

Eetu Niemelä

# RAG-JÄRJESTELMIEN AJALLISEN YHTENEVÄISYYDEN JA TIETOKONFLIKTIEN HAASTEET

Kandidaatintutkielma  
Informaatioteknologian ja viestinnän tiedekunta  
Toukokuu 2025

# TIIVISTELMÄ

Eetu Niemelä: RAG-järjestelmien ajallisen yhteneväisyyden ja tietokonfliktien haasteet  
Kandidaatintutkielma  
Tampereen yliopisto  
Tieto- ja sähkötekniikan kandidaattiohjelma  
Toukokuu 2025

---

Tämä kandidaatintutkielma tarkastelee RAG-järjestelmien kohtaamia ajallisia haasteita ja tietokonflikteja, jotka vaikuttavat näiden järjestelmien luotettavuuteen ja käytettävyyteen. Hakutehostettua generaatiota käyttävät järjestelmät (Retrieval-Augmented Generation, RAG) ovat nousseet keskeiseksi menetelmäksi modernien tekoälyratkaisujen kehittämisessä, yhdistäen suurten kielimallien generointikyvyt ulkoiseen tiedonhakuun.

Tutkielmassa kartoitetaan kirjallisuuskatsauksen menetelmin RAG-järjestelmissä ilmeneviä tietokonfliktityyppejä: konteksti-muistikonflikteja, kontekstin välisiä konflikteja ja muistin välisiä konflikteja. Lisäksi analysoidaan ajallisia haasteita, kuten ajallista sovituvirhettä, tietolähteiden epätasapainoa ja huomiovuonoutta aikariippuvaisissa kyselyissä. Erityisesti dynaamiset, nopeasti muuttuvat faktat ovat osoittautuneet haasteellisiksi RAG-järjestelmille.

Tutkielmassa esitellään ja vertaillaan teknisiä ratkaisumenetelmiä, kuten FLARE, Iter-RetGen, CD2 ja kontrastiivinen oppiminen, sekä tarkastellaan niiden vahvuuksia ja heikkouksia eri käyttötilanteissa. Kontrastiivinen oppiminen on osoittautunut erityisen tehokkaaksi aikariippuvaisten kyselyjen käsittelyssä, kun taas CD2-menetelmä tarjoaa ratkaisuja kielimallien harhojen vähentämiseen. Tutkimuksen perusteella optimaalinen ratkaisu RAG-järjestelmien ajallisiin haasteisiin saattaa löytyä eri lähestymistapojen vahvuuksien yhdistämisestä. Jatkotutkimuksissa voisi keskittyä kehittämään hybridimenetelmiä, jotka yhdistävät aikamääritteiden tarkan tunnistamisen ja luotettavan päätöksenteon ristiriitaisissa tilanteissa.

Avainsanat: hakutehostettu generaatio, tiedonhaku, kielimallit, RAG, RALM, tietokonfliktit, ajallinen yhteneväisyys

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin Originality Check -ohjelmalla.

# TEKOÄLYN KÄYTTÖ OPINNÄYTTEESSÄ

Opinnäytteessäni on käytetty tekoälysovelluksia:

- Ei
- Kyllä

Ilmoitukseni mukaan olen käyttänyt opinnäytteessäni tutkielmaproessin aikana seuraavia tekoälysovelluksia: Scopus AI, Perplexity, Claude Sonnet

Tekoälysovellusten nimet ja versiot: Scopus AI, Perplexity, Claude 3.5 Sonnet, Claude 3.7 Sonnet

Käyttötarkoitus: Scopus AI- ja Perplexity -työkaluja käytettiin aineiston haussa. Claude Sonnet -mallia käytettiin alustavan tutkimuskysymyksen laatimisessa, tutkimuksen rakenteen suunnittelussa, ajatusten jäsentelyssä, lähdeaineiston ymmärtämisessä ja tiivistämisessä, johdonmukaisten ilmaisujen muotoilussa, lähdetekstin kääntämisessä ja tekstin kielen parantelussa.

Osiot, joissa tekoälyä on käytetty: Scopus AI:ta ja Perplexityä käytettiin aineiston haussa, joka on esitelty luvussa 2. Claude Sonnetia käytettiin koko tutkielman laajuisesti.

Olen tietoinen siitä, että olen täysin vastuussa koko opinnäytteeni sisällöstä, mukaan lukien osat, joissa on hyödynnetty tekoälyä, ja hyväksyn vastuun mahdollisista eettisten ohjeiden rikkomuksista.

# SISÄLLYSLUETTELO

1. JOHDANTO.....	1
2. AINEISTON KERUU .....	2
3. RAG-JÄRJESTELMIEN PERUSTEET JA TOIMINTAPERIAATTEET .....	3
3.1 Kielimallit RAG-järjestelmien pohjana .....	3
3.2 Hakutehostetun generaation periaatteet ja toimintaprosessi.....	4
3.3 RAG-järjestelmien rajoitukset ja tehostamismenetelmät .....	4
3.4 RAG-järjestelmien arviointi ja mittaaminen .....	5
4. TIETOLÄHTEIDEN KONFLIKTIT RAG-JÄRJESTELMISSÄ .....	7
4.1 Tiedon tyypit ja konfliktit.....	7
4.2 RALM-mallien tietolähdepreferenssien vinoumat .....	8
5. AJALLISET HAASTEET .....	10
5.1 Tiedon ajalliset muutokset ja ristiriidat.....	10
5.2 Mallien ajallisen päättelyn haasteet.....	10
6. RATKAISUMENETELMÄT .....	12
6.1 Tekniset ratkaisut.....	12
6.1.1 FLARE: Ennakoiva tiedonhaku.....	12
6.1.2 Iter-RetGen: Iteratiivinen tiedonhaku.....	13
6.1.3 Kontrastiivinen oppiminen.....	15
6.1.4 CD2: Luottamustason kalibrointi.....	15
6.2 Metodologiset ratkaisut.....	17
7. ARVIOINTI JA ANALYYSI .....	19
7.1 Teknisten ratkaisumenetelmien vertailu ja tehokkuus .....	19
7.2 Käytännön haasteet ja rajoitukset .....	20
7.3 Tulevaisuuden tutkimussuunnat.....	21
8. YHTEENVETO .....	22
LÄHTEET .....	23

## **LYHENTEET JA MERKINNÄT**

CD2: Conflict-Disentangle Contrastive Decoding

FLARE: Forward-Looking Active Retrieval-Augmented Generation

GNN: Graph Neural Network

HyDE: Hypothetical Document Embeddings

LLM: Large Language Model

LM: Language Model

RAG: Retrieval-Augmented Generation

RALM: Retrieval-Augmented Language Model

RLHF: Reinforcement Learning from Human Feedback

# 1. JOHDANTO

Hakutehostettua generaatiota käyttävät järjestelmät (Retrieval-Augmented Generation, RAG) [1] ovat nousseet keskeiseen asemaan modernien tekoälyratkaisujen kehittämisessä. Nämä järjestelmät yhdistävät suurten kielimallien kyvyn tuottaa tekstiä tietokantahakuihin ja dokumenttien hakemiseen, mikä mahdollistaa ajantasaiseen ja tarkasti määriteltyyn tietopohjaan perustuvan vastausten generoinnin. Vastausten luotettavuus on kriittistä käytännön sovellusten kannalta, ja aihe on erityisen ajankohtainen tekoälyn yleistyessä eri toimialoilla. Tässä työssä kartoitetaan RAG-järjestelmissä esiintyviä tietokonflikteja ja ajallisen yhteneväisyyden haasteita sekä esitellään erilaisia ratkaisumalleja näiden ongelmien tunnistamiseen ja ratkaisemiseen.

Erityisesti ajallinen ulottuvuus tuottaa merkittäviä haasteita RAG-järjestelmille [2, 3]. Tiedon muuttuessa ajan myötä järjestelmän on pystyttävä tunnistamaan, mikä tieto on ajantasaista ja olennaista kussakin kontekstissa. Ongelma monimutkaistuu entisestään, kun otetaan huomioon kielimallien oma sisäinen, koulutusvaiheessa omaksuttu parametri-nen tieto, joka voi olla ristiriidassa haetun tiedon kanssa [3]. Tämä johtaa erilaisiin tietokonflikteihin, joiden ratkaiseminen on keskeistä RAG-järjestelmien luotettavuuden kannalta.

Tutkielmassa vastataan tutkimuskysymykseen ”Miten voidaan tunnistaa ja ratkaista RAG-järjestelmien tietokonfliktit ja ajalliset haasteet?”. Tutkielman tavoitteena on tarjota jäsennelty katsaus ajankohtaiseen tutkimukseen RAG-järjestelmien tietokonfliktien ja ajallisten haasteiden alueella. Ratkaisumenetelmiä käsitellään yleistajuisesti ilman syvällistä matemaattista analyysiä, mikä auttaa hahmottamaan keskeisiä konsepteja ja niiden merkitystä käytännön sovellusten kannalta. Luvussa 3 käsitellään RAG-järjestelmien perusteita ja toimintaperiaatteita. Luvussa 4 keskitytään tietolähteiden konflikteihin RAG-järjestelmissä, ja luvussa 5 perehdytään ajallisiin haasteisiin. Luvussa 6 käsitellään kirjallisuudessa esiintyneitä teknisiä ja metodologisia ratkaisumenetelmiä näihin tietokonflikteihin. Luvussa 7 vertaillaan kirjallisuudesta löytyneitä ratkaisumenetelmiä ja arvioidaan niiden vahvuuksia sekä heikkouksia.

## 2. AINEISTON KERUU

Tämä kandidaatintutkielma toteutettiin kirjallisuuskatsauksena. Aineistona käytettiin konferenssiartikkeleita, aikakauslehtiartikkeleita ja preprint-julkaisuja. Katsaus keskittyy vuodesta 2020 eteenpäin julkaistuihin tutkimuksiin, koska RAG-menetelmät ovat melko tuore tutkimusalue tekoälyn ja kielimallien kontekstissa. Ensimmäinen maininta termistä on Lewis et al. vuoden 2020 tutkimuksessa [1]. Vaikka tutkimuksen rajaus vuodesta 2020 eteenpäin on perusteltua RAG-järjestelmien tuoreuden vuoksi, saattaa rajaus toisaalta jättää huomiotta relevantteja menetelmiä ja lähestymistapoja aiemmasta tiedonhaku-tutkimuksesta, jotka voisivat tarjota arvokkaita näkökulmia nykyisten haasteiden ratkaisu-miseen. Lisäksi aiheen tuoreudesta johtuen osa käytetyistä lähteistä on preprint-versi-oita, joita ei ole vielä ehditty vertaisarvioimaan. Aineiston haussa käytettiin Scopus AI-, Google Scholar- ja Perplexity –palveluita.

Hakuprosessissa käytettiin seuraavia hakusanoja ja niiden yhdistelmiä: "retrieval-aug-mented generation", "information retrieval", "data retrieval", "time conflict", "temporal conflict", "solution", "resolution" ja "strategy". Näiden hakujen tuloksena syntyneitä viite-luetteloita käytettiin iteratiivisesti uusien relevanttien lähteiden tunnistamiseen. Tutkiel-massa käsiteltävien teknisten ratkaisujen valintaperusteita on kuvattu tarkemmin luvussa 6.1.

## 3. RAG-JÄRJESTELMIEN PERUSTEET JA TOIMINTAPERIAATTEET

### 3.1 Kielimallit RAG-järjestelmien pohjana

Kielimallit (Language Model, LM) ovat laskennallisia malleja, jotka ennustavat sanasekvenssien todennäköisyyksiä ja tuottavat tekstiä annetun syötteen perusteella. Suuret kielimallit (Large Language Model, LLM) kehittävät tätä konseptia pidemmälle hyödyntäen valtavaa määrää parametreja ja Transformer [4] -arkkitehtuurin itsehavainnointimekanismeja (self-attention). Tämä arkkitehtuuri on mullistanut luonnollisen kielen käsittelyn mahdollistamalla tehokkaan rinnakkaistamisen ja pitkien riippuvuussuhteiden tunnistamisen tekstissä. [5]

Suurten kielimallien keskeinen ominaisuus on kontekstissa oppiminen (in-context learning), joka mahdollistaa koherenttien ja kontekstuaalisesti relevanttien vastausten tuottamisen annetun kontekstin tai kehoitteen perusteella. Tätä peruskykyä täydentää ihmispalautteeseen perustuva vahvistusoppiminen (Reinforcement Learning from Human Feedback, RLHF), joka hienosäätää mallia ihmisen tuottamien vastausten avulla, mikä tehostaa mallin suorituskykyä ja soveltuvuutta erilaisiin käyttötarkoituksiin. [5]

Vaikka suuret kielimallit osoittavat huomattavaa kyvykkyyttä tekstin tuottamisessa, kielellisessä ymmärtämisessä ja monissa luonnollisen kielen käsittelytehtävissä, niillä on merkittäviä rajoituksia. Kielimallien haasteita ovat muun muassa harvinaisten sanojen käsittely, ylisovittuminen koulutusdataan (overfitting), vaikeus käsitellä abstraktia päätelyä ja altistuminen sosiaalisille vinoumille. Tämän tutkielman näkökulmasta erityisen merkittävä rajoite on suurten kielimallien heikkous reaaliaikaisen tai dynaamisen tiedon sisällyttämisessä. Rajoite tekee mallit soveltumattomiksi tehtäviin, jotka vaativat ajantasaista tietoa tai nopeaa sopeutumista muuttuviin konteksteihin. [5] Dynaamiseen tietoon liittyvät haasteet ovat osa laajempaa ongelmaa: suurten kielimallien käyttöönotossa erikoistuneilla aloilla on merkittäviä haasteita, kuten hallusinaatiot ja riittämätön erikoisalan osaaminen [6]. Ulkoisen datan integroiminen vähentää hallusinaatioita ankkuroimalla kielimallin vastaukset todelliseen tietoon ja mahdollistaa ajantasaisen tiedon hyödyntämisen [6]. Seuraavaksi esiteltävä hakutehostettu generaatio on keskeinen lähestymistapa dynaamisen, kielimallin ulkoisen tiedon hyödyntämiseen.

## 3.2 Hakutehostetun generaation periaatteet ja toimintaprosessi

Hakutehostettu generaatio (Retrieval-Augmented Generation, RAG) [1] viittaa menetelmään, jossa kielimalli tehostaa luonnollisen kielen tuottamisen kykyjään hakemalla dynaamisesti ulkoista tietoa generointiprosessin aikana. Tämä tekniikka yhdistää suurten kielimallien generatiiviset kyvyt tiedonhakuun laajoista tietokannoista tai dokumenteista. [6] Tiedonhaulla rikastettuja kielimalleja kutsutaan nimellä Retrieval-Augmented Language Model (RALM) [7].

RAG on yksi kolmesta pääasiallisesta suurten kielimallien optimointimenetelmästä kehotesuunnittelun (prompt engineering) ja hienosäädön (fine-tuning) ohella. Nämä tekniikat lisäävät mallin ohjattavuutta eri tavoin: RAG täydentää kehotetta ulkoisilla dokumenttikatkelmilla, kehotesuunnittelu ohjaa mallia tehtäväkohtaisilla ohjeilla (kuten ajatusketjukehoteilla), ja hienosäätö kouluttaa mallia tehtäväkohtaisella tietoaaineistolla. Näitä menetelmiä voidaan käyttää myös yhdessä paremman vaikutuksen saavuttamiseksi. [8]

RAG-järjestelmien hakukomponentin toiminta perustuu vektorisaatioon, joka on prosessi, jossa jäsentämätön data muunnetaan numeerisiksi vektoreiksi. Tekstidatan tapauksessa nämä korkean ulottuvuuden vektorit (tuhansia tai jopa miljoonia ulottuvuuksia) tallentavat tekstin semanttisia piirteitä, mahdollistaen merkitykseen perustuvan tiedonhaun. Vektorimuotoinen esitystapa mahdollistaa tekstien samankaltaisuuden arvioinnin niiden vektoriavaruuden läheisyyden perusteella. [9]

RAG hyödyntää tätä vektorisaatiota upottamalla sekä tekstikatkelmat että käyttäjän kyselyn samaan vektoriavaruuteen. Samankaltaiset tekstikatkelmat sijaitsevat lähellä toisiaan, jolloin ne voidaan löytää nopealla lähimmän naapurin haulla. Vektoriavaruudessa lähellä käyttäjän kyselyä olevat tekstikatkelmat lisätään kielimallin kontekstiin, mikä mahdollistaa faktuaalisesti paremmin perusteltujen vastausten tuottamisen. [10] Pelkkä semanttinen läheisyys ei kuitenkaan aina ole luotettavin tapa löytää sopivimmat dokumentit. Luvussa 5.2 esitellään RAG-järjestelmissä esiintyvä *huomiovino* (attention bias) [11], jonka seurauksena haussa saatetaan jättää huomiotta aikamääritteet. Tämän johdosta löytynyt tieto saattaa olla semanttisesti sopivaa, mutta vanhentunutta.

## 3.3 RAG-järjestelmien rajoitukset ja tehostamismenetelmät

RAG-järjestelmät kohtaavat useita merkittäviä haasteita liittyen tiedon käsittelyyn. Tiedonhaulla rikastaminen voi jopa heikentää kielimallin suorituskykyä tilanteissa, joissa haettu konteksti on epäolennaista tai harhaanjohtavaa. Tällöin malli saattaa epäonnistua vastaamaan kysymyksiin, joihin se ilman hakua olisi osannut vastata oikein. Tämä johtuu osittain siitä, että kielimalleja ei tyypillisesti ole koulutettu erottamaan ulkoisen kontekstin luotettavuutta. [12]

Näiden kontekstin laadun haasteiden lisäksi RAG-järjestelmät kohtaavat rakenteellisia rajoituksia. Perinteisen RAG-järjestelmän tekstipohjainen hakumoduuli käsittelee heikosti monimutkaista jäsennellyä tietoa, kuten graafirakenteita ja hierarkkisia suhteita. Käytännön sovelluksissa, kuten sosiaalisissa verkostoissa ja tieteellisessä kirjallisuudessa, tieto ei esiinny vain erillisinä tekstikatkelmina vaan sisältää runsaasti kontekstuaalisia yhteyksiä. Lisäksi, vaikka RAG-järjestelmät hyödyntävät ulkoista tietoa, ne tuottavat edelleen epätarkkoja tai epä johdonmukaisia tuloksia monimutkaisissa päättelytehtävissä, mikä rajoittaa niiden soveltamista vaativissa käyttötapauksissa. [13] Luvussa 4 käsitellään tarkemmin RAG-järjestelmien kohtaamia tietokonflikteihin liittyviä haasteita ja luvussa 5 ajallisia haasteita.

RAG-järjestelmien hakua voidaan tehostaa useilla menetelmillä. HyDE-haku (HyDE retrieval) [14] parantaa hakuprosessia pyytämällä kielimallia luomaan hypoteettisen dokumentin, joka toimii hakukyselynä. Tämä dokumentti on vektoriavaruudessa todennäköisesti lähempänä relevanttia tietoa kuin alkuperäinen kysely. Suojakaiteilla varustettu RAG (RAG with guardrails) puolestaan tarkistaa, osuuko kyselyn upotusarvo ennalta määritellyyn vektoriavaruuden alueeseen, mahdollistaen tiedonhaun kontrolloinnin tietyyttypisten kyselyiden kohdalla. [10] Luvussa 6.1 esitellään tarkemmin lisää haun tehostamisen menetelmiä (FLARE [15] ja Iter-RetGen [16]). Näistä FLARE ennustaa jo tekstin generointivaiheessa seuraavan lauseen ja muodostaa hakuja tämän perusteella, kun taas Iter-RetGen on iteratiivinen menetelmä, jossa kielimallin vastausta käytetään uuden haun pohjana, eli jokaiseen kyselyyn käytetään useampi kielimallin vastaus.

Lisäksi graafirakenteeseen yhdistettyjen RAG-mallien kehittäminen on nouseva tutkimusalue, joka pyrkii ratkaisemaan perinteisten mallien rajoituksia. Hyödyntämällä graafineuroverkkojen (GNN) teknologiaa, nämä mallit pystyvät käsittelemään tehokkaasti tietoa, jossa solmut edustavat entiteettejä ja kaaret niiden välisiä suhteita. Tämä mahdollistaa assosiativisten rakenteiden tallentamisen ja hyödyntämisen generointiprosessissa, parantaen merkittävästi mallin kykyä käsitellä monimutkaisia päättelytehtäviä. [13] Dokumenttien lisääntynyt kohina heikentää merkittävästi RAG-järjestelmien tehokkuutta, minkä vuoksi jäsennellyt, tiiviit tietomuodot kuten tietograafit ovat osoittautuneet erityisen hyödyllisiksi järjestelmien syötteinä [8].

### 3.4 RAG-järjestelmien arviointi ja mittaaminen

RAG-järjestelmien arviointi edellyttää lähestymistapoja, jotka huomioivat sekä tiedonhaun että tekstin generoinnin tehokkuuden. RAGVAL [10] on esimerkki arviointikehyksestä, joka mahdollistaa näiden järjestelmien automaattisen arvioinnin.

RAGVAL perustuu kolmeen periaatteeseen: 1) arvioinnin tulisi onnistua pelkästään RAG-järjestelmän tietokannan avulla, 2) tietoaaineistojen tulee soveltua automaattiseen arviointiin, ja 3) aineistojen tulee huomioida kielimallin sisäisen tiedon vaikutus tuloksiin. Viimeinen periaate on olennainen, sillä nykyaikaiset kielimallit sisältävät jo koulutusvaiheen johdosta merkittävän määrän tietoa. Tämä tekee RAG-järjestelmän lisäarvon arvioinnista haastavaa, koska malli voi vastata kysymyksiin oikein, vaikka tiedonhaku ei olisi toiminut. [10]

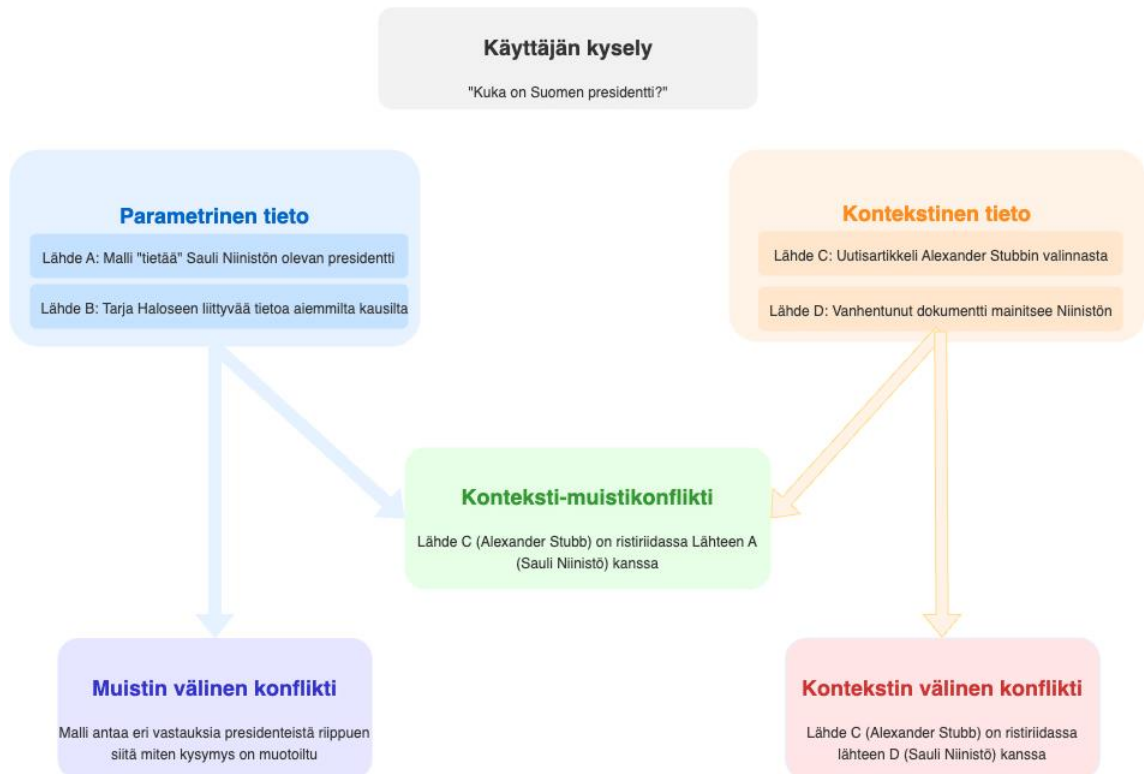
RAGVAL-menetelmässä tietokannasta poimitaan luonnollisesti esiintyviä tietolohkoja ja luodaan niistä kysymyksiä. Kielimalli arvioi tuotettuja vastauksia hyödyntäen tietoasymmetriaa: arvioijalla on käytössään kaikki relevantti tieto (luodut kysymykset, tietolohkot ja RAG-järjestelmän tuottamat vastaukset), joten se voi luotettavasti arvioida RAG-järjestelmän vastausten laatua. Kullekin vastaukselle luodaan pisteytys asteikolla 1–5 sekä relevanssin että totuudenmukaisuuden osalta. Arvioijamallina on suositeltavaa käyttää niin vahvaa kielimallia kuin mahdollista, mutta RAG-järjestelmän käyttämällä kielimallilla ei ole arvioinnin kannalta merkitystä. [10]

Tämän tutkielman kontekstissa arviointikehyksiä voidaan soveltaa erityisesti tietokonfliktien ja ajallisten haasteiden näkökulmasta. Keskeistä on mitata, miten RAG-järjestelmä pystyy tunnistamaan ajallisesti relevantin tiedon ja miten järjestelmä ratkaisee ristiriitoja eri lähteiden välillä. Esimerkkinä tällaisesta arviointikehyksestä toimii DynamicQA [17], joka on suunniteltu arvioimaan kielimallien kykyä käsitellä dynaamisia, ajallisesti muuttuvia faktoja. DynamicQA käyttää 11 378 kysymys-vastaus-paria, joissa aikasidonnaisuutta mitataan Wikipedia-muokkausten määrällä ja kiistanalaisuutta peruutusten määrällä [17]. Ilman tällaisia arviointityökaluja on vaikea mitata, miten hyvin järjestelmät todella selviävät ajallisista haasteista käytännön sovelluksissa.

## 4. TIETOLÄHTEIDEN KONFLIKTIT RAG-JÄRJESTELMISSÄ

### 4.1 Tiedon tyypit ja konfliktit

Suurten kielimallien sisäistä, koulutusvaiheessa omaksumaa tietoa kutsutaan parametrisiksi tiedoksi. Käyttäjän syötteiden, dialogihistorian ja haettujen dokumenttien muodostama kokonaisuutta nimitetään puolestaan kontekstiseksi tiedoksi. Näiden tietolähteiden välille voi syntyä kolmenlaisia konflikteja. Konteksti-muistikonfliktissa kontekstinen tieto on ristiriidassa parametrisen tiedon kanssa. Kontekstin välisessä konfliktissa kaksi kontekstiin sisällytettyä tietolähdettä esittävät vastakkaisia väitteitä. Muistin välinen konflikti puolestaan esiintyy siten, että kielimallin parametrinen tieto saattaa tuottaa erilaisia vastauksia samaan asiaan eri tavoin muotoiltuihin kysymyksiin. Tämä johtuu ristiriitaisesta tiedosta, joka on sisällytetty mallin parametreihin jo ennakkokoulutusvaiheessa monimuotoisista koulutustietolähteistä. [3] Kuvassa 1 on esitelty edellä mainitut tiedon tyypit ja konfliktit RAG-järjestelmissä.



**Kuva 1.** Tietokonfliktien tyypit RAG-järjestelmissä.

RAG-järjestelmien tietokonfliktit ilmenevät erityisen selkeästi silloin, kun tarkastellaan tiedon dynaamista luonnetta. Dynaamiset, usein muuttuvat faktat aiheuttavat merkittäviä haasteita kielimalleille, sillä niissä esiintyy tyypillisesti enemmän sisäisiä muistiritiriitoja

kuin staattisessa tiedossa. Tämä johtaa paradoksaaliseen tilanteeseen: vaikka RAG-järjestelmien tarkoitus on päivittää tietoa ulkoisen kontekstin avulla, juuri usein päivittyvää tietoa on kaikkein vaikeinta päivittää tehokkaasti. [17]

Sen sijaan RAG-järjestelmät onnistuvat parhaiten staattisten, harvoin muuttuvien faktojen käsittelyssä, vaikka tällaisen tiedon päivittäminen on harvemmin tarpeellista [17]. Tämä ristiriita korostaa RAG-järjestelmien rakenteellista haastetta: ne toimivat parhaiten tilanteissa, joissa niiden tuoma lisäarvo on pieni.

Kielimallien vuorovaikutuksessa kontekstin kanssa voidaankin havaita kaksi selkeää käyttäytymismallia. Vaikutetuissa tapauksissa malli omaksuu ulkoisen kontekstin tiedon ja muuttaa toimintaansa sen mukaisesti, kun taas itsepäisissä tapauksissa malli jättää ulkoisen kontekstin huomiotta ja pitäytyy sisäisessä tiedossaan [17]. Seuraavassa aluvussa esitellään RALM-malleissa esiintyviä vinoumia, jotka liittyvät näihin käyttäytymismalleihin.

## 4.2 RALM-mallien tietolähdepreferenssien vinoumat

Tiedonhaulla rikastetuilla suurilla kielimalleilla esiintyy erilaisia systemaattisia vinoumia tietokonfliktien käsittelyssä. Nämä vinoumat vaikuttavat siihen, miten mallit priorisoivat ja käsittelevät ristiriitaista tietoa. Vinouman ilmenemismuoto riippuu osittain käytössä olevan mallin koosta ja tiedon saatavuudesta.

Harvinaisemman tiedon suhteen RALM-mallit yleisimmin suosivat ulkoisia lähteitä. Tiedon yleistyessä malli siirtyy vähitellen luottamaan enemmän sisäiseen muistiinsa. Tätä helposti saavutettavan sisäisen muistin suosimista yksinkertaisille faktoille kutsutaan *saatavuusvinoumaksi* (availability bias). Jin et al. tutkimuksessa [7] tiedon yleisyyttä mitataan suosiotasolla, jossa suurempi arvo merkitsee yleisempää tietoa. LLama2 7B -mallilla suosiotasolla  $10^3$  malli nojaa noin 80–85-prosenttisesti ulkoisiin lähteisiin, jotka ovat ristiriidassa sen sisäisen muistin kanssa, kun taas suosiotasolla  $10^6$  se laskee noin 60% tasolle, osoittaen että malli alkaa “luottaa itseensä” enemmän tiedon ollessa yleisempää. [7]

RALM-malleilla esiintyy myös *vahvistusvinouma*: ne valitsevat mieluummin sisäisen muistinsa kanssa yhteensopivaa todistusaineistoa, riippumatta siitä onko tieto oikeaa tai väärää. Jin et al. tutkimuksen [7] tulosten mukaan GPT-3.5-Turbo-malli valitsee sisäistä muistiaan tukevia todisteita 86% tapauksista silloin, kun sen sisäinen tieto on oikeaa, mutta myös 74% tapauksista silloin, kun sen sisäinen tieto on virheellistä. Tämä osoittaa mallin vahvan taipumuksen suosia sisäistä muistiaan tukevia todisteita tiedon paikkansapitävyydestä riippumatta. [7]

Kolmas vinouma liittyy mallin koon kasvamiseen ja sen tuottamaan liialliseen "itsevarmuuteen". Kun yksinkertaiselle noutavalle kielimallille tarjotaan ulkoista tietoa, joka on ristiriidassa mallin sisäisen muistin kanssa, malli helposti sivuuttaa sisäisen tietonsa uskoen ulkoiseen tietoon. Mallin koon ja kyvykkyyksien kasvaessa malli saa enemmän it-seluottamusta sisäiseen muistiinsa. Kehittyneemmissä RALM-malleissa ilmeneekin *Dunning–Kruger-efekti*: ne suosivat jatkuvasti virheellistä sisäistä muistiaan, vaikka oikea tieto olisi saatavilla. [7]

Muistisuhteen erotus (IMR - CMR) mittaa, kuinka paljon useammin malli pitäytyy virheellisessä sisäisessä tiedossaan verrattuna oikeaan tietoon kohdatessaan ristiriitaista ulkoista informaatiota. Dunning–Kruger-efekti on selkeästi havaittavissa Jin et al. tutkimuloksissa [7], jotka osoittavat, että pienemmät mallit kuten FLAN-T5-XL 3B ovat valmiimpia muuttamaan virheellisiä uskomuksiaan kuin oikeita (muistisuhteen erotus - 5,88%), kun taas hieman suuremmat mallit kuten LLaMA2 13B osoittavat päinvastaista käyttäytymistä (muistisuhteen erotus +9,06%). Erityisen voimakkaasti ilmiö näkyy GPT-3.5-Turbo-mallissa, joka pitäytyy virheellisessä sisäisessä tiedossaan huomattavasti useammin kuin oikeassa (muistisuhteen erotus +32,55%). [7]

## 5. AJALLISET HAASTEET

Ajalliset haasteet muodostavat merkittävän ongelman RAG-järjestelmille. Suuret kielimallit sisällyttävät runsaasti tietoa parametreihinsa, ja tiedonhauulla voidaan saada paljon uutta tietoa käytettäväksi. Luonnollisesti tämä johtaa siihen, että osa tiedosta on ajallisessa ristiriidassa keskenään. Seuraavissa alaluvuissa tarkastellaan, miten tiedon ajallinen luonne synnyttää erilaisia ristiriitoja järjestelmissä, ja millaisia päättelyhaasteita mallit kohtaavat käsitellessään aikaan sidottuja kyselyitä.

### 5.1 Tiedon ajalliset muutokset ja ristiriidat

Ajallisen tiedon käsittelyn ensimmäinen merkittävä haaste on *ajallinen sovitusvirhe* (temporal misalignment). Tämä virhe syntyy malleissa, jotka on koulutettu menneisyydessä kerätyllä datalla, sillä ne eivät välttämättä heijasta tarkasti nykyisiä tai tulevia todellisuuksia, eli mallin käyttöönoton jälkeistä kontekstuaalista tietoa. Tällainen epäkohdistus voi heikentää mallin suorituskykyä ja merkityksellisyyttä ajan myötä, sillä malli ei välttämättä kykene havaitsemaan uusia trendejä, muutoksia kielen käytössä, kulttuurisia muutoksia tai tiedon päivityksiä. Ongelman odotetaan voimistuvan esikoulutusparadigman ja mallien skaalauksen kasvavien kustannusten vuoksi. [3]

Sovitusvirheen ohella on huomioitava *tietolähteiden ajallinen epätasapaino*. Tietolähteet, kuten Wikipedia ja verkkoaineistot, kasvavat jatkuvasti, mutta jakaantuvat epätasaisesti ajallisesti: uusia dokumentteja on enemmän, sillä vanhoja päivitetään ja verkkosisältöä tuotetaan nykyään aiempaa enemmän. Tämän vuoksi malli saattaa unohtaa faktoja, jotka olivat tosia vain vähemmän edustetuilla ajanjaksoilla, mikä heikentää sen kykyä vastata kaukaisempaa menneisyyttä koskeviin kysymyksiin. [2]

Näiden rakenteellisten haasteiden ohella myös konkreettiset ajallisesti muuttuvat faktat aiheuttavat mallille ristiriitoja, kuten "Suomen presidentti on Alexander Stubb / Sauli Niinistö". Koska kielimallien koulutuksessa ei yleensä huomioida aikaan liittyvää metatietoa, syntyy *keskiarvoistamisilmiö*, jonka seurauksena malli suhtautuu epävarmasti kaikkiin oikeisiin vastauksiin [2].

### 5.2 RALM-mallien ajallisen päättelyn haasteet

Edellä kuvatut tiedon ajallisiin muutoksiin liittyvät haasteet ovat läheisesti yhteydessä RALM-mallien kykyyn päätellä ja tehdä johtopäätöksiä ajallisesti määritellyistä tiedoista. Kielimallien "vanhentuuessa" niiltä kysytään yhä useammin tietoja, jotka ylittävät niiden koulutusaineiston aikajänteen. Vaikka arvaaminen voi vaikuttaa huonolta ratkaisulta,

monesti on kohtuullista olettaa tulevaisuuden muistuttavan nykyisyyttä: Alaskan pääkaupunki pysyy todennäköisesti samana 20 vuoden päästäkin, vaikka kuvernöörin ennustaminen olisi mahdotonta. Mallin vastausvarmuuden tulisi ideaalitulanteessa heijastaa tätä ennustettavuuden eroa. [2]

RAG-järjestelmien kyky käsitellä ajallisesti määrittyvää tietoa heikkenee osaltaan myös huomiovuonuman vuoksi, jossa mallit ylikorostavat semanttista samankaltaisuutta jättäen aikamääritteet huomiotta. Tämän seurauksena upotusmallit hakevat dokumentteja, jotka ovat semanttisesti sopivia mutta ajallisesti epäolennaisia. Ongelma korostuu, kun käyttäjät ilmaisevat ajanjakson epätasaisesti: hakujärjestelmä saattaa sivuuttaa vuoden 1969, vaikka se täyttäisi kyselyn "vuosien 1968 ja 1970 välillä" ehdot, tai se ei tunnista että "1990-luku" viittaa vuosiin 1990—1999. Lisäksi koulutusvaiheen aikana syntyneet semanttiset vinoumat heikentävät RALM-mallien päättelykykyä: jos malli altistuu tietyille aiheille useammin, se saattaa hakea niihin liittyviä dokumentteja myös epäolennaisissa kyselyissä. [11] Seuraavassa luvussa käsitellään teknisiä ja metodologisia ratkaisumenetelmiä näihin ajallisiin haasteisiin sekä aiemmin esiteltyihin tietokonflikteihin ja vinoumiin.

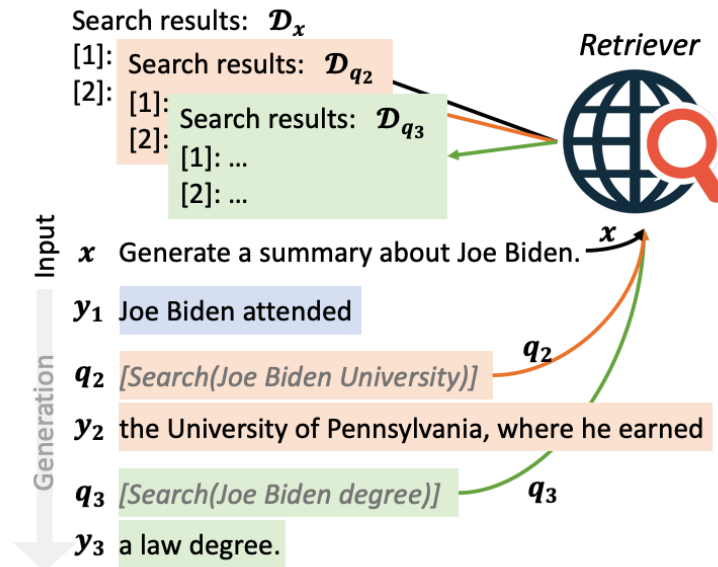
## 6. RATKAISUMENETELMÄT

### 6.1 Tekniset ratkaisut

RAG-järjestelmien ajallisiin haasteisiin ja tietokonflikteihin on kehitetty useita teknisiä ratkaisuja, jotka pyrkivät parantamaan mallien kykyä käsitellä tietoa täsmällisemmin ja luotettavammin. Seuraavaksi esitellään neljä keskeistä teknistä ratkaisua ja niiden tuomat konkreettiset hyödyt. Menetelmät valittiin sillä perusteella, että jokainen menetelmä keskittyy hieman eri osa-alueeseen tietokonfliktien ratkaisussa. Kaksi ensimmäistä menetelmää (FLARE ja Iter-RetGen) parantavat mallin kykyä yhdistää dynaamisesti tai iteraatiivisesti tiedonhakua ja vastausten generointia, jolloin mallin vastausten ajantasaisuus ja tarkkuus paranevat. Kolmatta menetelmää (kontrastiivista oppimista) voidaan hyödyntää sekä tiedonhaun että generoinnin osalta, jolloin mallin kyky erottaa relevantti, ajantasainen ja luotettava tieto parantuu. Neljäs menetelmä (CD2) on erityisesti suunniteltu parantamaan generaattorin päätöksentekokykyä tietokonfliktitilanteissa.

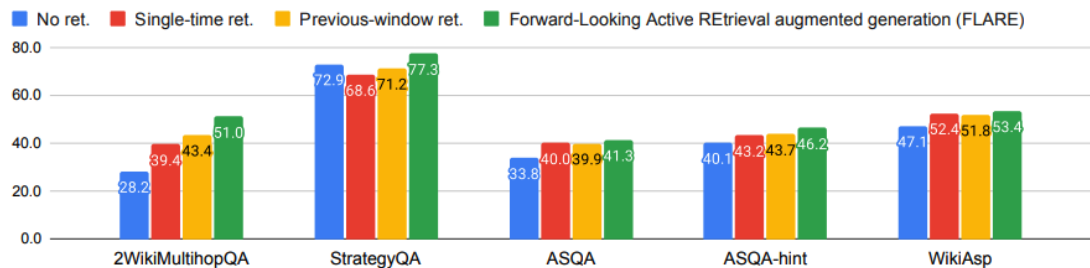
#### 6.1.1 FLARE: Ennakoiva tiedonhaku

Forward-Looking Active Retrieval-Augmented Generation (FLARE) [15] on menetelmä, joka ennustaa tekstin generointivaiheessa seuraavan lauseen ja käyttää tätä ennustetta hakukyselyinä, mikä mahdollistaa relevantin tiedon hakemisen jo ennen kuin malli on sitoutunut tiettyyn vastaukseen. Menetelmä tunnistaa lauseista epävarmat sanat ja pyrkii aktiivisesti parantamaan niitä hakemalla lisätietoa. [15] Tämä lähestymistapa tarjoaa ratkaisuja erityisesti luvussa 5.1 kuvattuun ajalliseen sovituserheeseen, sillä se mahdollistaa ajantasaisen tiedon hakemisen ennen kuin malli on sitoutunut mahdollisesti vanhentuneeseen parametriseen tietoonsa. FLARE auttaa myös luvussa 4.1 kuvattuihin konteksti-muistikonflikteihin tunnistamalla epävarmat sanat ja aktiivisesti hakemalla tarkempaa tietoa. Näin malli voi tarkistaa tiedon paikkansapitävyyden sekä ajallisen relevanssin ennen sen esittämistä. Kuvassa 2 on esitelty ennakoivan haun toiminta FLARE-menetelmässä.



**Kuva 2.** Ennakoivan haun toiminta (FLARE) [15].

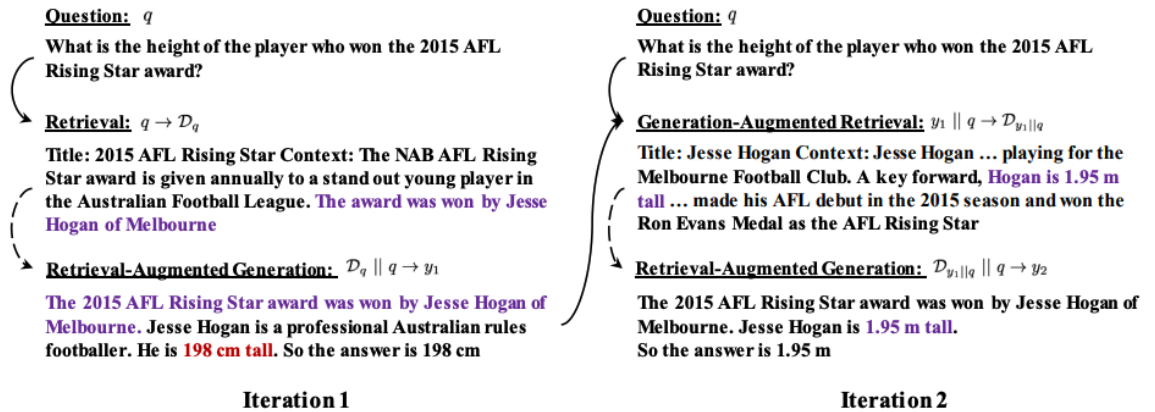
FLARE-menetelmää kokeiltiin erilaisissa tehtävissä, joissa pitää tuottaa pitkiä, paljon tietoa vaativia tekstejä. FLARE pärjäsikin joko paremmin tai yhtä hyvin kuin vertailumenetelmät kaikissa testeissä, kuten kuvassa 3 on esitelty. [15]



**Kuva 3.** FLARE-menetelmän ja lähtökohtien vertailu eri tietoaaineistoilla [15].

### 6.1.2 Iter-RetGen: Iteratiivinen tiedonhaku

Iter-RetGen [16] on menetelmä, jossa tiedonhaku ja tekstin tuottaminen yhdistyvät iteratiivisella, toisiaan vahvistavalla tavalla. Menetelmä hyödyntää mallin ensimmäistä vastausta tunnistaakseen, mitä lisätietoa tarvitaan. Tämän avulla voidaan muodostaa hakukysely uudelle tiedonhauulle, mikä mahdollistaa tarkemman ja relevanttimman tiedon hakemisen, kuten kuvassa 4 esitetään. [16] Menetelmä auttaa erityisesti luvussa 5.2 esiteltyyn huomiovinouman ongelmaan, sillä iteratiivinen prosessi ohjaa mallia keskittymään hakukyselyissä olennaisiin tietoihin, mukaan lukien mahdollisesti aiemmin huomiotta jätettyihin aikamääritteisiin



**Kuva 4.** Iter-RetGenin toiminta [16].

Toisin kuin monet muut menetelmät, Iter-RetGen käsittelee kaiken haetun tiedon kokonaisuutena, ja säilyttää suurelta osin tekstin tuottamisen joustavuuden ilman rakenteellisia rajoitteita [16]. Tämä tiedon kokonaisvaltainen käsittely on erityisen arvokasta tilanteissa, joissa tietolähteet sisältävät ristiriitaista tietoa.

Iter-RetGenin toimintaa arvioitiin monivaiheisessa kysymyksiin vastaamisessa, faktojen todentamisessa ja arkijärjen päättelyssä, ja tulokset osoittavat, että menetelmä pystyy joustavasti hyödyntämään sekä parametrissa että ei-parametrissa tietoa. Menetelmä on joko parempi tai kilpailukykyinen verrattuna vastaaviin tiedonhaulla täydennettyihin vertailumenetelmiin, kuten taulukossa 1 esitetyistä tuloksista voidaan havaita. Lisäksi Iter-RetGen aiheuttaa vähemmän tiedonhaku- ja generointiresurssien kulutusta. Suorituskykyä voidaan edelleen parantaa generointiin perustuvalla tiedonhaun mukautuksella. [16]

**Taulukko 1.** Iter-RetGenin tulosten vertailu eri tietoaineistoilla (mukailten) [16].

Menetelmä	HotpotQA (F1)	2WikiMultiHopQA (F1)	StrategyQA (Tarkkuus)
Perusmalli (ilman hakua)	36,8%	29,2%	66,5%
Perusmalli (haulla)	44,7%	35,4%	65,6%
Self-Ask	55,2%	48,8%	70,2%
Iter-RetGen 7	60,4%	47,4%	74,1%

### 6.1.3 Kontrastiivinen oppiminen

Kontrastiivinen oppiminen on keskeinen tekniikka koneoppimisessa, jonka tavoitteena on luoda esitysmuotoja vertaamalla positiivisia pareja negatiivisiin. Kontrastiivinen oppiminen esiintyy sekä valvotussa että valvomattomassa muodossa. Valvottu kontrastiivinen oppiminen hyödyntää luokkatunnisteita, kun taas itseohjatussa oppimisessa positiiviset näyteparit muodostetaan tyypillisesti ankkuridatan muunnoksilla. [11]

Kontrastiivinen oppiminen tarjoaa tehokkaan lähestymistavan erityisesti aikariippuvaisien kyselyjen käsittelyyn RAG-järjestelmissä, mahdollistaen tekstin aikamääritteiden suoran käytön positiivisten ja negatiivisten näyteparien määrittelyssä hyödyntämällä luokkatunnisteita. Tämä opettaa mallia tunnistamaan ajallisesti relevantin tiedon ja erottamaan sen epärelevantista tiedosta. Esimerkiksi malli oppii yhdistämään ilmaisun "1990-luku" vuosiin 1990—1999, mikä parantaa sen kykyä vastata ajallisesti määriteltyihin kyselyihin. [11]

Kontrastiivisen oppimisen avulla saadaan optimoitua mallin upotusavaruus tehokkaasti. Samankaltaisia ajallisia merkityksiä sisältävät ilmaisut sijoittuvat lähemmäs toisiaan, kun taas erilaiset ajalliset merkitykset etäännyvät toisistaan upotusavaruudessa. Aikamääritteiden täsmällinen huomioiminen parantaa merkittävästi hakutulosten tarkkuutta kielimalleissa. [11] Tämä vähentää luvussa 5.2 kuvattua huomiovinoumaa ja parantaa mallin kykyä tunnistaa ajallisesti relevantit dokumentit.

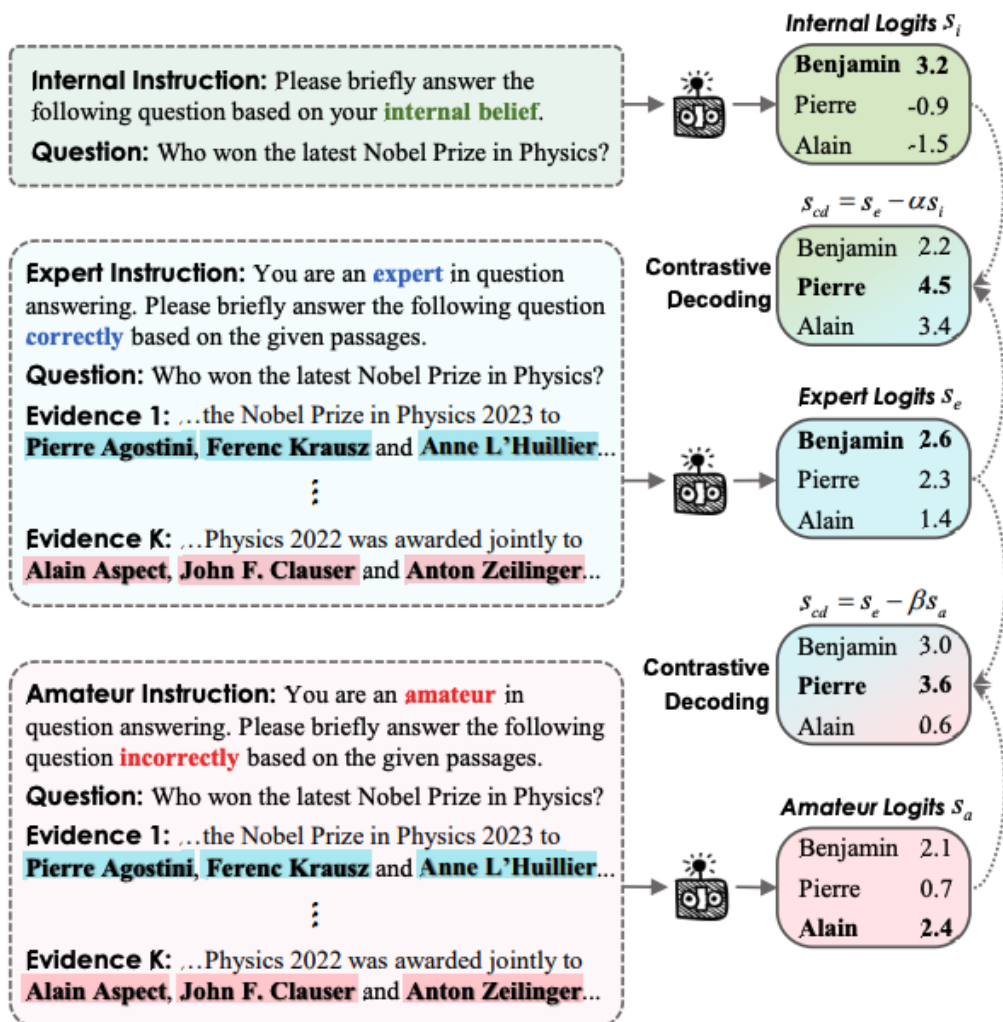
Perinteiset haut painottavat liikaa semanttista samankaltaisuutta aikamääritteiden kustannuksella, mikä johtaa usein epätarkkoihin hakutuloksiin. Kontrastiivinen lähestymistapa pakottaa mallin keskittymään juuri ajallisiin määreisiin. Wu et al. [11] tutkimuksen mukaan kontrastiiviseen oppimiseen perustuvat mallit suoriutuvat huomattavasti paremmin aikariippuvaisista kyselyistä kuin perinteiset mallit, jotka eivät erityisesti huomioi aikamääritteitä. [11]

### 6.1.4 CD2: Luottamustason kalibrointi

Conflict-Disentangle Contrastive Decoding (CD2) [7] on erityisesti RALM-mallien tietokonfliktien ratkaisemiseen suunniteltu menetelmä, joka auttaa kalibroimaan mallin luottamustason tietokonfliktien suhteen. Tämä mallin luottamustason kalibrointi vähentää luvussa 4.2 kuvattujen vinoumien kuten saatavuusvinouman, vahvistusvinouman ja Dunning–Kruger-efektin vaikutusta, mikä parantaa vastausten luotettavuutta ristiriitaisen tiedon yhteydessä. [7]

CD2 hyödyntää faktatietoista ohjeistusvirittämistä (fact-aware instruction tuning) totuudenmukaisten, epäolennaisten ja harhaanjohtavien todisteiden erottamisessa. Tämä

parantaa mallin kykyä tunnistaa luotettava tieto eri aikakausilta peräisin olevista lähteistä. Menetelmässä käytetään asiantuntijamallia tuottamaan totuudenmukaisia vastauksia ja amatöörimallia tuottamaan harhaanjohtavia vastauksia. Malli laskee kunkin vastausvaihtoehdon logit-arvot, jotka edustavat mallin sisäistä luottamusta ennen todennäköisyyksien normalisointia. Asiantuntijamalli tyypillisesti antaa korkean logit-arvon todenmukaiselle vastaukselle, kun taas amatöörimalli antaa korkean arvon harhaanjohtavalle. Näiden mallien tuottamien vastausten logit-arvojen eroa maksimoidaan, jolloin järjestelmä oppii painottamaan vastauksia, joihin asiantuntijamalli luottaa ja joita amatöörimalli ei suosi. Tämä parantaa mallin kykyä erottaa luotettavat vastaukset epäluotettavista erityisesti tilanteissa, joissa todisteet ovat ristiriidassa. Kuvassa 6 on havainnollistettu CD2-menetelmän käyttämää ohjeistusvirittämistä. [7]



Kuva 6. CD2-menetelmän toiminta [7].

Kokeelliset tulokset osoittavat, että CD2-menetelmä pystyy tehokkaasti ratkaisemaan RALM-mallien tietokonflikteja [7]. Tämä on keskeinen etu RAG-järjestelmille, jotka pyrkivät tarjoamaan luotettavia vastauksia nopeasti muuttuvassa tietoympäristössä.

## 6.2 Metodologiset ratkaisut

RAG-järjestelmien ajallisten haasteiden ja tietokonfliktien tehokkaaseen ratkaisemiseen tarvitaan systemaattinen lähestymistapa. Seuraavat metodologiset ratkaisut tarjoavat työkaluja näiden monimutkaisten ongelmien käsittelyyn.

Tietokonfliktien arvioinnissa on tärkeää määrittää asianmukaiset vertailukohtat. Kun tutkitaan esimerkiksi ajallista sovituvirhettä tai huomiiovinoumaa, RAG-järjestelmää voi verrata samaan kielimalliin ilman hakutoimintoa [18]. Tämä paljastaa, parantaako tiedonhaku todella järjestelmän ajallista johdonmukaisuutta. Vertailuasetelmien huolellinen suunnittelu mahdollistaa hakutoiminnon todellisen lisäarvon arvioinnin erityisesti aikariippuvaisissa tehtävissä.

Tietokonfliktien systemaattiseen tutkimiseen tarvitaan lisäksi erityisiä testiaineistoja [18], jotka sisältävät eri aikajaksojen tietoa ja osoittavat selkeästi, milloin tiedon tulisi muuttua tai pysyä samana. Tästä toimii esimerkkinä kappaleessa 3.4 mainittu DynamicQA [17], joka on erityisesti suunniteltu arvioimaan mallien kykyä käsitellä dynaamisia, ajallisesti muuttuvia faktoja. Tällaiset aineistot auttavat arvioimaan, kuinka hyvin järjestelmä käsittelee aikariippuvaisia kyselyjä ja tunnistaa aikatietoon liittyviä rajoitteita.

Tietokonfliktien, kuten konteksti-muistikonfliktien ja kontekstin välisten konfliktien, tutkimiseen tarvitaan myös erityisiä mittareita [3, 11]. Mittarina voi olla esimerkiksi oikeellisuus (correctness), jota voidaan mitata EM- ja F1-mittareilla tai Recall (R) -mittarilla, joka laskee, kuinka suuri osa kultaisen standardin vastauksesta sisältyy mallin tuottamaan vastaukseen [7]. Tärkeä mittari on myös uskollisuus (faithfulness), jota voidaan mitata K-Precision (KP) -mittarilla, joka laskee, kuinka suuri osa mallin tuottamista vastausten merkeistä esiintyy hakutuloksissa. Tämä mittari soveltuu myös totuudenmukaisten, epäolennaisten ja harhaanjohtavien todisteiden erottamiseen [7]. Lisäksi muistinkäyttöä (memorization) voidaan mitata muistisuhteen avulla, joka kuvaa mallin taipumusta turvautua sisäiseen tietoonsa ulkoisten lähteiden sijaan [7]. Nämä mittarit mahdollistavat RAG-järjestelmien systemaattisen vertailun ja kehittämisen erilaisilla tietoaineistoilla.

Järjestelmällinen virheiden analysointi [18] on erityisen tärkeää tietokonfliktien ja ajallisten haasteiden ratkaisemisessa. Tämä sisältää erilaisten vinoumien, kuten saatavuusvinouman, vahvistusvinouman ja Dunning–Kruger-efektin tunnistamisen mallien käyttäytymisessä [7]. Erityisen tärkeää on analysoida virheitä dynaamisten faktojen käsittelyssä, koska tämä on tyypillisesti RAG-järjestelmille haastavin osuus [17]. Analysointi auttaa ymmärtämään, milloin malli luottaa liikaa sisäiseen tietoonsa ulkoisen kontekstin sijaan tai päinvastoin, mikä mahdollistaa kohdenetetut parannukset järjestelmään.

Yhdistämällä edellä kuvatut metodologiset lähestymistavat aiemmin kuvattuihin teknisiin ratkaisut voidaan kehittää RAG-järjestelmiä, jotka käsittelevät ajallisia haasteita ja

tietokonflikteja entistä luotettavammin. Kirjallisuuden perusteella on selvää, että tehokas ratkaisu edellyttää sekä teknisiä että metodologisia innovaatioita, joita on sovellettava systemaattisesti koko järjestelmän kehitysprosessissa.

## 7. ARVIOINTI JA ANALYYSI

### 7.1 Teknisten ratkaisumenetelmien vertailu ja tehokkuus

Edellisessä luvussa esitellyt ratkaisumenetelmät tarjoavat erilaisia lähestymistapoja RAG-järjestelmien ajallisten haasteiden ja tietokonfliktien ratkaisemiseen. Tässä aluvussa vertaillaan näiden menetelmien tehokkuutta ja soveltuvuutta erilaisiin käyttötilanteisiin.

FLARE [15] ja Iter-RetGen [16] edustavat molemmat iteratiivisia lähestymistapoja, mutta eroavat merkittävästi toimintaperiaatteiltaan. FLARE on proaktiivinen menetelmä, joka ennustaa tekstin generointivaiheessa seuraavan lauseen ja käyttää tätä ennustetta hakukyselynä. Tämä mahdollistaa relevantin tiedon hakemisen jo ennen kuin malli on sitoutunut tiettyyn vastaukseen. Iter-RetGen puolestaan hyödyntää mallin ensimmäistä vastausta tunnistaakseen, mitä lisätietoa tarvitaan.

Tehokkuuden näkökulmasta FLARE toimii erityisen hyvin tilanteissa, joissa tarvitaan nopeaa päätöksentekoa ja epävarmuuksien tunnistamista tekstin tuottamisen aikana. Iter-RetGen taas soveltuu hyvin monimutkaisiin kyselyihin, joissa vastaus vaatii useita iteraatioita ja tietoa eri lähteistä. Iter-RetGen on osoittautunut tehokkaaksi erityisesti vähemmän hakuresursseja kuluttavana ratkaisuna verrattuna muihin iteratiivisiin menetelmiin [16], mikä on merkittävä etu laajaa käyttöä vaativissa järjestelmissä.

CD2 [7] ja kontrastiivinen oppiminen [11] molemmat keskittyvät tietokonfliktien ratkaisuun, mutta eri järjestelmän vaiheissa. CD2 toimii generointivaiheessa kalibroimalla kiellimallin luottamustasoa, kun kontekstissa on ristiriitaista tietoa, kun taas kontrastiivinen oppiminen parantaa hakukomponenttia luomalla upotusavaruuden, jossa ajallisesti samankaltaiset merkitykset sijaitsevat lähellä toisiaan. Näin CD2 ratkaisee konflikteja kiellimallin sisällä, kun taas kontrastiivinen oppiminen ehkäisee konflikteja parantamalla ajallisesti relevantin tiedon hakua.

Kontrastiivinen oppiminen on osoittautunut erityisen tehokkaaksi aikamääritteiden käsittelyssä ja huomioviinouden vähentämisessä [11]. CD2 taas tarjoaa tehokkaan ratkaisun erityisesti RALM-mallien viinoumiin, kuten Dunning–Kruger-efektiin ja vahvistusviinoumaan [7].

Tässä tutkielmassa tarkastellut ratkaisumenetelmät lähestyvät RAG-järjestelmien tietokonflikteja eri vaiheissa ja eri näkökulmista. Näin ollen optimaalinen ratkaisu RAG-järjestelmien tietokonflikteihin ja ajallisiin haasteisiin voisi löytyä monikerroksisesta lähestymistavasta, jossa kontrastiivinen oppiminen optimoi upotusavaruuden ajallisen

relevanssin suhteen, CD2 ratkaisee jäljelle jääviä konflikteja dekodausvaiheessa ja FLARE tai Iter-RetGen tarjoaa iteratiivisen tarkentamisen. Taulukkoon 2 on koottu edellä mainittujen menetelmien vertailu.

**Taulukko 2. Teknisten ratkaisumenetelmien vertailu.**

Menetelmä	Lähestymistapa	Soveltuvuus	Edut	Heikkoudet
FLARE [15]	Proaktiivinen, ennustava haku	Pitkien, faktatietoa vaativien tekstien generointi	Tunnistaa epävarmat tiedot etukäteen	Hidastaa vastausaikoja, enemmän API-kutsuja
Iter-RetGen [16]	Iteratiivinen, reaktiivinen haku	Monimutkaiset kyselyt, jotka vaativat tietoa useista lähteistä	Säilyttää tekstin tuottamisen joustavuuden, tehokkaampi kuin monet muut iteratiiviset hakumenetelmät	Hidastaa vastausaikoja, enemmän API-kutsuja
CD2 [7]	Luottamustason kalibrointi	Tilanteet, joissa esiintyy harhaisia päätelmiä	Vähentää tehokkaasti kielimallien harhojen ja vinoumien vaikutusta	Ei suoraan ratkaise hakuongelmia
Kontrastiivinen oppiminen [11]	Uputusavaruuden optimointi	Aikariippuvaiset kyselyt ja aikamääritteiden tunnistaminen	Parantaa merkittävästi hakutulosten tarkkuutta aikariippuvaisissa kyselyissä	Vaatii korkealaatuisia aikamääritteillä merkittyä tietoa

## 7.2 Käytännön haasteet ja rajoitukset

Lupaavista tuloksista huolimatta esiteltyillä ratkaisumenetelmillä on käytännön implementoinnin haasteita ja rajoituksia, joita esitellään tässä alaluvussa. Ensinnäkin kaikkien RAG-menetelmien tehokkuus riippuu olennaisesti käytettävissä olevien tietolähteiden laadusta ja kattavuudesta. Esimerkiksi kontrastiivinen oppiminen vaatii korkealaatuisia, aikamääritteillä merkittyä tietoa tehokkaaseen koulutukseen [11]. Jos tietolähteet ovat puutteellisia tai ne sisältävät itsessään ristiriitaista tietoa, mikään ratkaisumenetelmä ei voi täysin korjata ongelmaa. Kontrastiivinen oppiminen voi toimia hyvin englanninkielisillä aikamääritteillä, mutta vähemmän tutkituilla kielillä koulutusaineiston saatavuus voi olla rajoitettua. Tämä asettaa haasteita monikielisten RAG-järjestelmien kehittämiselle ja käyttöönotolle globaalissa mittakaavassa.

On myös huomioitava, että suurempi määrä ulkoista tietoa ei välttämättä paranna mallin suorituskykyä, sillä liian pitkä konteksti voi aiheuttaa mallin huomion hajaantumisen [7].

Tämä korostaa tietolähteiden huolellisen valinnan ja kontekstin optimoinnin tärkeyttä RAG-järjestelmien implementoinnissa.

Mikään esitellyistä menetelmistä ei myöskään tarjoa täydellistä ratkaisua dynaamisten, nopeasti muuttuvien faktojen käsittelyyn. RAG-järjestelmät toimivat edelleen parhaiten tilanteissa, joissa käsitellään staattisia tai hitaasti muuttuvia faktoja [17], mikä on paradoksaalista, sillä juuri dynaamiset faktat hyötyisivät eniten hakutehostetuista ratkaisuista. Näitä ongelmia ratkovat FLARE ja Iter-RetGen -menetelmät taas vaativat useita hakuja ja prosessointivaiheita, mikä lisää laskentakustannuksia ja voi hidastaa vastausaikoja. Tämä voi rajoittaa menetelmien käyttöä reaaliaikaisissa sovelluksissa, joissa vaaditaan nopeita vasteaikoja.

### 7.3 Tulevaisuuden tutkimussuunnat

RAG-järjestelmien ajallinen yhteneväisyys ja tietokonfliktien ratkaisu on tällä hetkellä aktiivinen mutta vielä kehittyvä tutkimusalue, jossa useita lupaavia lähestymistapoja on esitetty, mutta kattavaa ratkaisua ei ole vielä löydetty erityisesti dynaamisen tiedon käsittelyyn.

Eräs mahdollinen tulevaisuuden tutkimuskohde olisi hybridimenetelmien kehittäminen, jotka yhdistävät eri lähestymistapojen vahvuuksia. Esimerkiksi kontrastiivisen oppimisen periaatteita voitaisiin yhdistää CD2-luottamustason kalibrointiin, mikä mahdollistaisi sekä tehokkaan aikamääritteiden tunnistamisen, että luotettavan päätöksenteon ristiriitaisissa tilanteissa. Riskinä on järjestelmän liiallinen monimutkaistuminen ja suorituskyvyn lasku.

RAG-järjestelmien kehittäjien ja käyttäjien kannalta voisi olla myös hyödyllistä kehittää mekanismeja, jotka tekevät ajallisiin konflikteihin liittyvästä päätöksenteosta läpinäkyvämpää. Tulevaisuuden tutkimus voisi keskittyä kehittämään mallien kykyä selittää, miksi tietty ajallinen tulkinta valittiin tai miksi tietty tietolähde priorisoitiin toisen sijaan.

Lisäksi voisi olla mielenkiintoista kehittää menetelmiä, jota arvioivat faktojen ajallista relevanssia ja muutostodennäköisyyttä. Tämä mahdollistaisi dynaamisemman lähestymistavan, jossa malli voisi arvioida eri faktatyyppien pysyvyyttä ja priorisoida hakuja sen mukaisesti. Luvussa 5.2 mainittiin esimerkki, jossa Alaskan pääkaupunki pysyy todennäköisesti samana seuraavat 20 vuotta, kun taas kuvernöörin ennustaminen on lähes mahdotonta. Hauissa voitaisiin siis priorisoida mahdollisesti kuvernöörin tyyppistä tiheämpään muuttuvaa tietoa, kun taas pääkaupungin tyyppiseen vakaaseen tietoon ei tarvitsisi käyttää yhtä paljon resursseja.

## 8. YHTEENVETO

Tässä tutkielmassa on tarkasteltu hakutehostettua generaatiota käyttävien järjestelmien (RAG) ajallisia haasteita ja tietokonflikteja sekä niiden ratkaisumenetelmiä. RAG-järjestelmissä esiintyy kolmenlaisia keskeisiä tietokonflikteja: konteksti-muistikonflikteja, kontekstin välisiä konflikteja ja muistin välisiä konflikteja [3]. Näiden konfliktien taustalla on kielimallien erilaisten tietolähteiden väliset ristiriidat. Erityisen haastavia ovat dynaamiset, nopeasti muuttuvat faktat, joiden käsittelyssä mallit kohtaavat eniten vaikeuksia, vaikka juuri nämä faktat hyötyisivät eniten hakutehostetuista ratkaisuista [17].

Ajalliset haasteet ilmenevät monin tavoin, kuten ajallisena sovitusrvirheenä, tietolähteiden ajallisena epätasapainona ja päättelykyvyn puutteena aikamääritteiden käsittelyssä [2, 11]. Erityisen merkittävä ongelma on huomiovinouma, joka saa mallit ylikorostamaan tiettyjä tietoja kyselyissä samalla kun ne jättävät huomiotta aikamääritteet [11]. Tämä johtaa usein epätarkkoihin hakutuloksiin ja vastausten epäluotettavuuteen ajallisesti määrittäneissä kyselyissä.

Tutkielmassa tarkastellut tekniset ratkaisumenetelmät FLARE [15], Iter-RetGen [16], CD2 [7] ja kontrastiivinen oppiminen [11] tarjoavat lupaavia lähestymistapoja ajallisten haasteiden ja tietokonfliktien ratkaisemiseen. Erityisen tehokkaaksi on osoittautunut kontrastiivinen oppiminen aikariippuvaisten kyselyjen käsittelyssä, sillä se mahdollistaa ajallisesti merkityksellisten upotusavaruuksien luomisen ja vähentää huomiovinoumaa aikamääritteiden tunnistamisessa [11].

RALM-malleissa esiintyy useita vinoumia, kuten saatavuusvinouma, vahvistusvinouma ja Dunning–Kruger-efekti [7]. Nämä vinoumat vaikuttavat merkittävästi mallien kykyyn käsitellä ristiriitaista tietoa. CD2-menetelmä on osoittautunut lupaavaksi lähestymistavaksi näiden vinoumien ja muiden harhojen vaikutusten vähentämiseen kalibroimalla mallin luottamustasoa tietämysristiriidoissa [7].

Yksittäisten menetelmien sijaan optimaalinen ratkaisu RAG-järjestelmien ajallisiin haasteisiin ja tietokonflikteihin saattaa löytyä eri lähestymistapojen vahvuuksien yhdistämisestä. Erityisen lupaavia tutkimuskohteita ovat hybridiratkaisut, jotka yhdistäisivät kontrastiivisen oppimisen kaltaisia aikamääritteiden tunnistamisen menetelmiä ja CD2:n kaltaisia luottamustason kalibrointimenetelmiä mahdolliseen iteratiiviseen hakuun.

# LÄHTEET

- [1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, *Advances in Neural Information Processing Systems*, Vol. 33, 2020, pp. 9459-9474. Saatavissa: <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- [2] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, W. W. Cohen, Time-aware language models as temporal knowledge bases, *Transactions of the Association for Computational Linguistics*, Vol. 10, 2022, pp. 257-273. Saatavissa: [https://doi.org/10.1162/tacl\\_a\\_00459](https://doi.org/10.1162/tacl_a_00459)
- [3] R. Xu, Z. Qi, Z. Guo, C. Wang, H. Wang, Y. Zhang, W. Xu, Knowledge Conflicts for LLMs: A Survey, *arXiv preprint arXiv:2403.08319*, 2024. Saatavissa: <https://arxiv.org/abs/2403.08319>
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017. Saatavissa: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
- [5] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A Survey on Evaluation of Large Language Models, *ACM Transactions on Intelligent Systems and Technology*, Vol. 15, No. 3, 2024, Article No. 39, pp. 1-45. Saatavissa: <https://doi.org/10.1145/3641289>
- [6] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, L. Qiu, Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely, *arXiv preprint*, 2024. Saatavissa: <https://arxiv.org/abs/2409.14924>
- [7] Z. Jin, P. Cao, Y. Chen, K. Liu, X. Jiang, J. Xu, Q. Li, J. Zhao, Tug-of-War Between Knowledge: Exploring and Resolving Knowledge Conflicts in Retrieval-Augmented Language Models, *arXiv preprint*, 2024. Saatavissa: <https://arxiv.org/abs/2402.14409>
- [8] T. T. Procko, O. Ochoa, Graph Retrieval-Augmented Generation for Large Language Models: A Survey, 2024 Conference on AI, Science, Engineering, and Technology (AlxSET), Laguna Hills, CA, USA, 2024. Saatavissa: <https://ieeexplore.ieee.org/abstract/document/10771030>
- [9] T. Taipalus, Vector database management systems: Fundamental concepts, use-cases, and current challenges, *Cognitive Systems Research*, Vol. 85, 2024, 101216. Saatavissa: <https://www.sciencedirect.com/science/article/pii/S1389041724000093>

- [10] T. Kenneweg, P. Kenneweg, B. Hammer, RAGVAL: Automatic Dataset Creation and Evaluation for RAG Systems, 2024 2nd International Conference on Foundation and Large Language Models (FLLM), Dubai, United Arab Emirates, 2024, pp. 1-10. Saataavissa: <https://ieeexplore.ieee.org/document/10852482>
- [11] F. Wu, L. Liu, W. He, Z. Liu, Z. Zhang, H. Wang, M. Wang, Time-Sensitive Retrieval-Augmented Generation for Question Answering, Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), 2024, pp. 2544-2553. Saataavissa: <https://doi.org/10.1145/3627673.3679800>
- [12] O. Yoran, T. Wolfson, O. Ram, J. Berant, Making Retrieval-Augmented Language Models Robust to Irrelevant Context, arXiv preprint, 2023 (revised 2024). Saataavissa: <https://doi.org/10.48550/arXiv.2310.01558>
- [13] Y. Dong, S. Wang, H. Zheng, J. Chen, Z. Zhang, C. Wang, Advanced RAG Models with Graph Structures: Optimizing Complex Knowledge Reasoning and Text Generation, 2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC), Wuhan, China, 2024. Saataavissa: <https://ieeexplore.ieee.org/document/10810209>
- [14] L. Gao, X. Ma, J. Lin, J. Callan, Precise Zero-Shot Dense Retrieval without Relevance Labels, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers, 2023, pp. 1762-1777. Saataavissa: <https://aclanthology.org/2023.acl-long.99.pdf>
- [15] Z. Jiang, F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, G. Neubig, Active Retrieval Augmented Generation, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 7969-7992. Saataavissa: <https://doi.org/10.18653/v1/2023.emnlp-main.495>
- [16] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, W. Chen, Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy, Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 9248-9274. Saataavissa: <https://doi.org/10.18653/v1/2023.findings-emnlp.620>
- [17] S. V. Marjanovic, H. Yu, P. Atanasova, M. Maistro, C. Lioma, I. Augenstein, DYNAMICQA: Tracing Internal Knowledge Conflicts in Language Models, arXiv preprint, 2024. Saataavissa: <https://arxiv.org/abs/2407.17023>
- [18] S. Simon, A. Mailach, J. Dorn, N. Siegmund, A Methodology for Evaluating RAG Systems: A Case Study On Configuration Dependency Validation, arXiv preprint arXiv:2410.08801, 2024. Saataavissa: <https://arxiv.org/abs/2410.08801>