Tampereen yliopisto

Dani Bärlund

TILASTOLLISEN LUONNOLLISEN KIELEN KÄSITTELYN MALLIEN IBM-MALLI 1 JA IBM-MALLI 2 LAADUN JA KOULUTTAMISEN VERTAILU

Informaatioteknologian ja viestinnän tiedekunta
Tilastollisen data-analyysin kandidaattitutkielma
Toukokuu 2023

Tiivistelmä

Dani Bärlund: Tilastollisen luonnollisen kielen käsittelyn mallien IBM-malli 1 ja IBM-malli 2 laadun ja kouluttamisen vertailu
Kandidaattitutkielma
Tampereen yliopisto
Matematiikan ja tilastotieteen tutkinto-ohjelma
Toukokuu 2023

Tilastolliset käännösmallit ovat osa tilastollisen luonnollisen kielen käsittelyä, millä voidaan kääntää sanoja tai lauseita lähtökielestä kohdekieleen. Tutkielman aiheena on ensimmäisten tilastollisten käännösmallien, eli IBM-mallien 1 ja 2 soveltaminen Euroopan parlamentin saksa-englanti-korpukseen vuodelta 1996-2011. Tutkimuskysymyksenä on mallien väliset erot käännösten laadussa sekä kouluttamisen vaativuudessa. Vastauksia kysymyksiin saadaan vertaamalla mallien käännösten todennäköisyyksiä sekä perplexity-testauksen arvoja.

Tutkielmassa ensin esitellään Euroopan parlamentin korpus ja tähän vaadittavat käsittelyaskeleet. Tämän jälkeen teorialuvussa esitellään IBM-mallien 1 ja 2 matemaattiset esitysmuodot ja ominaisuuksia. Samassa luvussa esitellään myös EM-algoritmin määritelmä ja toiminta IBM-mallien kouluttamisessa, joita käytetään malleja luodessa. IBM-mallien ja EM-algoritmin määrittelyyn on käytetty apuna Philipp Koehnin teosta *Statistical machine translation*. Tutkielman lopuksi esitellään ja analysoidaan tulokset, sekä tehdään johtopäätökset tuloksien merkittävyydestä tutkimuskysymykseen.

Tutkielman tärkein tulos on IBM-mallin 2 huomattavasti parempi kyky kääntää kokonaisia lauseita suhteessa IBM-malliin 1. Mallien käyttäytyminen käännösten todennäköisyyksissä vastasi odotuksia ja tulokset tuottivat selviä eroja mallien välille. Perplexityn vertailu vahvisti muita havaintoja, mutta ei tuottanut tutkimuskysymyksen kannalta uusia tuloksia. Johtopäätös on se, että IBM-malli 2 on kaikin tavoin parempi kuin IBM-malli 1 ja kuuluisi yleensä tulla valituksi.

Avainsanat: EM-algoritmi, perplexity

Sisällys

1 Johdanto

Maailman globalisaation ja teknologian yleistymisen myötä ihmisten tarve kommunikoida sekä asioida eri kielillä on kasvanut. Tämä on nostanut tarpeen ja mahdollisuuden konekääntämiselle, jossa koulutetut mallit kääntävät sanoja sekä lauseita kieleltä toiselle. Esimerkkinä tällaisesta konekääntäjästä on Google kääntäjä. Konekääntäjät jakautuvat kahteen kategoriaan: tilastollinen luonnollisen kielen käsittely ja neuroverkostot. Nykyisin kääntäjät toimivat neuroverkostojen avulla, mitkä vaativat valtavia määriä dataa ja pitkää kouluttamista. Tämän vuoksi tutkielmassa perehdytään ensimmäisiin tilastollisiin käännösmalleihin IBM-malli 1 ja IBM-malli 2, mitkä ovat kouluttamisen ajan ja aineiston määrän suhteen kohtuullisia. Mallien kouluttamiseen käytetään Euroopan parlamentin saksa-englanti-korpus vuosilta 1996 – 2011, missä saksan- ja englanninkieliset lauseet ovat rinnakkaisilla riveillä.

Kyseiset mallit ovat valittu koska IBM-mallit toimivat perustana niin kolmelle muullekin IBM-mallille, kuin nykyisille neuroverkostoille. Mallien avulla voidaan tarkastella konekääntämisen perusteita teorian ja esimerkkien avulla sekä tutustua EM-algoritmin toimintaan. Myöskin mahdollistaa mallien välisen eron vertailun. Hypoteesina mallien erolle on, että IBM-malli 2 konvergoituu nopeammin sekä tarkemmin lokaaliin maksimiin, kuin IBM-malli 1. Tarkoittaen että IBM-mallin 2 käännösten tarkkuus on parempi, kuin IBM-mallin 1 pienemmällä kouluttamisella.

Tutkielman rakenne etenee seuraavasti: ensiksi Luvussa 2 esitellään aineiston alkuperä sekä rajaus ja aineiston käsittelyn askeleet. Aineiston ymmärrettävyyden vuoksi Liitteestä A löytyy esimerkki korpuksien muodosta ja välisestä rinnakkaisuudesta. Luvussa 3 esitellään IBM-mallien historiaa sekä kattavasti teoriaa. IBM-mallien teorian avulla esitellään EM-algoritmin toiminta mallien kouluttamisessa, jonka jälkeen tarkastellaan mallien testaamiseen tarkoitettua perplexityä. Teoria osuuden jälkeen esitellään tuloksia, missä aluksi tarkastellaan IBM-mallien 1 ja 2 yksittäisen sanan todennäköisyyden eroja kouluttamisen iteraatioihin sekä toisiinsa. Tämän avulla tarkastellaan kokonaisen lauseen käännöksiä ja mallien eroja. Lopuksi malleja vertaillaan keskenään perplexityn avulla. Tutkielman päätteeksi Luvussa 6 tehdään tuloksien pohjalta yhteenveto ja pohditaan tuloksien merkitsevyyttä.

2 Aineisto

2.1 Aineiston esittely

Työssä käytetään Euroopan parlamentin englanti-saksa-korpusta, joka on yli 2 miljoonan lauseen kokoelma parlamentin keskusteluista vuodesta 1996 vuoteen 2011 asti.

Aineisto muodostuu kahdesta tekstitiedostosta, jonka jokainen rivi on yksi lause. Tiedostojen rivit ovat toisiinsa suoranaisesti vertailtavissa tarkoittaen, että englannin kielen korpuksen rivi 5 on käännös saksankielisen korpuksen rivistä 5. Esimerkki käyttäytymisestä löytyy Liitteestä A. Tämä käännöksien välinen yhteys mahdollistaa käännösmallien kouluttamisen, selkeällä vertailulla riviltä riville.

Mallin kouluttaminen suuremmalla aineistolla tuottaa parempia tuloksia, mutta tämän työn tavoitteiden myötä aineistoa rajoitetaan pienempään 100 000 ensimmäiseen lauseeseen. Tämä mahdollistaa nopeamman mallin kouluttamisen ja täten useampien testien tuottamisen.

Aineistossa ilmenee pieniä ongelmia, kuten isojen ja pienten kirjainten aiheuttamat sanan eroavuudet, jolloin kouluttamisen aikana sanat jäävät laskennassa erilleen. Tämä sanojen eroavuus onkin suuri haaste mallin kouluttamisessa. Tutkielmaan valitun aineiston koko on myös yleistä pienempi vain 100 000, kun yleensä tahdottaisiin ainakin 500 000 lausetta. Tämä ei kuitenkaan tule vaikuttamaan paljoa tuloksiin, sillä aineiston laatu on tärkeämpi kuin määrä (Imam et al. 2011;Gavrila ja Vertan 2011). Lauseiden määrä saattaa kuitenkin johtaa pieneen käännösten puuttellisuudeen.

2.2 Aineiston käsittely

Käsittely jaetaan neljään askeleeseen Thanakin (2017) esittämän tekstin normalisoinnin mukaisesti. Askeleet suoritetaan yhtäaikaisesti niin saksan kuin englannin kielen korpukselle.

1. Virkkeet jaetaan yksittäisiin sanoihin.
2. Poistetaan ylimääräiset merkit, kuten väliviivat.
3. Sanojen isot kirjaimet muutetaan pieniksi kirjaimiksi.
4. Sanat muutetaan niiden perusmuotoon NLTK:n (*Natural language tool kit*) avulla.

Alla esitetään esimerkkinä, jokainen aineiston käsittelyn askel lauseelle “Today’s report is important“.

0. “Today’s report is important”
1. [“Today’s”, “report”, “is”, “important”]
2. [“Todays”, “report”, “is”, “important”]

3. [“todays”, “report”, “is”, “important”]

4. [“today”, “report”, “is”, “import”]

Käsittelyaskeleiden jälkeen aineisto on muuttunut ohjelmistolle helpommin käsiteltäväksi. Huomattavissa on kuitenkin, että neljännen askeleen tekemä sanan muutos perusmuotoon ei ole täydellinen vaan aikaisemman esimerkin muunnos “important” muotoon “import” on väärin. Tämä ei kuitenkaan ole työn kannalta merkittävä ongelma, joten se voidaan sivuuttaa.

Askeleiden jälkeen aineisto on muunnettu oikeaan muotoon ja on valmis käytettäväksi käännösmallien kouluttamiseen. Tulevassa luvussa esitellään kouluttamisen teoriaa ja mallien hyvyyden testaamista, minkä jälkeen teoriaa päästään hyödyntämään aineistoon ja tutkimaan tämän tuottavia tuloksia.

3 IBM-mallien teoriaa

Luvun teoria perustuu Koehnin (2010) lukuihin 4.1.5, 4.2.3, 4.2.3 ja 4.4.1. Teorian avulla esitetään mallien rakentamisen tilastotieteelliset perusteet sekä laskentaan käytetyt kaavat.

3.1 IBM-malli 1

IBM:n ensimmäinen 1980-luvulla kehittämä tilastollinen luonnollisenkielen kääntämisen malli on perusta niin useammille IBM-malleille, kuin nykyisille koneoppimisen kääntäjille. Malli perustuu kahteen osaan: sanastolliseen kääntämiseen (*engl. lexical translation*) ja kohdistamiseen (*engl. alignment*). Yhdistäminen mahdollistaa uusien lauseiden kääntämisen vieraalle kielelle. Seuraavaksi tutustutaan tarkemmin kohdistamiseen, joka toimii suuressa osassa mallin kouluttamisessa.

Kohdistamisella tarkoitetaan kahden erikielisen lauseen sanojen kohdistamista toisiinsa. Aina kuitenkaan jokaisella sanalla ei ole toisessa kielessä vastinetta, minkä takia kieleen johon kohdistetaan, lisätään NULL-sana, jolloin mahdollistuu kohdistamisfunktion täydellinen määrittely. Kohdistamisfunktio voidaan kirjoittaa matemaattiseen muotoon

$$a : j \rightarrow i.$$

Määritelmässä a on kohdistamisfunktio, i on indeksi sanalle käännettävästä lauseesta ja j on indeksi sanalle, johon i kääntyy. Esimerkki kohdistumisesta löytyy Liittestä B.

Sanastollinen kääntäminen perustuu tapahtuneiden tapahtumien määrään. Vertailemalla, kuinka usein kaksi sanaa kohdistuvat toisiinsa suhteutettuna sanan määrään,

$$t(e_j|f_{a_j}) = \frac{C(e_j, f_{a_j})}{\sum_{a \in A} C(e_j, f_{a_j})},$$

missä $C(e_f, f_{a_j})$ on count-funktio. Jolloin muodostuu tapahtumien määrästä koostuva taulukko, kuten esimerkissä Taulukko 3.1. Ja jokaiselle sanalle muodostuu käännöspöytä (*engl. translation table*), joka sisältää mahdollisuudet, että sana f kääntyy sanaan e, kuten Taulukossa 3.2.

Tapahtumapöytä		
Umwelt	ist	wichtig
environment : 85	is : 95	important : 73
circle : 15	yes : 5	main : 27

Taulukko 3.1. Yksittäisen sanan eri käännöksien tapahtumien määrää saksankielisestä sanasta englantiin

Todennäköisyyspöytä		
Umwelt	ist	wichtig
environment : 0.85	is : 0.95	important : 0.73
circle : 0.15	yes : 0.05	main : 0.27

Taulukko 3.2. Taulukko kuvaa yksittäisen sanan eri käännöksien todennäköisyyksiä saksankielisestä sanasta englantiin

Huomattavaa on myös, että Taulukon 3.2 todennäköisyydet summautuvat yhteen

$$\sum_{e \in A} t(e|f) = 1, \text{ missä } A = \{e_1, e_2, ..., e_n\}.$$

Kohdistamisen ja sanastollisen kääntämisen yhdistyessä muodostuu IBM-malli 1. Aluksi malli koulutetaan kohdistamisen avulla käyttäen EM-algoritmia, johon tutustutaan tarkemmin luvussa 3.3. Kouluttamisen valmistuessa sanojen käännöspöydät ovat saatavilla. Tällöin lauseita on mahdollista kääntää. Seuraavaksi suoritetaan käännös lauseesta $\mathbf{f}$ lauseeseen $\mathbf{e}$. Oletetaan, että $\mathbf{f} = \{f_1, f_2, ..., f_{l_f}\}$ on käännettävän kielen lauseen sanat ja $\mathbf{e} = \{e_1, e_2, ..., e_{l_e}\}$ on tunnetun kielen sanat, missä l_f ja l_e ovat lauseiden pituudet. Tällöin käännöksen todennäköisyys lasketaan kaavalla,

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}), \tag{3.1}$$

missä $\frac{1}{(l_f+1)^{l_e}}$ on kaikkien mahdollisten kohdistuksien määrä. Normalisointivakio ϵ on luku, jonka avulla $p(\mathbf{e}, a|\mathbf{f})$ on kelvollinen jakauma, kun

$$\sum_{\mathbf{e},a} p(\mathbf{e}, a|\mathbf{f}) = 1.$$

Esimerkki Kaavan 3.1 käytöstä, kun käännetään saksankielistä lausetta englanninkieliseen lauseeseen:

$$p(\text{environment is important}, a|\text{Umwelt ist wichtig})$$

$$= \frac{1}{(3+1)^3} \cdot t(\text{environment}|\text{umwelt}) \cdot t(\text{is}|\text{ist}) \cdot t(\text{important}|\text{wichtig})$$

$$= \frac{1}{64} \cdot 0.85 \cdot 0.95 \cdot 0.73 \approx 0.009$$

Käännöksen todennäköisyyden laskenta on helppoa, mikäli todennäköisyydet sanojen käännöksille on saatavilla. Näin ei kuitenkaan ole, eikä myöskään sanojen

kohdistuminen ole tiedossa. Tällöin kyseessä on puuttuvan aineiston ongelma, jota pyritään ratkaisemaan myöhemmän luvun aiheen EM-algoritmin avulla. Tätä varten tarvitaan count-funktiot, joiden avulla tapahtumien määrää voidaan laskea aineistosta. Aluksi muodostetaan todennäköisyys $p(a|\mathbf{e},\mathbf{f})$ ketjusäännön avulla, jolloin

$$p(a|\mathbf{e},\mathbf{f}) = \frac{p(\mathbf{e},a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})},$$

missä

$$p(\mathbf{e}|\mathbf{f}) = \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i), \tag{3.2}$$

sijoituksien jälkeen kaava saadaan muotoon:

$$p(a|\mathbf{e},\mathbf{f}) = \prod_{j=1}^{l_e} \frac{t(e_j|f_a(j))}{\sum_{i=0}^{l_f} t(e_j|f_i)}. \tag{3.3}$$

Tämän avulla pystytään muodostamaan count-funktio

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_a p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_a(j)),$$

missä $\delta(x,y)$ on Kronecker delta, eli

$$\delta(x,y) = \begin{cases} 0 & \text{jos } x \neq y \\ 1 & \text{jos } x = y \end{cases}.$$

Sijoitetaan $p(a|\mathbf{e},\mathbf{f})$, jolloin

$$c(e|f;\mathbf{e},\mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \cdot \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_j). \tag{3.4}$$

Funktio laskee kuinka usein vieraankielinen sana f_{k_f} kohdistuu englannin kielen sanaan e_{k_e}, missä $k_f \in \{0,1,...,l_f\}$ ja $k_e \in \{1,...,l_e\}$. Count-funktioiden avulla pystytään luomaan uusi käännösjakauma (*engl. translation probability distribution*) kaavalla

$$t(e|f;\mathbf{e},\mathbf{f}) = \frac{\sum_{(e,f)} c(e|f;\mathbf{e},\mathbf{f})}{\sum_e \sum_{(e,f)} c(e|f;\mathbf{e},\mathbf{f})}. \tag{3.5}$$

Alaluku on antanut pohjan IBM-mallin 1 käännösten todennäköisyyksien laskennalle $p(\mathbf{e},a|\mathbf{f})$ avulla sekä luonut count-funktion, joka mahdollistaa uusien käännösjakaumien muodostamisen.

3.2 IBM-malli 2

Alaluku käsittelee IBM-mallin 2 teoriaa laskennan sekä kouluttamisen kannalta. Luvun teoria perustuu Koehn 2010 lukuun 4.41

IBM-malli 2 on kaksiosainen, jonka perusta on mallissa 1. Malli 1 ei kuitenkaan ota huomioon eri kohdistamisien mahdollisuuksia, eli englannin kielen sanan i kohdistamisen todennäköisyys vieraan kielen sanalle j on sama, kuin sanalle $j+2$, missä i on positio englannin kielen lauseessa ja j on positio vieraan kielen lauseessa. Myöskin $i \leq l_e$ ja $j+2 \leq l_f$, kun l_e on englannin kielen lauseen pituus sekä l_f vieraan kielen lauseen pituus.

Ensimmäinen osa on mallin 1 tyyliin käännöstodennäköisyyksien jakauma $t(e|f)$. Lisänä malliin 1 verrattuna IBM-malli 2 käyttää kohdistamista. Jolloin jokainen vieraan kielen sana i kohdistetaan englannin kielen sanaan j. Tällöin kohdistumisjakauma on muotoa

$$a(i|j, l_e, l_f),$$

halutessaan kohdistamisjakauma (*engl. alignment probability distribution*) voidaan myös asettaa käänteiseen järjestykseen. Huomattavaa on, että a kuvaa jokaisen englannin kielisen tulosteen j vieraankieliseen syötteeseen $a(j)$. Tämän avulla kohdistamisjakauma, voidaan kirjoittaa muodossa

$$a(a(j)|j, l_e, l_f).$$

Molemmat IBM-mallin 2 osat voidaan yhdistää matemaattiseen muotoon

$$p(\mathbf{e}, a|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \cdot a(a(j)|j, l_e, l_f). \tag{3.6}$$

Kohdistamisen lisääminen laskentaa ei huomattavasti suurenna laskennan vaikeutta, eikä myöskään lisää mallin kouluttamisen monimutkaisuutta. Mallin 1 tapaan, mallin 2 kouluttaminen pystytään muuttamaan eksponentiaalisesta polynomiseksi ongelmaksi, derivoimalla todennäköisyyttä $p(\mathbf{e}|\mathbf{f})$. Todennäköisyys saadaan alkuperäisestä muodosta muotoon

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(\mathbf{e}, a|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_{a(j)}) \cdot a(a(j)|j, l_e, l_f). \tag{3.7}$$

Aikaisemman kaavan avulla pystytään laskemaan todennäköisyys $p(a|\mathbf{e}, \mathbf{f})$, joka mahdollistaa sanastollisen kääntämisen sekä kohdistamisen laskennan aineistosta. Sanastollisen kääntäminen kaavalla

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_{j=1}^{l_e} \sum_{i=0}^{l_f} \frac{t(e|f) \cdot a(a(j)|j, l_e, l_f) \cdot \delta(e, e_j) \cdot \delta(f, f_i)}{\sum_{i'=0}^{l_f} t(e|f_{i'}) \cdot a(i'|j, l_e, l_f)}. \tag{3.8}$$

Sekä kohdistaminen kaavalla

$$c(i|j, l_e, l_f; \mathbf{e}, \mathbf{f}) = \frac{t(e_j|f_i) \cdot a(a(j)|j, l_e, l_f)}{\sum_{i'=0}^{l_f} t(e_j|f_{i'}) \cdot a(i'|j, l_e, l_f)}. \tag{3.9}$$

Kaavojen avulla pystytään muodostamaan uudet arvot estimaateille. Käännösjakaumalle $t(e|f; \mathbf{e}, \mathbf{f})$ kaavalla (3.5) ja kohdentamisjakauman

$$a(i|j, l_e, l_f) = \frac{c(i|j, l_e, l_f)}{c(i, l_e, l_f)}. \tag{3.10}$$

Alaluku on antanut matemaattisen perustan IBM-mallin 2 toiminnalle sekä kouluttamiselle. Näiden oppien avulla käsitellään seuraavan alaluvun aiheitta EM-algoritmia.

3.3 EM-algoritmi

Korpuksissa on harvoin sana kohtaista kohdistamista a, jolloin aineisto on laskentaan puutteellinen. Ongelma pyritään ratkaisemaan EM-algoritmin (*engl. expectation-maximization algorithm*) avulla parametrejä estimoimalla. Algoritmi on kaksi osainen, mikä jakautuu E- (*engl. expectation*) ja M- (*engl. maximization*) askeleeseen. Askeleet tarkoittavat odotusarvon laskemista ja uskottavuusyhtälön maksimoimista. IBM-mallin 1 tapauksessa estimoitavaksi riittää ainoastaan käännöksien todennäköisyys $t(f|e)$ estimointi. Vaativammalle IBM-mallille 2, tarvitaan lisäksi kohdentamistodennäköisyyttä $a(j|i, l, m)$.

EM-algoritmi on iteratiivinen prosessi, jossa aikaisemman iteraation estimaattien t_n ja a_n avulla luodaan seuraavat estimaatit t_{n+1} ja a_{n+1}. Askelia suoritetaan, kunnes jokin ennalta määrätty ehto, kuten

$$L(t_{n+1}, a_{n+1}) - L(t_n, a_n) < \rho,$$

toteutuu. ρ:n ollessa riittävän pieni, estimaatit ovat konvergoituneet (*converge*) lokaaliin maksimiin.

Aluksi valitaan estimaattien aloituspiste t_0 ja a_0, johon Koehn 2010 ehdottaa IBM-mallin 1 kohdalla tasajakaumaa (*engl. uniform distribution*). IBM-mallin 2 aloituspisteiksi Brown et al. 1993 ehdottavat muutaman iteraation suorittanut IBM-mallin 1 estimoituja jakaumia.

3.3.1 E-askel

IBM-mallille 1: jokaisen lauseen **f** ja **e**, kohdalla lasketaan tapahtumien määrä count-funktion (3.4) avulla. Funktio antaa nollasta eroavan arvon ainoastaan, mikäli vieraankielen sana f ja englannin kielen sana e, löytyvät lauseista **f** ja **e**.

IBM-mallille 2: Vastaavasti IBM-mallin 1 tapaisesti lasketaan count-funktion (3.8) arvot sanastolliselle kääntämiselle. Lisänä lasketaan count-funktion $c(i|j, l_e, l_f; \mathbf{e}, \mathbf{f})$ (3.9) avulla kohdentamisen arvot.

3.3.2 M-askel

E-askeleen arvoja hyödynnetään uusien estimaattien laskemiseen.

IBM-mallille 1: Estimoidaan E-askelessa saadun count-funktion arvoja uuden käännösjakauman $t(e|f)$ (3.5)

IBM-mallille 2: Vastaavasti estimoidaan käännösjakauma $t(e|f)$ (3.5) sekä estimoidaan kohdentamisjakauma $a(i|j, l_e, l_f)$ (3.10) E-askeleen arvojen avulla.

EM-algoritmi kokonaisuudessaan

Algoritmi kiteytyy neljään kohtaan alla olevan taulukon mukaisesti:

1. Valitse alkuarvot t_0 ja a_0 mallille sopivalla tavalla
2. Laske count-funktioiden arvot t_n ja a_n estimaatteja käyttäen
3. Laske uusien estimaattien t_{n+1} ja a_{n+1} arvot count-funktioita käyttäen
4. Suorita kohtia 2 ja 3, kunnes arvot lähestyvät lokaaliin maksimiin

Neljännen askeleen jälkeen estimoidut käännös- ja kohdentamisjakaumat ovat saavuttaneet parhaimman arvonsa, jolloin IBM-mallin kouluttaminen on valmis.

EM-algoritmi on ratkaisu haastavalle puuttuvan tiedon ongelmalle. Tämä estimoi vaaditut jakaumat siten, että sanojen ja lauseiden kääntäminen onnistuu klassisen todennäköisyyden keinoin. On kuitenkin haastavaa tietää milloin estimaatit ovat riittävän hyviä. Kyseistä ongelmaa tarkistellaan tarkemmin seuraavassa alaluvassa.

3.4 Käännöksen laadun testaaminen Perplexityllä

Alaluvun teoria perustuu Koehnin (2010) lukuun 4.2.4. Oletetaan, että on malli $p(\mathbf{e}|\mathbf{f})$, missä vieraan kielen sanat f kääntyvät englannin kielen sanaan e. Valitaan ennalta valitsemasta testiaineistosta vieraan kielen lause f ja tämän käännös e. Käännöksen todennäköisyys lasketaan kaavalla (3.2) IBM-mallin 1 tapauksessa ja kaavalla (3.7) IBM-mallin 2 tapauksessa. Todennäköisyyksien arvot kasvavat, kun käännöksien varmuus kasvaa, tällöin $p(\mathbf{e}|\mathbf{f})$ todennäköisyydet saavat arvoja. IBM-mallin 1 todennäköisyys kasvaa kohti

$$p(\mathbf{e}|\mathbf{f})_{Model1} \rightarrow \frac{\epsilon}{(l_f+1)^{l_e}},$$

kun $t(e|f)$ lähestyvät arvoja 0 ja 1. Vastaavasti IBM-malli 2 lähestyy arvoa

$$p(\mathbf{e}|\mathbf{f})_{Model2} \rightarrow \epsilon,$$

kun $t(e|f)$ ja $a(i|j,l_e,l_f)$ lähestyvät arvoja 0 ja 1. Nämä parhaimmat arvot saavutetaan, kun lokaalimaksi on EM-algoritmin avulla löydetty.

Mallin kasvua kohti arvoa suurinta todennäköisyyttä voidaan mitata perplexity:llä:

$$log_2 PP = -\sum_{\mathbf{s}} log_2 p(\mathbf{e_s}|\mathbf{f_s}), \tag{3.11}$$

missä s on sana testiaineiston lauseesta. Jolloin PP arvo saadaan kaavalla

$$PP = 2^{-\sum_{\mathbf{s}} log_2 p(\mathbf{e_s}|\mathbf{f_s})}$$

Tällöin todennäköisyyksien $p(\mathbf{e}|\mathbf{f})$ kasvaessa perplexity laskee. Vastaavasti, kun $p(\mathbf{e}|\mathbf{f})$ saavuttaa maksimin perplexity saavuttaa minimin.

Perplexity on nopea ja suoraviivainen tapa testata mallin laatua. Tämä ei kuitenkaan ole täydellinen työkalu laadun mittaamiseen, minkä vuoksi yleensä käytetään useaa metodia, kuten BLEU-arvoa.

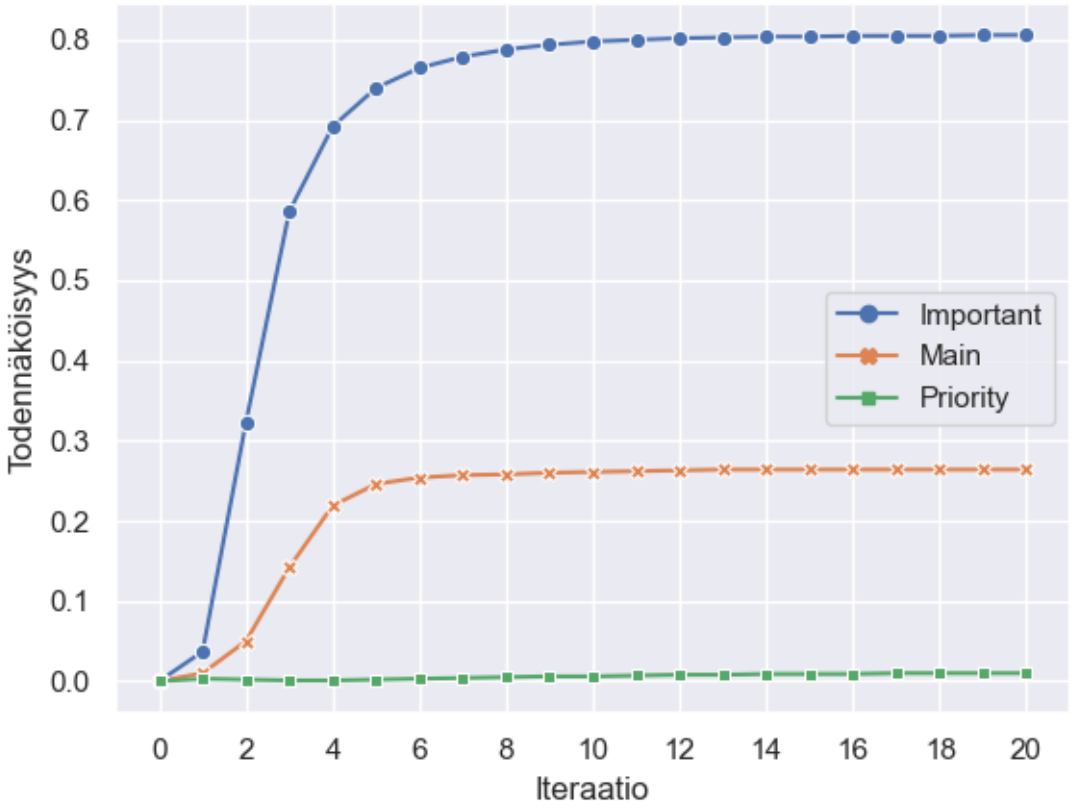
4 Tulokset

4.1 Iteraatioiden määrän vaikutus käännöksen todennäköisyyteen

4.1.1 Yksittäisen sanan käännöksen todennäköisyyden ero

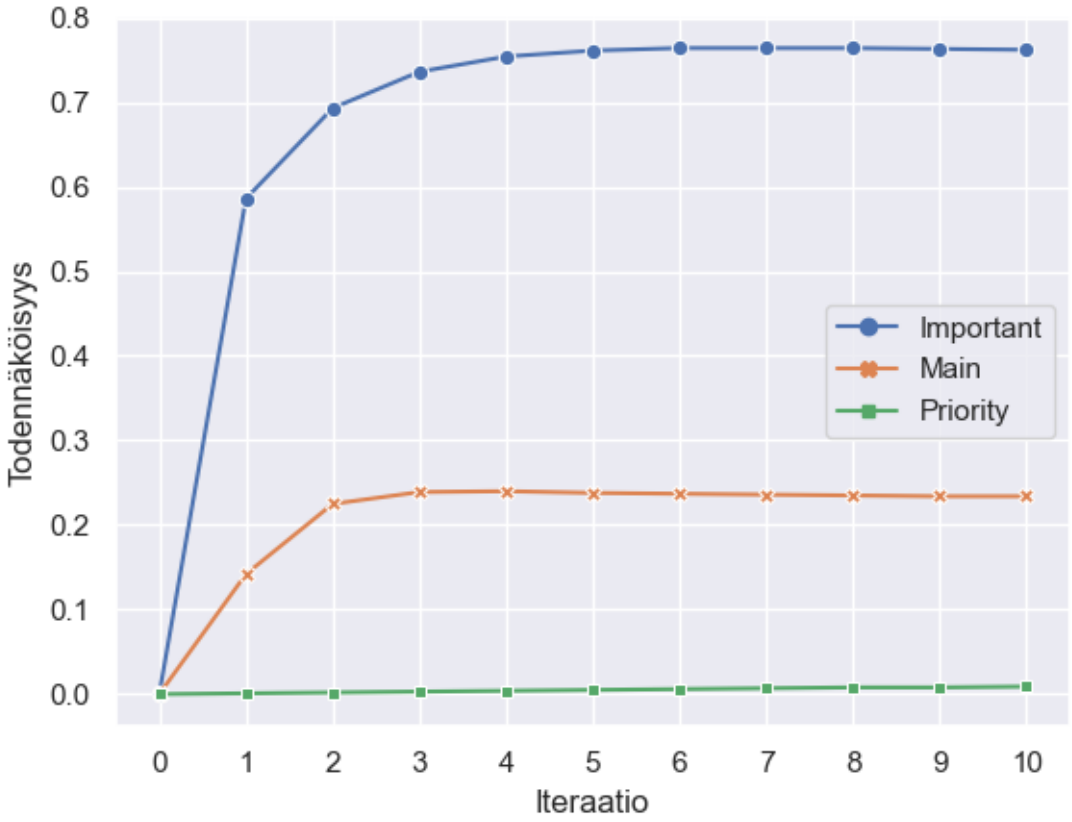
Luvussa tullaan esittämään kuvaajia, missä esiintyvät esimerkiksi nostetut sanat sekä näiden käännöksien todennäköisyydet. Kuvaajat eivät siis kuvaa koko aineistoa, vaan ainoastaan yhden tapauksen käyttäytymistä eri IBM-malleille ja iteraatioille.

EM-algoritmin konvergoitumisen ominaisuuden perusteella voidaan olettaa sanan käännöksen paranevan iteraatioiden kasvaessa. Tällöin paras käännös lähestyy normalisoimattomassa laskennassa kohti yhtä. Vastaavasti huonot käännökset lähestyvät nollaa. Mikäli sanalle on useampia oikeita käännöksiä, tällöin todennäköisyydet vaihtelevat nollan ja yhden välillä käännöksen yleisyydestä riippuen. Kaikki nämä kolme mahdollista tapausta nähdään Kuvaajassa 4.1, jossa on piirrettynä IBM-mallin 1 antamia todennäköisyyksiä sanan “Wichtig” (Tärkeä) käännöksille “Important” (Tärkeä), “Main” (Pääasiallinen) ja “Priority” (Prioriteetti), iteraatioiden kasvaessa nollasta kahteenkymmeneen.



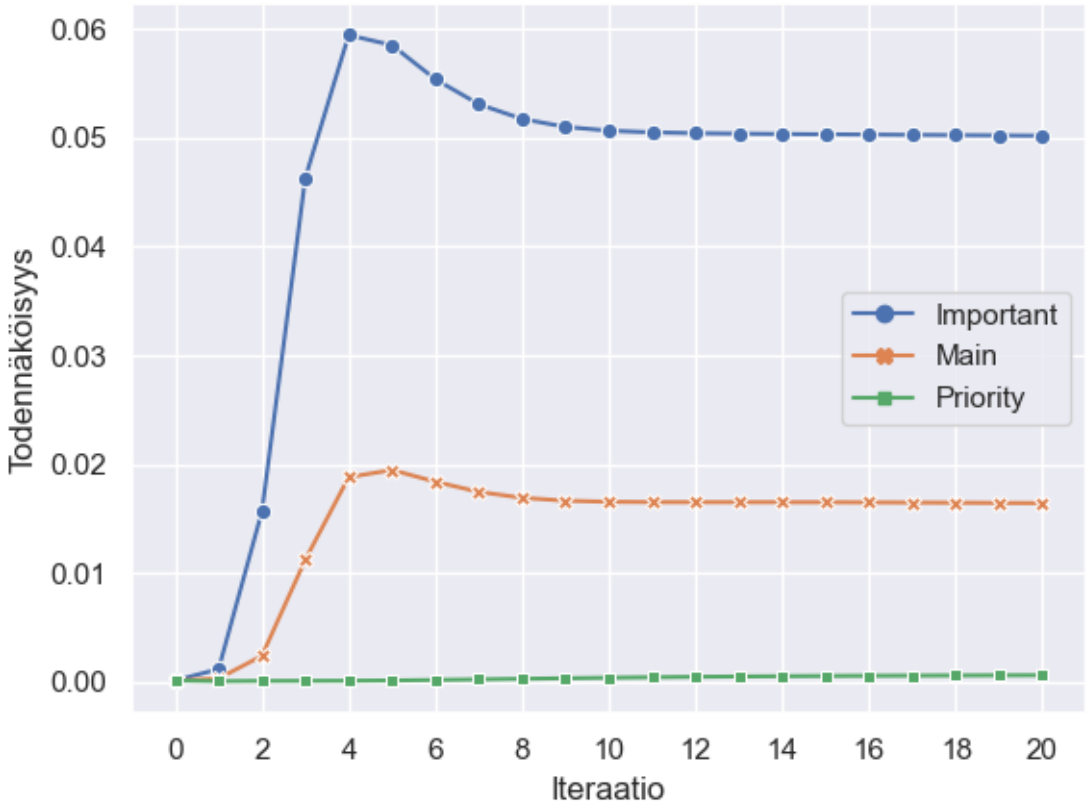
Kuva 4.1. IBM-mallin 1 iteraatioiden vaikutus Wichtig sanan käännöksien todennäköisyyksille

Mallia alustaessa kaikkien käännöksien todennäköisyydet asetetaan tasajakaumaksi, minkä takia käännöksien ensimmäinen piste on samassa kohdassa nollan tuntumassa. Heti ensimmäisten iteraatioiden aikana huomataan merkittävä ero käännöksien todennäköisyyksissä. Sana “Important” kasvaa nopealla vauhdilla kohti omaa maksimia, jonka saavuttaa noin kymmennenen iteraation kohdalla. Käännös on myös huomattavasti muita parempi. Vastaavasti “Main” kasvaa ensimmäisten iteraatioiden aikana, mutta tasaantuu iteraation kahdeksan kohdalla. Kaikista epätodennäköisimpänä käännöksenä on “Priority”, jonka todennäköisyys ei merkittävästä muutu iteraatioiden aikana. Tämä johtuu siitä, että mallin alustuksessa asettamat todennäköisyydet ovat hyvin lähellä nollaa, jota kohti kyseisen käännöksen todennäköisyys lähestyy. Jokaisen käännöksen todennäköisyydet huomioon ottaen voidaan päätellä, että IBM-mallin 1 konvergoitumisen kohta lokaaliin maksimiin kyseiselle 100 000 lauseen aineistolle on kymmenennen iteraation kohdalla.



Kuva 4.2. IBM-mallin 2 iteraatioiden vaikutus Wichtig sanan käännöksien todennäköisyyksille

Kuvaaja 4.2 on samanlainen Kuvaajan 4.1 kanssa, mutta käännöksien todennäköisyydet ovat laskettu IBM-mallilla 2. Myöskin iteraatioiden määrä on laskettu kahdestakymmenestä kymmeneen, mallin kouluttamisen vaativuuden takia. Odotuksena on tarkemmat käännökset sekä nopeampi konvergoitumisen suhteessa IBM-malliin 1, sillä IBM-malli 2 ottaa huomioon sanojen kohdistamisen. Käännöksien suuruudet ovat hyvin samankaltaiset Kuvaajan 4.1 kanssa, eli “Important” on selvästi parhain käännös, “Main” on toisiksi todennäköisin ja “Priority” huonoin. Erona Kuvaajan 4.1 kanssa on iteraatioiden määrä mallin konvergoitumiseen. Kuvaajan 4.2 perusteella IBM-mallin 2 konvergoitumiseen tarvittava iteraatioiden määrä on viisi.



Kuva 4.3. IBM-mallin 1 iteraatioiden vaikutus Wichtig sanan käännöksien normalisoiduille todennäköisyyksille

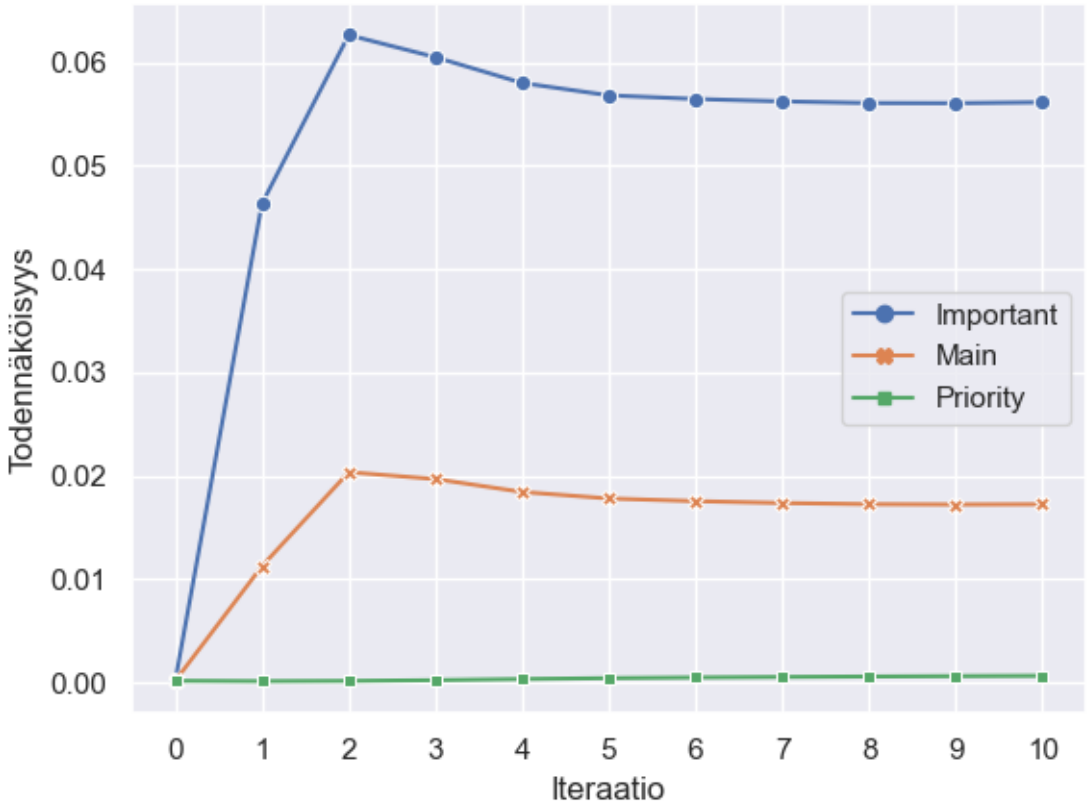
Kuvaajissa 4.1 ja 4.2 on käsitelty todennäköisyyksiä NLTK:n antamina yhtä lähestyminä todennäköisyyksinä. Jotta käännösjakaumat olisivat kunnollisia jakaumia, kuten teoria osuudessa esitetty. Tulee kaikkein todennäköisyyksien summautua yhteen. Tällöin hyödynnetään normalisaatiovakiota ϵ, jonka avulla todennäköisyydet voidaan normalisoida kaavalla

$$t(e|f)_\epsilon = t(e|f) * \epsilon$$

Kuvaaja 4.3 on hyvin samankaltainen Kuvaajan 4.1 kanssa. Molempien kuvaajien arvot on laskettu IBM-mallin 1 avulla ja näitä arvoja on kuvattu mallin iteraatioiden suhteen. Jälleen "Important" on selvästi muita parempi käännös ja konvergoituu kymmenennen iteraation kohdalla. Seuraavaksi todennäköisin on "Main" ja se konvergoituu tässä tapauksessa kymmenennen iteraation kohdalla toisin, kuin Kuvaajassa 4.1, jossa konvergoituminen tapahtui jo kahdeksannen iteraation kohdalla. Viimeisimpänä on "Priority", jonka todennäköisyyksien muutosta on kuvaajasta haastava huomata, todennäköisyyksien pienestä arvosta johtuen.

Kuvaajien 4.1 ja 4.3 välillä on paljon samankaltaisuuksia, mutta Kuvaajassa 4.3 on huomattava ero. Tämä tapahtuu neljännen iteraation kohdalla, kun todennäköisyyksien arvo lähtee laskemaan, toisin kuin Kuvaajassa 4.1. Todennäköisyyden lasku

viittaa siihen, että normalisointivakion arvo laskee nopeammin, kuin käännöksen todennäköisyys. Tämä tapahtuu silloin kun harvempien käännösten todennäköisyydet kasvavat, jolloin ϵ laskee. Kyseisen tapahtuman takia vastaavaa todennäköisyyden arvon laskua ei voida nähdä käännöksen “Priority”:n kanssa vaan voidaan havaita pientä kasvua neljännen iteraation jälkeen. Tapahtuma saattaa vaikuttaa aluksi huonolta käännösten todennäköisyyksien suhteen, mutta oikeasti todennäköisyydet jälleen konvergoituvat lokaaliin maksimiin noin kymmenen iteraation jälkeen ja ovat täten vertailu kelpoisia, muiden standardoitujen kuvaajien kanssa.



Kuva 4.4. IBM-mallin 2 iteraatioiden vaikutus Wichtig sanan käännöksien normalisoiduille todennäköisyyksille

Kuvaaja 4.4 on kuvaaja 4.2 standardoiduilla käännösten todennäköisyyksillä, joten toimii kuten aiemminkin standardoidun todennäköisyyden kohdalla. Tällöin käännösten todennäköisyydet ovat suhteessa yhtä laadukkaat Kuvaajan 4.2 kanssa, mutta huomattavasti pienemmät. Myöskin on huomattavissa todennäköisyyden lasku iteraation kaksi kohdalla, kuten Kuvaajan 4.3 neljännen iteraation kohdalla. Tarvittavien iteraatioiden määrä mallin konvergoitumiseen on myös sama, kuin standardoimattomassa tilanteessa.

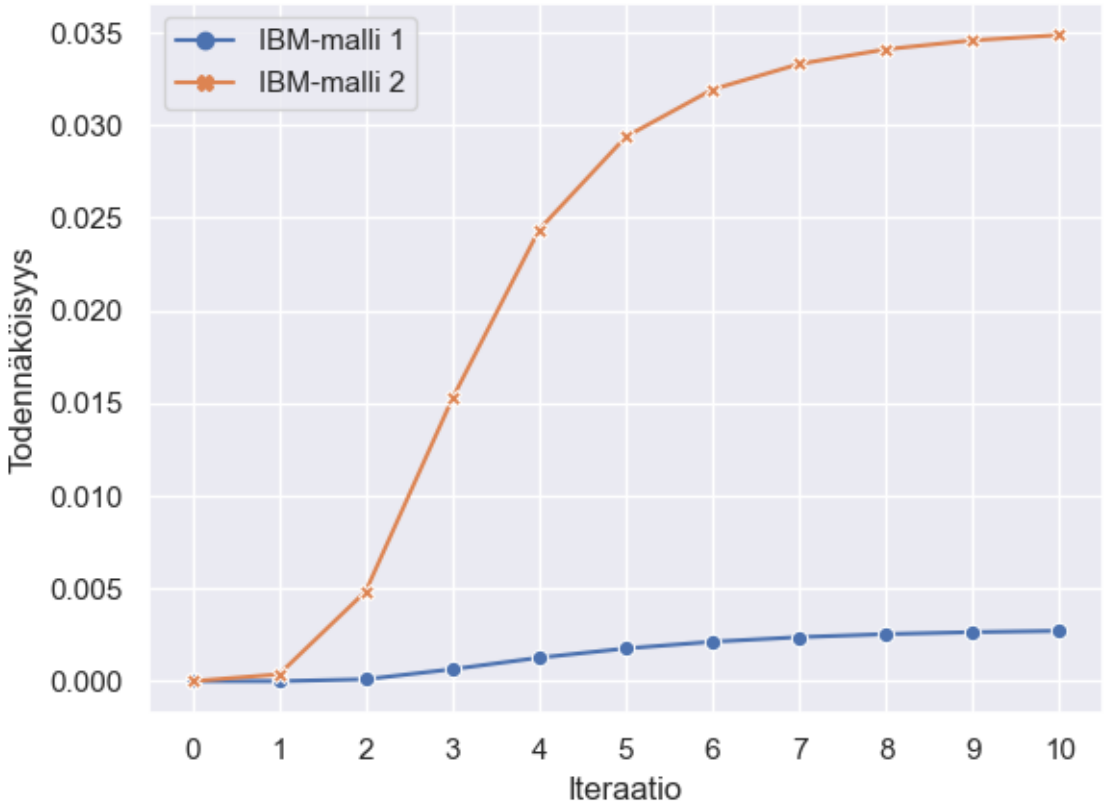
Kaikkien Kuvaajien 4.1, 4.2, 4.3 ja 4.4 avulla voidaan päätellä, että EM-algoritmin ominaisuuden avulla todennäköisyydet lähestyvät lokaalia maksimia iteraatioiden kasvaessa. Vaikka molemmat IBM-mallit 1 ja 2 lähestyvät omaa maksimia, tekee

IBM-malli 2 tämän huomattavasti nopeammin. Kuten Kuvaajien 4.1 ja 4.2 vertailussa iteraatiot ovat IBM-mallin 1 kohdalla kymmenen ja IBM-mallilla kuudennessa. Tämä toistuu myös standardoidussa tilanteessa Kuvaajissa 4.3 ja 4.4 missä vaadittavien iteraatioiden määrä ovat samat kuin aikaisemminkin. Tällöin siis IBM-malli 2 on huomattavasti parempi yksittäisen sanan käännökseen, sillä laskennan teho ei ole huomattavasti vaativampi kuin mallilla 1, mutta vaatii vain puolet vähemmän iteraatioita parhaimman tuloksen saavuttaakseen.

4.1.2 Lauseiden käännöksen todennäköisyyden ero

Aikaisemmassa osiossa esiteltyjen tulosten perusteella IBM-mallien 1 ja 2 yksittäisen sanan käännöksen todennäköisyydet ovat melkein samat. Huomattavasti suurempia eroja odotetaan lauseen kääntämisessä, sillä tämän laskennassa on mallien välillä suurta eroa. IBM malli 1 olettaa jokaisen kohdistamisen yhtä todennäköiseksi (3.1) $\frac{1}{(l_f+1)^{l_e}}$. Tämän vuoksi IBM-malli 1 lauseen kääntämisessä sanojen järjestyksellä ei ole lopputuloksen kannalta merkitystä. Toisin kuin IBM-malli 1, malli 2 sisältää kohdistamisfunktion (3.6), joka antaa todennäköisyyden kyseisille kohdistamisille.

Tuloksessa tarkastellaan yhden lauseen käännöstä molempien IBM-mallien mukaan suhteessa iteraatioon. Aikaisemmin IBM-mallin 1 kuvaajissa on kuvattu kahdenkymmenennen iteraatiota, mutta viime alaluvun perusteella voidaan iteraatiot rajoittaa kymmeneen kuvaajien vertailun helpottamiseksi. Kohdistamisen arvoina käytetään IBM-mallin 1 kohdalla $\frac{1}{(3+1)^3}$ ja IBM-mallin 2 kohdalla parasta mahdollista kohdistamista. Myöskin normalisaatiovakio on asetettu yhdeksi luettavuuden helpottamiseksi. Kuten aiemmista tuloksista huomattu kuten kuvaajat 4.3 ja 4.4 ei normalisointi vaikuta mallien suhteelliseen eroon.



Kuva 4.5. Lauseen “umwelt ist wichtig“ käännöksen “environment is important“ todennäköisyys IBM-malleille 1 ja 2 EM-algoritmin iteraatioiden suhteen.

Kuvaajasta 4.5 nähdään selvästi IBM-mallin 2 ylivoimainen paremmuus suhteessa malliin 1. Tämä johtuu melkein täysin kohdistamisen todennäköisyyksien erosta, sillä viime alaluvun tuloksien perusteella yksittäisen sanan kääntämisen todennäköisyydet ovat mallien välillä samankaltaiset. Jolloin ainut ero käännöksen laskennassa mallien välillä on kohdistaminen.

Kuvaaja 4.5 on kuitenkin puutteellinen, sillä siinä esitellään vain parhaimman kohdistamisen tapausta. Mikäli lauseen sanajärjestystä muutetaan, olisivat tulokset huomattavasti erilaiset. Tällöin IBM-mallin 1 antama todennäköisyys olisi uskottavasti suurempi, mikä jälleen vahvistaa IBM-mallin 2 paremmuutta suhteessa IBM-malliin 1.

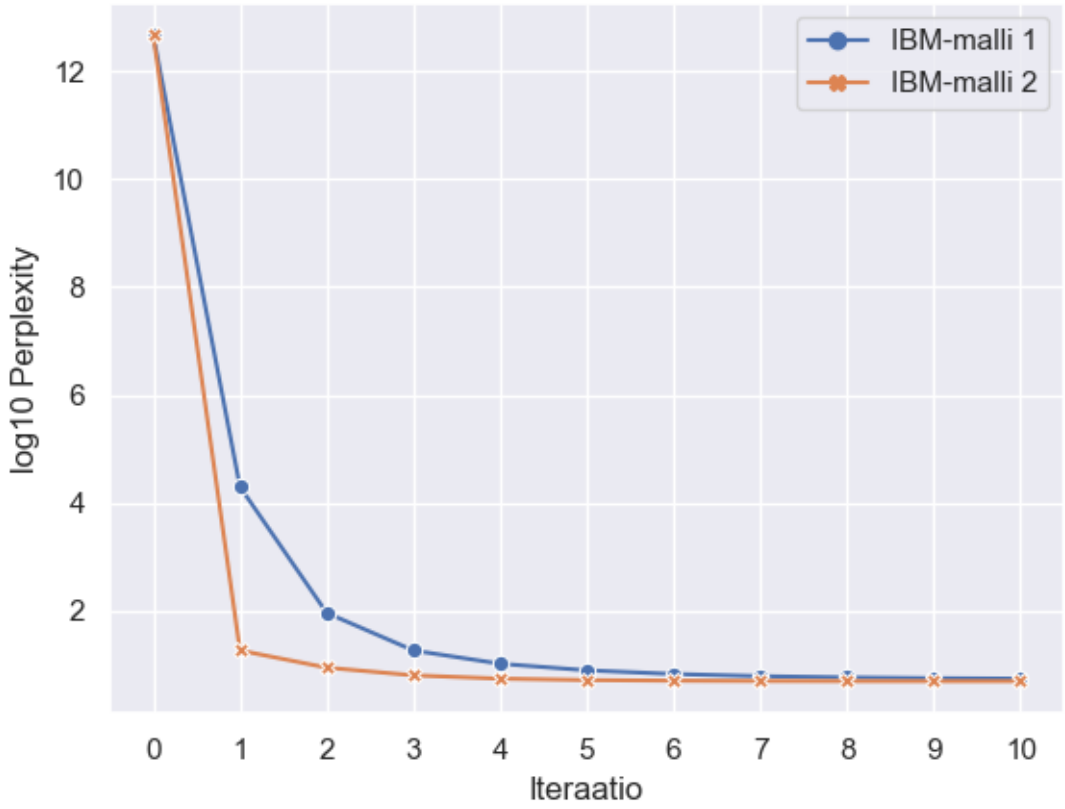
Mallien käännösten todennäköisyyksissä on pieniä eroja, kun kyseessä on yksittäinen sana. Tällöinkin malli 2 kouluttautuu huomattavasti nopeammin. Kun kyseessä on kokonainen lause nousevat esiin mallien suuremmat erot. IBM-mallin 1 lauseen todennäköisyys kasvaa vähän iteraatioiden suhteen. Sillä mallin antaman todennäköisyyden arvo muuttuu ainoastaan sanojen todennäköisyyksien tulon mukaan. Mallin 2 kohdalla on huomattavaa kasvua iteraatioiden välillä sekä myös suhteessa malliin 1. Koska mallin todennäköisyydet perustuvat sanojen käännösten tulon lisäksi kohdistamisfunktioon. Vaikka kuvaaja antaa selkeitä eroja mallien välillä tullaan

näitä tarkemmin testaamaan perplexityn avulla, mikä tuo numeerisen arvon mallin hyvyydelle.

4.2 Mallin laadun testaaminen perplexityn avulla

Mallien eroja on tarkastelu käännöksen todennäköisyyksien mukaan. Tämä tuo ongelmia, kuten todennäköisyyden pienet arvot, etenkin lauseiden todennäköisyyksiä käsitellessä. Vertailun parantamiseksi voidaan käyttää perplexityä, jonka arvo laskee, kun todennäköisyys kasvaa. Tällöin on konkreettinen numeerinen arvo, minkä avulla verrata malleja.

Tuloksien laskennassa käytetään edellisen alaluvun tuloksissa esitettyjä lauseita “umwelt ist wichtig” ja “environment is importat” sekä näiden parasta ja todennäköisintä käännöstä. Myöskin normalisatiovakio oletetaan yhdeksi. Ensimmäisten iteraatioiden perplexityn suuruuden vuoksi arvot logaritmoidaan kymmenenkantaisen logaritmin mukaan. Tällöin arvoista pystytään muodostamaan tarkkoja kuvaajia.



Kuva 4.6. Lauseen “umwelt ist wichtig“ käännöksen “environment is important“ kymmenkantainen logaritmi perplexity:stä ($log_{10}(PP)$) IBM-malleille 1 ja 2 EM-algoritmin iteraatioiden suhteen.

Kuvaajassa 4.6 nähdään kymmenlogaritmoitu perplexityn arvo IBM-malleille. Arvot alkavat samasta suuresta arvosta koska molemmat mallit asetetaan aluksi ta-

sajakaumaan. Tällöin jokaisen sanan käännöksen todennäköisyys on hyvin lähellä nollaa, milloin perplexity on suuri. Arvot laskevat huomattavalla nopeudella ensimmäisten iteraatioiden aikana, mutta jälleen IBM-malli 2 laskee nopeammin. Mallien väliset erot lähestyvät yhdeksännen iteraation kohdalla, milloin molempien arvot näyttävät olevan jo lokaalissa maksimissa. Arvot eivät kuitenkaan ole yhtä suuria, vaan kuvaajan tarkkuuden vuoksi on tulokset jäävät vajaiksi. Tarkemmin perplexityn arvot nähdään seuraavan taulukon avulla.

lg(Perplexity)						
IBM-malli 1	12.6696	4.2997	1.9636	$\cdots$	0.7904	0.7667
IBM-malli 2	12.6696	1.2668	0.9489	$\cdots$	0.6985	0.6983

Taulukko 4.1. lauseen “umwilt ist wichtig“ käännöksen “enviroment is important“ kymmen logaritmoidun perplexityn ($log_{10}(PP)$) arvo iteraatioista nollasta kymmeneen

Taulukosta 4.1 nähdään samat arvot kuin Kuvaajassa 4.6, mutta tarkemmin. Kun tarkastellaan ovatko IBM-mallit saavuttaneet saman lokaalin maksimin kuvaajan näyttävän yhdeksännen iteraation kohdalla, huomataan että tämä ei pidä paikkaansa. Tässäkin kohdassa mallien välinen perplexityn suhde on noin 1.13, eli IBM-mallin 1 perplexityn arvo kohdassa yhdeksän on 13 % korkeampi, kuin IBM-mallilla 2.

Kuvaajan 4.6 ja Taulukon 4.1 tuloksien perusteella IBM-mallien erot kasvavat entisestään. Aikaisemmin IBM-malli 2 on johtanut aikaisempaa versiotaan, niin yksittäisen sanan tai lauseen käännöksen todennäköisyydessä. Myöskin IBM-malli 2 on osoittanut jokaisessa tapauksessa nopeampaa konvergoitumista. Lisäksi malleja testatessa perplexityn avulla huomataan jälleen jopa yli 10 % parempi tulos IBM-malli 2 suhteessa malliin 1. Kaikkien näiden tuloksien pohjalta voidaan todeta, että IBM-malli 2 on huomattavasti nopeampi sekä tarkempi käännösmalli, kuin IBM-malli 1, ilman suuria laskennallisen tehon eroja. Tällöin käännösmallia valitessa, etenkin kokonaista lausetta kääntäessä on IBM-malli 2 ilmiselvä valinta.

5 Johtopäätökset

Tutkielman johdannossa tavoitteeksi on asetettu IBM-mallien 1 ja 2 välinen vertailu sanojen sekä lauseiden käännöksien todennäköisyyksissä, ja perplexityn arvoissa. Työhön valittiin 100 000 lauseen aineisto josta työn eri kouluttamisen iteraatioiden mallit ovat koulutettu. Tuloksissa huomattiin, että yksittäisen sanan käännöksen todennäköisyys ei paljoa vaihtele mallien välillä, tästä huolimatta IBM-malli 2 saavuttaa lopullisen arvon nopeammin. Kokonaisen lauseenkäännöksessä mallien erot kasvavat kohdistamisen myötä. IBM-mallin 1 todennäköisyydet kasvavat vain hieman iteraatioiden suhteen, kun taas IBM-malli 2 kasvaa hurjasti ja saa melkein kymmenkertaisen suuremman todennäköisyyden. Perplexityä tarkastellessa erot jälleen pienenevät, mutta IBM-malli 2 saa noin 10 % paremmat arvot ja saavuttaa tämän nopeammin, kuin IBM-malli 1.

Käännösten todennäköisyyksien tulokset vastaavat tutkielmassa asetettuja odotuksia. Perplexity mallien välillä on kuitenkin odotettua huonompi. Tämä johtuu siitä, että perplexity ei ota huomioon kohdistamista vaan ainoastaan sanojen käännöksien todennäköisyydet. Muita mahdollisia tuloksien haittakohtia ovat aineiston koko ja mallien testauksen yksinkertaisuus. Koko on suositeltua pienempi, minkä takia käännökset saattavat olla puutteellisia sanojen harvinaisuuden takia. Tällöin voi käydä esimerkiksi siten, että tiettyä sanaa ei ole aineistossa olemassa, jolloin käännöksen luominen on mahdotonta. Malleja on testattu ainoastaan yhtä sanaa tai lausetta käyttäen, milloin ei saada kokonaiskuvaa mallin käyttäytymisestä. Testauksen niukkuus voi aiheuttaa sen, että tutkielmaan on valittu sellaiset esimerkit, jotka käyttäytyvät haluamallaan tavalla. Ongelmista huolimatta tutkielma antaa yksinkertaisen käsityksen IBM-mallien 1 ja 2 eroille.

Vaikka IBM-mallit 1 ja 2 ovat jo vanhentuneita nykyisen konekääntämisen vierellä, on näiden toiminta näkyvissä nykyisissä käännösmalleissa. Tällöin IBM-mallien etenkin kohdistamisen ymmärtäminen on tärkeä osa tilastollista konekääntämisestä. Aihealuetta voisi jatkaa tutkimalla korkeamman tason IBM-malleja tai muita tilastollisia käännösmalleja. Sekä vertailemalla näitä neuroverkostoilla muodostettuihin käännösmalleihin pienillä aineistoilla.

Lähteet

Brown, P. F. et al. (1993). ”The Mathematics of Statistical Machine Translation: Parameter Estimation”. *Comput. Linguist.* 19.2, s. 263–311. ISSN: 0891-2017.

Gavrila, M. ja C. Vertan (2011). ”Training Data in Statistical Machine Translation - the More, the Better?” Teoksessa: s. 551–556.

Imam, A. H. et al. (2011). ”Impact of corpus size and quality on English-Bangla statistical Machine Translation system”. Teoksessa: *14th International Conference on Computer and Information Technology (ICCIT 2011)*, s. 566–571. DOI: 10.1109/ICCITechn.2011.6164853.

Koehn, P. (2010). *Statistical machine translation*. Cambridge: Cambridge University Press.

Thanaki, J. (2017). *Python Natural Language Processing*. Packt Publishing.

Liitteet

Liite 1. Esimerkki aineiston ensimmäisistä viidestä rivistä

1. Resumption of the session
2. I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.
3. Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful.
4. You have requested a debate on this subject in the course of the next few days, during this part-session.
5. In the meantime, I should like to observe a minute' s silence, as a number of Members have requested, on behalf of all the victims concerned, particularly those of the terrible storms, in the various countries of the European Union.

1. Wiederaufnahme der Sitzungsperiode
2. Ich erkläre die am Freitag, dem 17. Dezember unterbrochene Sitzungsperiode des Europäischen Parlaments für wiederaufgenommen, wünsche Ihnen nochmals alles Gute zum Jahreswechsel und hoffe, daß Sie schöne Ferien hatten.
3. Wie Sie feststellen konnten, ist der gefürchtete "Millenium-Bug " nicht eingetreten. Doch sind Bürger einiger unserer Mitgliedstaaten Opfer von schrecklichen Naturkatastrophen geworden.
4. Im Parlament besteht der Wunsch nach einer Aussprache im Verlauf dieser Sitzungsperiode in den nächsten Tagen.
5. Heute möchte ich Sie bitten - das ist auch der Wunsch einiger Kolleginnen und Kollegen -, allen Opfern der Stürme, insbesondere in den verschiedenen Ländern der Europäischen Union, in einer Schweigeminute zu gedenken.

Liite 2. Esimerkki kohdistamisen toiminnasta