Irene Venäläinen

Kirjain-äännevastaavuuden oppimisen mallinnus bayesilaisella menetelmällä

Tietotekniikan pro gradu -tutkielma

19. marraskuuta 2019

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Irene Venäläinen

Yhteystiedot: irene.venalainen@iki.fi

Ohjaaja: Raino Mäkinen, Janne V. Kujala

Työn nimi: Kirjain-äännevastaavuuden oppimisen mallinnus bayesilaisella menetelmällä

Title in English: Bayesian model for learning letter-sound correspondences

Työ: Pro gradu -tutkielma

Suuntautumisvaihtoehto: Tieteellinen laskenta

Sivumäärä: 65+0

Tiivistelmä: Tämä tutkielma kuvaa erään tavan mallintaa kirjain-äännevastaavuuksien oppimista. Malli on luotu Ekapeliä varten käyttäen apuna pelistä kerättyä dataa. Mallin toteutuksessa käytettiin bayesilaisen tilastotieteen menetelmiä. Tavoitteena oli käyttää mallia uuden adaptaation luomiseen. Malli ei kuitenkaan sopinut suoraan pelin adaptaatiossa käytettäväksi laskennallisista ongelmista johtuen. Mallin avulla haluttiin myös visualisoida pelaajan osaamista ja kuvaajien avulla voidaankin helposti näyttää kokonaiskuva kirjainten osaamisesta.

Avainsanat: Ekapeli, adaptaatio, bayesilainen tilastotiede, Markovin ketju Monte Carlo, Markovin piilomalli, Gibbs-otanta

Abstract: This thesis describes a bayesian model for learning letter-sound correspondences. The model was created for Ekapeli using data from the game. The model was created using bayesian methods. Purpose of the model was to create a new adaptation for Ekapeli. Because of high computational time, the model doesn't suite for an adaptation without simplifications. Another goal for the model was to help visualize the player's learning. The model suited well for visualizing the player's knowledge of the letter-sound correspondences.

Keywords: Ekapeli, adaptation, Bayesian statistics, Hidden Markov model, Markov

Chain Monte Carlo, Gibbs sampling

Termiluettelo

Suomi	Englanti	Määritelmä
Adaptaatio	Adaptation	Sopeutumistapa. Pelin sovittautuminen pelaanan osaamisen mukaan. Liite
Assosiaatio	Association	Kirjoitetun ja puhutun kielen väliset yhteydet, joita Ekapelissä harjoitellaan. Liite
Gibbs-otanta	Gibbs sample	Otantamenetelmä, jolla todennäköisyysjakauman otoksen seuraava arvo valitaan.
Häiriöärsyke, distraktori	Distracter, Distractor	Pelaajalle esitetty kirjainvaihtoehto, joka ei vastaa kuultua äännettä. Liite
Häiriöarvo	Lapserate	Todennäköisyys, jolla pelaaja häiriintyy tai vahingossa vastaa satunnaisesti.
Kenttä	Level	Trialeista muodostuva kokonaisuus. Ennaltamäärätty tai esitettyjen trialien tai peliajan mukaan määräytyvä. Liite
Kohdeärsyke	Target stimulus	Pelaajalle esitetty kirjainvaihtoehto, joka vastaa kuultua äännettä. Liite
Markovin ketju Monte Carlo	Markov chain Monte carlo	Mallinnusmenetelmä, jota käytetään laskemaan otos todennäköisyysjakaumasta. Lyhenne MCMC.
Markovin piilomalli	Hidden Markov model	Mallinnusmenetelmä, jota käytetään, kun tiedetään tulos, mutta ei tilaa.
Pelikerta	Game session	Pelissä määritelty kokoelma kenttiä, jotka pelaaja suorittaa kerralla. Uudeksi pelikerraksi lasketaan, kun edellisestä pelikerrasta on kulunut yli 15 minuuttia. Liite

Termiluettelo

Suomi	Englanti	Määritelmä
Priori, Priorijakauma	Prior, Prior distribution	Alkuperäistodennäköisyys, todennäköisyysjakauma ennen kuin dataa on havaittu, $P(A)$.
Posteriori, Posteriorijakauma	Posterior, Posterior distribution	Todennäköisyysjakauma A ehdolla B $P(A\|B)$.
Triali	Trial	Yksi pelikenttä, jossa pelaajalle esitetään äänne ja kirjainvaihtoehtoja, joista pelaajan tulee valita äännettä vastaava kirjain. Liite
Ärsyke	Stimulus	Pelissä esitettävä äänne tai kirjain, joka esiintyy kohdeärsykkeenä tai häiriöärsykkeenä. Liite

Kuviot

Taulukot

Sisältö

1 Johdanto

Adaptoituva oppimispeli muokkautuu pelaajan vastausten perusteella. Adaptoituvan pelin avulla saadaan pidettyä pelaaja motivoituneena säätämällä peli siten, ettei se ole pelaajalle liian helppo tai liian vaikea. Jotta peli osaisi valita pelaajalle sopivan sisällön, tulee pelin sisältää tieto siitä mitä pelaaja osaa ja miten pelaaja oppii. Ihanteellisessa tilanteessa peli pystyy pelaajalle sopivan haasteellisena, jolloin pelaaja on motivoitunut ja myös pelaa mielellään (Ronimus ym. 2014). Tässä tutkielmassa kuvataan eräs tapa mallintaa pelaajan osaamista ja oppimista. Mitä tarkemmin peli tietää pelaajan osaamisesta, sitä paremmin esitettävät sisällöt voidaan valita.

Pelin saama tieto pelaajasta perustuu yleiseen tietoon opittavasta aihepiiristä ja pelin aikana saatavaan informaatioon. Voidaan esimerkiksi tehdä ennakko-oletus, että pelaaja vastaa osaamisensa mukaan tai täysin satunnaisesti. Pelin aikana saatava tieto koostuu mm. pelaajan vastauksista ja vastaamiseen kuluneesta ajasta. Jotta peli osaisi näyttää pelaajalle sopivaa sisältöä, täytyy sen jollain tapaa päätellä näiden ennakkotietojen ja vastausten perusteella millaisia sisältöä pelaajalle tulee näyttää seuraavaksi. Peli voidaan luoda esimerkiksi sellaiseksi, että se tuottaa mahdollisimman paljon tietoa pelaajan osaamisesta.

Ekapeli on lukemaanoppimispeli, jonka avulla pelaaja pystyy harjoittelemaan aakkosia ja tavuja. Pelin tavoitteena on, että pelaaja oppisi tunnistamaan kirjaimet ja niitä vastaavat äänteet.

Adaptaation luonti vaatii johtopäätösten tekemistä pelaajan vastauksista. Kuinka monta kertaa pelaajan tulee vastata oikein, että voidaan sanoa pelaajan osaavan tietyn kirjaimen? Jos pelaaja vastaa oikein kysyttäessä kirjainta P, paitsi silloin, kun kohdekirjain P esitetään kirjaimen B kanssa, voidaanko sanoa, että pelaaja osaa kirjaimen P?

Tukielman tavoitteena oli suunnitella ja toteuttaa uudenlainen malli pelaajan oppimisesta ja osaamisesta Ekapeliä varten ja tutkia kuinka hyvin malli sopii pelistä saatuun pelaajadataan. Erityisesti haluttiin tietää kuinka tarkkaan voidaan sanoa pelaajan osaavan yksittäisen kirjaimen pelkästään pelaajan vastausten perusteella.

Koska pelistä saatu data on kaikki mitä pelaajan osaamisesta tiedetään, haluttiin luoda malli, joka kuvaisi mahdollisimman tarkkaan pelaajan osaamista eri pelikerroilla. Bayesilainen lähestymistapa sopii tällaiseen tarkasteluun, sillä sen avulla voidaan approksimoida tuntematonta, eli pelaajan todellista osaamista.

Mallin avulla haluttiin myös visuaalisesti hahmottaa pelaajan osaamista. Kuvaajien avulla nähdään helpommin kuin pelkkiä vastauksia katsomalla mitkä kirjaimet pelaaja osaa ja mitkä tarvitsevat vielä harjoitusta. Lisäksi mallin avulla pyrittiin tutkimaan voisiko pelissä käytettyä adaptaatiota muokata entistä paremmin oppimista tukevaksi.

Luvussa 2 kuvataan lukemaanoppimispelin Ekapelin toimintaa ja tavoitteita sekä eri peliversioita. Tutkimussuunnitelma kuvataan luvussa 3. Tutkimuksessa käytetyt menetelmät esitellään luvussa 4. Oppimismallin luonnissa käytetään tilastollisia menetelmiä kuten bayesilaista tilastotiedettä 4.1, Markovin ketju Monte Carlo-menetelmää 4.5 ja forward-backward-algoritmia 4.7. Peliä varten luotu oppimismalli ja sen toteutus esitellään luvussa 5 ja mallin mukaisten kirjainten osaamistodennäköisyyksien laskenta luvussa 6. Kappaleessa 6.6 esitellään mallin pohjalta piirretyt kuvaajat. Luku 7 käsittelee mallin perusteella tehtyjen laskelmien sopivuutta pelaajadataan ja luku 8 kuinka mallin pohjalta voisi toteuttaa adaptaation peliä varten.

2 Ekapeli

Ekapeli on Jyväskylän yliopiston ja Niilo Mäki Instituutin kehittämä adaptiivinen oppimispeli, jonka tavoitteena on auttaa lapsia lukemaan oppimisessa. Ekapeli pyrkii pelin avulla auttamaan erityisesti lukemisvaikeuksista kärsiviä lapsia hahmottamaan kirjainten ja äänteiden välisen yhteyden (Lyytinen ym. 2007).

2.1 Ekapelin pelaaminen

Peli koostuu useista kentistä, jotka koostuvat tehtävistä, joissa pelaajan tulee vastata kuulemaansa kirjainta vastaava kirjain. Tehtävässä ruudun ylälaidasta tippuu palloja, joista jokaisessa on eri kirjain. Samalla pelaaja kuulee äänteen, joka vastaa yhtä ruudulla esiintyvistä kirjaimista. Pelaajan tulee valita palloista se, jossa on hänen kuulemaansa äännettä vastaava kirjain. Yhtä tällaisia tehtävää kutsutaan trialiksi. Trialissa kysyttävää kirjainta kutsutaan kohdeärsykkeeksi ja muita kirjaimia häiriöärsykkeiksi. Kenttien ja trialien rakennetta on kuvattu liitteen Ekapelisanasto kuviossa 2.

Pelin aluksi trialeissa on kaksi vaihtoehtoa: kohdeärsyke ja häiriöärsyke, kuvio 1. Jos pelaaja vastaa trialiin oikein, häiriöärsykkeiden määrä ja pallojen tippumisnopeus kasvavat, kuva 2. Pelaajan vastatessa väärin vastausvaihtoehtojen määrä vähenee ja niiden tippumisvauhti hidastuu. Lisäksi, jos pelaaja vastaa trialiin väärin, samaa kohdekirjainta kysytään uudestaan ja trialia tehdään helpommaksi siten, että kysytty kohde korostetaan.

Ekapelistä on tehty useita eri versioita, jotka esitellään kappaleessa 2.2. Joissain peliversioissa väärän vastauksen jälkeen ei esitetä uutta trialia, vaan kysytty äänne toistetaan ja muut vaihtoehdot poistetaan, jolloin pelaaja pakotetaan valitsemaan kysytty kohdekirjain. Esitettävien trialien määrä yhdellä pelikerralla riippuu peliversiosta ja käytetystä adaptaatiosta. Pelin edetessä harjoiteltavat kohteet ja ärsykkeet muuttuvat Ekapeli Ykkösessä kirjaimista tavuiksi ja sujuvuuspelissä tavuista sanoiksi.

2.2 Ekapelin adaptaatioiden historiaa

Ekapelistä on tehty useita eri versioita eri ikäisille ja taustaisille lapsille. Lukimat-palvelun verkkosivuilla ("Lukimat-palvelu" 2019) on kuvattu saatavilla olevat ja vanhat peliversiot ja eri peliversioiden sisällöt.

Ekapelin 'klassisessa' versiossa Ekapeli-Lukeminen kirjainten esitysjärjestys oli vakio ja se vastasi suomenkielisten aapisten kirjainten esittämisjärjestystä. Uudempien Ekapeli-Eskarin, Ekapeli-Yhden ja Ekapeli-Sujuvuuden sisällöt ovat etukäteen valittuja. Ekapeli-Eskari sisältää ainoastaan kirjain-äänne-vastaavuustehtäviä, Ekapeli-Yksi sisältää lisäksi tavu- ja sanatehtäviä ja Sujuvuus tavu- ja sanatehtäviä. Ekapeli-Sujuvuus on suunnattu erityisesti lukemisen sujuvuuden harjoittamiseen. Uusin versio Ekapelistä on Ekapeli-Alku, joka on tarkoitettu esikoululaisille ja koululaisille. Ekapeli-Alussa on kirjain-äänne-tehtäviä, tavutehtäviä ja sanatehtäviä.

Ekapeli-Lukeminen, jossa uusien kirjainten esitysjärjestys on vakio, perustuu tasoadaptaatioon. Kun pelaaja vastaa trialiin oikein, peli lisää opetusjärjestyksestä seuraavan kirjaimen kirjainlistaan, josta seuraava triali arvotaan. Kun pelaaja vastaa väärin, vähennetään kirjainlistaan viimeksi lisätty kirjain. Kun pelaaja on vastannut kolme kertaa oikein tiettyä kirjainta kysyttäessä, peli olettaa, että kirjain on opittu ja kirjain poistetaan pelikenttään arvottavien kirjainten listasta. Peli loppuu, kun kaikki kirjaimet ovat siirtyneet pois arvottavien kirjainten listalta.

Ekapeli-Eskari perustui ennalta valittuihin sisältöihin. Trialin sisältö valitaan ennaltamäärätyistä kirjainlistoista, jotka vaihtelevat sen mukaan kuinka pitkälle pelaaja on pelissä edennyt.

Ekapeli-Yksi toimi kuten Ekapeli-Eskari, mutta kirjainlistat, joista trialit valitaan ovat laajempia. Ekapeli-Yksi sisältää kirjain-äänne-tehtävien lisäksi tavu- ja sanaharjoituksia.

Ekapelistä on tehty lisäksi versio Ekapeli-Maahanmuuttaja maahanmuuttajataustaisille lapsille. Pelin sisältö on mukautettu pelaajan äidinkielen mukaan siten, että pelissä harjoitellaan erityisesti sen kielisille vaikeita suomen kielen kirjain- ja sanatehtäviä.

2.3 Ekapelin tutkimus

Ekapelistä on tehty tutkimusta Jyväskylän yliopiston ja Niilo Mäki Instituutin yhteisessä Lukimat-projektissa. Pelistä on tehty lisäksi GraphoGame-nimellä useita kansainvälisiä versioita, joiden avulla on tutkittu muiden kielten kuten englannin, kiinan ja ranskan oppimista (Richardson ja Lyytinen 2014).

2.4 Pelitulosten esittäminen

Pelituloksia voidaan tarkastella erilaisilla menetelmillä, joista tässä esitellään testitulostaulukot, tulostaulukot ja viuhkakuvaajat. Esitysmenetelmät on kuvattu Lukimat-palvelun verkkosivulla ("Lukimat-palvelu" 2019). Viuhkakuvaajia ja niiden taustaa kuvataan tarkemmin artikkelissa Kujala, Richardson ja Lyytinen (2010b).

Testitulostaulukot ja tulostaulukot kuvaavat pelaajan vastauksia taulukkomuodossa. Testitulostaulukoissa, kuvio 3, näytetään onko pelaaja vastannut arviointikenttiin oikein (O) vai väärin (V). Testitulostaulukot olivat käytössä Ekapeli-Eskarissa.

Tulostaulukot, kuvio 4, näyttävät kunkin kirjaimen osalta kuinka monesti kirjain on pelin aikana esitetty pelaajalle. Taulukoissa näytetään lisäksi kirjainkohtaiset ensimmäisten ja viimeisten 7 trialin oikeinvastausprosentit sekä kaikkien trialien oikeinvastausprosentit. Lisäksi esitetään nuolella onko pelaajan kirjaimen osaaminen parantunut, pysynyt samana vai huonontunut. Tulostaulukoita käytettiin pelaajien osaamisen seurannassa pelissä Ekapeli-Yksi.

Viuhkakuvaajassa, kuvio 5, esitetään yksittäisen kirjaimen erottamista toisista kirjaimista. Kirjain, jota halutaan tarkastella, on kuvaajan keskellä ja muut kirjaimet esitetään viuhkoina keskikirjaimen ympärillä. Viuhkakuvaajassa esitetty data kuvaa trialeita, joissa viuhkan keskellä oleva kirjain ja viuhkan kirjain ovat esiintyneet yhtä aikaa.

Viuhkan muoto kertoo kuinka hyvin pelaaja on erottanut kyseisen kirjaimen keskellä olevasta kirjaimesta; mitä kauempana keskustasta viuhkan paksuin osuus on, sitä paremmin pelaaja on erottanut kirjaimet toisistaan. Viuhkat on skaalattu epälineaarisesti, jonka hahmottamiseen kuvaajissa on keskustan ympärillä kolme kehää. Ensimmäinen kehä kuvaa tilannetta, jossa pelaaja on vastannut oikein puolet ajasta, toinen 75% ajasta ja uloimmalla kehällä 100% ajasta eli pelaaja on erottanut kirjaimet toisistaan joka kerta. Viuhkan muotoon vaikuttaa erotustodennäköisyyden lisäksi montako kertaa kirjainpari on esiintynyt pelikerran aikana.

Viuhjakuvaajien ongelmana on se, että kuvaajia on yhtä monta kuin esitettyjä kirjaimia. Tällöin kokonaiskatsauksen saaminen pelaajan pelimenestyksestä vaatii jokaiselle kirjaimelle tehdyn kuvaajan tarkastelua. Pelikertojen välisiä tuloksia voidaan esittää viuhkakuvaajien avulla lisäämällä kirjaimille aiemman pelikerran viuhkat taustalle. Tällöin nähdään kuinka viuhkan muoto ja erityisesti paksuimman kohdan sijainti muuttuu.

Viuhkakuvaajista on lisäksi tehty versio, jossa kuvataan palloina pelaajan tai pelaajien edistymistä useilla eri pelikerroilla. Esimerkkikuvassa 6 esitetään kirjaimen P osaamista suhteessa muihin kirjaimiin. Kuvan data on laskettu yhden koululuokan kaikkien pelaajien vastausten perusteella ja päälimmäisimpänä oleva pallo kuvaa viimeisimpien pelikertojen osaamista. Pallojen vieressä oleva numero kertoo kuinka monta kertaa kyseinen kirjainpari on esiintynyt kaikissa pelaajille näytetyissä trialeissa.

3 Tutkimussuunnitelma

Tutkimuskysymyksenä oli kuinka pelaajan osaamista ja oppimista voisi mallintaa, ja kuinka mallia voisi hyödyntää pelin adaptaatiossa. Tukimuskysymys voidaan jakaa kahteen osaan:

1. Miten mallinnetaan milloin pelaaja oppii eri kirjaimet pelatessaan Ekapeliä, ja millainen malli vastaa tarpeeksi hyvin todellista pelaajaa?
2. Voidaanko mallin pohjalta toteuttaa uudenlainen adaptaatio Ekapeliin, ja kuinka tällainen adaptaatio toimisi?

Mallin lisäksi haluttiin toteuttaa myös kuvaaja, jolla pelaajan osaamisesta saisi tietoa. Koska mallin oli tarkoitus kuvata pelaajan osaamista kaikkina pelikertoina ja kaikkien esitettyjen kirjainten osalta, haluttiin luoda selkeä kuvaaja, josta jokaisen kirjainten osaaminen eri pelikerroilla olisi nähtävissä. Kuvaajien avulla opettaja, ohjaaja tai vanhempi voisi saada helpommin käsityksen pelaajan taidoista ja niiden kehittymisestä. Saatavien kuvaajien tavoitteena oli olla helposti tulkittavia ja sisältää mahdollisimman paljon tietoa pelaajan pelimenestyksestä.

Aiemmat kuvaajat pelaajien osaamisesta kuvasivat tuloksia vain yhdeltä pelikerralta ja jokainen kuvaaja esitti yhden kirjaimen osaamista suhteessa muihin kirjaimiin. Uuden mallin avulla haluttiin selvittää voisiko tämän tiedon yhdistää yhteen kuvaajaan.

3.1 Tutkimuksen rakenne

Tässä tutkielmassa kuvataan kuinka jo kerättyä pelaajadataa 3.3 voidaan käyttää oppimismallin luonnissa. Tutkimuksen aluski päätettiin peliversio, josta kerättyä dataa käsiteltäisiin. Kun käytettävä data oli valittu, hahmoteltiin malli pääpiirteittäin ja valittiin mallintamisessa käytettävät menetelmät. Mallissa päätettiin käyttää bayesilaisen tilastotieteen menetelmiä, jotka on kuvattu osiossa 4. Oppimismallin rakenne luotiin iteratiivisesti lisäämällä ja poistamalla mallin parametreja, jotta malli vastaisi mahdollisimman hyvin pelaajadataa. Mallin sopivuus pelaajadataan ei kuitenkaan ollut ainoa kriteeri. Myös mallista saatavaan otokseen tarvittava laskenta-aika vaikutti valittuihin menetelmiin ja mallin rakenteeseen. Eri mallien tuloksia tarkasteltiin visuaalisesti sekä numeerisesti tarkkailemalla onko malli konvergoitunut.

3.2 Menetelmien valinta

Pelaajan osaamisen selvittäminen vastausten perusteella vaikutti suoraviivaiselta, mutta yksittäisten kenttien perusteella ei kuitenkaan voinut laskea suoraa todennäköisyyttä, jolla pelaaja osaisi tietyn kirjaimen. Tästä syystä mallia lähdettiin toteuttamaan bayesilaisen tilastotieteen keinoja käyttäen. Bayesilaisen tilastotieteessä pyritään perinteisen todennäköisyyden sijaan laskemaan varmuus, jolla tuntematon asia tiedetään. Bayesilaisen tilastotieteen menetelmistä Markovin ketju Monte Carlo soveltuu erityisesti tuntemattomien asioiden todennäköisyyksien laskentaan. Markovin ketju Monte Carlo -menetelmällä pyritään ratkomaan tuntematon, tässä tapauksessa pelaajan osaaminen, tunnettujen havaintojen eli pelaajan vastausten perusteella.

MCMC-menetemä vaatii tietokoneelta laskentatehoa, mutta sen ei ajateltu muodostuvan ongelmaksi. Menetelmän käyttö onkin yleistynyt koneiden laskentatehon kasvaessa.

3.3 Käytetty data

Mallin testauksessa käytettiin 500 satunnaisesti valittujen Ekapeli-Yhden pelaajien pelaajadataa. Pelaajien datasta poistettiin pelaajat, joilla oli alle 3 tai yli 40 pelikertaa. Yhdeksi pelikerraksi laskettiin perättäiset pelikerrat mikäli edellisen kerran lopettamisesta oli kulunut alle 15 minuuttia. Pelikerroista karsittiin ne, joissa pelaaja oli vastannut alle 10 trialiin. Lisäksi pelaajadatasta poistettiin helpotetut trialit sekä arviointikentät. Pelaajien karsimisen jälkeen 500 pelaajadatasta jäi jäljelle 318 pelaajan pelitiedot.

Pelaajadata valittiin pelistä Ekapeli-Yksi, jonka adaptaatio mukautuu pelaajan osaamiseen siten, että esitettävien häiriöärsykkeiden määrä ja kohteiden tippumisnopeus riippuvat pelaajan pelimenestyksestä. Eri kentissä esitettävät kirjaimet valitaan ennalta määritellystä listasta. Yksittäisten trialien sisältö vaihtelee kuitenkin pelaajakohtaisesti, sillä trialin kohde- ja häiriöärsykkeet arvotaan. Pelaajalle esitettävät kirjaimet valitaan ennalta valituista listoista. Listan kirjaimet vaihtuvat, kun adaptaation mukaan pelaaja osaa esitetyt kirjaimet. Tästä johtuen pelaajien etenemistahti pelissä ja yhdellä pelikerralla esitettyjen trialien määrä vaihtelevat suuresti.

Pelaajien trialeista otettiin mukaan vain ne trialit, joissa kysyttiin isoja kirjaimia. Valinta perustui siihen, että isot kirjaimet esitetään pelissä ennen pieniä kirjaimia. Tällöin isojen kirjainten osaaminen saattaa vaikuttaa pienten kirjainten oppimiseen, joka puolestaan johtaisi monimutkaisempaan malliin.

Tutkimuksessa ei ole eritelty ovatko pelaajat pelannet peliä kotona vai koulussa tai itsenäisesti vai ohjatusti. Valittuja pelaajia ei myöskään ole karsittu iän, lukivaikeuksien tai äidinkielen mukaan. Kyseinen peliversio oli tarkoitettu suomen kieltä äidinkielenään puhuville lapsille.

Pelaajadata saatiin tietokannasta, johon tallennettiin suomenkielisen peliversion pelaajadata. Alkuperäisessä datassa oli mukana pelaajien tunnistenumerot ja tiedot trialeista ja pelaajan vastauksista. Pelaajien alkuperäiset tunnistenumerot korvattiin uusilla satunnaisilla tunnisteilla anonymiteetin varmistamiseksi. Lopullinen data sisälsi ainoastaan uudet satunnaiset tunnistenumerot ja listauksen trialeista eri pelikerroilla. Tutkimuksessa käytetty data ei sisältänyt henkilötietoja, jolla yksittäisen pelaajan datan saisi yhdistettyä tiettyyn henkilöön.

4 Teoria

Tässä luvussa esitellään mallissa ja adaptaation luonnissa käytetyt menetelmät ja niiden teoria. Kappaleessa 4.1 esitellään bayesilaisen tilastotietteen peruskäsitteitä. Kappaleissa 4.2 ja 4.3 esitellään Markovin ketjuja ja piilomallia ja kappaleessa 4.5 Markovin ketju Monte Carlo -menetelmä. Kappaleissa 4.6 ja 4.7 Markovin ketju Monte Carlo -menetelmässä käytettyjä algoritmeja. Bayesilaisen tilastotieteen keinoja ja Markovin ketju Monte Carlo -menetelmää on käytetty mallinnuksessa esimerkiksi osiovasteanalyysissa, jolla voidaan myös mallintaa oppimista (Béguin ja Glas 2001), (Mislevy 1986).

4.1 Bayesilainen tilastotiede

Bayesilainen tilastotiede on klassisen tilastotieteen lisäksi tilastotieteen toinen suuri päähaara (Bernardo ja Smith 2009). Bayesilainen tilastotieteessä todennäköisyys kuvaa ehdollista epävarmuutta perinteisen todennäköisyyden sijaan. Bayesilainen tilastotiede pyrkii havaintojen perusteella päättelemään tiedot tutkittavasta kohteesta.

Bayesilainen tilastotiede perustuu Bayesin kaavaan soveltamiseen. Kaava käsittelee ehdollisia todennäköisyyksiä ja tapahtumien välisiä suhteita. Yleisen notaation mukaisesti merkitään tuntemattomia parametreja θ:lla ja havaintoja y:llä. θ:n todennäköisyyttä $p(\theta)$ kutsutaan prioriksi. $p(\theta \mid y)$ kutsutaan θ:n posteriorijakaumaksi, kun y tiedetään.

Todennäköisyydet θ:lle ehdolla y voidaan nyt laskea Bayesin kaavasta

$$p(\theta \mid y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y \mid \theta)p(\theta)}{p(y)},$$

jossa

$$p(y) = \int p(\theta)p(y \mid \theta)d\theta,$$

kun θ on jatkuva. $p(y)$ kuvaa y:n marginaalijakaumaa.

4.2 Markovin ketju

Markovin ketju on menetelmä, joka kuvaa tutkittavan kohteen tiloja eri ajanjaksoina, ja erityisesti tilojen välisiä siirtymiä ja siirtymien todennäköisyyksiä. Tutkittavien kohteiden tilojen tulee olla diskreettejä. Tämä kappale on kirjoitettu teosten Gamerman ja Lopes (2006), Meyn ja Tweedie (2012) ja Grinstead ja Snell (2012) pohjalta.

Olkoon tutkittavalla kohteella N-kappaletta mahdollisia eri tiloja $S = s^{(1)}, s^{(2)}, \dots s^{(N)}$ ja kohteen tila vaihtelee säännönmukaisesti näiden välillä. Tällöin kohteen tilat $s_1, s_2, \dots s_M \in S$ ajanhetkillä $t \in 1, 2, \dots M$ muodostavat Markovin ketjun, jos tila s_t hetkellä t riippuu vain sitä edeltäneestä tilasta s_{t-1}. Merkitään nyt tilan s todennäköisyyttä hetkellä $t+1$

$$P(s_{t+1}|s_t),$$

jossa s_t on tila edellisellä ajanhetkellä t. Annetun ketjun $s = (s_1, s_2, \dots, s_M)$ todennäköisyys saadaan kertomalla eri ajanhetkien tilojen todennäköisyydet keskenään

$$P(s) = \prod_{i}^{N} P(s_{t+1}|s_t)$$

Siirtymätodennäköisyyttä tilasta $s^{(i)}$ tilaan $s^{(j)}$ ajanhetkellä t voidaan merkitä nyt

$$p_{(ij)}(t) = P(s_{t+1} = s^{(j)}|s_t = s^{(i)}).$$

Kun tiloja $S = s^{(1)}, s^{(2)}, \dots s^{(N)}$ on N kappaletta, siirtymätodennäköisyydet voidaan esittää matriisina

$$A = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N,1} & p_{N,2} & \cdots & p_{N,N} \end{pmatrix}.$$

Matriisin yksi rivi kuvaa siirtymätodennäköisyyksiä tilasta $s^{(1)}$ ja rivikohtaisen todennäköisyyksien summan tulee olla 1.

Olkoon esimerkiksi kohde, jolla on 2 tilaa, $s^{(1)}$ ja $s^{(2)}$ joiden välillä kohteen tila vaihtelee taulukon 1 todennäköisyyksien mukaisesti. Todennäköisyys sille, että tilaa $s^{(1)}$

seuraa tila $s^{(2)}$ on siis $p_{1,2}$ ja todennäköisyys tilalle $s^{(1)}$ tilan oltua edellisellä askeleella kumpi tahansa on $p_{1,1}(t) + p_{2,1}(t)$. Todennäköisyys $p_{1,2}$ voidaan esittää myös muodossa $p_{1,2} = 1 - p_{1,1}$ ja vastaavasti $p_{2,1} = 1 - p_{2,2}$, kun mahdollisia tiloja on vain kaksi. Taulukossa 2 on esitetty tilojen todennäköisyydet ajanhetkestä t hetkeen $t+1$.

Taulukko 1. Tilojen $s^{(1)}$ ja $s^{(2)}$ väliset siirtymätodennäköisyydet.

s_t s_{t+1}	$s^{(1)}$	$s^{(2)}$
$s^{(1)}$	$p_{1,1}$	$p_{1,2}$
$s^{(2)}$	$p_{2,1}$	$p_{2,2}$

Taulukko 2. Tilojen $s^{(1)}$ ja $s^{(2)}$ väliset siirtymätodennäköisyydet ajanhetkinä t ja $t+1$.

$$
\begin{tikzcd}
\textit{Tilat} & s^{(1)} \arrow[r, "p_{1,1}"] \arrow[dr, "p_{1,2}"'] & s^{(1)} \\
 & s^{(2)} \arrow[r, "p_{2,2}"'] \arrow[ur, "p_{2,1}"'] & s^{(2)} \\
\textit{Aika} & t \arrow[r] & t+1
\end{tikzcd}
$$

Tässä tutkielmassa tarkastellaan tapausta, jossa tilojen väliset siirtymätodennäköisyydet pysyvät samoina eri ajanhetkinä, eli käsiteltävä Markovin ketju on aikahomogeeninen. Tällöin tilojen väliset siirtymätodennäköisyydet voidaan laskea kohteesta kerätystä datasta laskemalla jokaiselle tilalle monestiko tila on vaihtunut toiseen tilaan ja monestiko tila on pysynyt samana.

4.3 Markovin piilomalli

Markovin piilomalli (Hidden Markov model, HMM) on tilastollinen menetelmä, jota käytetään, kun tutkittavan kohteen tiloja ja sen muutoksia ei voida suoraan havainnoida (Rabiner 1989), (Cappé, Moulines ja Rydén 2009). Kohteen tilojen S sijaan kohteesta saadaan havaintoja y, joiden oletetaan olevan riippuvaisia kohteen tilasta.

Markovin piilomalli pyrkii selvittämään kohteen tilan ja toiminnan näiden havaintojen y perusteella. Toisin kuin Markovin ketjuille, Markovin piilomallissa kaikkia mahdollisia tiloja S ei välttämättä tiedetä. Oletetaan, että tutkittavalla kohteella on N eri tilaa $S = s^{(1)}, s^{(2)}, \ldots, s^{(N)}$.

Merkitään nyt havaintoja $y = y_1, y_2, \ldots, y_M$ ajanhetkillä $t = (1, 2, \ldots, M)$. Havainnot $y_1, y_2, \ldots, y_M$ riippuvat tuntemattomasta Markovin ketjusta $s_1, s_2 \ldots, s_M$, jonka tiloja ja mahdollisesti myöskään tilojen välisiä siirtymätodennäköisyyksiä ei tiedetä. Mallissa pyritään havaintojen perusteella arvioimaan tuntemattomat tilat s ja niiden väliset siirtymätodennäköisyydet. Havainnot y voidaan esittää tilojen s ilmentyminä, kuten taulukossa 3. Siirtymätodennäköisyydet voidaan nyt määritellä samoin kuin Markovin ketjulle 4.2.

Taulukko 3. Markovin piilomallin tilat ja havainnot ajanhetkinä $t = (1, 2, \ldots, M)$.

$$\begin{tikzcd}
s_1 \arrow[r] \arrow[d] & s_2 \arrow[r] \arrow[d] & s_3 \arrow[r] \arrow[d] & \cdots & s_M \arrow[d] \\
y_1 & y_2 & y_3 & \cdots & y_m
\end{tikzcd}$$

Annetun ketjun $y = (y_1, y_2, \ldots, y_M)$ todennäköisyys voidaan laskea Bayesin säännön mukaisesti summaamalla kaikkien tilojen S yli

$$P(y) = \sum_S P(y|s)P(s).$$

4.4 Monte carlo -menetelmä

Monte Carlo -menetelmää käytetään approksimointiin silloin, kun tarkan arvon laskeminen olisi laskennallisesti hankalaa tai mahdotonta (Rubinstein ja Kroese 2016). Menetelmä vaatii ongelmasta tietoa kuten miten jokin asia toistuu. Menetelmän perusperiaate on toistaa tapahtumaa ja katsoa mihin lopputulokseen kullakin toistolla päädyttiin. Lopputulosten jakautuminen approksimoi kohteen todellista jakaumaa.

Yksinkertaisin esimerkki algoritmin toiminnasta on osuman ja erehdyksen menetelmä. Esimerkiksi loton voittotodennäköisyydet voidaan laskea Monte Carlo

-menetelmällä valitsemalla jokin lottorivi ja tämän jälkeen arpomalla lukuisia voittorivejä ja laskemalla kuinka moni voittorivin numeroista oli valitulla lottorivillä kullakin kierroksella. Voittotodennäköisyydet kullekin voittoluokalle saadaan nyt laskemalla kuinka monta prosenttia kaikista kierroksista osui kuhunkin voittoluokkaan.

4.5 Markovin ketju Monte Carlo

Markovin ketju Monte Carlo (MCMC) on simulointimenetelmä, jolla voidaan approksimoida posteriorijakaumaa silloin, kun posteriorijakauman tarkat arvot eivät ole laskettavissa. Menetelmän käyttäminen voi olla helpompaa kuin tarkkojen arvojen laskeminen mikäli tilojen todennäköisyyksiin vaikuttaa laaja joukko parametreja. Menetelmän käyttö voi olla epävarmempaa kuin tarkkojen arvojen laskeminen, jos laskenta ei konvergoidu. Konvergenssi voidaan kuitenkin laskea, ja usein Markovin ketju Monte Carlo -menetelmällä saadaan riittävän tarkka tulos pienemmällä laskenta-ajalla kuin tarkkaa arvoa laskettaessa. Markovin ketju Monte Carlo -menetelmää on kuvattu tarkemmin kirjoissa Neal (1993) ja Gamerman ja Lopes (2006), joiden pohjalta tämä luku on kirjoitettu.

Menetelmän käyttö on yleistynyt laskentatehon kasvaessa (Gelman ym. 2004). Markovin ketju Monte Carlo -menetelmällä saadaan otos kappaleessa 4.1 esiteltyjen tuntemattomien parametrien θ:n arvoista. θ:n arvojen ajatellaan kuvaavan otosta todellisesta posteriorijakaumasta silloin, kun laskenta konvergoi. Approksimointi suoritetaan laskemalla θ:lle uusi tila ja valitsemalla θ^k:n arvo vain edellisen tilan θ^{k-1}perusteella. Tällöin saatu otos θ:n arvoista toteuttaa Markovin ominaisuuden, jota kuvattiin luvussa 4.3.

Markovin piilomallin priori saadaan kaavalla

$$p(\theta) = p(\theta_0) \prod_{k=1} p(\theta_k \mid \theta_{k-1})$$

ja havaintojen todennäköisyys tilalle θ kaavalla

$$p(y \mid \theta) = \prod_{k} p(y_k \mid \theta_k).$$

Markovin ketjun posteriorijakauma saadaan nyt kaavalla

$$\begin{aligned} p(\theta_k \mid y) &\propto p(\theta_k \mid y_{1:k}) p(y_{k+1:n} \mid \theta_k, y_{1:k}) \\ &= p(\theta_k \mid y_{1:k}) p(y_{k+1:n} \mid \theta_k) \end{aligned}.$$

4.6 Gibbs-otanta

Gibbs-otanta on eräs Markovin ketju Monte Carlo -otantamenetelmä, jonka periaatteena on, että jokaisen iteraation approksimaatio θ:sta kuuluu otokseen posteriorijakaumasta (Neal 1993).

Gibbs-otannassa θ:a kuvataan komponentein $\theta = (\theta_1, \theta_2 \ldots \theta_n)$, jolloin seuraava askel lasketaan edellisen askeleen määräämästä ehdollisesta jakaumasta

$$p(\theta_j \mid \theta_{-j}^{t-1}, y),$$

jossa θ_j kuvaa θ:n komponenttia j ja θ_{-j} muita θ:n arvoja. Komponentin θ_j arvoja päivittäessä muiden komponenttien arvot pysyvät vakioina. Yksi MCMC-otannan askel saadaan päivittämällä jokainen θ:n komponentti. Komponenttien päivitysjärjestys voi olla vakio tai sattumanvarainen.

Gibbs-otanta on paljon käytetty menetelmä, sillä se konvergoituu nopeasti eikä tarvitse normalisointia. Gibbs-otanta voidaan nähdä myös Metropolis-Hastings-algoritmin erikoistapauksena, jossa jokainen siirtymä tilojen välillä hyväksytään. Tarvittavien iteraatioiden määrä on pienempi kuin Metropolis-Hastings -algoritmia käytettäessä, sillä Metropolis-Hastings ei salli kaikkien θ:n tilojen välisiä siirtymiä.

4.7 Forward backward -algoritmi

Forward-backward -algoritmi on datan siloitusmenetelmä (Eisner 2002). Menetelmän etuna on se, että tilan todennäköisyyksiä laskettaessa huomioidaan sekä edeltävän että seuraavan tilan todennäköisyydet. Menetelmässä lasketaan ensin forward-tekijät ja backward-tekijät ja lopuksi kerrotaan niiden arvot keskenään, jolloin saadaan eri tilojen todennäköisyydet.

Forward-backward -algoritmia käyttäen lauseen 4.5 posteriorijakauman tekijöistä saadaan

$$p(\theta_k \mid y_{1:k}) \propto \sum_{\theta_{k-1}} p(\theta_{k-1} \mid y_{1:k-1}) p(\theta_k \mid \theta_{k-1}) p(y_k \mid \theta_k)$$

ja

$$p(y_{k+1:n} \mid \theta_k) = \sum_{\theta_k} p(\theta_k \mid \theta_{k-1}) p(y_k \mid \theta_k) p(y_{k+1:n} \mid \theta_k).$$

Forward backward -algoritmin lisäksi käytetään menetelmää backward sampling. Sitä käytetään, kun posteriorijakaumaa ei voida laskea tai halutaan otos jakaumasta. Tällöin tiloille lasketaan ensin todennäköisyydet forward- ja backward-tekijöiden avulla. Seuraavassa vaiheessa valitaan muuttujille uudet arvot. Valinta tapahtuu käänteisessä järjestyksessä siten, että ensin valitaan viimeisen tilan arvo ja sen ehdollistamasta jakaumasta sitä edeltävän tilan arvo kunnes kaikki arvot on laskettu.

5 Oppimismallin rakenne

Tässä luvussa kuvataan pelaajan oppimismallin rakenne. Mallin tavoitteena on kuvata pelaajan osaamista ja toimintaa mahdollisimman yksinkertaisesti, mutta riittävän tarkasti kuvaamaan todellista pelaajaa. Yksinkertainen malli on myös helpompi ymmärtää ja tilojen väliset yhteydet ovat helpommin hahmotettavissa. Mallin avulla halutaan hahmottaa mitkä kirjaimet pelaaja osaa kullakin pelikerralla. Bayesilaisen tilastotieteen mukaan voidaan ajatella, että mallin pohjalta lasketaan millä varmuudella pelaaja osaa kunkin kirjaimen.

Mallin halutaan kuvaavan mahdollisimman tarkkaan pelaajan osaamista ja antavan näin tietoa pelaajasta. Osaamistilojen todennäköisyyksien laskemista kuvataan luvussa 6, ja laskettujen osaamistilojen sopivuutta pelaajien datoihin kuvataan myöhemmin luvussa 7.

Mallista toteutettiin kaksi eri versiota. Aluksi kuvataan mallin yksinkertaista versiota, jossa pelaaja joko osaa yhdistää kuullun äänteen esitettyyn kirjaimeen tai ei osaa sitä ja valitsee todennäköisemmin väärin. Myöhemmin kappaleessa 5.4 malliin lisätään mahdollisuus, että pelaaja osaa erottaa kirjaimen lähes kaikista muista kirjaimista, mutta sekoittaa sen johonkin toiseen kirjaimeen, usein joko samankuuloiseen tai -näköiseen. Näiden kahden version lisäksi esitetään kuinka malleista voidaan tehdä tarkempia muuttamalla mallissa käytettyjä vakioita MCMC-otoksessa laskettaviksi arvoiksi.

5.1 Oppimismallin taustaa

Malli perustuu Atkinsonin, Crothersin ja Calfeen esittämiin pariassosiaatiomalleihin. Artikkeleissa Atkinson ja Crothers (1964) ja Calfee ja Atkinson (1965) kuvataan erilaisia malleja pari-assosiaation oppimisen mallinnukseen.

Artikkelissa Calfee ja Atkinson (1965) kuvataan oppimismalli, jossa on kolme eri tilaa. Ensimmäisessä tilassa U kohdetta ei ole vielä esitetty pelaajalle lainkaan. Toisessa tilassa S kohde on esitetty, mutta pelaaja osaa sen vain väliaikaisesti. Kolmannessa tilassa L pelaaja on oppinut kohteen pysyvästi. Mallille esitetään siirtymämatriisi (s. 254)

$$\begin{array}{c} \\ L_n \\ S_n \\ U_n \end{array} \begin{array}{c} \begin{array}{ccc} L_{n+1} & S_{n+1} & U_{n+1} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ a & 1-a & 0 \\ ca & c(1-a) & 1-c \end{pmatrix} \end{array} \quad \begin{array}{c} Pr(oikein|tila) \\ \begin{pmatrix} 1 \\ 1-f+fg \\ g \end{pmatrix} \end{array}$$

Siirtymämatriisissa n kuvaa monettako kertaa kohde esitetään. Parametri a kuvaa todennäköisyyttä, että pelaaja oppii tällä esityskerralla n kohteen pysyvästi ja c todennäköisyyttä, että pelaaja oppii kohteen esityskerralla n ehdolla, ettei pelaaja ole vielä oppinut kohdetta esityskertaan n mennessä. Parametri f kuvaa todenäköisyyttä, että pelaaja unohtaa kohteen ja g arvaustodennäköisyyttä. Malli täyttää Markovin ketjun ehdot, sillä pelaajan osaamisen tila riippuu vain edeltävästä tilasta.

R. C. Atkinson on kirjoittanut myös lukuisia muita artikkeleita oppimisen mallinnuksesta (Chant ja Atkinson 1973) ja tietokoneavusteisesta lukemaanoppimisesta. Erityisen kiinnostavia olivat artikkelit Groen ja Atkinson (1966), jossa kuvataan esitettävän kohteen valintaa ja Fletcher ja Atkinson (1972), joka kuvaava Stanfordin yliopiston CAI-ohjelmaa (computer-assisted instruction). CAI-ohjelmaa käytettiin englannin kielen lukemaanoppimisessa. Pelajan tehtävänä oli valita kuulemansa sana näytöllä näkyvistä vaihtoehdoista.

Englannin kieltä voidaan opetella harjoittelemalla kirjain-äänne-vastaavuuksia kuten Ekapelissä tai yhdistämällä kokonaisia lausuttuja sanoja kirjoitukseen kuten CAI-ohjelmassa (Ehri ym. 2001) ja myös Ekapelissä. Toisin kuin Atkinsonin kuvaamissa malleissa, joissa käsiteltiin kokonaisten sanojen oppimista, tässä tutkielmassa keskitytään mallintamaan kirjainten oppimista. Matemaattisessa mielessä opittavilla kohteilla ei kuitenkaan ole eroa.

5.2 Kirjainten osaaminen eri pelikerroilla

Mallissa pelaajalla on kaksi mahdollista tilaa kirjaimen osaamiselle: osattu ja ei osattu. Jos pelaaja on pelikerralla mallin mukaan osannut kirjaimen, merkitään sen osaamistilaa 1, ja mikäli pelaaja ei ole osannut kirjainta, kirjaimen osaamistila on 0. Koska oppimispeliä pelataan useilla eri kerroilla, pelaajan kirjainten osaamistilat esitetään listana, jossa kukin alkio kuvaa kirjaimen osaamistilaa kullakin pelikerralla.

Merkitään kirjainten osaamistiloja eli tuntematonta θ:lla ja pelaajadataa eli tunnetua y:llä, kuten bayesilaisessa notaatiossa on tapana. θ_1 tarkoittaa nyt pelaajan kirjainten tiloja ensimmisellä pelikerralla ja $\theta^{(A)}$ pelaajan A-kirjaimen osaamista eri ajanhetkinä. $\theta_1^{(A)} = 1$ tarkoittaa siis kirjaimen A-osaamistilaa ensimmäisellä pelikerralla ja osaamistila 1, että pelaaja on mallin mukaan osannut kirjaimen.

Kirjainten osaamistilat voidaan nyt esittää matriisin muodossa, jossa yksi rivi kuvaa kunkin kirjaimen osaamistiloja eri pelikerroilla ja yksi sarake kaikkien kirjainten osaamistiloja kyseisellä pelikerralla. Taulukossa 4 on esitetty esimerkki pelaajan osaamistiloista, jossa ensimmäisen pelikerran osaamistilat ovat $\theta_1 = [1,0,1,0,1,0,0,0,1,0,0]$ ja kirjaimen A osaamistilat $\theta^{(A)} = [1,0,1,1,1,1,1]$.

Mallissa haluttiin, että pelaaja voi oppia pelikertojen välissä kirjaimen tai unohtaa sen. Olkoon x todennäköisyys sille, että jos pelaaja osaa kirjaimen nyt, osaa hän sen myös seuraavalla pelikerralla. Olkoon lisäksi z todennäköisyys sille, että jos pelaaja ei osaa kirjainta nyt, hän ei osaa sitä seuraavallakaan kerralla. Kirjaimen osaamistiloille saadan nyt taulukon 5 kuvaama siirtymämatriisi. Siirtymämatriisin mukaan pelaaja voi siis oppia kirjaimen pelikertojen välillä todennäköisyydellä $1 - z$ ja unohtaa kirjaimen todennäköisyydellä $1 - x$. Siirtymämatriisin avulla voidaan laskea kirjaimelle sen tilojen todennäköisyydet tietyllä pelikerralla, kun sen tilojen todennäköisyydet tiedetään edeltävältä pelikerralta. Kirjaimen osaaminen tietyllä pelikertaa riippuu vain edellisen pelikerran osaamisesta, jolloin kirjaimen osaamistilat toteuttavat Markovin ominaisuuden. Siirtymämatriisit voivat olla joko samanlaisia kaikille kirjaimille tai jokaisen kirjaimen tilojen siirtymämatriisi voi olla erilainen.

Kirjainten osaaminen on mallissa riippumatonta toisistaan eli pelaajan tietyn kirjai-

Kirjain pelikerta	1	2	3	4	5	6	7
A	1	0	1	1	1	1	1
B	0	1	1	0	1	0	0
D	1	0	0	0	0	0	0
F	0	1	0	1	0	1	0
G	1	1	0	0	1	0	1
H	0	0	0	0	1	0	1
I	0	1	1	0	1	1	1
J	0	1	1	1	1	0	0
K	1	0	1	1	0	1	1
L	0	0	0	1	1	0	1
M	0	0	1	0	1	0	1

Taulukko 4. Esimerkki kirjainten osaamistiloista

	0	1
0	z	$1-z$
1	$1-x$	x

Taulukko 5. Kirjaimen tilojen siirtymistodennäköisyydet.

men osaaminen ei vaikuta muiden kirjainten osaamiseen. Todellisuudessa samankaltaisten kirjainten osaaminen kuitenkin korreloi keskenään. Tätä mallinnettiin kirjainten sekoittumisella, jota on kuvattu luvussa 5.4.

Jokaisen kirjaimen osaaminen eri pelikerroilla muodostaa nyt Markovin piilomallin, jossa pelikerralla esitetyt trialit ovat havainto pelaajan osaamisesta tällä pelikerralla. Kirjainten oppimistodennäköisyys, eli siirtymätodennäköisyys tilasta 0 tilaan 1, on tässä mallissa päätetty pitää vakiona.

5.3 Pelaajan vastausalgoritmi

Pelaajan oletetaan käyttäytyvän loogisesti ja mallissa oletetaan, että mikäli pelaaja osaa kysytyn kirjaimen, hän myös valitsee oikean vaihtoehdon. Mikäli pelaaja ei osaa kohdeärsykettä, hänen oletetaan osaavan sulkea pois vastausvaihtoehdoista ne kirjaimet, jotka hän osaa. Tällöin pelaaja valitsee satunnaisesti jonkun kentässä esiintyvän kirjaimen niistä, joita hän ei vielä osaa. Pelaaja voi kuitenkin vastata vahingossa väärin, vaikka osaisikin kirjaimen. Pelaaja voi esimerkiksi häiriintyä pelin ulkoisesta tekijästä johtuen. Mallissa tätä kuvataan häiriöarvolla (lapse rate) δ. Merkitään nyt n:llä trialissa olleiden kirjainten lukumäärää ja k:lla niiden kentässä olleiden kirjainten lukumärää, joita pelaaja ei mallin mukaan ole osannut eli joiden tila tällä pelikerralla on ollut 0. Tällöin todennäköisyys valita kohteena ollut kirjain a kentässä t on

$$p(a,t \mid \theta) = (1-\delta) + \frac{1}{n}\delta,$$

kun pelaaja osaa kirjaimen a ja

$$p(a,t \mid \theta) = \frac{1}{k}(1-\delta) + \frac{1}{n}\delta,$$

kun pelaaja ei osaa kirjainta.

Oletetaan, että pelaajalle esitetään nyt triali $\{A, E, I, O\}$, jossa kohteena on A-kirjain, ja olkoon pelaajan osaamisen tilat kirjaimille $\{A = 1, E = 1, I = 0, O = 1\}$. Olkoon lisäksi häiriöarvo δ 0,05. Tällöin todennäköisyys valita kirjain A on

$$p(a,t|\theta) = 0{,}95 + \frac{1}{4} \times 0{,}05 = 0{,}9625$$

ja muiden kirjainten vastaustodennäköisyys on $\frac{1}{4} \times 0{,}05 = 0{,}0125$.

Jos pelaaja ei osaisi kirjainta A ja kirjainten tilat olisivat $\{A = 0, E = 1, I = 0, O = 1\}$, todennäköisyys valita vaihtoehto A olisi

$$p(a,t|\theta) = \frac{1}{2} \times 0{,}95 + \frac{1}{4} \times 0{,}05 = 0{,}4875$$

ja samoin kirjaimelle I 0,4875 ja kirjaimille E ja O 0,0125.

Valittavan häiriöarvon tulee olla tarpeeksi pieni, jotta olisi todennäköisempää, ettei pelaaja osaa kirjaimia kuin se, että pelaaja häiriintyisi jatkuvasti ja vastaisi tämän takia sattumanvaraisesti.

5.4 Kirjainten sekoittuminen

Edellisen yksinkertaisen mallin mukaan pelaaja ei välttämättä osaisi m-kirjainta, vaikka hän vastaisikin oikein kaikissa muissa kentissä kuin sellaisissa, joissa häiriöärsykkeenä esiintyy n-kirjain. Kirjainparien, joissa joko äänteet ovat lähellä toisiaan tai kirjainmerkit muistuttavat toisiaan, sekoittuminen on yleistä esimerkiksi lukihäiriöisillä. Kirjainparien sekoittuvuutta varten lisätään malliin jokaista pelikertaa kohden matriisi, joka ilmaisee mitkä kirjaimet pelaajalla menevät sekaisin. Merkitään kirjainparille (a,b) sekoittumista $\rho^{(a,b)} = 1$. Vastaavasti, jos kirjaimet a ja b eivät mene pelaajalla sekaisin keskenään, merkitään $\rho^{(a,b)} = 0$.

	A	E	K	M	N
A	1	0	0	0	0
E	0	1	0	0	0
K	0	0	1	0	0
M	0	0	0	1	1
N	0	0	0	1	1

Taulukko 6. Esimerkki pelaajan kirjainten sekoittuvuudesta, kun pelaaja erottaa muut kirjamet toisistaan, mutta sekoittaa kirjaimet M ja N keskenään.

Sekoittumista kuvaavat matriisit oletetaan symmetrisiksi, eli jos pelaajalta kysytään kirjainta a ja pelaaja sekoittaa kirjaimen a kirjaimeen b, niin kirjainta b kysyttäessä pelaaja yhtälailla sekoittaa kirjaimet a ja b keskenään. Todellisuudessa näin ei välttämättä ole, vaan kirjaiten sekoittuminen toisiinsa voi olla epäsymmetristä. Pelaaja voi esimerkiksi sekoittaa kirjaimet N ja M kuullessaan kirjaimen N-äänteen ja nähdessään häiriöärsykkeen M, mutta kuullessaan kirjaimen M-äänteen erottaa sen häiriöärsykkeestä N, kuvattu taulukossa 6. Sekoittuvuus oletetaan tässä symmetriseksi, sillä epäsymmetrisen sekoittuvuuden löytyminen vaatisi suuremman määrän dataa kuin symmetrisen, sillä muuttujien määrä mallissa kasvaisi.

Sekoittumismatriisien diagonaalialkiot ovat kaikki 1, sillä mikäli pelikentässä esiintyisi kaksi samaa kirjainta, pelaaja ei voisi mitenkään erottaa niitä toisistaan. Mallin kannalta diagonaalialkioiden arvoilla ei ole väliä, koska pelissä ei esitetä kenttiä, joissa sama kirjain olisi useammassa kuin yhdessä vastausvaihtoehdossa.

Kirjainten sekoittuvuuden vaikuttaa valintatodennäköisyyksiin seuraavasti: mikäli pelaaja sekoittaa trialissa esiintyvän häiriöärsykkeen kohdeärsykkeeseen, todennäköisyydet, että pelaaja valitsee minkä tahansa näistä vaihtoehdoista, ovat yhtä suuret. Kun pelaaja osaa kirjaimen, todennäköisyys valita kirjain a trialissa t on

$$p(a,t \mid \theta) = \frac{1}{k}(1-\delta) + \frac{1}{n}\delta,$$

jossa k on kohdeärsykkeen kanssa sekaisinmenevien häiriöärsykkeiden määrä trialissa. Vastaavasti, kun pelaaja ei osaa kirjainta, todennäköisyys valita kirjain a on sama kuin edellä, mutta k kuvaa trialissa olleiden kirjainten lukumäärää, joita pelaaja ei osaa tai jotka pelaaja sekoittaa kohdeärsykkeeseen. Täten yhden trialin kohdalta sekoittuvuudet ovat yhdentekeviä, mikäli pelaaja ei osaa kirjaimia. Esimerkiksi olkoon pelaajan tilat seuraavat $\{A=1, K=0, M=1, N=1\}$ ja pelaajalle esitettävä triali $t = \{A, K, M, N\}$, jossa kysytään M-kirjainta. Nyt pelaajan vastaukseen vaikuttavat kirjaimen M osaamistilan lisäksi kirjaimen M muihin kirjaimiin sekoittumista kuvaavat tilat

$$\{\{A,M\}=0, \{K,M\}=0, \{N,M\}=0\}$$

ja häiriöarvo $\delta = 0{,}05$, 3. pelikerta taulukoissa 7 ja 8. Nyt pelaajan vastaustodennäköisyydet kirjaimille olisivat

$$p(t|\theta) = \{A: 0{,}0125;\ K: 0{,}0125;\ M: 0{,}9625;\ N: 0{,}0125\}.$$

Jos taas kirjainparien sekoittumistilat olisivat

$$\{\{A,M\}=0, \{K,M\}=0, \{N,M\}=1\},$$

taulukoissa 7 ja 8 2. pelikerta, vastaustodennäköisyydet olisivat

$$p(t|\theta) = \{A : 0{,}0125\,;\ K : 0{,}0125\,;\ M : 0{,}4875\,;\ N : 0{,}4875\}.$$

Kirjain\Pelikerta	1	2	3
A	0	1	1
K	0	0	0
M	0	1	1
N	0	1	1

Taulukko 7. Esimerkki pelaajan kirjainten osaamistiloista eri pelikerroilla.

1. pelikerta

	A	K	M	N
A	1	0	0	0
K	0	1	0	0
M	0	0	1	1
N	0	0	1	1

2. pelikerta

	A	K	M	N
A	1	0	0	0
K	0	1	0	0
M	0	0	1	1
N	0	0	1	1

3. pelikerta

	A	K	M	N
A	1	0	0	0
K	0	1	0	0
M	0	0	1	0
N	0	0	0	1

Taulukko 8. Esimerkki pelaajan kirjainten sekoittumistiloista eri pelikerroilla.

Jos taas kirjainten osaamistilat olisivat $\{A = 1, K = 0, M = 0, N = 1\}$ ja kirjainparien sekoittumistilat $\{\{A,M\} = 0, \{K,M\} = 0, \{N,M\} = 0\}$, taulukoissa 7 ja 8 1. pelikerta, vastaustodennäköisyydet olisivat

$$\{0{,}0125\,;\ 0{,}4875\,;\ 0{,}4875\,;\ 0{,}0125\}$$

ja sekoittumistiloilla $\{\{A,M\} = 0, \{K,M\} = 0, \{N,M\} = 1\}$ vastaavasti

$$p(t|\theta) = \{A : 0{,}0125\,;\ K : 0{,}329166\ldots\,;\ M : 0{,}329166\ldots\,;\ N : 0{,}329166\ldots\}.$$

Mallissa halutaan, että pelaaja voi oppia erottamaan sekoittamansa kirjaimet. Tätä kuvataan samanlaisella siirtymämatriisilla kuin kirjainten oppimista 5. x kuvaa nyt todennäköisyyttä, jolla pelaaja sekoittaa kirjaimet seuraavallakin pelikerralla, kun hän sekoittaa ne edellisellä pelikerralla. Vastaavasti z kuvaa todennäköisyyttä sille, ettei pelaaja sekoita kirjaimia seuravallakaan pelikerralla mikäli hän ei sekoita niitä nyt. Samoin kuin kirjainten osaamistilat, tietyn kirjainparin sekoittumisen tilat ajassa muodostavat nyt Markovin piilomallin.

6 Kirjainten osaamistilojen laskenta

Tässä luvussa kuvataan kuinka osaamistilojen todennäköisyyksiä voidaan approksimoida Markovin ketju Monte Carlo -menetelmällä. Simuloinnin tavoitteena oli selvittää millä todennäköisyydellä pelaaja osaa kunkin kirjaimen, eli millä varmuudella kirjainten osaamistilat ovat tilassa 1 eri pelikerroilla. Pelaajan valinta-algoritmin seurauksena näiden todennäköisyyksien laskeminen analyyttisesti on hyvin vaikeaa tai jopa mahdotonta. Tämä johtuu siitä, että jokaisen trialin vastaukseen vaikuttavat kaikkien siinä esiintyvien kirjainten osaamistilojen todennäköisyydet, ja todennäköisyydet osaamistiloille lasketaan kaikkien pelikerralla esitettyjen trialien yli. Yhden pelikerran osaamistilojen laskeminen on jo nyt hyvin monimutkaista, vaikka jätettäisi huomioimatta se, että jokaisen kirjaimen osaamistilaan vaikuttaa myös sen osaamistila edellisellä pelikerralla.

Teoriassa tarkan vastauksen voisi laskea myös kaikkien pelaajan mahdollisten tilojen yli. Kun pelaajalle esitettyjen kirjainten määrä on k ja pelikertojen määrä p, on mahdollisten eri tilojen määrä $2^{k \times p}$. Kun kirjainparien sekoittumiset lisätään malliin, mahdollisten tilojen määrä kasvaa hurjasti, sillä jokaista pelikertaa kohden tulee $\frac{(k-1)^2}{2}$ uutta tilaa. Näiden lisäksi virhearvon ja siirtymämatriisien laskenta kasvattaa laskentaan kuluvaa aikaa. Todennäköisyyksien laskeminen kaikille mahdollisille tiloille veisi paljon aikaa ja muistia, sillä tarve kasvaa eksponentiaalisesti. MCMC:llä saadaan otos tilojen posteriorijakaumasta, joka approksimoi tarkkaan laskettuja todennäköisyyksiä.

6.1 Markovin ketju Monte Carlo -otoksen laskenta

Pelaajan jokaisen tilan arvot päivitetään Markovin ketju Monte Carlo -menetelmää käyttäen. Yhdellä iteraatiokerralla päivitetään jokaisen kirjaimen osaamistilat ja mallin versiosta riippuen myös kirjainten siirtymämatriisit ja sekoittumismtilat. Uuden tilan valitseminen tapahtuu käymällä tilat yksitellen läpi, laskemalla todennäköisyydet tilan eri vaihtoehdoille ja valitsemalla uusi arvo. Arvo valitaan satunnaisesti mahdollisista tiloista niille lasketuilla todennäköisyyksillä.

Todennäköisyydet eri tiloille saadaan laskemalla käyttäen Gibbs-otantaa ja forward backward -menetelmää käyttäen. Muiden tilojen arvot pidetään tällöin vakiona ja kirjaimen tilat valitaan näin ollen muiden kirjainten tilojen määräämästä ehdollisesta jakaumasta.

Jokaisen kirjaimen tiloja $\theta^{(a)}$ kohdellaan θ:n osavektoreina Gibbs-otannan mukaisesti. Päivityksen ajan muiden kirjainten tilat pidetään vakioina ja $\theta^{(a)}$:n arvot valitaan ehdollisesta jakaumasta

$$p(\theta^{(a)} \mid \theta^{(-a)}, y),$$

jossa $\theta^{(-a)}$ kuvaa muiden kirjainten kuin a:n osavektoreita.

Uusien arvojen valinnassa käytetään forward-backward -menetelmää. Siinä jokaiselle arvolle lasketaan luvussa 4.7 kuvatut forward-backward -tekijät. Näiden tekijöiden perusteella arvotaan tilojen todennäköisyyksien suhteessa ensin tilan arvo viimeisellä pelikerralla. Seuraavaksi valitaan toiseksi viimeisen pelikerran arvo jakaumasta

$$p(s_k|s_{k+1})\alpha_k,$$

jossa s_{k+1} on viimeisen pelikerran arvo. Iteraatiota jatketaan lopusta päin ehdollistamalla uuden arvon valinta seuraavan pelikerron arvolla kunnes kaikki uudet arvot on valittu. Näin kirjaimen tilan arvoon vaikuttaa pelikerralla pelattujen trialien lisäksi myös muilla pelikerroilla pelatut trialit, joissa kirjain esiintyy.

Kun jokaisen kirjaimen ja kirjainparin sekoittumisen tila on päivitetty, valitaan uudet arvot siirtymämatriiseille sekä häiriöarvolle. Nämä päivitetyt arvot tallennetaan ja iteraatio aloitetaan alusta. Seuraavalla iteraatiolla valitut arvot riippuvat vain edellisen iteraation arvoista. Iterointia jatkamalla saadaan otos kaikkien tilojen eli θ:n posteriorijakaumasta.

6.2 Kirjainten osaamistilojen siirtymämatriisien laskenta

Osaamistilojen siirtymämatriisien laskenta voidaan toteuttaa joko kirjainkohtaisesti tai laskemalla jokaisen kirjaimen tiloille oma siirtymämatriisi. Mallissa siirtymätodennäköisyyden päätettiin pysyvän samana eri ajanhetkinä. Siirtymämatriisit päivitetään joka iteraatiolla laskemalla montako kertaa kirjaimen (tai kaikkien kirjainten) osaamistila on pysynyt samana ja montako kertaa tila on vaihtunut. Olkoon nyt a lukumäärä, jona tila on pysynyt samana ja b lukumäärä, jona tila on vaihtunut. Uudet arvot siirtymämariisille saadaan beta-jakaumasta parametrein $\alpha = a + 1$ ja $\beta = b + 1$.

Lopuksi päivitetään mallin prioria siirtymämatriisin arvojen mukaan. Merkitään kirjaimen A prioria $p_A(\theta_0)$, jossa θ_0 kuvaa pelaajan osaamista ennen pelaamista. Prioritodennäköisyydet voidaan nyt laskea kaavalla

$$p_A(\theta_0) = \frac{1 - z}{(1 - z) + (1 - x)},$$

jossa x on todennäköisyys sille, että osaamistila on 1, kun se aiemmalla kerralla on ollut 1 ja z todennäköisyys, että tila pysyy nollassa, kun se aiemmalla pelikerralla on ollut 0.

Tilan vaihtumista kuvaavat matriisit valittiin aluksi kirjainkohtaiksi, koska useat pelaajat osaavat jo aloittaessaan pelaamisen joitain kirjaimia (esimerkiksi nimessään olevat kirjaimet). Näiden kirjainten osaaminen pysyy yleensä vakiona pelin ajan. Siirtymätodennäköisyyksien x ja z oletettaminen samoiksi kaikille kirjaimille saataisi johtaa ongelmiin tilojen päivitysessä. Tällöin jo alusta osattujen kirjainten osaaminen kaikilla pelikerroilla kasvattaisi kaikkien kirjainten todennäköisyyden $p(\theta_k^a = 1 \mid \theta_{k-1}^a = 1)$ hyvin lähelle yhtä, sillä siirtymätodennäköisyydet lähenisivät nollaa.

Yhtenäinen siirtymätodennäköisyys olisi kuitenkin perusteltua, jotta malli pysyisi mahdollisimman yksinkertaisena. Tällöin siirtymätodennäköisyydet voisi pitää vakioina, joiden arvot on laskettu kaikkien pelaajien vastausten perusteella. Toinen vaihtoehto olisi antaa yhtenäiselle siirtymätodennäköisyydelle minimi- ja maksimiarvot, ja näin estää sitä painumasta liian pieneksi.

6.3 Kirjainparien sekoittumisten siirtymämatriisi

Osaamistilojen välistä siirtymää kuvataan kaikille kirjainpareille yhteisellä matrisilla. Arvojen laskeminen jokaisen kirjainparin sekoittumiselle vaatisi hyvin suuren määrän pelidataa, sillä tietyt kirjainparit eivät välttämättä esiinny trialeissa juuri lainkaan. Kirjainten sekoittumisen halutaan myös olevan eräänlainen erikoistapaus ja täten huomattavasti epätodennäköisempää kuin se, etteivät kirjaimet sekoitu. Mallin tulisi kuitenkin toimia siten, että on todennäköisempää, ettei pelaaja osaa tiettyä kirjainta kuin se, että pelaaja sekoittaisi kirjaimen kaikkiin muihin. Tästä syystä sekoittumisen prioriksi asetettiin 0,2.

Yksittäisen siirtymätodennäköisyysmatriisin voidaan ajatella kuvaavan pelaajan kykyä oppia mikä tahansa yksittäinen kirjain. Kirjainten sekoittumistodennäköisyys taas kuvaa erikoistilannetta, jossa pelaaja ei erota tiettyjä kahta kirjainta toisistaan. Aloittaessaan kirjainten opettelun lapsi hyvin todennäköisesti sekoittaa useat kirjaimet keskenään, ja tällainen tilanne haluttiin tulkita ennemmin siten, ettei pelaaja osaa kirjainta. Lukutaidon kehittyessä kehittyy myös taito erottaa kirjaimet ja äänteet toisistaan.

6.4 Laskennan optimointi

Laskenta toteutettiin Python-ohjelmointikielellä käyttäen apuna erityisesti matriisilaskentaa helpottavaa Numpy-kirjastoa. Laskennan nopeuttamiseksi toteutin C-kielellä kirjoitetun Python-moduulin, joka laski iteraatioiden sisimmän osuuden eli kirjaimen osaamistilan todennäköisyydet arvoille 0 ja 1, kun muut arvot pysyivät vakioina. Laskenta-aika tippui C-kielellä kirjoitettua moduulia käyttämällä alle viidesosaan pelkästään Python-kielellä kirjoitettuun ohjelmaan verrattuna.

Trialien vastaustodennäköisyydet laskettiin myös etukäteen eri häiriöarvoille. Yksittäisen trialin vastauksen todennäköisyys saatiin indeksoimalla trialin vastausten perusteella. Indeksointi tapahtui järjestämällä kohdeärsyke ensimmäiseksi ja merkitsemällä todennäköisiä vastausvaihtoehtoja 1:llä ja epätodennäköisiä 0:lla. Olkoon esimerkiksi pelaajan kirjainten osaamistilat kirjaimille E ja K $\{E = 0, K = 1\}$, ja pe-

Vaihtoehdot			Indeksi	Valintatodennäköisyys		
	1	0	2		0,9725	0,025
	1	1	3		0,5	0,5
1	0	0	4	0,9833...	0,0166...	0,0166...
1	0	1	5	0,49166...	0,49166...	0,0166...
1	1	0	6	0,49166...	0,0166...	0,49166...
1	1	1	7	0,33...	0,33...	0,33...

Taulukko 9. Vastaustodennäköisyydet kahden ja kolmen kirjaimen trialeille.

laajalle esitetty triali $t = \{A, E, K\}$, jossa kysytään kirjainta A. Kun yhdessä iteraatiossa laskettiin osaamistodennäköisyyksiä trialissa esitetylle kirjaimelle K, valittiin listasta molempia K:n osaamistiloja $\{A = 1, E = 0, K = 1\}$ ja $\{A = 1, E = 0, K = 0\}$ vastaavat vaihtoehdot $\{1,0,0\}$ ja $\{1,0,1\}$. Seuraavaksi laskettiin vaihtoehdoille indeksit $\{1,0,0\} => 100_2 = 4_{10}$ ja $\{1,0,1\} => 101_2 = 5_{10}$, ja saaduista indekseistä valittiin todennäköisuuslistat, jotka kertovat todennäköisyydet eri vastauksille ottaen huomioon kirjainten osaamistilat. Taulukossa 9 on esitetty todennäköisyydet vastausvaihtoehdoille kahden ja kolmen kirjaimen trialeissa, kun häiriöarvo $\delta = 0{,}05$.

6.5 Otannan koko

Jokaista mallia kohden otannan koko pidettiin samana. Simuloinnin pituuteen vaikuttaa mm. estimoitavien parametrien määrä. Koska mallissa, jossa kirjaimet voivat sekoittua on enemmän parametreja, tarvitaan myös pidempi otanta. Tämä johtuu siitä, että lähtötilanne on satunnaisesti valittu, jolloin mallilla kestää hetki konvergoitua todennäköisimpään tilaan. Tätä kutsutaan sisäänajoksi. Tässä tapauksessa sisäänajojakso oli 50 iteraatiota yksinkertaisemmalle mallille ja 100 sekoittumismallille.

Koska yksittäisen iteraation laskentaan kuluu paljon aikaa ja malli konvergoituu hitaasti, jokaiselle pelaajalle laskettiin yksi otos. Kappaleessa 7.4 kuvataan tästä seuranneita ongelmia.

6.6 Kuvaajat

Pelaajan mallinmukaista osaamista haluttiin tarkastella visuaalisesti. Toteutin tulosten esittämiseksi kuvaajan, josta näkee pelaajan kirjainten osaamistilat eri pelikerroilla. Kuvaajissa on y-akselilla esitetty pelissä esiintyneet kirjaimet ja x-akselilla pelaajan pelikerrat. Pelikertojen pituus kuvissa vaihtelee pelikerralla pelattujen trialien määrän mukaan suhteessa trialien kokonaismäärään. Saadusta MCMC-otoksesta laskettiin jokaiselle tilalle keskiarvot, joiden perusteella kirjainten laatikot eri pelikerroilla värjättiin. Kuvaajiin merkittiin jokaisen kirjaimen tiloista punaisella alle 0,2:n keskiarvot ja vihreällä yli 0,8:n keskiarvot. Värin peittävyys kuvaa kuinka lähellä nollaa tai yhtä keskiarvo on. Kuvassa 7 on esimerkki mallista, jossa ei ole mukana kirjainten sekoittumista.

Kuvaajista haluttiin nähdä myös mitkä kirjaimet pelaaja sekoittaa keskenään. Kirjainparien sekoittumista kuviin piirrettiin nuolet parien välille, joiden keskiarvo oli yli 0,8. Kuva 8 on esimerkkikuvaaja mallista, jossa pelaaja voi sekoittaa kirjaimet.

Saatuja kuvaajia verrattiin pelaajadataan sekä muihin Ekapelin analysointimenetelmiin. Sekoittuneet kirjaimet olivat myös viuhkakuvissa sellaisia, jotka pelaajien oli hankala erottaa toisistaan. Kuvissa oli selkeitä yhteneväisyyksiä viuhkakuvaajiin joidenkin sekoittuvien kirjainten kohdalla, mutta ei kaikkien. Joissain tapauksissa kaksi kirjainparia näytti menevän viuhkakuvaajien perusteella yhtä todennäköisesti sekaisin, mutta mallin mukaan pelaaja sekoitti vain toisen kirjainparin keskenään.

7 Oppimismallin vastaavuus pelaajadataan

Tässä luvussa esitellään simuloinnin avulla laskettujen osaamistodennäköisyyksien sopivuutta pelaajadataan. Pelaajadatan sopivuutta testattiin kahdella erilaisella mallilla: ensimmäinen yksinkertainen malli ilman kirjainten sekoittamista ja toinen, jossa pelaaja saattoi sekoittaa kirjaimet keskenään. Pelaajien datalle laskettiin eri versioiden mukaiset todennäköisyydet, joilla pelaaja osaa kunkin kirjaimen. Tämän jälkeen laskettiin kuinka hyvin saadut mallit sopivat pelaajasta saatuun dataan. Tarkoituksena oli laskea todennäköisyydet jokaiselle pelaajalle, mutta laskennallisista ongelmista johtuen päädyttiin tarkastelemaan tuloksia pelaajakohtaisesti.

Vastauksena ensimmäiseen tutkimuskysymykseen, kuvattu malli sopii pelaajan oppimisen mallintamiseen ja sen avulla saadaan pelaan osaamisesta ja oppimisesta tietoa. Yksinkertaisen version ja kirjainten sekoittumimallin sopivuuden vertaaminen vaatisi kuitenkin suuremman pelaajamäärän tarkastelua, joka on tästä tutkimuksesta jätetty laskennallisista ongelmista johtuen.

7.1 Pelaajadatan vaikutus mallin sopivuuteen

Pelin tasoadaptaation vaikutusta pelaajadataan on vaikea arvioida. Kirjainten esitysjärjestys tasoadaptaatiossa on aina sama, jolloin pelaaja näkee aina ensimmäisenä kirjainparin A ja I. Tällöin ensimmäisenä esitetyistä kirjaimista kertyy enemmän dataa kuin myöhemmin esitettävistä kirjaimista. Pelin adaptaation vaikutusta siihen, kuinka malli sopii pelaajadataan, on hankala arvioida. Pelissä esitetyt sisällöt määräytyvät pelaajakohtaisesti, joten toistuvuuksien löytäminen tässä esitetyn mallin ja adaptaation väliltä vaatisi suuremman otoksen tarkastelua.

7.2 Mallin sopivuus pelaajadataan

Testaus suoritettiin laskemalla p-arvot valituille pelaajille. Laskentaa toteutettiin luomalla jokaiselle MCMC-otokselle replikaattidata, jossa trialit pysyivät samoina, mutta trialeille arvottiin uudet vastaukset mallin antamien osaamistodennäköisyyksien perusteella. Olkoon esimerkiksi pelaajan osaamistodennäköisyydet kirjaimille jollain pelikerralla $p(\theta_A = 1) = 0{,}8, p(\theta_U = 1) = 0{,}6$ ja kysyttävä kirjain A. Nyt todennäköisyys sille, että pelaaja osaa vähintään toisen kirjaimen $0{,}8 * 0{,}6 + 0{,}8 * 0{,}4 + 0{,}2 * 0{,}6 = 0{,}92$, ja todennäköisyys, että pelaaja ei osaa kumpaakaan, on $0{,}2 * 0{,}4 = 0{,}08$. Tällöin, ottaen huomioon häiriöarvon, todennäköisyys valita kirjain A on $0{,}92 * (1 - \delta) + \delta * 0{,}5 + 0{,}08 * 0{,}5$ ja todennäköisyys valita U $0{,}08 * 0{,}5 + \delta * 0{,}5$. Näiden replikaattien pohjalta laskettiin todennäköisyydet, joilla replikaattidata sopisi malliin paremmin kuin alkuperäinen pelaajadata.

Todennäköisyys sille, että replikaattidata sopii paremmin kuin alkuperäinen data, saadaan kaavalla

$$Pr(T(y_{\mathtt{rep}}, \theta) \geq T(y, \theta) \mid y),$$

jossa tilastollinen poikkeavuus (deviance) on $T(y, \theta) = -2ln(p(y \mid \theta))$.

Jotta mallien välisestä paremmuudesta voisi sanoa tarkemmin, tulisi p-arvot laskea kaikille pelaajille ja verrata kumpi malleista sopii paremmin pelaajista saatuun dataan. Tämä vaatisi huomattavasti laskentatehoja ja -aikaa.

7.3 Variaatioita malleista

Mallista kokeiltiin myös versiota, jossa häiriöarvot lasketaan pelaajakohtaisesti. Pelaajakohtainen häiriöarvo ei juurikaan parantanut mallin sopivuutta pelaajadataan, joten häiriöarvo on perusteltua pitää vakiona.

Lisäksi kokeiltiin mallia, jossa kirjainten osaamistilojen siirtymätodennäköisyydet määräytyvät pelaajakohtaisesti. Pelaajakohtaiset siirtymämatriiseja kokeiltiin kirjainkohtaisina ja yhtenäisenä kaikille kirjaimille. Kumpikin lähestymistapa osoittautui ongelmalliseksi, kuten seuraavassa kappaleessa kuvataan. Parempi lähestymistapa häiriöarvon ja siirtymämatriisien arvojen suhteen on laskea pienestä satunnaisesta pelaajaotoksesta sopivimmat arvot ja käyttää niitä vakioina.

7.4 Mallien ongelmia

Suurin ongelma kirjainten sekoittamismallissa, verrattuna yksinkertaiseen malliin, oli sen vaatima laskenta-aika. Eräs tapa yksinkertaistaa kirjainten sekoittumista olisi, että se sallittaisiin vain kirjainpareille, jotka yleensä sekoitetaan helposti. Tämä vähentäisi huomattavasti mallissa olevien parametrien määrää ja laskenta-aikaa. Haittapuolena tällaisessa ratkaisussa olisi pelaajasta mahdollisesti menetettävä tieto, mikäli keskenään sekoittuva kirjainpari ei ole valitussa joukossa.

Mallissa haluttiin, että on todennäköisempää, ettei pelaaja osaa kirjaimia kuin että kirjaimet sekoittuvat keskenään. Yksittäisen trialin kohdalta tilannetta ei aina voi erottaa, sillä todennäköisyydet valita väärä vaihtoehto trialissa, jossa ei ole muita tuntemattomia kirjaimia, ovat samat silloin, kun pelaaja ei osaa kirjainta ja silloin, kun se sekoittuu toiseen kirjaimeen.

Kuvissa 9 ja 10 on laskettu samalle pelaajalle kaksi eri otantaa, jotka ovat lähteneet eri alkupisteistä. Ensimmäisessä tapauksessa 9 malli on konvergoitunut tilaan, jossa pelaaja osaa ensimmäisellä pelikerralla esitetyt kirjaimet, mutta sekoittaa kirjaimet E ja R keskenään. Toisessa tapauksessa kuvassa 10 malli on konvergoitunut tilaan, jossa pelaaja ei osaa kirjaimia E ja R.

Koska kuvissa 9 ja 10 esitetty pelaaja on mallin mukaan osannut muut kirjaimet, ovat trialikohtaiset todennäköisyydet olleet samoja. Näin ollen mallin tulisi päätellä sellaisten trialien pohjalta, joissa E ja R-kirjaimet ovat esiintyneet muiden kirjainten kanssa. Pelaajalle on esitetty pelikerralla yli sata trialia, mutta malli päätyy siltikin eri tilanteisiin eri aloituspisteistä. Useisiin eri tiloihin konvergoitumista voisi korjata sillä, että pelaajalle laskettaisi useita otantoja eri alkutiloista. Lopullinen otos pelaajalle olisi näiden otantojen keskiarvo tai se tila, johon malli konvergoituu useimmiten. Tällainen lähestymistapa vaatisi moninkertaisen laskenta-ajan, mutta antaisi stabiilimpia tuloksia.

Mallista kokeiltiin myös versiota, jossa häiriöarvo ei ole vakio. Tällaisen mallin ongelmaksi muodostui kuitenkin, ettei häiriöarvo määräydy tarkkaan. Häiriöarvon laskenta toteutettiin valitsemalla sallittu vaihteluväli 1-10% ja asettamalla sallituiksi arvoiksi lukuvälin arvot 0,1 % välein. Häiriöarvo diskretisoitiin, jotta luvussa 6.4 esitetyt trialien vastaustodennäköisyydet voitiin laskea etukäteen eri häiriöarvoille. Jatkuvan häiriöarvon käyttö olisi vaatinut todennäköisyyksien laskentaa joka iteraatiolla ja näin kasvattanut laskenta-aikaa huomattavasti. Diskretisoidun häiriöarvon laskeminen lisää laskentaan kuluvia iteraatioita suoraan suhteessa häiriöarvon mahdollisten arvojen lukumäärään nähden.

Samoin mallissa, jossa siirtymämatriisit määritettiin pelaaja- ja kirjainkohtaisiksi, siirtymätodennäköisyydet eivät määräytyneet tarkkaan edes pelaajille, joilla oli yli 30 pelikertaa. Tämä johtunee siitä, että yksittäisten kirjainten osalta dataa ei saada riittävästi. Mallissa, jossa pelaajakohtaisesti on kaikille kirjaimille yhtenäinen siirtymämatriisi, siirtymätodennäköisyydet saattoivat olla hyvin lähellä yhtä. Tämä johtuu siitä, että kirjaimet, jotka pelaaja osaa hyvin, eivät muuta tilaansa. Tällöin mallin mukaan on myös epätodennäköistä, että muiden kirjainten tilat muuttuisivat.

Mielestäni paras vaihtoehto kirjainten siirtymille on laskea pienestä otannasta sopivat arvot, joita käytetään kaikille pelaajille vakioina. Tällöin malli pysyisi yksinkertaisena ja helpommin ymmärrettävänä. Yksittäisen pelaajan kohdalla on mielestäni kiinnostavampaa tietää kokonaiskyvystä oppia uusi kirjain. Kaikkien pelaajien tietoja tutkiessa voisi kuitenkin olla kiinnostavaa kokeilla mallia, jossa olisi kirjainkohtaiset siirtymämatriisit, jotka olisivat yhteisiä kaikille pelaajille. Näin voisi saada tietoa siitä, kuinka hankalaa jonkin tietyn kirjaimen oppiminen pelaajille on.

8 Adaptaation toteutus mallin pohjalta

Tässä kuvussa esitellään kuinka mallin pohjalta voisi luoda adaptaation Ekapeliin. Lyhyesti toiseen tutkimuskysymykseen vastaten, malli ei suoraan sovi adaptaatiossa käytettäväksi, mutta siitä saatavaa tietoa voisi hyödyntää adaptaation toteutuksessa eri tavoin. Kappaleessa 8.1 esitellään eri versioiden sopivuutta adaptaation toteutukseen, ja kappaleessa 8.3 pohditaan mitä ongelmia adaptaation toteutuksessa ja toiminnassa voisi tulla.

Oppimispeli, joka toimii aina samalla tapaa, ei välttämättä ole pelaajan oppimisen kannalta hyvä. Adaptiivinen oppimispeli mukautuu pelaajan vastausten mukaan. Adaptiivisen pelin hyötynä on se, että vaikeustaso voidaan määrätä pelaajan osaamisen mukaan. Jos peli on liian vaikea tai helppo, pelaaja turhautuu ja pelaajan motivaatio laskee. Adaptaation tavoitteena on opettaa halutut kohteet pelaajalle mahdollisimman tehokkaasti. Tehokkuuteen vaikuttavat mm. kirjainten esiintymisjärjestys, trialin koko ja pelaajan motivaatio. Jos opetettava kirjain esitetään jo osatun kirjaimen tai kirjainten kanssa, pelaaja todennäköisemmin vastaa oikein kuin tuntemattomien kirjainten kanssa esitettäessä. Tällöin ei kuitenkaan saada tietoa siitä, osaako pelaaja kirjaimen oikeasti, vai osaako pelaaja sulkea varmasti väärät vaihtoehdot, eli jo osaamansa kirjaimet vastausvaihtoehdoista, ja näin ollen valita ainoan jäljellejääneen vaihtoehdon.

8.1 Adaptaation valinta

Adaptaation valinnassa ei kuitenkaan riitä se, että pelaajasta saadaan mahdollisimman paljon tietoa. Pelin tulee myös olla pelaajan kannalta motivoiva. Jos esimerkiksi pelaajalle esitettäisi useampi tuntematon kirjain samassa kentässä, saattaisi se pelaajan kannalta vaikuttaa liian hankalalle. Mallin perusteella pelkästään jo osattujen kirjainten näyttäminen uudestaan ei tuottaisi uutta tietoa.

8.2 Pelaajasta saatavan tiedon laskenta

Kun pelaajan oletetaan toimivan mallin mukaisesti, voidaan jokaiselle esitettävälle trialille laskea siitä saataava tietomäärä. Esitettävä triali voidaan täten valita joko maksimoimalla trialista saatava tiedon määrä, tai pelaajaystävällisesti, jolloin tiedon määrä suhteutetaan todennäköisyyteen, jolla pelaaja vastaa väärin. Informaation määrän laskentaa on kuvattu tarkemmin julkaisuissa Kujala (2008) ja Kujala, Richardson ja Lyytinen (2010a).

Tiedon määrä yksittäiselle trialille saadaan kaavalla

$$I(R_x;\Theta \mid y) = \int\int p(r_x,\theta|y) log \frac{p(r_x,\theta|y)}{p(r_x|y)p(\theta|y)}$$

(Kujala, Richardson ja Lyytinen 2010a, s. 3), jossa R_x kuvaa seuraavan esitettävän trialin satunnaista tulosta.

Pelaajaystävällisessä versiossa saatu informaatio suhteutetaan todennäköisyydellä, että pelaaja vastaa väärin. Väärän vastauksen todennäköisyys toimii näin esitettävän trialin hintana. Trialista saatava informaatio voidaan pelaajaystävälliselle versiolle laskea kaavalla

$$I_F(R_x;\Theta|y) = \frac{I(R_x;\Theta|y)}{E(C_x|y)}$$

(Kujala, Richardson ja Lyytinen 2010a, s. 5), jossa C_x kuvaa väärää vastausta seuraavalle trialille. Odotusarvo väärän vastauksen todennäköisyydelle saadaan kaavalla

$$E(C_x|y) = \int p(R_x = 0|\theta)p(\theta|y)d\theta,$$

jossa $p(R_x = 0|\theta)$ kuvaa todennäköisyyttä, että pelaaja vastaa väärin esitettyyn triaaliin. Trialista saatava informaatio lasketaan tässä esitetylle mallille laskemalla keskiarvot MCMC-otoksesta, jolloin saadaan todennäköisyydet. joilla pelaaja osaa kunkin kirjaimen.

8.3 Adaptaation toteutuksessa huomioitavaa

Tässä osiossa kerrotaan, kuinka valittu adaptaatio voitaisi toteutetaan osaksi ekapeliä, ja mitä adaptaation toteutuksessa tulisi ottaa huomioon.

Esitellyt mallit sopivat paremmin datan tarkasteluun kuin adaptaation luomiseen, sillä MCMC-otoksen laskentaan kuluu huomattavasti aikaa. Mallien tuntemattomien muuttujien määrät ovat melko suuria, ja täten adaptaatiolla saattaa kestää hyvinkin kauan ennen kuin pelaajakohtaiset muuttujat ovat määräytyneet tarkasti. Tätä ennen adaptaatio saattaisi vaikuttaa näennäisesti mielivaltaiselle, eikä se välttämättä innostaisi pelaajaa pelaamaan. Lisäksi suuri määrä tuntemattomia muuttujia vaatii paljon laskenta-aikaa, jolloin toteutettu adaptaatio ei toimisi tarpeeksi nopeasti. Pelaajan ei haluta joutuvan odottamaan trialin jälkeen seuraavaksi esitettävää trialia, koska peliaika halutaan käyttää mahdollisimman hyödyllisesti.

Edellä mainitut ongelmat voidan korjata korvaamalla osa muuttujista, kuten siirtymätodennäköisyydet, peliä jo pelanneiden lasten pelaajadatoista lasketuilla ja valituilla vakioilla. Tällöin adaptaation toimintaa voidaan ennustaa huomattavasti paremmin kuin erittäin monimuuttujaista mallia käyttämällä. Laskenta-aikaan tämä ei vielä välttämättä auta, sillä pelaajan mahdollisten eri tilojen määrä kasvaa eksponentiaalisesti uusia kirjaimia esitettäessä, samoin kuin mahdollisten esitettävien trialien määrä. Lisäksi vakioiden tulee olla huolellisesti ja perustellusti valittuja pelaajista saadusta datasta.

Adaptaation toimintaa voi nopeuttaa esimerkiksi jakamalla kirjaimet eri ryhmiin ja rajoittamalla trialeissa esitettävien ärsykkeiden määrää. Vaihtoehtoisesti valintafunktiota voidaan muokata siten, ettei ole tarpeen laskea jokaisen trialin antamaa informaatiomäärää. Tällaisen valintafunktion voisi toteuttaa esimerkiksi laskemalla saatava tietomäärä vain satunnaisesti valituista trialeista ja valitsemalla niistä kaikkein eniten halutulla menetelmällä tietoa antava. Valintaa voisi vielä tarkentaa arpomalla satunnaisia trialeja, jotka poikkeavat hyvin vähän edellisellä kertaa valitusta trialista, ja valitsemalla näistä trialeista taas eniten tietoa antava.

Yksinkertainen vaihtoehto olisi käyttää hyödyksi vain todennäköisyyksiä sille, miten hyvin pelaaja osasi kirjaimet viimeisimmällä pelikerralla, ja jättää mallista huomiotta meneillään oleva pelikerta. Tällöin malli ei hidastaisi pelin toimintaa, sillä tarvittavat todennäköisyydet voisi laskea pelikertojen välissä. Pelaajalle ei myöskään näytettäisi niin paljon pelaajan osaamia kirjaimia.

9 Yhteenveto

Tässä tutkielmassa kuvattiin eräs malli siitä, kuinka oppimispeli Ekapelin pelaajan oppimista voisi kuvata. Mallin avulla tutkittiin, kuinka pelaaja oppii kirjaimet ja kuinka hyvin pelaaja osaa kirjaimet eri pelikerroilla. Tutkielmassa tarkasteltiin kuinka hyvin malli sopisi kuvaamaan pelaajan toimintaa ja osaamista, ja kuinka mallin pohjalta voisi toteuttaa adaptaation Ekapeliä varten.

Adaptaation luonti mallin pohjalta osoittautui monimutkaiseksi, ja vaatisi useita kompromisseja ja yksinkertaistuksia toimiakseen tarpeeksi nopeasti. Mallin avulla saatavaa tietoa voisi kuitenkin käyttää osana adaptaatiota.

Pelitulosten visualisointiin malli sopii mielestäni hyvin. Ongelmaksi nousee lähinnä otoksen laskentaan kuluva aika ja vaadittava laskentateho. Kuvaajien avulla saa nopeasti tietoa pelaajan osaamisesta, ja kuinka osaaminen on pelaamisen aikana kehittynyt. Malli myös tunnistaa tilanteet, joissa pelaaja osaa kirjaimen muuten, mutta sekoittaa sen, kun se esitetään jonkin tietyn kirjaimen kanssa.

Tässä tutkielmassa käsiteltiin suomenkielisestä pelistä kerättyä dataa. Olisi kiinnostava verrata kuinka eri kielisistä peliversioista lasketut tulokset eroavat. Kirjaimia vastaavat äänteet ovat erilaisia eri kielille, jonka vuoksi myös toisiltaan kuulostavat ja keskenään sekoittuvat kirjaimet vaihtelevat kielikohtaisesti. Olisi kiinnostava nähdä, voidaanko kerätystä datasta tässä esitetyn mallin avulla löytää kielikohtaisia eroavaisuuksia.

Vastaavaa oppimismallia voisi hyödyntää myös muissa peleissä, jotka pytkivät luomaan suoria assosiaatioita esitettyjen kohteiden väleille. Olisi myös kiinnostava tutkia, kuinka kappaleessa 5.1 kuvattu kolmitilainen oppimismalli sopisi pelaajadataan ja eroaisi tässä esitetystä kaksitilaisesta mallista.

Lähteet

Atkinson, Richard C, ja Edward J Crothers. 1964. “A comparison of paired-associate learning models having different acquisition and retention axioms”. *Journal of Mathematical Psychology* 1 (2): 285–315.

Béguin, Anton A, ja Ceec AW Glas. 2001. “MCMC estimation and some model-fit analysis of multidimensional IRT models”. *Psychometrika* 66 (4): 541–561.

Bernardo, José M, ja Adrian FM Smith. 2009. *Bayesian theory.* Nide 405. John Wiley & Sons.

Calfee, Robert C, ja Richard C Atkinson. 1965. “Paired-associate models and the effects of list length”. *Journal of Mathematical Psychology* 2 (2): 254–265.

Cappé, Olivier, Eric Moulines ja Tobias Rydén. 2009. “Inference in hidden markov models”. Teoksessa *Proceedings of EUSFLAT Conference,* 14–16.

Chant, Verne G, ja Richard C Atkinson. 1973. “Optimal allocation of instructional effort to interrelated learning strands”. *Journal of Mathematical Psychology* 10 (1): 1–25.

Ehri, Linnea C, Simone R Nunes, Steven A Stahl ja Dale M Willows. 2001. “Systematic phonics instruction helps students learn to read: Evidence from the National Reading Panel’s meta-analysis”. *Review of educational research* 71 (3): 393–447.

Eisner, Jason. 2002. “An interactive spreadsheet for teaching the forward-backward algorithm”. Teoksessa *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1,* 10–18. Association for Computational Linguistics.

Fletcher, JD, ja Richard C Atkinson. 1972. “An evaluation of the Stanford CAI program in initial reading (Grades K through 3)”. *Journal of Educational Psychology* 63 (6).

Gamerman, Dani, ja Hedibert F Lopes. 2006. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference.* Chapman / Hall/CRC.

Gelman, Andrew, John B. Carlin, Hal S. Stern ja Donald B. Rubin. 2004. *Bayesian data analysis.*

Grinstead, Charles M, ja James Laurie Snell. 2012. *Introduction to probability.* American Mathematical Soc.

Groen, Guy J, ja Richard C Atkinson. 1966. "Models for optimizing the learning process." *Psychological Bulletin* 66 (4): 309.

Kujala, Janne V. 2008. *Bayesian adaptive estimation under a random cost of observation associated with each observable variable.* University of Jyväskylä.

Kujala, Janne V, Ulla Richardson ja Heikki Lyytinen. 2010a. "A Bayesian-optimal principle for learner-friendly adaptation in learning games". *Journal of Mathematical Psychology* 54 (2): 247–255.

———. 2010b. "Estimation and visualization of confusability matrices from adaptive measurement data". *Journal of Mathematical Psychology* 54 (1): 196–207.

Lyytinen, Heikki, Miia Ronimus, Anne Alanko, Anna-Maija Poikkeus ja Maria Taanila. 2007. "Early identification of dyslexia and the use of computer game-based practice to support reading acquisition". *Nordic Psychology* 59 (2): 109–126.

Meyn, Sean P, ja Richard L Tweedie. 2012. *Markov chains and stochastic stability.* Springer Science & Business Media.

Mislevy, Robert J. 1986. "Bayes modal estimation in item response models". *Psychometrika* 51 (2): 177–195.

Neal, Radford M. 1993. "Probabilistic inference using Markov chain Monte Carlo methods".

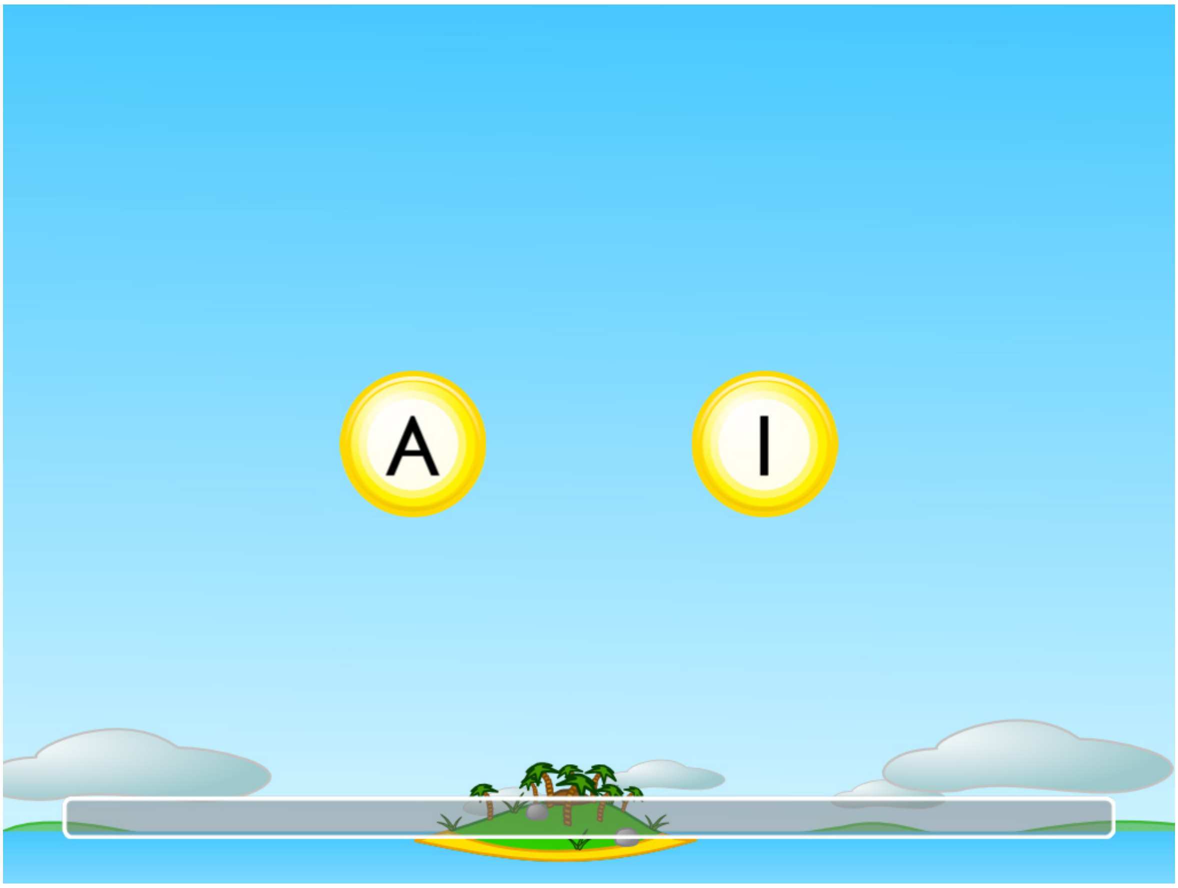
"Lukimat-palvelu". 2019. Niilo Mäki -Instituutti. Viitattu 13. huhtikuuta 2019. `http://www.lukimat.fi`.

Rabiner, Lawrence R. 1989. “A tutorial on hidden Markov models and selected applications in speech recognition”. *Proceedings of the IEEE* 77 (2): 257–286.

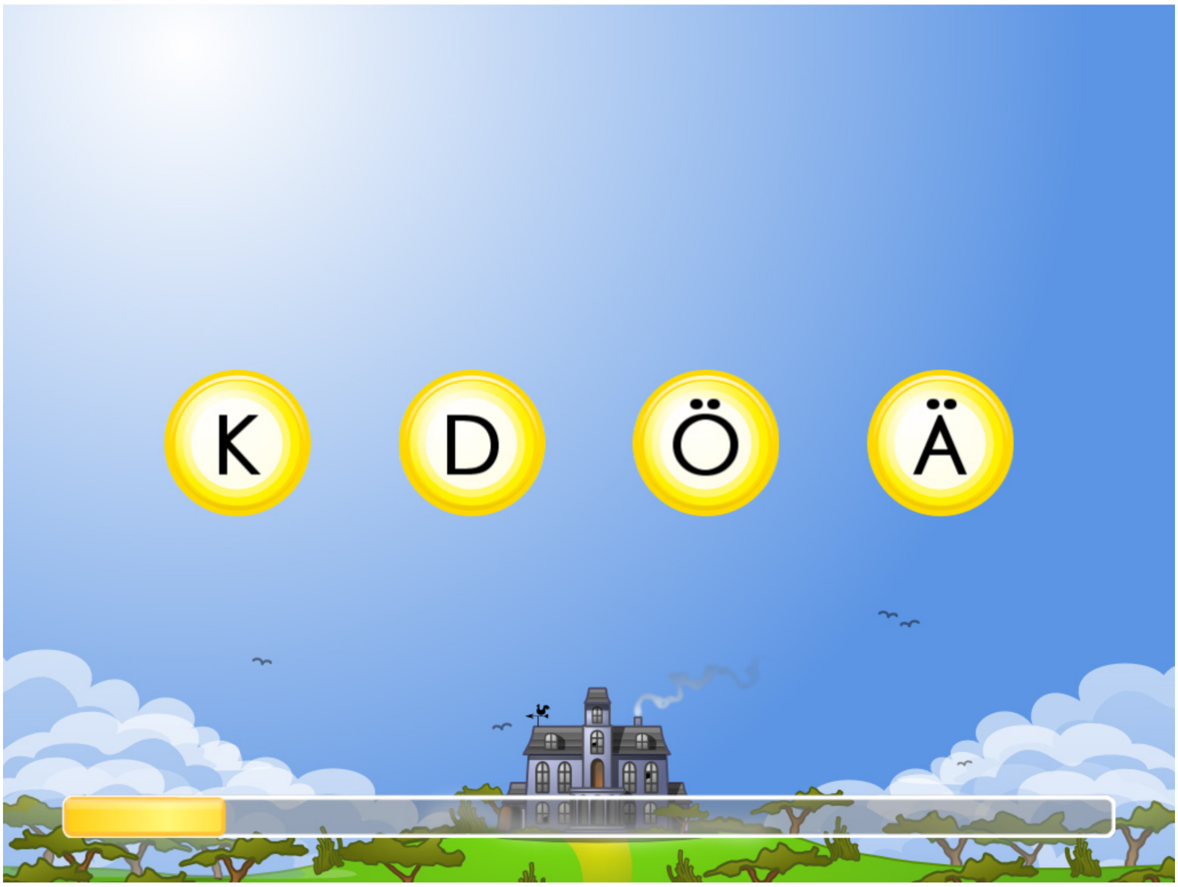
Richardson, Ulla, ja Heikki Lyytinen. 2014. “The GraphoGame method: The theoretical and methodological background of the technology-enhanced learning environment for learning to read”. *Human Technology* 10 (1).

Ronimus, Miia, Janne Kujala, Asko Tolvanen ja Heikki Lyytinen. 2014. “Children’s engagement during digital game-based learning of reading: The effects of time, rewards, and challenge”. *Computers & Education* 71:237–246.

Rubinstein, Reuven Y, ja Dirk P Kroese. 2016. *Simulation and the Monte Carlo method.* Nide 10. John Wiley & Sons.



Kuvio 1. Ruutukaappaus Ekapelistä. (“Lukimat-palvelu” 2019)



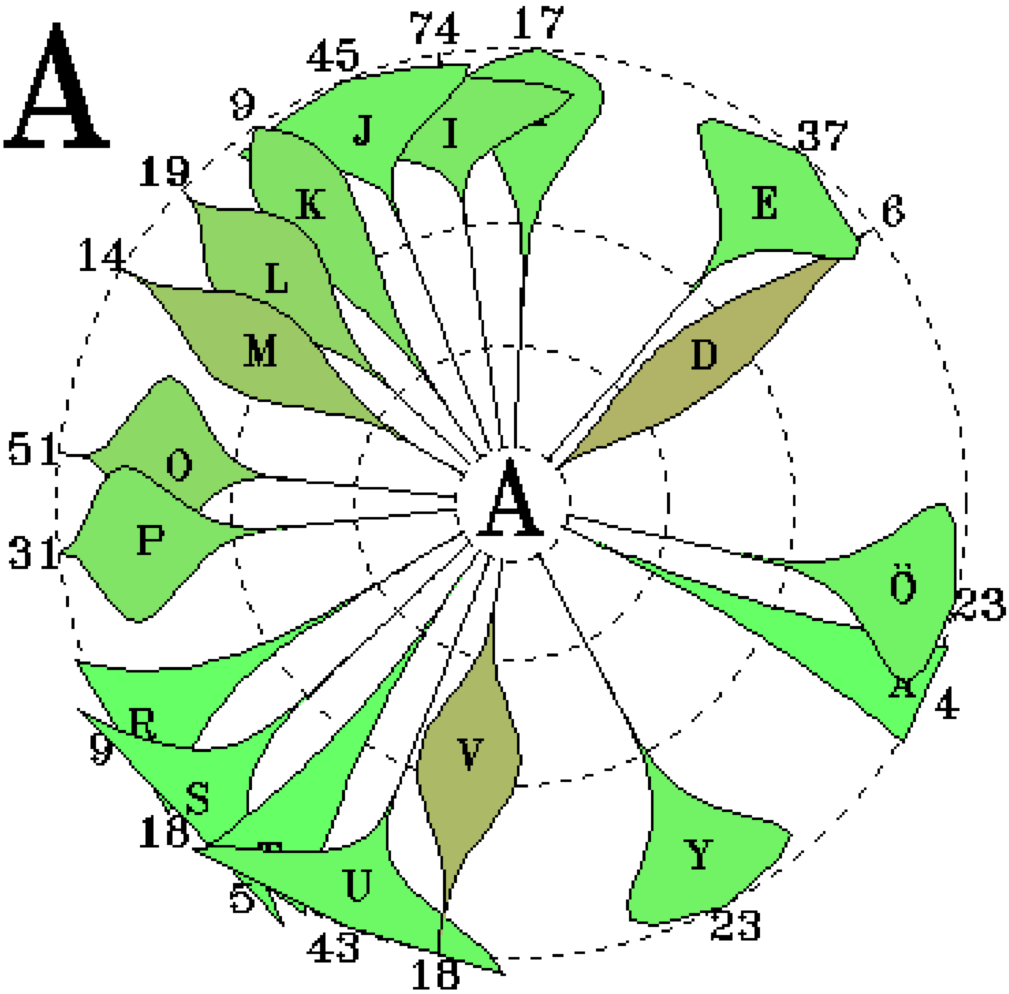
Kuvio 2. Ruutukaappaus, jossa esitetään neljän ärsykkeen triali. ("Lukimat-palvelu" 2019)

	1	2	3
A	O	O	V
B	O	O	V
D	V	O	V
E	V	O	V
F	O	O	V
G	V	O	V
H	O	O	V
I	O	O	O
J	V	O	V
K	O	O	V
L	V	O	V
M	V	O	O
N	O	O	V
O	O	O	O
P	V	O	V
R	V	O	V
S	V	O	O
T	V	O	V
U	V	O	V
V	V	O	V
Y	O	O	V
Ä	V	O	V
Ö	O	O	V

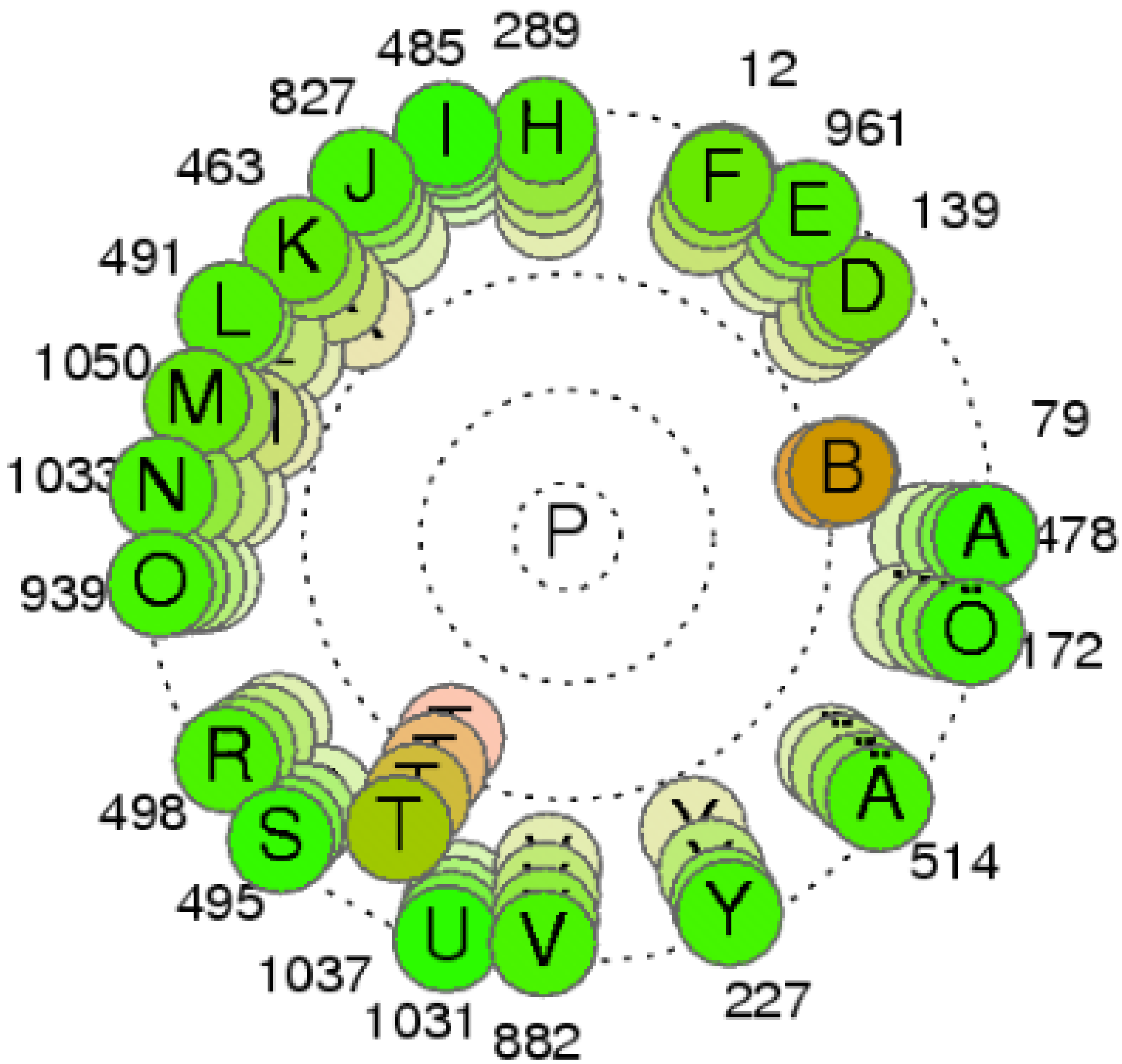
Kuvio 3. Testitulostaulukko. ("Lukimat-palvelu" 2019)

Kohdeärsyke	Esiintymisten lukumäärä	Ensimmäisten 7 trialin %	Viimeisten 7 trialin %	Kaikkien trialien %	
a	9	100%	100%	100%	—
b	8	86%	100%	88%	↑
B	8	71%	71%	75%	—
D	8	86%	86%	88%	—
f	8	86%	86%	88%	—
F	8	100%	100%	100%	—
G	8	71%	86%	75%	↑
H	8	86%	100%	88%	↑
i	9	100%	100%	100%	—
J	8	86%	86%	88%	—
k	8	86%	100%	88%	↑
L	12	71%	86%	83%	↑
l	12	71%	100%	83%	↑
m	16	29%	100%	63%	↑
M	12	71%	86%	75%	↑
n	17	71%	86%	76%	↑
N	12	71%	71%	75%	—
o	8	86%	100%	88%	↑
P	8	71%	71%	75%	—
r	8	100%	100%	100%	—
s	9	100%	100%	100%	—
t	17	100%	100%	100%	—
T	12	71%	86%	83%	↑
u	8	100%	100%	100%	—
v	16	86%	100%	94%	↑
V	12	71%	100%	83%	↑
Y	8	86%	86%	88%	—
Ö	8	100%	100%	100%	—
ä	8	100%	100%	100%	—

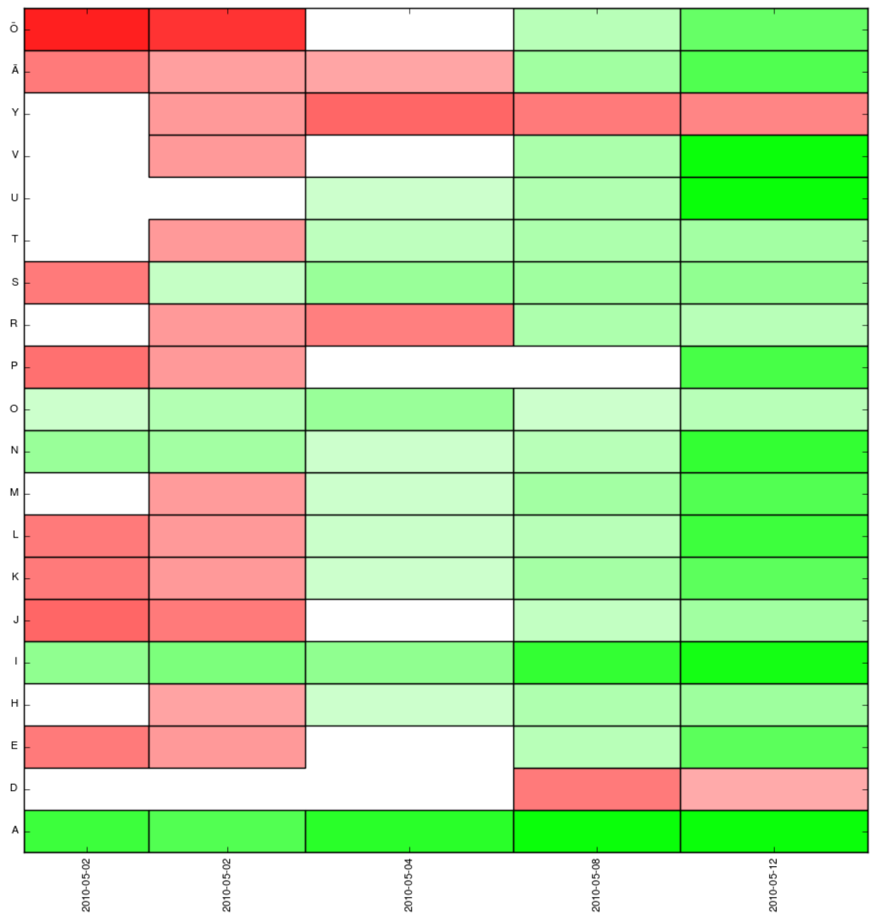
Kuvio 4. Tulostaulukko. (“Lukimat-palvelu” 2019)



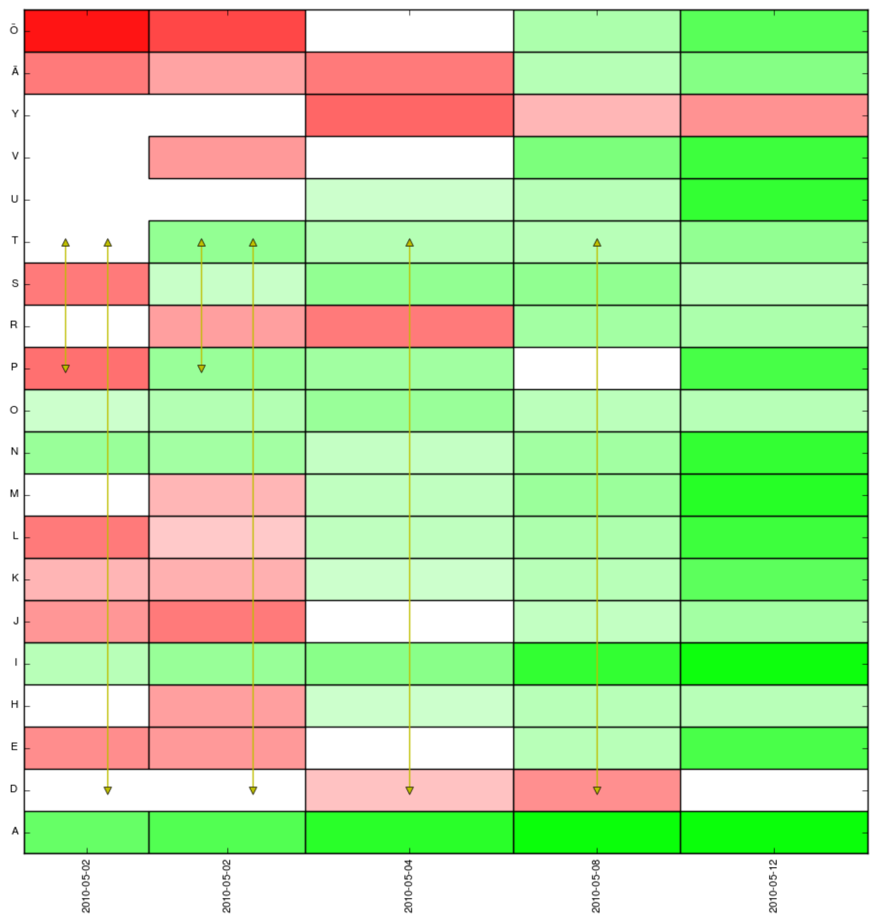
Kuvio 5. Viuhkakuvaaja kirjaimen A osaamisesta. (“Lukimat-palvelu” 2019)



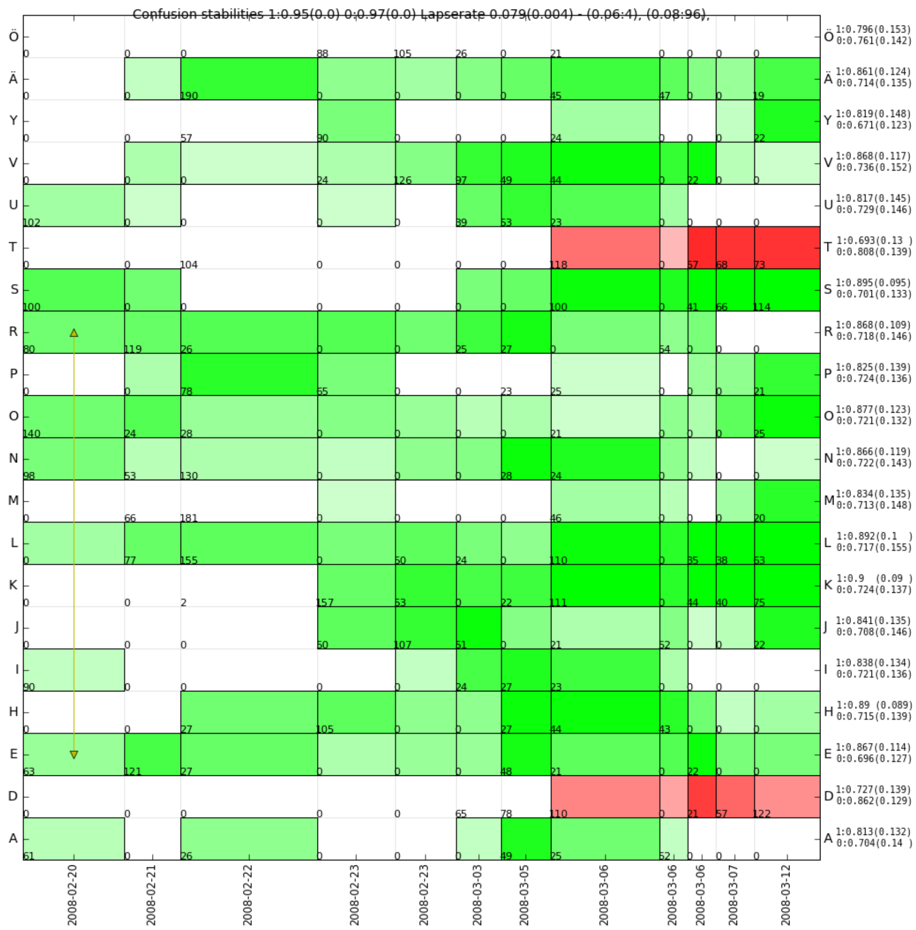
Kuvio 6. Viuhkakuvaaja kirjaimen P osaamisesta. (“Lukimat-palvelu” 2019)



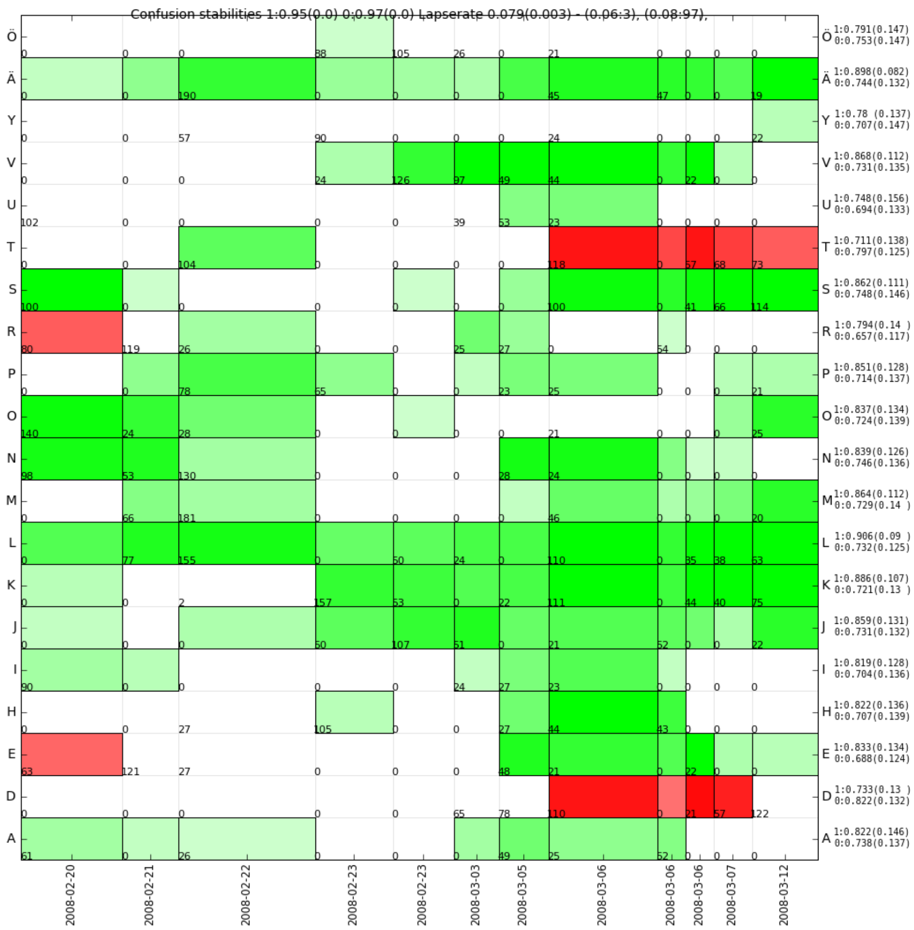
Kuvio 7. Esimerkkikuvaaja mallista, jossa ei ole mukana kirjainten sekoittumista.



Kuvio 8. Esimerkkikuvaaja mallista, jossa pelaaja voi sekoittaa kirjaimet keskenään.



Kuvio 9. Kuvaaja pelaajan osaamisesta, kun kirjaimet E ja R menevät pelaajalla sekaisin ensimmäisellä pelikerralla.



Kuvio 10. Kuvaaja pelaajan osaamisesta, kun pelaaja ei osaa kirjaimia E ja R ensimmäisellä pelikerralla.

Liitteet

Liite Ekapelisanasto

Ekapelin käsitteiden määrittely

Termi	in English	Selitys
Adaptaatio	Adaptation	Sääntö- tai laskentapohjainen itemien esittämisjärjestys ja -tapa, johon pelaajan pelaamisen edetessä tekemät valinnat vaikuttavat. Adaptaation vaihtelemat tekijät ovat muun muassa vaihtoehtojen määrä, pallojen putoamisnopeus, sisältöjen laatu tai vaikeusaste.
Altistuminen (altistus)	Exposure	Ärsykkeen visuaalinen tai auditiivinen havaitseminen, jolloin pelaaja aktiivisesti harjoittelee(=tekee valinnan) visuaalisten ja auditiivisten kielellisten yksikköjen välisiä vastaavuuksia.
Altistumisaika (altistusaika)	Exposure time	Altistumisen kesto alkaen ärsykkeen auditiivisesta esittämisestä ja päättyen trialin **viimeisen aktiivisen valinnan** tekemiseen. Vrt. vastausaika. Vrt. esittämisaika Ks. kokonaisaltistumisaika.
Altistumiskerta (altistuskerta)	Single exposure	Yksi ärsykkeelle altistuminen alkaen ärsykkeen auditiivisesta esittämisestä ja päättyen trialin viimeisen valinnan tekemiseen. (Esiintyy vain peruspeli –tehtävätyypin harjoittelu- ja lisäharjoittelu –kenttätyypeissä.
Arviointikenttä[1]	Evaluation task	Pelin kenttätyyppi, jossa pelaajan taitotasoa mitataan. (Myös vanhoissa peleissä olevat testikentät lasketaan arviointikentiksi).
Assosiaatio	Association	Kirjoitetun ja puhutun kielen väliset yhteydet, joita Ekapelissä harjoitellaan.
Esittäminen (altistaminen)	Presentation	Ärsykkeen visuaalinen tai auditiivinen **esiintyminen**, jolloin pelaajalla on mahdollisuus harjoitella visuaalisten ja auditiivisten kielellisten yksiköiden välisiä vastaavuuksia.
Esittämisaika (altistamisaika)	Presentation time	**Esittämisen kesto** alkaen ärsykkeen auditiivisesta esiintymisestä ja päättyen trialin päättymiseen (esim. oikean valinnan tekemiseen tai trialin ohimenemiseen).
Esityskerta (altistamiskerta)	Single presentation	**Yksi ärsykkeen esittäminen** alkaen ärsykkeen auditiivisesta esiintymisestä ja päättyen trialin päättymiseen (esim. oikean valinnan tekemiseen).
Harjoitteluaika	Training time	Kaikkiin lukemisen perustaitoja aktiivisesti harjoittaviin tehtäviin kulunut peliaika. Sisältää harjoittelu- ja lisäharjoittelu-kenttätyypit. Kentän valinnan ja palkkioiden kuten tarinoiden ja tarrakirjan parissa vietettyä aikaa ei lasketa harjoitteluajaksi.
Harjoittelukenttä	Training game	Kenttätyyppi, jossa aktiivisesti harjoitellaan pelin oppimateriaalia. Voi sisältää myös helpotetun harjoittelun. Ks. kuvio 2.
Helpotettu lisäharjoittelu[1]	Simplified extra game	Ärsykkeille altistumisia tarjoava lisäharjoittelukentän osa, jossa vaihtoehtojen määrä vähennetty ja/tai ärsykkeiden putoamisnopeutta vähennetty ja/tai kohdeärsyke esitetty korostettuna. (Lisäselvitys: jos lisäharjoittelukentän lopetuskriteeriksi on määritelty oikeiden vastausten määrä, niin lisäharjoittelun lopuksi esitetään kentän lopetuskriteerin saavuttamiseksi tarvittava trialimäärä helpotettua lisäharjoittelua.)
Helpotettu harjoittelu[1]	Simplified game	Ärsykkeille altistumisia tarjoava harjoittelukentän osa, jossa vaihtoehtojen määrä vähennetty ja/tai ärsykkeiden putoamisnopeutta vähennetty ja/tai kohdeärsyke esitetty korostettuna. (Lisäselvitys: jos harjoittelukentän lopetuskriteeriksi on määritelty oikeiden vastausten määrä, niin harjoittelukentän lopuksi esitetään kentän lopetuskriteerin saavuttamiseksi tarvittava trialimäärä helpotettua peruspeliä.)
Häiriöärsyke (distraktori)	Distracter (distractor)	Pelissä esiintyvä visuaalinen ärsyke (esim. kirjain), joka **ei vastaa** auditiivisesti esitettyä ärsykettä (esim. äännettä). Vrt. kohdeärsyke.
Implementaatio	Implementation	Asiantuntijan määrittämän oppimateriaalin asettaminen peliympäristöön.
Itemi	Item	Sisältöyksikkö. (Lisäselvitys: esim. ärsykkeen A eri esiintymiskerrat muodostavat itemin A).

LUKIMAT

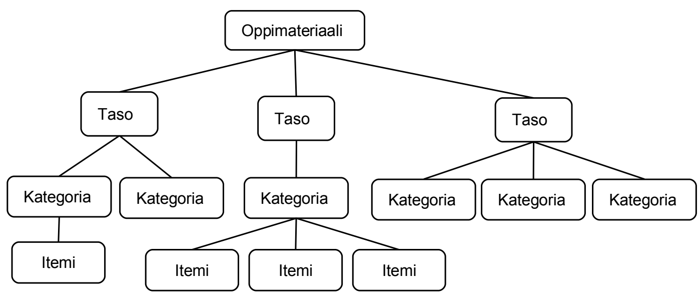
Kategoria	Category	Sisällöllisesti rajattu itemikokonaisuus, jonka sisältämät, ennalta päätettyjen kriteerien perusteella määritellyt itemit ovat keskenään lähes samankaltaisia (esim. yhtä vaikeita). Eri kategoriat ovat ennalta päätettyjen kriteerien perusteella keskenään erilaisia (esim. vaikeustasoltaan). Kategoria voi muodostua yhdestä tai useammasta itemistä (joka toimii kohdeärsykkeenä) sekä ennaltamääritellyistä tai ennaltamäärittelemättömistä itemeistä (jotka toimivat häiriöärsykkeinä). Ks. taso.
Kenttä	Level	Trialeista muodostuva pelaamista jaksottava kokonaisuus, joka voi olla itemien osalta ennaltamääritelty tai adaptaation mukaan määrittyvä. Kentän päättyminen riippuu ennalta määritellyistä peli- tai kenttäkohtaisista lopetuskriteereistä (esim. altistumisaika, itemien määrä, jne.)
Kenttätyyppi	Level type	Kentän tavoitetta tai tarkoitusta kuvaava luokka. Kenttätyyppejä ovat esimerkiksi pelaajan osaamista mittaava arviointikenttä ja palaajalle tukiharjoittelua tarjoava lisäharjoittelukenttä. (Lisäselvennys: kenttätyyppi osoittaa minkä tavoitteen vuoksi tai miksi kenttää pelataan). Vrt. tehtävätyyppi.
Kohdeärsyke	Target stimulus	Pelissä esiintyvä visuaalinen ärsyke, joka vastaa auditiivisesti esitettyä ärsykettä. (Lisäselvitys: ns. oikea vaihtoehto).
Kokonaisaltistumisaika	Total exposure time	Yksittäisten altistumisaikojen yhteenlaskettu summa.
Korostettu triali	Highlighted trial	Triali, jossa kohdeärsyke esitetään visuaalisesti tehostettuna (eli korostettuna). Pelaajan tekemät valinnat määritellään erilaisiksi kuin tavanomaisissa (ei-korostetuissa) trialeissa tehdyt valinnat.
Lisäharjoittelukenttä[1]	Extra game	Ärsykkeille altistumisia tarjoava kenttätyyppi, jossa keskitytään harjoittelemaan harjoittelukentässä huonosti osattuja ärsykkeitä. Myös hyvin osattuja ärsykkeitä on mukana motivoinnin takia. Voi sisältää myös helpotetun lisäharjoittelun.
Ohitriali	Missed trial	Triali, jossa ei ole tehty aktiivista valintaa. Myös taukovalinta tai trialin meno tauolle määritetään analyyseissa ohitrialiksi riippumatta pelaajan vastauksen oikeellisuudesta.
Oppimateriaali	Learning content	Auditiivisten ja visuaalisten itemien sisältökokonaisuus, joka jakautuu tasoihin ja edelleen kategorioihin.
Palaute	Feedback	Valinnan tekoa lähes välittömästi seuraava visuaalinen ja/tai auditiivinen tapahtuma (tai tehoste), joka ilmaisee valinnan oikeellisuuden.
Palkkio	Reward	Ennalta määritellyn pelitilanteen saavuttamisesta pelaajalle annettava visuaalinen ja/tai auditiivinen tapahtuma (tai tehoste) ja/tai harjoittelusta poikkeava toiminto (Lisäselvitys: esimerkiksi virtuaalisen tarran valinta ja asettaminen tarrakirjaan on palkkio).
Pelaajadata	Player data	Rekisteriselosteessa mainitut pelaajakohtaiset profiilitiedot, yleisesti ja yksilöllisesti määritellyt peliasetukset ja pelaamisesta kertyneet tiedot (=lokitiedot, peliloki).
Peliaika	Playing time	Kaikki pelaamiseen ja palkkioihin yms. käytetty aika pelaajan peliin kirjautumisesta pelistä poistumiseen/pelaajan vaihtumiseen tauot pois luettuna.
Pelikerta (sessio)	Playsession	Yhtäjaksoisesti vähintään 1 minuutin kestävä altistuminen harjoiteltavalle sisällölle. Uusi pelikerta lasketaan alkaneeksi, kun edellisestä pelikerrasta on kulunut vähintään 15 minuuttia.
Peliloki (lokitiedot)	Gamelog	Pelaamisesta automaattisesti palvelimelle kertyneet pelitiedot. Ks. pelaajadata.

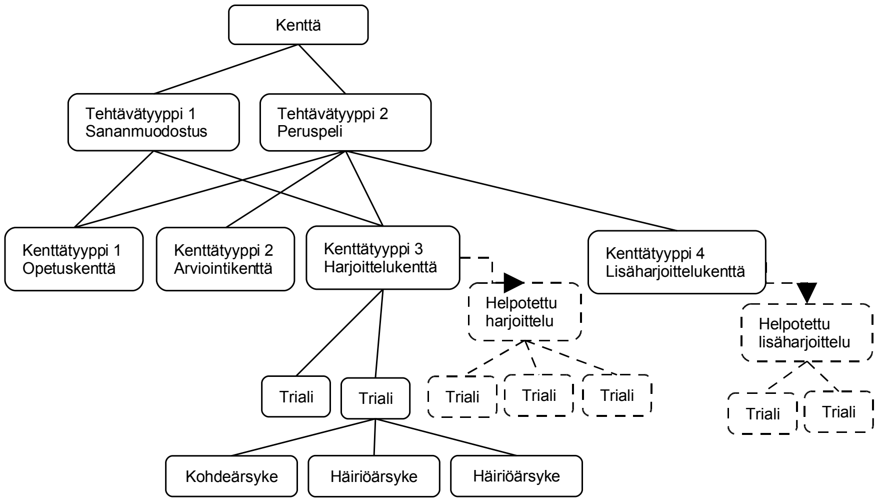
LUKIMAT

Peruspeli	Basic game	Pelin tehtävätyyppi, jossa harjoitellaan visuaalisten ja auditiivisten ärsykkeiden yhdistämistä. Pelaajan tehtävänä on valita kohdeärsyke häiriöärsykkeiden joukosta. Valinnasta annetaan palaute.
Sananmuodostustehtävä[1]	Wordforming task	Pelin tehtävätyyppi, jossa pelaaja muodostaa annetuista visuaalisista ärsykkeistä suurempia kokonaisuuksia (esim. kirjaimista tavuja tai tavuista sanoja) auditiivisesti annetun vihjeen mukaisesti.
Taso	Stream	Yhdestä tai useammasta kategoriasta ennalta määrätyin kriteerein muodostuva kokonaisuus (esim. kirjaintaso, tavutaso, sanataso, epäsanataso). Tasoja voidaan käyttää kategorioiden ryhmittelyyn.
Tauko	Pause	Pelitapahtumien pysähtyminen pelaajan passiivisuuden tai pelaajan taukovalinnan vuoksi. Tauolle menneet trialit käsitellään analyyseissa ohitrialeina.
Tehtävätyyppi	Task type	Pelin eri tehtävätyypeissä tehtävien esittämistapa ja pelaajalta vaadittavat toiminnot ovat erilaisia. Tehtävätyyppejä ovat esimerkiksi sananmuodostustehtävä ja peruspeli. (Lisäselvennys: tehtävätyyppi osoittaa millä tavalla pelataan). Vrt. kenttätyyppi.
Testikenttä	Test level	Peruspeli –tehtävätyypin muotoinen arviointikenttä (mm. Graphogamessa ja Klassisessa Ekapelissä).
Triali	Trial	Yksi pelitapahtuma, jossa pelaajalla on mahdollisuus valita näytöltä vaihtoehtojen joukosta auditiivisesti esitetyn kohdeärsykkeen visuaalinen vastine. Ks. ohitriali. Ks. vastattu triali.
Vastattu triali	Answered trial	Triali, jossa on tehty aktiivinen valinta. (Myös korostetuissa trialeissa tehdyt valinnat ovat vastattuja trialeita). Ks. ohitriali.
Vastausaika (~reaktioaika)	Response time	Ärsykkeen auditiivisesta esiintymisestä **ensimmäisen** aktiivisen valinnan tekoon (hiiren klikkaukseen) kulunut aika. Riippumaton valinnan oikeellisuudesta. Vrt. altistumisaika. Vrt. esittämisaika.
Ärsyke	Stimulus	Pelissä esiintyvä visuaalisesti tai auditiivisesti esitettävä kohde, joka esiintyy kohdeärsykkeenä tai häiriöärsykkeenä.

[1]Eivät mukana vanhemmissa peliversioissa

Kuvio 1. Ekapelin sisältöjen ja pelitapahtumien rakenne





Kuvio 2. Ekapelin pelikenttien rakenne