset ihmisten elämään. Chatbottien uudesta ja tulevasta aikakaudesta huolimatta olemassa olevat chatbottien dialogien toteuttamisen periaatteet ovat ajankohtaisia nyt ja tulevaisuudessa eri tavoin keskusteleville chatboteille. Kuitenkaan ohjenuoria ei tulisi noudattaa liian orjallisesti. Esitettyä Nielsenin heuristiikkaa dialogeille voi pitää kuitenkin kulmakivenä tietoa jakaville ja kysymyksiin vastaaville chatboteille myös tulevaisuudessa.

8 Johtopäätökset

Transformereihin perustuvat abstrahoivat yhteenvetomenetelmät luovat tulosten perusteella laadukkaampia yhteenvetoja kuin ydinvirkkeitä poimivat ekstraktiiviset menetelmät. Koska abstrahoidun yhteenvedon idea ei ole sisällyttää sellaisenaan kokonaisia lähdetekstin virkkeitä ja se voi käyttää lähdetekstin ulkopuolista sanastoa, niin abstrahoitu yhteenveto voi olla tiiviimpi kuin ekstraktiivinen yhteenveto.

Yhteenvetojen automaattisessa laadunarvioimisessa *n*-grammien päällekkäisyyteen perustuvana menetelmänä *de facto* ROUGE ei välttämättä anna abstrahoivan transformerin kyvystä generoida yhteenvetoa parasta mahdollista kuvaa. Päällekkäisiin *n*-grammeihin perustuvat metriikat soveltuvat yhteenvetojen arviointiin, joissa lähdetekstiä ei ole muutettu, vaan poimittu sellaisenaan ydinajatukset virkkeineen. Tiivis abstrahoitu yhteenveto puolestaan voi käsittää myös lähdetekstin ulkopuolista sanastoa. Vaihtoehtoisia sanastoja voi olla useita.

Tämän pro gradu -tutkimuksen tulosten pohjalta vaikuttaakin siltä, että metriikka, joka huomioi semanttisen samankaltaisuuden kontekstin kanssa tai faktojen sisältymisen yhteenvetoon lähdetekstistä, antaa abstrahoidun yhteenvedon laadusta realistisemman kuvan, kuin leksikaaliseen *n*-grammien päällekkäisyyteen perustuvat metriikat. Myös olemassa oleva tutkimus tukee tätä päätelmää. Siksi suositellaan tarkistamaan abstrahoitujen yhteenvetojen automaattista laadunarviointia.

Chatbotin tulosteen lukemisaika vaikuttaa käyttäjätyytyväisyyteen. Oli tuloste osa laajempaa lähdetekstiä tai generoitu, sen on syytä olla tietoa jakavilla tai kysymyksiin vastaavilla chatboteilla informatiivinen ja napakka. Sosiaalisen chatbotin tulee vastata hyvin lyhyesti vain muutamalla sanalla mahdollisesti myös ei-formaalilla kielellä minimoimisen periaatetta noudattaen tarjotakseen inhimillisen vuorovaikutuksen kaltaisen käyttäjäkokemuksen.

Chatbotin tekstin tuottamista arvioitaessa on huomioitava myös chatbotin käyttötarkoitus. Jos siihen kuuluu myös esseen luominen, niin tuloste voi olla niinkin pitkä, että käyttäjä ei lue sitä heti välttämättä kerralla. Kuitenkaan tulosteessa ei tule olla ylimääräisiä tekstiyksiköitä. Turingin testin läpäisemistä ei siis ehkä välttämättä aina tule pitää chatbotin vuorovaikutustaidon mittarina, vaan chatbotin suoriutumista käyttötarkoituksestaan.

Lähteet

Agrawal, Kanika. 2020. “Legal case summarization: an application for text summarization”. Teoksessa *2020 International conference on computer communication and informatics (ICCCI),* 1–6. IEEE.

Amala, Ihsan Ahsanu, Donni Richasdy ja Mahendra Dwifebri Purbolaksono. 2022. “Telkom University News Topic Modeling Using Latent Semantic Analysis (LSA) Method on Online News Portal”. *Building of Informatics, Technology and Science (BITS)* 4 (1): 110–115.

Amalia, Amalia, Maya Silvi Lydia, Siti Dara Fadilla, Miftahul Huda ja Dani Gunawan. 2017. “Document clustering optimization with synonym dictionary check function”. Teoksessa *2017 International Conference on Electrical Engineering and Informatics (ICELTICs),* 286–291. IEEE.

Amiri, Parham ja Elena Karahanna. 2022. “Chatbot use cases in the Covid-19 public health response”. *Journal of the American Medical Informatics Association* 29 (5): 1000–1010.

Anantharaman, Aditya, Arpit Jadiya, Chandana Tulasi Sai Siri, Bharath NVS Adikar ja Biju Mohan. 2019. “Performance evaluation of topic modeling algorithms for text classification”. Teoksessa *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),* 704–708. IEEE.

Arun, Rajkumar, Venkatasubramaniyan Suresh, CE Veni Madhavan ja MN Narasimha Murthy. 2010. “On finding the natural number of topics with latent dirichlet allocation: Some observations”. Teoksessa *Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part I 14,* 391–402. Springer.

Aydın, Ömer ja Enis Karaarslan. 2022. “OpenAI ChatGPT generated literature review: Digital twin in healthcare”. *Available at SSRN 4308687.*

———. 2023. “Is ChatGPT Leading Generative AI? What is Beyond Expectations?” *What is beyond expectations.*

Babar, Samrat, M Tech-Cse ja Rit. 2013. “Text Summarization:An Overview” (lokakuu).

Banerjee, Satanjeev ja Alon Lavie. 2005. “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”. Teoksessa *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization,* 65–72.

Bathija, Richeeka, Pranav Agarwal, Rakshith Somanna ja GB Pallavi. 2020. “Guided interactive learning through chatbot using bi-directional encoder representations from transformers (bert)”. Teoksessa *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA),* 82–87. IEEE.

Bhandari, Manik, Pranav Gour, Atabak Ashfaq, Pengfei Liu ja Graham Neubig. 2020. “Re-evaluating evaluation in text summarization”. *arXiv preprint arXiv:2010.07100.*

Blei, David M. 2012. “Probabilistic topic models”. *Communications of the ACM* 55 (4): 77–84.

Blei, David M, Andrew Y Ng ja Michael I Jordan. 2003. “Latent dirichlet allocation”. *Journal of machine Learning research* 3 (Jan): 993–1022.

Brandtzaeg, Petter Bae ja Asbjørn Følstad. 2018. “Chatbots: changing user needs and motivations”. *Interactions* 25 (5): 38–43.

Cajueiro, Daniel O, Arthur G Nery, Igor Tavares, Maísa K De Melo, Silvia A dos Reis, Li Weigang ja Victor RR Celestino. 2023. “A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding”. *arXiv preprint arXiv:2301.03403.*

Caldarini, Guendalina, Sardar Jaf ja Kenneth McGarry. 2022. “A Literature Survey of Recent Advances in Chatbots”. *Information* 13 (1): 41.

Cañizares, Pablo C, Sara Pérez-Soler, Esther Guerra ja Juan de Lara. 2022. “Automating the measurement of heterogeneous chatbot designs”. Teoksessa *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing,* 1491–1498.

Cao, Juan, Tian Xia, Jintao Li, Yongdong Zhang ja Sheng Tang. 2009. “A density-based method for adaptive LDA model selection”. *Neurocomputing* 72 (7-9): 1775–1781.

Chen, Chen ja Wei Emma Zhang. 2023. “Incorporating Knowledge into Document Summarization: an Application of Prefix-Tuning on GPT-2”. *arXiv preprint arXiv:2301.11719.*

Chen, Yingying, Zhao Peng, Sei-Hill Kim ja Chang Won Choi. 2023. “What We Can Do and Cannot Do with Topic Modeling: A Systematic Review”. *Communication Methods and Measures,* 1–20.

Clevert, Djork-Arné, Thomas Unterthiner ja Sepp Hochreiter. 2015. “Fast and accurate deep network learning by exponential linear units (elus)”. *arXiv preprint arXiv:1511.07289.*

Cosma, Georgina ja Mike Joy. 2012. “Evaluating the performance of lsa for source-code plagiarism detection”. *Informatica* 36 (4).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee ja Kristina Toutanova. 2018. “Bert: Pre-training of deep bidirectional transformers for language understanding”. *arXiv preprint arXiv:1810.04805.*

Dis, Eva AM van, Johan Bollen, Willem Zuidema, Robert van Rooij ja Claudi L Bockting. 2023. “ChatGPT: five priorities for research”. *Nature* 614 (7947): 224–226.

Else, Holly. 2023. “Abstracts written by ChatGPT fool scientists”. *Nature* 613 (7944): 423–423.

Engel, Nico, Vasileios Belagiannis ja Klaus Dietmayer. 2021. “Point transformer”. *IEEE Access* 9:134826–134840.

Fan, Huilong ja Yongbin Qin. 2018. “Research on text classification based on improved tf-idf algorithm”. Teoksessa *2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018),* 501–506. Atlantis Press.

Gebre, Binyam Gebrekidan, Marcos Zampieri, Peter Wittenburg ja Tom Heskes. 2013. “Improving native language identification with tf-idf weighting”. Teoksessa *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications,* 216–223.

Ghojogh, Benyamin ja Ali Ghodsi. 2020. “Attention mechanism, transformers, BERT, and GPT: Tutorial and survey”.

Gholipour Ghalandari, Demian, Chris Hokamp ja Georgiana Ifrim. 2022. “Efficient Unsupervised Sentence Compression by Fine-tuning Transformers with Reinforcement Learning”. Teoksessa *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 1267–1280. Dublin, Ireland: Association for Computational Linguistics, toukokuu. https://arxiv.org/abs/2205.08221.

Ghosal, Tirthankar, Ondřej Bojar, Muskaan Singh ja Anja Nedoluzhko. 2021. “Overview of the first shared task on automatic minuting (automin) at interspeech 2021”. *Proceedings of the First Shared Task on Automatic Minuting at Interspeech,* 1–25.

Gliwa, Bogdan, Iwona Mochol, Maciej Biesek ja Aleksander Wawer. 2019. “SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization”. Teoksessa *Proceedings of the 2nd Workshop on New Frontiers in Summarization,* 70–79. Hong Kong, China: Association for Computational Linguistics, marraskuu. https://doi.org/10.18653/v1/D19-5409. https://aclanthology.org/D19-5409.

Goodrich, Ben, Vinay Rao, Peter J Liu ja Mohammad Saleh. 2019. “Assessing the factual accuracy of generated text”. Teoksessa *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* 166–175.

Gozalo-Brizuela, Roberto ja Eduardo C Garrido-Merchan. 2023. “ChatGPT is not all you need. A State of the Art Review of large Generative AI models”. *arXiv preprint arXiv:2301.04655.*

Greene, Derek, Derek O’Callaghan ja Pádraig Cunningham. 2014. “How many topics? stability analysis for topic models”. Teoksessa *Joint European conference on machine learning and knowledge discovery in databases,* 498–513. Springer.

Gregor, Shirley ja Alan R Hevner. 2013. “Positioning and presenting design science research for maximum impact”. *MIS quarterly,* 337–355.

Grootendorst, Maarten. 2022. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. *arXiv preprint arXiv:2203.05794.*

Guan, Wang, Ivan Smetannikov ja Man Tianxing. 2020. “Survey on automatic text summarization and transformer models applicability”. Teoksessa *2020 International Conference on Control, Robotics and Intelligent System,* 176–184.

Gupta, Som ja S. K Gupta. 2019. “Abstractive summarization: An overview of the state of the art”. *Expert Systems with Applications* 121:49–65. ISSN: 0957-4174. https://doi.org/https://doi.org/10.1016/j.eswa.2018.12.011. https://www.sciencedirect.com/science/article/pii/S0957417418307735.

Halliday, Michael Alexander Kirkwood ja Ruqaiya Hasan. 2014. *Cohesion in english.* 9. Routledge.

Hartl, Philipp ja Udo Kruschwitz. 2021. “University of Regensburg at CheckThat! 2021: Exploring Text Summarization for Fake News Detection.” *CLEF (Working Notes)* 2936:508–519.

Havrlant, Lukáš ja Vladik Kreinovich. 2017. “A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)”. *International Journal of General Systems* 46 (1): 27–36.

Hendrycks, Dan ja Kevin Gimpel. 2016. “Gaussian error linear units (gelus)”. *arXiv preprint arXiv:1606.08415.*

Hermann, Karl Moritz, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman ja Phil Blunsom. 2015. “Teaching machines to read and comprehend”. *Advances in neural information processing systems* 28.

Hevner, Alan R, Salvatore T March, Jinsoo Park ja Sudha Ram. 2004. “Design science in information systems research”. *MIS quarterly,* 75–105.

Hill, Jennifer, W Randolph Ford ja Ingrid G Farreras. 2015. “Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations”. *Computers in human behavior* 49:245–250.

Hingu, Dharmendra, Deep Shah ja Sandeep S Udmale. 2015. “Automatic text summarization of Wikipedia articles”. Teoksessa *2015 international conference on communication, information & computing technology (ICCICT),* 1–4. IEEE.

Hobbs, Jerry R. 1974. “A model for natural language semantics, Part I: The model”. *Yale University Department of Computer Science Research Report,* numero 36.

Holmes, Samuel, Anne Moorhead, Raymond Bond, Huiru Zheng, Vivien Coates ja Michael McTear. 2019. “Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces?” Teoksessa *Proceedings of the 31st European Conference on Cognitive Ergonomics,* 207–214.

Höhn, Sviatlana ja Kerstin Bongard-Blanchy. 2020. “Heuristic evaluation of COVID-19 chatbots”. Teoksessa *International Workshop on Chatbot Research and Design,* 131–144. Springer.

Jain, Mohit, Pratyush Kumar, Ramachandra Kota ja Shwetak N Patel. 2018. “Evaluating and informing the design of chatbots”. Teoksessa *Proceedings of the 2018 designing interactive systems conference,* 895–906.

Jones, Karen Sparck. 1972. “A statistical interpretation of term specificity and its application in retrieval”. *Journal of documentation.*

Kalepalli, Yaswanth, Shaik Tasneem, Pasupuleti Durga Phani Teja ja Suneetha Manne. 2020. “Effective comparison of lda with lsa for topic modelling”. Teoksessa *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS),* 1245–1250. IEEE.

Karl, Andrew, James Wisnowski ja W Heath Rushing. 2015. “A practical guide to text mining with topic extraction”. *Wiley Interdisciplinary Reviews: Computational Statistics* 7 (5): 326–340.

Kherwa, Pooja ja Poonam Bansal. 2020. “Topic modeling: a comprehensive review”. *EAI Endorsed transactions on scalable information systems* 7 (24).

Khurana, Alka ja Vasudha Bhatnagar. 2022. “Investigating entropy for extractive document summarization”. *Expert Systems with Applications* 187:115820.

Koh, Huan Yee, Jiaxin Ju, Ming Liu ja Shirui Pan. 2022. “An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics”. *ACM Comput. Surv.* (New York, NY, USA) 55, numero 8 (joulukuu). ISSN: 0360-0300. https://doi.org/10.1145/3545176. https://doi-org.ezproxy.jyu.fi/10.1145/3545176.

Kullback, Solomon ja Richard A Leibler. 1951. “On information and sufficiency”. *The annals of mathematical statistics* 22 (1): 79–86.

Kung, Tiffany H, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo ym. 2023. “Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models”. *PLOS Digital Health* 2 (2): e0000198.

Kågebäck, Mikael, Olof Mogren, Nina Tahmasebi ja Devdatt Dubhashi. 2014. “Extractive summarization using continuous vector space models”. Teoksessa *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC),* 31–39.

Langevin, Raina, Ross J Lordon, Thi Avrahami, Benjamin R Cowan, Tad Hirsch ja Gary Hsieh. 2021. “Heuristic evaluation of conversational agents”. Teoksessa *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems,* 1–15.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov ja Luke Zettlemoyer. 2019. “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. *arXiv preprint arXiv:1910.13461.*

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov ja Luke Zettlemoyer. 2020. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. Teoksessa *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 7871–7880. Online: Association for Computational Linguistics, heinäkuu. https://doi.org/10.18653/v1/2020.acl-main.703. https://aclanthology.org/2020.acl-main.703.

Lin, Chin-Yew. 2004. “Rouge: A package for automatic evaluation of summaries”. Teoksessa *Text summarization branches out,* 74–81.

Lin, Tianyang, Yuxin Wang, Xiangyang Liu ja Xipeng Qiu. 2022. “A survey of transformers”. *AI Open.*

Liu, Yang, Sheng Shen ja Mirella Lapata. 2020. “Noisy self-knowledge distillation for text summarization”. *arXiv preprint arXiv:2009.07032.*

Mao, Yuning, Ming Zhong ja Jiawei Han. 2022. “CiteSum: Citation Text-guided Scientific Extreme Summarization and Low-resource Domain Adaptation”. *arXiv preprint arXiv:2205.06207.*

March, Salvatore T ja Gerald F Smith. 1995. “Design and natural science research on information technology”. *Decision support systems* 15 (4): 251–266.

Mays, W. 1952. “Can Machines Think?” *Philosophy* 27 (101): 148–162.

Mihalcea, Rada ja Paul Tarau. 2004. “Textrank: Bringing order into text”. Teoksessa *Proceedings of the 2004 conference on empirical methods in natural language processing,* 404–411.

Mimno, David, Hanna Wallach, Edmund Talley, Miriam Leenders ja Andrew McCallum. 2011. “Optimizing semantic coherence in topic models”. Teoksessa *Proceedings of the 2011 conference on empirical methods in natural language processing,* 262–272.

Miner, Adam S, Liliana Laranjo ja A Baki Kocaballi. 2020. “Chatbots in the fight against the COVID-19 pandemic”. *NPJ digital medicine* 3 (1): 1–4.

Moore, Robert J, Margaret H Szymanski, Raphael Arar ja Guang-Jie Ren. 2018. “Studies in Conversational UX Design”.

Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, Bing Xiang ym. 2016. “Abstractive text summarization using sequence-to-sequence rnns and beyond”. *arXiv preprint arXiv:1602.06023.*

Naseem, Usman, Imran Razzak, Shah Khalid Khan ja Mukesh Prasad. 2021. “A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models”. *Transactions on Asian and Low-Resource Language Information Processing* 20 (5): 1–35.

Newman, David, Jey Han Lau, Karl Grieser ja Timothy Baldwin. 2010. “Automatic evaluation of topic coherence”. Teoksessa *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics,* 100–108.

Nguyen, Elisa, Daphne Theodorakopoulos, Shreyasi Pathak, Jeroen Geerdink, Onno Vijlbrief, Maurice Van Keulen ja Christin Seifert. 2020. “A hybrid text classification and language generation model for automated summarization of Dutch breast cancer radiology reports”. Teoksessa *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI),* 72–81. IEEE.

Nicolescu, Luminița ja Monica Teodora Tudorache. 2022. “Human-Computer Interaction in Customer Service: The Experience with AI Chatbots—A Systematic Literature Review”. *Electronics* 11 (10): 1579.

Pasquali, Arian Rodrigo. 2016. “Automatic coherence evaluation applied to topic models”.

Patel, Sajan B ja Kyle Lam. 2023. “ChatGPT: the future of discharge summaries?” *The Lancet Digital Health.*

Peffers, Ken, Tuure Tuunanen ja Björn Niehaves. 2018. *Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research,* 2.

Peffers, Ken, Tuure Tuunanen, Marcus A Rothenberger ja Samir Chatterjee. 2007. “A design science research methodology for information systems research”. *Journal of management information systems* 24 (3): 45–77.

Principe, Vitor Ayres, Rodrigo Gomes de Souza Vale, Juliana Brandao Pinto de Castro, Luiz Marcelo Carvano, Roberto Andre Pereira Henriques, Victor de Almeida e Sousa Lobo, Rodolfo de Alkmim Moreira Nunes ym. 2022. “A computational literature review of football performance analysis through probabilistic topic modeling”. *Artificial Intelligence Review* 55 (2): 1351–1371.

Rahali, Abir ja Moulay A Akhloufi. 2023. “End-to-End Transformer-Based Models in Textual-Based NLP”. *AI* 4 (1): 54–110.

Reimers, Nils ja Iryna Gurevych. 2019. “Sentence-bert: Sentence embeddings using siamese bert-networks”. *arXiv preprint arXiv:1908.10084.*

Rooein, Donya, Devis Bianchini, Francesco Leotta, Massimo Mecella, Paolo Paolini ja Barbara Pernici. 2022. “aCHAT-WF: Generating conversational agents for teaching business process models”. *Software and Systems Modeling* 21 (3): 891–914.

Ruane, Elayne, Sinead Farrell ja Anthony Ventresque. 2020. “User perception of text-based chatbot personality”. Teoksessa *International Workshop on Chatbot Research and Design,* 32–47. Springer.

Sacks, Harvey ja Emanuel A Schegloff. 1979. “Two preferences in the organization ofreference to persons in conversation and their interaction”. *Everyday Language: Studies in Ethnomethodology. New York.*

Sarkar, Dipanjan. 2019. *Text analytics with Python: a practitioner’s guide to natural language processing.* Springer.

Scialom, Thomas, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano ja Alex Wang. 2021. “Questeval: Summarization asks for fact-based evaluation”. *arXiv preprint arXiv:2103.12693.*

Sezgin, Emre, Joseph Sirrianni, Simon L Linwood ym. 2022. “Operationalizing and Implementing Pretrained, Large Artificial Intelligence Linguistic Models in the US Health Care System: Outlook of Generative Pretrained Transformer 3 (GPT-3) as a Service Model”. *JMIR Medical Informatics* 10 (2): e32875.

Shen, Yiqiu, Laura Heacock, Jonathan Elias, Keith D Hentel, Beatriu Reig, George Shih ja Linda Moy. 2023. *ChatGPT and Other Large Language Models Are Double-edged Swords.*

Shleifer, Sam ja Alexander M Rush. 2020. “Pre-trained summarization distillation”. *arXiv preprint arXiv:2010.13002.*

Singh, Sushant ja Ausif Mahmood. 2021. “The NLP cookbook: modern recipes for transformer based deep learning architectures”. *IEEE Access* 9:68675–68702.

Song, Shuangyong, Xiangyan Chen, Chao Wang, Xiaoguang Yu, Jia Wang ja Xiaodong He. 2022. “A Two-stage User Intent Detection Model on Complicated Utterances with Multi-task Learning”. Teoksessa *Companion Proceedings of the Web Conference 2022,* 197–200.

Steinberger, Josef, Karel Jezek ym. 2004. “Using latent semantic analysis in text summarization and summary evaluation”. *Proc. ISIM* 4 (93-100): 8.

Suanmali, Ladda, Naomie Salim ja Mohammed Salem Binwahlan. 2009. “Fuzzy logic based method for improving text summarization”. *arXiv preprint arXiv:0906.4690.*

Sugisaki, Kyoko ja Andreas Bleiker. 2020. “Usability guidelines and evaluation criteria for conversational user interfaces: a heuristic and linguistic approach”. Teoksessa *Proceedings of the Conference on Mensch und Computer,* 309–319.

Syed, Ayesha Ayub, Ford Lumban Gaol ja Tokuro Matsuo. 2021. “A survey of the state-of-the-art models in neural abstractive text summarization”. *IEEE Access* 9:13248–13265.

Thompson, Laure ja David Mimno. 2020. “Topic modeling with contextualized word representation clusters”. *arXiv preprint arXiv:2010.12626.*

Tsivitanidou, Olia ja Andri Ioannou. 2021. “Envisioned Pedagogical Uses of Chatbots in Higher Education and Perceived Benefits and Challenges”. Teoksessa *International Conference on Human-Computer Interaction,* 230–250. Springer.

Wang, Alex, Kyunghyun Cho ja Mike Lewis. 2020. “Asking and answering questions to evaluate the factual consistency of summaries”. *arXiv preprint arXiv:2004.04228.*

Wang, Xuequn, Xiaolin Lin ja Bin Shao. 2022. “Artificial intelligence changes the way we work: A close look at innovating with chatbots”. *Journal of the Association for Information Science and Technology.*

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser ja Illia Polosukhin. 2017. “Attention is all you need”. *Advances in neural information processing systems* 30.

Vayansky, Ike ja Sathish AP Kumar. 2020. “A review of topic modeling methods”. *Information Systems* 94:101582.

Wei, Xing ja W Bruce Croft. 2006. “LDA-based document models for ad-hoc retrieval”. Teoksessa *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval,* 178–185.

Wesslen, Ryan. 2018. “Computer-assisted text analysis for social science: Topic models and beyond”. *arXiv preprint arXiv:1803.11045.*

Wiering, Marco A ja Martijn Van Otterlo. 2012. “Reinforcement learning”. *Adaptation, learning, and optimization* 12 (3): 729.

Wilson, James R. 2002. “Responsible authorship and peer review”. *Science and engineering ethics* 8 (2): 155–174.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz ym. 2020. “Transformers: State-of-the-art natural language processing”. Teoksessa *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations,* 38–45.

Völkel, Sarah Theres, Ramona Schoedel, Lale Kaya ja Sven Mayer. 2022. “User Perceptions of Extraversion in Chatbots after Repeated Use”. Teoksessa *CHI Conference on Human Factors in Computing Systems,* 1–18.

Xie, Yuexiang, Fei Sun, Yang Deng, Yaliang Li ja Bolin Ding. 2021. “Factual consistency evaluation for text summarization via counterfactual estimation”. *arXiv preprint arXiv:2108.13134.*

Yamaguchi, Atsuki, Gaku Morio, Hiroaki Ozaki, Ken-ichi Yokote ja Kenji Nagamatsu. 2021. “Team hitachi@ automin 2021: Reference-free automatic minuting pipeline with argument structure construction over topic-based summarization”. *arXiv preprint arXiv:2112.02741.*

Yang, Xi ja Marco Aurisicchio. 2021. “Designing conversational agents: A self-determination theory approach”. Teoksessa *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems,* 1–16.

Zhang, Mengli, Gang Zhou, Wanting Yu, Ningbo Huang ja Wenfen Liu. 2022. “A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning”. *Computational Intelligence and Neuroscience* 2022.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q Weinberger ja Yoav Artzi. 2019. “Bertscore: Evaluating text generation with bert”. *arXiv preprint arXiv:1904.09675.*

Zhang*, Tianyi, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger ja Yoav Artzi. 2020. “BERTScore: Evaluating Text Generation with BERT”. Teoksessa *International Conference on Learning Representations.* https://openreview.net/forum?id=SkeHuCVFDr.

Zhao, Mingjun, Shengli Yan, Bang Liu, Xinwang Zhong, Qian Hao, Haolan Chen, Di Niu, Bowei Long ja Weidong Guo. 2021. “QBSUM: A large-scale query-based document summarization dataset from real-world applications”. *Computer Speech & Language* 66:101166.

Österle, Hubert, Jörg Becker, Ulrich Frank, Thomas Hess, Dimitris Karagiannis, Helmut Krcmar, Peter Loos, Peter Mertens, Andreas Oberweis ja Elmar J Sinz. 2011. “Memorandum on design-oriented information systems research”. *European journal of information systems* 20 (1): 7–10.

Liitteet

A Mimno ym. 2011 metriikalla LDA-mallinnuksen aiheet

=== Topic 1 ===

An inquest determined his death as a suicide, but it has been noted that the known evidence is also consistent with accidental poisoning. Following a public campaign in 2009, the British prime minister Gordon Brown made an official public apology on behalf of the British government for "the appalling way [Turing] was treated".

=== Topic 2 ===

He accepted hormone treatment with DES, a procedure commonly referred to as chemical castration, as an alternative to prison. Turing died on 7 June 1954, 16 days before his 42nd birthday, from cyanide poisoning.

=== Topic 3 ===

In 1948, Turing joined Max Newman's Computing Machine Laboratory, at the Victoria University of Manchester, where he helped develop the Manchester computers and became interested in mathematical biology. He wrote a paper on the chemical basis of morphogenesis and predicted oscillating chemical reactions such as the Belousov–Zhabotinsky reaction, first observed in the 1960s.

=== Topic 4 ===

He graduated at King's College, Cambridge, with a degree in mathematics. Whilst he was a fellow at Cambridge, he published a proof demonstrating that some purely mathematical yes–no questions can never be answered by computation and defined a Turing machine, and went on to prove that the halting problem for Turing machines is undecidable. Turing played a crucial role in cracking intercepted coded messages that enabled the Allies to defeat the Axis powers in many crucial engagements, including the Battle of the Atlantic. After the war, Turing worked at the National Physical Laboratory, where he designed the Automatic Computing Engine (ACE), one of the first designs for a stored-program computer.

=== Topic 5 ===

Queen Elizabeth II granted a posthumous pardon in 2013. The term "Alan Turing law" is now used informally to refer to a 2017 law in the United Kingdom that retroactively pardoned men cautioned or convicted under historical legislation that outlawed homosexual acts.

=== Topic 6 ===

For a time he led Hut 8, the section that was responsible for German naval cryptanalysis. Here, he devised a number of techniques for speeding the breaking of German ciphers, including improvements to the pre-war Polish bomba method, an electromechanical machine that could find settings for the Enigma machine. A 2019 BBC series, as voted by the audience, named him the greatest person of the 20th century.

=== Topic 7 ===

Alan Mathison Turing (; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. He is widely considered to be the father of theoretical computer science and artificial intelligence. Born in Maida Vale, London, Turing was raised in southern England. Turing has an extensive legacy with statues of him and many things named after him, including an annual award for computer science innovations. He appears on the current Bank of England £50 note, which was released on 23 June 2021, to coincide with his birthday.

=== Topic 8 ===

In 1938, he obtained his PhD from the Department of Mathematics at Princeton University. During the Second World War, Turing worked for the Government Code and Cypher School (GC & CS) at Bletchley Park, Britain's codebreaking centre that produced Ultra intelligence.

=== Topic 9 ===

Despite these accomplishments, Turing was never fully recognised in Britain during his lifetime because much of his work was covered by the Official Secrets Act. Turing was prosecuted in 1952 for homosexual acts.

B Cao ym. 2009 metriikalla LDA-mallinnuksen aiheet

=== Topic 1 ===

In 1938, he obtained his PhD from the Department of Mathematics at Princeton University. During the Second World War, Turing worked for the Government Code and Cypher School (GC & CS) at Bletchley Park, Britain's codebreaking centre that produced Ultra intelligence. An inquest determined his death as a suicide, but it has been noted that the known evidence is also consistent with accidental poisoning. Following a public campaign in 2009, the British prime minister Gordon Brown made an official public apology on behalf of the British government for "the appalling way [Turing] was treated".

=== Topic 2 ===

He accepted hormone treatment with DES, a procedure commonly referred to as chemical castration, as an alternative to prison. Turing died on 7 June 1954, 16 days before his 42nd birthday, from cyanide poisoning. A 2019 BBC series, as voted by the audience, named him the greatest person of the 20th century.

=== Topic 3 ===

Alan Mathison Turing (; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. He is widely considered to be the father of theoretical computer science and artificial intelligence. Born in Maida Vale, London, Turing was raised in southern England. Turing has an extensive legacy with statues of him and many things named after him, including an annual award for computer science innovations. He appears on the current Bank of England £50 note, which was released on 23 June 2021, to coincide with his birthday.

=== Topic 4 ===

He graduated at King's College, Cambridge, with a degree in mathematics. Whilst he was a fellow at Cambridge, he published a proof demonstrating that some purely mathematical yes–no questions can never be answered by computation and defined a Turing machine, and

went on to prove that the halting problem for Turing machines is undecidable. Despite these accomplishments, Turing was never fully recognised in Britain during his lifetime because much of his work was covered by the Official Secrets Act. Turing was prosecuted in 1952 for homosexual acts. Queen Elizabeth II granted a posthumous pardon in 2013. The term "Alan Turing law" is now used informally to refer to a 2017 law in the United Kingdom that retroactively pardoned men cautioned or convicted under historical legislation that outlawed homosexual acts.

=== Topic 5 ===

For a time he led Hut 8, the section that was responsible for German naval cryptanalysis. Here, he devised a number of techniques for speeding the breaking of German ciphers, including improvements to the pre-war Polish bomba method, an electromechanical machine that could find settings for the Enigma machine. Turing played a crucial role in cracking intercepted coded messages that enabled the Allies to defeat the Axis powers in many crucial engagements, including the Battle of the Atlantic. After the war, Turing worked at the National Physical Laboratory, where he designed the Automatic Computing Engine (ACE), one of the first designs for a stored-program computer. In 1948, Turing joined Max Newman's Computing Machine Laboratory, at the Victoria University of Manchester, where he helped develop the Manchester computers and became interested in mathematical biology. He wrote a paper on the chemical basis of morphogenesis and predicted oscillating chemical reactions such as the Belousov–Zhabotinsky reaction, first observed in the 1960s.