Transformerit luonnollisen kielen käsittelyssä: koulutus ja sovellukset

Turun yliopisto
Tietotekniikan laitos
TkK-tutkielma
Joulukuu 2024
Veera Ollila

TURUN YLIOPISTO
Tietotekniikan laitos

Veera Ollila: Transformerit luonnollisen kielen käsittelyssä: koulutus ja sovellukset

TkK-tutkielma, 20 s.
Joulukuu 2024

Tämän kandidaatin tutkielman aiheena on transformer-arkkitehtuuriin perustuvat kielimallit, erityisesti niiden toiminta ja rooli luonnollisen kielen käsittelyssä. Tutkielman tarkoituksena on selvittää, miten transformerien ydintekniikat, kuten huomiomekanismit, mahdollistavat tehokkaan tekstin analysoinnin ja tuottamisen.
Tutkimus on luonteeltaan teoreettinen ja perustuu merkittävään alan kirjallisuuteen. Pääpaino on transformeriarkkitehtuurissa ja siihen liittyvissä innovaatioissa, sekä myöhemmin kehittyneissä sovelluksissa.
Keskeinen havainto on, että transformeriarkkitehtuuri tarjoaa skaalautuvuutta ja monimutkaisten riippuvuuksien mallintamista pitkien tekstien käsittelyssä. Tämä johtuu erityisesti huomiomekanismien kyvystä painottaa merkityksellisiä tekstinosia tehokkaasti. Lisäksi tutkielmassa analysoidaan mallien koulutusprosesseja ja niihin liittyviä haasteita, kuten resurssivaatimuksia ja optimointistrategioita.
Päätelmänä todetaan, että transformerit ovat mullistaneet luonnollisen kielen käsittelyn monipuolisuutensa ja suorituskykynsä ansiosta. Samalla korostetaan tarvetta jatkotutkimukselle, joka keskittyy niiden energiatehokkuuden ja eettisten kysymysten, kuten väärinkäyttömahdollisuuksien, parantamiseen.

Asiasanat: kielimallit, kielimallien koulutus, transformerit, crewAI

Sisällys

Kuvat

1 Johdanto

Luonnollisen kielen käsittely, eng. Natural Language Processing (NLP), on tekoälyn osa-alue, joka keskittyy ihmiskielen ymmärtämiseen ja hyödyntämiseen tietokoneohjelmien avulla. Transformeriarkkitehtuuriin perustuvat kielimallit ovat viime vuosina mullistaneet NLP-sovelluksia niiden tehokkuuden ja monipuolisuuden ansiosta. Näiden mallien kyky analysoida ja tuottaa kieltä tarkasti perustuu erityisesti huomiomekanismin käyttöön, joka painottaa tekstin merkityksellisiä osia.

Tässä tutkielmassa tarkastellaan transformereiden toimintaa, sovelluksia ja koulutusstrategioita luonnollisen kielen käsittelyssä. Aiheen valinta on ajankohtainen, sillä transformerit ovat keskeinen osa modernia tekoälytutkimusta ja herättävät myös kysymyksiä liittyen laskennallisiin vaatimuksiin ja eettisiin näkökulmiin. Oleellisia käsitteitä ovat muun muassa huomiomekanismi, tokenisointi ja esikoulutetut kielimallit.

Tutkielman toisessa luvussa käsitellään kielimallien kouluttamista ja siinä hyödynnettäviä erilaisia teknologioita. Kolmannessa luvussa keskitytään kielimallien sovellukseen, tarkemmin Crew AI -systeemiin. Neljännessä luvussa käydään yhteenvetona tulokset tutkimuskysymyksiin.

1.1 Tutkimuskysymykset

Tutkielma on toteutettu kirjallisuuskatsauksena. Kirjallisuus koostuu tieteellisistä julkaisuista, tutkimuksista, artikkeleista ja kirjoista. Tutkielman tavoitteena on vas-

tata seuraaviin tutkimuskysymyksiin:

TK1 Mitä teknologioita hyödynnetään kielimallien kouluttamisessa?
Kielimallien kouluttamisessa käytetään monia teknologioita, kuten neuroverkkoarkkitehtuureja, esimerkiksi transformerit sekä laskennan optimointimenetelmiä, esimerkiksi tensorisuorittimet sekä jakautunut laskenta. Työssä tarkastellaan myös teknologioiden, kuten vastavirta-algoritmin, roolia mallien tehokkuuden parantamisessa.

TK2 Miten Crew AI toimii teknisesti?
Työssä perehdytään siihen, miten Crew AI käyttää transformeriarkkitehtuuria huomiomekanismeineen monimutkaisten tehtävien ratkaisemisessa. Työssä analysoidaan myös kuinka Crew AI hyödyntää lokaalia asennettavuutta tietoturvan ja suorituskyvyn parantamiseksi. Jätän tarkastelun ulkopuolelle sen liiketoimintakäyttöön liittyvät sovellukset.

1.2 Metodologia

Alan nopean kehityksen vuoksi lähteissä on kiinnitetty huomiota julkaisuaikaan. Suurin osa lähteistä on julkaistu vuoden 2020 jälkeen, muutamaa poikkeusta lukuun ottamatta, joista esimerkkinä tutkimus "Attention is All you Need"[1], joka on julkaistu vuonna 2017. Artikkelilla on ollut merkittävä vaikutus alan tutkimukseen, ja kyseisessä tutkimuksessa on luvussa 2.2 käsiteltävät transformerit ensimmäisen kerran esitelty.

Tutkielman lähdeaineistona on pääosin käytetty vertaisarvioituja tieteellisiä julkaisuja ja alan keskeisiä artikkeleita. Tietoa on kerätty ensisijaisesti englanninkielisistä lähteistä, sillä suomenkielistä materiaalia on aiheesta saatavilla niukasti. Lähteiden valinnassa on huomioitu julkaisujen luotettavuus ja ajankohtaisuus. Osa läh-

teistä on peräisin arxiv.org-sivustolta, kuitenkin vahvalla harkinnalla. Kolmannessa luvussa käsiteltäessä Crew AI:ta on lähteinä jouduttu käyttämään nettisivuja, sillä aihe on niin tuore, ettei siitä ole juurikaan vielä tieteellisiä julkaisuja tehty.

Tiedonhakuun käytettiin pääosin Google-Scholaria ja hakulauseina esimerkiksi ("artificial intelligence"OR "AI") AND ("training"OR "neural networks") sekä "neural networks" AND ("dataset quality" OR "data preprocessing"). Tutkielmassa on käytetty tekoälyä (ChatGPT 4.0) avuksi termien suomentamiseen.

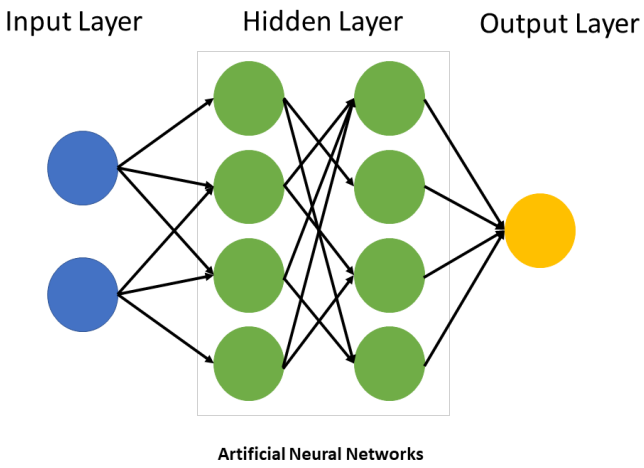
2 Teknologiat kielimallien kouluttamisen taustalla

Tämän kirjallisuuskatsauksen tavoitteena on vastata tutkimuskysymykseen TK1: Mitä teknologioita hyödynnetään kielimallien kouluttamisessa? Lähtökohtaisesti kielimalleja koulutetaan suurella määrällä dataa. Tulevissa luvuissa käsitellään, kuinka koulutusta voidaan tehostaa sekä kuinka koulutusmateriaalin laatua voidaan parantaa. Suurten kielimallien vahvuuksia ja heikkouksia käsitellään lisää katsauksessa "A Survey on Evaluation of Large Language Models" [2].

Suurten kielimallien esikouluttaminen kuluttaa paljon resursseja, kuten laskentatehoa ja energiaa. Artikkelissa "How to Train Data-Efficient LLMs"[3] käsitellään, kuinka kielimalleja voidaan kouluttaa tietotehokkaasti, josta yhtenä esimerkkinä perpleksisuodatus. Artikkelissa "Perplexed by Perplexity: Perplexity-Based Pruning With Small Reference Models"[4] todetaan, että perpleksisuodatuksessa, eng. perplexity-filtering, priorisoidaan näytteitä matalalla perpleksillä (epävarmuus) ja suodatetaan pois epätodennäköiset esimerkit. Lisäksi kielimallien kouluttamisen optimointiin voidaan hyödyntää muun muassa koulutusajon simulaattoreita, jakautuneita laskentaklustereita sekä erilaisia datan esikäsittelyteknologioita, joita käsitellään lisää luvussa 2.4.

2.1 Neuroverkot

Neuroverkot ovat syväoppimisen perusta ja jäljittelevät ihmisaivojen rakennetta ja toimintaa. Niissä on useita keinotekoisia neuroneita kerroksittain, joista kukin kerros käsittelee tietoa ja siirtää sen seuraavalle. Neuroverkon perusrakenne koostuu sisääntulo-, piilo- ja ulostulokerroksista, joissa piilokerrokset suorittavat suurimman osan laskennallisesta työstä. Artikkelissa "Neural network models and deep learning"[5] käsitellään aihetta lisää. Kuvassa 2.1 esitellään perinteinen neuroverkko.



Kuva 2.1: Neuroverkko

Erilaisia neuroverkkorakenteita ovat esimerkiksi eteenpäinkytketyt sekä yksi- että monikerroksiset perseptroniverkot, eng. single / multilayer perceptron, takaisinkytketyt neuroverkot, eng. recurrent neural network, sekä konvoluutioneuroverkot, eng. convolutional neural networks.

Perinteisessä eteenpäinkytketyssä arkkitehtuurissa informaatio kulkee vain yhteen suuntaan, kerroksesta toiseen. Artikkelissa "A survey of techniques for optimizing transformer inference"[6] kerrotaan, että lisäämällä huomiolohkoja malli kykenee oppimaan, mitkä syötteiden osat ovat tärkeimpiä kunkin tehtävän ratkaisemisessa. Huomiolohko laskee kunkin syötesekvenssin osan vaikutuksen suhteessa muihin osiin painotettujen pistetulojen avulla. Tämä mahdollistaa mallin keskittymisen

merkityksellisiin piirteisiin ja samalla vähentää hälyä, joka voi haitata ennustustarkkuutta.

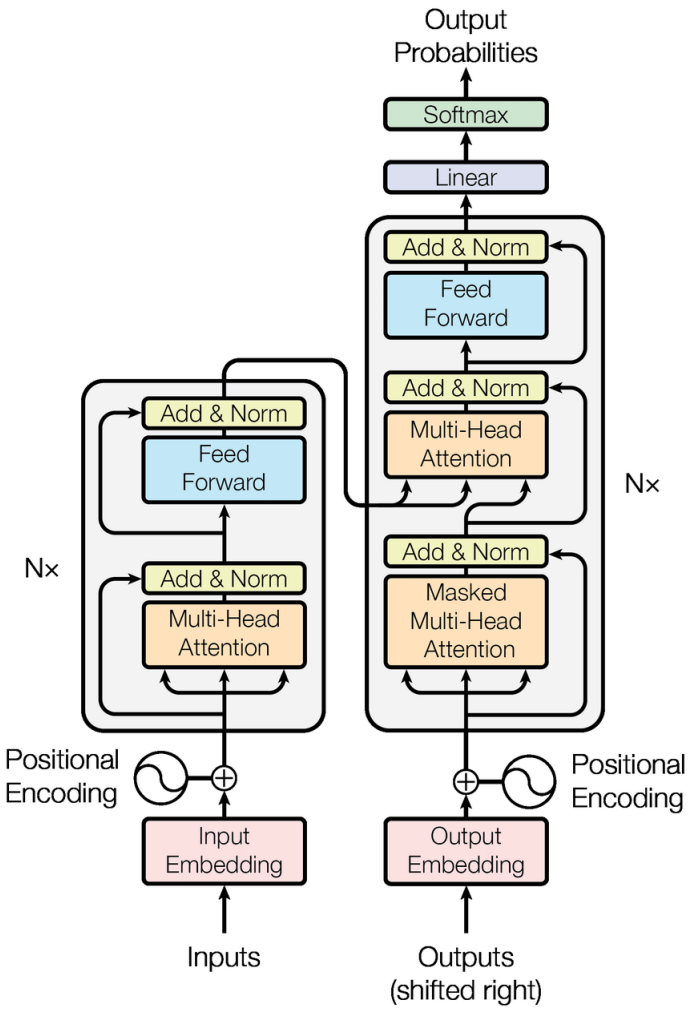
Syvät neuroverkot, eng. deep neural networks (DNN), eroavat perinteisistä neuroverkoista monikerroksisuutensa ansiosta, joka antaa niille mahdollisuuden käsitellä syvällisempiä ja monimutkaisempia tietorakenteita. Koulutuksessa tarvittava laskentateho on huomattava, minkä vuoksi grafiikkasuorittimet (GPU) ja tensorisuorittimet (TPU) ovat tulleet välttämättömiksi suurten neuroverkkojen tehokkaassa käsittelyssä. GPU:n vahvuus on rinnakkainen suoritus. Vaikka prosessori, eli CPU, onkin tehokkaampi ja pystyy suorittamaan nopeita kontekstinvaihtoja, GPU:n vie johtoasemaan sen kyky prosessoida rinnakkain.

Tensorisuorittimet, eng. Tensor Processing Unit (TPU), ovat Googlen tutkijoiden käsialaa, ja se on dokumentoitu Google Cloud-palveluun [7]. TPU:t ovat optimoitu neuroverkkojen matriisilaskentaan ja ovat olleet kriittisiä esikoulutustekniikoiden kehittämisessä. TPU:t ovat erityisen hyödyllisiä massiivisten kielimallien kouluttamisessa, koska ne lyhentävät koulutusaikaa ja vähentävät kustannuksia suuria datamääriä käsiteltäessä [7].

Vastavirta-algoritmi, eng. backpropagation, on keskeinen menetelmä neuroverkkojen koulutuksessa. Se laskee miten yksittäinen koulutusdata haluaisi muuttaa painoja ja siirtymiä, ei vain sitä pitäisikö arvojen kasvaa vai laskea. Artikkelissa "Comprehensive Review of Backpropagation Neural Networks"[8] kerrotaan, että algoritmin avulla neuroverkko oppii pienentämään ennustusvirhettä toistuvien laskelmien avulla. Tämä mahdollistaa monimutkaisten kuvioiden ja trendien tunnistamisen suurista tietomassoista, minkä ansiosta menetelmä on hyödyllinen erilaisissa tehtävissä kuten kuvan- ja puheentunnistuksessa sekä luonnollisen kielen käsittelyssä.

2.2 Transformerit

Transformeri on Googlen tutkijoiden vuonna 2017 esittelemä neuroverkkoarkkitehtuuri, joka pohjautuu huomiomekanismiin [1]. Transformeri on nykyään eniten käytetty arkkitehtuuri suurissa kielimalleissa, joten kyseistä tutkimusta voidaan pitää nykyisen tekoälyn pohjustana. Kuvassa 2.2 esitellään transformeri.



Kuva 2.2: Transformeri

Transformerit poikkeavat perinteisistä takaisinkytketyistä neuroverkoista siinä, että ne voivat käsittellä sekvenssidataa, kuten tekstiä, ilman että niiden tarvitsee käydä läpi dataa tietyssä järjestyksessä [1]. Sen sijaan transformerit käyttävät huomiomekanismia, eng. attention mechanism, joka painottaa eri osia datasekvenssis-

tä riippuen niiden merkityksestä kulloinkin käsiteltävälle osalle (vrt. kieli*malli* ja auto*malli*). Tämä mahdollistaa tehokkaamman ja joustavamman tavan käsitellä sekvenssejä ja on johtanut merkittäviin parannuksiin monilla sovellusalueilla.

Kielimallien prosessointiin käytetään tokenien numeerisia esityksiä eli vektoreita, joita käsitellään mallin sisällä toistuvien huomiolohkojen ja monikerroksisten perseptroniverkkojen avulla. Tokenit tarkoittavat pienempiä tekstin osia, eivät välttämättä aina tavuja, mutta hieman siihen suuntaan. Malli määritellään sen painoarvoilla, eng. weights, jotka säätävät mallin toimintaa ja ennusteita.

Huomiolohkokerrokset keskittyvät siihen, mitkä sanat ovat oleellisia toistensa merkityksen päivittämisessä ja kuinka nämä merkitykset tarkalleen päivitetään. Monikerroksisissa perseptroniverkoissa suoritetaan vektorien ja matriisien laskentaa. [1]

Huomiomekanismi

Huomiomekanismi painottaa tiettyjä syötteen osia laskennan aikana. Se auttaa mallia keskittymään syötteen olennaisiin osiin, esimerkiksi tiettyihin sanoihin lauseessa tai pikseleihin kuvassa. Mekanismi määrittää, mitkä osat syötteestä ovat tärkeämpiä tietyssä kontekstissa, ja antaa niille suuremman huomioarvon, eng. attention weight. Mekanismi jäljittelee ihmisen kykyä kohdistaa huomionsa olennaisiin asioihin [1]. Huomiomekanismi on merkittävä edistysaskel koneoppimisessa, sillä se parantaa mallien suorituskykyä monimutkaisissa tehtävissä ja mahdollistaa joustavamman tiedon käsittelyn verrattuna perinteisiin neuroverkkoarkkitehtuureihin.

Esimerkkinä huomiomekanismista on transformerimallien self-attention, jossa jokainen syöte-elementti huomioi suhteensa muihin elementteihin. Huomiomekanismi toimii siis osana suurempaa laskennallista prosessia, eikä ole itsessään konkreettinen rakenne.

Huomiolohkot sen sijaan ovat konkreettisia rakenteita neuroverkoissa, jotka käyttävät huomiomekanismia osana laskentaa. Se sisältää kaikki tarvittavat osat meka-

nismin toteuttamiseksi, kuten painot, matriisitoiminnot ja laskennalliset operaatiot. Huomiolohko on siis fyysinen osa mallin arkkitehtuuria.

2.3 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely, eng. Natural Language Processing (NLP), tarkoittaa tekoälyn osa-aluetta, jonka avulla tietokoneohjelmat tulkitsevat ja käyttävät ihmiskieltä. Artikkelissa "An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools"[9] käsitellään syväoppimisen tekniikoiden soveltamista luonnollisen kielen käsittelyssä.

Eräs NLP:n osalta tärkeä työkalu on transformerit, joita käsiteltiin luvussa 2.2. Transformeripohjaiset esikoulutetut kielimallit, eng. Transformer-based pretrained language models (T-PTMLs), kuten GPT-1 ja BERT, ovat menestyneet NLP:ssä kiitos niiden kyvyn oppia suuresta määrästä luokittelematonta dataa. Tätä käsitellään artikkelissa "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing"[10]

2.4 Transformeripohjaisten kielimallien kehitys ja optimointi

Artikkelissa "Taking ChatGPT as an example to analyze the main technologies used in large language models"[11] todetaan, että transformeripohjaisten esikoulutettujen kielimallien evoluutio alkoi malleilla BERT ja GPT. BERT (Bidirectional Encoder Representations from Transformers), tarkoittaa kaksisuuntaista kieliedustusten mallia. GPT (Generative Pre-trained Transformer) tarkoittaa generatiivista esikoulutettua transformeria.

BERT

BERT on kaksisuuntainen kieliedustusten malli, joka yhdistää esikoulutuksen ja hienosäädön NLP-tehtävissä (Natural Language Processing). Sen esikoulutus sisältää kaksi keskeistä tehtävää: maskattu kielimallinnus ja seuraavan lauseen ennustus. BERT on Googlen julkaisema, ja se on dokumentoitu Google Cloud-palveluun [12].

Artikkelissa "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"[13] kerrotaan, että esikoulutetut BERT-mallit tarjoavat vahvan pohjan, jota käyttäjät voivat hienosäätää vastaamaan erilaisia NLP-tarpeita. Googlen tutkijat ovat kehittäneet menetelmiä, joilla yleiskieliedustuksia voidaan kouluttaa massiivisen, merkkaamattoman tekstiaineiston avulla. Näin saatuja malleja voidaan soveltaa pienempien aineistojen NLP-tehtäviin, kuten kysymysten vastaamiseen tai tunneanalyysiin, mikä tuo merkittäviä tarkkuusparannuksia verrattuna kouluttamiseen täysin tyhjästä.

Suuret ja monimutkaiset syväoppimismallit, kuten BERT, vaativat huomattavaa laskentatehoa. Tensorisuorittimet, eng. Tensor Processing Unit (TPU), ovat Googlen tutkijoiden käsialaa, ja se on dokumentoitu Google Cloud-palveluun [7]. TPU:t ovat optimoitu neuroverkkojen matriisilaskentaan ja ovat olleet kriittisiä muun muassa BERT:n esikoulutustekniikoiden kehittämisessä. TPU:t ovat erityisen hyödyllisiä massiivisten kielimallien kouluttamisessa, koska ne lyhentävät koulutusaikaa ja vähentävät kustannuksia suuria datamääriä käsiteltäessä. [7]

GPT

Generatiivinen esikoulutettu transformeri, eng. Generative Pre-Trained Transformer (GPT), hyödyntää transformeriarkkitehtuuria ihmismäisen tekstin tuottamiseen.

Esikoulutuksessa malli opetetaan valtavan tekstiaineiston avulla oppimaan kielellisiä rakenteita ja merkityksiä. Esikoulutuksessa GPT ennustaa seuraavaa sanaa annetussa kontekstissa, mikä kehittää sen kykyä ymmärtää sanojen välistä yhteyttä

ja kieliopillisia rakenteita. Tämä vaihe antaa mallille laajan kielituntemuksen, vaikka koulutus ei sisälläkään erityisiä tehtäviä.

Kun perusrakenteet on opittu, malli voidaan hienosäätää pienemmän aineiston avulla tiettyihin tehtäviin, kuten kysymys-vastaus -toimintoihin, yhteenvedon luontiin tai keskusteluihin.

GPT-mallin ytimessä ovat luvussa 2.2 mainitut huomiolohkot. Painoarvot määrittelevät, miten malli käsittelee sanoja ja lauseita. Malli tuottaa ennusteita hyödyntämällä softmax-funktiota, joka arvioi todennäköisimmät sanat tai lauseet annetun kontekstin perusteella.

Viime vuosina GPT-mallit ovat kehittyneet huomattavasti. Esikoulutetut tekniikat ovat avainasemassa suuren luokan kielimallien menestykseen. GPT-mallit ymmärtävät paremmin kielen rakenteita. Kielen generointi on myös kehittynyt. [11]

Vertailu kielimallien välillä

Transformeriarkkitehtuuriin perustuvat kielimallit, kuten GPT ja BERT, ovat suosituimpia ja laajimmin käytettyjä luonnollisen kielen käsittelyn malleja. Molemmat mallit ovat mullistaneet alaa omilla ainutlaatuisilla ominaisuuksillaan ja sovelluksillaan, mutta niillä on myös merkittäviä eroja, jotka vaikuttavat niiden käyttöön eri konteksteissa.

GPT on ensisijaisesti generatiivinen malli, joka on suunniteltu tuottamaan ihmismäistä tekstiä. Sen koulutuksessa keskitytään seuraavan sanan ennustamiseen annetussa kontekstissa. GPT:n arkkitehtuuri on yksisuuntainen, mikä tarkoittaa, että se käsittelee tekstin kontekstia vasemmalta oikealle. Tämä yksisuuntaisuus antaa GPT:lle erityisen etulyöntiaseman tekstin generoinnissa, sillä se voi keskittyä intuitiivisesti rakentamaan johdonmukaisia ja luovia vastauksia tai tarinoita.

Toisaalta BERT on kaksisuuntainen malli, joka käsittelee tekstin kontekstia samanaikaisesti sekä vasemmalta oikealle että oikealta vasemmalle. Tämä tekee siitä

erityisen hyödyllisen tehtävissä, joissa tarvitaan tarkkaa ymmärrystä tekstin merkityksestä, kuten kysymys-vastaus -järjestelmissä, tekstiluokittelussa ja hakukoneoptimoinnissa. BERT:n esikoulutus perustuu maskatun kielimallinnuksen ja seuraavan lauseen ennustamisen yhdistelmään, mikä mahdollistaa sen syvällisen kielentuntemuksen. Kaksisuuntainen prosessointi antaa BERT:lle kyvyn ymmärtää monimutkaisia kielirakenteita ja asiayhteyksiä.

Vaikka molemmat mallit hyödyntävät transformeriarkkitehtuuria ja huomiomekanismeja, niiden sovellusalueet ja painopisteet eroavat toisistaan. GPT on ihanteellinen tehtäviin, joissa tarvitaan luovuutta ja tekstin tuottamista, kun taas BERT soveltuu paremmin analysointiin ja tarkkaan kielelliseen ymmärrykseen.

Optimointi

Välitysmalli, eng. proxy model, on alkuperäistä mallia yksinkertaisempi, joten sen suoritus on nopeampaa ja vaatii vähemmän resursseja. Artikkelissa "A convolutional neural network-based proxy model for field production prediction and history matching"[14] mukaan malli suunnitellaan usein tiettyyn tehtävään tai osittaiseen ongelmanratkaisuun, kuten testaamiseen, ennustamiseen tai optimointiin. Esimerkiksi monimutkainen neuroverkko voidaan korvata yksinkertaisemmalla mallilla, joka jäljittelee alkuperäisen verkon käyttäytymistä. Simulaatioissa käytetään myös usein välitysmalleja laskennan nopeuttamiseksi.

Ydinjoukot, eng. coresets, ovat pieniä tietotiivistelmiä, jotka ovat tehokkaita mallikoulutuksessa. Artikkelissa "Coresets via Bilevel Optimization for Continual Learning and Streaming"[15] kerrotaan, että ydinjoukko viittaa pienempään alijoukkoon alkuperäisestä materiaalista, joka pystyy säilyttämään sen tärkeimmät ominaisuudet. Ydinjoukon tavoitteena on vähentää laskentatehoa ja parantaa algoritmien tehokkuutta ilman, että menetetään liikaa tietoa. Ydinjoukkoja käytetään muun muassa optimoinnissa ja klusteroinnissa, koska ne ovat alkuperäistä materiaalia pienempiä

ja helpommin käsiteltäviä, mutta sisältävät kuitenkin riittävän tiedon materiaalista.

Koulutusajon simulaattorit ovat suhteellisen edullisia ajaa, ja niitä voidaan käyttää arvioimaan koulutusdatan epävarmuutta. Koulutusajossa malli käy läpi useita ajoja ennustustarkkuuden parantamiseksi. Esimerkkinä neuroverkkojen koulutus, jossa malli oppii säätämällä painojaan ja parametrejaan toistuvasti. Simulaattorin etuna on, että ei ole tarvetta käyttää aitoja resursseja tai riskejä. [3]

Jakautuneet laskentaklusterit, eng. distributed computing, mahdollistavat kielimallien kouluttamisen rinnakkaisesti useilla palvelimilla tai pilvilaskentayksiköillä, minkä ansiosta voidaan hyödyntää massiivista määrää laskentatehoa. Tämä tekee mahdolliseksi käsitellä valtavia tekstiaineistoja tehokkaasti ja nopeuttaa mallin koulutusta.

Datan esikäsittelyteknologiat, kuten maskattu kielimallinnus ja tokenisointi, varmistavat, että teksti saadaan mallin syötteeksi optimaalisessa muodossa. Maskattu kielimallinnus auttaa mallia oppimaan sanojen merkityksiä eri konteksteissa. Tokenisointi puolestaan muuntaa sanat numeerisiksi esityksiksi, joita neuroverkko pystyy käsittelemään.

3 Kielimallien sovellus ja mahdollisuudet

Tässä luvussa vastataan tutkimuskysymykseen TK2: Miten Crew AI toimii teknisesti? Crew AI on tekoälypohjainen järjestelmä, joka on suunniteltu tukemaan erityisesti tiimityöskentelyä ja yhteistyön tehostamista. Crew AI:sta on vähän tieteellisiä julkaisuja, mutta sen kehittäjät ovat koonneet dokumentaatiota nettisivuilleen (crewai.net) [16]. Crew AI:lle on myös GitHub-repositorio (github.com/crewAIInc/crewAI) [17].

Crew AI on transformeriarkkitehtuuriin perustuva luonnollisen kielen käsittelyn (NLP) järjestelmä, joka on suunniteltu tehokkaaseen tekstianalyysiin ja -generointiin. Se hyödyntää esikoulutettuja kielimalleja, huomiomekanismeja ja jakautunutta laskentaa suurten datamäärien käsittelyyn. Crew AI:n tärkeimpiä sovellusalueita ovat monimutkaisten raporttien luominen, automaattinen tekstin tiivistäminen ja tietojen analysointi reaaliaikaisesti. Tekninen toteutus perustuu useisiin transformerikerroksiin, jotka mahdollistavat syötteiden kontekstin ymmärtämisen ja monimutkaisten riippuvuuksien mallintamisen. Käytännön sovelluksissa Crew AI korostaa käytettävyyttä ja sisäistä yhteistyötä, mikä tekee siitä aidosti käytännöllisen työkalun todellisessa maailmassa.

3.1 Kielimallien sovellus

Kielimallit ovat luonnollisen kielen käsittelyn keskeinen osa-alue, ja niiden sovellukset ulottuvat monille elämän osa-alueille. Viime vuosien kehitys, erityisesti transformeriarkkitehtuuriin perustuvissa malleissa, on tehnyt kielimallien sovelluksista entistä monipuolisempia ja tehokkaampia. Nämä mallit eivät ole ainoastaan mullistaneet tapaa, jolla tietokoneet ymmärtävät ja tuottavat kieltä, vaan ne ovat myös avanneet uusia mahdollisuuksia eri toimialoille. Teknologian jatkuva kehitys on mahdollistanut mallien soveltamisen yhä monimutkaisempiin tehtäviin, mikä puolestaan on lisännyt niiden merkitystä niin yritysmaailmassa kuin akateemisessa tutkimuksessakin.

Kielimallien sovelluskohteet kattavat laajan skaalan, esimerkkinä automatisoidut asiakaspalvelut, sisällöntuotanto, tiedonhaku ja analyysi, käännösteknologiat sekä terveydenhuollon ratkaisut. Esimerkiksi asiakaspalvelun chatbotit voivat vastata monimutkaisiin kysymyksiin ja käsitellä suuria määriä asiakasviestintää reaaliajassa, mikä vapauttaa resursseja muihin tehtäviin. Tämän kaltaiset sovellukset eivät pelkästään lisää tehokkuutta, vaan ne voivat myös tuottaa merkittäviä kustannussäästöjä, kun aikaa vieviä tehtäviä voidaan automatisoida.

Käännösteknologioissa, kuten Google Kääntäjässä (translate.google.fi) [18] ja DeepL:ssä (www.deepl.com) [19], esikoulutetut kielimallit mahdollistavat tarkemmat ja luonnollisemmat käännökset eri kielten välillä. Tämä on erityisen hyödyllistä globaaleille organisaatioille, jotka tarvitsevat tehokkaita ratkaisuja monikieliseen viestintään. Samalla terveydenhuollossa kielimallit voivat analysoida suuria määriä potilasdataa, tunnistaa sairauksien merkkejä sekä tukea diagnoosiprosessia. Tällaisten ratkaisujen ansiosta voidaan paitsi parantaa potilasturvallisuutta, myös säästää resursseja ja vähentää inhimillisiä virheitä. Tämä osoittaa, miten teknologia voi auttaa ratkaisemaan kriittisiä yhteiskunnallisia ongelmia.

3.2 Crew AI

Crew AI on yhteistoiminnallinen työskentelyjärjestelmä, jonka tärkeimpiin ominaisuuksiin kuuluvat roolipohjaiset agentit ja tiimityöominaisuudet. Agentit hoitavat kukin tarkasti määritellyn roolinsa, ja tiimityöominaisuuksien avulla agentit voivat kommunikoida, jakaa tehtävätietoja ja avustaa toisiaan. Crew AI organisoi useita älykkäitä agentteja tiimiksi ja korostaa yhteistyötä saumattoman suorituskyvyn varmistamiseksi tehtävien suorittamisen aikana. [16]

Crew AI soveltuu erilaisiin skenaarioihin, erityisesti sellaisiin, jotka edellyttävät yhteistoiminnallista työskentelyä monimutkaisissa tehtävissä. Se parantaa tiimin kokonaisvaltaisia valmiuksia, yksinkertaistaa päätöksentekoprosesseja, lisää luovuutta ja vastaa monimutkaisiin haasteisiin. Crew AI:n keskeinen ero muihin tekoälytyökaluihin verrattuna on sen painotus tiimityöskentelyyn, jolla saavutetaan saumaton koordinointi useiden älykkäiden agenttien välillä. [16]

Crew AI:n agentit

Crew AI hyödyntää roolipohjaisia agentteja, jotka on suunniteltu hoitamaan tiettyjä tehtäviä yhteistyössä tiimin kanssa. Nämä agentit kommunikoivat keskenään, jakavat tietoa ja koordinoivat tehtävien suorittamista. Esimerkiksi agentit voivat vastata reaaliaikaisista tietoanalyysitehtävistä, luovasta sisällöntuotannosta tai päätöksenteon tukemisesta monimutkaisissa ympäristöissä.

Järjestelmä perustuu agenttien autonomisuuteen ja yhteistyökykyyn, mikä erottaa sen monista muista tekoälysovelluksista. Agentit voivat esimerkiksi mukautua dynaamisiin tilanteisiin, oppia suorituksistaan ja optimoida prosessejaan jatkuvasti. Tämä tekee Crew AI:sta erityisen hyödyllisen monimutkaisissa ja nopeasti muuttuvissa skenaarioissa, kuten projektinhallinnassa. [16]

Crew AI:n lähestymistapa hyödyntää huomiomekanismia ja transformeriarkkitehtuuria, mikä mahdollistaa kontekstuaalisen ja yksityiskohtaisen tiedon proses-

soinnin tehokkaasti. Näin järjestelmä voi käsitellä suuria tietomääriä ja tarjota merkityksellisiä vastauksia sekä analyysejä reaaliajassa. Crew AI:n agenttien keskeinen etu on kyky työskennellä itsenäisesti samalla, kun ne ylläpitävät tiivistä yhteistyötä tiiminsä muiden komponenttien kanssa

Crew AI -järjestelmässä "työkalu"määritellään taitona tai toimintona, jota tekoälyagentti voi hyödyntää erilaisten tehtävien suorittamiseen. Tämä sisältää CrewAI Toolkitin ja LangChain-työkalut, jotka tukevat kaikkea yksinkertaisista hauista monimutkaisiin vuorovaikutuksiin ja tehokkaaseen yhteistyöhön agenttien välillä. [16]

Keskeisiä ominaisuuksia ovat käytännöllisyys, integrointi ja muokattavuus. Työkalut on suunniteltu laajaa tehtäväkirjoa varten, kuten verkkohakuihin, data-analyysiin, sisällöntuotantoon ja agenttiyhteistyöhön. Ne parantavat agenttien kyvykkyyttä integroimalla työkalut suoraan agenttien työnkulkuun ja tarjoavat joustavuutta mukauttaa tai käyttää olemassa olevia työkaluja erityistarpeiden mukaan. Tämä työkalupakki tarjoaa tehokkaita keinoja laajentaa tekoälyagenttien kyvykkyyksiä, mahdollistaen monenlaisten tehtävien suorittamisen ja tehokkaan yhteistyön.

3.3 Paikallisuus

Yksi Crew AI:n merkittävistä piirteistä on mahdollisuus asentaa se lokaalisti, mikä tarjoaa etuja verrattaessa pilvipohjaisiin ratkaisuihin. Lokaalisti asennettu tekoälyjärjestelmä käsittelee ja tallentaa tiedot paikallisesti, mikä vähentää tietovuotojen riskiä. Tämä on erityisen tärkeää organisaatioille, jotka käsittelevät luottamuksellista tai arkaluontoista dataa, kuten henkilötietoja tai liikesalaisuuksia.

Paikallisesti toimiva tekoäly voi prosessoida dataa ilman, että se lähetetään ulkoisiin palvelimiin. Tämä vähentää viivettä ja parantaa järjestelmän suorituskykyä, mikä on kriittistä sovelluksissa, joissa tarvitaan reaaliaikaista analyysiä. Lokaalit järjestelmät eivät ole riippuvaisia internet-yhteyksistä tai pilvipalveluiden saatavuudesta, mikä tekee niistä luotettavampia erityisesti tilanteissa, joissa verkkoyhteydet

ovat epävarmoja. Tämä ominaisuus on erityisen arvokas esimerkiksi teollisuuden ja logistiikan sovelluksissa.

Lokaalisti asennettua järjestelmää voidaan mukauttaa organisaation erityistarpeisiin. Käyttäjät voivat hienosäätää mallia paikallisesti saatavilla olevan datan avulla ilman ulkopuolisten palveluntarjoajien rajoituksia. Tämä mahdollistaa paremman integraation olemassa oleviin prosesseihin ja infrastruktuuriin.

Vaikka lokaalit tekoälyratkaisut tarjoavat useita etuja, ne vaativat myös merkittäviä resursseja, kuten laskentatehoa ja teknistä osaamista järjestelmien ylläpitämiseen. Tästä huolimatta niiden tarjoama tietoturva, nopeus ja riippumattomuus tekevät niistä varteenotettavan vaihtoehdon. Crew AI edustaa tätä yhdistäen transformeriarkkitehtuurin edut lokaalisti asennettujen järjestelmien tarjoamiin hyötyihin. [16]

4 Yhteenveto

Tässä tutkielmassa tarkasteltiin transformeriarkkitehtuuriin perustuvia kielimalleja ja niiden koulutusstrategioita luonnollisen kielen käsittelyssä. Tutkimuksen päätavoitteina oli vastata kahteen tutkimuskysymykseen, TK1: Mitä teknologioita hyödynnetään kielimallien kouluttamisessa? ja TK2: Miten Crew AI toimii teknisesti? Tutkielma perustui kirjallisuuskatsaukseen, jonka aineisto koostui pääasiassa viime vuosina julkaistuista tieteellisistä artikkeleista ja tutkimusraporteista.

Ensimmäiseen tutkimuskysymykseen "Mitä teknologioita hyödynnetään kielimallien kouluttamisessa?"voidaan todeta, että kielimallien kouluttamisessa hyödynnetään useita teknologioita, jotka varmistavat mallien suorituskyvyn ja skaalautuvuuden. Neuroverkot, erityisesti transformerit, muodostavat kielimallien teknisen perustan. Transformereiden huomiomekanismi mahdollistaa merkityksellisten tekstinosien tehokkaan käsittelyn ja pitkien tekstiriippuvuuksien mallintamisen. Maskattu kielimallinnus ja tokenisointi ovat keskeisiä datan esikäsittelymenetelmiä, joiden avulla tekstisyöte muokataan neuroverkon käsittelemään muotoon.

Koulutusprosessien tehokkuutta parannetaan myös jakautuneen laskennan ja ydinjoukkojen avulla. Jakautunut laskenta mahdollistaa suurten aineistojen rinnakkaisen käsittelyn useilla palvelimilla, mikä nopeuttaa koulutusta ja tehostaa laskentaresurssien käyttöä. Ydinjoukot puolestaan tiivistävät laajoja aineistoja säilyttäen niiden olennaisimmat ominaisuudet, minkä ansiosta laskennalliset kustannukset pienenevät ilman merkittävää suorituskyvyn heikkenemistä.

Toiseen tutkimuskysymykseen "Miten Crew AI toimii teknisesti?"voidaan todeta, että Crew AI on tekoälyyn perustuva yhteistoiminnallinen järjestelmä, joka hyödyntää transformeriarkkitehtuuria monimutkaisten tehtävien suorittamisessa. Järjestelmän keskeinen piirre on sen roolipohjaiset tekoälyagentit. Jokaisella agentilla on selkeästi määritelty tehtävä, ja agenttien välinen yhteistyö mahdollistaa saumattoman tiimityöskentelyn.

Crew AI hyödyntää esikoulutettuja kielimalleja, jotka on optimoitu tekstin analysointiin, tiivistämiseen ja generointiin. Huomiomekanismit ja transformerikerrokset mahdollistavat syötteiden kontekstin ymmärtämisen ja riippuvuuksien mallintamisen. Teknisen toteutuksen ytimessä ovat jakautunut laskenta ja lokaali tietojenkäsittely, jotka parantavat järjestelmän suorituskykyä ja tietoturvaa. Lokaali asennettavuus antaa mahdollisuuden prosessoida tietoja ilman ulkoisia palveluntarjoajia.

Tulevaisuudessa transformereiden laajamittainen käyttö asettaa merkittäviä haasteita, jotka samalla avaavat uusia mahdollisuuksia syvälliselle jatkotutkimukselle. Ensinnäkin energiatehokkuuden parantaminen on keskeinen tutkimusaihe, sillä nykyiset mallit vaativat huomattavia laskentaresursseja. Optimoidut koulutusstrategiat, kuten ydinjoukkojen laajempi käyttö, voisivat merkittävästi vähentää energiankulutusta. Lisäksi mallien eettisiin näkökohtiin, kuten tietoturvaan ja väärinkäytön mahdollisuuksiin, tulisi kiinnittää lisähuomiota. Tulevaisuudessa olisi myös hyödyllistä tutkia, miten Crew AI:n roolipohjaisia agentteja voitaisiin mukauttaa entistä monimutkaisempiin tehtäviin ja eri toimialojen tarpeisiin.

Yhteenvetona voidaan todeta, että transformerit ja niiden pohjalta kehitetyt järjestelmät, kuten Crew AI, ovat mullistaneet luonnollisen kielen käsittelyn tarjoamalla tehokkaita ja skaalautuvia ratkaisuja monimutkaisiin ongelmiin. Vaikka haasteita on vielä ratkaistavana, transformereiden sovelluskenttä laajenee jatkuvasti.

Lähdeluettelo

[1] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention Is All You Need", 2017. url: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (viitattu 28. 10. 2024).

[2] Y. Chang, X. Wang, J. Wang et al., "A Survey on Evaluation of Large Language Models", en, *ACM Trans. Intell. Syst. Technol.*, vol. 15, nro 3, s. 1–45, kesäkuu 2024, ISSN: 2157-6904, 2157-6912. DOI: 10.1145/3641289. url: https://dl.acm.org/doi/10.1145/3641289 (viitattu 11. 11. 2024).

[3] N. Sachdeva, B. Coleman, W.-C. Kang et al., "How to Train Data-Efficient LLMs", en, helmikuu 2024. url: http://arxiv.org/abs/2402.09668 (viitattu 19. 10. 2024).

[4] Z. Ankner, C. Blakeney, K. Sreenivasan, M. Marion, M. L. Leavitt ja M. Paul, "Perplexed by Perplexity: Perplexity-Based Pruning with Small Reference Models", en, huhtikuu 2024. url: https://openreview.net/forum?id=0r0Bg1NY1X (viitattu 29. 11. 2024).

[5] N. Kriegeskorte ja T. Golan, "Neural Network Models and Deep Learning", en, vol. 29, s. 231–236, huhtikuu 2019. DOI: https://doi.org/10.1016/j.cub.2019.02.034. (viitattu 19. 12. 2024).

[6] K. T. Chitty-Venkata, S. Mittal, M. Emani, V. Viswanath ja A. K. Somani, "A Survey of Techniques for Optimizing Transformer Inference", en, vol. 29,

marraskuu 2023. DOI: https://doi.org/10.1016/j.sysarc.2023.102990. (viitattu 19. 12. 2024).

[7] *Introduction to Cloud TPU*, en. url: https://cloud.google.com/tpu/docs/intro-to-tpu (viitattu 22. 10. 2024).

[8] M. Li, "Comprehensive Review of Backpropagation Neural Networks", en, vol. 9, tammikuu 2024. DOI: https://doi.org/10.54097/51y16r47. (viitattu 11. 12. 2024).

[9] I. Lauriola, A. Lavelli ja F. Aiolli, "An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools", *Neurocomputing*, vol. 470, s. 443–456, tammikuu 2022, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.05.103. url: https://www.sciencedirect.com/science/article/pii/S0925231221010997 (viitattu 18. 11. 2024).

[10] K. Kalyan, A. Rajasekharan ja S. Sangeetha, *AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing*. elokuu 2021. DOI: 10.48550/arXiv.2108.05542. (viitattu 29. 11. 2024).

[11] M. Liao, "Taking ChatGPT as an example to analyze the main technologies used in large language models", en, *Science and Technology of Engineering, Chemistry and Environmental Protection*, vol. 1, nro 4, 2023, Number: 4, ISSN: 2960-1339. DOI: 10.61173/qecdqw17. url: https://www.deanfrancispress.com/index.php/te/article/view/311 (viitattu 11. 11. 2024).

[12] J. Devlin ja M.-W. Chang, *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing*, en, marraskuu 2018. url: http://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processing/ (viitattu 21. 10. 2024).

[13] J. Devlin, M.-W. Chang, K. Lee ja K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", lokakuu 2018. url: http://arxiv.org/abs/1810.04805 (viitattu 22. 10. 2024).

[14] B. Yan, Z. Zhong ja B. Bai, "A convolutional neural network-based proxy model for field production prediction and history matching", *Gas Science and Engineering*, vol. 122, helmikuu 2024, ISSN: 2949-9089. DOI: 10.1016/j.jgsce.2024.205219. url: https://www.sciencedirect.com/science/article/pii/S2949908924000153 (viitattu 18. 11. 2024).

[15] Z. Borsos, M. Mutn'y ja A. Krause, "Coresets via Bilevel Optimization for Continual Learning and Streaming", kesäkuu 2020. url: https://www.semanticscholar.org/paper/40a094a8afaa45454ab9c76c1d9933b2f9b1a0ff (viitattu 19. 11. 2024).

[16] *Crew AI*. url: https://crewai.net/ (viitattu 12. 12. 2024).

[17] *GitHub: Crew AI*. url: https://github.com/crewAIInc/crewAI (viitattu 20. 12. 2024).

[18] *Google Translate*. url: https://translate.google.fi/ (viitattu 23. 12. 2024).

[19] *DeepL*. url: https://www.deepl.com/ (viitattu 23. 12. 2024).