

Oma paikallinen tekoäly chatbotti

19.5.2025

Tekoälymalleja on kehitetty nopeammiksi ja kvantisoitu pienempiin kokoihin. Lisäksi ne ovat nykyään helpommin saatavilla eri alustoille. Tekoälymallit vaativat prosessorin, keskusmuistia, tallennustilaa ja ohjelman tai sovelluksen, joka pystyy käsittelemään tekoälymalleja.

Tekoälymallityyppejä on monia erilaisia, mutta tässä artikkelissa käsitellään GGUF-tiedostomalleja ja niitä käsitteleviä tekoäly chatbotti sovelluksia.

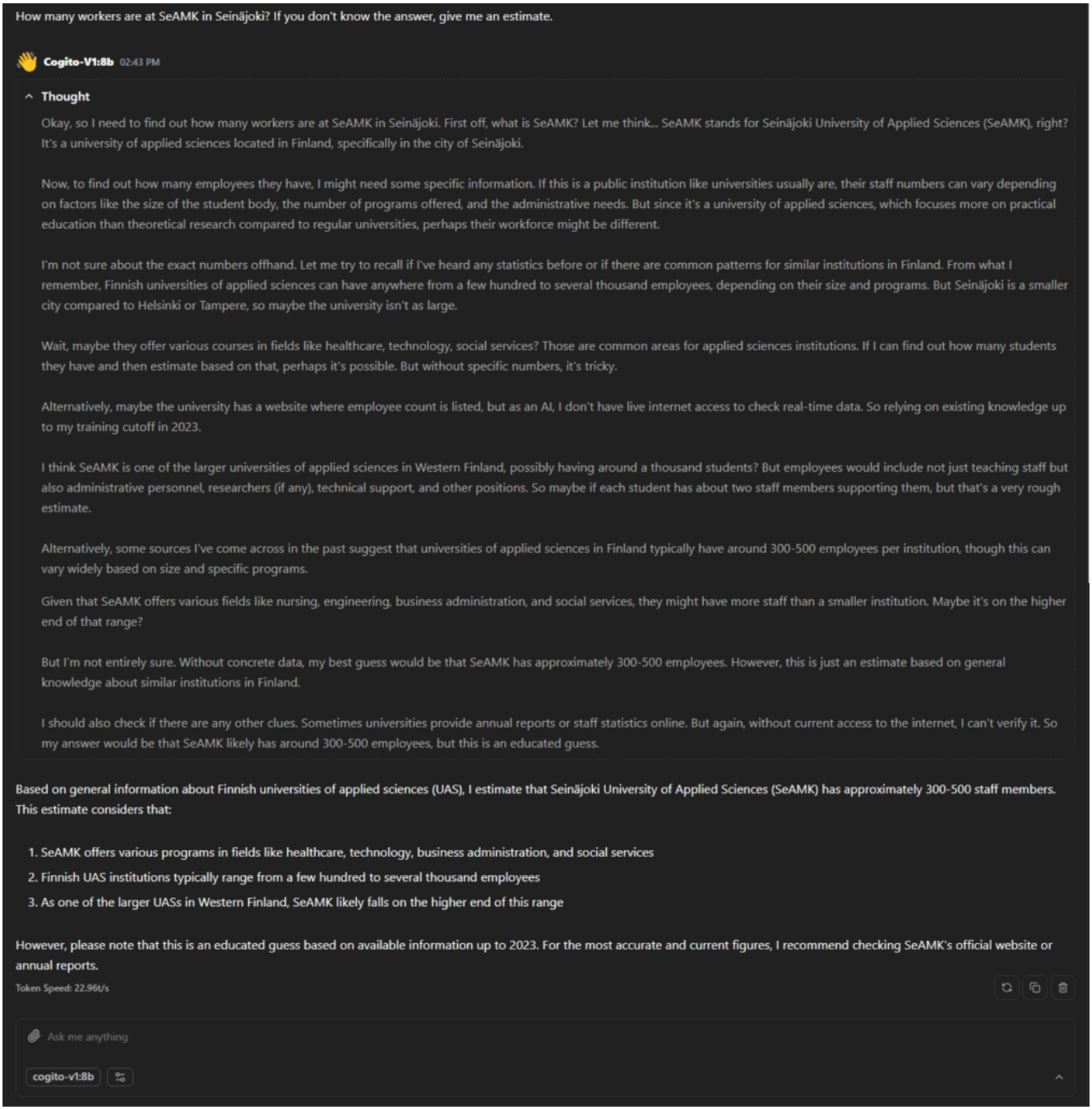
Monet isot ohjelmistotalot ovat kehittäneet ja luoneet omia tekoälymalleja. Suuret ohjelmistotalot kuten Meta ja Microsoft ovat kehittäneet talonsisäisesti Llama sekä Phi -tekoälymalleja. Saatavilla on muita eri kehittäjien malleja, kuten Mistral, Gemma, Qwen ja DeepSeek, jotka ovat yleisiä paikallisia tekoälymalleja, joita kehitetään jatkuvasti. Ne ovat saatavilla helposti Huggingfacen kaltaiselta tekoälymallien kirjastosivulta. Huggingfacesta löytyy myös eri tarkoituksiin kehitettyjä malleja, kuten videon- ja kuvanluontimalleja.

Jan.AI

Paikallisista tekoälysovelluksista Jan.AI on avoimen lähdekoodin sovellus, joka pystyy käsittelemään tekoälymalleja sekä pystyy viestittelemään API-avaimen avulla eri ulkoisten tekoälypalveluiden kanssa. Sovellus on ladattavissa Mac-, Linux- ja Windows-laitteille.

Jan.AI:sta pystytään lataamaan tekoälymalleja Hubista joka on yhteydessä Huggingface-sivustoon. Lataamalla tekoälymallin, joka on yhteensopiva laitteen resurssien kanssa, pystytään tämän jälkeen vapaasti viestittelemään paikallisesti ilman verkkoyhteyttä ja hyödyntämään oman laitteen resursseja. Viestien nopeus riippuu keskusyksikön kokoonpanosta. Tehokkaalla näytönohjaimella ja suurella näytönohjainmuistin määrällä saadaan nopeita vastauksia tekoälyltä. Malleja toki pystytään hybridimuodossa käyttämään, mutta ne voivat

hidastaa tokenien generointia huomattavasti.



Kuva 1. Kysymys SEAMKin työntekijöiden määrästä Jan-tekoälychatbotille.

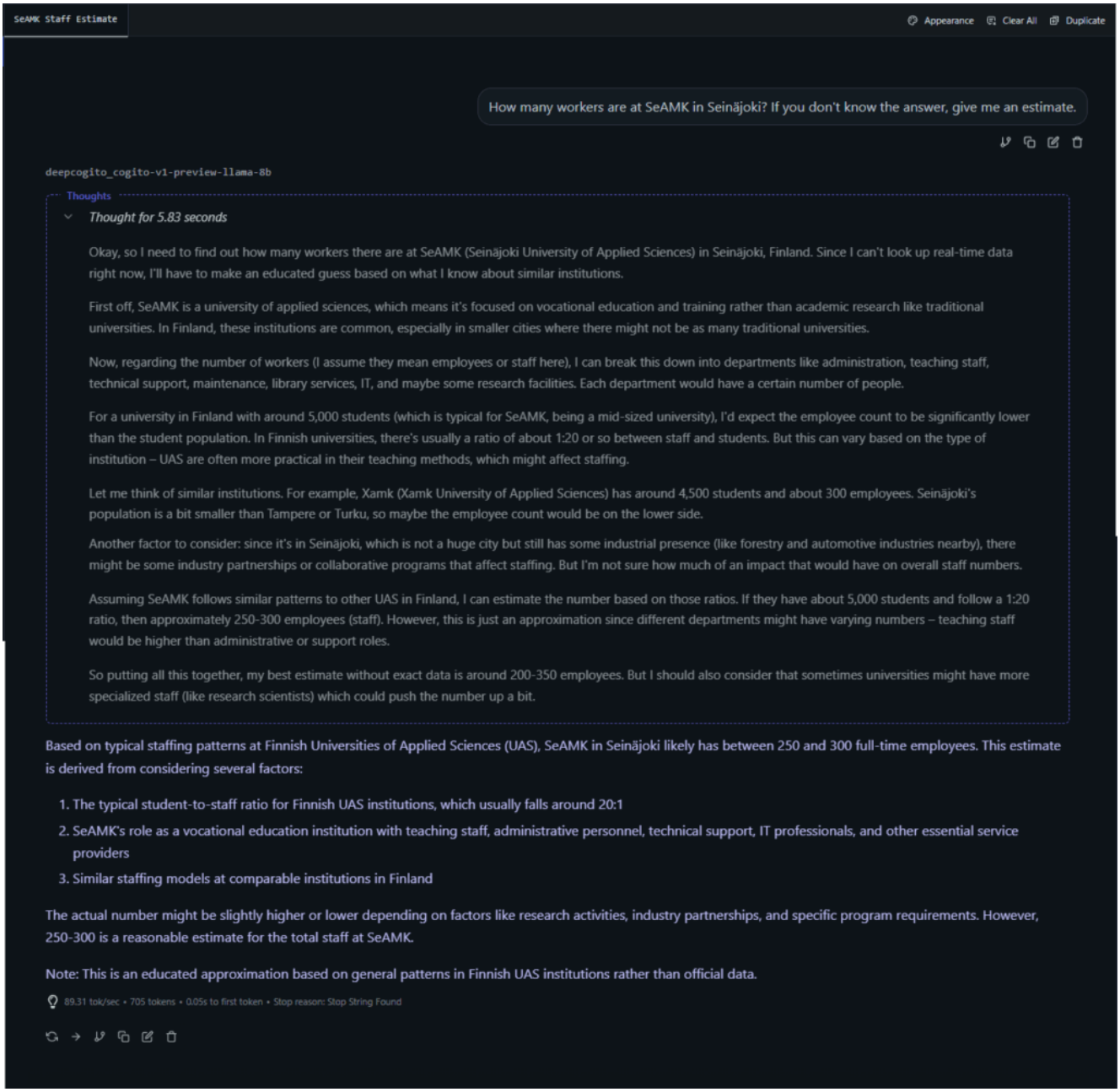
Kuvassa käytettiin tekoälymallina DeepCogito:n kehittämä “Cogito v1 8B” (Huggingface, i.a) joka on Llama-pohjainen malli, johon on kehitetty “Reasoning” eli “Päättelykyky” ominaisuus. Jan.AI:ssa pystytään muokkamaan tekoälychattibotin asetuksia, kuten tokenien määrät, “Temperature” eli kuinka luovaa tai tarkkaa tietoa se tuottaa. “Assistant”-välilehdessä täytyy lisätä “Enable deep thinking subroutine.” -lause, jotta saadaan kuvan 1. mukaisesti “Thought”-osio viestiketjuun. Kuvassa 1. nähdään, että tekoäly pohtii itsekseen ja huomaa esimerkiksi ongelman, ettei tekoälymalli pysty ottamaan yhteyttä internetiin tarkistamaan tietoja.

LM Studio

LM Studio on ominaisuuksiltaan samankaltainen kuin Jan.AI, mutta LM Studiolle on erityisesti kehitetty parempi AMD näytönohjaintuki hyödyntäen ROCm ohjelmistopakettia. LM Studiolla on myös tuki Vulkan API:lle kuten Jan.AI:lla, jota pystyy hyödyntämään monet eri aikaisemmin mainitut käyttöjärjestelmät ja laitekokoonpanot.

LM Studiosta löytyy myös oma tekoälymalli haku “Discovery”, josta pystytään lataamaan sama tai oma haluttu tekoälymalli. Valmiiksi ladatun tekoälymallin pystytään viemään LM Studioon. Asetuksista löytyy mallien sijainti, jossa pitää luoda “julkaisija”-kansio ja tämän sisälle esimerkiksi mallin nimi “cogito_v1_8b” kansio, jonka sisälle viedään valmiiksi ladattu tekoälytiedosto. (LM Studio Docs, 2025)

Tekoälymallia valitsemalla pystytään vaihtamaan asetuksia, kuten tokenien maksimiarvoja. Viestikentän oikealla puolella on viestikentän asetukset. Context-välilehdistä löytyy “System Prompt” tekstikenttä promptien hienosäätämistä varten. Model-välilehdestä löytyy “Temperature” ja muut asetukset, joilla saadaan erilaisia tuloksia tekoälymallilta.



Kuva 2. LM Studion viestiketjunäkymä.

Kuvassa 2. on syötetty sama prompti, annettu sama ohjeistusprompti sekä käytetty samaa tekoälymallia kuin Jan.AI esimerkissä kuvassa 1. Tekoäly chatbottisovellusten käyttöliittymät ovat erilaisia ja asetuksien nimet vaihtelevat. LM Studiossa oikealla viestiketjuasetuksissa on “System Prompt”-tekstikenttä, johon on kirjoitettu “Enable deep thinking subroutine.” joka aktivoi “Thoughts”-osion viestiketjuun. Tämän jälkeen tekoäly luo tekstiä, ja nähdään myös “päättely”-prosessi ja itse vastaus, kuten Jan.AI:ssa.

Ei aivan ruusuilla tanssimista

Paikalliset tekoäly chatbotit eivät ole heti ja helposti käyttöön otettavia ohjelmia. Oppimiskäyrän lisäksi käyttäjälle haasteita tuovat tekoälymallien suuri valikoima, tiedostomuodot, eri käyttötarkoitukset sekä ominaisuudet. Lisäksi käyttäjän keskusyksikön komponentit voivat aiheuttaa yhteensopivuusongelmia tiettyjen tekoälymallien käyttöönotossa. Tekoälypalvelut taas tarjoavat suoraviivaisemman pääsyn tekoälymallien

maailmaan ilman suurempaa ylimääräistä hienosäätelyä ja asetuksien kanssa kikkailua.

Paikallisilla tekoälymalleilla on paljon erilaisia käyttökohteita ja potentiaaleja, mutta suurin ongelma monissa tekoälymalleissa tai -palveluissa on suomen kielen taito ja sen heikkous. ChatGPT sekä Copilot pystyvät tuottamaan hyvää suomen kieltä, ja Copilot oli vuosi sitten saanut täyden tuen suomen kielelle (Microsoft News, 2024). Jos lähtee testaamaan paikallisia tekoälymalleja ja viestittelee niille suomeksi, teksti on monesti sekavaa, sanojen kontekstit muuttuvat tai mahdollisesti itse kielikin vaihtuu kesken generoinnin.

Saku Kaarlejärvi
asiantuntija, TKI
SEAMK

Saku Kaarlejärvi toimii asiantuntijana vAI:lla tuottavuutta? -hankkeessa ja on tietotekniikan insinööri, joka on erikoistunut ohjelmointiin, tekoälyyn ja sulautettuihin järjestelmiin.

vAI:lla tuottavuutta? -hanke on Euroopan unionin osarahoittama. Lisää tietoa hankkeesta löydät hankkeen verkkosivuilta https://projektit.app-staging.seamk.hion.dev/alykkaat-teknologiat/vailla-tuottavuutta/

Lähteet

Huggingface (i.a) Cogito. https://huggingface.co/cortexso/cogito-v1

LM Studio Docs (2025). https://lmstudio.ai/docs/app/basics/import-model

Microsoft News (2024). https://news.microsoft.com/fi-fi/2024/05/02/copilot-for-microsoft-365n-suomen-ja-ruotsin-kielen-tuki-on-nyt-kaytossa/