

語彙の誕生・語彙の誕生・語彙の誕生・語彙の
誕生・語彙の誕生・語彙の誕生・語彙の誕生・
語彙の誕生・語彙の誕生・語彙の誕生・語彙の
誕生・語彙の誕生・語彙の誕生・語彙の誕生・
語彙の誕生・語彙の誕生・語彙の誕生・語彙の
誕生・語彙の誕生・語彙の誕生・語彙の誕生・
語彙の誕生・語彙の誕生・語彙の誕生・語彙の
誕生・語彙の誕生・語彙の誕生・語彙の誕生・
語彙の誕生・語彙の誕生・語彙の誕生・語彙の
誕生・語彙の誕生・語彙の誕生・語彙の誕生・
語彙の誕生・語彙の誕生・語彙の誕生・語彙の
誕生・語彙の誕生・語彙の誕生・語彙の誕生・

『Sanaston synty』

Petäjä-opisto / Olli Ruohomäki
CC BY 4.0 / Päivitetty 2026-08-18

Sisällysluettelo

Alkusanat	3
1. Taustaa	3
Idea	4
2. Opi ja opeta	4
Miten sanoja opitaan	4
Itsetehty sanasto	5
Sanaston opettamisesta	5
Miltä sanasto näyttää	6
3. Luo oma kieli	6
4. Kehitysvaiheita	7
Versiohistoriaa	7
Ongelmia ja ratkaisuja	8
Parhaat opit	9
5. Sanaston luominen	10
Näin luot sanaston	10
Ulkoasun viimeistely	10
Muista metadata	10
Vinkit tiedonhallintaan	11
6. Sanaston tarkastaminen	11
Millainen on hyvä sanasto	11
Miten sanasto tarkastetaan	11
Miten tarkistettu sanasto merkitään	12
Miten virheitä korjataan	12
7. Fraasit ja kokoelmat	12
PET-fraasityökalu	13
Fraasisanaston luominen	13
GCC-kokoelmatyökalu	13
Sanastokokoelman luominen	14
Vinkkejä tiedonhallintaan	14
8. Arvot ja periaatteet	14
Arvot kaiken takana	14
Suunnitteluperiaatteita	15
Parhaat opit	15
9. Kokeile itse	16
Sanaston luominen	16
Sanaston tarkastaminen	16
Fraasisanaston luominen	17
Tekijänoikeuksista	17
Demolyriikasta	17
Tulevat julkaisut	18
Lopuksi	18
Liitteet	18

Alkusanat

Tämä julkaisu kokoa yhteen vuonna 2026 toteuttamani hankkeen, jossa kehitin menetelmää tuottaa kieltenopiskelussa tarvittavia sanastoja hyödyntämällä tekoälyä. Tämän julkaisun tavoitteena on dokumentoida tehtyä työtä ja muuntaa aineistot saavutettavampaan, siirrettävämpään ja pysyvämpään formaattiin.

Tämä tarina julkaistiin alun perin yhdeksänsaisena blogijuttusarjana heinäkuussa 2026. Blogi löytyy osoitteesta: <https://blogit.sasky.fi/ruohomaki>. Pdf-versio ja liitteet löytyvät myös OPH:n Avoimien oppimateriaalien kirjastosta, eli <https://aoe.fi> -palvelusta.

TLDR; Idea sanaston luomisesta itse valittavasta lähdetekstistä syntyi tarpeesta tehdä itselleni käyttökel-poista oppimateriaalia. Satunnaiset kokeilut muuttuivat pian määrityksen kehittämiseksi ja yksittäisten aineistojen tuottaminen tuotantomenetelmän rakentamiseksi. Idea alkoi eskaloitua mukavasti, kun tajusin, että valmis tuotantoprosessi olisi helppo siirtää, skaalata ja muokata eri käyttäjien tarpeisiin. Se edellyttäisi käytännössä täyttä avoimuutta, hyvää dokumentointia, siistiä koodia ja esimerkkejä pedagogisista malleista. Nämä kaikki ovat kuitenkin täysin tehtävissä.

Kankaanpää,

18.8.2026

Olli Ruohomäki

1. Taustaa

Kaikki alkoi siitä, että tarvitsin oppimateriaalia japanin kielen opiskeluani varten. Halusin ymmärtää, mitä suosikkilauluissani sanotaan ja mitä suosikkiartistini haastatteluissaan puhuvat.

Itseopiskelu ei sujunut hyvin, kun halusin opetella juuri tietynlaista sanastoa. En turistifraaseja enkä työelämän kohteliasta puhetapaa, vaan sitä epämuodollista puhekieltä, mitä kuulee *J-Rockissa* ja *J-Popissa*.

Sanaston opiskelu oli kuitenkin vaikeaa, kun en vielä osannut perinteisiä japanilaisia merkistöjä (hiraganaa, katakanaa ja kanjia). Vaikka näkisin merkit edessäni ja kuulisin, miten sana lausutaan, en osaisi yhdistää niitä toisiinsa.

Pyysin aluksi tekoälyä kääntämään kokonaisen laulutekstin ja pääsin sitä kautta kiinni laulun sanomaan. Sanavarastoni ei kuitenkaan siitä karttunut, sillä en osannut yhdistää alkuperäiskielisiä sanoja käännöksiin. Sanoja taivutetaan ihan yhtä paljon kuin suomessa ja myös sanajärjestys on erilainen. Lisäksi japanissa kaikkea ei sanota ääneen, vaan lukijan pitää aina tulkita osa.

Aloin tämän jälkeen kysellä yksittäisiä sanoja tekoälyltä: miten sana kirjoitetaan länsimaisilla aakkosilla (romaji), mitä se tarkoittaa ja mikä tämä taivutusmuoto on. Opin pian, että sanoilla on useita merkityksiä.

Esimerkiksi sana **akane** (茜) (*subst.*) voi tarkoittaa:

1. (*kirjaimellisesti*) punamatarakaasvilajia (*Rubia akane*)
2. (*tavallisesti*) syvän tai hehkuvanpunaista väriä
3. (*abstraktisti*) iltaruskoa tai nostalgiaa
4. (*historiallisesti*) vaurautta
5. (*filosofisesti*) katoavaisuutta tai itsereflektion hetkeä

Kirjaimellinen käännos ei siis riitä ymmärtämään, mitä sana tarkoittaa. Pitää katsoa sanan merkityskerrok-sia ja valita niistä tilanteeseen parhaiten sopiva.

Sanojen kyseleminen tekoälyltä oli hyvin mielenkiintoista, mutta myös hyvin hidasta. Oli vaivalloista kirjoittaa pitkä kehote yhden sanan selvittämiseksi. Lisäksi yksittäiset vastaukset hukkuivat nopeasti keskusteluketjujen syövereihin. Ne täytyisi noukkia sieltä johonkin talteen.

Idea

Mitpä, jos valjastaisi tekoälyn tuottamaan sanaston kokonaisuudesta laulusta yhdellä kertaa? Tuollainen 60–80 sanaa voisi olla sopivan kokoinen annos opiskella yhdellä kertaa. Jos sanastot vielä tallentaisi tiedostoiksi, ne löytäisi helposti myös jatkossa.

Yksittäisen sanan kysyminen on normaalia tiedonhakua, mutta kokonaisen sanaston tuottamiseen tarvitaan jo jonkinlainen systeemi. Siihen pitäisi vain suunnitella sujuva työnkulku ja tiedonhallinta. Niitä olisi hyvin mielenkiintoista alkaa tutkia ja kehittää.

Teen parhaani! – Gambarimasu! (頑張ります)

2. Opi ja opeta

Tämä luku on kirjoitettu oppijan ja opettajan näkökulmista. Aluksi pohditaan, miten sanoja opitaan koulun ulkopuolella, miten sanastot tukevat oppimista ja millaisia sanastoja tässä on rakennettu.

Miten sanoja opitaan

Jos puhutaan muualla kuin koulussa tai kurssilla tapahtuvasta kielten opiskelusta, sanoja ei yleensä opita kerralla. Niitä ensin huomaa, sitten arvailee ja vähitellen ne alkavat tarttua ja avautua. Seuraavassa omia havaintojani täysin vieraan uuden kielen kohtaamisesta.

- Kun kuulet sanan ensimmäistä kertaa, ymmärrät, ettet tiedä mitä se tarkoittaa, etkä ole varma, miten se kirjoitetaan. Lisäät sanan mielesi selviteltävien tapausten listalle.
- Vertaat uutta sanaa jo tuntemiisi, samalta kuulostaviin sanoihin myös muihin osaamiisi kieliin.
- Joskus uusi sana alkaa häiritä niin paljon, että sen selitys täytyy katsoa netistä. Opit sen heti, epä-tietoisuus ei enää häiritse ja uteliaisuutesi tyydyttyi.
- Kun kohtaat sanan fraasissa, voit ymmärtää sen käytännön merkityksen, vaikket vielä yksittäisen sanan kirjaimellista merkitystä. Muistat vain, että *"tuo sana oli siinä fraasissa"*.
- Alat arvailla ja päätellä monenlaisia asioita puhujan ympäriltä: Kuka puhuja on? Millaisista asioista tuossa asemassa / roolissa yleensä puhutaan?
- Jos ymmärrät kielen sanoja ja rakenteita, voit alkaa haarukoida uuden sanan merkitystä "verbi, jota voi käyttää meri-sanalla kanssa"
- Kun et vielä osaa uutta sanaa, olet virittynyt huomaamaan sen erilaisissa lauseyhteyksissä, esim. tv-ohjelmien tekstityksissä.
- Olet oppinut sanan, kun tunnistat sen kuultuna ja kirjoitettuna, osaat sanoa ja kirjoittaa sen, tiedät sen merkityksen ja missä yhteydessä sitä voi käyttää.
- Kun alat käyttää oppimaasi sanaa, olet saanut sen aktiiviseen sanavarastoosi.
- Huomaat, että sanaa käytetään muissakin merkityksissä. Alat oppia erilaisia merkityksiä. Kun osaat tulkita sanan merkityksen sen käyttöyhteyden mukaan, olet lopulta ymmärtänyt sanan.

Edellinen oli lineaarisesti eteneväksi siistitty lista. Oikeasti sanoja kuitenkin unohtaa, kuulee, ymmärtää, sanoo ja kirjoittaa väärin. Usein sanan merkityksen sekoittaa samalta kuulostavaan sanaan. Joskus näitä virheitä korjataan pitkänkin ajan päästä.

Itsetehty sanasto

Itse tuotettuja sanastoja voi hyödyntää monessa eri oppimiseen liittyvässä vaiheessa ja tehtävässä.

- Jos sinulla on lähdeteksti ja sen rinnalla siitä generoitu sanasto, voit opetella kääntämään ja tulkitsemaan tekstiä ilman muuta oppimateriaalia.
- Jos sanasto on luotu lähdeaineistosta, jonka olet kuullut, mutta et nähnyt, löydät sanastosta sanoille oikeat kirjoitusasut.
- Tekstin kääntäminen on helpompaa, kun sanasto on järjestetty siten, että sanat esitetään siinä järjestyksessä, kun ne lähdetekstissä on esitetty.
- Elävässä puhekielessä sanat sulautuvat paljon toisiinsa. Sanasto auttaa näkemään, missä yksi sana loppuu ja toinen alkaa. Kuvittele vaikka olevasi ulkomaalainen, joka yrittää ilman sanastoa tulkita suomen kielen repliikkiä *”emmäänyoikeintostatiia”*.
- Se, että sanastoon on merkitty myös sanaluokat, helpottaa kielen rakenteiden hahmottamista. Esi-merkki omasta äidinkielestä poikkeava sanajärjestys alkaa hahmottua, kun ymmärtää, esimerkiksi että *”tämä on substantiivi ja tuo on adverbi”*.
- Sanan eri merkitysten tarjoaminen syventää sen ymmärtämistä. Se myös totuttaa ajatukseen, että lukeminen on samalla tulkintaa, asiayhteyteen sopivan merkityksen valitsemista.

Sanaston opettamisesta

Tämän hankkeen perusidea on tehdä työkalu, joka tuottaa sanaston vapaasti valittavasta lähdetekstistä. Se sopii yhtä hyvin itseopiskelijan kuin opettajanakin käyttöön. Opettajat selittävät sanoja ja kokoavat sanastoja jatkuvasti. Tähän työhön valjastetaan nyt tekoäly avuksi.

Itse valitusta lähdetekstistä luotu sanasto tukee tosielämässä käytettävän, ajankohtaisen ja oppijalle merkityksellisen kielen oppimista. Oppijan motivaatio on korkeampi, kun sisältö on oikeasti oppijaa kiinnostavaa ja merkityksellistä. Fani haluaa oppia suosikkiartistinsa laulunsanoja, Japaninmatkaa suunnittelevan on tärkeä osata lentokentän ja metron kylttejä ja aasialaiseen keittiöön hurahtaneen on hyvä osata lukea pakkausmerkintöjä, esim. soijakastiketyypit pullon kyljestä.

Normaalisti sanakirjoissa ja sanastoissa selitettävä sana on perusmuodossa. Tekstissä sanat ovat joskus perusmuodossa, mutta useammin eivät. Suomen ja Japanin kielet ovat siinä mielessä samanlaisia, että sanan perusmuodosta otetaan loppuosa pois ja se korvataan taivutuspäätteillä. Ne ovat aloitteleville opiskelijoille vaikeita, koska:

- taivutuspäätteitä on useita erilaisia
- sanojen taivuttaminen ei ole täysin kiinteää (osa on epäsäännöllistä)
- sanan juuren perään voi olla liimattu useitakin taivutuspäätteitä

Tällainen räätälöity sanasto voitaisiin tuottaa myös niin, että selitettävä sana on lähdetekstissä esiintyvässä muodossa. Näin voitaisiin tuottaa tekstin nopeaa ymmärtämistä tukevia sanastoja generoimalla esimerkiksi suomenkielisestä uutissähkeestä ukrainankielinen sanasto. Tällöin sanan perusmuodon voisi esittää selitysosana lopussa tukemaan sanojen oppimista.

Sanaston opiskelu on entistäkin vaikeampaa, jos vieraan kielen teksti on kirjoitettu merkistöllä, jota oppija ei vielä tunne. Japanin kielen alkeisoppikirjat on ulkomaalaisia varten yleensä kirjoitettu länsimaisin aakkosin (romaji). Lisäetuna on, että ne myös kertovat, miten sanat äännetään. Suomalaisille tämä on valtava etu, sillä sekä suomea että japania lausutaan pääosin niin kuin kirjoitetaan. Eli: suomalainen voi lukea romajilla kirjoitettua japania aivan suoraan.

Sanastoon on tuotu sanat myös perinteisillä kirjoitusmerkeillä (kanji, hiragana ja katakana), jotta sanasto auttaa oppimaan ja käyttämään myös perinteisiä japanilaisia merkistöjä.

Opettajan näkökulmasta itse tehty sanasto avaa uudenlaisia mahdollisuuksia. Oppijalle voidaan luoda opiskelupaketti, jossa on 1. linkki oppijan lempibiisiin, 2. biisin sanasto ja 3. pieni tehtävänanto, esim.

- Opettele sanasto romaji-aakkosilla
- Harjoittele ääntämistä laulajan perässä
- Käännä kaksi ensimmäistä säkeistöä suomeksi
- Tutki, mitä taivutuspäätteitä laulunsanoissa esiintyy
- kuuntele kappale seuraten sanastoa samalla paperilta
- Piirrä paperille kappaleen kymmenen ensimmäistä kanji-merkkiä.

Miltä sanasto näyttää

Sanastojen rakenne suunniteltiin aloittelijan tasolla olevan japanin itseopiskelijan tarpeita varten. Yksittäinen sanastoentry näyttää tältä:

ikiru 生きる (*verb*) 1. To live; to be alive 2. To continue existing despite difficulty 3. To live with purpose or existential awareness (lyric form: *ikil*ta – past; completed action, may imply memory or finality)

- Selitettävä sana on perusmuodossa länsimaisilla merkeillä (romaji)
- Selitettävä sana perinteisillä japanilaisilla merkeillä (kanji ja hiragana) tekemään niitä tutuiksi alusta pitäen ja helpottamaan siirtymistä käyttämään niitä opiskelun edetessä
- Sanaluokka helpottaa lauseen jäsentämistä ja japanin sanajärjestyksen ymmärtämistä.
- Merkitystasot ohjaavat tulkitsemaan sanojen merkityksiä opiskelun alusta pitäen. Tasoja on useita: kirjaimellinen, käytännöllinen, historiallinen, abstrakti, teoreettinen, poeettinen, vertauskuvallinen, filosofinen. Tähän sanastoon otettiin kolme (kirjaimellinen, abstrakti ja runollinen).
- Lyric form: sana siinä muodossa, miten se esiintyy ja sen käännös.

Esimerkkejä valmiista sanastoista:

- Atozome: [Na mo naki tamashii](#) (名もなき魂) (pdf 4 s. CC BY 4.0, FI)
- HACHI: [Hoshi wo matsu](#) (星を待つ) (pdf, 4 s. CC BY 4.0, EN)
- Hakubi: [Twilight](#) (pdf, 3 s. CC BY 4.0, EN)

3. Luo oma kieli

Sanastoihin liittyvä hanke ei olisi mitään ilman omaa sanastoa. Tässä on otettu käyttöön koodikieli, joka koostuu englanninkielisistä lyhenteistä. Ne eivät ole minkään olemassa olevan standardin mukaisia, vaan luovat tälle kohdealueelle omaa kieltä (*Domain-Specific Language*). Näiden lyhenteiden tehtävät ovat:

- nimetä tekoälylle annettavat määrittelyt
- nimetä tekstisisältöjen käsittelyvaiheet
- toimia tekstisisältöjen tunnistena tiedostonimissä
- viitata asioihin tiiviisti ja nopeasti blogissa ja käyttöohjeissa

SD (*Source Document*) tarkoittaa tiedostoksi tallennettua lähdetekstiä. Se liitetään kehoitteeseen ja sanasto generoidaan siitä. Tekoäly ei siis etsi lähdetekstiä verkosta, vaan se pitää sille antaa. Tiedonhallinnan kannalta lähdetekstiin on tärkeää lisätä metadataa lähdetekstistä.

GGs (*Glossary Generation Specification*) on määrittely, joka kertoo tekoälylle, miten sanasto generoidaan: mitä sanoja lähdetekstistä (SD) poimitaan, mitä selityksissä on ja missä muodossa ne esitetään. Määrittely poistaa käyttäjältä sanaston luomiseen tarvittavan kehoitteen laatimisen vaivan ja pienentää tarvittavan tekoälyosaamisen määrää. Määrittely myös yhtenäistää sanastojen sisältöä, mikä on tärkeää, kun sanastoja kootaan suuremmaksi kokoluokaksi.

GD (*Glossary Document*) on valmis sanasto, joka on tallennettu tiedostoksi. Sanat ovat lähdetekstin esiintymisjärjestyksessä helpottamaan lähdetekstin seuraamista ja opiskelua.

GDV (*Glossary Document Verifier*) on valmiin sanaston laadunvarmistustyökalu. Se on tekoälylle annettava määrittäminen, jolla tarkistetaan sanaston termien ja selitysten oikeellisuus. GDV suunniteltiin aluksi opettajille, mutta on hyödyllinen työkalu kaikille sanastoja tekeville.

GVR (*Glossary Verification Report*) on GDV:n tuottama raportti, josta näkee luotettavien, tarkistettavien ja väärin sanastotietojen määrät, listaa tarkistettavat sanat ja selitykset ja ehdottaa niihin korjauksia.

VGD (*Verified Glossary Document*) on valmis sanasto, joka on tarkastettu GDV:llä, muotoiltu tekstieditorilla ja tallennettu tiedostoksi.

PET (*Phrase Extraction Tool*) on työkalu, joka tutkii SD:n ja etsii sieltä fraasit, kääntää ne 1. kirjaimellisen ja 2. kulttuurisen ja sosiaalisen merkityksen. PET tuottaa fraasilistan selityksin varustettuna.

PGD (*Phrase Glossary Document*) on PET työkalun tuotos, joka on tallennettu tiedostoksi. Se sisältää useammasta merkistä muodostuvan kokonaisuudet, joiden merkitykset eroavat niiden kirjaimellisista merkityksistä.

GCC (*Glossary Collection Compiler*) on tekoälylle annettava määrittäminen, jolla erillisistä sanastoista (GD) koostetaan laajempi kokonaisuus. Esimerkiksi kokonaisen albumin laulun sanat niin, että koostesanastossa on kukin sana vain yhteen kertaan ja aakkostettuna.

GCD (*Glossary Collection Document*) tarkoittaa erillisistä sanastoista koostettua sanastoa, missä sanat ovat sanakirjamaisesti aakkosjärjestyksessä. Jos yksittäiset sanastot (VGD) tehdään musiikkikappaleista, sanastokokoelma voisi olla kokonaisen albumin kappaleiden sanastojen yhdistelmä.

GDC (*Glossary Document Collection*) on sanastojen kokoelma, eli joukko erillisiä sanastotiedostoja. Termi ei ota kantaa siihen, onko niiden oikeellisuus varmistettu tai missä ne sijaitsevat.

GDA (*Glossary Document Archive*) on yleistermi, joka viittaa sanastotiedostojen tallennuspaikkaan. Se voi olla yksittäinen tietokone, oppilaitoksen palvelin, verkkoasema tai siirrettävä tallennusväline. Ehkä jopa mappi, missä sanastot ovat paperitulosteina. Termi on käytössä siksi, että sillä voidaan tarkoittaa mitä tahansa tallennuspaikkaa niitä tarkemmin nimeämättä. Jokaisella sanastojen käyttäjällä ja esimerkiksi koululla voi olla oma niille varattu tallennuspaikka.

GGM (*Glossary Generation Manual*) on ihmiskäyttäjälle tarkoitettu dokumentti, joka neuvoo, miten sanastoja generoidaan, mitä ongelmia on havaittu, miten niitä voi välttää, miten sanaston ulkoasu on hyvä muotoilla, miten sanastotiedosto tehdään ja miten sen laatu varmistetaan. Lisäksi siinä kerrotaan lyhyesti GGS:n muokkaamisesta.

4. Kehitysvaiheita

Tämä luku kertoo sanastogeneraattorin rakentamisen vaiheista, ongelmista ja ratkaisuista. Siitä, miten hanke muuttui oppimateriaalin tuottamisesta ohjelmoinniksi luonnollisella kielellä. Tämä on suunnattu erityisesti tekoälyn aktiivisille käyttäjille. Kielenopettajaa saattaa kiinnostaa lopussa liitteenä oleva sanojen luokitus, mutta kielenopiskelijan kannattaa ihan oikeasti jättää tämä luku väliin.

Versiohistoriaa

Spesifikaatio kasvoi aluksi nopeasti, mutta myöhemmin sitä alettiin myös yksinkertaistaa. Tarkoituksena oli erottaa toisistaan dokumentointiin, tiedostonhallintaan ja varsinaiseen sanaston muodostamiseen liittyvät tehtävät. Uusimmissa versioissa painopiste siirtyi pedagogisiin ja kielellisiin ratkaisuihin. Seuraava taulukko antaa yleiskuvan versiokehityksestä:

Versio	Pääosioita / alakohtia	Tärkeimmät muutokset	Toimivuus
0.1	1 / 6	Sanastoentryn rakenne, kolmen merkitystason selitys, selitettävä sana perusmuodossa	testi
0.2	4 / 15	ulkoasun muotoiluja, lyric form ja taivutukset	testi
0.3	4 / 18	Tuotti kokonaisen sanaston. Dokumentin muoto ja rakenne vakiintuvat, mallina painetut sanakirjat.	toimiva ja hidas
0.5	7 / 30	Mukaan How to Use, tulostus docx-tiedostoon, tiedoston nimeäminen ja metadata.	epävakaa
0.6	8 / 34	Generation Assurance Rules lisätään. Pitkien ajojen ja keskeytysten hallinta.	epävakaa
0.7	9 / 38	Ihmishojeet poistuvat. Sisältösäännöt tarkentuvat, Assurance säilyy.	epävakaa
0.8	9 / 39	Lyric Glossary Spec nimettiin uudelleen Glossary Generation Speciksi, sillä ei ole käytännön syytä rajoittaa käyttöä pelkkiin laululyriikoihin.	vakaa ja hidas
0.81	5 / 39	Säännöt ryhmitellään (Formatting, Content, Heading, Metadata...). Dokumenttirakenne selkeytyy.	vakaa ja hidas
0.82	13 / 32	Suuri pelkistys. Completeness Check, automaattinen regenerointi ja AI-keskeinen prosessi.	vakaa ja hidas
0.83	14 / 34	Output Header ja Excluded-lista. Konjugaatiosääntö tarkentuu.	vakaa ja hidas
0.84	14 / 34	Semanttiset kerrokset ja prosessi hioutuvat, rakenne vakiintuu.	vakaa ja hidas
0.85	14 / 35	Romanisointi ja kielikohtaiset säännöt tarkentuvat.	vakaa ja hidas
0.86	14 / 36	Lyric Form muuttuu ehdolliseksi. Romanisointi (macronit) vakiintuu.	vakaa ja hidas
0.87	15 / 38	Quality Principle, GD-ajattelu, Ignore Metadata ja tuotannon tilastointi.	vakaa
0.90	14 / 31	Partikkelit ja funktiosanat poistuvat, fraasit erotellaan.	vakaa
0.91	15 / 33	Word Boundary Rule.	vakaa
0.92	15 / 35	Sanaluokittelua tarkennetaan (mm. laskusanat, prefiksit, suffiksit).	vakaa
0.93	15 / 38	Merkitystasot nimetään (literal / abstract / poetic), Lyric Form uudistuu.	vakaa
0.94	15 / 38	Lyric Form yksinkertaistuu, glossaryentryjen puhtaus varmistetaan.	vakaa ja nopea
0.95	17 / 50	Luovuttiin vaatimasta ylitarkkoja sana- ja alaluokkia, Otettiin käyttöön pehmeäreunainen neljän ish-korin malli.	vakaa ja nopea
0.96	17 / 50	Lisättiin sanaston ylätunnistetiedot	vakaa ja nopea
0.97	17 / 50	Development Release	vakaa ja nopea

Versiohistoria osoittaa, ettei kehitys ollut suoraviivaista ominaisuuksien lisäämistä vaan tasapainoilua ominaisuuksien lisäämisen ja yksinkertaistamisen välillä. Osa säännöistä poistui kokonaan, osa yhdistettiin ja osa korvattiin myöhemmin tarkemmilla periaatteilla. Lopputuloksena syntyi yksinkertaisempi mutta samalla käsitteellisesti vahvempi spesifikaatio.

Version 0.95 suuri alakohtien määrä (50 kpl) selittyy sillä, että määrittämisessä piti luetella ne 17 mahdollista sana- ja alaluokkaa, joita luokiteltiin lempeämmin neljään koriin. Esim. "nounish" = substantiivit ja substantiivin tapaan käyttäytyvät.

- Lisätietoa versiokehityksestä [GGs-versiohistoria](#) (pdf, 2 s. CC BY 4.0, 2026-07-28)
- Maksimaalista kielinörrteilyä sanaluokista [GGs-sanaluokitus](#) (pdf 2 s. CC BY 4.0)

Ongelmia ja ratkaisuja

Testattaessa havaittiin, että tekoäly on altis monenlaisille toimintahäiriöille.

- Tekoäly tuottaa pienen näytteen ja kysyy, halutaanko käsitellä koko aineisto. Vaikka siihen vastaa myöntävästi, tekoäly jää jumiin.
- Tekoäly tuottaa muutaman entryn ja ilmoittaa, että tehtävä on suoritettu. Sillä ei näytä olevan aavistustakaan, että tehtävä jäi kesken.
- Tekoäly alkaa keskeyttää generoimista varmistukseen käyttäjältä, kummalla tavalla toimitaan.

Tekoäly voi epäonnistua tehtävän suorittamisessa:

- jos tekoälylle annettava lähdeteksti sisältää jotain yllättävää, mitä ei ole otettu huomioon.
- jos tehtävään kuuluu tekoälyn mielestä liikaa aikaa
- jos sääntöjä on liikaa, ne ovat monimutkaisia tai ovat ehdottomia
- jos tekoäly suorittaa yhtä aikaa erilaisia, monivaiheista ja raskasta tehtäviä
- jos sanastosta on tulossa niin laaja, että se ylittäisi tekoälylle asetetun ruudulle tuotettavan vastauksen pituusrajan

Yritettäessä ratkaista versio 0.5:n keskeytymisongelmia, kysyttiin siitä myös tekoälyltä. Tämä oli suuri virhe, sillä tekoäly on hyvä generoimaan uusia sääntöjä, mutta kovin huono arvioimaan, ovatko säännöt ja tehtävät sille liian monimutkaisia.

Yksi kehittämistyön perusongelma on, että tekoäly ei anna selkeitä virheilmoituksia. Jos siltä kysyy, mikä meni pieleen, se alkaa generoida mahdollisia selityksiä. Vaikuttaa siltä, että se ei tiedä itsekään. Olen kohdannut tämän useasti.

Fraasit ovat sanaston kannalta hankalia: Fraasin merkitys on erilainen kuin sen osien summa. Ongelma ratkaistiin ohittamalla fraasit sanastosta ja luomalla niiden kääntäminen erillisen työkalun tehtäväksi. Tästä tarkemmin myöhemmin.

On hyvä huomata, että speksin laatu ja generoinnin vakaus ovat eri asioita. Hyväkin määrittäminen voi epäonnistua yksittäisessä ajossa. Ja toisin päin: heikkokin määrittäminen saattaa toimia vakaasti.

Parhaat opit

Nyt kun jälkikäteen palauttaa mieleen GGS:n kehittämisestä saatuja kokemuksia, ymmärtää aiempaa paremmin tekoälyn tuotantokäytön rajoituksia ja ylipäänsä sitä, miten nykyiset keskustelubotit toimivat. Esi-merkiksi sääntöjen kohdalla on löydettävä tarkka tasapaino:

- Jos tärkeitä sääntöjä puuttuu, tekoäly joutuu päättämään, mitä pitää tehdä. Paljon toistuvassa silmukassa tämä hidastaa tehtävän suoritusta ja kuluttaa turhaan laskentatehoa,
- Jos säännöt ovat tulkinnanvaraisia, tekoäly saattaa tuottaa tuloksia, jotka eivät ole sisällöltään sitä tai siinä muodossa kuin käyttäjä ne olisi halunnut.
- Jos sääntöjä on liikaa, tehtävän suoritus hidastuu ja saattaa jumiutua.
- Jos säännöt ovat ehdottomia, normaalista poikkeava lähdeteksti keskeyttää tehtävän suorituksen

Tekoäly ei kerro, milloin käytettävissä oleva kapasiteetti ylittyy. En välitä, onko häiriön syy token-budjetti vai hetkellinen käyttäjäkuorma, mutta yritys- ja erehdys-periaatteella tapahtuvalta kehittämistyöltä puuttuu laskentapohja. Se on kuin lähtisi vielä huhtikuussa pilkille ja toivoisi, että jäät edelleen kantavat.

Tärkein oppi: tekoälyä ei kannata yrittää saada tekemään kaikkea ihmisen puolesta. Parempi työnjako on kolmikohtainen:

1. Tekoälylle annetaan yksi selkeä tehtävä, jossa se on hyvä: sanaston sisältötekstin tuottaminen
2. Ihminen hoitaa tehtävät, jotka vaativat harkintaa. Sanaston viimeistely julkaisukuntoon. sisältötekstin siivoaminen, merkkien ja kappaleiden muotoileminen, linkin ja metadatan lisääminen, pdf-viennin tekeminen ja tiedoston nimeäminen
3. Päätehtävästä irrotettiin osatehtäviä, joita tekoäly suorittaa vain pyydettyäessä. Tällaisia ovat sanaston tarkastaminen ja fraasien kääntäminen.

5. Sanaston luominen

Tässä viidennessä luvussa päästään viimein käytäntöön ja jutun sävy muuttuu kerralla käyttöoppaaksi: miten sanastoja luodaan ja miten tiedonhallinta ei heti karkaa käsistä.

Kun halutaan laatia juuri tietynlainen sanasto, tekoälylle on annettava tarkat ohjeet, miten sanasto tehdään, miltä valmiin sanaston pitää sisältää ja miltä se näyttää ulkoasultaan. Tällainen määrittäminen on siksi monimutkainen, ettei sitä ole järkevää kopioida kehoitteeseen, vaan antaa tekoälylle valmiina tiedostona. Se sisältää ohjeita, sääntöjä ja tehtäviä ja onkin enemmän spesifikaatio kuin kehote. Siitä se sai myös nimensä GGS (*Glossary Generation Specification*). Tämä speksi on julkaistu edelleenkin kehitysversiona, mutta se toimii jo hyvin vakaasti ja nopeasti.

[GGGS-097](#) (docx, 3 s, versio 0.97, CC BY 4.0, 2026-08-11)

Näin luot sanaston

Projektin periaatteena on olla sitoutumatta mihinkään tiettyyn tekoälyyn tai ohjelmistokohtaisiin edistyneisiin ominaisuuksiin. Tästä syystä seuraavat ohjeet eivät anna tarkkoja ”paina tuosta” -tyyppisiä ohjeita eri valmistajien tekoälypalveluille.

1. Etsi lähdeteksti netistä ja kopioi se leikepöydälle
2. Ota esiin tekstieditori, liitä lähdeteksti asiakirjaan ja tallenna se tiedostoksi.
3. Lisää tiedoston nimeen SD-lyhenne (*Source Document*)
4. Ota esiin chatbotti
5. Lataa kehoitteeseen SD ja GGS -tiedostot
6. Kirjoita kehoitteeksi ”luo sanasto SD-tiedostosta noudattaen tarkasti GGS-speksiä”
7. Paina Enter ja jos kaikki menee hyvin, näytölle tulostuu valmis sanasto
8. Kopioi sanasto tekstieditoriin ja tallenna se tiedostoksi (joistain chatboteista tulostuksen voi viedä suoraan tiedostoon)
9. Lisää tiedoston nimeen GD-lyhenne (*Glossary Document*)

Ulkoasun viimeistely

Kun sanastoja alkaa kertyä enemmän, yhtenäinen ulkoasu helpottaa niiden yhdistämistä kokoelmiksi. Itse käytän Segoe UI -fonttia, ykkösen riviväliä sekä korostan selitettävän sanan lihavoinnilla ja sanaluokan kursivoinnilla. Dokumentin alkuun lisään otsikon, linkin, päivämäärän, lisenssitiedot sekä käytetyn GGS-version. Lisää muotoiluasetuksista tulee erilliseen GGM-oppaaseen (*Glossary Generation Manual*).

Esimerkki valmiiksi muotoillusta sanastosta. ReoNa: [ANIMA](#) (pdf, 2 s. CC BY 4.0, EN)

Muista metadata

Lisää sanastotiedostoon metadata jo tekstieditorissa. Tällöin PDF-vienti poimii ne mukaan automaattisesti, otsikko näkyy oikein myös intrassa ja metadataa voi käyttää hakusanoina.

- Otsikko: mikä sanasto tämä on
- Tekijä: Oma nimesi, jos haluat sen julkaista
- Asiasanat: sanasto, japani, tekoäly tms.
- Kommentti: GGS ja sen versio

Jos haluat ilmoittaa olevasi sanaston tekijä, lisää oma nimesi sen metatietoihin tai sanaston alkuun. Jos et halua ilmoittautua tekijäksi, lisää Tekijä-kenttään GGS [versio].

- Wordin sovellusversion metadatakentät löytää valitsemalla Tiedosto > Tiedot.
- LibreOfficessa metadataa pääsee syöttämään valitsemalla Tiedosto > Ominaisuudet.

Vinkit tiedonhallintaan

Sanastot kannattaa tehdä alusta alkaen niin kuin niitä alkaisi tuottaa massoittain. Tiedostojen ja tekoälyn kanssa puuhatessa systemaattisuus on välttämätöntä, kun aletaan käyttää näitä uusia työkaluja. Tiedostoihin liittyvää:

- Saat ruudulle tuotetun sanaston talteen näin: (ChatGPT: Copy Response, Gemini: Tallenna Driveen, Copilot: Vie kohteeseen > Word)
- Voi kopioida sanaston leikepöydälle, liittää tekstieditoriin ja tallentaa sieltä tiedostoksi
- SD:lle ja GD:lle kannattaa tehdä oma tallennuspaikka, jonne ne kaikki tallentaa
- SD ja GD kannattaa nimetä niin, että ne yhdistää ilman, että tiedostoa täytyy avata. Esim. "energetic-sd.docx" ja "energetic-gd.docx" näkyvät kansion tiedostolistassa peräkkäin.

Sanastoa luodessa kannattaa avata chatbotissa uusi keskustelu. Tällöin sen mahdolliset (todennäköiset) virheilmoitukset löytää heti säikeen alkupäästä. Näin se ei myöskään pääse takertumaan johonkin, mitä säikeessä aiemmin on keskusteltu.

6. Sanaston tarkastaminen

Nyt puhutaan sanaston tarkastamisesta virheiden varalta. Jos edellinen osa tuntui käyttöönoton koulutukselta, tässä osassa on äänessä laatupäällikkö.

Millainen on hyvä sanasto

On tärkeää voida luottaa sanastoon, jota opiskelee. Tarve tarkastaa sanaston oikeellisuus kasvaa vielä lisää, kun sanastoja jaetaan eteenpäin tai luovutetaan yhteisiin kokoelmiin. Listataan muutama laatuksiteeri ja mietitään, mitä tämä merkitsee tarkastamisen kannalta.

1. Sanasto kattaa lähdetekstin, eli tekoäly ei ole luovuttanut kesken kaiken. Tarkastusohjelma laskee lähdetekstin sanat, vähentää summasta ohitetut ja duplikaatit ja katsoo, päätyykö samaan summaan kuin sanastossa on entryjä.
2. Sanasto noudattaa rakennetta, eli tekoäly ei ole jättänyt pois jotain sisältöjä liian monimutkaisina. Tarkistuksessa tekoäly käy läpi jokaisen entryn ja liputtaa tarkastettavaksi ne, jotka poikkeavat määritellystä rakenteesta.
3. Sanat kirjoitettuna eri merkistöillä vastaavat toisiaan. Tarkistuksessa tekoäly vertaa sanan kirjoitusasuja luotettaviin nettilähteisiin.
4. Sanojen merkitykset vastaavat yleisiä tulkintoja. Tarkistuksessa tekoäly vertaa eri merkityksiä luotettaviin nettilähteisiin

Tekoäly tarkastaa siis vain sanaston sisällön. Ihmiskäyttäjän tehtävänä on tämän jälkeen tarkastaa sen ulkoasu. Tämä tarkoittaa merkkien, kappaleiden ja sivun ulkoasun muotoilua sekä tiedoston nimeämistä.

Miten sanasto tarkastetaan

GDV (*Glossary Document Verifier*) on erillinen sanaston sisällön tarkastustyökalu. Se syntyi aluksi opettajan käyttötärpeeseen: tutkimaan opiskelijoiden luomia sanastoja. Siitä on kuitenkin hyötyä yhtä lailla oppijoille itselleen sekä muillekin kielten kanssa askarteleville. GDV on omassa tiedostossa oleva määrittely, joka vertaa sanastoa lähdetekstiin, tarkastaa sanaston rakenteen ja sanojen selitykset. Lopuksi se tekee raportin tarkastuksen tuloksista.

Sanaston tarkastustyökalu on julkaistu kehitysversiona, mutta se toimii jo vakaasti ja antaa enää hyvin vähän vääriä hälytyksiä kieliopillisista rajatapauksista.

[GDV-053](#) (docx, 3 s. versio 0.53, CC BY 4.0, 2026-08-11)

Tarkastus on käytännössä yhtä suoraviivainen kuin sanaston luominen. Työvaiheita on hieman enemmän, mutta perusidea on sama: annetaan määrittäminen, annetaan aineisto ja luetaan lopuksi raportti. Seuraavassa työvaiheet, jos sanasto läpäisee tarkastuksen heti ensimmäisellä kerralla.

1. Ota esiin chatbotti
2. Lataa kehoitteeseen SD, GD ja GDV-tiedostot
3. Kirjoita kehoitteeksi "tarkasta oheinen GD-sanasto vertaamalla sitä SD-lähteeseen noudattaen tarkasti GDV-speksiä" ja paina Enter
4. Jos kaikki menee hyvin, näytölle tulostuu valmis raportti
5. Kopioi raportti tekstieditoriin ja tallenna se tiedostoksi (joistain chatboteista sen voi viedä suoraan tiedostoon)
6. Lisää tiedoston nimeen GVR-lyhenne (*Glossary Verification Report*)

Tarkastusraportit voivat näyttää esimerkiksi tällaisilta:

- Band-Maid: [energetic-gvr](#) (pdf, 2 s. CC BY 4.0, 2026-07-15)
- Ai Otsuka: [cherish-gvr](#) (pdf, 1 s. CC BY 4.0, 2026-08-04)

Miten tarkistettu sanasto merkitään

Jos tarkastus ei tuottanut virheitä, voit muuttaa sanaston GD-tunnisteen tarkastetun sanaston VGD-tunnisteksi (*Verified Glossary Document*). Se on sekä dokumentin otsikotiedoissa että tiedostonimessä.

1. Lisää sanaston ylätunnisteseen maininta "Verified by GDV [versio]"
2. Muuta tiedostonimen lyhenne esim. energetic-gd → [energetic-vgd](#)

Miten virheitä korjataan

GDV tuottaa raportin, jossa sanaston tarkistettavat kohdat on listattu ja virhe kuvattu lyhyesti. Tekoäly ei siis korjaa virheitä, vaan sanasto palautuu ihmiskäyttäjän pöydälle.

Jos teet korjauksia sanastoon, jonka kielet tunnet hyvin, huomaat ehkä virheen heti ja voit korjata sen suoraan GD-tiedostoon.

Jos tarkastat oppilaiden tekemiä sanastoja, joissa on suurempia virheitä, laita heille viestissä tarkastusraportti ja pyydä tekemään sanasto uudelleen.

Jos virhe on epävastaavuus länsimaisilla aakkosilla kirjoitetun selitettävän sanan ja perinteisillä kirjoitusmerkeillä kirjoitetun välillä, voit tarkistaa oikean kirjoitustavan SD:stä, lähteen verkkosivulta, katsomalla sanakirjasta tai kysymällä tekoälyltä.

Jos virhe on merkitysten kuvauksissa eri merkitystasoilla, se kannattaa kysyä erikseen tekoälyltä ja tehdä korjaus sanastoon.

Kesken jäänyt tai paljon virheitä sisältävä sanasto kannattaa luoda GGS:llä uudelleen eikä yrittää korjata virheitä yksitellen. Tämän jälkeen sille tehdään uusi tarkastus.

Hyvä keino ratkoa ongelmia on kysyä tekoälyltä, miksi näin kävi. Siltä saa usein hyvää analyysiä, vaikka välillä vain todennäköisyyksiin perustuvaa spekulointia.

7. Fraasit ja kokoelmat

Tässä juttusarjan seitsemännessä osassa kerron kahdesta kätevästä työkalusta. **PET** (*Phrase Extraction Tool*) etsii lähdetekstistä fraaseja ja kääntää ne. **GCC** (*Glossary Collection Compiler*) kokoaa sanastoja yhteen. Hanke eteni hyvää tahtia, sillä nämä kaksi työkalua sekä tämä blogijuttu syntyivät käytännössä yhden päivän aikana.

PET-fraasityökalu

Fraasit ovat kielessä useamman sanan mittaisia merkkijonoja, joiden merkitys on eri kuin niiden osien summa. Japanissa kirjaimellisen ja kulttuurisen merkitysten ero voi olla niin suuri, ettei sanonta mitenkään avaudu tuijottamalla yksittäisiä sanoja. Seuraavassa esimerkki arkielämän fraasista:

yoroshiku onegaishimasu (よろしくお願ひします) 1. (literal) "Please treat me favorably" 2. (social) almost anything from "Nice to meet you" or "Please take care of this" to "I look forward to working with you"

Sanaston ja sanakirjan kannalta fraasit ovat hankalia. Tuodaanko ne sanastoon sana kerrallaan vai kokonaisuutena, aakkostetaanko ne fraasin ensimmäisen sanan vai ydinsanansa mukaan ja kuinka monta kulttuurista merkitystä voidaan tuoda mukaan. Erilaisissa sosiaalisissa tilanteissa fraasi voi merkitä erilaisia asioita, joten pitäisikö myös näitä tilanteita selittää sanastossa? Tällainen sosiaalinen tulkinta on hyvin mielenkiintoista, mutta myös aivan liian laaja aihe tähän juttuun. Se voisi olla kokonaisen juttusarjan aihe.

Aiemmin kehitetyn GGS-sanastogeneraattorin käskettiin ohittaa lähdetekstissä olevat fraasit ja listata ne sanaston loppuun. Tuolloin päätettiin tehdä oma työkalu fraasien etsimiseen ja kääntämiseen. Fraasit opetellaan kokonaisuuksina, joten on hyvä, että ne voi lukea ja ymmärtää myös niin.

Fraasin tunnistustyökalu PET löytyy alla olevasta linkistä. Se on melko yksinkertainen ja ensimmäiseen versioon on tullut vain pientä hienosäätöä. Se on edelleen kehitysversio, mutta toimii vakaasti ja tuottaa kielen opiskeluun hyvin hyödyllistä materiaalia.

[PET-013](#) (docx, 2 s. versio 0.13, CC BY 4.0, 2026-08-11)

Fraasisanaston luominen

1. Ota esiin chatbotti
2. Lataa kehotteen liitteeksi SD-lähdeteksti ja PET-määrittäminen.
3. Kirjoita kehotteeksi "etsi fraasit sd-tiedostosta noudattaen tarkasti pet-speksiä. priorisoi laatu. unohda tehtävään käytettävä aika" ja paina Enter
4. Jos kaikki menee hyvin, näytölle tulostuu valmis sanasto
5. Tallenna tiedosto ja lisää sen nimeen PGD-lyhenne (*Phrase Glossary Document*)

Tässä pari esimerkkiä julkaisukelpoisiksi muotoilluista fraasilistoista:

- Atarayo: [Shinaide-pgd](#) (pdf, 2 s. CC BY 4.0, EN)
- Hakubi: [Twilight-pgd](#) (pdf, 1 s. CC BY 4.0 EN)

GCC-kokoelmatyökalu

GCC (*Glossary Collection Compiler*) on työkalu, mikä kokoaa yksittäiset sanastot isommaksi kokoelmaksi. Yhdessä tiedostossa ja aakkosjärjestyksessä se muistuttaa jo sanakirjaa. Työkalu on yksinkertainen ja hyvin toimiva. Sekin on julkaistu kehitysversiona.

[GCC-013](#) (docx, 2 s. versio 0.13, CC BY 4.0 2026-08-11)

GCC suorittaa käytännössä seuraavat tehtävät:

- lukee sanastoja ja sanastokokoelmia
- laatii niistä lähdeluettelon
- kokoaa sanastot yhteen (ei samaa sanaa kahta kertaa)
- järjestää sanastot aakkosjärjestykseen
- lisää lähdeluettelomerkinnän, missä sanastoissa sana esiintyi
- tulostaa sanaston yhtenä kokonaisuutena
- listaa lopuksi ehdokkaiden, ohitettujen ja hyväksytyjen sanojen määrät

Sanastokokoelman luominen

1. Ota esiin chatbotti
2. Lataa kehotteen liitteiksi GCC-työkalu sekä GD- tai VGD -sanastot
3. Kirjoita kehoitteeksi esim. "kokoa gd- ja vgd-tiedostoista sanastokokoelma noudattaen tarkasti gcc-speksiä. Priorisoi laatu. Unohda käytettävä aika." ja paina Enter
4. Jos kaikki menee hyvin, näytölle tulostuu valmis sanasto
5. Tallenna tiedosto ja lisää sen nimeen GCD-lyhenne (*Glossary Collection Document*)

Vinkkejä tiedonhallintaan

Kun alat tehdä sanastoja, tiedostoja alkaa kertyä nopeasti. Parhaat vinkit asioiden pitämiseen hallussa.

1. Nimeä tiedostot huolellisesti, esim. [laulun_nimi]-[dokumenttikoodi]
2. Tallenna kaikki luomasi tiedostot samaan hakemistoon
3. Älä muuttele tiedostonnimiä jälkikäteen
4. Älä luota tekoälyn tiedostopuskuriin arkistona

8. Arvot ja periaatteet

Kokoan seuraavassa yhteen arvoja ja periaatteita, joiden varaan tätä hanketta on rakennettu. Äänessä on nyt esseisti, mutta paikalla oleva tekninen dokumentoija palauttaa välillä puheen käytännön tasolle.

Arvot kaiken takana

Arvoista puhuminen tuntuu usein vähän juhlalliselta ja poseeraamiselta, Niiden kautta kuitenkin näkee, mitkä ovat yleiset tavoitteet ja millaisilla keinoilla niihin halutaan päästä. Niistä puhutaan harvoin arjessa, mutta yritän tehdä nyt juuri sitä.

Avoimuus on aina hyvä periaate. En lähtenyt kokoamaan valmista sanakirjaa, enkä digitaalista oppimis-sovellusta, vaan tutkimaan, voisiko kehittää menetelmän, jolla sanastoja voisi luoda ihan itse. Halusin, ettei menetelmä jäisi yhden ihmisen tai yhden palvelun varaan, joten halusin julkaista tuotokset vapaasti käytettäväksi. Sanastoja kun tarvitsevat kaikki kieltä opiskelevat ja opettavat. Tuskinpa tekoälylle annettavaa kehoitetta voisi edes suojatakaan, joten yritin lähteä tekemään ratkaisua, jonka jokainen voisi ottaa käyttöönsä helposti, ilmaiseksi ja laillisesti.

Toimijuus on luonteva jatko edelliselle. Kun oppija voi luoda itselleen oppimateriaalin aiheesta, joka häntä aidosti kiinnostaa, esim. japanilaisesta ruokareseptistä, hän on heti motivoituneempi opiskelemaan juuri hänelle räätälöityä sanastosisältöä. Musiikkifani voi etsiä netistä suosikkibändinsä uutuussinglen sanoitukset ja opetella sen sanat heti, kun kappale on julkaistu. Käyttämällä GGS:n kaltaista työkalua oppija voi ilman syvällistä tekoälyosaamista valita itse, mitä opiskelee. Tuolloin hän ei ole pelkkä oppimateriaalin kuluttaja, vaan hänestä tulee sen tuottaja. Samaa toimijuutta voi myös nähdä sanastojen luomisessa ajan-kohtaisista sisällöistä. Uutissähkeen antaminen sanastogeneraattorin lähdetekstiksi vahvistaa oppijan omaa toimijuutta yhteiskunnan aktiivisena jäsenenä.

Sitoutumattomuus on myös yksi hankkeen kulmakivistä. En lähtenyt kehittämään ratkaisua millekään tietylle tekoälypalvelulle, en mitoittanut kapasiteettitarvetta niiden mukaan enkä käyttänyt niiden kehittyneitä ominaisuuksia, kuten agenttia, APlä tai koodi-ikkunaa. Tämä siksi, että tekoälyjen toiminnot muuttuvat jatkuvasti. Yleisiin perusominaisuuksiin tyytymällä työkalun elinkaari on varmasti pidempi.

En ole testannut, kuinka laajoja lähdeaineistoja jonkun tietyn tekoälyn ilmainen tai maksullinen versio kestää, sillä järjestelmät rajoittavat käyttäjän toimintaa mm. hetkittäisen käyttäjämäärän ja järjestelmän kuormitusasteen mukaan. Tekoälyt eivät myöskään osaa tai niillä ei ole lupa itse kertoa kapasiteetin rajoituksistaan. Niihin törmää vasta kun tehtävän suorittaminen ei onnistu. Jos esimerkiksi antaisit GGS:lle

kokonaisen kirjan lähdetekstiksi, tavallisen kotikäyttäjän ilmaiset chatbotit eivät siitä selviytyisi (kesällä 2026). Teknologia kehittyy, mutta sen nykyiset rajat löytää vain kokeilemalla itse.

Muokattavuus on tärkeää. Jos työkalua voi itse muokata, sen saa paremmin vastaamaan omia tarpeita. Ihmiset myös oppivat eri tavoilla. Joku saattaa kaivata lyhyttä esimerkkilauseetta, mistä voi opetella sanan käyttötapaa, miten se näkyy lauseen muodossa tai miten sitä on taivutettu. Tällaisen esimerkkilauseen liittäminen sanastointien speksiin ei ole edes vaikeaa. GGS:ä saa ja sitä voi vapaasti muokata.

- Kaikki aineisto on julkaistu Creative Commons -lisenssillä BY 4.0.
- Speksit ja tehtävän määritykset ovat selkeää englantia.
- Speksitiedostot ovat normaaleja Word-muotoisia dokumentteja.
- Lyhenteet on listattu ja niitä käytetään systemaattisesti.
- Kryptisiä muuttujien nimiä ei ole käytetty. Lähes kaikki on toteutettu luonnollisella kielellä.

Järjestelmä on rakennettu tuottamaan japaninkielisestä lähdetekstistä englanninkielisin selityksin varustettuja sanastoja. Pienillä muutoksilla sen saa tuottamaan esimerkiksi suomalaisin selityksin varustettuja sanastoja englanninkielisestä lähdetekstistä.

Suunnitteluperiaatteita

Heti alussa oli selvää, ettei sanastoja ja muita tuotoksia ollut mielekästä toteuttaa minkään tietyn verkkopalvelun varaan. Ei siis natiivisti blogiin, kyselytyökaluun tai oppimisalustalle, vaan jakelualustasta riippumattomaan ja siirrettävään muotoon. Yksinkertaisin vaihtoehto oli tehdä tuotokset tiedostopohjaisiksi.

- Helppo viedä koulun käyttämälle oppimisalustalle
- Helppo arkistoida kotikoneelle, matkapuhelimeen tai koulun palvelimelle
- Tiedostot eivät saastu palvelutarjoajan mainoksista
- Luettavuus on hyvä myös mobiililaitteilla ja linkit toimivat
- docx- ja pdf-tiedostoja voi luoda ilman maksullisia ohjelmia tai verkkopalveluita
- Opettaja voi helposti lähettää oppilaalle sanastotiedoston ja tehtävänannon
- Näppärimmät oppilaat voivat luoda sanastoja itse ja lähettää niitä opettajalle

Laadin GGS:n suoraan englanniksi, sillä ajattelin sen käyttäjiksi J-rock-fanit ja japanin itseopiskelijat ympäri maailmaa. Suomessa meitä on vain vähän, mutta maailmalla enemmän. Työkalua ei siis kannata sitoa vain yhdelle kielialueelle. Tämän lisäksi kielimallien käyttämä perustason luonnollinen kieli on englanti. Kun tekoälylle antaa ohjeet suoraan englanniksi, sen ei tarvitse ensin kääntää niitä suomesta.

Tavoitteena oli aluksi laatia kehote, joka tuottaisi heti julkaisuvalmiin sanaston. Sellaisen, jota ei tarvitsisi ohjata eikä muotoilla manuaalisesti. Eipä käynyt aivan noin helposti. Heti ensimmäiset kokeilut osoittivat, että tekoälylle täytyy määritellä sanaston sisältö ja ulkoasu todella tarkasti. Jos tulkinnanvaraa jää, tekoäly tekee, miten parhaaksi näkee tai kyselee käyttäjältä tarkempia ohjeita.

Parhaat opit

Sanaston luominen ei ole kehotteessa annettava käsky, vaan enemmän määritystä, millainen sanaston pitää olla. Tämä oli kätevintä laatia erilliseen tiedostoon ja antaa kehotteen liitteenä. Hankkeen alkupuolen ongelmat keskittyivät juuri tämän GGS-määrittelyn (*Glossary Generation Specification*) kehittämiseen, testaamiseen, vianetsintään ja tulkinnanvaraisuuden eliminoimiseen.

kokeilut opettivat, että on parempi tyytyä maksimaalisia tavoitteita vähempään, riittävään ja kohtuulliseen tulokseen. Tämä tuli vastaan monessa kohtaa:

- Tekoälylle on parempi antaa yksi selkeä tehtävä ja kohdella sitä kuin työkalua, joka tekee yhden vaiheen isommasta kokonaisuudesta. Asioita ei ehkä kannata tehdä yhdellä jättikäskyllä alusta loppuun vaan tehtävät muodostavat työnkulun, prosessin tai tuotantoketjun.

- Tekoälyn ohjauksessa oli tärkeä löytää tasapaino sääntöjen määrässä, tarkkuudessa ja ehdottomuudessa. Liian paljon, liian tarkkoja ja ehdottomia sääntöjä estää tehtävän suorittamisen. Toisaalta puuttuvat tai liian väljät säännöt tuottavat vaihtelevan laatuista tuloksia.
- Tekoäly menee helposti liian syvälle yksityiskohtiin. Turhia virheilmoituksia alkoi tulla esimerkiksi siitä, että sana voitiin tulkita kuuluvan kahteen eri sanaluokkaan. Ratkaisu löytyi niin, että määriteltiin sanoille pehmeärajaiset luokitukset (esim. "verbish").

Toisinaan tekoäly epäonnistui tehtävän suorittamisessa, kun se pyrki täyteen varmuuteen sanan selityksessä. Tämä korjattiin lopulta lisäämällä laatuperiaate: "Kun täysi varmuus on mahdotonta, käytä todennäköisintä lingvististä analyysiä ennen kuin ohitat sanan tai keskeytät tehtävän suorittamisen."

Yksi japanilaisen estetiikan, yhteiskunnan ja päätöksenteon keskeisiä käsitteitä on **wa** (和). Se tarkoittaa mm. harmoniaa, rauhaa ja ryhmähenkeä. Sovelsin siitä käyttäjän ja tekoälyn tasapainoa kuvaavan termin **Alwa** (アイ和). Se kuvaa omien tavoitteiden sovittamista ilman konflikteja siihen, mikä on yleisesti mahdollista. Se on minusta realistinen ja käytännöllinen asenne, kun toimitaan tekoälyn kanssa.

"Tehdään, mitä voidaan sillä, mitä meillä on."

Alwa näkyy ehkä parhaiten yhdessä projektin ratkaisevista keskusteluista. Liitän tähän transkription, jossa GGS:n ja GDV:n sääntöjä sovitetaan yhteen. Kieliopin yksityiskohtia tärkeämpää on nähdä, miten rakentava vuoropuhelu toimii kehittämisen välineenä. [convo-2026-07-13](#) (pdf, 14 s. CC BY 4.0, FI)

9. Kokeile itse

Tässä viimeisessä luvussa pääset kokeilemaan työkaluja itse. Voit käyttää haluamaasi chatbottia. Prosessi vie noin kymmenen minuuttia. Sen jälkeen puhutaan tekijänoikeuksista ja tulevista julkaisuista.

Sanaston luominen

1. Lataa demolyriikka tästä: [mukaigawanoseki-sd](#) (pdf, 1 s. JP)
2. Lataa sanastogeneraattori tästä: [GGS-097](#) (docx, 3 s. EN)
3. Ota esiin chatbotti
4. Liitä kehoitteeseen SD ja GGS-tiedostot
5. Kirjoita kehote "Luo sd-tiedostosta sanasto noudattaen tarkasti ggs-speksiä"
6. Paina Enter ja jos kaikki menee hyvin, näytölle tulee valmis sanasto.
7. Kopioi sanasto leikepöydälle ja siitä tekstinkäsittelyohjelmaan.
(joissain chatboteissa voit viedä tuotoksen suoraan tiedostoon)
8. Tallenna sanasto nimellä "mukaigawanoseki-gd" (*Glossary Document*)

Nyt sinulla on valmis sanasto!

Sanaston tarkastaminen

1. Lataa tarkastustyökalu tästä: [GDV-053](#) (docx, 3 s. EN)
2. Ota esiin chatbotti
3. Liitä kehoitteeseen SD, GD ja GDV -tiedostot
4. Kirjoita kehote "tarkasta gd-tiedosto. vertaa sitä sd-tiedostoon. noudata tarkasti gdv-speksiä"
5. Paina Enter ja jos kaikki menee hyvin, näytölle tulee tarkastusraportti
6. Kopioi se leikepöydälle ja siitä tekstinkäsittelyohjelmaan
(joissain chatboteissa voit viedä tuotoksen suoraan tiedostoon)
7. Tallenna raportti nimellä "mukaigawanoseki-gvr"
(*Glossary Verification Report*)

Voit nyt raportin perusteella tehdä sanastoon tarvittavat korjaukset.

Fraasisanaston luominen

1. Lataa fraasityökalu tästä: [PET-013](#) (docx, 2 s. EN)
2. Ota esiin chatbotti
3. Liitä kehoitteeseen SD ja PET-tiedostot
4. Kirjoita kehote "etsi sd-lähdetekstistä fraasit noudattaen tarkasti pet-speksiä" ja paina Enter
5. Jos kaikki menee hyvin, näytölle tulee fraasisanasto
6. Kopioi sanasto leikepöydälle ja liitä se tekstieditoriin (joissain chatboteissa voit viedä tuotoksen suoraan tiedostoon).
7. Tallenna fraasisanasto nimellä "mukaigawanoseki-pgd" (*Phrase Glossary Document*)

Nyt sinulla on fraasisanasto!

Tekijänoikeuksista

Alkuperäisten artistien oikeuksia on tärkeää kunnioittaa. Laulusanoitukset ovat tekijänoikeuden suojaamia taiteellisia teoksia. Sen sijaan sanasto ei ole laulutekstin kopio, vaan sanojen ja niiden merkitysten luettelo. Se on laadittu oppimateriaaliksi ja julkaistu vapaasti käytettäväksi. Älä kuitenkaan julkaise muiden tekemiä lähdetekstejä. Pidä ne vain omassa käytössäsi.

Lisään luomiini sanastoihin aina kappaleen ja artistin nimet sekä YouTube-linkin artistin kanavalle kyseiseen kappaleeseen. Se on reilua ja auttaa artistia tekemään musiikkiaan tunnetuksi. Se auttaa myös opiskelijaa, kun kappaleen voi kuunnella yhdellä hiiren napsauksella.

Lopuksi vielä tärkeä ero: tekijänoikeus ja käyttö lupa ovat eri asioita. Tekijänoikeuden haltija voi myöntää teokselleen käyttö lupia eli lisenssejä. Julkaisen tässä luomani aineistot CC BY 4.0 -lisenssillä, jolloin niitä saa käyttää, kehittää ja jakaa edelleen tekijä mainiten. Lisätietoa [Creative Commons BY 4.0](#) -lisenssistä.

Demolyriikasta

Sanastotyökalujen kokeileminen on helppoa, kun lähdeteksti on jo valmiiksi tiedostossa. En kuitenkaan voi julkaista täällä mitään tekijänoikeuden suojaamaa sisältöä. Enkä halua pyytää tekoälyä generoimaan demolyriikoita, sillä en voi olla varma, ottaako se ne jostain olemassa olevasta kappaleesta.

Niinpä kirjoitin pienen laulutekstin itse. Voin julkaista sen tässä, sillä minulla on sen tekijänoikeudet ja voin myöntää sen käytölle lisenssin. Kirjoitin lyriikan suomeksi ja tekoäly käänsi sen minulle japaniksi.

Demolyriikka: [Mugaikawa no seki \(向かい側の席\)](#) (pdf, 1 s. CC BY 4.0 JP)

Laulun sanoitus on kahdessa muodossa. Japanilaiset merkit kertovat, mitä sanat tarkoittavat ja länsimaiset merkit, miten ne lausutaan. Näin myös aloitteleva japanin opiskelija voi ne lukea.

Tavoittelin sanoituksessa *Hakubin* inhimillisyyttä, *Yuurin* kerrontatapaa ja *Atozomen* rohkeutta. Laulusta tuli myös mieleen *Aisagasu Eye Shadown* kappale *Mayonaka no Riri (Midnight Lily)*. Samanlaisuus ei kuitenkaan ole sanoissa eikä fraaseissa, vaan syvemmällä tarinan asetelmassa ja sanomassa. Siinä, miten jotkut satunnaiset kohtaamiset jäävät viipymään ja muuttavat meitä vähän.

En julkaise tässä alkuperäistä suomenkielistä sanoitustani, sillä sehän pilaisi tutkimisen ja ymmärtämisen ilon. Jos joku haluaa nähdä tuon kirjoittamani laulutekstin, voi laittaa viestiä, niin lähetän sen.

Tulevat julkaisut

Juttusarjan päättymisen jälkeen on vielä tulossa erimuotoista dokumentaatiota.

- **Sanaston synty e-kirja** on koko yhdeksänosainen juttusarja yksissä kansissa. Sen saa PDF-muodossa Saskyn blogista (blogit.sasky.fi) sekä OPH:n Avoimien oppimateriaalien kirjastosta (aoe.fi).
- **GCD** (*Glossary Collection Document*) on esimerkki sanastokokoelmasta. Se on sanakirja, joka sisältää suosikkibändini parhaan albumin kaikki kappaleet. Julkaisuaikataulu: elokuu 2026.
- **GGM** (*Glossary Generation Manual*) on PDF-dokumentti, jossa neuvotaan tässä hankkeessa kehitettyjen työkalujen käyttöä, muokkaamista, dokumenttien ulkoasun muotoilua ja oppilaitoksen sanastoarkiston tiedonhallintaan. Julkaisuaikataulu: syyskuu 2026.

Lopuksi

Hankkeen aikana mittakaava muuttui itseopiskelijan tiedontarpeesta uudenlaisia mahdollisuuksia avaavaksi sisällöntuotannon menetelmäksi. Lopputuloksena on joukko avoimia määrittämiä ja työnkulkua, joita kuka tahansa voi käyttää ja kehittää edelleen. Se on paljon enemmän kuin osasin odottaa. Pieni kesäprojektini karkasi hauskesti käsistä. Se opetti myös, ettei hyväkään idea tule yhtäkkiä valmiina. Sen kehittäminen toimivaksi ratkaisuksi vaatii paljon kypsyttelyä ja monta pientä kehitysaskelta.

Liitteet

[GGG-097](#) sanastogeneraattori (docx, 3 s, versio 0.97, CC BY 4.0, 2026-08-11, EN)

[GDV-053](#) tarkastustyökalu (docx, 3 s. versio 0.53, CC BY 4.0, 2026-08-11, EN)

[PET-013](#) fraasityökalu (docx, 2 s. versio 0.13, CC BY 4.0, 2026-08-11)

[GCC-013](#) kokoelmatyökalu (docx, 2 s. versio 0.13, CC BY 4.0 2026-07-31, EN)

[convo-2026-07-13](#) tekoälykeskustelu (pdf, 12 s. 2026-07-13, FI)

[GGG-versiohistoria](#) (pdf, 2 s, 2026-07-28, CC BY 4.0, FI)

[GGG-sanaluokitus](#) (pdf, 1 s. 2026-07-26, CC BY 4.0, FI)

[GGG-wordcategories](#) (pdf, 1 s. 2026-07, CC BY 4.0, EN)

Demolyriikka: [mukaigawanoseki-sd](#) (pdf, 1 s. CC BY 4.0, JP)

Atozome: [namonakitamashii-gd](#) (名もなき魂) -sanasto (pdf 4 s. CC BY 4.0, FI)

Atarayo: [shinaide-vgd](#) -sanasto (pdf, 3 s. CC BY 4.0 EN)

Atarayo: [shinaide-pgd](#) -fraasisanasto (pdf, 2 s. CC BY 4.0, EN)

Hakubi: [twilight-vgd](#) -sanasto (pdf, 3 s. CC BY 4.0, EN)

Hakubi: [twilight-pgd](#) -fraasisanasto (pdf, 1 s. CC BY 4.0 EN)

HACHI: [hoshiwomatsu-gd](#) (星を待つ) -sanasto (pdf, 4 s. CC BY 4.0, EN)

ReoNa: [anima-vgd](#) -sanasto (pdf, 2 s. CC BY 4.0, EN)

Band-Maid: [energetic-vgd](#) -sanasto (pdf, 2 s. CC BY 4.0 EN)

Band-Maid: [energetic-gvr](#) -tarkastusraportti (pdf, 2 s. CC BY 4.0, EN)

Ai Otsuka: [cherish-gvr](#) -tarkastusraportti (pdf, 1 s. CC BY 4.0, EN)