

Loppuraportti - Lausuntoyhteenvetojen tuottamisen palveludemo lainvalmistelutyön tueksi (Valtioneuvoston kanslia)

Projektin tausta ja tavoitteet	2
Projektin toteutus	2
Vaatimusten määrittely ja haastattelut	2
Haastatteluiden löydökset	2
Demo-ratkaisun kehittäminen	3
Tulosten arviointi ja palautteen kerääminen	3
Ratkaisun kuvaus	4
Demoratkaisun ominaisuudet	4
Yhteenvedon tuottamisen osavaiheet	8
Tulokset ja johtopäätökset	9
Käyttäjäpalautte	9
Johtopäätökset ja huomiot	9
Liitteet	12
Terminologia	12

Projektin tausta ja tavoitteet

Valtioneuvoston kanslia hakee tekoölyavusteista työkalua, jota käytettäisiin lausuntoyhteenvedon tuottamisen apuna, osana lainvalmistelutyötä. Tämän kokeilun tavoitteena on kehittää lainvalmistelutyötä helpottava tekoölyavusteinen työkalu, jolla voidaan testata ja varmentaa miten generatiivista tekoölyä voidaan hyödyntää tyyppillisissä lainvalmisteluun liittyvissä työtehtävissä, kuten lausuntoyhteenvedon tuottamisessa. Alla on tiivistetty projektin pääasialliset tavoitteet:

- Projektin tavoitteena on tehdä kokeilu, jossa toteutetaan tekoölyn avulla demoratkaisu lausuntoyhteenvedosta.
- Tarkoituksena on, että tekoölyn avulla tuotetaan luonnos lausuntoyhteenvedoksi, jonka valmistelija viimeistelee loppuun.
- Kokeilu tehdään lainvalmisteluun liittyvällä lausunto-aineistokokonaisuudella, johon kuuluu lausuntopyyntöjä ja niihin annettuja lausuntoja.
- Demoratkaisun avulla kokeillaan vähintään neljän lausuntoyhteenvedon tuottamista neljään lausuntopyyntöön annettujen lausuntojen perusteella.
- Kokeilussa saatuja oppeja huomioidaan tulevissa ICT-hankinnoissa ja lausuntokierrosten toteuttamisen kehittämisessä

Projektin toteutus

Vaatimusten määrittely ja haastattelut

Vaatimusten määrittelyn tavoitteena oli ymmärtää paremmin käyttäjien työprosessia ja tarpeita ja näin ollen toimia perustana ratkaisun kehittämiseksi. Tavoitteena oli selvittää työskentelyprosessi eli miten lainvalmistelussa tuotettu yhteenvedo luodaan, miltä yhteenvedon tulisi näyttää ja mitä lainvalmistelijat toivovat tekoölyavusteiselta työkalulta. Vaatimukset kerättiin haastattelemalla kahta avainhenkilöä: Jyrki Jauhiainen, joka toimii lainsäädäntöneuvoksena Oikeusministeriön lainvalmisteluosastolla ja Susanna Siitonen, joka toimii lainsäädäntöjohtajana oikeusministeriössä.

Haastatteluiden löydökset

Lausuntokierroksen ja tiedon rakenteen osalta on tärkeää varmistaa, että kerätyt tiedot ovat jäsenneiltyjä ja helposti analysoitavissa, jotta niistä saadaan mahdollisimman paljon hyötyä. Monivalintakysymykset voivat auttaa saamaan tarkempia ja jäsenneilympiä vastauksia, ja jäsenneilymmät lausuntokierrokset voivat auttaa saamaan selkeämmän kuvan yleisestä mielipiteestä. Tekoölykokeilua voidaan käyttää näiden parannusten vaikutusten todentamiseen, mikä voi auttaa osoittamaan näiden muutosten tuoman lisäarvon.

Lausuntokierroksen tulosten sisällön ja laadun osalta on tärkeää keskittyä tunnistamaan, mikä ehdotetussa lakipykälässä on onnistunutta ja mitä ongelmia siinä on. Tämä auttaa varmistamaan, että

palaute on hyödyllistä ja että sitä voidaan käyttää tehokkaasti lain muuttamiseen tai poistamiseen. Lisäksi on tärkeää keskittyä palautteen laadulliseen sisältöön, jotta voidaan tunnistaa, millä perusteella kannatetaan tai vastustetaan tiettyä ehdotusta.

Tekoälytyökalun osalta on tärkeää, että se ei ainoastaan tuota lopullista yhteenvetoa, vaan myös tuottaa listauksen huomioista ja esimerkkipalautteista, jotka on syytä ottaa huomioon. Tämä auttaa varmistamaan, että kaikki olennaiset tiedot on otettu huomioon ja että palaute on mahdollisimman hyödyllistä. Lisäksi työkalun tulisi pystyä toimimaan ihmisen apulaisena eikä tuottaa valmiita asiakirjoja, jolla varmistetaan, että se on mahdollisimman hyödyllinen ja että se vastaa käyttäjien tarpeita.

Demo-ratkaisun kehittäminen

Jokaisesta lausuntokierroksesta kerättiin CSV-tiedosto, joka sisältää kaikki lausuntokierroksen kommentit taulukkomuodossa. Näitä kommentteja, kommenttien kirjoittajia sekä muita metatietoja annetaan yhdessä ohjeen (esim. "Luo seuraavasta kommentista tiivistelmä, joka..." tai "Tunnista seuraavasta kommentista kirjoittajan kanta.") kanssa suurelle kielimallille.

Ylläolevan lisäksi, pohjamallia kehitettiin few-shot prompting menetelmän avulla. Few-shot prompting tarkoittaa sitä, että suurille kielimalleille näytetään käyttäjän kysymyksen lisäksi muita esimerkkejä, jotka sisältävät kommentteja ja niiden mallitiivistelmiä. Tämä auttaa kielimallia ymmärtämään varsinaista tehtävää (kommenttien tiivistäminen, kannan tunnistaminen, jne.) ja auttaa kielimallia vastaamaan käyttäjän alkuperäiseen kysymykseen esimerkkien mukaisesti. Projektin aikana on käytetty sekä todellisia esimerkkejä muista lausuntokierroksista että itse kirjoitettuja esimerkkejä jotta saatiin mahdollisimman kattava aineisto.

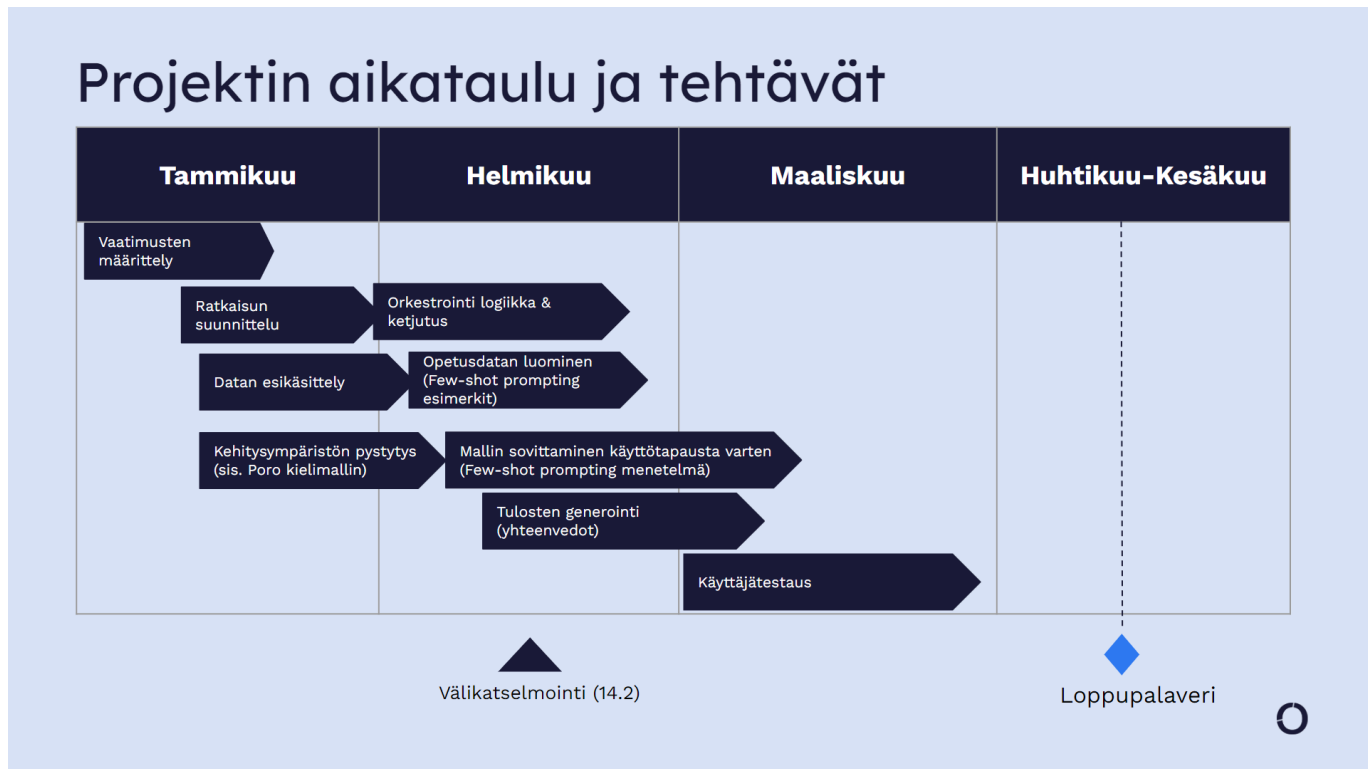
Tulosten arviointi ja palautteen kerääminen

Tekoälyn tuottama lausuntoyhteenveto arvioitiin ja palautetta kerättiin usealta projektiin mukaan valitulta arvioijalta, joka edusti lausuntokierrosta. Arvioijat saivat luonnoksen pdf-muodossa tuotetusta yhteenvetodokumentista ja heitä pyydettiin arvioimaan sen rakennetta, sisältöä ja kieltä sekä ilmoittamaan kaikista havaitsemistaan virheistä. Lisäksi heiltä pyydettiin arviota siitä, kuinka paljon ihmistyötä luonnoksen viimeistely vaatisi ja kuinka paljon aikaa tekoälyn tuottama luonnos olisi voinut säästää. Palautetta käsiteltiin yksityiskohtaisemmin viikoittaisissa kokouksissa, joihin osallistui sekä projektiin tiimi sekä arvioijat. Arvioijat kutsuttiin osallistumaan kokouksiin 25. maaliskuuta ja 8. huhtikuuta keskustelemaan havainnoistaan ja kommentteistaan.

Seuraavat lausuntokierrokset olivat mukana tulosten arvioinnissa:

- Esitys asunto-osakeyhtiölain ja toiminimilain 7 §:n muuttamisesta
- Arkeologinen kulttuuriperintö
- Luonnos hallituksen esitykseksi rakentamislain muuttamisesta
- Luonnos hallituksen esitykseksi laiksi jätelain muuttamisesta
- Työryhmän arviomuistiolounnos: data retention -lainsäädännön täsmentämisen vaihtoehtoista

Lausuntokierrokset erosivat rakenteeltaan siten, että esimerkiksi Arkeologisen kulttuuriperinnön lausuntokierros ei sisältänyt eri aihekohtia, vaan lausunnonantajat kirjoittivat kommenttinsa vain yhteen kenttään. Näissä rakenteistamattomissa lausuntokierroksissa luotiin aihekohdat tekoälyn avulla. Aihekohtien luontia varten kerättiin alkuperäinen taustamateriaali, josta poimittiin lakipykälät ja niiden ehdotetut muutokset. Sitten tekoäly teki semanttisen vertailun jokaisen lausunnon kappaleen ja lakipykälien välillä. Semanttisesti eniten vastaava lakipykälä kertoo, mihin aihekohtaan kommentin kappale kuuluu.



Kuva: Projektin aikataulu ja päävaiheet

Ratkaisun kuvaus

Tässä kappaleessa on kuvattu projektissa kehitetyn demoratkaisun ominaisuudet ja toimintaperiaate.

Demoratkaisun ominaisuudet

Demoratkaisu luo .pdf -tiedoston, joka sisältää useita kappaleita ja tiivistelmiä, jotka ovat kielimallin tuottamia. Tiedosto mallintaa muista lausuntokierroksista tehtyjä esimerkitiivistelmiä. Jokaisen aihekohdan tiivistelmän mukana kerrotaan lausunnonantajat ja heidän kannatusluokat (Kannatan muutettuna, Vastustan, jne.). Tekoäly- ja kielimallit eivät siis luo yksittäistä tiivistelmätiedostoa kerralla, vaan tiivistävät yksittäisiä kommentteja tai niputettuja kommentteja kappaleiksi, jolloin nämä kappaleet, tunnistetut kannat sekä aihekohdat kirjoitetaan Python -skriptillä haluttuun rakenteeseen.

Jos aihekohtia ei ole alkuperäisessä .csv -tiedostossa, nämä tunnistetaan tekoälymallien kanssa. Tällaisia lausuntokierroksia kutsutaan rakenteistamattomiksi lausuntokierroksiksi, sillä kommentit ovat kerätty vain yhteen kenttään ilman, että lausuntokierroksessa on minkäänlaista rakennetta.

Aihekohdat tiivistetään niin, että kielimallille näytetään kaikki saman aihe- sekä kannatusluokan kommentit (esim. Aihe A ja kaikki "Vastustan" kommentit). Tekoäly tuottaa näiden niputettujen kommenttien perusteella kappaleen. Jos kommentteja on niin monta tai ne ovat liian pitkiä, kommentteja jaetaan kahteen osaan kunnes kaikki jaetut osat mahtuvat yksinään syötteeseen.

Aihekohtaisten tiivistelmien lisäksi tuotetaan johdon yleinen yhteenveto, joka tuotetaan sitä varten räätälöidyllä syötteellä. Tätä syötettä varten tiivistetään ensin jokainen kommentti. Aihekohta kerrallaan, jokainen kommentin tiivistelmä annetaan syötteessä johdon yhteenvedon tuottamista varten. Näin jokaisesta aihekohdasta saadaan yksi kappale. Näin saadaan yleensä 4-5 kappaletta, joista jokainen käsittelee yhtä aihekohtaa.

Kun tuotetaan aihekohtaisia tai yksittäisten kommenttien tiivistelmiä, kommenttien lisäksi kielimallille näytetään niiden kannatusluokka sekä lausunnonantajien tai organisaatioiden nimet.

Kielimallia käytetään myös kannatusluokkien tunnistamiseen. Tuetut kannatusluokat ovat *Kannatan*, *Vastustan* ja *Ei kantaa*. Ennen kuin kommentti annetaan kielimallille luokiteltavaksi, katsotaan, kuuluuko kommentti Ei kantaa -luokkaan käyttäen semanttista vertailua ja tekstinkäsittelyä (onko kommentti tarpeeksi lyhyt ja onko siinä "ei kommentoitavaa/lausuttavaa" rakenne, jne.). Lausunnonantajat antavat siis joskus esimerkiksi kommentin: "Toteamme, että ei ole lausuttavaa." Jos kommentti ei ole tällainen, se annetaan kielimallille tunnistettavaksi niin, että kielimalli tuottaa kannan.

Tiivistelmien sekä kannatusluokkien tuottamisessa käytetään few-shot prompting tekniikkaa. Tällöin syötteeseen sisältyy (kommenttien ja muiden metatietojen lisäksi) myös few-shot (ks. kohdasta Termit "few-shot prompting") esimerkkejä. Few-shot esimerkit ovat malliesimerkkejä, jotka sisältävät syötteen sekä halutun ulostulon. Näyttämällä kielimallille tällaisia esimerkkejä, malli oppii toistamaan tehtävää ja antamalla halutunlaisen ulostulon.

Seuraavalla sivulla on esimerkki aihekohtaisesta tiivistelmästä, jossa näkyy lista lausunnonantajista ja heidän kannatuksista, sekä osa myönteisten kommenttien tiivistelmätekstistä,

2.2 Muutosehdotuksen otsikko

Muutosehdotus: Asunto-osakeyhtiölakia ehdotetaan muutettavaksi siten, että tavallisella enemmistöpäätöksellä voi päättää myös sellaisesta yhtiön kiinteistön tai rakennuksen käytön kestävyttä merkittävästi parantavasta uudistuksesta, joka on vasta yleistymässä uustuotannossa (AOYL 6 luvun 31 §:n 2 momentin uusi 3 kohta).

Lausunnonantajien kannat:

Ehdotusta **kannattavat sellaisenaan** Oulun kaupunki, asuntotoimenpäällikkö, yhdyskunta- ja ympäristöpalvelut, Pohjois-Pohjanmaan liitto, maakuntajohtaja, yritys juridiikan professori, Tampereen yliopisto, Vammaisfoorumi ry, Invalidiliitto ry, Yhteiskuntasuhteet -yksikkö, Vanhustyön keskusliitto - Centralförbundet för de gamlas väl ry, Vuokralaiset VKL ry, Suomen Isännöintiliitto ry, Suomen Yrittäjät ry, Suomen Asianajajaliitto, Elinkeinoelämän keskusliitto EK, Näkövammaisten liitto ry, Osuustoimintakeskus Pellervo ry, Kynnys ry

Ehdotusta **kannattavat muutettuna** Ympäristöministeriö, RYMO, Rakennusteollisuus RT ry, Suomen Vuokranantajat ry, Suomen Kiinteistöliitto ry, Kuluttajaliitto ry, Konsumentförbundet rf

Dahlström Jorma, Brummer Simo **vastustavat ehdotusta.**

Liikenne- ja viestintäministeriö, Maanmittauslaitos, Pääjohtaja Arvo Kokkonen ja maanmittausneuvos Markku Markkula, Finanssiala ry, Johtaja Hannu Ijäs, Ulosottolaitos, Ammattiliitto Pro ry, Yhdenvertaisuusvaltuutettu, Suomen Kuntaliitto ry **eivät ota kantaa ehdotukseen.**

Ehdotukseen myönteisesti suhtautuvien lausunnonantajien perustelut ja kehittämissuhteet:

Lausunnonantajat pitivät lakiehdotusta perusteltuna, mutta esittivät huolensa merkittävyyden kriteerin monitulkintaisuudesta ja sen mahdollisesti aiheuttamista tulkintaongelmista sekä riittäisyydestä. Osa lausunnonantajista toivoi esteettömyyden parempaa huomioimista sähköautojen latauspistokepaikkojen

4 / 9

mitoituksessa. Muutamit lausunnonantajat toivoivat myös lain perustelujen tarkentamista kestävyden ja merkittävyyden osalta.

Lausunnonantajat pitivät ehdotettua muutosta pääosin kannatettavana. Ehdotettu muutos nähtiin järkevänä ja tarpeellisena, sillä se helpottaisi kestävien uudistusten toteuttamista vanhoissa asunto-osakeyhtiöissä. Muutosta pidettiin myös teknologianeutraalina.

Kestävän asumisen määrittäminen voi kuitenkin olla haastavaa ja jäädä tulkinnanvaraiseksi. Tämän vuoksi on tärkeää kytkeä määrittely esimerkiksi samankaltaisten uudistusten yleisyyteen vastaavassa rakentamisessa. Sähköautojen latauspisteitä ja latausjärjestelmiä koskevan päätöksenteon ja kustannustenjaon osalta nähtiin välttämättömänä selkeyttä tilannetta. Myös aurinkosähkön ja muun energian paikallisen tuotannon osalta nähtiin tarpeelliseksi mainita selkeämmin, miten tuotettu sähkö hyödyttää osakkeenomistajia.

Alla on osa syötteestä, joka näytetään Poro -kielimallille. `{input}` korvataan varsinaisella kommentilla, josta halutaan tunnistaa kanta. Allaolevia few-shot esimerkkejä on todellisuudessa yhdeksän.

Tehtävänäsi on luokitella kommentteja. Käytä ainoastaan niitä luokkia, jotka on annettu ennen kommenttia. Älä käytä mitään muuta tekstiä. Esimerkiksi:

ESIMERKIT ALKAA

1. Kannatan

2. Vastustan

Perustelu:Ehdotus, joka vaatii kaikilta kouluilta päivittäin tarjottavaa kasvisruokavaihtoehtoa, on askel kohti kestävämpää ja terveellisempää ruokakulttuuria. Kasvisruokavalion edistäminen kouluissa voi vaikuttaa positiivisesti lasten ruokailutottumuksiin ja tukea ympäristön suojelua. Lisäksi se tarjoaa kaikille oppilaille mahdollisuuden tutustua monipuolisempiin ruokavalintoihin, mikä on erityisen tärkeää kasvavassa iässä. Muuttaisimme silti kohtaa, jossa kasvisruokaa tarjoiltaisiin neljänä päivänä viikossa, ja ehdottaisimme sen tarjottavan myös perjantaisin.

Kanta:Kannatan DONE

1. Kannatan

2. Vastustan

Perustelu:Ehdotettu laki sähköisten viestintien valvonnasta työpaikoilla loukkaa työntekijöiden yksityisyyttä ja on vastoin perusoikeuksia. Se saattaa lisätä epäluottamusta työyhteisöissä ja vähentää työntekijöiden motivaatiota. Vaikka työnantajilla on oikeus suojella yrityksen omaisuutta ja varmistaa työntekijöiden tuottavuus, tämä ei saisi tapahtua yksityisyyden kustannuksella.

Kanta:Vastustan DONE

1. Kannatan

2. Vastustan

Perustelu:Ehdotus kieltää yksityisautoilun kaupungin keskustassa ja edistää näin julkisen liikenteen käyttöä sekä kävelyä ja pyöräilyä. Tämä ei ainoastaan vähennä ilmansaasteita ja liikenteen ruuhkia, vaan myös parantaa kaupunkilaisten elämänlaatua tarjoamalla turvallisempia ja miellyttävämpiä kaupunkitiloja.

Kanta:Kannatan DONE

ESIMERKIT LOPPUU

Nyt tehtäväsi on luokitella seuraava kommentti:

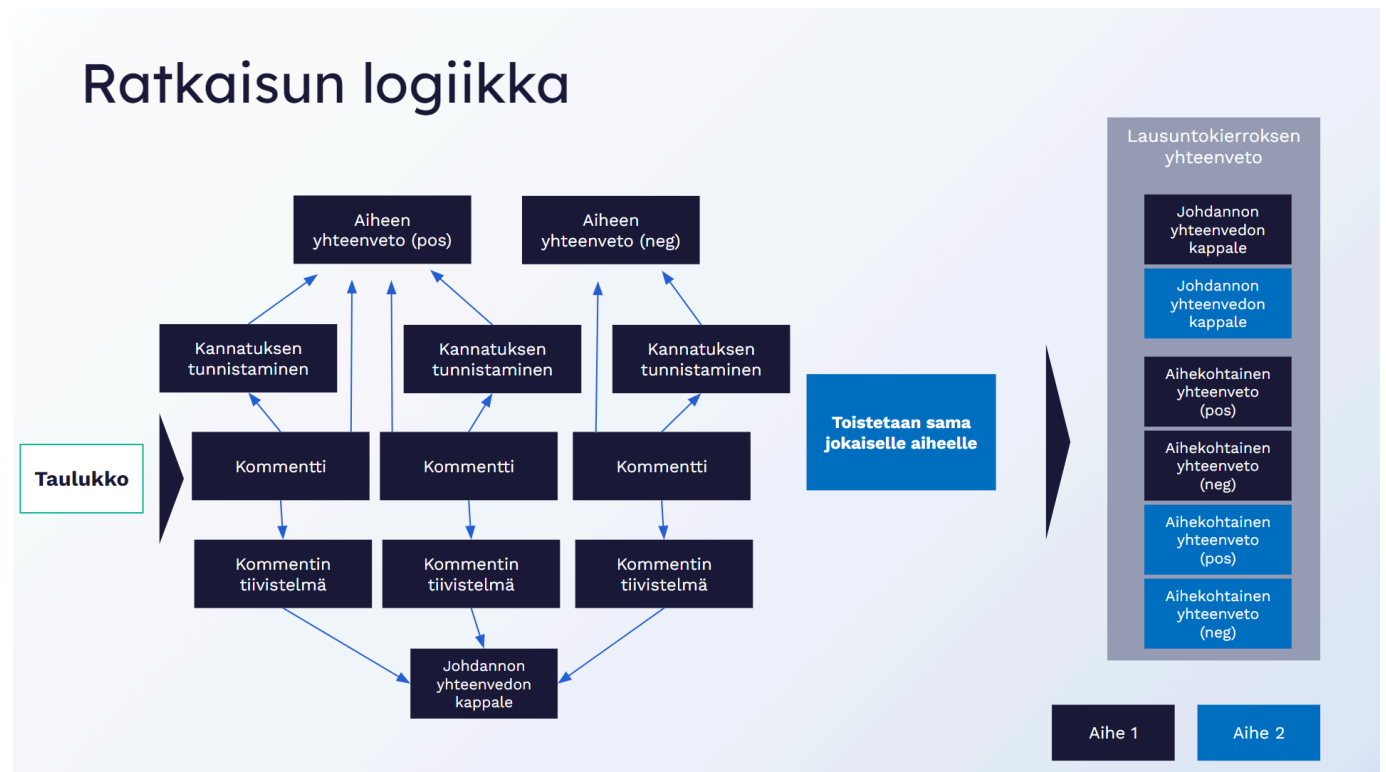
1. Kannatan

2. Vastustan

Perustelu:`{input}`

Poro -kielimallin kanssa käytetään siis few-shot prompting tekniikkaa sen sijaan, että kielimalli oltaisiin jatkokoulutettu lausuntoaineistolla. Kielimallien jatkokoulutuksella malli erikoistuu tiettyyn tehtävään, ja tähän tarvitaan tyypillisesti useita tuhansia esimerkkejä, kun taas few-shot -tekniikkaan yleensä riittää muutamasta kymmeneen malliesimerkkiä.

Semanttiseen vertailuun käytetään TurkuNLP:n [Finnish SBERT paraphrasing mallia](#). Tällä mallilla voidaan vertailla lauseiden semantiikkaa, ja nähdä kuinka samanlaisia lauseet ovat sisällön suhteen. Kommenttien tekstikäsittelyyn, lauseisiin tai kappaleiden pilkkomiseen, ym. Käytetään SpaCy kirjastoa, sekä valmiiksi koulutettua [fi_core_news_sm](#) mallia.



Kuva: Ratkaisun toimintaperiaate - yhteenvetön tuottaminen

Yhteenvetön tuottamisen osavaiheet

Ensimmäiseksi Python -skripti lukee taulukon, joka sisältää kommentit ja tiedon siitä, kuka on kirjoittanut kommentin. Taulukon kommentit ovat jaettu sarakkeisiin niin, että yksi sarake edustaa yhtä aihekohtaa. Näiden sarakkeiden nimet luetaan aihekohtien nimiksi. Esimerkiksi, Valtionneuvoston kanslia on kirjoittanut kommentit aihekohtiin A, B ja D, mutta on jättänyt C -sarakkeen tyhjäksi.

Otetaan ensimmäinen sarake käsittelyyn. Ensimmäiseksi, jokaisesta kommentista tehdään tiivistelmä. Nämä kommenttien tiivistelmät annetaan syötteenä kielimallille, jota ohjeistetaan muodostamaan johdon yhteenveto. Malli tuottaa tiivistelmistä sekä johdon yhteenvetosta kappaleen. Tämä toistetaan jokaiselle aihekohdalle.

Kommenteista myös tunnistetaan kanta, joka annetaan kommentin sekä muiden saman kannan kommenttien kanssa syötteenä kielimallille. Syötteestä - joka sisältää esimerkiksi kaikki "Vastustan"

kommentit yhdestä aihekohdasta - tuotetaan aihe- ja kantakohtainen tiivistelmä. Tämä toistetaan myös jokaiselle aihekohdalle.

Kun kaikki tiivistelmät ja tekstit ovat tuotettu, ne kirjoitetaan tekstitiedostoon haluttuun rakenteeseen. Ennen jokaista aihekohtaa listataan myös kaikki lausunnonantajat ja heidän kannat.

Tulokset ja johtopäätökset

Tässä kappaleessa käsitellään tulokset, ratkaisun rajoitukset ja ratkaisuehdotukset näiden korjaamiseksi.

Käyttäjäpalautte

Tässä kappaleessa käsitellään tiivistetysti käyttäjiltä tulleet palautteet.

Lausuntopyyntöä yhteenvedossa käytetty demoratkaisu on osoittanut lupaavia tuloksia, tarjoten arvokasta pohjadataa ja alustavia tiivistelmiä lausunnonantajalle. Tekoälytyökalu on kyennyt tehokkaasti käsittelemään suuria määriä dataa ja tuottamaan johdonmukaisia ja lausuntopyyntöä rakennetta noudattavia tuloksia.

Yhteenvedossa käytetty kieli on pääosin selkeää ja ymmärrettävää, ja ihmisen tekemä jatkokäsittely kohdistuisi lähinnä sisällön parantamiseen.

Demoratkaisu on kuitenkin tuottanut myös virheellisiä ja puutteellisia tulkintoja lausuntopalautteesta, mikä on vaikeuttanut eri lausuntopyyntöä kysymysten vastauksiin liittyvien palautteiden erottamista toisistaan erityisesti tiivistelmäosiossa.

Yksi merkittävimmistä haasteista on ollut ratkaisun taipumus yleistää palautetta ja jättää huomiotta lausunnonantajien voimassa olevaa lainsäädäntöä koskevat kommentit ja lausuntopyyntöä tarkoitettua ehdotusta täydentävät kommentit. Tämä osoittaa, että tekoälyratkaisu tarvitsee lisäkoulutusta ja hienosäätöä, jotta se voi erottaa ja käsitellä eri tyyppisiä lausuntoja tehokkaasti.

Yhteenvedon yhteydessä, demoratkaisu on osoittanut potentiaalia toimia avustajana lausunnonantajien työn helpottamisessa, mutta se tarvitsee vielä ihmisen ohjausta ja valvontaa toimiakseen tehokkaasti. Aito ja hyödyllinen testiympäristö vaatisi ihmisen osallistumista ja tekoälyratkaisun parempaa ohjausta, jotta se voi oppia ja sopeutua lausuntoihin liittyviin erityisiin vaatimuksiin ja kontekstiin.

Johtopäätökset ja huomiot

Tässä kappaleessa käsitellään ratkaisun rajoitukset, päähuomiot ja ratkaisuehdotukset näiden korjaamiseksi.

Kielimallin rajallinen konteksti-ikkuna

Kielimallin konteksti-ikkuna vaikuttaa siihen kuinka monta sanaa kielimalli pystyy käsittelemään kerralla. Tähän lasketaan sekä syöte että tuotettu teksti. Kommentteja ja muuta kontekstia käytetään antamaan lisää tietoa ja taustaa kielimallille. Tämä on tärkeää, jotta malli kykenee tuottamaan tarkkoja ja asiaankuuluvia tiivistelmiä. Lyhyt konteksti-ikkuna ei mahdollista kaikkien kommenttien ja muun taustatiedon syöttämistä mallille, jolloin mallilta puuttuu tarvittava taustatieto ja tiivistämiseen vaadittavat kommentit.

Kielimallin rajallinen konteksti-ikkuna rajoittaa tiivistettävän tekstin määrää, mistä syystä kommentteja on pitänyt supistaa. Kielimallin konteksti-ikkuna ei mahdollista kaikkien kommenttien ja muun kontekstin syöttämistä mallille, jolloin mallilta puuttuu tarvittava taustatieto ja samaan aiheeseen liittyvät kommentit. Kontekstia käytetään myös mallin “maadoittamiseen” jolloin voidaan vaikuttaa vastauksen oikeellisuuteen. Maadoittaminen viittaa prosessiin, jossa annetaan mallille asiaankuuluvaa lakikontekstia, jotta se voi tuottaa tarkempia ja täsmällisempiä vastauksia.

Toinen merkittävä haaste on kielimallin keksityt vastaukset ja omat tulkinnat (“hallusinoiminen”) johtuen siitä että malli ei ole tarpeeksi “maadoitettu” relevantilla lakikontekstilla. Hallusinaatiot viittaavat kielimallin tuottamiin vastauksiin, jotka eivät ole relevantteja tai tarkkoja, koska mallilla ei ole tarpeeksi kontekstia.

Suosituksena on jatkossa käyttää suuremman konteksti-ikkunan kielimallia (esim. Viking-malli tai vastaava), jotta mallille voidaan antaa suurempi määrä sisältöä syötteenä mikä parantaisi sen tarkkuutta ja luotettavuutta. Esimerkiksi malli ymmärtäisi paremmin nykyistä lainsäädäntöä ja osaisi erotella lakiehdotuksen ja ehdotetut muutosehdotukset lakiin.

Suosituksena on myös luoda yhteenveto vaiheistetusti pienemmissä osissa, jotta kommentit mahtuvat mallin konteksti-ikkunaan. Vaiheittainen yhteenveto viittaa prosessiin, jossa kommentit jaetaan pienempiin osiin, jotta ne mahtuvat kielimallin konteksti-ikkunaan.

Kielimallin kyvykkyys ja erikoistuneet mallit

Lausuntoyhteenvedon tuottaminen on hyvin erikoistunut käyttötapaus, jossa pelkän tekstin tiivistäminen ei riitä, vaan ratkaisun tulisi tunnistaa olennaiset kohdat kommentteista, esimerkiksi lausunnonantajan muutosehdotukset suhteessa nykyiseen lakiin. Projektissa käytetty yleinen kielimalli ei sovellu tähän tarkoitukseen, vaan vaatii erikoistuneemman mallin, joka on koulutettu näiden tehtävien tunnistamiseen tarvittavalla opetusaineistolla, jonka avulla se oppii tunnistamaan kannatukset, perustelut ja muutosehdotukset.

Kaikkea opetusaineistoa ei tarvitse kerätä ennakkoon vaan mallia voidaan opettaa ja kehittää jatkuvasti keräämällä käyttäjäpalautetta jota käytetään opetusaineistona. Tämä tapahtuu siten että käyttäjät voivat arvioida tekoälyn tuottamia vastauksia ja korostaa kommentteista olennaisia kohtia (kannatus, muutosehdotus, perustelu) joita käytetään sitten mallin kouluttamiseen. Tämä jatkuvan oppimisen lähestymistapa mahdollistaa mallin sopeutumisen uusiin tilanteisiin ja parantaa sen tarkkuutta ajan myötä.

Suosituksena olisi myös käyttää enemmän lakiainestoa seuraavien suomenkielisten kielimallien kouluttamisessa jotta kielimalli osaisi tuottaa sujuvampaa lakitekstiä huomioiden lakitekstin erityispiirteet ja tyyli.

Tekoälyn ohjeistaminen

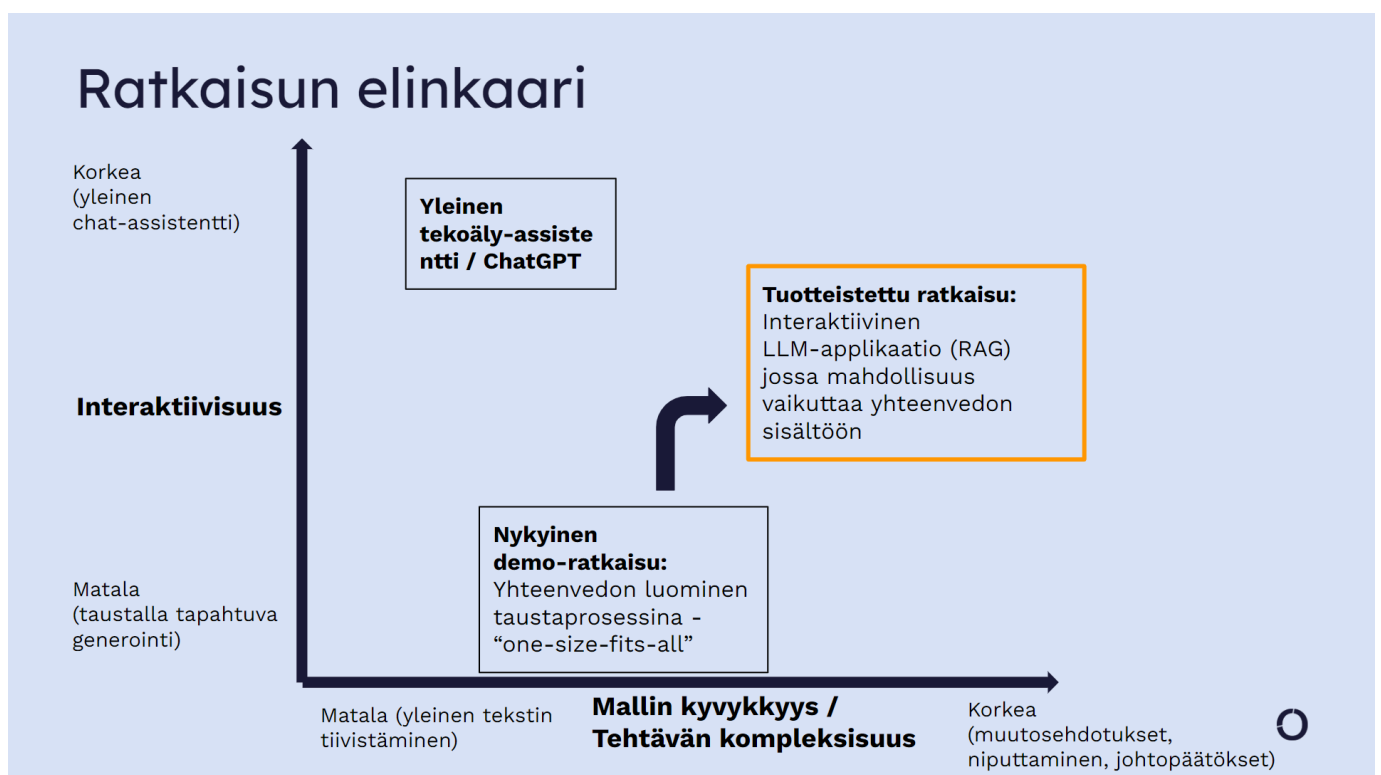
Nykyinen demo-ratkaisu tuottaa koko yhteenvedon kerralla taustalla mikä kestää tällä hetkellä noin

80-160 minuuttia riippuen kommenttien määrästä ja pituudesta. Tästä syystä nykyinen demo-ratkaisu ei mahdollista yhteenvedon ohjaamista haluttuun suuntaan eikä huomioi lausuntokierroksen erityispiirteitä.

Nykyinen ratkaisu ei myöskään mahdollista alkuperäisten kommenttien linkittämistä yhteenvedoon, mikä johtaa siihen, että tiivistettäessä menetetään tietoa eikä nähdä lähdeaineistoa.

Jotta nämä ylläolevat ongelmat voitaisiin ratkaista, tarvitaan interaktiivisempi ratkaisu jossa on mukana hakutoiminnallisuus ja tietokanta. Tämä voisi sisältää RAG (Retrieval Augmented Generation) -tyyppisen ratkaisun, jossa valmistelija voi hakea tietoja ja ohjata tekoälyä tuottamaan halutunlaisen yhteenvedon, joka huomioi lausuntokierroksen erityispiirteet. RAG-tyyppinen ratkaisu mahdollistaa alkuperäisten kommenttien linkittämisen yhteenvedoon referenssien kautta, joten tiivistettäessä ei menetetä tietoa alkuperäisiin lähteisiin.

Lisäksi ratkaisun ohjaamista voidaan parantaa kehittämällä parempia kehoitteita (tunnetaan myös nimellä kehoitteiden suunnittelu). Tämä tarkoittaa sellaisten kehoitteiden luomista, jotka ohjaavat tekoälyä tuottamaan halutunlaisen yhteenvedon, joka huomioi lausuntokierroksen erityispiirteet.



Kuva: Ratkaisun interaktiivisuutta tulisi jatkossa lisätä jotta käyttäjä pysyy paremmin ohjeistamaan ratkaisua ja antamaan palautetta mallin jatkokehitystä varten.

Datan laatu & rakenne on avainasemassa

Lausuntokierrokset, jotka on valittu tähän projektiin, eroavat toisistaan tiedon rakenteen osalta, mikä vaikuttaa merkittävästi projektin lopputuloksiin. Kommentteja tarkasteltaessa on havaittu, että joissakin tapauksissa lausuntoihin sisältyy tarpeetonta johdantoa lakiehdotuksesta, mikä voi tehdä kommentteista vaikeampia tulkita ja analysoida.

Jotta tekoälyä voitaisiin hyödyntää tehokkaasti tässä projektissa, on tärkeää yhdenmukaistaa lausuntoaineisto. Tämä tarkoittaa kommenttien ja niiden sisältämien kannatusluokkien jäsentämistä ja

yhdenmukaistamista. Tämä mahdollistaisi tarvittavan koulutusaineiston tehokkaamman keräämisen, mikä parantaa kielimallin kykyä analysoida ja tulkita kommentteja.

Lisäksi, olisi suositeltavaa selkeyttää kysymysasettelua tai antaa vastaajille ohjeita siitä, miten heidän tulisi muotoilla vastauksensa. Tämä auttaisi vähentämään turhaa jaarittelua ("kohinaa") ja varmistaisi, että vastaukset ovat selkeitä ja ytimekkäitä, mikä helpottaisi niiden koneellista analysointia ja tulkintaa.

Tämän projektin onnistumisen varmistamiseksi on tärkeää ottaa huomioon lausuntokierrosten rakenne, yhdenmukaistaa aineisto ja selkeyttää kysymysasettelua tai antaa ohjeita vastaajille. Näin varmistetaan, että saadut tiedot ovat selkeitä, ytimekkäitä ja helposti tekoälyn analysoitavissa. Lisäksi ehdotetaan, että seuraavassa pilottivaiheessa keskitytään rakenteistettuihin lausuntokierroksiin, jolloin nämä lausunnot ovat helpommin käsiteltäviä tekoälyn kannalta, mikä mahdollistaa tehokkaamman tietojen analysoinnin ja tulkinnan.

Liitteet

Terminologia

Retrieval Augmented Generation (RAG)

Suurta generatiivista kielimallia avustetaan älykkäällä haulla. Kun käyttäjä kysyy kysymyksen kielimallilta, käytetään älykstä hakuja vastauksen tiedon löytämiseen. Tämä tieto annetaan kielimallille kysymyksen kanssa kontekstina, jolloin kielimalli hyödyntää vastauksessaan annettua tietoa. Esimerkiksi:

Käyttäjän kysymys:

Milloin on Sauli Niinistön syntymäpäivä?

Tieto:

Sauli Väinämö Niinistö (s. 24. elokuuta 1948 Salo) toimi Suomen tasavallan 12. presidenttinä kahden kauden ajan vuosina 2012–2024.

Vastaus:

Sauli Niinistön syntyi 24. elokuuta 1948, Salossa.

RAG -ratkaisuja käytetään siihen, että malli vastaa kysymyksiin annetun oikean tiedon perusteella, eikä "hallusinoi" (keksi omia vastauksia sekä tietoa).

RAG:n älykäs haku perustuu semanttiseen hakuun, eli verrataan kysymyksen semantiikkaa tietokantaan. Tietokannasta haetaan tulos, joka eniten vastaa kysymyksen semantiikkaa. Esimerkiksi *auto* ja *rekka* ovat semanttisesti enemmän lähekkäin, kuin *auto* ja *talo*.

Chunking:

Chunking viittaa prosessiin, jossa tietokannan tekstit jaetaan pienempiin osiin. Näitä osia kutsutaan "chunkeiksi". Chunkit tarjoavat kontekstia ja avustavat kielimallia sen vastauksissa. Tekstejä voidaan jakaa chunkeiksi esimerkiksi lauseittain, tai vaikka niin, että jokainen chunkki sisältää enintään 200 merkkiä tai jonkun määrän sanoja.

Guardrails:

“Guardrailit” ovat menetelmiä, joilla voidaan minimoida erilaisia riskejä tekoälyyn liittyen. Kielimallit voivat ilman guardraileja generoida ennakoasenteellista, haitallista tai jopa laitonta sisältöä. Malli voidaan näillä menetelmillä ohjata turvallisesti organisaation omien määritelmien ja vaatimusten perusteella.

Älykäs haku

Älykäs haku on hakuprosessi, joka RAG -sovelluksessa antaa relevanttia tietoa kielimallille, joka muodostaa vastauksen annetun tiedon perusteella. Haku perustuu usein yhdistelmään semanttisen samankaltaisuuden algoritmeihin, avainsanoihin, sekä metadata -filttereihin.

Prompt engineering

Tavallisesti tekoälymalleja jatkokoulutetaan jopa kymmenillä- tai jopa sadoillatuhansilla esimerkeillä, jotta malli saadaan erikoistumaan haluttuun tehtävään. Prompt engineering on taas erityinen menetelmä, jossa suuri kielimalli erikoistetaan tehtävään muokkaamalla syötettä erilaisilla tekniikoilla. Näitä tekniikoita ovat esimerkiksi few-shot prompting, chain of thought, jne.

Few-shot prompting

Few-shot promptaus on yleinen prompt engineering -tekniikka, jossa syötteessä sisällytetään esimerkkejä, joita malli pyrkii toistamaan ja täydentämään. Esimerkiksi, voidaan antaa syötteessä seuraavat neljä esimerkkiä, joista malli täydentää viimeisen esimerkin:

Syöte

Ruoka oli kylmää ja oli liian suolaista! // Negative

Hieno! // Positive

Elokuva oli upea! // Positive

Olipas kamala asiakaspalvelu.. //

Generointi

Negative

Epäyhtenäisen tiedon puhdistaminen (esim. Alaviitteet keskellä tekstiä) ja asiaankuuluvan metatiedon lisääminen (esim. otsikot) tietokantaan auttaa LLM-malleja ymmärtämään eri tietotyyppisiä tehokkaammin.

Tietolähteiden jakaminen:

Vaikka voi olla houkuttelevaa sekoittaa kaikki tietotyypit keskenään ja käyttää semanttista samankaltaisuushakua, eri tietotyypeillä voi olla erilaisia vaatimuksia, jolloin niiden erillinen käsittely johtaa parempiin tuloksiin.

Domain tuntemus

Eriyistiedon sisällyttäminen esimerkiksi sanastojen, kehoitteiden suunnittelun tai hienosäädön avulla on välttämätöntä, jotta LLM:t ymmärtävät toimiala- tai yrityskohtaista tietoa.

Yhteenvedona, on tärkeää ottaa huomioon nämä tekijät, jotta voidaan parantaa LLM-sovellusten tehokkuutta ja tarkkuutta, sekä varmistaa, että ne vastaavat paremmin käyttäjien tarpeita ja vaatimuksia.

