Tampereen yliopisto

Nea Peltola

VEKTORITIETOKANNAN HYÖDYNTÄMINEN KONEOPPIMISEN SOVELLUKSISSA

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Lokakuu 2024

TIIVISTELMÄ

Nea Peltola: Vektoritietokannan hyödyntäminen koneoppimisen sovelluksissa
Kandidaattitutkielma
Tampereen yliopisto
Tieto- ja sähkötekniikan koulutus
Lokakuu 2024

Vektoritietokantojen kehitys on kiihtynyt nopeasti viime vuosien aikana. Kehitys sai alkunsa koneoppimissovellusten kuten luonnollisen kielen käsittelyn, konenäön ja suosittelujärjestelmien yleistymisestä, jolloin huomattiin tarve vektorimuotoisen datan tehokkaaseen hallinnointiin. Lisäksi jäsentämättömän datan määrä on ollut nousussa jo pitkään, mikä on aiheuttanut haasteita monille perinteisempien tietokantatyyppien järjestelmille.

Vektoritietokannat ovat rakenteen ja varastointimenetelmien avulla laajasti skaalautuvia, mutta mahdollistavat myös indeksoinnin avulla nopean tiedon hakemisen. Tietokannasta hakeminen perustuu vektorien eli datan samanlaisuuteen. Tallennettava tieto vektoroidaan määrittelemällä sille ominaisuuksia, jolloin sen semanttisuus eli merkitys säilyy suhteessa muuhun dataan. Näin hakemisessa voidaan ottaa huomioon tiedon merkitys täydellisen hakuosuman sijasta.

Tutkielmassa vektoritietokannan käyttökohteiksi löydettiin semanttinen haku, luonnollisen kielen käsittely, laajat kielimallit, suosittelujärjestelmät, tunnistus sekä poikkeavuuksien havainnointi. Käyttöä sovelletaan useilla eri aloilla kuten terveydenhuollossa, teollisuuden tuotannossa, markkinoinnissa ja kyberturvallisuudessa. Useimmat käyttökohteet hyötyvät vektoritietokannoista erityisesti samankaltaisuushaun, semanttisen merkityksen säilyttämisen ja skaalautuvuuden vuoksi.

Avainsanat: Vektoritietokanta, tietokantajärjestelmä, vektorointi, koneoppiminen

SISÄLLYSLUETTELO

1 Johdanto

Tässä työssä esitellään vektoritietokantojen toiminnallisuutta, saatavilla olevia hallintajärjestelmiä, sekä käyttökohteiden mahdollisuuksia koneoppimisen sovelluksissa. Tietokantojen kehityksessä on kyetty vastaamaan alati kasvavan datamäärään, joka on varsinkin viime vuosina tekoälyn ja muiden koneoppimissovellusten myötä kasvanut entisestään. Määrän lisäksi myös datan luonne on muuttunut, esimerkiksi kuvia ja tekstiä halutaan tallentaa moniulotteisina objekteina, jotka sisältävät tietoa tallennettavan datan sisällöstä (Taipalus, 2024). Nämä moniulotteiset objektit kuvataan useimmiten vektoreina, jossa jokainen arvo kuvaa tiettyä ominaisuutta tallennettavassa tiedossa. Näiden vektoreiden varastoimiseen ja hallinnointiin on kehitetty erityinen tietorakenne, vektoritietokanta. Vektoritietokantojen ympärille on myös kehitetty vektoritietokantojen hallintajärjestelmiä, jotka sisältävät tietokannan lisäksi datan vektoroinnin sekä tiedon hakemisen. Vektoritietokantojen hallintajärjestelmät ovat kehitetty erityisesti moniulotteista tietoa sisältävien vektorien tehokkaaseen varastoimiseen ja tiedon hakemiseen (J. J. Pan ym., 2024).

Vektoritietokannat vastaavat nykyaikaisen datan haasteisiin, joihin perinteisemmät relaatiotietokannat eivät samalla mittakaavalla pysty (Han ym., 2023). Vektoritietokantoja hyödynnetäänkin erittäin paljon koneoppimisen ohjelmissa, sillä ne mahdollistavat esimerkiksi laskennallisesti tehokkaan samankaltaisuushaun myös suurista data-aineistoista. Tietokannat ovat hyvin skaalautuvia ja käsittelevät myös laajoja, jopa tuhansia ulottuvuuksia sisältäviä vektoreita.

Tässä työssä vastataan tutkimuskysymykseen: Miten vektoritietokannat toimivat ja miten niitä voidaan hyödyntää koneoppimisen sovelluksissa? Tutkimuskysymykseen pyritään vastaamaan esittelemällä vektoritietokantojen toimintaa yksinkertaisilla esimerkeillä ja kaavioita hyödyntämällä. Toimintaa vertaillaan myös yleisimmin tunnetuimpiin tietokantoihin, kuten relaatiotietokantoihin. Käyttökohteet käsitellään kategorioihin luokiteltuna, ja jokaiseen sisältyy esimerkkejä tehdyistä tutkimuksista ja vektoritietokantoja hyödyntävistä ohjelmista. Tutkielmassa esitellään myös nykyisiä vektoritietokantojen hallintajärjestelmiä, sekä niiden eroavaisuuksia.

Tutkielman menetelmänä on kirjallisuuskatsaus. Vaikka datan muuttaminen vektorimuotoon on ollut keskeistä monissa koneoppimisen ohjelmissa jo pitkään, on itse vektoritietokannat ja tietokantojen hallintajärjestelmät kehittynyt vasta viime vuosien aikana. Tutkimukset aiheesta on julkaistu 2010- ja 2020-luvuilla, ja suurin osa tässä työssä käytetyistä artikkeleista on 2020-luvulta. Tutkimuskysymyksen aiheeseen liittyy erityisen vahvasti Taipaluksen artikkeli (Taipalus, 2024), joka kokoaa yhteenvedon vektoritietokantojen hallintajärjestelmien toiminnasta sekä käytöstä. Yhteenvedon lisäksi lähteinä on

käytetty yksityiskohtaisempia tutkimuksia vektoritietokantojen toiminnasta sekä tutkimusartikkeleita ohjelmista, jotka hyödyntävät vektoritietokantaa.

Kirjallisuuskatsauksen tiedonhakuun käytettiin tiedonhakupalveluja ACM Digital Library, Andor sekä Google Scholar. Hakutermeinä käytettiin aluksi laajoja käsitteitä kuten ’vector database’ ja ’vector embedding database’, mutta niitä rajattiin tiedonhaun edetessä yksityiskohtaisempaan kuten ’inserting data to vector database’ sekä hyödynnettiin käyttökohteita, esimerkiksi hakemalla ’vector database’ AND ’natural language processing’. Lisäksi käytettiin aiheeseen liittyviä lyhenteitä kuten ’VDBMs’ ja ’LLM’. Tärkeä osa tiedonhakua oli myös löydettyjen artikkelien lähteiden hyödyntäminen. Uudempia julkaisuja löytyi myös artikkeleista, joissa löydettyyn artikkeliin on viitattu sen julkaisemisen jälkeen. Kaikista artikkeleista karsittiin paljon vanhoja ja epärelevantteja julkaisuja pois, mutta joitain vanhempia julkaisuja jätettiin esimerkkejä varten. Lisäksi tietoa etsittiin myös vektoritietokantojen hallintajärjestelmien omilta nettisivuilta sekä alaan liittyvistä blogi- ja uutisartikkeleista.

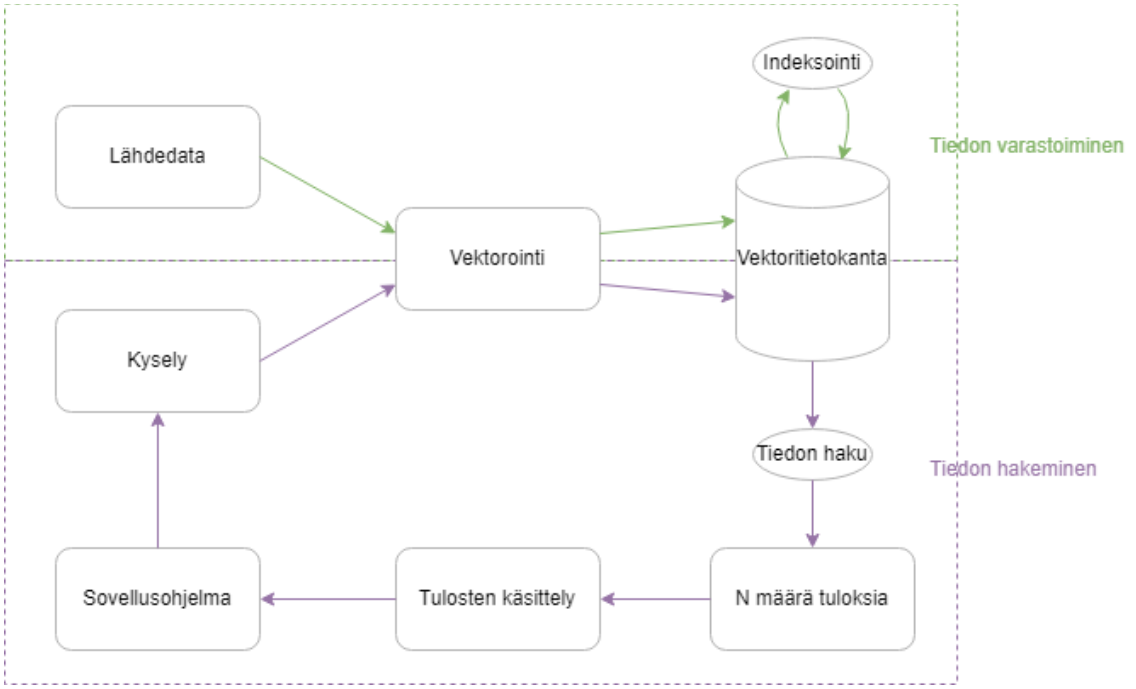
Tutkielman rakenne koostuu kuudesta luvusta. Luku 2 esittelee vektoritietokantojen rakennetta ja toimintaa. Luku 3 käsittelee vektoritietokantojen hallintajärjestelmiä, eli kokonaisuutta johon vektoritietokannat kuuluvat. Luku 4 esittelee käyttökohteita ja esimerkkejä niistä jaoteltuna kategorioihin. Luvussa 5 on keskustelua vektoritietokantojen haasteista ja tulevaisuudesta sekä luvussa 6 yhteenveto koko tutkielmasta.

2 Vektoritietokanta

Tietokanta on olennainen osa lähes jokaista nykyaikaista ohjelmistoa. Tietokantatyyppi valitaan useimmiten sen mukaan, millaista dataa halutaan tallentaa ja kuinka sitä käytetään. Relaatiotietokannat, kuten SQL, ja uudemmat NoSQL tietokannat ovat olleet vallitsevia vaihtoehtoja, sillä ne käsittelevät jäsenneltyä (engl. structured) dataa tehokkaasti esimerkiksi rivien ja sarakkeiden avulla (Zhang ym., 2014).

Viime vuosikymmenen aikana tallennettavan datan luonne on kuitenkin muuttunut entistä jäsentämättömäksi (engl. unstructured) (Wang ym., 2021). Jäsentämättömäksi dataksi kutsutaan tietoa, jonka hallinta, kuten järjestäminen ja kategorisointi vaatii datan semanttisen merkityksen ja kontekstin ymmärtämistä (Kukreja ym., 2023). Tällaista dataa voi olla esimerkiksi kuvat, tekstit, musiikkikappaleet tai lääketieteellinen data. Jäsentämättömän datan määrä on kasvanut räjähdysmäisesti älypuhelimien, IoT laitteiden ja tekoälyn kehityksen myötä, ja vuonna 2019 arvioitiin, että 80% kaikesta datasta on jäsentämätöntä vuoteen 2025 mennessä (Wang ym., 2021). Samanaikaisesti koneoppimisen alalla on ke-

hitetty vektorointifunktioita, joilla jäsentämätöntä dataa voidaan esittää ominaisuusvektoreina (engl. feature vector), eli datan ominaisuudet esitetään numeerisina arvoina vektorissa. Nämä ovat yhdessä johtaneet vektoritietokantojen ja -hallintajärjestelmien kehitykseen.



Kuva 1 Kaavioesitys vektoritietokantajärjestelmän rakenteesta ja tiedon kulusta

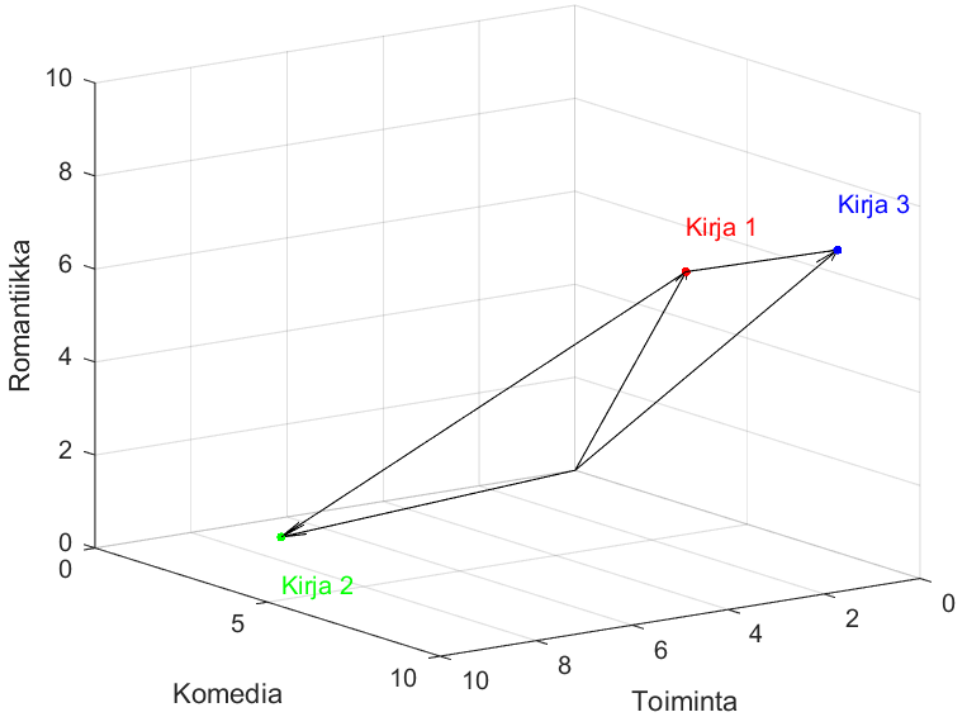
Vektoritietokanta varastoi ja hallinnoi dataa vektorimuodossa (Taipalus, 2024). Lähdedata tulee vektoroida eli muuttaa ominaisuusvektoreiksi, jotka indeksoidaan ja tallennetaan tietokantaan. Tiedon hakeminen tietokannasta tehdään kyselyn avulla ja se perustuu vektoreiden etäisyyksien tai samankaltaisuuksien tarkasteluun. Tämä kappale käsittelee vektoritietokannan perusperiaatteita eli datan vektorointia ja lisäämistä tietokantaan, rakennetta ja tiedon hakemista kyselyn avulla. Vektoritietokannan toiminta ja rakenne on havainnollistettu kuvan 1 kaaviolla.

2.1 Vektori

Vektoreita on hyödynnetty datarakenteissa pitkään. Käyttö yleistyi esimerkiksi paikkatietoaineistojen (engl. geospatial data), eli sijaintitietojen varastoimiseen paikkatietoanalyysejä varten (Liu ym., 2008). Vektoreihin tallennettiin esimerkiksi maantieteellisiä koordinaattipisteitä, ja näiden etäisyyksiä laskemalla löydettiin tietyn pisteen lähellä sijaitsevat muut pisteet (Taipalus, 2024). Koordinaattien esittäminen ja vertailu vektoreiden avulla on luonnollista, sillä kaksiulotteiset koordinaatit asettuvat kaksiulotteiseen vektoriavaruuteen kuten maantieteellisellä kartalla.

Vektoriavaruuden ulottuvuudet voidaan kuitenkin ajatella myös erilaisina ominaisuuksina, jolloin mitä tahansa dataa voidaan esittää vektorimuodossa (Kukreja ym., 2023). Vektoriesitys on siis yksinkertaistettuna datan ominaisuudet eli attribuutit muutettuna numeerisiksi arvoiksi, jotka yhdessä muodostavat vektorin ja sijoittavat datan vektoriavaruuteen. Tällaista vektoriesitystä kutsutaan ominaisuusvektoriksi (engl. feature vector), joka viittaa siihen, että vektorin numeeriset arvot ovat määritetty datan ominaisuuksien perusteella.

Esimerkiksi kirjaston kirjakokoelma voitaisiin vektoroida siten, että jokainen kirja olisi oma vektori. Yksinkertaisen ominaisuusvektorin luonnissa kirjat luokiteltaisiin kategorian perusteella, jolloin kategorioille annetaan numeerisia arvoja, esimerkiksi asteikolla 1-10. Kirja 1 voitaisiin luokitella olevan 2 toimintaa, 6 komediaa ja 5 romantiikkaa, jolloin vektori on muotoa [2,6,5]. Kirja 2 taas määritellään [9,4,1] ja kirja 3 [1,9,7], jolloin kirjat sijoittuvat vektoriavaruuteen kuvan 2 mukaisesti. Kirjojen luokittelua voisi tarkentaa lisäämällä vektoriin enemmän kategorioita tai muita tietoja kuten kohdeyleisö ja aihealueet. Jokainen eri ominaisuus kasvattaa vektorin ulottuvuutta eli dimensiota, ja paikka vektoriavaruudessa muuttuu.



Kuva 2 Kirjoista luodut ominaisuusvektorit vektoriavaruudessa.

Todellisuudessa datan ominaisuudet ja numeeriset arvot määritellään vektorointifunktiolla (engl. vector embedding) (Kukreja ym., 2023). Tavoitteena on luoda vektori, joka

kuvaa datan semanttisuutta eli merkitystä mahdollisimman hyvin, jotta sen sijainti vektoriavaruudessa on oikeassa suhteessa muuhun dataan. Esimerkiksi sanojen vektoroinnissa kontekstilla on hyvin suuri merkitys siihen, mitä sana tarkoittaa.

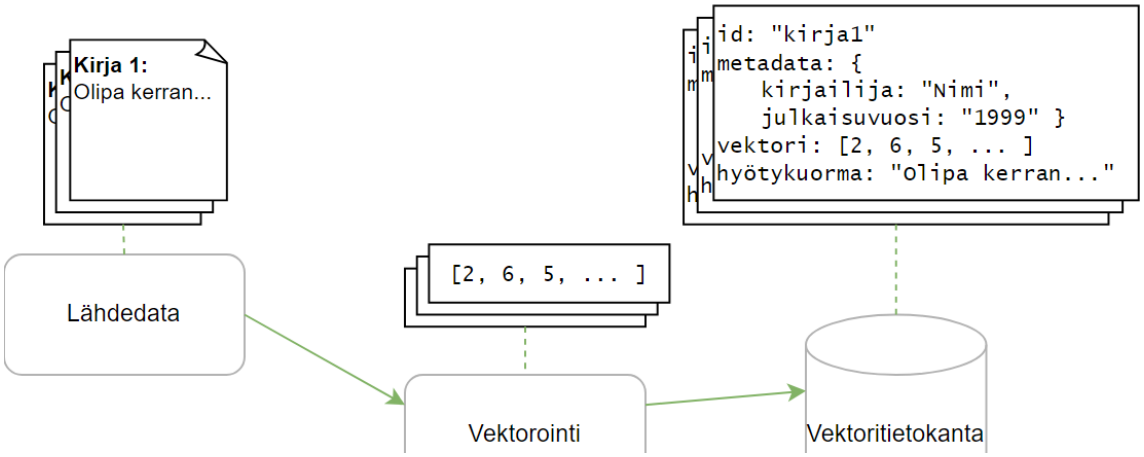
Vektorointifunktioon vaikuttaa tallennettavan datan tyyppi eli modaliteetti, minkä vuoksi vektorointifunktioiden toiminta perustuu moniin eri tekniikoihin (Kukreja ym., 2023). Sanojen ja dokumenttien vektoroinnin tunnettuja tekniikoita ovat esimerkiksi Word2Vec, GloVe ja Doc2Vec. Kuvan, videon ja äänen vektoroinnissa hyödynnetään muun muassa konvolutiivisia neuroverkkoja. Vektorointifunktio voidaan luoda myös täysin käyttötarkoituksen mukaan, esimerkiksi sivuston käyttäjästä vektori, jossa ominaisuuksina on erilaiset käyttäjämieltymykset, tai tuotevektori, jossa tuotteen ominaisuudet ovat määritelty suhteessa muihin tuotteisiin (Taipalus, 2024).

Yhteen vektoritietokantaan voidaan tallentaa myös useita eri modaliteetteja, eli esimerkiksi sekä kuvia että sanoja (Kukreja ym., 2023). Tällöin vektorointi tehdään multimodaalisesti, jossa keskitytään semanttisen yhteyden säilyttämiseen eri modaliteettien välillä. Multimodaalinen vektorointimalli on esimerkiksi Weaviaten kehittämä multi2vec-clip, joka mahdollistaa kuvien hakemisen sanojen avulla ja toisinpäin (Hasan, 2023).

Jäsentämättömän ja semanttisesti monipuolisen datan vektorointi kasvattaa vektorin ulottuvuuksien määrää nopeasti jopa tuhansiin. Vektoritietokantojen hallintajärjestelmät voivat kuitenkin parhaimmillaan tukea jopa yli 65 000 ulottuvuuden vektoreita (Jing ym., 2024).

2.2 Tiedon lisääminen ja rakenne

Valmiit ominaisuusvektorit varastoidaan tietokantaan. Vektorin kanssa tietokantaan tallennetaan usein lisätietoja, kuten yksilöllinen tunniste, metadataa sekä alkuperäinen data, jota vektori kuvaa. Kirjaesimerkissä metadataa voisi olla esimerkiksi kirjailijan nimi tai julkaisuvuosi. Metadata ei kuulu numeeriseen vektoriin, mutta sitä voidaan hyödyntää tiedon hakemisessa esimerkiksi hakutulosten rajaamiseen (Taipalus, 2024). Vektoriin voidaan liittää myös alkuperäinen data, kuten kirjan koko teksti, ja tätä kutsutaan vektorin hyötykuormaksi. Kirjallisuudessa koko tallennettavaa yksikköä kutsutaan yleensä pelkästään vektoriksi, tai vektorimonikoksi. Kuvassa 3 on havainnollistettu miltä kirjaesimerkin tallennettava yksikkö voisi näyttää. Metadata ja alkuperäinen data voidaan varastoida myös kokonaan erillään vektoritietokannasta, ja liittää datan mukaan vasta tiedon haun vaiheessa.



Kuva 3 Datan lisäämisen vaiheet vektoritietokannassa.

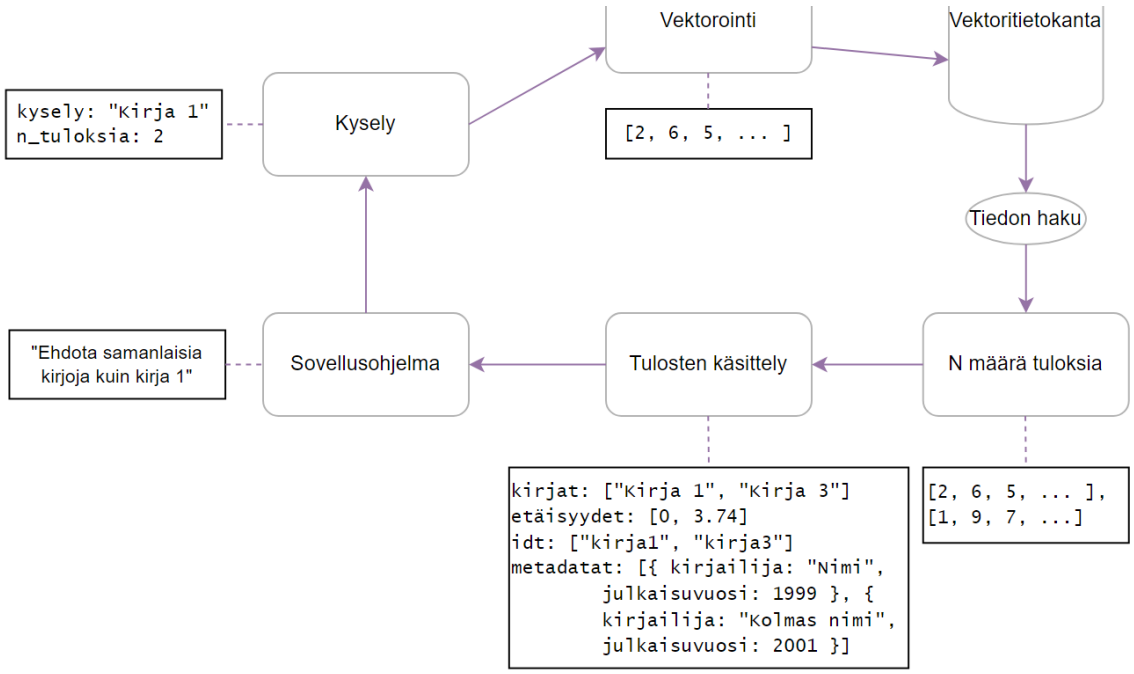
Suurten aineistojen varastoiminen aiheuttaa haasteita työmäärän jakamiseen. Vektoritietokannoissa hyödynnetään eri menetelmiä, joilla tietokantaa jaetaan osiin tai ryhmiin hallinnan parantamiseksi (Han ym., 2023). Esimerkiksi jaottelu (engl. partitioning) on menetelmä, joka kriteerin, kuten sijainnin tai kategorian perusteella jakaa tietokantaa pienempiin paloihin. Menetelmät parantavat esimerkiksi tietokannan skaalautuvuutta, suorituskykyä ja hakua.

Tietokantaan lisättävä data tulee myös indeksoida tiedon hakemisen tehostamiseksi. Vektoritietokantojen indeksointimenetelmiä on käsitelty tarkemmin kappaleessa 2.3.

Vektoritietokannan rakenne voidaan esittää vektoriavaruutena, joissa jokainen piste on oma tallennettu vektori eli datayksikkö (Taipalus, 2024). Vaikka vektorimuotoista dataa voitaisiin tallentaa myös esimerkiksi avain-arvo-tyyppiseen tietokantaan, se ei luo samanlaisia mahdollisuuksia samankaltaisuuksien hakuun, kuin mitä matemaattisia malleja hyödyntävä vektoritietokanta tarjoaa. Lisäksi vektoritietokannat pienentävät laskennallisen kuormaa, joka syntyy suurista aineistoista jäsentämätöntä dataa.

2.3 Tiedon hakeminen

Vektoritietokannasta haetaan tietoa kyselyn avulla. Kysely vektoroidaan datan kaltaisesti, eli sen tulee olla tietokannan tukemassa datatyypissä (Kukreja ym., 2023). Tätä varten kyselyä voidaan käsitellä ennen vektorointia. Kirjaesimerkissä luonnollisen kielen hakulause voisi olla ”Ehdota samanlaisia kirjoja kuin Kirja 1”, ja koska tietokantamme ei sisällä luonnollisen kielen lauseita, tulisi siitä algoritmien avulla erotella haluttu kirja kyselyksi. Samalla määritellään haluttu määrä tuloksia, eli kuinka monta samankaltaisinta kirjaa halutaan etsiä.



Kuva 4 Tiedon hakemisen vaiheet vektoritietokannassa.

Hakemisen tavoitteena on löytää kyselyvektoria lähimpänä sijaitsevat vektorit vektoriavaruudessa eli tietokannassa (Taipalus, 2024). Lähimmät vektorit vastaavat semanttiselta merkitykseltään eniten kyselyvektoria. Vektoritietokannasta hakemisessa etsitään siis semanttisia samankaltaisuuksia, eikä mahdollisimman tarkkaa yhtäläisyyttä kuten perinteisemmissä tietokantatyypeissä haetaan. Vektorien etäisyyttä mitataan esimerkiksi euklidisella etäisyydellä, kosinisamankaltaisuudella tai Jaccard-indeksillä.

Tietokannan kirjavektoreille voidaan nyt laskea etäisyys kyselyvektoriin. Tietokannasta halutaan löytää kategorisesti mahdollisimman samanlaisia kirjoja, eli pientä etäisyyttä. Euklidisella metriikalla laskettuna kirjan 1 ja 2 etäisyys on 8,31 ja kirjan 1 ja 3 etäisyys on 3,74. Kuvassa 4 on havainnollistettu datan muotoa haun eri vaiheissa. Koska tietokanta sisälsi myös kyselyvektorin, on sen etäisyys 0 ja tuloksissa ensimmäinen, mutta toisena suositellaan kirjaa 3.

Tietokantajärjestelmä ei kuitenkaan laske joka ikisen vektorin etäisyyttä kyselyvektoriin vaan vertailujen määrää minimoidaan indeksointimenetelmillä (J. J. Pan ym., 2024). Menetelmät nopeuttavat hakua jakamalla vektoriavaruutta alueisiin. Vektoritietokantojen indeksoinnin tulee sopia laajoihin aineistoihin ja korkean ulottuvuuden vektoreihin, joita on haastava lajitella, minkä vuoksi perinteiset indeksointimenetelmät eivät usein sovi. Vektoritietokantoja varten onkin kehitetty rakenteelle sopivammat indeksointimenetelmät (Han ym., 2023).

Kaikki indeksointimenetelmät perustuvat lähimmän naapurin hakuun (Nearest Neighbour Search) tai approksimoituun lähimmän naapurin hakuun (Approximate Nearest Neighbor Search), ja jakautuvat kolmeen kategoriaan: taulukot (tables), puut (trees) ja

graafit (graphs) (J. J. Pan ym., 2024). Paljon käytettyjä indeksointimenetelmiä ovat muun muassa Hierarchical Navigable Small World (HNSW), KD-tree, Product Quantization (PQ) ja Approximate Nearest Neighbor Oh Yeah (Annoy) (Han ym., 2023; Kukreja ym., 2023; Taipalus, 2024). Menetelmiä ja tietokannasta hakemista on esitelty yksityiskohtaisemmin artikkelissa ”Vector Database Management Techniques and Systems” (J. J. Pan ym., 2024). Indeksointimenetelmän valinnassa vaikuttavat haun toivottu tarkkuus, tehokkuus sekä suoritusnopeus. Usein vektoritietokantajärjestelmät tukevat monia eri indeksointimenetelmiä, joista erillinen optimointikomponentti valitsee käytön mukaan sopivimman (Taipalus, 2024).

3 Vektoritietokantojen hallintajärjestelmät

Vektoritietojen hallintajärjestelmät (engl. vector database management systems, VDBMS) ovat luotu erityisesti vektorimuotoisen datan hallinnointiin (Taipalus, 2024). Vektoritietokantojen kehitys on tapahtunut nopeasti muutaman viime vuoden aikana. Järjestelmiä alettiin kehittämään erityisesti vektoreita hyödyntävien koneoppimismallien kehityksen myötä, kun ilmeni tarve tehokkaaseen vektorien käsittelyyn sovelluksissa (P. N. Singh ym., 2023). Vektoritietokantojen hallintajärjestelmät mahdollistavat näille sovelluksille tarvittavan samankaltaisuuksien ja semanttisuuden ymmärtämisen. Samalla vektoritietokantojen hallintajärjestelemät ratkaiset datan räjähdysmäiseen kasvuun liittyviä haasteita, sillä tietokannat pystyvät hallitsemaan miljoonien moni-ulotteisten vektorien kokoisia aineistoja.

Useat perinteisempiin tietokantatyyppeihin keskittyvät järjestelmät ovat liittäneet ohjelmiinsa myös vektoritietokantojen tukemisen. Nämä lisäosat mahdollistavat vektoritietokantojen hallinnan, mutta optimointi ja ominaisuudet voivat olla puutteellisia (Taipalus, 2024). Tällaisia järjestelmiä ovat esimerkiksi Redis ja PostgreSQL. Lisäksi on luotu järjestelmiä vektori-indeksointiin, jotka mahdollistavat vektorihaun tietokannasta (J. J. Pan ym., 2024). Tällaisista järjestelmistä puuttuu kuitenkin usein esimerkiksi tiedonhallinta-ominaisuuksia, reaaliaikainen päivitys sekä integraatiomahdollisuudet. Esimerkiksi Faiss ja Elasticsearch ovat vektorihakujärjestelmiä.

Pelkästään vektoritietokantoihin suunnitellut hallintajärjestelmät käsittävät ohjelmasta riippuen yleensä ainakin datan vektoroinnin, varastoinnin ja siihen liittyvän indeksoinnin, hakukyselyn vektorisoinnin sekä tulosten palautuksen (Taipalus, 2024). Järjestelmiin on usein myös mahdollista yhdistää koneoppimisen kehysympäristöjä (engl. framework), kuten TensorFlow ja PyTorch. Vektoritietokannan valinta perustuu usein käyttökohteen

eli tallennettavan datan mukaan. Lisäksi valintaan voivat vaikuttaa indeksointimenetelmät, integrointimahdollisuudet, vektoriulottuvuuksien määrä sekä hinta (Superlinked, 2024).

Tällä hetkellä kirjallisuuden perusteella tunnetuimpia, ainoastaan vektoritietokantojen käsittelyyn keskittyviä hallintajärjestelmiä ovat Chroma, Vespa, Milvus, LanceDB, Qdrant, Weaviate, Pinecone ja Deep Lake. Järjestelmät ovat kaikki verrattain uusia, julkaistu muutaman viime vuoden aikana, minkä vuoksi käyttöaste ja suosio on vielä vähäistä verrattuna tietokantojen hallintajärjestelmiin yleisesti (DB-Engines, 2024). DB-engines -sivustolla on ajantasainen listaus saatavilla olevista järjestelmistä, ja niiden suosiosta. Avoimen lähdekoodin järjestelmiä ovat esimerkiksi Chroma, Weaviate, Qdrant ja Milvus (LangChain, 2024).

4 Käyttökohteet koneoppimisessa

Datan vektorointi sekä samankaltaisuuksien vertailu ovat monien koneoppimissovellusten ydintoimintaa. Esimerkiksi luonnollisen kielen käsittelyn tarkoituksena on vektoroida sanojen ja lauseiden merkitys, konenäössä vektoroidaan ja vertaillaan kuvia, ja suosittelujärjestelmissä etsitään samankaltaisuutta (Han ym., 2023). Vektoritietokannat ovatkin kehitetty erityisesti koneoppimissovelluksia varten hallinnoimaan vektorimuotoista dataa. Koneoppimismallien ja vektoritietokantojen yhdistelmää voidaan hyödyntää monissa käytännön sovelluksissa eri aloilla, sillä lähes mitä tahansa dataa voidaan muuttaa vektoriesityksiksi. Käyttökohteita löytyy niin biologian tutkimuksesta, teollisuuden tuotannosta, markkinoinnista kuin kyberturvallisuudesta.

Tässä kappaleessa on esitelty kirjallisuudesta löydettyjä vektoritietokantojen käyttökohteita koneoppimisen sovelluksissa. Käyttökohteet ovat jaoteltu käytetyn teknologian perusteella kategorioihin. Jaottelu ei ole ehdoton, sillä monet ohjelmistot yhdistävät useita teknologioita, esimerkiksi chatbotit voivat perustua sekä laajaan kielimalliin että luonnollisen kielen käsittelyyn. Esittely ei ole myöskään täysin kattava, sillä käyttökohteet ovat hyvin monipuolisia, mutta luo yleiskuvan siitä minkälaisiin käyttökohteisiin vektoritietokannat soveltuvat.

4.1 Semanttinen haku

Vektoritietokannan mahdollistamaa semanttista eli merkityksen huomioivaa hakua voidaan hyödyntää monilla eri aloilla ja erilaisilla data-aineistoilla. Perinteisten tietokantatyyppien samankaltaisuushaussa haetaan tuloksia, jotka vastaavat tarkasti kyselyä tai muita määriteltyjä kriteerejä, eikä niissä huomioida kyselyn semanttista merkitystä (Han ym., 2023). Semanttisen haun lisäksi vektoritietokantoja voidaan hyödyntää suurien data-aineistojen samankaltaisuushakuun, jolloin hakeminen vaatii tehokkuutta. Vektoritietokantaa hyödynnettiin esimerkiksi yleisen NFT (engl. non-fungible token) tietokannan luomiseen (Sahoo ym., 2023). Palvelun käyttäjät voivat etsiä tietokannasta samanlaisia NFT:eitä kuvan avulla, ja näin varmistetaan että NFT:n yksilöllisyys säilyy ja yhdenkin pikselin ”duplikaatit” tunnistetaan nopeasti. Tietokannan NFT:t muutetaan tiiviiksi 2016 ulotteiseksi vektoreiksi RegNetY-080 algoritmin avulla.

Vektoritietokantajärjestelmä Qdranttia hyödynnetään biologisen datan varastoimiseen ja samankaltaisuuksien hakuun luodussa PEPhub palvelussa (LeRoy ym., 2023). Palvelu keskittyy biologisen metadatan jakamiseen tiedeyhteisölle, ja sen aineisto kerättiin yli 100 000 biologisesta tutkimusprojektista. Data standardisoitiin taulukkomuotoon ja vektoroitiin käyttämällä lause-vektorointitekniikkaa. Palvelussa käyttäjä pystyy hakemaan tietokannasta luonnollisen kielen hakulauseella, joka vektoroidaan samalla lausetransformaatiolla, minkä jälkeen etsitään semanttisella haulla tietokannasta kyselyä vastaava biologinen data.

4.2 Luonnollisen kielen käsittely

Tallennettavan tiedon merkityksen säilytys korostuu erityisesti luonnollisen kielen käsittelyn (Natural Language Processing, NLP) sovelluksissa. Vektoroidut sanat ja lauseet ovat usein korkean ulottuvuuden dataa, jonka käsittelyyn vektoritietokannat tuovat parempaa suorituskykyä (Kukreja ym., 2023). Tietokannan avulla sanojen kieleen, kontekstiin ja kulttuurisiin eroihin liittyvät ominaisuudet pystytään mallintamaan ja tehokkaasti hyödyntämään.

Käyttökohteita on esimerkiksi kielen kääntämisessä, tekstin luokittelussa, mielipidelouhinnassa ja verkkosivustojen chatboteissa. Tekstin luokittelua voidaan hyödyntää esimerkiksi sosiaalisen median kommenttien tarkastuksessa, jossa kommentit jaetaan haitallisiin ja ei-haitallisiin (Weaviate, 2024). Chatbottien eli virtuaalisten asiakaspalvelijoiden käyttö on suosittua, ja niiden avulla voidaan parantaa käyttäjän asiakaspalvelukokemusta (Taipalus, 2024). Mielipidelouhinta (engl. sentiment analysis) voi myös parantaa chatbotin vastauksia luomalla käyttäjästä sentimentin ja muuttamalla vastauksia sen mukaan. Luonnollisen kielen käsittelyn alalla suurin kehitys on tapahtunut laajojen kielimallien

sovelluksissa, joista erityisesti kysymys-vastaus ohjelmien suosio on kasvanut (Jing ym., 2024).

4.3 Laajat kielimallit

Laajat kielimallit (Large Language Models, LLM) pystyvät käsittelemään, ymmärtämään ja luomaan tekstiä ihmisen kaltaisesti (Jing ym., 2024). Mallien kouluttamiseen tarvitaan suuria määriä dataa, jonka indeksointiin ja varastointiin voidaan yhtenä keinona hyödyntää vektoritietokantaa (Taipalus, 2024). Generatiivisiin malleihin, kuten kysymys-vastaus ohjelmiin, liittyy kuitenkin myös monia tunnettuja haasteita, kuten hallusinaatiot ja edellisten keskustelujen unohtaminen (Jing ym., 2024).

Hallusinaatiot generatiivisessa mallissa tarkoittavat, että ohjelma luo uskottavaa, mutta tosiasiassa virheellistä tai järjetöntä tietoa (Jing ym., 2024). Hallusinaatioden vähentämiseen on kehitetty Retrieval-Augmented Generation (RAG) teknologia, jossa käyttäjän antamalle syötteelle voidaan hakea kontekstia ulkoisesta vektoritietokannasta. Näin on tehty esimerkiksi Tangin ym. tutkimuksessa, jossa kehitettiin ChatSOS palvelua (Tang ym., 2024). ChatSOS on chatbot, joka laajaan kielimallin (GPT-3.5) lisäksi hyödyntää vastauksissaan erillistä vektoritietokantaa, joka on luotu 117 räjähdysonnettomuusraportin pohjalta. Tietokanta paransi selkeästi vastaustuloksia räjähdyksiin liittyvissä aiheissa verrattuna ainoastaan laajaa kielimallia käyttävään chatbottiin kuten ChatGPT.

Edellisten keskustelujen unohtaminen vaikuttaa ohjelman kontekstin ymmärtämiseen sekä käyttäjäkokemukseen. Useissa malleissa edellistä keskustelua voidaan ottaa tekstin generoinnissa huomioon vain rajoitettu määrä, minkä vuoksi pitkät ja monimutkaiset keskustelut aiheuttavat haasteita ohjelman toiminnalle (Taipalus, 2024). Vektoritietokantojen avulla voidaan poistaa nämä rajoitukset, käyttämällä tietokantaa ulkoisena pitkän aikavälin muistina. Siinä tietokanta sisältää kaikki edelliset kysely ja vastaus -yhdistelmät, ja sitä päivitetään aina jokaisesta uudesta parista. Käyttäjän lähettämä syöte vektoroidaan ja sillä haetaan ensin pitkän aikavälin vektoritietokannasta kaikista samankaltaisimmat keskustelut. Sen jälkeen alkuperäistä syötevektoria sekä sen samankaltaisimpia keskusteluvektoreita käytetään yhdessä kyselyvektoreina itse generatiiviseen malliin, josta vastaus luodaan. Tällä tavalla ohjelma muistaa keskustelut paremmin sekä voi myös metadatan avulla esimerkiksi aikaleimata keskustelua.

RAG:n käyttöä laajan kielimallin tekstin generoinnissa voi hyödyntää myös tiivistelmien tekemiseen. Saban ym. tutkimuksessa luotiin terveydenhuollon henkilökunnalle kysymys-vastaus ohjelma, joka luo potilasasiakirjoista yhteenvetoja (Saba ym., 2024). Tutkimuksessa potilasasiakirjat jaettiin kappaleisiin, jotka vektoroitiin ja tallennettiin vektoritietokantaan. Vektoritietokantajärjestelmänä käytettiin Chromaa. Palveluun esitetty ky-

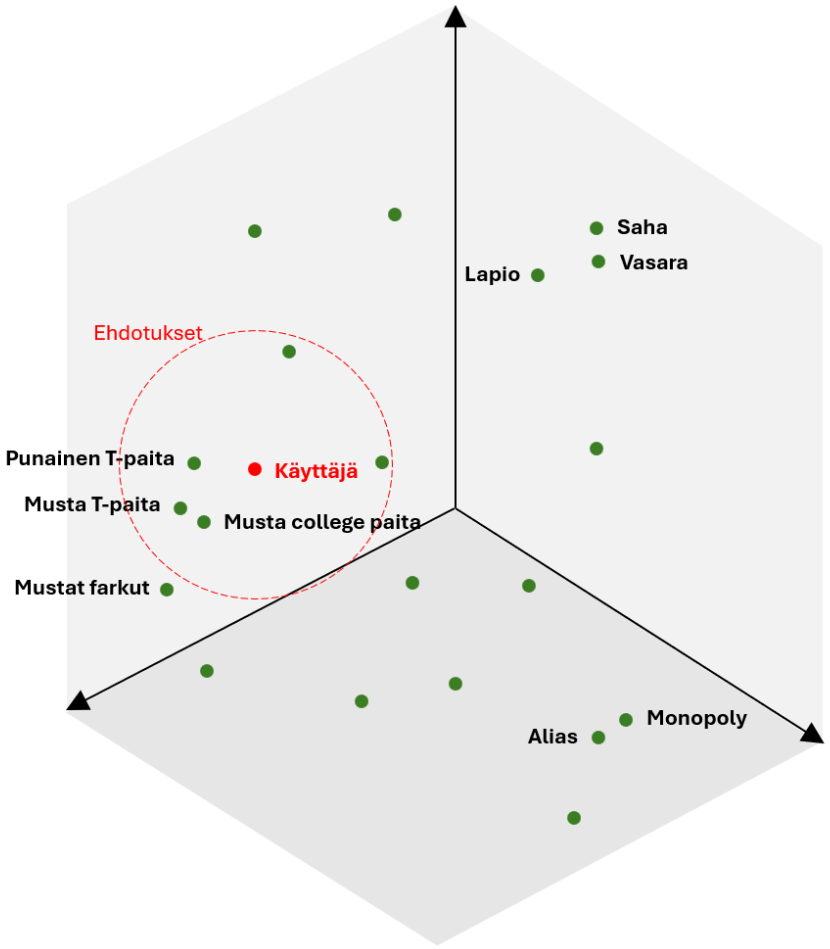
symys vektoroitiin ja tietokannasta etsittiin sitä vastaavat kappaleet, jotka palautettiin laajaan kielimalliin, joka loi yhtenäisen vastaustekstin ja viittauksen alkuperäiseen tekstiin. Yhteenvedon luomisessa voitiin ottaa myös huomioon kyselyn tekijän rooli potilaan hoidossa. Palvelu koettiin hyödyllisenä ja se voisi merkittävästi nopeuttaa terveydenhoitohenkilökunnan työtehtäviä.

4.4 Suosittelu ja personointi

Suosittelujärjestelmä on tärkeä osa useita ohjelmia ja sivustoja, kuten hakukoneita, sisällöntuottosivustoja ja verkkokauppoja (Grbovic & Cheng, 2018). Suosittelujärjestelmät hyötyvät vektoritietokannoista erityisesti nopean ja tarkan semanttisen haun vuoksi, mutta tietokannan laajuus ja päivitettävyys ovat myös tärkeitä etuja. Esimerkiksi verkkokaupan tuotteet voidaan vektoroida, ja suositella toisiaan samankaltaisia tuotteita. Weaviate on tehnyt kirjasuositteludemon, jossa suositellaan käyttäjän valitseman kirjan pohjalta samankaltaisia kirjoja (Weaviate, 2024).

Personoinnissa myös käyttäjä vektoroidaan, jolloin tuotteita ja käyttäjiä voidaan vertailla ja näin suositella käyttäjälle sopivia tuotteita (Kukreja ym., 2023). Verkkokaupat kuten IKEA ja VIPShop hyödyntävät personoiduissa tuotesuosituksissa Milvus vektoritietokantaa (Milvus, 2024). Lisäksi esimerkiksi pelejä, musiikkia ja elokuvia suositellaan usein käyttäjäpreferenssin ja aikaisemman kokemuksen avulla. Esimerkiksi Youtube käyttää vektoritietokantaa videoiden suositteluun käyttäjille (Covington ym., 2016). Suosittelua tehdään samanaikaisesti, kun uutta sisältöä ladataan tietokantaan jatkuvasti, ja lisäksi myös käyttäjän preferenssivektori muuttuu katsottujen videoiden pohjalta. Samankaltaisesti Airbnb hyödyntää kohteidensa ja käyttäjiensä vektorointia suosittelujärjestelmässään (Grbovic & Cheng, 2018). Vektorit luodaan käyttäjien klikkausten ja varausten tekemisen yhteisesiintymisen pohjalta, ja järjestelmä mahdollistaa reaaliaikaisen personoidun haun.

Personoidun suosittelujärjestelmän toimintaperiaate vektoritietokannan avulla on kuvattu kuvassa 5. Kuvan vektoriavaruuteen on sijoiteltu sekalaisia tuotteita myyvän verkkokaupan tuotteet. Käyttäjävektori luodaan esimerkiksi aikaisempien hakujen tai katseltujen tuotteiden perusteella, jolloin jokainen uusi toiminta verkkokaupassa päivittää vektorin ja sen sijainnin. Käyttäjän personoituihin tuotesuosituksiin voidaan ottaa kuvan kaltaisesti tietyllä etäisyydellä sijaitsevat tuotteet, tai vaihtoehtoisesti N määrä lähimpiä tuloksia.



Kuva 5 Personoidut tuotesuositukset vektoritietokannan avulla.

Personoidun suosittelujärjestelmän yhdistäminen laajaan kielimalliin (GPT-3.5) on tehty tutkimuksessa, jossa luotiin chatbot, joka suosittelee perulaisia ravintoloita käyttäjän mieltymyksien mukaan (Romero ym., 2023). Luotu chatbot käyttää suosittua LangChain kehystä, joka on integroitu useaan vektoritietokantajärjestelmään, kuten Chroma, Pinecone ja LanceDB.

Personointia käytetään myös mainostuksessa. Käyttäjävektoriin voidaan kerätä tietoa esimerkiksi selainhistoriasta, sosiaalisen median toiminnasta ja edellisistä ostoksista (L. Singh, 2023). Käyttäjistä voidaan muodostaa vektorien avulla käyttäytymismalleja ja -ryhmiä, joiden avulla voidaan esimerkiksi kohdentaa mainontaa.

4.5 Kuvan-, videon- ja puheentunnistus

Kuvan-, videon- ja audion vektorointi mahdollistaa vektoritietokantojen hyödyntämisen tunnistuksessa, luokittelussa ja samankaltaisuushaussa (Taipalus, 2024). Kuvantunnistamista voidaan käyttää esimerkiksi kasvojentunnistukseen autentikaatiota varten. Kuvien samankaltaisuushakua hyödynnetään monissa sovelluksissa, esimerkiksi aikaisemmin

mainituissa NFT-tietokannassa sekä ruokakuvien suosittelujärjestelmässä. Näiden lisäksi vektoritietokantajärjestelmä Milvuksen kehittäjät esittelevät artikkelissaan kahta kuvantunnistusohjelmistoa (Wang ym., 2021). Näistä ensimmäinen etsii kyselykuvaa vastaavia tavaramerkkejä asiakkailleen rekisteröinnin tarkastamista varten. Toinen hyödyntää tietokantaa samankaltaisten asuntojen etsintään pohjapiirustuskuvien perusteella.

Audion vektoroinnissa tallenne pilkotaan lyhyiksi osiksi, jotka normalisoidaan, suodatetaan ja muunnetaan vektoriksi, jolloin koko audio muodostuu vektorijoukoksi (Taipalus, 2024). Vektoritietokantaan tallennettuja audiovektorijoukkoja voidaan hyödyntää puheen- ja puhujantunnistuksessa. Tietokantaan voitaisiin tallentaa esimerkiksi äänitteitä puhutuista avainsanoista ja verrata kyselyäänitettä tietokantaan, jolloin sitä voi käyttää esimerkiksi autentikaatioon. Autentikaatio puheentunnistuksessa on esimerkki korkean toleranssin vaativasta samankaltaisuudesta, eli osuman tulee täsmätä erittäin hyvin toimiakseen oikein.

4.6 Poikkeavuuksien havainnointi

Poikkeavuuksien havainnointi vektoritietokannan avulla perustuu siihen, että tietokantaan tallennetaan normaaleja käyttäytymismalleja, jonka avulla epänormaali eli vähiten samankaltainen käytös huomataan nopeasti (P. N. Singh ym., 2023). Näitä epälineaarisia malleja voidaan tunnistaa esimerkiksi petoksen tunnistamiseen, kyberturvallisuudessa hyökkäyksen estoon sekä teollisuudessa laitteiden monitorointiin. Datan käsittelyssä vektoritietokannat mahdollistavat myös esimerkiksi kloonien ja duplikaattien tunnistamisen ja poistamisen (L. Singh, 2023).

RAGLog ohjelma luotiin poikkeuksien havainnointiin lokitiedoissa (J. Pan ym., 2023). RAGLog hyödyntää nimensä mukaisesti RAG teknologiaa ja vektoritietokantaa laajan kielimallin kanssa. Tietokantaan tallennetaan suuri aineisto normaaleja lokeja. Laajaan kielimalliin perustuvaan kysymys-vastaus -ohjelmaan annetaan lokimerkintä, johon tietokanta hakee samankaltaiset normaalit lokit ja tämän perusteella arvioidaan onko merkintä normaali vai ei. Poikkeavuuksien havainnointi monien muiden käyttökohteiden mukaisesti hyötyy erityisesti vektoritietokantojen skaalautuvuudesta ja nopeasta vektorihausta, jonka avulla poikkeavuuksiin voidaan reagoida nopeasti.

5 Keskustelu

Vektoritietokannat ovat tämänhetkisen tutkimuksen perusteella uusi ja mielenkiintoinen tietokantateknologia, joka vastaa moniin nykyisen datan hallinnoinnin ongelmiin. Vektoritietokannoissa on kuitenkin myös vielä omat haasteensa, joihin tutkimusta ja kehitystyötä tulee kohdentaa. Yksi haaste vektoritietokannoissa on haun tarkkuus, erityisesti kun niitä käytetään päätöksenteon tukena. Tietokannan antama tieto on oltava luotettavaa, jotta vältetään väärän tiedon aiheuttamat virheet ja ongelmat. Tämä on erityisen tärkeää esimerkiksi poikkeavuuksien havainnoinnissa, autentikaatiossa tai ympäristön tunnistuksessa. Lisäksi vektoritietokantojen laajempaa käyttöä rajoittaa tässä vaiheessa kehitystä saatavuus sekä hinta. Vektoritietokantateknologialle on kuitenkin kiinnostusta monilla aloilla, joka edistää käytön yleistymistä.

Tutkielma esittelee vektoritietokantojen käyttökohteita monipuolisesti. Aiheen tuoreus on tuonut haasteita tiedonhakuun, sillä artikkeleja julkaistiin lisää jopa kesken kirjoitusprosessin. Vektoritietokantojen toiminnasta on tehty kattavia katsauksia, mutta käyttökohteita käsitellään usein vain hyvin pintapuolisesti. Ohjelmakohtaisia tutkimusartikkeleita on julkaistu jonkun verran, mutta kaupallista käyttöä on paljon myös tutkimuksen ulkopuolella. Tutkielmassa löydetyt kategoriat käyttökohteille ovat kuitenkin saaneet paljon mainintoja kirjallisuudessa, minkä lisäksi löydetyt ohjelmakohtaiset käyttökohteet tukevat tätä jaottelua. Tämän vuoksi tuloksen voidaan olettaa olevan yleistettävissä.

Tutkielman tulosta voivat hyödyntää yritykset tai muut aiheesta kiinnostuneet. Tutkielma tarjoaa lukijalle yleiskuvan vektoritietokantojen toiminnasta ja mahdollisuuksista. Sen seurauksena teknologian tarjoamia etuja voitaisiin hyödyntää entistä tehokkaammin ja laajemmin.

Kirjallisuuden perusteella tulevaisuudessa vektoritietokantojen hallintajärjestelmien ominaisuuksia monipuolistetaan, käyttö yleistyy ja mahdollisia käyttökohteita löydetään myös lisää. Kehitystä varten olisi kuitenkin tärkeää saada lisää etenkin vertailevaa tutkimusta, jossa vektoritietokantoja verrataan perinteisempiin tietokantojärjestelmiin tai toisiin vektoritietokantajärjestelmiin. Tutkimuksen avulla voidaan arvioida minkälaiset käyttökohteet hyötyvät vektoritietokannoista sekä mitä ominaisuuksia järjestelmissä tulisi kehittää.

6 Yhteenveto

Tutkielmassa käsiteltiin vektoritietokantojen toiminnallisuutta sekä käyttömahdollisuuksia koneoppimisen sovelluksissa. Vektoritietokantoja ja niiden hallintajärjestelmiä on kehitetty muutaman viime vuoden aikana nopeasti. Kehityksen motivaationa on ollut jäsentämättömän datan määrän kasvu sekä koneoppimissovellusten kehitys ja yleistyminen. Vektoritietokannat mahdollistavat laajojen aineistojen varastoimisen ja hakemisen tehokkaasti. Lisäksi datan tallentaminen vektorimuodossa mahdollistaa tietokannan vektorihaun, jota voidaan hyödyntää samankaltaisuushaussa.

Vektoritietokannassa tallennettava tieto esitetään sen ominaisuuksia kuvaavana vektorina. Datan vektoroinnissa, omaisuuksille määritellään numeerisia arvoja suhteessa muuhun dataan, ja vektorin ulottuvuus voi kasvaa jopa tuhansiin. Mitä tahansa dataa voidaan vektoroida, mutta vektorointifunktioita on kehitetty erityisesti sanojen, dokumenttien, kuvien ja videoiden vektorointiin.

Vektorit muodostavat tietokannassa vektoriavaruuden, jossa tietoa voidaan vertailla mittaamalla etäisyyksiä toisiin vektoreihin. Tietokannasta hakeminen perustuukin samankaltaisuushakuun, jossa kyselyvektoria vertaillaan tietokannan vektoreihin. Hakua tehostetaan indeksointimenetelmillä, jotka vähentävät tehtyjen vertailujen määrää.

Vektoritietokantojen hallintajärjestelmät varastoivat, indeksoivat ja hakevat tietoa vektoritietokannasta. Useat perinteiset tietokantojen hallintajärjestelmät ovat lisänneet vektoridatan käsittelemisen tai vektorihaun järjestelmiinsä, mutta viime vuosien aikana on myös julkaistu täysin vektoridatan hallinointiin keskittyviä tietokantajärjestelmiä. Suosituimpia järjestelmiä ovat esimerkiksi Chroma, Pinecone, Milvus ja Weaviate.

Vektoritietokantojen hyöty verrattuna perinteisiin tietokantatyyppeihin tulee erityisesti tiedon merkityksen säilymisestä, samankaltaisuushausta ja laajasta skaalautuvuudesta. Näitä ominaisuuksia voidaan hyödyntää useissa koneoppimisen sovelluksissa, kuten semanttisessa haussa, luonnollisen kielen käsittelyssä, laajoissa kielimalleissa, suosittelujärjestelmissä, kuvantunnistuksessa ja poikkeavuuksien havainnoinnissa.

Koneoppimissovellusten yleistymisen ja datamäärien kasvun myötä vektoritietokantoja voidaan hyödyntää lukuisilla eri aloilla ja erilaisissa käyttökohteissa. Käyttöä rajoittavat ensisijaisesti saatavuus ja hinta, mutta myös teknologian uutuus, tutkimusten vähäinen määrä ja toiminnalliset haasteet. Vektoritietokantojen nopea kehitys ja käyttökohteiden monipuolisuus lupaavat kuitenkin käytön laajempaa yleistymistä tulevien vuosien aikana.

Lähdeluettelo

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198. https://doi.org/10.1145/2959100.2959190

DB-Engines. (2024). *DB-Engines Ranking*. DB-Engines. Noudettu 18. kesäkuuta 2024, osoitteesta https://db-engines.com/en/ranking/vector+dbms

Grbovic, M., & Cheng, H. (2018). Real-time Personalization using Embeddings for Search Ranking at Airbnb. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 311–320. https://doi.org/10.1145/3219819.3219885

Han, Y., Liu, C., & Wang, P. (2023). *A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge* (arXiv:2310.11703). arXiv. http://arxiv.org/abs/2310.11703

Hasan, Z. (2023, kesäkuuta 27). *Multimodal Embedding Models*. https://weaviate.io/blog/multimodal-models

Jing, Z., Su, Y., Han, Y., Yuan, B., Xu, H., Liu, C., Chen, K., & Zhang, M. (2024). *When Large Language Models Meet Vector Databases: A Survey* (arXiv:2402.01763). arXiv. http://arxiv.org/abs/2402.01763

Kukreja, S., Kumar, T., Bharate, V., Purohit, A., Dasgupta, A., & Guha, D. (2023). Vector Databases and Vector Embeddings-Review. *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*, 231–236. https://doi.org/10.1109/IWAIIP58158.2023.10462847

LangChain. (2024, huhtikuuta 6). *Top 5 Open Source Vector Databases in 2024*. LLM & Langchain Blogs. https://www.langchain.ca/blog/top-5-open-source-vector-databases-2024/

LeRoy, N. J., Khoroshevskyi, O., O'Brien, A., Stepień, R., Arslan, A., & Sheffield, N. C. (2023). *PEPhub: A database, web interface, and API for editing, sharing, and validating biological sample metadata*. https://doi.org/10.1101/2023.08.15.551388

Liu, X., Zhang, S., He, J., Han, J., Han, C., & Xu, Z. (2008). VDMS: A Vector Data Management System for GIS Applications. *2008 International Conference on Networking, Architecture, and Storage*, 278–284. https://doi.org/10.1109/NAS.2008.58

Milvus. (2024). *Use cases | Milvus*. Noudettu 18. kesäkuuta 2024, osoitteesta https://milvus.io/use-cases

Pan, J. J., Wang, J., & Li, G. (2024). Vector Database Management Techniques and Systemss. *Companion of the 2024 International Conference on Management of Data*, 597–604. https://doi.org/10.1145/3626246.3654691

Pan, J., Wong, S. L., & Yuan, Y. (2023). *RAGLog: Log Anomaly Detection using Retrieval Augmented Generation* (arXiv:2311.05261). arXiv.

Romero, F. C., Pajes León, S., & Wong, L. (2023). Approach for Personalized Recommendations to Enhance Customer Service Process in Peruvian Restaurants using OpenAI Contextual Chatbot. *2023 IEEE XXX International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 1–8. https://doi.org/10.1109/INTERCON59652.2023.10326059

Saba, W., Wendelken, S., & Shanahan, J. (2024). *Question-Answering Based Summarization of Electronic Health Records using Retrieval Augmented Generation* (arXiv:2401.01469). arXiv.

Sahoo, S., Paul, N., Shah, A., Hornback, A., & Chava, S. (2023). *The Universal NFT Vector Database: A Scaleable Vector Database for NFT Similarity Matching* (arXiv:2303.12998). arXiv. http://arxiv.org/abs/2303.12998

Singh, L. (2023, elokuuta 31). Top 10 Types of Vector Databases & Libraries [2024 Guide]. *Redblink*. https://redblink.com/vector-databases/

Singh, P. N., Talasila, S., & Banakar, S. V. (2023). Analyzing Embedding Models for Embedding Vectors in Vector Databases. *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, 1–7. https://doi.org/10.1109/ICTBIG59752.2023.10455990

Superlinked. (2024). *Vector DB Comparison*. Noudettu 18. kesäkuuta 2024, osoitteesta https://superlinked.com/vector-db-comparison

Taipalus, T. (2024). Vector database management systems: Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*, *85*, 101216. https://doi.org/10.1016/j.cogsys.2024.101216

Tang, H., Chen, D., & Chu, Q. (2024). *ChatSOS: Vector Database Augmented Generative Question Answering Assistant in Safety Engineering* (arXiv:2405.06699). arXiv.

Wang, J., Yi, X., Guo, R., Jin, H., Xu, P., Li, S., Wang, X., Guo, X., Li, C., Xu, X., Yu, K., Yuan, Y., Zou, Y., Long, J., Cai, Y., Li, Z., Zhang, Z., Mo, Y., Gu, J., … Xie, C. (2021). Milvus: A Purpose-Built Vector Data Management System. *Proceedings of the 2021 International Conference on Management of Data*, 2614–2627. https://doi.org/10.1145/3448016.3457550

Weaviate. (2024). *Example use cases and demos | Weaviate—Vector Database*. Noudettu 18. kesäkuuta 2024, osoitteesta https://weaviate.io/developers/weaviate/more-resources/example-use-cases

Zhang, N., Zheng, G., Chen, H., Chen, J., & Chen, X. (2014). HBaseSpatial: A Scalable Spatial Data Storage Based on HBase. *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 644–651. https://doi.org/10.1109/TrustCom.2014.83