



Pro gradu -tutkielma

Tietojenkäsittelytieteen maisteriohjelma

# Kielimallien luovuuden kriteerit

Iikka Hauhio

31.8.2022

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA  
HELSINGIN YLIOPISTO

**Ohjaaja(t)**

Prof. Hannu Toivonen, FT Anna Kantosalo

**Tarkastaja(t)**

Prof. Hannu Toivonen, FT Anna Kantosalo

**Yhteystiedot**

PL 68 (Pietari Kalmin katu 5)

00014 Helsingin yliopisto

Sähköpostiosoite: [info@cs.helsinki.fi](mailto:info@cs.helsinki.fi)

URL: <http://www.cs.helsinki.fi/>

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen maisteriohjelma	
Tekijä — Författare — Author			
Iikka Hauhio			
Työn nimi — Arbetets titel — Title			
Kielimallien luovuuden kriteerit			
Ohjaajat — Handledare — Supervisors			
Prof. Hannu Toivonen, FT Anna Kantosalo			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu -tutkielma		31.8.2022	54 sivua, 13 liitesivua
Tiivistelmä — Referat — Abstract			
<p>Kielimallit ovat tilastollisia malleja, jotka ennustavat kuinka todennäköinen annettu jono sanoja on. Kielimallin tuottamasta jakaumasta on mahdollista arpoa sanoja, lauseita ja kokonaisia tekstejä. Viime aikoina kielimalleja on onnistuneesti käytetty taiteen, kuten tarinoiden, runojen ja musiikkikappaleiden generointiin. Tutkijat ovat kutsuneet kielimallien tuottamia tekstejä erinomaisiksi, mutta samalla niiden luovuutta on myös kyseenalaistettu.</p> <p>Tutkielmassa esittelen riittävän kriteeristön kielimallien luovuuden arviointiin. Kriteereiksi määrittelen teosten <i>arvokkuus</i> ja <i>uutuus</i> sekä kielimallin <i>ennustamattomuus</i>. Tutkin kriteerit täyttävän kielimallin luovia tekoja FACE-mallin avulla. Poissuljen sen mahdollisuuden, että kaikki luovat teot olisivat mallin kouluttajan tai lähdeaineston tai koulutusalgoritmin tekijöiden tekemiä.</p> <p>Lopputulokseni on, että kriteerit täyttävä kielimalli tekee useita luovia tekoja. Kriteerien toteutumisen mittaaminen voi olla käytännössä haastavaa, sillä kriteerien tarkka määrittely on tehtävä tapauskohtaisesti, eikä niiden mittaamiseen ole yhtä selkeää tapaa.</p>			
<p><b>ACM Computing Classification System (CCS)</b> Computing methodologies → Cognitive science Applied computing → Arts and humanities</p>			
Avainsanat — Nyckelord — Keywords			
laskennallinen luovuus			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsingin yliopiston kirjasto			
Muita tietoja — övriga uppgifter — Additional information			



# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Tausta</b>	<b>3</b>
2.1	Laskennallinen luovuus . . . . .	3
2.1.1	Bodenin luovuuden eri lajit . . . . .	3
2.1.2	Ritchien menetelmä . . . . .	4
2.1.3	Coltonin luova kolmijalka . . . . .	5
2.1.4	FACE . . . . .	6
2.1.5	IDEA . . . . .	7
2.1.6	”Lovelace-testi” . . . . .	8
2.1.7	Pelkkä generointi ja luovuus . . . . .	9
2.2	Kielimallit . . . . .	10
2.2.1	Neuroverkkokielimallit . . . . .	11
2.2.2	Kielimallien käyttö taiteessa ja laskennallisessa luovuudessa . . . . .	12
2.2.3	Aiempiä arvioita kielimallien luovuudesta . . . . .	13
<b>3</b>	<b>Kielimallin luovuuden kriteerit</b>	<b>15</b>
3.1	Kriteerien yksityiskohtainen kuvaus . . . . .	16
3.1.1	Teosten arvokkuus ja uutuus . . . . .	16
3.1.2	Kielimallien ennustamattomuus . . . . .	20
3.2	Luovat teot . . . . .	21
3.2.1	Konseptin luominen . . . . .	21
3.2.2	Esteettisyysmetriikan luominen . . . . .	24
3.2.3	Huomio deterministisyydestä . . . . .	25
3.3	Tavoitteellisuus . . . . .	26
3.3.1	Generoinnin aikaiset tavoitteet . . . . .	27
3.3.2	Tavoitteet kielimallin taustalla . . . . .	28

<b>4</b>	<b>Esimerkkitapaus: säveltävä</b>	
	<b>kielimalli</b>	<b>31</b>
4.1	Aineisto . . . . .	31
4.2	Tekstipohjainen muoto . . . . .	32
4.3	Kielimalli . . . . .	32
4.4	Ennustamattomuus . . . . .	35
4.5	Luovuuden arvioinnin mittarit . . . . .	35
	4.5.1 Uutuus . . . . .	35
	4.5.2 Tyypillisuus . . . . .	37
	4.5.3 Arvo . . . . .	37
4.6	Arvioinnin tulokset . . . . .	38
	4.6.1 Uutuus . . . . .	38
	4.6.2 Tyypillisuus . . . . .	38
	4.6.3 Arvo . . . . .	39
4.7	Tulosten tulkinta . . . . .	39
<b>5</b>	<b>Pohdintaa</b>	<b>43</b>
5.1	Arvokkuuden ja uutuuden määrittely . . . . .	43
5.2	Ennustamattomuuden mittaaminen . . . . .	43
5.3	Generoinnin aikaisten tavoitteiden syntyminen . . . . .	44
5.4	Tietoisuus . . . . .	44
5.5	Johtopäätösten yleistyminen . . . . .	45
<b>6</b>	<b>Johtopäätökset</b>	<b>47</b>
	<b>Kirjallisuus</b>	<b>51</b>
	<b>A Tunnehaikuja</b>	
	<b>B Säveltävän ohjelman generoimia kappaleita</b>	

# 1 Johdanto

Kielimalli on tilastollinen malli, joka pyrkii ennustamaan, mitkä sanat kuuluvat annettuun kontekstiin. Usein tämä toimii antamalla mallille *kehote*, minkä jälkeen malli ennustaa, mikä sana tulisi todennäköisesti kehotteen jälkeen. Esimerkki yksinkertaisesta tekstiä tuottavasta kielimallista on Markovin ketju, joka ennustaa yhden sanan mittaisesta kontekstistä, mikä on seuraava sana [9]. Markovin ketju on liian yksinkertainen tuottaakseen hyviä tuloksia, sillä kielen sanat eivät riipu vain edellisestä sanasta. Tuloksena on yleensä kieliopillisesti vääriä tekstikatkelmia, joilla ei ole mitään havaittavaa merkitystä.

Viime aikoina on kuitenkin pystytty kehittämään malleja, joilla voi generoida hyvää tekstiä. Suuret syväoppimiseen pohjautuvat kielimallit, kuten GPT-2 [31] ja GPT-3 [8] kykenevät tuottamaan kieliopillisesti oikeaa ja ainakin näennäisesti merkityksellistä tekstiä, kuten runoja, tarinoita, näytelmäkirjoituksia, tiivistelmiä ja kuvitteellisia uutisia. GPT-3:n tuloksia on kutsuttu laadultaan erinomaisiksi [19]. Kielimallien kykyjä on myös kyseenalaistettu. Floridi ja Chiriatti [19] toteavat, että GPT-3 on yhtä älykäs kuin vanha kirjoituskone, eikä ymmärrä mitään mitä kirjoittaa.

Laskennallinen luovuus on tietokoneohjelmien luovuutta tarkasteleva tieteenala [14]. Eräs suosittu tutkimuskohde allalla on kirjallisten taideteosten, kuten tarinoiden ja runojen, koneellinen tuottaminen. Kielimallit ovatkin laskennallisen luovuuden näkökulmasta erittäin kiinnostavia, ja sekä Markovin ketjuja että neuroverkkokielimalleja on käytetty luovien järjestelmien osina [1][2].

Eräs tärkeä osa laskennallista luovuutta on sen arviointi, kuinka luovia kehitetyt järjestelmät ovat [14]. Neuroverkkokielimallien arviointi on kuitenkin ongelmallista, sillä monimutkaiset neuroverkot ovat ”mustia laatikoita”, joiden sisäistä toimintaa on hyvin vaikea tutkia. Tästä syystä on tarve työkalulle, jolla kielimallien luovuutta voi tarkastella.

Tavoitteeni tässä gradussa on esitellä riittävä kriteeristö, jonka täyttämällä kielimallin voi sanoa tekevän luovia tekoja. Tarkastelen kriteeristön mielekkyyttä sekä luovien tekojen merkitystä. Demonstroin kriteeristöä soveltamalla sitä kouluttamaani säveltävään kielimalliin.



# 2 Tausta

Tässä osassa kerron taustatietoa laskennallisesta luovuudesta ja kielimalleista.

## 2.1 Laskennallinen luovuus

Laskennallinen luovuus on tietokoneohjelmien luovuutta tutkiva tieteenala. Laskennalliselle luovuudelle ei ole olemassa yhtä hyväksyttyä määritelmää. Eräs yleinen määritelmä on:

Laskennallisesti luovaa on sellaisten tehtävien suorittaminen, joita pidettäisiin luovina, jos ihminen suorittaisi ne. [38]

Tässä tutkielmassa käytän niin kutsuttua ”luovuuden standardimääritelmänä”, jonka mukaan luovuus on *arvokkaiden* ja *uusien* teosten tuottamista [34]. Ventura [37] pitää laskennallisen luovuuden allalla näiden lisäksi tekojen *tarkoituksellisuutta* edellytyksenä luovuudelle.

Luovuuden arviointiin ja mallintamiseen on kehitetty useita menetelmiä, joista osaa käsittelen tässä luvussa. Koska menetelmiä on monia, on suositeltavaa käyttää useaa eri menetelmää ohjelman luovuuden arviointiin [24]. Jordanous [26] ehdottaa menetelmien valintaan omaa SPECS-menetelmäänsä, jossa valitaan ohjelman kannalta relevantit luovuuden osa-alueet ja näitä osa-alueita arvioivat menetelmät. Hän on lingvistisellä analyysillä tunnistanut 14 luovuuteen liittyvää sanaa, joita voi käyttää hyödyksi luovuuden osa-alueita valittaessa.

Kuvailen tässä osassa myöhempien osien kannalta relevantteja käsitteitä sekä luovuuden arviointimenetelmiä.

### 2.1.1 Bodenin luovuuden eri lajit

Boden määrittelee kirjassaan *The Creative Mind* [5] useita tapoja lajitella koneluovuutta. Yksi jaottelu koskee *H-luovuutta* (”H-creativity”) ja *P-luovuutta* (”P-creativity”). H-luovat teokset tai ideat ovat uusia koko ihmiskunnalle, kun taas P-luovat teokset

tai ideat ovat uusia vain tekijälle itselleen. Laskennallisen luovuuden yhteydessä saat-  
taa olla kiinnostavampaa tarkastella P-luovuutta kuin H-luovuutta, vaikka H-luovat  
teokset ovatkin arvokkaampia ihmisille kuin P-luovat teokset.

Toinen Bodenin käyttämä jaottelu koskee luotujen teosten suhdetta aiempiin teoksiin  
ja konseptiavaruuteen eli mahdollisten teosten joukkoon. *Yhdistelevässä luovuudessa*  
("combinatory creativity") uudet ideat ovat yhdistelmiä aiemmista tutuista ideoista.  
*Tutkivassa luovuudessa* ("exploratory creativity") tutkitaan konseptiavaruutta. *Mullis-  
tavassa luovuudessa* ("transformative creativity") koko konseptiavaruutta muunnetaan.  
Boden pitää mullistavaa luovuutta kaikista luovimpana näistä kolmesta.

### 2.1.2 Ritchien menetelmä

Ritchie [32][33] esittää, että luovuuden arvioinnin tulisi perustua niihin empiirisiin te-  
kijöihin, jotka saavat ihmiset pitämään toisia ihmisiä luovina. Koska ihmisillä ei toisia  
ihmisiä arvioitaessa ole useinkaan pääsyä tekijän sisäisiin tuntemuksiin, ehdottaa hän  
pelkästään teosten arviointia [33]. Jos teoksen luomismenetelmä on tiedossa, voi si-  
tä arvioida hänen mukaansa erikseen omana artefaktinaan. Ritchie toteaa, että pelkän  
teoksen arvionnissa on myös se hyöty, ettei arviointimenetelmä ota ennenaikaisesti kan-  
taa luovan prosessin rakenteeseen eikä siten rajaa itseään perusteetta vain tietynlaisiin  
luoviin prosesseihin [33].

Ritchie [33] ehdottaa artefaktien arviointiin menetelmää, jossa mitataan artefaktien  
*tyypillisyyttä* ja *arvoa*. Tyypillisuus kertoo, onko teos artefaktiluokkansa edustaja, ja  
arvo mittaa teoksen koettua laatua.

Ritchien [33] mukaan tyypillisyyttä on syytä mitata, sillä monet luoviksi väitetyt oh-  
jelmat eivät oikeasti kykene tuottamaan halutunlaisia teoksia: ihminen ei tunnista ru-  
noilevan ohjelman tekstejä runoiksi eikä vitsigeneraattorin tuotoksia vitseiksi. Ritchie  
pitää yhtenä luovuuden kriteereistä, että riittävän suuri osuus tuotetuista teoksista on  
tyypillisiä. Kuitenkin hän toteaa myös, että tyypillisuus on keino mitata uutuutta. Epä-  
tyypilliset mutta arvokkaat teokset saattavat kertoa, että ohjelmalla on epätyypillisiä  
taitoja ja että ohjelma kykenee rikkomaan artefaktiluokan rajoja. Hän pitää tätä koko-  
naan toisena luovuuden tasona [32]. Luovuus vaatii siis sekä kykyä tuottaa tyypillisiä  
teoksia että kykyä rikkoa tyypillisyyden rajoja tarvittaessa.

Tyypillisyyden ja arvon metriikkoja ei määritellä yleisesti, vaan ne jäävät mallin käyt-  
täjän päätettäväksi. Näiden kahden tekijän lisäksi Ritchie pyrki tutkimaan teosten *uu-*

*tuutta* testaamalla, kuuluvatko ne lähdeaineistoon (engl. *inspiring set*). Ritchie toteaa vastauksena hänelle esitettyyn kritiikkiin, että identtisten kopioiden etsimisen lisäksi on mahdollista tutkia teosten *samankaltaisuutta* lähdeaineiston teosten kanssa.

Ritchie [33] ehdottaa 18 näistä johdettuja kriteeriä luovuudelle, kuten ”hyvien tyypillisten artefaktien osuus tyypillisistä artefakteista” ja ”hyvien epätyypillisten artefaktien osuus uusista artefakteista”.

Ritchie [33] kutsuu hyvien tyypillisten ja epätyypillisten artefaktien osuutta uusista artefakteista ”tavanomaiseksi” ja ”epätavanomaiseksi taidoksi”. Viimeinen kriteeri mittaa hänestä ehkä kaikista eniten luovuutta.

### 2.1.3 Coltonin luova kolmijalka

Colton [11] esittää luovuuden koostuvan kolmesta osa-alueesta: *taidosta* (”skill”), *arviointikyvystä* (”appreciation”) ja *mielikuvituksesta* (”imagination”). Hän nimittää tätä mallia *luovaksi kolmijalaksi* (”the creative tripod”). Coltonin mukaan luovalla järjestelmällä on oltava taitoa artefaktien tuottamiseen, arviointikykyä hyvien artefaktien tunnistamiseen, ja mielikuvitusta uusien ideoiden tuottamiseen vanhan mukailemisen sijaan. Jokainen näistä kolmesta osa-alueesta voi ilmetä sekä ohjelmoijassa, ohjelmassa että yleisössä.

Coltonin mukaan luovalla järjestelmällä on oltava kaikki kolme osa-aluetta. Hän sanoo: ”ilman taitoa [taiteilija] ei voi tuottaa mitään, ilman arviointikykyä ei mitään arvokasta ja ilman mielikuvitusta hän voi korkeintaan tehdä mukaelmia muiden töistä”\* [11, s. 4]. Jos ohjelmalta puuttuu jokin osa-alue, se ei itsessään ole luova, mutta se voi yhdessä ihmisen kanssa muodostaa luovan järjestelmän. Jos ohjelmalta esimerkiksi puuttuu arviointikyky, mutta ihminen valitsee ohjelman luomuksista parhaat julkaitavaksi, ovat ihminen ja ohjelma yhdessä luovia.

Colton ym. [13] täydentävät luovaa kolmijalkaa lisäämällä taidon, arviointikyvyn ja mielikuvituksen lisäksi *oppimisen*, *tavoitteellisuuden*, *vastuun*, *uudistamisen*, *subjektii-visuuden* ja *pohdiskelun*. Heidän mukaansa järjestelmällä on oltava myös nämä ominaisuudet, jotta ihminen kokisi sen luovaksi.

---

\*”Without skill, they would never produce anything; without appreciation, they would never produce anything of value; and without imagination, at best they would only produce pastiches of other people’s work.” [11, s. 4]

### 2.1.4 FACE

Colton ym. [12] ehdottavat luovien tekojen arviointiin FACE-mallin, joka tarkastelee seuraavanlaisten olioiden luomista:

$f^g$	kehystiedon osa	$f^p$	menetelmä kehystiedon luomiseen
$a^g$	esteettinen metriikka	$a^p$	menetelmä esteettisen metriikan luomiseen
$c^g$	konsepti	$c^p$	menetelmä konseptin luomiseen
$e^g$	konseptin esitys	$e^p$	menetelmä konseptin esityksen luomiseen

Mallissa merkitään suuraakkosilla *luovia tekoja*, jotka tuottavat edellä mainittuja olioita. Yläindeksi  $^g$  viittaa olioon ja  $^p$  olion luomiseen käytettyyn menetelmään. Esimerkiksi  $f^p$  on siis menetelmä kehystiedon luomiseen ja  $F^p$  luova teko, jossa tällainen menetelmä luodaan. Colton ym. [13] ovat myöhemmin täydentäneet tätä merkintää merkitsemällä kolmannen asteen luovia tekoja yläindeksillä  $^m$  ( $f^m$  on siis menetelmä kehystiedon luomismenetelmän luomiseen) ja ihmisen tekemää luovaa tekoa yläviivalla – esimerkiksi  $\overline{F^p}$  – jolloin pelkkä  $F^p$  olisi koneen tekemä luova teko.

Osa FACE-mallissa käytetyistä käsitteistä tarvitsee selitystä. Colton ym. [12] antavat termeille seuraavat määritelmät: *Kehystieto*  $f^g$  tarkoittaa varsinaisen teoksen ulkopuolista tietoa, kuten tekijän nimen ja taustan, julkaisutavan ja teoksen historian. *Konsepti*  $c^g$  on tietokoneohjelma, joka kuvaa syötteen  $i$  teokseksi  $o$ . *Konseptin esitys* on pari  $e^g = (i, o)$ . *Esteettinen metriikka* on funktio  $a^g : (c^g, e^g) \rightarrow [0, \infty)$ , joka mittaa konseptin esityksen esteettistä arvoa.

FACE-malli täydentää Coltonin luovaa kolmijalkaa, sillä luovien tekojen avulla voi kuvata, missä osaa luovaa järjestelmää luovuus sijaitsee. Itse teoksen tuottaminen on vain kapea osa luovuutta: menetelmät teoksen tuottamiseen ja arviointiin ovat yhtä lailla tärkeitä, ja niiden tuottaminen on myös luovaa toimintaa. FACE-mallilla voi selvittää, mitkä luovat teot ovat ohjelman vastuulla ja mitkä ohjelmoijan, käyttäjän ja muiden luovan järjestelmän osien.

Colton ym. [13] ovat kehittäneet FACE-mallin pohjalta menetelmän, jossa luovat teot voidaan järjestää kaavioiksi, joissa ne on kategorisoitu luovan järjestelmän osan, ajankohdan ja riippuvuussuhteiden mukaan. Esimerkki tällaisesta kaaviosta on kuvassa 3.2. Kaavioissa luovia tekoja merkitään kulmasuluilla  $\langle \rangle$  ja ei-luovia tekoja hakasuluilla  $[]$ . Ei-luovia tekoja ovat esimerkiksi teoksen valitseminen metriikan avulla  $S(a^g(e^g))$  ja metriikan muuntaminen ohjelmakoodiksi  $T(a^g)$ . Toistettuja luovia tekoja esitetään tähdellä  $*$ .

FACE-mallissa vaikuttaa olevan ristiriita: konseptin esitys  $e_g = (i, c_g(i))$  on määritelty vain syöte-tulos-parina, joka syntyy, kun konsepti suoritetaan. Luova teko  $E^g$  on siis ennaltamääritelty ja tarkoittaa aina konseptin suorittamista. Luova teko  $E^p$  olisi merkityksetön (ja siten myös  $E^m$ ), sillä on vain yksi tapa suorittaa konsepti. Konsepti on erityisesti määritelty tietokoneohjelmaksi, ja on vaikea kuvitella, että olisi useita eri tapoja suorittaa sama ohjelma samalla syötteellä. Colton ym. [13] kuitenkin puhuvat konseptin esityksen luomisesta kuin kyseessä voisi olla monimutkaisempikin luova teko. He antavat esimerkkinä luovasta teosta  $\overline{E^p}$  jälkikäsitteijän, joka muuntaa jotenkin konseptin tuottamaa teosta [13, kuva 2, ID 1.1.2]. Ilmeisesti konseptin esitys tulisikin paremmin määritellä  $e^g = (i, e^p(c^g(i)))$ . Toisaalta he myös käyttävät merkintää  $\overline{C^m}, \overline{E^m}$  merkitsemään luovaa tekoa, jossa ihminen ensin suunnittelee menetelmän konseptien luomiseen ( $\overline{C^m}$ ) ja sitten toteuttaa sen ( $\overline{E^m}$ ) [13, kuva 2, ID 5.1.2], eli  $E^m$  merkitsisi vaihetta, jossa ihminen toteuttaa jo suunnitellun menetelmän koodina. Kuitenkin esteettisen metriikan toteuttamista koodina he merkitsevät  $T(a_g)$  [13, kuva 2, ID 5.4.2], missä  $T$ -funktio on syntynyt luovassa teossa  $E^p$ . Ilmeisesti  $E^p$  ja  $E^m$  tuottavat siis muunnoksia (joita merkitään joskus  $T$  ja joskus  $e^p/e^m$ ), joilla algoritmeja, kuten konsepteja ja esteettisyysmetriikoita, muutetaan muodosta toiseen. Valitettavasti Colton ym. eivät anna näille teoille tarkkoja määritelmiä.

### 2.1.5 IDEA

FACE-mallin mukaisten luovien tekojen arviointiin Colton ym. [12] ehdottavat IDEAMallia ("Iterative Development Execution Appreciation"). Mallissa kuvataan luovien ohjelmien kehitysprosessia, jossa luovan ohjelman tekemiä tekoja arvioidaan kehityksen aikana sekä suhteessa "ideaaliseen kontekstiin" että "ideaaliseen yleisöön".

Konteksti koostuu kaikista maailman luovista teoista  $\alpha$ , ohjelman kehityksessä käytyistä ja ohjelmalle syötteenä annetuista teoista  $\beta$  sekä ohjelman tekemistä luovista teoista  $\kappa$ . Coltonin ym. uskovat, että luovilla teoilla on eniten merkitystä, jos ne ovat tarpeeksi erilaisia kuin ennestään tunnetut  $\alpha$ :n luovat teot, mutta tarpeeksi samanlaisia, että niitä voi arvioida osana samaa kontekstia.

Luovien tekojen vaikutusta ideaaliseen yleisöön  $i$  he arvioivat kahden muuttujan,  $wb_i \in [-1, 1]$  ("yleisön hyvinvoinnin") ja  $ce_i \in [0, 1]$  ("kognitiivisen vaivan") avulla. Näistä  $wb_i$  mittaa yleisön mielipidettä luovasta teosta ja  $ce_i$  yleisön teon arviointiin käyttämää vaivaa.

### 2.1.6 ”Lovelace-testi”

Bringsjord ym. [6] ehdottavat Turingin testin korvaajaksi ”Lovelace-testiä”, joka mittaa, ymmärtääkö ohjelman tekijä itse, mitä hänen ohjelmansa tekee. Jos ohjelmoija ei ymmärrä ohjelmansa toimintaa, on ohjelma testin mukaan luova. Formaalisti testi on määritelty seuraavasti:

Ihmisen  $H$  kehittämä agentti  $A$  läpäisee Lovelace-testin, jos

1.  $A$  tuottaa artefaktin  $o$ ,
2. tämä prosessi on deterministinen ja toistettava ja
3.  $H$  tai muu, jolla on  $H$ :n tiedot ja resurssit, ei voi selittää, miksi  $A$  tuotti  $o$ :n.

Bringsjord ym. itse uskovat, ettei mikään tietokoneohjelma voi läpäistä testiä, sillä niiden toiminta on täysin selitettävää. He toteavat, että esimerkiksi keinotekoiset neuroverkot voi aina esittää kaavana, ja ihminen voi ”manipuloida näitä symboleita” ja ennustaa, miten ohjelma toimii.

Testi on nimetty Charles Babbagen analyyttistä konetta tutkineen Ada Lovelacen mukaan. Hänen kerrotaan väittäneen, että tietokone ei voi luoda mitään uutta, sillä se tekee vain sen mitä sen käsketään tehdä. Bringsjord ym. siteeraa Turingia [36], joka siteeraa Hartreetä [23], joka siteeraa itse Lovelacea [29]. Historiallisen tarkkuuden vuoksi on todettava, että Lovelace itse asiassa ei väitä, ettei tietokone voisi luoda mitään. Kyseinen sitaatti on irrotettu asiayhteydestään, eli kommentista [29] M. Menabrean tutkielmaan Babbagen analyyttisestä koneesta. Lovelace toppuuttelee Menabreaa, joka pohtii, että analyyttiselle koneelle voisi antaa älykkyyttä ja autonomiaa koodaamalla älykkyyteen vaadittavat ominaisuudet koneelle annettaviin käskyihin. Lovelace toteaa, että kone ei yritä teeskennellä tuottavansa mitään itsenäisesti, vaan noudattaa vain sille annettuja käskyjä. Hänen mukaansa on varottava koneen kykyjen liioittelusta, sillä ihmisillä on taipumus yliarvioida uusien teknologioiden ominaisuuksia. Lovelace sanoo, että itsenäinen älykkyys on pelkkää spekulatiota. Hän ei siis väitä sitä mahdottomaksi, vaan ainoastaan muistuttaa, ettei se välttämättä ole mahdollista analyyttisen koneen kohdalla. Lovelace sanoo aiemmin tekstissään, että analyttinen kone voisi mahdollisesti tuottaa ”taidokkaita ja tieteellisiä musiikkikappaleita minkätahansa asteen monimutkaisuudella tai laajuudella”, eli hän puolusti tietokoneen kykyä jonkinasteiseen luovaan työhön.

### 2.1.7 Pelkkä generointi ja luovuus

Ventura [37] määrittelee luovuuden skaalana, joka alkaa *pelkästä generoinnista* ja asteittain etenee kohti *luomista*. Luovaksi hän määrittelee ohjelman, joka voi *tarkoitukSELLISESTI* tuottaa *uusia* ja *arvokkaita* teoksia.

Luovuuden askeleet Venturan mukaan ovat:

1. *Satunnaisuus*. Täysin satunnainen ohjelma, joka tuottaa kohinaa (satunnaisia pikseleitä, satunnaisia sanoja, satunnaisia nuotteja jne.).
2. *Plagiarisointi*. Ohjelma palauttaa valmiin teoksen lähdeaineistostaan.
3. *Muistaminen*. Plagiarisointi epätäydellisellä muistilla, joka korruptoi lähdeaineiston teoksia.
4. *Yleistäminen*. Ohjelma luo yleistyksiä (samaaan alaan kuuluvia uusia teoksia) lähdeaineistonsa teoksista.
5. *Valikointi*. Ohjelma valikoi luomistaan teoksista parhaat metriikan avulla.
6. *Tiedon istuttaminen*. Ohjelma käyttää lähdeaineiston lisäksi hyödykseen tietopankkia.
7. *Luominen*. Ohjelma arvioi teosten tuottamia kokemuksia.

Vaikka plagiarisointi tuottaa arvokkaita teoksia, sitä ei voida pitää luovana uutuuden puutteen vuoksi. Yleistäminen tuottaa uusia teoksia ja valikointi arvokkaita teoksia. Koska kielimalleissa ei ole erillistä suodatusvaihetta, voi niitä nähdäkseni pitää yleistävänä, mutta ei suodattavana. Rajanveto on kuitenkin hankalaa: Ventura [37] jakaa suodattamisen jälkikäteiseen *post hoc* -suodattamiseen ja generoinnin aikaiseen *in situ* -suodattamiseen. Jos kielimalli määrittää todennäköisyydet riittävän monimutkaisella tavalla, voinee tätä pitää yksittäisiin sanoihin kohdistuvana *in situ* -suodattamisena. Ventura ei ota kantaa siihen, kumpi suodattamistapa on luovempi.

Hyppy yleistämisestä valikointiin on merkittävä ja haastava, eikä se välttämättä ole edes mahdollinen kaikille yleistäville ohjelmille [20]. Jos ohjelma tuottaa liian paljon huonoja tuloksia ja liian harvoin hyviä tuloksia, valikointi saattaa karsia pois lähes kaiken, jolloin ohjelmalla ei voi tuottaa mitään.

## 2.2 Kielimallit

Ennen kielimallien määrittelyä määrittelen peruskäsitteitä. *Teksti* on jono saneita järjestyksessä. *Sane* (engl. [word] token) on sanan ilmentymä [9]. Yleiskielessä sanat ”sana” ja ”sane” ovat joskus synonyymisiä, mutta kielimallien yhteydessä niillä on ero. Esimerkiksi lauseessa ”Minä teen mitä teen” on neljä sanetta ( $\{\text{Minä, teen, mitä, teen}\}$ ) ja kolme sanaa ( $\{\text{Minä, teen, mitä}\}$ ).

Vaikka puhun tässä ”sanoista”, kielimalleilla ei välttämättä generoida kokonaisia sanoja, vaan perusyksikkö voi olla myös sanaa pienempi merkkijono. Musiikin generoinnin yhteydessä sane voi tarkoittaa nuottia, sointumerkkiä, tahtiviivaa tai muuta vastaavaa musikaalisen notaation merkintää.

*Kielimalli* on tilastollinen malli, joka pyrkii laskemaan todennäköisyyden  $P(W_{1,t} = w_{1,t})$ , missä  $W_{1,t} = W_1, W_2, \dots, W_t$  on jono satunnaismuuttujia, joiden arvoina voi olla mitä tahansa kielen sanoja ja  $w_{1,t} = w_1, w_2, \dots, w_t$  ovat tietty  $t$ :n pituinen jono saneita. Kielimalleja voi käyttää esimerkiksi puheentunnistuksessa päättelemään, mikä mahdollisista tulkinnoista annetulle äänelle on todennäköisin. [9]

Käytetään todennäköisyydestä  $P(W_{1,t} = w_{1,t})$  lyhennettä  $P(w_{1,t})$ . Todennäköisyys voidaan pilkkoa seuraavalla tavalla:

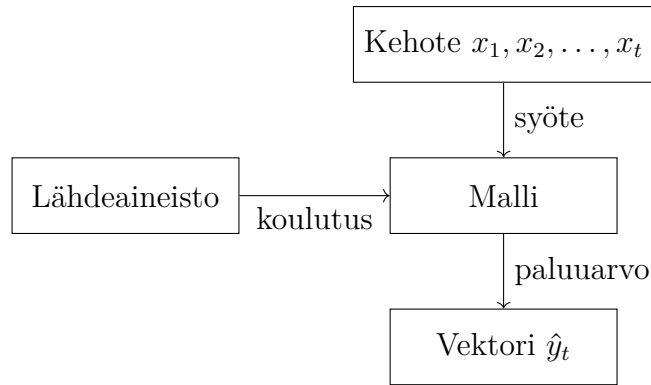
$$P(w_{1,t}) = P(w_1) \prod_{t'=2}^t P(w_{t'} | w_{1,t'-1}).$$

Monet kielimallit laskevat yllä olevan kaavan mukaisesti seuraavan saneen todennäköisyyden edeltävien saneiden perusteella sen sijaan, että ne laskisivat koko lauseen todennäköisyyden suoraan.

*n-gram-malliksi* sanotaan mallia, joka ottaa huomioon vain  $n - 1$  edellistä sanetta [9], eli sisältää oletuksen

$$P(w_t | w_{1,t-1}) = P(w_t | w_{t-n+1,t-1})$$

*1-gram-malliksi* eli *unigram-malliksi* sanotaan mallia, jossa saneen todennäköisyys ei riipu muista saneista. *2-gram-malli* vastaa yksinkertaista Markovin ketjua, jossa vain edellinen sane määrittää seuraavan saneen todennäköisyyden. [9]



**Kuva 2.1:** Kielimallin toiminta.

### 2.2.1 Neuroverkkokielimallit

Tässä tutkielmassa keskityn monimutkaisiin kielimalleihin, kuten GPT-2-tyylisiin[31] transformer-kielimalleihin. Tällaiset mallit ennustavat edellisten saneiden perusteella, mikä on seuraava sane tekstissä. Toistamalla ennustamista ja lisäämällä aina uusi sane tekstin loppuun voidaan tuottaa pitkiä tekstejä. Kielimalleja koulutetaan suurilla aineistoilla peräkkäisiä saneita.

Graves [21] kuvaa tekstingenerointimenetelmää RNN-neuroverkoilla. Tutkielmassani mainitsemani kielimallit kuten GPT-2, GPT-3 ja osassa 4 esittelemäni kielimalli eivät ole tällaisia neuroverkkoja, mutta generointi toimii niillä samalla periaatteella.

Kielimallin syötteenä on *kehotteeksi* kutsuttu joukko saneita  $x_1, x_2, \dots, x_t$ . Mallin tulos taas on vektori  $\hat{y}_t$ , jonka komponentit  $\hat{y}_t^k$  vastaavat mallin sanaston sanoja  $k$ . Sanaston kokoa merkitään  $K$ :lla. Kielimallin syöte ja tulos on havainnollistettu kuvassa 2.1.

Todennäköisyys sille, että seuraava sane  $x_{t+1}$  on sana  $k$  saadaan softmax-kaavalla [21]:

$$\Pr(x_{t+1} = k | y_t) = \text{softmax}(\hat{y}_t^k) = \frac{e^{\hat{y}_t^k}}{\sum_{k'=1}^K e^{\hat{y}_t^{k'}}$$

Kielimallille voi kehotteen lisäksi antaa syötteenä erillisen kontekstivektorin, joka voi sisältää tietoa esimerkiksi siitä, millä tyylillä tekstiä tulee kirjoittaa. Tällöin kielimallia sanotaan *ehdollistetuksi kielimalliksi*. [18]

Niin kutsutulla *lämpötilalla* voi skaalata todennäköisyysjakaumaa kasvattamalla todennäköisyyseroja (eli tekemällä todennäköisistä saneista vieläkin todennäköisempiä, jolloin kielimalli on ”varmempi”) tai pienentämällä niitä (jolloin kielimalli tuottaa sa-

tunnaisempaa tekstiä) [18]. Lämpötilaa muutetaan jakamalla tulos  $T$ :llä ennen softmax-operaatiota:

$$\Pr(x_{t+1} = k|y_t) = \text{softmax}\left(\frac{\hat{y}_t^k}{T}\right) = \frac{e^{\frac{\hat{y}_t^k}{T}}}{\sum_{k'=1}^K e^{\frac{\hat{y}_t^{k'}}{T}}}$$

Kielimallilla tuotetaan tekstiä siten, että todennäköisyysjakaumasta arvotaan kehoitteeseen uusi sana. On mahdollista tuottaa tekstiä interaktiivisesti ihmisen tai toisen tietokoneohjelman kanssa siten, että kielimalli ja toinen osapuoli lisäävät vuorotellen tekstiä kehoitteen loppuun. Kerron esimerkkejä tästä seuraavassa luvussa.

### 2.2.2 Kielimallien käyttö taiteessa ja laskennallisessa luovuudessa

Kielimalleja on käytetty erilaisissa projekteissa, joista jotkut ovat laskennallisen luovuuden tutkijoiden tekemiä ja jotkut muiden alojen taiteilijoiden ja tutkijoiden tekemiä. Kaikki tässä esittelemistäni järjestelmistä eivät välttämättä ole laskennallisen luovuuden näkökulmasta luovia.

OpenAI-yhtiön Transformer-kielimalleja GPT-2 [31] ja GPT-3 [8] on käytetty runojen ja muiden kirjallisten teosten generoinnissa.\* Malleja on käytetty myös vuorovaikutteisissa järjestelmissä. Austin [3] esittelee GPT-2:een perustuvan hypertekstiteoksen, jossa Wikipedia-tyylisellä sivulla on loputtomasti generoituja artikkeleita, joiden välillä voi liikkua klikkaamalla mallin tuottamia linkkejä. AI Dungeon<sup>†</sup> taas on GPT-3:a käyttävä interaktiivinen tarinankerrontajärjestelmä, jossa käyttäjä voi ohjata tarinan kulkua syöttämällä tekstiseikkailu-tyylisiä komentoja kesken tekstin tuottamisen. OpenAI-yhtiö on esitellyt edellä mainittujen mallien lisäksi myös musiikkia tuottavan MuseNet-mallin.<sup>‡</sup>, joka osaa tuottaa moniäänisiä musiikkikappaleita.

Ammanabroly ym. [2] ovat vertailleet Markovin ketjua ja neuroverkkopohjaista kielimallia tekstiseikkailujen generoinnissa. He toteavat, että neuroverkkokielimalli tuottaa arvokkaampia ja koherentimpia seikkailuja, kun taas Markovin ketju tuottaa yllättäviä ja uusia seikkailuja.

---

\*Ks. esimerkiksi <https://www.gwern.net/GPT-2> ja <https://www.gwern.net/GPT-3> sekä <http://cs.uccs.edu/~jkalita/work/reu/REU2019/12Bena.pdf>

<sup>†</sup><https://play.aidungeon.io>

<sup>‡</sup><https://openai.com/blog/musenet/>

Muun muassa Tikhonov ja Yamshchikov [35], Peterson [30] sekä Zhang ja Zhang [40] ovat käyttäneet LSTM-verkkoja runouden generointiin. Eck ja Schmidhuber [17] esittävät, että LSTM-kielimallit voivat tuottaa rakenteellisesti koherentteja ja miellyttäviä musiikkikappaleita. Choi ym. [10] ovat käyttäneet LSTM-verkkoja sointukulkujen ja rumpuraitojen generointiin.

GPT-mallien lisäksi myös muita Transformer-verkkoja on käytetty taiteen tuottamiseen. Agafonova ym. [1] kouluttivat tällaisen mallin tuottamaan ”vainoharhaista” tekstiä, ja pohtivat luovien ohjelmien suhdetta pelkoon ja minäkäsitteeseen. Huang ym. [25] käyttävät musiikilla koulutettua Transformer-mallia sekä melodioiden että säestysten tuottamiseen. Wu ym. [39] käyttävät Transformer-XL-mallia [16] musiikin tuottamiseen.

### 2.2.3 Aiempia arvioita kielimallien luovuudesta

Huolimatta kielimallien yleisyydestä taideprojekteissa niiden luovuutta on arvioitu suhteellisen vähän.

Berns ym. [4] toteavat, että syväoppivat neuroverkot voivat olla luovia ainoastaan ”minimaalisessa mielessä”. Heidän mukaansa neuroverkko täyttää luovuuden määritelmän, sillä se osaa arvioida teostensa hyvyttä hukkafunktion avulla, mutta vain minimaalisella tavalla, sillä arvokkaampien teosten tuottaminen tarkoittaa samalla niiden uutuu- den pienenemistä, sillä hukkafunktio mittaa teosten samankaltaisuutta lähdeaineiston kanssa. Broad ym. [7] toteavat, että neuroverkkomallit tuottavat usein tyypillisiä ja arvokkaita teoksia. Koska mallin koulutuksen tavoite on saada mallin todennäköisyysjakauma muistuttamaan täysin lähdeaineiston jakaumaa, teoksia ei kuitenkaan voi pitää täysin uusina. He listaavat useita menetelmiä, joilla neuroverkkojen tuottaman todennäköisyysjakauman voi yrittää saada poikkeamaan lähdeaineiston todennäköisyysjakaumasta. Näitä ovat ekstrapoloiminen, lähdeaineiston annotoiminen tarkoituksellisesti väärin, kouluttaminen ilman lähdeaineistoa, mallin hienosäätäminen erilaisella datalla kuin alkuperäinen aineisto, mallin hienosäätäminen pois päin lähdeaineistosta, useiden mallien yhdisteleminen ja ketjuttaminen, useiden mallien tai niiden todennäköisyysjakaumien laskeminen yhteen ja mallin muuttaminen satunnaisesti.

”Musta laatikko” -ominaisuuden suhdetta luovuuteen on pohdittu aiemminkin. Bringsjord ym. esittävät luovuuden mittaamiseen ”Lovelace-testiä” [6] (ks. luku 2.1.6), jonka mukaan ohjelma on luova, jos sen tekijä ei ymmärrä, miksi se on tuottanut luoman-

sa teokset. Oletuksena on, että ohjelma on täysin deterministinen ja toistettava. He viittaavat siis ohjelman jälkikäteiseen *selitettävyyteen*, eivät *ennustettavuuteen*. En pidä tätä testiä lähtökohdiltaan täysin vakavasti otettavana. Tietynlainen selitettävyyden on osa luovuutta: ihmistäiteilijät käyttävät usein paljon aikaa teostensa perusteluun ja luomisprosessinsa avaamiseen. Testi ei myöskään ota kantaa teosten arvoon tai uutuuteen.

Bringsjord ym. [6] uskovat, että mikään olemassaoleva ohjelma ei läpäise heidän testiään. Esimerkkinä he tutkivat muun muassa tarinageneraattoria ja keinotekoisia neuroverkkoja. He toteavat, että ihminen on luonut nämä järjestelmät, ja voi käydä läpi kaikki ohjelman tekemät laskutoimitukset ja päästä samaan lopputulokseen. Tämä on kehäpäätelmä, sillä he olettivat järjestelmän olevan deterministinen ja toistettava. Ajatteluketjussa on myös useita muita ongelmia. Ensinnäkin ei ole merkityksellistä, suorittaako ohjelmaa ihminen vai kone. Ihmisenkin suorittamassa ohjelmassa tapahtuvat luovat teot ovat ohjelman tekemiä. Pelkkien laskutoimitusten tekeminen ei myöskään tarkoita, että ihminen ymmärtäisi ohjelmaa kokonaisuutena tai sen toimintaperiaatetta korkeammalla abstraktiotasolla. On toki vaikea määritellä, kuinka korkealla tasolla ohjelmaa on ymmärrettävä, jotta sitä voi kutsua täysin selitettäväksi. Tämän vuoksi on mielestäni järkevää tutkia B-ennustamattomuutta (ks. luku 3.1.2) selitettävyyden sijaan. B-ennustamattomuuden määritelmästä seuraa, ettei ennustavalla ihmisellä ole käytössään kaikkia tietokoneen tekemiä laskutoimituksia.

Puutteistaan huolimatta ajatusleikkinä Lovelace-testi on kuitenkin mielenkiintoinen, sillä ohjelmien selitettävyyden saattaa vaikuttaa ihmisten mielikuvaan ohjelman luovudesta. Käsittelen hieman samanlaista argumenttia deterministisyydestä luvussa 3.2.3.

# 3 Kielimallin luovuuden kriteerit

Tutkielman tavoitteena on muodostaa riittävä kriteeristö kielimallin luovuudelle. Ajatuksena on, että kriteerejä voi käyttää kielimallien luovuuden arviointiin: kriteerit täyttävä kielimalli tekee varmasti tiettyjä luovia tekoja.

Lähtökohtana kriteeristölle on luovuuden standardimääritelmä [34]: luovan järjestelmän on tuotettava arvokkaita ja uusia teoksia. Nämä kaksi ovat välttämättömiä kriteerejä: jos teokset eivät ole arvokkaita ja uusia, järjestelmä ei ole luova. Ne eivät kuitenkaan yksinään riitä, sillä luovia tekoja on voitava kohdentaa juuri kielimalliin.

Kirjallisuudessa luovuuden katsotaan usein voivan ilmetä tekijässä, teoksessa, luomisprosessissa ja yleisössä [27]. Yhdessä nämä eri osat muodostavat luovan järjestelmän. Kielimallin tapauksessa järjestelmässä ovat ainakin seuraavat osat, joista mikä vain voi olla luovuuden lähde:

1. Kielimalli itse, mukaan lukien koulutusalgorithmi, neuroverkon painot ja verkkoa ajava generointiohjelma.
2. Kielimallin koulutukseen käytetty lähdeaineisto ja sen tekijät ja valikoijat sekä kielimallille annettu kehote (jäljempänä ”L-osa”).
3. Kielimallin koulutusalgorithmien tekijä ja kielimallin kouluttaja (jäljempänä ”K-osa”).
4. Yleisö, joka kuluttaa kielimallin tuottaman teoksen (jäljempänä ”Y-osa”).

Luovassa järjestelmässä on myös muita osia, kuten generointiin käytetty tietokone. Oletan näiden osien vaikutuksen luovuuteen olevan pieni. Lisäksi kielimalli on usein vain pieni osa laajempaa luovaa järjestelmää. Esimerkiksi musiikin generoinnin tapauksessa tärkeä osa on ihmissoittajalla tai syntetisaattorilla, joka tulkitsee kappaleen yleisölle. Tarkoitukseni ei ole tutkia näiden osien luovuutta, vaan rajoitun pelkästään kielimalliin ja sen tuottamaan tulokseen (esimerkiksi nuotteihin äänitteiden sijasta). Järjestelmän muiden osien luovuutta on arvioitava erikseen.

Kohdentaakseni luovat teot kielimalliin muiden osien sijasta otan arvokkuuden ja uutuuden lisäksi kolmanneksi kriteeriksi mallin *ennustamattomuuden*. Kaikki kolme kriteeriä ovat siis:

1. **Teosten arvokkuus.** Välttämätön kriteeri luovuudelle.
2. **Teosten uutuus.** Välttämätön kriteeri luovuudelle. Käytetään kohdentamaan luovuus pois L-osasta.
3. **Kielimallin ennustamattomuus.** Käytetään kohdentamaan luovuus pois K-osasta.

On huomattava, että kriteeristöstä puuttuu kriteeri, joka kohdentaa luovuuden pois Y-osasta. En siis tarkastele sellaista tilannetta, jossa kriteerit täyttyvät, mutta kaikki luovuus syntyy yleisön tulkinnasta. Tällaisessa tilanteessa teokset ovat sekä arvokkaita että uusia, mutta eivät siltikään luovia. On vaikea määritellä, mikä mitattava ominaisuus teoksissa tällöin tekee niistä ei-luovia. Tämä johtuu siitä, että arvokkuus ja uutuus on määritelty suhteessa johonkin yleisöön: teokset ovat arvokkaita ja uusia *jollekulle*. Ongelma voidaan osittain ratkaista uutuuden kohdalla tarkastelemalla vain P-luovuutta, eli määrittelemällä ”uusi” tarkoittamaan uutta ohjelmalle, ei yleisölle. Arvon suhteen ongelma on monimutkaisempi.

Eräs mahdollinen Y-osan huomioonottava kriteeri olisi *tavoitteellisuus*, jota Ventura [37] pitää edellytyksenä luovuudelle uutuuden ja arvon lisäksi. Tavoitteellisuus liittyy käsitteenä *teostietoisuuteen* eli luovan ohjelman kykyyn tulkita itse omia teoksiaan. Ajatus näissä kriteereissä on, että teosten arvo ei synny vain yleisön tulkinnasta, vaan niillä on arvoa suhteessa ohjelman tavoitteisiin. Tietoisuus mahdollistaa sen, että ohjelma voi antaa tulkintoja omille teoksilleen, mikä siirtää vastuuta tulkinnasta pois yleisöltä. Näiden käsitteiden määrittely on huomattavan hankalaa. Olen käsitteellyt niitä kriteeristöstä erikseen luvussa 3.3.

## 3.1 Kriteerien yksityiskohtainen kuvaus

Määrittelen tässä luvussa käyttämäni kriteerit tarkemmin ja pohdin niiden mielekkyyttä.

### 3.1.1 Teosten arvokkuus ja uutuus

Ensimmäiset kaksi kriteeriä kielimallin luovuudelle ovat, että sen tuottamat teokset ovat *arvokkaita* (”value”) ja *uusia* (”novelty”). Nämä ominaisuudet ovat edellytys luovuudelle [34][37], ja ilman niitä kielimallia ei voisi pitää luovana. Koska arvokkuus ja

uutuus eivät ole binäärisiä totuusarvoja, määrittelen, että riittävän moni teos on riittävän arvokas ja riittävän uusi. Vaikka nämä kaksi kriteeriä ovat välttämättömiä, ne ovat myös monin tavoin ongelmallisia.

Sekä arvokkuus että uutuus olettavat yleisön, jolle teokset ovat arvokkaita ja uusia. Yleisön määrittelyminen on kuitenkin hankalaa. Teoriassa kaikki teoksia näkevät ihmiset ovat osa yleisöä, jopa sellaiset, jotka näkevät teokset vasta pitkän ajan päästä. On vaikea ottaa näitä huomioon arvoa ja uutuutta arvioitaessa. Toisaalta vain pienen testiyleisön ottaminen huomioon saattaa vinouttaa dataa. Arvioidessani esimerkkijärjestelmää osassa 4 oletan uutuuden tarkoittavan, että teos on kielimallille uusi (Bodeinin [5] termeillä mitaan siis P-luovuutta), jolloin yleisöä ei tarvitse ottaa huomioon. Arvoa mitaan vähäisten resurssieni vuoksi pienellä testiyleisöllä.

## Uutuuden arviointi

Uutuus on käsitteenä huonosti määritelty. Osan 4 esimerkikielimallin arvoinnissa käyttämäni määritelmä on, että teos on uusi, jos se ei suoraan kopioi mitään tiettyjä aiempia teoksia. Tässä määritelmässä on kaksi ongelmaa. Ensinnäkin aiempia teoksia kopioivakin teos voi olla luova, jos aiempia teoksia on yhdistelty luovalla tavalla. Toisekseen teos voi olla hyvin kliseinen ja epäkiinnostava, vaikka se ei kirjaimellisesti kopioisi mitään aiempia teoksia. Määritelmän etu kuitenkin on, että se on helposti mitattavissa koneellisesti. Sen matemaattinen kuvaus musiikkiin sovellettuna on annettu luvussa 4.5.1.

Avaan yllä esitetyn uutuuden määritelmän seurauksia esimerkillä. Tämä ei ole realistinen esimerkki siitä, mitä kielimalli voi tehdä, vaan tarkoitukseni on kuvata nimenomaan määritelmän ominaisuuksia. Oletetaan, että luova järjestelmä oppii tuottamaan uusia teoksia siten, että se valitsee lähdeaineistostaan teoksen ja muuttaa kaikki sen pienet kirjaimet isoiksi ja isot kirjaimet pieniksi. Tällöin järjestelmän luomat teokset ovat siinä mielessä uusia, etteivät ne sisällä mitään minkään kokoisia pätkiä lähdeaineiston tekstiä. Oletetaan myös, että näin syntynyttä teosta pidetään arvokkaana. Onko järjestelmä luova?

Esimerkki voidaan analysoida niin, että järjestelmä on todellakin luonut jotain: uuden konseptin eli kirjainkoon muuttamisen. Koska tarkastelemme vain P-luovuutta, voi tämä konsepti olla täysin uusi, jos vastaavanlaisia muunnoksia ei esiinny lähdeaineistossa.

Vaikka ihmisen näkökulmasta ohjelma ei teekään mitään kovin vakuuttavaa, on se silti tehnyt yksinkertaisen luovan teon.

### **Konseptien arviointi**

Eräs tapa arvioida uutuutta on tutkia kielimallin käyttämän konseptin saamia vaikutteita. Colton ym. [12] määrittelevät konseptin tietokoneohjelmaksi, joka kuvaa syötteen  $i$  teokseksi  $o$ . Minusta tämä määritelmä on rajoittava ja liian konkreettinen: on epäselvää, mitä ”tietokoneohjelma” kielimallin ja sen koulutusalgoritmin yhteydessä tarkoittaa. Siksi tässä luvussa viitataan konseptilla yleisemmin mihin tahansa menetelmään, jolla voi tuottaa teoksia. Koska kielimalli on koulutettu jollain lähdeaineistolla, perustuu konsepti väistämättä jotenkin aineistoon, vaikka se ei olisi suoraan sama kuin mikään aineiston tekijöiden käyttämä konsepti.

Luettelen alla viisi eri tapaa, joilla kielimallin konsepti voi pohjautua lähdeaineistoon. Näistä tavoista ensimmäinen ei määritelmällisesti ole luova.

**Kopioiminen.** Yksinkertaisimmassa tilanteessa mallin jakauma on diskreetti ja vastaa täysin lähdeaineistoa, eli ylioppinut malli siis toistaa täysin lähdeaineiston teoksia. Tällöin teokset eivät ole millään tavalla uusia, jolloin malli ei ole tehnyt luovaa tekoa, jota tutkin tässä luvussa. Venturan asteikolla tämä vastaa tasoa 2 eli plagiarisointia [37].

Varsinaisia luovia tapoja on neljä:

**Muuttaminen.** Malli ottaa yhden lähdeaineiston teoksen ja tuottaa muunnelman siitä. Alkuperäinen teos on selkeästi tunnistettavissa.

**Yhdisteleminen.** Malli ottaa useita lähdeaineiston teoksia ja yhdistää ne uudella tavalla. Alkuperäiset teokset ovat selkeästi tunnistettavissa.

**Interpoloiminen.** Malli yhdistelee useita lähdeaineiston ideoita uudella tavalla siten, että alkuperäisiä teoksia ei voi enää tarkasti tunnistaa.

**Ekstrapoloiminen.** Mallin tuottamissa teoksissa on ideoita, jotka eivät ole yhdistettävissä mihinkään lähdeaineiston teoksiin.

Näistä muuttaminen ja yhdisteleminen vastaavat Bodenin [5] (ks. luku 2.1.1) määrittelemää yhdistelevää luovuutta, kun taas ekstrapolointi vastaa tutkivaa luovuutta. Interpoloiminen voi olla joko tutkivaa tai yhdistelevää.

On huomattava, että sama teos voi hyödyntää lähdeaineistoa eri tavoin eri tasoilla. Jos malli tuottaman teoksen merkitys on täsmälleen sama kuin jonkun lähdeaineiston teoksen, mutta se on esitetty eri sanamuodoin, saattaa teos olla merkityksen tasolla kopioimista mutta kielen tasolla yhdistelemistä. Toisaalta, jos ohjelma tuottaa perusidealtaan täysin uuden tekstin, mutta jokainen lause on kopioitu eri lähdeaineiston teoksista, voi teos olla merkityksen tasolla ekstrapoloiva mutta kielen tasolla yhdistelevä.

On helppo arvioida peräkkäisten sanojen perusteella, minkälaisia vaikutteita kielimallin konsepti sisältää. Ylioppinut kielimalli kopioi sanasta sanaan pitkiä tekstipätkiä, jolloin se joko muuttaa tai yhdistelee lähdeaineiston teoksia. On mahdollista koneellisesti jäljittää kielimallin käyttämät lauseet tai sanavalinnat niihin lähdeaineiston teoksiin, josta ne on kopioitu. Merkityksen tasolla esiintyvä lähdeaineiston jäljittelemineen on paljon vaikeampaa mitata automatisoidusti, vaikka ihmiselle merkitys voi olla yhtä tärkeä ellei jopa tärkeämpi kuin käytetyt sanavalinnat.

Pinnallisella sanojen tasolla muuttaminen ja yhdisteleminen on mahdollista selvästi tunnistaa, jolloin on mahdollista arvioida, kuinka luova käytetty konsepti on. Tarkastellaan esimerkkinä tilannetta, jossa malli toistaa lähdeaineiston teoksen siten, että yksi sana on korvattu toisella sanalla. Jos tämä korvaus on hyvin yleinen, esimerkiksi sana ”ja” on korvattu sanalla ”sekä”, ei tekoa voi pitää kovin luovana. Jos taas korvaus on täysin ennennäkemätön ja yleisön mielestä arvokas, on kyseessä selvästi luova teko.

Merkityksen tasolla on käytettävä ihmisarvioijaa tulkitsemaan teoksen saamia vaikutteita. Tällöin on oltava tarkka siitä, tutkitaanko P-luovuutta vai H-luovuutta (ks. luku 2.1.1). Jos lähdeaineisto on erittäin suuri, mikä on yleistä suurten kielimallien kohdalla, ihminen ei välttämättä voi käydä sitä kokonaan läpi. Tällöin hän voi ainoastaan tutkia H-luovuutta. Tämä rajoittaa merkittävästi P-luovuuden arviointia. Eräs mahdollinen ratkaisu voisi olla arvioinnin joukkoistaminen suurelle määrälle ihmisiä.

Konseptien arviointi vaatii, että luova järjestelmä on selitettävä ja arvioija osaa päätellä, millainen järjestelmän käyttämä konsepti on. Mallin ennustamattomuus ei välttämättä tarkoita, että malli olisi selittämätön. Myös ennustamaton malli voi käyttää selvästi tunnistettavia konsepteja. Uskon kuitenkin, että käytännössä suurin osa isoista kielimalleista on myös selittämättömiä, jolloin teoksista ei voi enää suoraan nähdä,

miten ne tarkalleen hyödyntävät lähdeaineistoaan. Tällöin ei ole enää mahdollista arvioida, kuinka uusia käytetyt konseptit ovat.

Kun arvioija on tunnistanut ohjelman käyttämät konseptit, niitä voi arvioida esimerkiksi IDEA-mallin (ks. luku 2.1.5) tai Ritchien menetelmän (ks. luku 2.1.2) avulla.

### 3.1.2 Kielimallien ennustamattomuus

Kolmas kriteerini on, että kielimallit ovat *ennustamattomia*, eli ihminen ei voi ennustaa niiden suorittamatta itse mallia. Ennustamattomuus ulottuu vain järjestelmän deterministisiin osiin. Määrittelen, että reilu satunnaisluvun arpominen on ennustettava tapahtuma, sillä ihminen voi yhä ennustaa satunnaisluvun jakauman.

Boden [5] jakaa ennustamattomuuden absoluuttiseen ennustamattomuuteen eli *A-ennustamattomuuteen* ("A-unpredictability"), relatiiviseen ennustamattomuuteen eli *R-ennustamattomuuteen* ("R-unpredictability") ja perhosennustamattomuuteen eli *B-ennustamattomuuteen* ("B-unpredictability"). A-ennustamattomuus tarkoittaa, että prosessia ei voi ennustaa mitenkään. R-ennustamattomuus tarkoittaa, että ihminen ei voi ennustaa prosessia. B-ennustamattomuus taas tarkoittaa, että ainoa tapa ennustaa prosessia on suorittaa se. Valitsemani kriteeri on siis, että kielimallit ovat B-ennustamattomia.

Ennustettavuuden käsitteeseen liittyy läheinen selitettävyyden käsite. Ennustettavuus liittyy ohjelman etukäteiseen arviointiin, kun taas selitettävyyys jälkikäteiseen. Selitettävyyys on mahdollista toteuttaa ilman ennustettavuutta esimerkiksi siten, että ohjelma kertoo teoksen luotuaan, miksi loi sen. Toisin kuin ennustamattomuus, selittämättömyys ei ole käyttämäni kriteeri.

On huomattava, että jotkut kielimallit ovat ennustettavia. Esimerkiksi Markovin ketju on kielimalli, joka valitsee todennäköisyysjakauman vain kehotteen viimeisen tai viimeisten saneiden perusteella. Sekä jakauman haku tietorakenteesta että itse arvonta ovat yksinkertaisia tapahtumia, jotka ihminen voi ymmärtää etukäteen. Markovin ketjut siis eivät ole ennustamattomia.

Tarkoitukseni ei ole väittää, että ennustamattomuus olisi edellytys luovuudelle. Kriteeristöni on tarkoitus olla riittävä, ei välttämätön. Enustettavien ja luovien kielimallien tarkastelu on tämän työn rajauksen ulkopuolella.

## 3.2 Luovat teot

Tässä luvussa tarkastelen kriteerit täyttävän kielimallin tekemiä luovia tekoja. Luovalla teolla tarkoitan sellaista luovan järjestelmän tekoa, jonka lopputuloksena on syntynyt jotain arvokasta ja uutta.

Käytän kriteerejä kohdentaakseni luovat teot luovan järjestelmän sisällä kielimalliin L-, K-osien sijasta. Päättelyketjuni on seuraava:

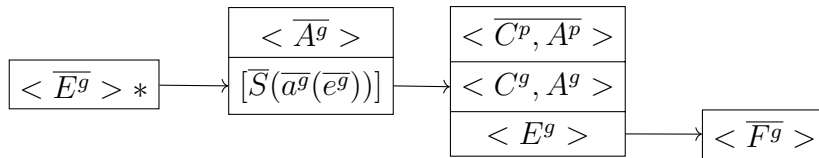
1. **Oletus 1.** Kielimallin tuottamat teokset ovat arvokkaita ja uusia.
  - (a) **Korollaari.** Arvokkaiden ja uusien teosten on täytynyt syntyä luovista teoista.
  - (b) **Korollaari.** Koska teokset ovat uusia, kaikki luovat teot eivät ole tapahtuneet lähdeaineistoa luodessa (L-osassa).
2. **Oletus 2.** Kielimallin algoritmin kehittäjät eivät osaa ennustaa mallin toimintaa.
  - (a) **Korollaari.** Kaikki luovat teot eivät ole syntyneet kielimallin algoritmia kehitettäessä (K-osassa).
3. **Johtopäätös.** Siispä osa luovista teoista ei ole lähdeaineiston luojien tai kielimallin tekijöiden ennalta suunniteltavia, vaan ne tapahtuvat kielimallin koulutuksen tai generoinnin aikana.

Puhun tässä luvussa teoksista ”arvokkaina” ja ”uusina” aivan kuin ne olisivat binäärisiä totuusarvoja, mikä ei pidä paikkansa. Ritchien menetelmässä tarkastellaankin arvoa ja tyypillisyyttä lukuina välillä  $[0, 1]$  ja kehoitetaan tutkimaan näiden jakaumia [33]. Konkreettista järjestelmää arvioitaessa tämä on välttämätöntä. Tässä luvussa oletan, että teokset ovat ”riittävän usein riittävän uusia ja arvokkaita”.

Avaan seuraavaksi päättelyketjua ja luovia tekoja tarkemmin FACE-mallin avulla. Tarkastelen sekä konseptin luomista (FACE-mallissa  $\langle C^g \rangle$ ) että esteettisyysmetriikan luomista ( $A^g$ ). Kielimallin luovat teot on esitetty kuvassa 3.1.

### 3.2.1 Konseptin luominen

Coltonin luovan kolmijalan jokainen jalka jakaantuu kolmeen haaraan: ohjelmoijaan, ohjelmaan ja yleisöön. Tutkitaan esimerkkinä kuvassa 3.2 esiteltyä runoilevaa ohjel-



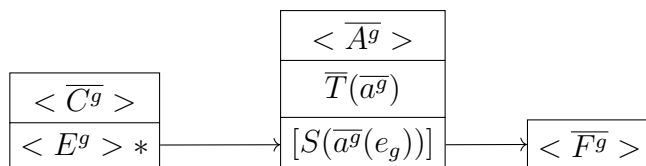
**Kuva 3.1:** Kielimallin luovat teot FACE-mallin mukaan:

1. Ihmiset ovat luoneet useita teoksia ( $\langle \overline{E^g} \rangle *$ )
2. Ihminen on luonut metriikan teosten arviointia varten ( $\langle \overline{A^g} \rangle$ )
3. Ihminen on valikoinut metriikan perusteella teoksia mallin lähdeaineistoon ( $[\overline{S}(\overline{a^g}(\overline{e^g}))]$ )
4. Ihminen luo kielimallin koulutusalgoritmin, eli menetelmän konseptien ja metriikoiden luomiseen ( $\langle \overline{C^p}, \overline{A^p} \rangle$ )
5. Kielimallin koulutusalgoritmi määrittää mallin parametrit, jotka ovat samalla konsepti teosten luomiseen ja metriikka ( $\langle C^g, A^g \rangle$ )
6. Kielimalli luo teoksen konseptin pohjalta ( $\langle E^g \rangle$ )
7. Ihminen kirjoittaa teokselle kehystiedon ( $\langle \overline{F^g} \rangle$ )

maa. Ohjelma luo runoja ohjelmoijan luoman konseptin (algoritmin) pohjalta, min­kä jälkeen se valikoi parhaat runot ohjelmoijan luoman metriikan pohjalta ja lopuksi ihminen kirjoittaa näille runoil­le kehystiedon. Yhdessä ohjelmoija, ohjelma ja yleisö, kolmijalan kolme haaraa, voivat täyttää luovuuden kriteerit ja tuottavat teoksia, jotka ovat arvokkaita ja uusia. Suurin osa luovuudesta kuitenkin on järjestelmän ihmisissä. On mahdollista väittää, että ohjelma ei ole ollenkaan luova, sillä sen toiminta on täysin ennustettavaa ja se vain toteuttaa ilman autonomiaa ohjelmoijansa suunnitelmaa sen sijaan, että se osaisi itse luoda konsepteja ja uusia generoimismenetelmiä.

Myös kielimallien tapauksessa pitää tutkia, onko teoksen luomiseen käytetyn konseptin luonut oikeasti kielimallin kouluttaja tai koulutusalgoritmin kehittäjä ("K-osa"). Koska kielimalli on koulutettu lähdeaineiston pohjalta, myös lähdeaineiston tekijät ovat henkilöitä, joiden osallisuutta pitää tutkia. Lähdeaineiston tekijöillä tarkoitan sekä aineiston teosten varsinaisia tekijöitä että aineiston valikoijia, jotka ovat päättäneet, mitä aineistoon sisällytetään ("L-osa").

**Kohdistus pois K-osasta.** Kielimallin koulutusalgoritmin kehittäjää ei minusta voi pitää konseptin luoja­na, jos hän ei ole etukäteen tutkinut lähdeaineistoa ja erityisesti suunnitellut koulutettua kielimallia tuottamaan sillä haluamanslaisia teoksia. Koska oletan, että kielimalli on ennustamaton "musta laatikko", ei koulutusalgoritmin kehit-



**Kuva 3.2:** Runoileva ohjelma FACE-mallin mukaan. Kaavion pohjana on Coltonin esimerkkikaavio [13, kuva 1, kohta b], johon on lisätty kehystiedon luonti. Luova järjestelmä koostuu seuraavista vaiheista:

1. Ihminen luo konseptin runojen luomiseen ( $\langle \overline{C^g} \rangle$ )
2. Kone luo useita runoja konseptin pohjalta ( $\langle E^g \rangle *$ )
3. Ihminen luo esteettisen metriikan ( $\langle \overline{A^g} \rangle$ )
4. Ihminen muuntaa metriikan ohjelmakoodiksi ( $\overline{T}(\overline{a^g})$ )
5. Ohjelma valitsee metriikan (merkitään  $\overline{a^g}$ ) avulla luoduista runoista parhaan (merkitään  $e^g$ ) ( $[S(\overline{a^g}(e_g))]$ )
6. Ihminen kirjoittaa runolle kehystiedon ( $\langle \overline{F^g} \rangle$ )

täjä voi etukäteen ennustaa, miten eri algoritmeilla koulutetut mallit käyttäytyvät eri lähdeaineistojen kanssa. Ennustamattomuuden määritelmästä seuraa, että ainoa tapa, jolla tämän oletuksen pätiessä kehittäjä voisi ohjata mallia toimimaan haluamallaan tavalla, olisi tuottaa useita eri koulutusalgoritmeja, suorittaa niitä, tutkia tuottavatko ne halutunlaisia teoksia ja valita algoritmeista se, joka toimii halutulla tavalla. Tällaisessa tilanteessa koulutusalgoritmia optimoitaisiin vain kehittäjän käyttämän lähdeaineiston suhteen. Jos tällä tavalla optimoitua algoritmia sovellettaisiin johonkin muuhun aineistoon, joka poikkeaisi kehittäjän käyttämästä aineistosta, ennustamattomuuden mukaisesti tulokset eivät enää olisi kehittäjän päätettävissä, vaan syntyneet koulutuksen aikana.

Neuroverkkokielimallien koulutusalgoritmeja ei tietääkseni ole kehitetty tuottamaan juuri tietyn tyyppisiä teoksia. Päinvastoin niiden kouluttamista testataan usein useilla eri aineistoilla (esimerkiksi [16]). Koska tällöin algoritmin kehittäjä ei ole valinnut algoritmia nimenomaan tuottamaan haluamanslaisia teoksia, ei algoritmin tekijän voida katsoa ennalta luoneen teoksia eikä niiden konseptia (koulutettua mallia), vaan hän tosiaan on luonut vain koulutusalgoritmin (FACE-mallissa menetelmän konseptin luomiseen,  $\langle \overline{C^p} \rangle$  [12]).

**Kohdistus pois L-osasta.** On myös mahdollista, että koulutusalgoritmi ottaa konseptin lähdeaineistosta siten, että lähdeaineiston tekijöitä pidetään konseptin luoja.

Tämä on mahdollista, jos malli on ylioppinut ja toistaa joko suoraan lähdeaineiston teoksia tai hyvin kaavamaisesti niiden osia. Jos konsepti on otettu lähdeaineistosta, voisi lähdeaineistoa tutkimalla ennustaa, millaisia teoksia malli tuottaa, jolloin teoksia ei voisi pitää uusina. Koska oletan teosten olevan uusia, täytyy konseptin olla syntynyt koulutuksen aikana, eli koulutusalgorithmi on tehnyt luovan teon  $\langle C^g \rangle$ .

### 3.2.2 Esteettisyysmetriikan luominen

Runoilevan ohjelman tapauksessa ihminen on myös luonut metriikan runojen arviointia varten, jota ohjelma vain toteuttaa, eikä ohjelma näin ollen ole itsenäisesti luova. Kielimallit eivät sisällä tällaista jälkikäteistä arviointia, mutta kielimallin tuottamaa seuraavan saneen todennäköisyysjakaumaa voi pitää arviointina. Kielimalli tuottaa vain saneita, joita se pitää todennäköisinä eli hyvinä. Kielimallilla on siis esteettisyysmetriikka. On jälleen tarkasteltava, ovatko tämän metriikan voineet luoda koulutusalgorithmin tai lähdeaineiston tekijät.

Vaikka kielimallilla olisi oma konsepti, eli se loisi teoksia omalla menetelmällään, joka eroaa lähdeaineiston teosten konsepteista, on mallin määritelmällinen tarkoitus kuitenkin olla yleistys lähdeaineistostaan, eli mallin tulisi pitää lähdeaineiston teoksia todennäköisinä eli ”hyvinä”. Mallin luovuus syntyy ekstrapoloinnista: mallin on tarkoitus pitää muitakin kuin lähdeaineiston teoksia todennäköisinä. Koska todennäköisyyksien pohjana on kuitenkin käytetty lähdeaineistoa, täytyy näillä muilla teoksilla olla joitakin yhteisiä piirteitä lähdeaineiston teosten kanssa. Jos aineiston valikoijien valintakriteerinä on ollut esteettisyysmetriikka\*, malli saattaa pyrkiä kopioimaan tätä metriikkaa. On huomattava, että koulutusalgorithmi ei tiedä, mitkä teokset jätettiin valitsematta aineistoon, joten se ei voi mallintaa metriikkaa kuitenkaan täydellisesti. Tästä syystä on mallin koulutuksen aikana tapahtunut luova teko  $\langle A^g \rangle$ , sillä koulutusalgorithmin on täytynyt tehdä omaa tulkintaa siitä, mitkä aineiston piirteet ovat oleellisia metriikan kannalta ja mitkä eivät.

---

\*Kaikki aineistot eivät välttämättä perustu esteettisyysmetriikkaan. Äärimmäinen tapaus on tilanne, jossa aineisto on muodostettu satunnaisesti, esimerkiksi etsimällä hakurobotilla Internetistä mitä tahansa tietyn kielisiä tekstejä suodattamatta niitä muuten lainkaan. Voi myös olla, että aineisto perustuu johonkin muuhun kriteeriin kuin esteettisyyteen. Jos näin on, voi olla, etteivät lähdeaineiston teokset ihmisen mielestä ole hyviä. Koska kielimalli mallintaa lähdeaineistoa, eivät sen generoimat teoksetkaan enää välttämättä ole hyviä tai arvokkaita.

Koska kielimallilla ei ole erillistä esteettisyysmetriikkaa, voidaan sanoa, että luovat teot  $\langle C^g \rangle$  ja  $\langle A^g \rangle$  ovat itse asiassa sama teko, jossa kielimalli muodostaa oman todennäköisyysjakaumansa, josta voimme arpoa kielimallin hyvänä pitämiä teoksia.

Vaikka kielimalli tekeekin enemmän luovia tekoja kuin esimerkin runoileva ohjelma, on yhä muistettava, että ihminen on sekä kehittänyt kielimallin koulutusalgoritmin että valikoinut lähdeaineiston, jonka pohjalta malli on luotu. Edellytys tekojen pitämiselle luovina on, että mallin tuottamat teokset ovat riittävän arvokkaita ja uusia esimerkiksi Ritchien menetelmällä tutkittuna.

### 3.2.3 Huomio deterministisyydestä

Tarkastellaan seuraavaa vasta-argumenttia:

Vaikka kielimallin koulutusalgoritmin tehnyt ihminen ei osannut ennustaa, millaisia teoksia malli tekee, hän on kuitenkin algoritmia suunnitellessaan päättänyt, miten malli kouluttuu ja siksi hän on tehnyt luovan teon  $\langle \overline{C^g} \rangle$ . Mallin kouluttaminen ja suorittaminen ovat vain ihmisen luoman konseptin suorittamista, eli ovat molemmat osa luovaa tekoa  $\langle E^g \rangle$ .

Tässä vasta-argumentissa tarkastellaan kielimallin kouluttamista ja siitä generoimista yksittäisenä luovana tekona  $\langle E^g \rangle$  ilman sisäistä rakennetta. Colton ym. [12] eivät tarkkaan määrittele, missä luovien tekojen rajat menevät. ”Konsepti” on määritelty vain tietokoneohjelmaksi. On mahdollista tulkita sekä koulutusalgoritmin että generointialgoritmin olevan osa yhtä tietokoneohjelmaa ja siten yhtä konseptia, mutta tällainen tulkinta ei mielestäni ole mielekäs. Argumentissa yritetään kiistää ohjelman sisäisen rakenteen olemassaolo, mikä ei ole hyödyllistä, koska juuri sisäinen rakenne on se, mitä haluamme tutkia. Vaikka merkitsisimme ohjelman toimintaa yhdellä merkinnällä  $E^g$ , se ei muuta sitä, että päätökset siitä, millaisia teoksia ohjelma tuottaa, tapahtuvat koulutusalgoritmin suorituksen aikana, eivät sen ohjelmoinnin aikana. Olisi hyödyllistä käyttää niistä merkintöjä  $A^g$  ja  $C^g$  sen sijaan että ne niputetaisiin yhteen tekoon  $E^g$ .

On mahdollista argumentoida, että päätökset siitä, mitä ohjelma tekee, itse asiassa ovat tapahtuneet koulutusalgoritmin luomisen aikana. Oletetaan yksinkertaisuuden vuoksi, että algoritmi on täysin deterministinen, mikä ei välttämättä kaikkien algoritmien tapauksessa pidä paikkaansa. Algoritmi siis käyttäytyy samalla aineistolla aina samalla

tavalla ja tuottaa saman mallin. Se on siis ennalta määrätty tai ”predestinoitu” tuotamaan tietynlaisia teoksia, eikä sillä ole todellista valtaa omaan toimintaansa. Tämäkään argumentti ei minusta ole mielekäs. Ennustamattomuuden mukaan algoritmin luoja ei tiedä, miten algoritmi tulee käyttäytymään. Tämä tieto syntyy vasta, kun algoritmi suoritetaan.

Jos hyväksyttäisiin, että deterministisen järjestelmän tuottama tieto on olemassa sen syntymästä lähtien, ei mikään deterministinen tietokoneohjelma voisi olla luova. On huomattava, että myös pseudosatunnaislukugeneraattorit ovat ennustamattomia ja kaaottisia, mutta deterministisiä järjestelmiä, joten niidenkään käyttäminen ei takaisi luovuutta. Oikeastaan ainoa tapa olla luova olisi käyttää jotakin ulkoista, ”aitoa” satunnaisuuden lähdettä. Tämä on sinänsä mahdollinen tapa määritellä luovuus, mutta sitä ei ole tunnustettu laskennallisen luovuuden alalla. Se ei esimerkiksi ole osan 2 alussa olevien määritelmien edellytys. Boden [5] ei pidä edes ennustamattomuutta edellytyksenä luovuudelle.

### 3.3 Tavoitteellisuus

Ventura [37] pitää uutuuden ja arvon rinnalla *tavoitteellisuutta* yhtenä tekijänä, joka erottaa luovuuden pelkästä generoinnista. Myös Colton ym. [13] pitävät tavoitteellisuutta edellytyksenä sille, että ihmiset kokevat ohjelman luovaksi. Luovalla järjestelmällä on siis heidän mukaansa oltava tarkoitus tai tavoite.

Tavoitteellisuuden avulla on mahdollista kohdentaa luovuutta pois Y-osasta eli yleisöstä: jos generoidulla tekstillä tavoitellaan jotain tiettyä tulkintaa, on yleisön vastuu tulkinnasta pienempi.

Tässä luvussa käsittelen kahdenlaisia tavoitteita, joita kielimallilla voi sanoa olevan. Ensimmäinen ryhmä tavoitteita on ”generoinnin aikaiset tavoitteet” eli ne tavoitteet, jotka on jotenkin koodattu kielimallin kehoitteeseen. Koska kehote muuttuu, kun sen perään arvotaan sanoja, voi kielimallin ajatella voivan myös muuttaa kehoitteeseen koodattuja tavoitteita.

Toinen ryhmä on ”taustalla olevat tavoitteet”, eli ne tavoitteet, jotka kielimallin tekijöillä on ollut. Nämä tavoitteet saattavat olla hyvin epämääräisiä ja niihin on voinut vaikuttaa suurikin joukko ihmisiä.

### 3.3.1 Generoinnin aikaiset tavoitteet

Kielimallin määritelmällinen tavoite on ennustaa, mikä sana sopii annettuun kontekstiin eli kehoitteeseen. Kun teosta generoidaan, lisätään kehoitteen loppuun uusi sane todennäköisyysjakauman perusteella, ja tätä uutta kehoitetta käytetään taas seuraavan saneen lisäämiseen. Voi siis sanoa, että kielimalli voi muuttaa omaa kehoitettaan ja näin omaa tavoitettaan. Nämä tavoitteet ovat luonteeltaan paikallisia ja väliaikaisia, mutta myös kaikkein konkreettisimpia.

Tutkitaan esimerkkinä musiikkia generoivaa kielimallia, joka tuottaa nuottien lisäksi merkintöjä kappaleiden halutusta tunnetilasta. Kappaleen alussa saattaa olla esimerkiksi merkintä, että kappaleen on tarkoitus olla surullinen. Tällöin kielimallin tavoitteena olisi generoida surullista musiikkia. Kesken kappaleen tuottamisen malli saattaa kuitenkin generoida uuden merkinnän, jonka mukaan kappale on tästä pisteestä alkaen surullisen sijasta haikeaa. Tämän jälkeen malli yrittää tuottaa uuden tavoitteen mukaisesti haikeaa musiikkia. Voidaan siis kuvitella malli, joka pystyy sekä toteuttamaan tavoitteitaan että muuttamaan niitä.

Tällaiset tavoitteet vaativat, että kielimallin koulutusaineisto sisältää tavoitteita ilmaisevia sanoja, kuten tunnetilaa ilmaisevia merkintöjä. Kaikilla kielimalleilla ei näitä ole, mutta väitän, että niilläkin on samankaltaista toimintaa. Esimerkiksi esseetä kirjoittava GPT-3 saattaa kirjoittaa ensin kappaleen yhdestä näkökulmasta ja sitten vaihtaa näkökulmaa tuottamalla tätä ilmaisevan luonnollisen kielen ilmauksen, minkä jälkeen tavoite on vaihtunut.

On syytä todeta, että puhuessani ”luonnollisen kielen ilmauksista” en väitä, että kielimallilla olisi todellista ymmärrystä siitä, mitä sanat tosielämässä tarkoittavat, vaan että tietyt fraasit muuttavat tietynlaiset lauseet todennäköisemmiksi kuin toisenlaiset, ja tuottamalla tällaisen fraasin malli voi muuttaa tuottamaansa todennäköisyysjakautta. Sama voidaan todeta myös musiikkiteknoalustasta: vaikka malli osaa tuottaa tunnetilaa ilmaisevan merkinnän, ei se tietenkään oikeasti ymmärrä, mitä tunteet kuten suru ja ilo tarkoittavat. Tämä ei kuitenkaan tarkoita, etteikö tunnetila voisi olla tekoälyn tavoite. Jos ihmissäveltäjälle kerrottaisiin, että hänen teoksensa pitäisi kuulostaa ”nuokolta” ja että kokosävelaskeleet luovat hyvin nuokon tunnelman, olisi hänellä tavoite jota toteuttaa. Jos säveltäjä ei tietäisi, mitä ”nuokko” tarkoittaa, hän lisäisi kappaleeseensa paljon kokosäveliä ja toivoisi parasta. Tietysti jos hän ymmärtäisi sanan

merkityksen, hän osaisi tuottaa parempia kappaleita, mutta hänellä on tavoite joka tapauksessa.

Koska monimutkaiset kielimallit ottavat huomioon pitkiä riippuvuussuhteita, kehoitetta muuttamalla voi muuttaa eri tason tavoitteita, kuten kokonaisen teossarjan tavoitteita, yhden teoksen tavoitteita tai teoksen eri osien tavoitteita. Mallin kykyyn muuttaa näitä tavoitteita vaikuttaa merkittävästi se, kuinka pitkä mallin kehote voi maksimissaan olla. Ääritapauksena Markovin ketju, jonka kehoteena on vain edellinen sana, ei pysty asettamaan itselleen mitään tavoitteita, sillä se ”unohtaa” ne heti yhden sanan tuottamisen jälkeen. Monimutkainen neuroverkko, jonka kehoitteena on tuhansia sanoja, voi ottaa huomioon useita yhtäaikaista tavoitteita pitkissäkin teoksissa. Esimerkki pitkän riippuvuussuhteen teossarjaa koskevasta tavoitteesta voisi olla mallin aivan aluksi kirjoittama sisällysluettelo, jonka perusteella malli tuottaa useita eri aiheita käsitteleviä tekstejä. Vaikka malli tuottaakin tekstiä sane kerrallaan, se voi siis ikään kuin suunnitella sisällysluettelon avulla tekstin rakenteen etukäteen ja luoda tekstiin korkean tason rakennetta.

Liitteessä A on GPT-3:a käyttävä esimerkki runogeneraattorista, joka valitse itse ennen runon generoimista, millainen tunne runossa on. Esimerkin tarkoitus on havainnollistaa, millainen tavoitetta merkitsevä ilmaisu käytännössä voisi olla. Tarkoitukseni ei ole ottaa kantaa GPT-3:n luovuuteen tai kykyyn generoida runoja.

Kielimallin kykyä muuttaa tavoitteita rajoittaa merkittävästi se, ettei mallilla ole kykyä vuorovaikuttaa suoraan maailman kanssa eikä sillä ole todellista ymmärrystä siitä, miten maailma toimii. Kielimallilla voi siis olla tavoitteena tuottaa musiikkia, joka assosioituu sanaan ”surullinen”, mutta mallin tavoite ei voi olla tuottaa musiikkia, joka saa kuuntelijan tuntemaan itsensä surulliseksi, sillä ei voi ymmärtää, mitä surullisuus tarkoittaa tai mitä seurauksia surullisella musiikilla on kuuntelijalle. Ventura [37] pitää tällaista ”tietoisuutta” edellytyksenä varsinaiselle luovuudelle. Jotta järjestelmä osaisi ymmärtää tekojensa seurauksia todellisessa maailmassa, olisi siinä oltava kielimallin lisäksi jokin muu osa, joka havainnoi maailmaa ja kouluttaa kielimallia lisää havaintojensa pohjalta.

### 3.3.2 Tavoitteet kielimallin taustalla

Guckelsberger ym. [22] toteavat, että ns. ”miksi?”-kysymys on tärkeä: *miksi* järjestelmä päätti toimia niin, kuin se toimi? He arvelevat, ettei mikään olemassaoleva järjestel-

mä osaa vastata tähän kysymykseen itsenäisesti, vaan toistavat ihmisten järjestelmälle antamia tavoitteita. Tämä johtuu siitä, että laskennallisessa luovuudessa käsitellyt järjestelmät ovat olemassa ollakseen luovia, jolloin niiden pohjimmainen syy luoviin tekoihin on, että ihminen on suunnitellut ne tekemään luovia tekoja.

Kielimalleja itseasiassa ei ole suunniteltu olemaan luovia, vaan alunperin ne on kehitetty yksinkertaisesti kertomaan, mitkä lauseet ovat todennäköisempiä kuin toiset, mikä on tärkeää esimerkiksi puheentunnistuksessa, jossa järjestelmän on päätettävä, mikä on oikea tulkinta epäselvästi lausutulle lauseelle. Kielimalleja on käytetty myös muiden monitulkintaisuuksien ratkomiseen, kuten sanojen sanaluokkien selvittämiseen. Se, että kielimalleilla voi generoida hyvää tekstiä lähdeaineiston avulla on uusi ilmiö, jota 1990-luvulla pidettiin vielä mahdottomana. [9] Tutkiessani kielimallien luovuutta en siis yritä perustella, miksi luovaksi suunniteltu järjestelmä on luova, vaan miten luovuus on syntynyt sivuvaikutuksena aivan toisesta tarkoituksesta, eli lauseiden todennäköisyyksien määrittämisestä.

Yksittäisen kielimallin syntyyn vaikuttaneilla ihmisillä on usein hyvin erilaisia tavoitteita. Koulutusalgoritmin kehittäjät toivovat usein, että malli toimii hyvin niillä osaluilla, joihin he ovat keskittyneet. Esimerkiksi Transformer XL -mallin kehittäjien Dai ym. [16] tavoite oli, että malli osaisi tulkita pitkän matkan riippuvuussuhteita. Lähdeaineiston kokoajat taas saattavat valikoida teoksia joko omien subjektiivisten kriteeriensä perusteella, kuten esteettisen arvon, tai mahdollisesti ilman mitään kriteerejä, yrittäen koota mahdollisimman ison aineiston laadusta riippumatta. Mallin kouluttaja saattaa esikäsitellä aineistoa vielä omien kriteeriensä pohjalta. Lopulta mallin käyttäjä, joka määrittää kehotteet, joiden pohjalta tekstiä generoidaan, tekee tämän taas omien tavoitteidensa mukaisesti.

Näiden lähtökohtien vuoksi mallilla voi katsoa olevan paljon ulkoisia tavoitteita, jotka vaikuttavat malliin. Vaikka malli voikin siis muuttaa kehotteeseen koodattuja tavoitteita, vaikuttavat mallin toimintaan myös nämä ulkoiset tavoitteet.

Minusta on kyseenalaista, kuinka paljon tekijöiden tavoitteilta voi välttyä. Tekoälyt ovat määritelmällisesti lopulta ihmisen luomia, sillä täysin spontaanisti syntyneet tekoälyt eivät enää ole tekoälyjä vaan luonnollisia älyjä. Vaikka tekoäly olisi *tavoitetietoinen*, eli kykenevä arvioimaan ja muuttamaan omia tavoitteitaan [28], on silti joku luonut sen omien tavoitteidensa vuoksi. On epäselvää, kuinka paljon painoarvoa tavoitteellisuuden arvioinnissa on annettava paikallisille ja ulkoisille tavoitteille.

Se, että kielimallin taustalla on paljon eri ihmisten tavoitteita antaa sille mahdollisesti tilaisuuden löytää omia tavoitteitaan näiden tavoitteiden välimaastosta, mikä olisi hankalampaa, jos tekoäly olisi täysin yhden ihmisen tai yhtä tavoitetta edistävän ryhmän suunnittelema. Laajalla lähdeaineistolla koulutetun kielimallin on esimerkiksi opittava yleistämään useiden eri tavoitteista syntyneiden lähdeaineiston teosten piirteitä, joten mallin tuottamat teosten tavoitteiden ei voi sanoa suoraan toteuttavan lähdeaineiston tekijöiden tavoitteita.

Järjestelmän tuottamat vastaukset ”miksi?”-kysymyksiin ovat osa laajempaa *kehystietoon* käsitettä, jonka tuottaminen on FACE-mallin luova teko  $F^g$ . On epävarmaa, voiko kielimalli luoda kehystietoa. GPT-3-tyyppiset mallit osaavat jäljitellä kehystietoa ja tuottaa generoituja selityksiä, taustoja ja perusteluja teoksilleen, mutta nämä selitykset eivät välttämättä mitenkään vastaa todellista luomisprosessia. Vaikka kielimallin tavoitteet muuttuvatkin, malli ei osaa perustella näitä muutoksia. Tämä ei kuitenkaan tarkoita, etteikö malli olisi muilla tavoin luova.

# 4 Esimerkkitapaus: säveltävä kielimalli

Esimerkkinä kielimallien luovuuden analysoinnista koulutin itse oman musiikkia generoivan kielimallin. Valitsin musiikin luonnollisen kielen sijaan, sillä haluan välttää pohdintoja siitä, ymmärtääkö kielimalli sanojen merkityksiä. Kieltä generoivia malleja analysoidessa on helppo syyllistyä antropomorfisointiin. Esimerkiksi jos GPT-3:lle antaa luonnollisella kielellä kuvauksen siitä, millainen essee sen tulee kirjoittaa, voidaan perustellusti kyseenalaistaa, ymmärtääkö GPT-3 tuosta selityksestä mitään muuta kuin pinnallisia sanojen todennäköisyyksiä. Jos taas säveltävälle kielimallilleni taas antaa symbolin, joka merkitsee kappaleen pituuden olevan 16 tahtia, ja malli tosiaan tuottaa 16 tahdin mittaisen kappaleen, voi mallin sanoa oikeasti ymmärtävän mitä tuo symboli tarkoittaa.

Vaikka nuotit eivät sinällään liity mihinkään tosielämän käsitteisiin toisin kuin luonnollisen kielen sanat, on niissä kuitenkin joitakin merkityksiä, joita malli ei voi ymmärtää, esimerkiksi musiikin aiheuttamia tunnetiloja tai mielikuvia. Tästä huolimatta on silti mahdollista ajatella, että malli kykenee arvostamaan jollakin muulla tavalla tuottamaan musiikkia, sillä se voi arvioida sen todennäköisyyttä eli ”hyvyyttä”.

## 4.1 Aineisto

Käytin lähdeaineistona Wikifonia-korpusta<sup>†</sup>, joka koostuu 6 712 MusicXML-tiedostosta. Tiedostot sisältävät sekä klassista että populaarimusiikkia. Jokaisessa teoksessa on vain yksi melodia sekä sointumerkinnät, eli ne ovat ns. ”lead sheet” -nuotteja.

Käytin Music21-kirjastoa [15] jäsentääkseni kappaleet tietokoneen muistiin. Tämän jälkeen transponoin jokaisen kappaleen viidellä satunnaisella tavalla<sup>‡</sup>, ja muutin nämä viisi muunnosta sekä alkuperäisen kappaleen tekstipohjaiseen muotoon. Tällä menetelmällä muodostin 38 466 kappaleesta, yhteensä 21 840 948 saneesta koostuvan aineiston.

---

<sup>†</sup><https://web.archive.org/web/20131004182741/http://www.wikifonia.org/>

<sup>‡</sup>Toivon transponoinnin parantavan modulaation laatua ja mahdollisesti ”sekoittavan” mallia, sillä sen on opittava kaikki tahtilajit eikä vain yksi.

## Adott Isten szekeret

*Hungarian folk song*

Voice

A - dott Is - ten sze - - ke - ret, Ü - ve - gem - nek fe - - ne - ket,  
Sze - ke - rem - nek ke - - re - ket,

ab - ból i - szom e - - le - get.

Kuva 4.1: Eräs Wikifonia-korpuksen teos. Tämä on eräs sen lyhyimmistä teoksista.

## 4.2 Tekstipohjainen muoto

Käyttämässäni tekstimuodossa kappale on esitetty joukkona saneita. Saneet vastaavat musikaalisen notaation merkintöjä ja ne on esitetty taulukossa 4.1. Esimerkki kokonaisuudesta kappaleesta tekstimuodossa on esitetty ohjelmalistauksessa 1.

Koska useampi sävel saattaa soida yhtä aikaa sointuna, nuottisaneiden lisäksi muodossa on erillinen kursoria siirtävä sane. Koodi **note:C5;1.0 note:E5;1.0 wait:1.0** merkitsee siis kahdesta sävelestä koostuvaa neljäsosanuotin mittaista sointua, kun taas **note:C5;1.0 wait:1.0 note:E5;1.0 wait:1.0** merkitsee kahta peräkkäistä neljäsosanuottia.

Tekstipohjaiseen muotoon olen lisännyt musikaalisten merkintöjen lisäksi tahtinumerot, jotka ovat negatiivisia ja kasvavat kohti nollaa. Tämä on tehty, jotta kielimalli tietäisi kehotteen perusteella, kuinka pitkä kappale sen on tuotettava. Kielimalli voi mahdollisesti oppia hyödyntämään lukuja osatakseen aloittaa uudet kappaleen osat järjestyksessä kohdissa koko kappaletta, sillä kehotteesta käy ilmi, missä kohdassa kappaletta generaatio on.

## 4.3 Kielimalli

Koulutin tekstipohjaisella aineistolla Transformer XL -arkkitehtuuriin [16] pohjautuvan kielimallin. Valitsin arkkitehtuurin, sillä sen avulla on aiemmin saatu hyviä tuloksia musiikin säveltämisessä [39]. Koulutin mallia Helsingin yliopiston Ukko 3 -laskentaklusterilla noin kahden tunnin ajan.

```
|| bar:-3 key:0 time:4/4
note:A4;0.5 wait:0.5
note:C5;0.5 wait:0.5
note:B4;0.5 wait:0.5
note:A4;0.5 wait:0.5
note:B4;1.0 wait:1.0
note:G4;0.5 wait:0.5
note:E4;0.5
repeat:end wait:0.5
|| bar:-2 repeat:start
note:D4;0.5 wait:0.5
note:D4;0.5 wait:0.5
note:D4;0.5 wait:0.5
note:D4;0.5 wait:0.5
note:G4;1.0 wait:1.0
note:G4;0.5 wait:0.5
note:G4;0.5 wait:0.5
| bar:-1
note:A4;0.5 wait:0.5
note:A4;0.5 wait:0.5
note:G4;0.5 wait:0.5
note:F4;0.5 wait:0.5
note:E4;1.0 wait:1.0
note:E4;0.5 wait:0.5
note:E4;0.5
repeat:end
end
```

**Ohjelmalistaus 1:** Kuvassa 4.1 esitetty kappale tekstimuodossa. Rivinvaihdot on lisätty lukemisen helpottamiseksi. Kappaleen alussa ei ole `repeat:start`-sanetta, sillä toistoa ei ollut merkitty kappaleen alkuun musikaalisessa notaatiossa.

Sanetyyppi	Esimerkki	Selitys
Kursorin siirto	<b>wait:0.5</b>	Siirtää kursoria yhden kahdeksasosanuotin verran
Tahtilaji	<b>time:4/4</b>	4/4-tahtilajimerkintä
Sävellaji	<b>key:3</b>	Kolme korotusta (A-duuri, fis-molli)
Tempo	<b>tempo:120</b>	Kappaleen nopeus on 120 iskuu/ minuutti
Nuotti	<b>note:C5;1.0</b>	Neljäsosanuotti c oktaavissa 5
Tauko	<b>rest:0.5</b>	Kahdeksasosatauko
Kaari	<b>tie:start</b>	Yhdyskaari alkaa edeltävästä nuotista
	<b>tie:continue</b>	Yhdyskaari jatkuu saapuu ja lähtee edeltävästä nuotista
	<b>tie:stop</b>	Yhdyskaari loppuu edeltävään nuottiin
Sointumerkki	<b>chord:C#m</b>	cis-molli
Kertaus	<b>repeat:start</b>	Nykyisen tahdin vasemmassa reunassa on kertauksen alku
	<b>repeat:end</b>	Nykyisen tahdin oikeassa reunassa on kertauksen loppu
Voltan hakaset	<b>bracket:1_start</b>	Nykyisestä tahdistä alkaa Voltan hakanen
	<b>bracket:1_end</b>	Nykyiseen tahtiin loppuu Voltan hakanen
Tahtiviiva		Yksi tahtiviiva
		Kaksoistahtiviiva
Tahdin numero	<b>bar:-16</b>	Kappaleessa on jäljellä 16 tahtia
Lopetussane	<b>end</b>	Jokaisen kappaleen lopussa

Taulukko 4.1: Tekstipohjaisen muodon sanetyypit.

## 4.4 Ennustamattomuus

Ennustamattomuus on eräs kriteereistäni. Ennustamattomuuden arviointi käytännössä on vaikeaa, koska en voi puhua mallin koulutusalgoritmin tekijöiden puolesta. Siispä tämä osuus jää arvelujen tasolle. Itse en osaa ennustaa algoritmin toimintaa.

Ensinnäkin mallin koulutusalgoritmin suunnittelijat Dai ym. [16] eivät todennäköisesti suunnitelleet mallia musiikin generointia varten. He käyttävät esimerkkinä mallin sovelluskohteista luonnollisen kielen käsittelyn ongelmia. Siispä on epäselvää, missä määrin he pystyvät ennustamaan mallin käyttäytymistä hyvin erilaisen tehtävän eli säveltämisen yhteydessä.

Toisekseen käyttämäni malli on iso: se sisältää 33 469 243 erillistä parametria. Näin suuresta määrästä puhuttaessa on mahdotonta, että ihminen ymmärtäisi jokaisen parametrin vaikutuksen mallin toimintaan. Koska erilaisia parametrien yhdistelmiä on mittaamattoman paljon, on hyvin todennäköistä, että osa yhdistelmistä toimii ennustamattomalla tavalla.

Pidän näiden seikkojen vuoksi todennäköisenä, että malli on määrittelemälläni tavalla ennustamaton, mutta tätä on vaikeaa tai mahdotonta todistaa täysin.

## 4.5 Luovuuden arvioinnin mittarit

Käytän luovuuden arvioinnissa Ritchien menetelmän [33] (ks. luku 2.1.2) *tyypillisyyttä* ja *arvoa*. Lisäksi mittaan Ritchien menetelmästä poiketen *uutuutta* omilla mittareillani, sillä hänen mallissaan tutkitaan vain tyypillisyyttä ja sitä, löytyykö identtinen teos lähdeaineistosta. Säveltävän kielimallin tapauksessa on hyvin epätodennäköistä, että täsmälleen identtinen kappale olisi olemassa, vaan on minusta järkevämpi mitata kappaleiden saamia vaikutteita tutkimalla samankaltaisuutta lähdeaineiston kappaleisiin.

### 4.5.1 Uutuus

Kappaleiden uutuuden mittaamiseksi vertaan niiden samankaltaisuutta lähdeaineistoon kahdella eri mittarilla, *kopiointiasteella* sekä *sumealla kopiointiasteella*.

Kopiointi mittaa, kuinka moni kuuden nuotin tai tauon jono esiintyy sellaisenaan lähdeaineistossa. Olkoon  $L_n$  lähdeaineisto, jonka saneina on pelkkiä nuotteja ja tauko-

ja. Kappaleen  $w$  6-grammien joukko  $G_6(w) = \{w_{t,t+6} \mid 1 \leq t \leq |w| - 6\}$ . Olkoon  $L_n^6 = \{G_6(w) \mid w \in L\}$  joukko kappaleista esitettynä 6-grammeina. Määritellään *kopiointiaste*  $K$  seuraavasti:

$$A_K = A_{L_n^6}(w) = \frac{100}{|w|} \sum_{g_6 \in G_6(w)} \begin{cases} \frac{\sum_{w' \in L \setminus w} f_{g_6, w'}}{n_{g_6}} & \text{jos } n_{g_6} > 0 \\ 0 & \text{muuten} \end{cases}$$

missä  $g_6$  on 6-grammi,  $n_{g_6}$  on sen esiintyvyys lähdeaineistossa

$$n_{g_6} := |\{l \in L_n^6 \mid g_6 \in l\}|$$

ja  $f_{g_6, w}$  on  $g_6$ :n esiintymien määrä  $w$ :ssä

$$f_{g_6, w} := |\{t \mid 1 \leq t \leq |w| - 6, w_{t,t+6} = g_6\}|.$$

Jaan esiintymien määrän niiden dokumenttien määrällä, joissa 6-grammi esiintyy ( $\frac{1}{n_{g_6}}$ ), sillä jotkut yksinkertaiset sävelkulut ovat hyvin yleisiä ja esiintyvät hyvin monessa kappaleessa. Teoksia ei tule rangaista liikaa siitä, että ne käyttävät tällaisia yleisiä sävelkulkuja. Kopiointiaste kerrotaan sadalla, jotta tulos on välillä 0–100.

Määritellään vastaavasti *rytmien samankaltaisuus*  $A_R = A_{L_r^6}(r_{1,n})$  suhteessa lähdeaineistoon  $L_r^6$ , joka sisältää nuottien pituuksien  $r_t = \text{pituus}(w_t)$  6-grammeja, sekä *intervallien samankaltaisuus*  $A_I = A_{L_i^6}(i_{1,n-1})$ , missä intervalli  $i_t = \text{korkeus}(w_{t+1}) - \text{korkeus}(w_t)$ . Intervallien eli korkeuserojen mittaaminen sävelten korkeuksien sijasta on hyödyllistä, sillä ihmiset mieltävät eri korkeudelta soitetun (transponoidun) kappaleen samaksi kuin alkuperäisen kappaleen, eli absoluuttisten sävelkorkeuksien sijasta intervallit määrittävät kappaletta.

Lasken rytmien ja intervallien samankaltaisuudet yhteen muodostaen *sumean kopiointiasteen*  $\frac{A_I + A_R}{2}$ . Tämä luku kertoo siis yksi-yhteen-kopioiden lisäksi tilanteista, joissa rytmi on sama, mutta sävelkorkeudet eroavat, sekä toisinpäin.

Arvioin sekä lähdeaineiston kappaleita että mallin tuottamia kappaleita näillä kriteereillä. Tällöin poistan ennen kopiointiasteen ja samankaltaisuuden laskemista lähdeaineistosta kaikki samannimiset teokset. Poistan myös ne teokset, joiden kopiointiaste on yli 50, sillä nämä ovat oletettavasti versioita toisista teoksista, jotka vääristävät jakaumaa. On odotettavissa, että monet ihmissävellykset ottavat vaikutteita toisistaan.

Musiikissa on joitakin hyvin yleisiä rytmejä ja sävelkulkuja, jotka esiintyvät monessa kappaleessa, minkä lisäksi säveltäjät tekevät joskus viittauksia toistensa kappaleisiin.

Valitsin 6-grammit, koska kuuden nuotin mittainen melodia on selvästi tunnistettavissa.

### 4.5.2 Tyypillisuus

Ritchie määrittelee tyypillisyyden kysymyksenä ”Missä määrin teos on artefaktiluokansa edustaja?” [33]. Hän antaa esimerkkinä vitsigeneraattorin, jonka teosten tyypillisyyttä mitataan kysymyksellä ”Onko tämä teos vitsi?” ja arvoa kysymyksellä ”Onko tämä teos hauska?”. Ritchie toteaa, että epätyypilliset mutta arvokkaat teokset saattavat kertoa suuremmasta luovuudesta kuin tyypilliset teokset, mutta luovan ohjelman pitäisi osata tuottaa myös tyypillisiä teoksia [32]. Ritchie mallintaa tyypillisyyttä luvulla välillä  $[0, 1]$ .

Säveltävän kielimallin tapauksessa artefaktiluokka on ”melodiat sointumerkintöineen”. Teokset eivät edusta mitään tiettyä genreä. Päätin tutkia tyypillisyyttä musiikin teorian näkökulmasta ja selvittää, noudattavatko kappaleet ”sääntöjä”. Kävin läpi kymmenen generoitua, valikoimatonta kappaletta läpi harrastajasäveltäjä Tuukka Rusin kanssa. Pyysin häntä huomauttamaan mistä tahansa epätyypillisyyksistä, outouksista tai musiikin teorian vastaisista kohdista. Tutkin tyypillisyyttä laadullisesti, sillä tätä pohdintaa on vaikea tiivistää yhteen lukuun etenkin, kun koehenkilöitä on vain yksi.

Tyypillisyyden arviointiin käytetyt kappaleet on esitetty liitteessä B.

### 4.5.3 Arvo

Arvioin teosten arvoa soittamalla koeyleisölle viisi satunnaista kappaletta lähdeaineistosta sekä viisi valikoimatonta generoitua kappaletta. Pyysin koeyleisön jäseniä antamaan jokaiselle teokselle arvosanan asteikolla 1–5. Koeyleisössäni oli kolme henkilöä. Yksi heistä harrasti säveltämistä ja soitti useita soittimia. Toinen soitti vain pianoa. Kolmas ei harrastanut musiikkia. Yleisölle kerrottiin etukäteen, että osa kappaleista on tietokoneen ja osa ihmisen. Kappaleet soitettiin satunnaisessa järjestyksessä FluidSynth-ohjelman pianosyntetisaattorilla. Jokaisen kappaleen soittamisen jälkeen kuuntelijoita pyydettiin kirjoittamaan kyseisen kappaleen järjestysnumero ja arvosana paperille.

Mittari	Lähdeaineisto	Generoitu otos
keskiarvo( $A_K$ )	6,30	6,17
mediaani( $A_K$ )	4,55	4,30
max( $A_K$ )	45,00*	63,61
keskiarvo( $\frac{A_I+A_R}{2}$ )	14,07	12,26
mediaani( $\frac{A_I+A_R}{2}$ )	13,34	11,31
max( $\frac{A_I+A_R}{2}$ )	55,74	29,17

\* Aineistosta on poistettu kappaleet, joiden kopiointiaste on yli 50.

**Taulukko 4.2:** Yhteenveto lähdeaineiston ( $n = 6049$ ) ja generoidun otoksen ( $n = 100$ ) kopiointiasteista  $A_K$  ja sekä sumeista kopiointiasteista  $\frac{A_I+A_R}{2}$ .

Viiden kappaleen arviointi kolmella henkilöllä on liian vähän, jotta tuloksia voisi pitää tarkkoina. Koska resurssini ovat rajatut ja koska tarkoitukseni on kuitenkin enemmän demonstroida kuinka kielimallia voisi arvioida kuin oikeasti arvioida kielimallia, en suorittanut laajempaa koekuuntelua.

## 4.6 Arvioinnin tulokset

### 4.6.1 Uutuus

Generoin kielimallilla 100 kappaletta, joille laskin sekä kopiointiasteen että sumean kopiointiasteen. Tulokset on esitetty taulukossa 4.2. Tulosten perusteella kielimallin tuottamat kappaleet muistuttavat lähdeaineiston kappaleita keskimäärin vähemmän kuin lähdeaineiston kappaleet muistuttavat toisiaan.

Generoiduista kappaleista 63 %:n kopiointiaste oli pienempi kuin lähdeaineiston keskimääräinen kopiointiaste sekä 70 %:n sumea kopiointiaste oli pienempi kuin lähdeaineiston keskimääräinen sumea kopiointiaste.

### 4.6.2 Tyypillisuus

Kaikki tutkitut kappaleet olivat Rusin mukaan musiikkikappaleita. Hän arvioi yhden kappaleen muistuttavan pop-kappaletta, yhden klassisen musiikin kappaletta. Muiden kappaleiden genre ei ollut hänen mukaansa selvä.

Kappaleet noudattivat pääasiallisesti musiikin teorian sääntöjä. Ne pysyivät pääosin sävellajissa ja sisälsivät mielekkäitä sointukulkuja. Lisäksi kappaleet sisälsivät tyypillisiä lopetuksia. Kappaleissa oli myös epätyypillisiä piirteitä. Nämä on lueteltu alla:

- sointuun sopimattoman sävelen käyttäminen
- sointukulun loppuminen kesken kappaleen lopussa
- epätavalliset rytmit
- epätavalliset soinnut
- useampi päällekkäinen sointumerkintä
- sävellajin vaihtuminen kesken kappaletta
- tahtilajin vaihtuminen kesken kappaletta
- sointujen liian säästeliäs käyttö
- Voltan hakasten merkitseminen väärin

Näistä osa on selitettävissä lähdeaineiston huonolla laadulla, sillä se sisältää paljon virheellisiä merkintöjä, kuten useita päällekkäisiä sointumerkkejä.

Kokonaisuudessaan kappaleet eivät olleet kovinkaan epätyypillisiä.

### 4.6.3 Arvo

Koekuuntelun tulokset on esitetty taulussa 4.3 sekä kuvassa 4.2. Ihmisten säveltämien kappaleiden arvosanojen keskiarvo oli 2,20 ja ohjelman säveltämien kappaleiden 2,73.

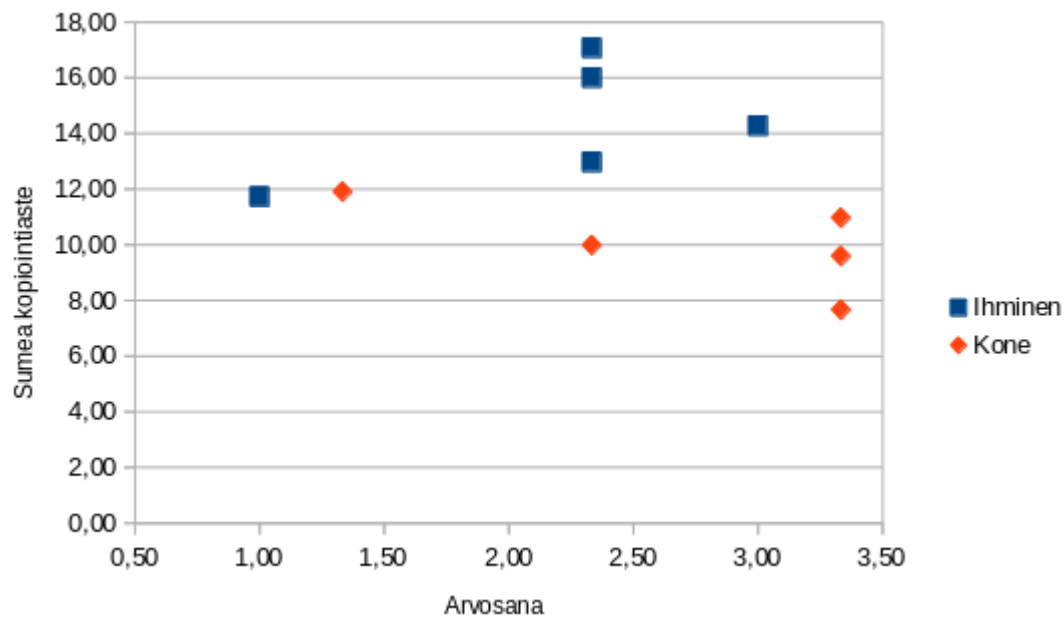
## 4.7 Tulosten tulkinta

Koska kielimallin teosten kopiointiaste ja sumea kopiointiaste ovat samalla tasolla tai pienempiä kuin lähdeaineistossa olevien teosten, ovat teokset näillä mittareilla mitattuna keskimäärin yhtä uusia kuin ihmissäveltäjienkin teokset.

Uutuuden mittaamiseen käytetty metriikka ei kuitenkaan ole paras mahdollinen, sillä se ottaa huomioon vain pinnallisia nuotti nuotilta kopioituja sävel- ja rytmijonoja eikä se mittaa korkeamman tason musiikillisia ideoita. Näiden arviointi on huomattavan

Kappale #	Tekijä	$A_K$	$\frac{A_I+A_R}{2}$	Arvosana
0	Ohjelma	16,67	9,99	2,33
1	Ohjelma	10,83	<b>7,67</b>	<b>3,33</b>
2	Ohjelma	6,25	11,91	1,33
3	Ohjelma	1,74	10,97	<b>3,33</b>
8	Ohjelma	2,48	9,59	<b>3,33</b>
13	Ihminen	7,20	17,07	2,33
14	Ihminen	11,45	14,27	3,00
15	Ihminen	<i>19,74</i>	11,73	<i>1,00</i>
16	Ihminen	10,83	<i>16,00</i>	2,33
17	Ihminen	<b>0,00</b>	12,97	2,33

**Taulukko 4.3:** Koekuuntelun ( $n = 3$ ) tulokset. Paras tulos on **tummennettu** ja huonoin *kursivoitu*.



**Kuva 4.2:** Koekuuntelun ( $n = 3$ ) tulokset. Pystyakselilla on sumea kopiointiaste  $\frac{A_I+A_R}{2}$  ja vaakakselilla arvosanojen keskiarvo.

vaikeaa ja tarvitsee musiikin teoriaan perehtyneitä ihmisarvioijia. Tuukka Rusin mielestä teokset eivät olleet erityisen epätyypillisiä, joten tämän perusteella teoksissa ei ollut selvästi havaittavia uusia ideoita, jotka kiinnittäisivät arvioijan huomion.

Koekuuntelun perusteella kielimallin säveltämät kappaleet ovat keskimäärin pidettympiä kuin ihmisen säveltämät kappaleet. Korkean arvosanan saaneiden kappaleiden kopiointiaste ei ole suurempi kuin ihmisten säveltämien kappaleiden.

Kokeen perusteella katson, että arvokkuuden kriteeri täyttyy selvästi ja uutuuden kriteeri täyttyy riittävästi, mutta teokset eivät ole niin uusia kuin ne voisivat olla. Kolmannen kriteerin eli ennustamattomuuden arviointi on haastavaa, mutta arvioin sen todennäköisesti täyttyvän. Näin ollen kielimallin voi sanoa olevan jossakin määrin luova.

Pienen otoskoon vuoksi tutkimus ei kuitenkaan ole kovin tarkka. Lisäksi vain yhden tyyppillisyyden arvioijan ja kolmen koekuuntelijan käyttäminen tekee arvioinnista epätarkkaa.



# 5 Pohdintaa

Tässä osassa käsittelen vastaamattomia kysymyksiä ja mahdollisia seuraavia tutkimusaiheita.

## 5.1 Arvokkuuden ja uutuuden määrittely

Luvussa 3.1.1 kuvailin arvokkuuden ja uutuuden käsitteitä, sekä niiden määrittelyn ongelmallisuutta. On selvää, että arvokkuus ja uutuus on määriteltävä tapauskohtaisesti: esimerkiksi novellin arvon mittaaminen on tehtävä väistämättä eri tavalla kuin musiikkikappaleen arvon mittaaminen.

Esittelin useita tapoja uutuuden mittaamiseen, jotka ottavat huomioon eri merkityksen tasoja. Eri tasojen uutuuden arviointi on vaikeaa, eikä ole selvää, minkä tasojen mittaaminen on luovuuden kannalta merkityksellisintä.

Mahdollinen uusi tutkimussuunta on luoda selkeät ohjeistukset arvon ja uutuuden määrittelmien valitsemiseen ja niiden mittaamiseen tekstilajikohtaisesti.

## 5.2 Ennustamattomuuden mittaaminen

Kolmas määrittelemäni kriteeri on kielimallin ennustamattomuus. Käytännössä, kuten luvussa 4.4 totesin, ennustamattomuuden arviointi formaalisti on vaikeaa. On jossakin mielessä mahdotonta todistaa ennustamattomuutta, koska on aina teoriassa mahdollista, että *joku* ymmärtää mallin toimintaa, vaikka sellaista ihmistä ei olisi vielä löytynyt.

Argumentti, että tarpeeksi suuria malleja on mahdotonta ymmärtää niiden koon vuoksi, ei myöskään ole täysin pitävä. Voisi olla, että malli on niin säännönmukainen ja sisältää helposti hahmotettavia osia, että ihminen voi ymmärtää sitä riittävällä tasolla, vaikka ei ymmärtäisikään kaikkia pieniä yksityiskohtia. Olisi tarpeen kehittää tapa mitata mallin koon lisäksi myös sen monimutkaisuutta. Tämä on mahdollisista tulevista tutkimussuunnista mielestäni tärkein.

### 5.3 Generoinnin aikaisten tavoitteiden syntyminen

Luvussa 3.3.1 kuvasin mahdollista menetelmää, jolla malli voi luoda omia tavoitteita ja muuttaa niitä kesken generoinnin. Tämän käytöksen aikaansaamiseksi lähdeaineiston pitäisi olla oikeanlainen ja sisältää tavoitemerkintöjä. Jos lähdeaineistossa on näitä merkintöjä, on epäselvää, missä määrin malli asettaa aidosti itselleen uusia tavoitteita, ja missä määrin se vain kopioi lähdeaineistossa esiintyviä tavoitemuutoksia. Eräs mahdollinen tutkimussuunta olisi selvittää, millainen lähdeaineiston tulisi olla ja miten tavoitteet tulisi merkitä, jotta malli oppisi niiden avulla luomaan omia tavoitteitaan.

### 5.4 Tietoisuus

Riippuen tulkinnasta täysi tavoitteellisuus saattaa vaatia tekoälyltä tietoisuutta. Linkola ym. [28] määrittelevät tietoisuuden (”self-awareness”) kyvyksi arvioida ja muuttaa omaa toimintaa. He jakavat tietoisuuden useisiin osa-alueisiin, joista tavoitteellisuuden kannalta oleellimmat ovat *teostietoisuus* ja *tavoitetietoisuus*. Teostietoinen tekoäly osaa arvioida tuottamiaan teoksia ja tuottaa havaintojensa pohjalta uudenlaisia teoksia. Tavoitetietoinen tekoäly osaa arvioida havaintojensa pohjalta omia tavoitteitaan ja tarvittaessa muuttaa niitä.

Ventura [37] pitää teostietoisuutta asteikkonsa viimeisenä askeleena kohti todellista luovuutta. Vaikka teostietoisuus ei ole edellytys sille, että tekoäly voi olla jossakin määrin luova, on se kuitenkin tärkeä ominaisuus edistyneemmän luovuuden saavuttamiseksi.

Jotta tekoäly voisi aidosti asettaa itselleen tavoitteita, tulisi sen olla tavoitetietoinen. Muussa tapauksessa tavoitteet ovat joko satunnaisia tai jonkun muun asettamia. Kuten luvussa 3.3.2 totesin, tämän määrittelemine on hankalaa, sillä tekoälyn itse asettamien tavoitteiden on pohjaututtava korkeamman tason tavoitteisiin, joista lopulta jonkun on oltava tekoälyn luojan asettama.

Koska tietoisuuteen liittyy määrittelyongelmia, on tärkeä uusi tutkimussuunta määrittelmien täydentäminen.

Jotta kielimalli voisi olla aidosti tietoinen, pitäisi sen pystyä muuttamaan omaa toimintaansa. Tämä ei ole kielimallille staattisena oliona mahdollista, joten olisi kiinnostavaa tutkia, minkälainen ohjelma tarvittaisiin vähintään kielimallin oheen.

## 5.5 Johtopäätösten yleistyminen

Käsittelen tässä tutkielmassa aihetta erityisesti kielimallien näkökulmasta, mutta monet esittämäni argumentit yleistyvät todennäköisesti myös muihin järjestelmiin. Osa argumenteista saattaa yleistyä vain muihin koneoppimisalgoritmeihin ja osa ennustamattomiin järjestelmiin ylipäätensä. Seuraava tutkimusaihe voisi olla argumenttien jäsenteleminen niin, että pelkästään kielimalleja koskevat argumentit erotetaan yleisemmin pätevistä argumenteista, ja jälkimmäisten argumenttien pätemisen laajuuden selvittäminen.



## 6 Johtopäätökset

Tavoitteeni oli muotoilla selkeä menetelmä kielimallien luovuuden arviointiin, jossa tietyt kriteerit täyttävä kielimalli tekee varmasti tietynlaisia luovia tekoja. Kriteereinä käytin teosten arvokkuutta ja uutuutta sekä kielimallin ennustamattomuutta, joita kuvaan luvussa 3.1. Tarkoitukseni on, että ennustamattomuutta lukuunottamatta pelkkä teosten arviointi on riittävää kielimallien luovuuden tutkimiseksi.

Arvioin luvussa 3.2 kriteerit täyttävän kielimallin tekemiä luovia tekoja. Kielimalli tekee FACE-mallin mukaiset luovat teot  $C^g$ ,  $A^g$  ja  $E^g$ .

Osassa 4 esittelen kouluttamani säveltävän kielimallin ja arvioin sen teoksia Ritchien menetelmän avulla. Pohdin, täyttävätkö sen luomat teokset arvokkuuden ja uutuuden ehdot sekä arvioin kielimallin tekemiä luovia tekoja.

Kuvaan osassa 5 menetelmäni ongelmia ja ratkaisemattomia kysymyksiä.

Eräs menetelmän ongelma on, ettei kriteeristöön ei ole sisällytetty tavoitteellisuutta, jota pidetään arvokkuuden ja uutuuden ohella luovuuden ehtona [37]. Käsittelin aiheita luvussa 3.3. Arviointini mukaan kielimalleilla voi olla generoinnin aikana generointia ohjaavia tavoitteita, jotka voivat myös muuttua generoinnin edetessä. Lisäksi kielimalleilla on monia ulkoa asetettuja tavoitteita. Mallin luovuutta rajoittaa kuitenkin se, ettei se ole ”tietoinen”: malli ymmärrä tavoitteidensa vaikutuksia tosimaailmassa tai pysty arvioimaan tekojensa kontekstia tai seurauksia.

Toinen ongelma on, että kriteerejä ei ole tarkkaan määritelty. Uutuuden ja arvon käsitteitä tulee soveltaa tapauskohtaisesti, eikä ole selvää, kuinka ”tiukka” määritelmä tulisi valita: riittääkö uutuuden arvioimiseksi esimerkiksi editointietäisyyden mittaaminen, vai vaatiiko se merkitysten ja ideoiden tutkimista. Lisäksi kolmannen kriteerin eli ennustamattomuuden arviointi täsmällisesti on vaikeaa.

Johtopäätökseni on, että menetelmäni soveltuu kielimallin tekemien luovien tekojen määrittämiseen ja sitä voi käyttää osana kielimallien luovuuden arviointia. Käytännössä mallin käyttämien konseptien sekä mallin tavoitteiden arviointi on kuitenkin haastavaa.



# Kiitokset

Käytin työssäni Helsingin yliopiston Ukko 3 -laskentaklusteria, mistä kiitän FCCI-konsortiota (Finnish Computing Competence Infrastructure). Lisäksi haluan kiittää Tuukka Rusia kärsivällisyydestä ja vastauksista musiikin teoriaan liittyviin kysymyksiini.



# Kirjallisuus

- [1] Y. Agafonova, A. Tikhonov ja I. P. Yamshchikov. "Paranoid transformer: Reading narrative of madness as computational approach to creativity". Teoksessa: *Proceedings of the 11th International Conference on Computational Creativity*. Coimbra, Portugal, 2020.
- [2] P. Ammanabrolu, W. Broniec, A. Mueller, J. Paul ja M. O. Riedl. "Toward automated quest generation in text-adventure games". Teoksessa: *Proceedings of the 11th International Conference on Computational Creativity*. Coimbra, Portugal, 2020.
- [3] J. Austin. "The book of endless history: Authorial use of GPT2 for interactive storytelling". Teoksessa: *International Conference on Interactive Digital Storytelling*. Springer. 2019, s. 429–432.
- [4] S. Berns, T. Broad, C. Guckelsberger ja S. Colton. "Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities". Teoksessa: *Proceedings of the 12th International Conference on Computational Creativity*. Mexico City, Mexico, 2021.
- [5] M. A. Boden. *The creative mind: Myths and mechanisms*. Routledge, 2004.
- [6] S. Bringsjord, P. Bello ja D. Ferrucci. "Creativity, the Turing test, and the (better) Lovelace test". Teoksessa: *The Turing Test*. Springer, 2003, s. 215–239.
- [7] T. Broad, S. Berns, S. Colton ja M. Grierson. "Active Divergence with Generative Deep Learning—A Survey and Taxonomy". Teoksessa: *Proceedings of the 12th International Conference on Computational Creativity*. Mexico City, Mexico, 2021.
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al. "Language models are few-shot learners". *Advances in neural information processing systems* 33 (2020), s. 1877–1901.
- [9] E. Charniak. *Statistical language learning*. MIT press, 1996.
- [10] K. Choi, G. Fazekas ja M. Sandler. "Text-based LSTM networks for automatic music composition". *arXiv preprint arXiv:1604.05358* (2016).

- [11] S. Colton. "Creativity Versus the Perception of Creativity in Computational Systems." Teoksessa: *AAAI spring symposium: creative intelligent systems*. Vol. 8. 2008.
- [12] S. Colton, J. W. Charnley ja A. Pease. "Computational Creativity Theory: The FACE and IDEA Descriptive Models." Teoksessa: *Proceedings of the Second International Conference on Computational Creativity*. Mexico City, Mexico, 2011, s. 90–95.
- [13] S. Colton, A. Pease, J. Corneli, M. Cook ja T. Llano. "Assessing Progress in Building Autonomously Creative Systems." Teoksessa: *Proceedings of the Fifth International Conference on Computational Creativity*. Ljubljana, Slovenia, 2014, s. 137–145.
- [14] S. Colton, G. A. Wiggins et al. "Computational creativity: The final frontier?" Teoksessa: *Proceedings of the 20th European Conference on Artificial Intelligence*. Vol. 12. Montpellier, France, 2012, s. 21–26.
- [15] M. S. Cuthbert ja C. Ariza. "music21: A toolkit for computer-aided musicology and symbolic music data" (2010).
- [16] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le ja R. Salakhutdinov. "Transformer-xl: Attentive language models beyond a fixed-length context". *arXiv preprint arXiv:1901.02860* (2019).
- [17] D. Eck ja J. Schmidhuber. "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks". Teoksessa: *Proceedings of the 12th IEEE workshop on neural networks for signal processing*. IEEE. 2002, s. 747–756.
- [18] J. Fidler ja Y. Goldberg. "Controlling linguistic style aspects in neural language generation". *arXiv preprint arXiv:1707.02633* (2017).
- [19] L. Floridi ja M. Chiriatti. "GPT-3: Its nature, scope, limits, and consequences". *Minds and Machines* 30.4 (2020), s. 681–694.
- [20] P. Glines, B. Biggs ja P. M. Bodily. "A Leap of Creativity: From Systems that Generalize to Systems that Filter." Teoksessa: *Proceedings of the 11th International Conference on Computational Creativity*. Coimbra, Portugal, 2020, s. 297–302.
- [21] A. Graves. "Generating sequences with recurrent neural networks". *arXiv preprint arXiv:1308.0850* (2013).

- [22] C. Guckelsberger, C. Salge ja S. Colton. "Addressing the "Why?" in Computational Creativity: A Non-Anthropocentric, Minimal Model of Intentional Creative Agency". Teoksessa: *Proceedings of the Eight International Conference on Computational Creativity*. Atlanta, USA, 2017.
- [23] D. R. Hartree. *Calculating instruments and machines*. New York, 1949.
- [24] I. Hauhio. "Koneluovuuden arviointi". Kandidaatintutkielma. Helsingin yliopisto, 2019.
- [25] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu ja D. Eck. "Music transformer". *arXiv preprint arXiv:1809.04281* (2018).
- [26] A. Jordanous. "A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative". *Cognitive Computation* 4.3 (2012), s. 246–279.
- [27] A. Jordanous. "Four PPPerspectives on Computational Creativity in theory and in practice". *Connection Science* 28.2 (2016), s. 194–216.
- [28] S. Linkola, A. Kantosalu, T. Männistö ja H. Toivonen. "Aspects of Self-awareness: An Anatomy of Metacreative Systems". Teoksessa: *Proceedings of the 8th International Conference on Computational Creativity*. 2017, s. 189–196.
- [29] A. Lovelace. "Translator's Notes to M. Menabrea's Memoir". Teoksessa: *Babbage's Calculating Engines*. London: E. & F. N. Spon, 125, Strand, 1889, s. 21–50.
- [30] C. Peterson. "Generating rhyming poetry using LSTM recurrent neural networks". Tutkielma. Victoria, Canada: University of Victoria, 2019.
- [31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al. "Language models are unsupervised multitask learners". *OpenAI blog* 1.8 (2019). Viitattu: 16.6.2022, s. 9. URL: <https://openai.com/blog/better-language-models/>.
- [32] G. Ritchie. "Assessing creativity". Teoksessa: *Proc. of AISB'01 Symposium*. Citeseer. 2001.
- [33] G. Ritchie. "Some empirical criteria for attributing creativity to a computer program". *Minds and Machines* 17.1 (2007), s. 67–99.
- [34] M. A. Runco ja G. J. Jaeger. "The standard definition of creativity". *Creativity research journal* 24.1 (2012), s. 92–96.

- [35] A. Tikhonov ja I. P. Yamshchikov. "Guess who? Multilingual approach for the automated generation of author-stylized poetry". Teoksessa: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, s. 787–794.
- [36] A. M. Turing. "Computing machinery and intelligence". Teoksessa: *Minds and Machines*. Englewood Cliffs, NJ: Prentice-Hall, 1964, s. 4–30.
- [37] D. Ventura. "Mere generation: Essential barometer or dated concept". Teoksessa: *Proceedings of the Seventh International Conference on Computational Creativity*. Paris, France, 2016, s. 17–24.
- [38] G. A. Wiggins. "A framework for description, analysis and comparison of creative systems". Teoksessa: *Computational Creativity*. Springer, 2019, s. 21–47.
- [39] X. Wu, C. Wang ja Q. Lei. "Transformer-XL based music generation with multiple sequences of time-valued notes". *arXiv preprint arXiv:2007.07244* (2020).
- [40] H. Zhang ja Z. Zhang. "Automatic Generation Method of Ancient Poetry Based on LSTM". Teoksessa: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE. 2020, s. 95–99.

## Liite A Tunnehaikuja

Seuraava teksti on generoitu GPT-3:lla. Kursivoidut osat ovat ihmisen antama kehote. Tekstin on tarkoitus osoittaa, kuinka kielimallin voi ohjata päättämään itse, millaisia teoksia se haluaa tehdä. Ensimmäistä runoa lukuunottamatta malli on itse valinnut runojen tunteet. Esimerkin tarkoitus ei ole ottaa kantaa GPT-3:n luovuuteen.

### *10 Haikus with emotions*

#### *1. Sad Haiku:*

Lonely, I sit  
Alone in my room  
With only my thoughts

#### *2. Angry Haiku:*

Why does he do that?  
It's always the same thing  
I don't deserve this

#### *3. Loving Haiku:*

I love you so much  
I want to be with you always  
I'm so happy

#### *4. Hateful Haiku:*

I hate you so much  
I wish you would die  
I can't stand to see you

#### *5. Fearful Haiku:*

I'm so afraid  
What if something happens to you?

I don't know what I'd do

6. Grateful Haiku:

Thank you for being here  
Thank you for being my friend  
Thank you for everything

7. Hopeful Haiku:

I hope that one day  
We can be together again  
I miss you so much

8. Surprised Haiku:

I never expected  
That you would feel that way too  
I'm so happy

9. Confused Haiku:

I don't understand  
Why you would say that to me  
I don't know what to do

10. Ecstatic Haiku:

I'm so happy  
I can't believe this is happening  
I'm so in love with you

## **Liite B Säveltävän ohjelman generoimia kappaleita**

Seuraavilla sivuilla on säveltävän ohjelman tyypillisyyden arviointiin käytetyt kappaleet, joihin Tuukka Rusi on tehnyt merkintöjä.



# Kappale 1

musikorpus3-18

Musical notation for measures 1-6. Chords: C# (triple), Gdim C#6, G#7 (triple), C# (triple), C#dim C#6. Tempo: ♩ = 160.

Musical notation for measures 7-12. Chords: Ebm7 (triple), G#7, G#7 (circled), Ebm7 (circled), C# (triple), C#dim C#, G#7 (triple).

Musical notation for measures 13-19. Chords: C# (triple), C#dim C#6, G#7 (triple), C# (triple), C#dim C#, G#7 (triple).

Musical notation for measures 20-27. Chords: C# (triple), C#dim C#, F#7 (triple), F# (triple), F#m6, C# (triple), Bbm.

Musical notation for measures 28-34. Chords: Eb7 (triple), G#7, C#dim (circled), C# (circled), Ebm7, G#aug G#7, 2 C# (boxed). Handwritten note: *tähän sopi vain alampi*.

# Kappale 2

musikorpus3-18

1 2  
G# G#

G# Fm Bbm G# G#

$\text{♩} = 160$

6 C# Eb G# Fm Bbm Eb Ab

# Kappale 3

musikorpus3-18

F F7 B $\flat$  G7 C7 F C7 F

♩ = 160

9 C7 F F7 B $\flat$  G7 C7 F F G

*Osa vaihtuu,  
mutta jaa tää  
irrollisiksi*

18 C7 F G7 C7 F

27 Dm G7 C7 F Dm G7 C7 F

# Kappale 4

musikopus3-18

C Dm7 G7 C Am Dm G Dm G7 C

$\text{♩} = 160$

*outo valinta*

10 Dm7 G7 C Am Dm G Dm G7 → Bbm Fm Bbm7

20 Eb7 Ab Bbm7 Eb7 Ab Dm7 G7 C Am Dm

28 G Dm 1G7 C Dm7 G7 2 C

# Kappale 5

musikorpus3-18

Handwritten annotations in red ink:

- outo sointuväline / tähän astelkseen / kontakettiin
- tässä parempi
- nieluummin joku soinnun väli

Chord progression: G, D7, G, Em, F, D7, G, G, F, D7, G, Em, Am, D7, G, Em, Am, C, C/D, D7, G.

Tempo: ♩ = 160

Measure numbers: 10, 20, 30, 39.

# Kappale 6

musikorpus3-18

G Gmaj7 G6 G C D7sus4 D7 C

♩ = 160

8 Dsus4 D7 G Gmaj7 G6 G C D7sus4 D7 C

16 Dsus4 D7 G Gmaj7 Cmaj7 D7 Am7D7 G Cmaj7 C#dim

25 <sup>G/H</sup> Gm/Bb D7 D D7 G G Cmaj7 C#dimD Bm Gaug

35 C A7 D7 G Cmaj7 C#dimGm/D <sup>G/O</sup> D7 D D7 G

46 G7 C C#dim G/D D7 G/B G#dim <sup>Am7</sup> D7 G

# Kappale 7

musikorpus3-18

D A7 F#m Bm D B7 E7 A7 A18

$\text{♩} = 160$

tähän nuotti

6 Em F#m7 Em7 A7 F#m7 Bm7 C#m7 F#7 Am7 Bm7

tähän nuotti

15 C#m7 F#7 Bm7 F#7 Bm7 Bm7 E7

3

ei triolia

23 A Dm A

# Kappale 8

musikopus3-18

$\text{♩} = 80$

Chord symbols:  $Bb$ ,  $Bb/Eb$ ,  $Bb\text{add}2$ ,  $Bb\text{maj}7/D$ ,  $Gm$ ,  $Eb$ ,  $Dm$ ,  $F$ ,  $Bb/D$ ,  $F/Eb$ ,  $F$ ,  $Bb/D$ ,  $Eb$ ,  $Bb$ ,  $C$ ,  $F$ ,  $F$ ,  $Bb/D$ ,  $Eb$ ,  $Bb$ ,  $C$ ,  $F$ ,  $F$ ,  $Cm$ ,  $Eb$ ,  $Bb$ ,  $C$ ,  $Bb$ ,  $F$ ,  $Cm$ ,  $Bb$ ,  $F$ ,  $Cm$ ,  $Bb$ ,  $F$ ,  $Bb$ ,  $Cm$ ,  $Dm$ ,  $Cm$ ,  $Bb$ ,  $Cm$ ,  $Dm$ ,  $F$ ,  $Bb$ ,  $Cm$ ,  $Dm$ ,  $F$ ,  $Bb$ ,  $Cm$ ,  $Dm$ ,  $F$ ,  $Bb$ ,  $Eb$ ,  $Bb$ ,  $Eb$ ,  $Bb$ ,  $Eb$ ,  $Bb$ ,  $Eb$ ,  $Bb$ .

*epäsäännöllisellä rytmi tähän asti*

2

43

E $\flat$

B $\flat$

E $\flat$ ?

~~1~~

B $\flat$

2

B $\flat$



# Kappale 9

musikorpus3-18

*Lisää sointuja!  
Eräs ehdotus:*

A Bm E A E A E A A7

♩ = 160

11 D A Bm E A E D A E

21 A E A E A A7 D A Bm E

31 A E A E D A A E A A7

41 D A