Piltide automaatne kirjeldamine eesti keeles - visuaalse ja semantilise ühisesituse õppimine neurovõrkudega

Magistritöö matemaatilise statistika erialal (30 EAP)

Tanel Pärnamaa

Matemaatika-informaatikateaduskond
Matemaatilise statistika instituut

Tartu Ülikool

Juhendajad: Leopold Parts, Sven Laur

2015

Piltide automaatne kirjeldamine eesti keeles - visuaalse ja semantilise ühisesituse õppimine neurovõrkudega

Selle töö eesmärgiks on treenida *statistiline masin* ehk algoritm, mis on võimeline pilte eesti keeles kirjeldama. Vastav mudel oleks kasulik nii pildiotsingul kui ka nägemisvaegustega inimestele navigeerimisel.

Eesti keel on morfoloogiliselt rikas (palju käändied ja pöördeid), mis teeb selle modelleerimise keeruliseks. Enne kui on võimalik genereerida grammatiliselt korrektset kirjeldust, tuleb osata lauseid ja sõnu informatiivselt esitada. Selleks uurin neurovõrkudel põhinevaid meetodeid.

Lisaks on eestikeelsed andmekogud tihti väiksemad kui analoogilised ingliskeelsed korpused. Uurin, kuidas kanda tarkust üle suurtest ingliskeelsetest andmekogudest, et eesti keeletehnoloogia rakenduste tulemusi parandada.

Treenin uudse neurovõrkudel põhineva tõlkesüsteemi ingliskeelsete lausete tõlkimiseks eesti keelde. Näitan, et analoogilise mudeliga saab tõlkida ka pilte tekstiks. Töö käigus valmib esimene mudel, mis on edukalt võimeline pilte loomulikus eesti keeles kirjeldama.
Märksõnad: *masinõpe, loomuliku keele töötlus, masintõlge, tehisnägemine, statistiline masin, neurovõrgud*

Translating pictures to Estonian - learning shared representations of images and languages using neural networks

The aim of this thesis is to train a *statistical machine* to describe images in natural Estonian language. The model could have an enormous impact on image search and would be helpful for visually impaired people to better understand the content of images.

Estonian language is highly inflective (one noun lemma could have 28 different forms) making it one of the hardest language to model. Before we can generate novel image descriptions, great care must be taken on how to represent words and sentences. For that, I study the effectiveness of neural embeddings.

Moreover, Estonian is a low resource language. I study how to use English as a pivot to improve Estonian natural language applications (i.e increase the semantic information in word embeddings, classify documents and describe images without proper Estonian corpora, translate from one language to another).

Recent advances in computer vision (the use of deep convolutional neural networks), language modelling (the use of recurrent neural networks) and multimodal modelling (combining the models) have made it possible to achieve novel and morphologically rich image descriptions in Estonian.
Keywords: *machine learning, natural language processing, machine translation, computer vision, statistical machine, neural networks*

Sisukord

Sissejuhatus **1**

1 Sõnadest arusaamine **5**
1.1 Sõnade esitamine statistilistes algoritmides 5
1.2 Distributiivsemantika algoritmid 6
1.2.1 Singulaarväärtuslahutus 7
1.2.2 word2vec 8
1.2.3 Edasileviga neurovõrk 11
1.2.4 Rekurrentne neurovõrk 12
1.3 Eksperimendid 14
1.3.1 Analoogiaülesanne 15
1.3.2 Sõnade tõlkimine 17
1.3.3 Keeltevaheline siirdeõpe 20

2 Lausetest arusaamine **23**
2.1 Lausete esitamise viisid statistilistes algoritmides 23
2.1.1 Sageduspõhised meetodid 23
2.1.2 Mudelipõhised meetodid 25
2.2 Eksperimendid 27
2.2.1 Lause meelsuse klassifitseerimine 27
2.2.2 Sõnavektorite algväärtustamine 28
2.2.3 Kvalitatiivne võrdlus 29
2.2.4 Keeltevaheline siirdeõpe 31

3 Lausete tõlkimine **33**
3.1 Fraasipõhine statistiline masintõlge 33
3.2 Neurovõrkudel põhinev tõlkimine 35
3.2.1 Kodeerija-dekodeerija süsteem 35

3.3 Automaatne tõlkekvaliteedi hindamine 38
3.4 Eksperimendid 39
3.4.1 Kvalitatiivne analüüs 40
3.4.2 Kvantitatiivne analüüs 41

4 Piltidest arusaamine **47**
4.1 Piltide kirjeldamine ühe sõnaga 48
4.2 Piltide kirjeldamine mitme sõnaga 50
4.3 Piltide kirjeldamine lausega 51
4.4 Eksperiment: automaatne piltide kirjeldamine eestikeelse lausega 54

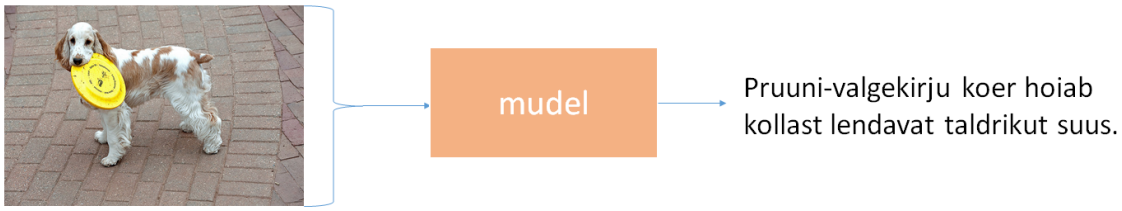
Kokkuvõte **59**

Kirjandus **61**

Lisa A Sõnavektorite lisatulemused **65**

Sissejuhatus

Selle töö eesmärgiks on treenida *statistiline masin* ehk algoritm, mis on võimeline mõistma ja kirjeldama pilte eesti keeles. Näiteks andes mudelile ette koera pildi (joonis 1), peaks mudel saama aru, et pildil on koer, lendav taldrik ning lendav taldrik on koeral suus. Mõistes seda, peaks mudel tagastama grammatiliselt korrektse eestikeelse kirjelduse.



Joonis 1 Sisestades pildi, tagastab mudel loomulikus keeles kirjelduse.

Selline mudel oleks kasulik nii pildiotsingul (näiteks soovides otsida personaalsest pildipangast fotosid, kus kass on jõehobu seljas) kui ka nägemisvaegustega inimestele navigeerimisel ning piltide mõistmisel.

Eelnimetatud eesmärk tundub aga utoopilisena. Hiljuti ei suutnud pilditöötluse algoritmid piisavalt hästi tuvastada sedagi, kas pildil on koer või kass. Rämpskommentaaride vastu võitlemiseks on paljud veebilehed kaitstud ülesandega, mida on lihtne lahendada inimesel aga keeruline automatiseerida läbi algoritmi. Üheks selliseks ülesandeks oligi tuvastada, kas pildil on kass või koer (joonis 2) [10]. Tehisnägemise teeb raskeks, et tüüpiliselt esitatakse pilte pikslite intensiivsustena ja vaja on arvesse võtta objekti suurust, suunda, asukohta, taustamüra, valgustust, moonutust, kattuvust, klassisisest erinevust ja muud.

Lisaks on keele modelleerimine keeruline. Näiteks ei võta tüüpilised keeletehnoloogia rakendused arvesse informatsiooni, et sõna *kass* on tähenduslikult palju lähedasem sõnale *koer* kui sõnale *lubjabakter*. Selle kõrvalt tundub grammatiliselt korrektse vabavormilise kirjelduse genereerimine mõeldamatuna.

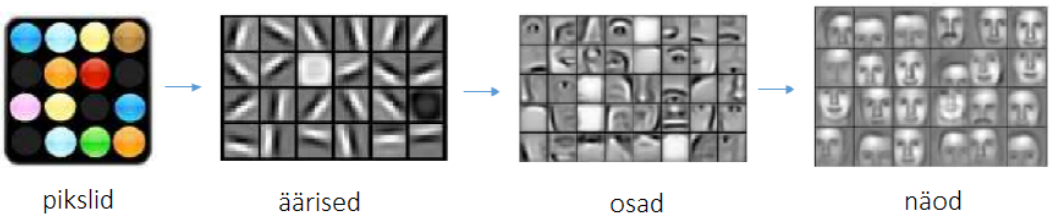
Joonis 2 2007. aastal töötasid Microsofti teadurid välja ülesande, mida on lihtne lahendada inimesel, aga raske masinal [10]. Oli vaja tuvastada, kas pildil on kass või koer. Kuigi 2007. aastal polnud algoritmid eriti paremad juhuslikust arvamisest, siis kasutades süvaõppemeetodeid on masinad 2014. aastal võimelised 99% juhtudel eristama kassi koerast [17].

Ülesandele lisab keerukust, et mudel peab tagastama eestikeelse kirjelduse. Eesti keel on morfoloogiliselt rikas (palju käändeid ja pöördeid), mis muudab võrreldes inglise keelega modelleerimise raskemaks. Lisaks on eestikeelsed tekstikogud (ehk korpused) väiksemad ning teatud ülesannete jaoks vajalikku andmestikku polegi.

Kui inglise keele modelleerimisel kasutatakse korpusi, milles on üle 100 miljardi sõna, siis sarnase kvaliteediga eesti keelse tekstikogu saamiseks läheks üle 4000 aasta (võttes arvesse viimase 10 aasta meediaväljannete teksti hulka). Lisaks on erinevate korpuste tekitamine ja sildistamine kallis. Seega on eesti keeletehnoloogia seisukohast ääretult oluline küsimus, kuidas kanda tarkust üle võõrkeelsest tekstist, et eestikeelsete rakenduste tulemusi parandada. Siirdeõpe (*transfer learning*) on olnud läbivaks teemaks selles töös.

Teiseks keskseks teemaks on olnud süvaõppe meetodite (*deep learning*) kasutamine. Need mudelid põhinevad ideel, et andmetest õpitakse automaatselt mitu kihti tunnuseid (joonis 3). Süvaõpe on hiljuti revolutsioneerinud pildituvastust [27], kõnetuvastust [14] ja on kiiresti muutmas keeletehnoloogiat (näiteks masintõlget [49]).

Grammatiliselt korrektse kirjelduseni jõuan sammhaaval. Enne kui saame tegeleda lausetega, peab *statistiline masin* mõistma sõnade tähendust. Esimeses peatükis uuringi, kuidas statistilistes algoritmides esitada sõnu, et need sisaldaksid endas ka semantilist ja grammatilist informatsiooni. Muuhulgas uurin, kas distributiivsemantika algoritmid suudavad automaatselt tuvastada keelelisi konseptsioone nagu käänded ja pöörded ning kuidas automaatselt tõlkida



Joonis 3 Valik tunnustest, mida süvaõppe meetodid leiavad nägude andmestikust [32]. Iga järgmise kihiga õpitakse kõrgematasemelised tunnused.

ingliskeelseid sõnu eesti keelde. Selle käigus koostasin eestikeelse analoogiaandmestiku, mille peal saab võrrelda sõnade vektoresituste headust.

Sõnadest üksi ei piisa, et keelt modelleerida. II peatükis uurin, kuidas sõnavektorite esitusi üldistada lausetele. Võrdlen erinevaid lausete esitusi eestikeelse teksti meelsuse tuvastamisel. Selle käigus näitan, kuidas edukalt kasutada suurt tekstikorpust spetsiifilise ülesande (meelsuse tuvastamise) täpsuse parandamiseks. Lisaks õpin klassifitseerija ingliskeelsel tekstil ning tutvustan, kuidas seda sama klassifitseerijat kasutada eestikeelsete tekstide kategoriseerimisel.

Eesti keele jaoks pole andmestikku piltidest ja nende kirjeldustest, mille põhjal saaks otse õppida generatiivse mudeli, mille sisendiks on pilt ja väljundiks lause. Seega valisin tee, kus pildile genereerin kõigepealt ingliskeelse kirjelduse ja seejärel tõlgin selle eestikeelseks. Lausete tõlkimiseks treenin rekurrentsetel neurovõrkudel põhineva inglise-eesti tõlkemudeli (peatükk III). Kuigi vastav süvaõppemeetoditel põhinev tõlkesüsteem pakuti välja alles 2013. aastal [18], on see andnud väga häid tulemusi inglise keelest prantsuse keelde tõlkimisel [49]. Näitan, et ka eesti keelde tõlkimisel on tulemused keeleliselt ilusad (tihti paremad kui näiteks aastaid häälestatud tõlkesüsteemi *Google Translate* tõlked).

Neurovõrkudel põhineva masintõlke ideed rakenduvad ka piltide *tõlkimisele*. Märkamine, et automaatne pildikirjeldamine pole muud kui pilditunnuste tõlkimine teise keelde, lubab sujuvalt liikuda ühest äärmusest (keeletehnoloogia) teise (tehisnägemine). IV peatükis kirjutan, kuidas pilti iseloomustada ühe sõna, mitme sõna ja lausega. Koostasin esimese süsteemi, mis on edukalt võimeline pilte kirjeldama loomulikus eesti keeles.

Soovin tänada Ruslan Salakhutdinovi süvaõppe meetodite põhitõdede õpetamise eest. Olen tänulik Sven Laurile, kellega rääkides jääb tahvlipinnast alati puudu. Erilised tänusõnad kuuluvad Leopold Partsile, kelle maagilised kommentaarid ning innustamine jätavad sind mõtlema, uudistama ja katsetama aastateks.

Peatükk 1

Sõnadest arusaamine

Kui *statistilisele masinale* anda ette 10 aasta Postimehe artiklid, kas ta on võimeline aru saama sõnade tähendusest?

Täpsemalt mõtlen sõna all tekstisõna (iga eraldi sõna tekstis). Näiteks linna *New York* käsitlen kahe eraldi sõnana: *New* ja *York*. Samuti käsitlen kirjavahemärke (näiteks *.,!*) kui eraldi sõnu.

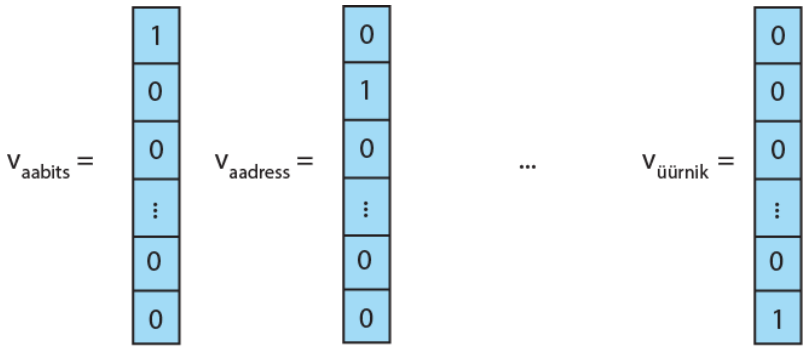
Sõnade mõistmise all mõtlen arusaamist nii leksikaalsest ehk semantilisest tähendusest (sõna *kass* on mõiste sisult lähemal sõnale *koer* kui sõnale *sinepipulber*) kui ka grammatilisest tähendusest (sõnad *kassiga* ja *sinepipulbriga* on samas käändes).

1.1 Sõnade esitamine statistilistes algoritmides

Tüüpiliselt esitatakse sõnu keeletehnoloogilistes rakendustes 1-S kodeeringus. See tähendab, et sõna esitatakse S-elemendilise indeksvektorina, kus sõna indeksile vastav element on 1, kõik ülejäänud elemendid on nullid (joonis 1.1).

Sellisel esitusviisil on mitmeid puudusi. Esiteks puudub neil vektoritel semantiline tähendus. Sõnale *kass* vastav vektor on sama kaugel nii vektorist *koer* kui ka vektorist *lubjakivibakter*. Teiseks probleemiks on kõrge dimensionaalsus. Sõnastikes on tihti üle 100 000 sõna ning harva esinevate sõnade modelleerimine on keeruline. Samuti võib arvata, et eksisteerib S'-mõõtmeline ruum, kus S' on palju väiksem S-st, mis on piisav kogu semantilise informatsiooni kodeerimiseks. Näiteks üks mõõde võib kodeerida mitmust, teine ajavormi. Lisaks on palju mugavam töötada väiksemate, näiteks 300-mõõtmeliste vektoritega.

Nende puuduste parandamiseks on välja töötatud mitmeid statistilisi algoritme, mis õpivad sõnadele vektoresituse. Enamik neist algoritmidest kasutavad ära sõnade koosesinemise sagedust ja põhinevad distributiivsuse hüpoteesil - sõnadel, mis esinevad samades kontekstides, on sarnane tähendus.



Joonis 1.1 Tüüpiliselt esitatakse sõnu keeletehnoloogia rakendustes vektorina, kus kõik elemendid on nullid, välja arvatud sõnale vastava indeksi element, mis on üks. Vektori mõõde on võrdne sõnastiku suurusega.

1.2 Distributiivsemantika algoritmid

Sõna w_i kontekstiks nimetatakse teda ümbritsevaid sõnu $w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}$, kus c on etteantud akna laius. Sõnade koosesinemise maatriksit tähistan tähega X, kus X_{ij} näitab, mitu korda esines sõna w_j sõna w_i kontekstis. Maatriks X on mõõtmetega $S \times S$, kus S on sõnastiku suurus.

Maatriksi X rida X_i on i-nda sõna üks võimalik distributiivne esitus. Vastav esitus sisaldab nüüd ka semantilist informatsiooni, kuid seda domineerivad sageli esinevad rämpssõnad (näiteks sidesõnad *ja, et, ning*).

Seega kaalutakse tihti sõnade koosesinemise maatriksit X. Sage kaalumisviis on positiivne punktiviisiline vastastikune informatsioon (PPMI):

$$PPMI(x,y) = \max\{PMI(x,y),0\}, \quad \text{kus}$$

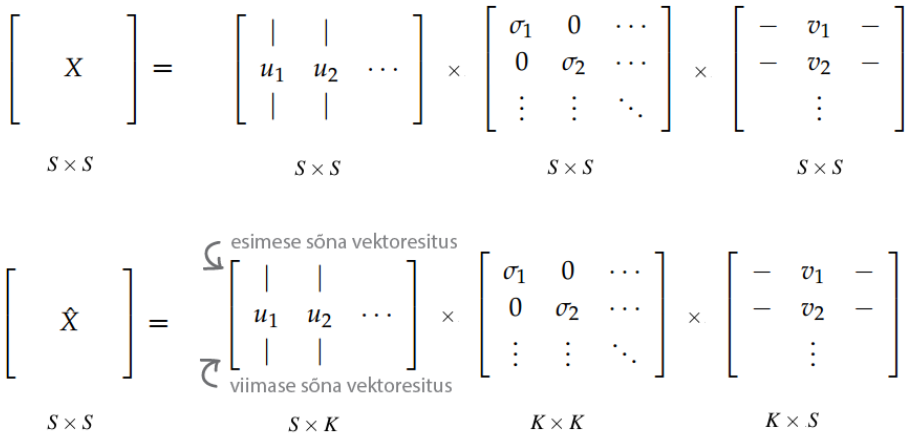
$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)}.$$

PMI mõõdab omavahelist seost sõna w_i ja kontekstisõna w_j vahel arvutades logaritmi ühisjaotuse $p(w_i,w_j)$ ja marginaaljaotuste $p(w_i), p(w_j)$ suhtest. PMI maatriksi kasutamine tekitab arvutuslikke raskusi, sest maatriksis on mitmeid sõna-konteksti paare, mida ei esinenudki korpuses. Sel juhul oleks $PMI(w_i,w_j) = \log 0 = -\infty$. Üheks võimaluseks on siluda seda maatriksit Dirichlet' eeljaotusega (lisades maatriksile pisikese võltsesinemissageduse). Teiseks võimaluseks on kasutada seosemõõtu $PPMI$.

Vastavate esituste mõõtmed on aga ikka võrdsed sõnastiku suurusega S. Singulaarväärtuslahutus on üks viis vektori dimensionaalsuse vähendamiseks.

1.2.1 Singulaarväärtuslahutus

Singulaarväärtuslahutus (SVD) faktoriseerib $S \times S$ maatriksi X korrutiseks $U\Sigma V^T$, kus U on $S \times S$ ortogonaalne sõnavektorite maatriks, Σ on $S \times S$ diagonaalne singulaarväärtuste maatriks ja V on $S \times S$ ortogonaalne kontekstivektorite maatriks (joonis 1.2). Dimensionaalsuse vähendamiseks võime väiksemad singulaarväärtused muuta nulliks ning jätta alles K suurimat singulaarväärtust ning vastavad K rida maatriksitest U ja V. See annab parima astakuga K lähenduse $\hat{X} = U_{:,1:K}\Sigma_{1:K,1:K}V^T_{:,1:K}$ maatriksile X Forbeniuse normi mõttes. Selliselt saame i-ndale sõnale K-elemendilise esituse $U_{i,1:K}$.



Joonis 1.2 SVD faktoriseerib maatriksi X kolme maatriksi korrutiseks: $U\Sigma V^T$. Jättes alles K suurimat singulaarväärtust ning vastavad read maatriksites U ja V, saame esialgsele maatriksile parima lähendi $\hat{X}$ Forbeniuse normi mõttes, kus $\hat{X}$ astak on K.

Vaistlikult võib SVD rakendamisest sõnade koosesinemise maatriksile X mõelda kui tekstis olevate semantiliste kontseptsioonide automaatset leidmist[30]. Näiteks sõnad *ämber* ja *pang* on väga sarnased ning dimensionaalsuse vähendamise meetodid võiksid esitada neid ühe semantilise klassina. Keeletehnoloogid kutsuvadki SVD rakendamist sõnade koosesinemise maatriksile X kui varjatud semantiline analüüs (*latent semantic analysis*).

SVD abil saadud sõnade vektoresitustel on mitmeid puudusi. Esiteks tuleb koostada sõnade koosesinemise maatriks X, mille mõõtmed on väga suured ($S \times S$, kus S on sõnastiku suurus) ning kus enamik elemente on nullid. Maatriksit X tuleb kuidagi kaaluda, et võtta arvesse sõnade erinevat esinemissagedust. Kui soovime sõnade vektoresitust uuendada uute andmete peal, tuleb maatriksi kaalumine ja SVD faktoriseerimine uuesti läbi teha.

1.2.2 word2vec

Selle asemel, et sõnade koosesinemise sagedusi loendada ning mõelda, kuidas oleks sõnade koosesinemiste maatriksit kõige optimaalsem kaaluda ning selle dimensionaalsust vähendada, võiks olla üks mudel, mis teeb seda kõike korraga.

Üks võimalus selleks on otsida sõnade vektoresitusi, mis aitavad modelleerida keelt. Näiteks vektoresitused, mis aitavad välja valida grammatiliselt ning semantiliselt korrektseid lauseid, võiksid olla kasulikud mitmetes rakendustes (näiteks nimeolendite tuvastamisel).

Et valida välja keeleliselt ilusaid lauseid, on vaja lausetele osata määrata skoore või tõenäosusi. Näiteks lause *"Magistritööd on tore teha!"* on täiesti korrektne lause ning mudel peaks sellisele lausele tagastama suure tõenäosuse. Lause *"Teha ! tore magistritööd on"* peaks saama aga väiksema tõenäosuse.

Seega soovime lause $w_1, w_2, ..., w_n$ korral modelleerida tõenäosust $p(w_1, w_2, ..., w_n)$. Vastavat tõenäosust saab faktoriseerida ketireegliga:

$$p(w_1, w_2, ..., w_n) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \cdots p(w_n|w_1, w_2, ..., w_{n-1}).$$

Praktikas on tõenäosust $p(w_i|w_1, w_2, ..., w_{i-1})$ keeruline modelleerida ning tihti tehakse lihtsustav Markovi eeldus:

$$p(w_i|w_1, w_2, ..., w_{i-1}) \approx p(w_i|w_{i-k}, ..., w_{i-1}).$$

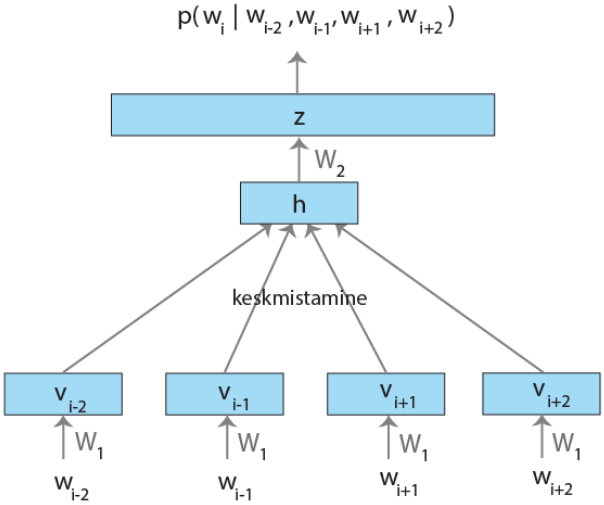
Modelleerides tõenäosust $p(w_i|w_{i-k}, ..., w_{i-1})$ otsime sõnadele vektoresitusi, mis aitavad ennustada järgmist sõna. Selle käigus aga ignoreerime sõna w_i parempoolset konteksti $w_{i+1}, ..., w_{i+k}$. Seega võib olla mõistlikum õppida sõnadele vektoresitusi modelleerides hoopis $p(w_i|w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k})$. Või pöörata mudel ümber ja otsida vektoresitusi, mis aitavad kontekstisõnu ennustada (st modelleerides $p(w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k}|w_i)$).

2013. aasta artiklis [39] pakuti välja kaks moodust, kuidas sõnadele õppida vektoresitusi (modelleerides $p(w_i|w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k})$ või $p(w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k}|w_i)$). Esimest mudelit nimetati *continuous bag of words* (CBOW) mudeliks ning teist *skip-gram* mudeliks. Muuhulgas avaldati tarkvara *word2vec*[1], millega vastavaid mudeleid õppida. Tarkvara kogus kiiresti populaarsust, sest mudeleid oli kerge ja kiire õppida ning leitud vektoresitused sisaldasid üllatavalt palju semantilist informatsiooni. Tarkvaras *word2vec* implementeeritud mudelid on tuntud just *word2vec* nime all.

[1] https://code.google.com/p/word2vec/

CBOW mudel

CBOW otsib selliseid vektoresitusi, mis kontektsisõnade $w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}$ olemasolul aitavad ennustada keskmist sõna w_i (joonis 1.3). 1-S kodeeringus kontekstisõnad projitseeritakse K-mõõtmelisse esitusse kasutades $K \times S$-mõõtmelist maatriksit W_1. Vastavad esitused kombineeritakse K-elemendiliseks peidetud kihiks h ning kasutades $S \times K$-mõõtmelist maatriksit W_2 saadakse S-mõõtmeline skoorivektor, mis muudetakse tõenäosusteks.



Joonis 1.3 CBOW mudel otsib vektoresitusi, mis kontekstisõnade olemasolul aitavad ennustada keskmist sõna.

Algoritmilised sammud prognoosi saamiseks on järgmised:

1. Olgu sõna w_i ja selle kontekst $w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}$ esitatud 1-S kodeeringus.

2. Transformeeri konteksti sõnad uude esitusse:

$$v_{i-c} = W_1 w_{i-c}, ..., v_{i-1} = W_1 w_{i-1}, v_{i+1} = W_1 w_{i+1}, ..., v_{i+c} = W_1 w_{i+c}.$$

3. Võta neist vektoritest aritmeetiline keskmine:

$$h = \frac{v_{i-c} + ... + v_{i-1} + v_{i+1} + ... + v_{i+c}}{2c}.$$

4. Leia skoorivektor $z = W_2 h$.

5. Muuda skoorivektor z tõenäosusteks kasutades multinomiaalset logistilist funktsiooni:

$$p(w_i|w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}) = \frac{e^{z_{I(w_i)}}}{\sum_{m=1}^{S} e^{z_m}},$$

kus $I(w_i)$ tähendab sõnale w_i vastavat indeksit.

Tundmatud on maatriksid W_1 ja W_2. Need õpitakse selliselt, et mudeli prognoosid oleksid võimalikult lähedased näidisandmetele (minimiseerides ristentroopiat). Sõnade esitustena kasutatakse tüüpilselt maatriksi W_1 veerge.

skip-gram mudel

Skip-gram mudel modelleerib $p(w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}|w_i)$ ja otsib sõnadele vektoresitusi, mis on kasulikud kontekstisõnade ennustamiseks (joonis 1.4). Modelleerimise lihtsustamiseks tehakse naiivne tingliku sõltumatuse eeldus:

$$p(w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}|w_i) = \prod_{j=0,j\neq c}^{2c} p(w_{i-c+j}|w_i).$$

Tõenäosuse $p(w_j|w_i)$ leidmiseks projitseeritakse 1-S kodeeringus olev sõna w_i K-mõõtmelisse esitusse v_i kasutades $K \times S$-mõõtmelist maatriksit W_1. Esitust v_i käsitletakse kui peidetud kihti. See muudetakse S-mõõtmeliseks skoorivektoriks z kasutades $S \times K$-mõõtmelist maatriksit W_2, mis omakorda muudetakse tõenäosusteks.

Algoritmiliselt leitakse $p(w_j|w_i)$ järgmiselt (esitatud selliselt, et oleks võimalikult analoogiline CBOW mudelile):

1. Olgu sõna w_i ja selle kontekst w_j esitatud 1-S kodeeringus.

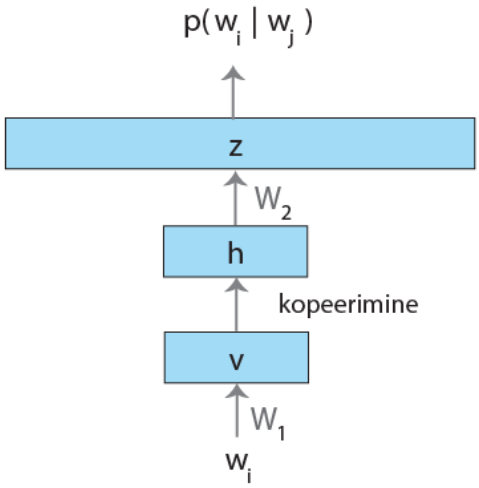
2. Transformeeri sõna w_i uude esitusse:

$$v_i = W_1 w_i.$$

3. Rakenda samasusfunktsiooni:

$$h = v_i.$$

4. Leia skoorivektor $z = W_2 h$.



Joonis 1.4 *Skip-gram* mudel otsib selliseid sõnade esitusi, mis aitavad prognoosida vasak- ja parempoolset konteksti.

5. Muuda skoorivektor z tõenäosusteks kasutades multinomiaalset logistilist funktsiooni:

$$p(w_j|w_i) = \frac{e^{z_{I(w_j)}}}{\sum_{m=1}^{S} e^{z_m}}.$$

Tundmatud on maatriksid W_1 ja W_2. Sõnade vektoresitustena kasutatakse tüüpilselt W_1 veerge.

1.2.3 Edasileviga neurovõrk

CBOW on log-lineaarne mudel. Keelemudeli seisukohast tahaksime mudelit, mis on võimeline modelleerima keerulisemaid otsustuspiire. Lisaks ei arvesta CBOW sõnade järjekorda. Neid kahte puudust saab parandada, kui lisada CBOW mudelisse mittelineaarsust ning konteksti sõnade vektoresitusi mitte keskmistada (joonis 1.5).

Edasileviga neurovõrgu keelemudeli ainuke erinevus võrreldes CBOW mudeliga ongi see, kuidas leitakse peidetud kiht h. Kontekstisõnade vektoresitused $v_{i-c}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+c}$ transformeeritakse maatrikskorrutise abil uude ruumi (kusjuures kasutatakse $2c$ erinevat maatriksit W_2^j, igaüks neist mõõtmetega $K \times H$, kus H on valitud peidetud kihi h suurus), vastavad tulemused liidetakse kokku, millele omakorda rakendatakse mõnd mittelineaarset funktsiooni f.

Mudeli algoritmiline kirjeldus prognoosi saamiseks on järgmine:

1. Olgu sõna w_i ja selle kontekst $w_{i-c},...,w_{i-1},w_{i+1},...,w_{i+c}$ esitatud 1-S kodeeringus. (Kui soovime saada keelemudeli, siis vaatama vaid vasakpoolset konteksti).

2. Transformeeri konteksti sõnad uude esitusse:

$$v_{i-c} = W_1 w_{i-c}, ..., v_{i-1} = W_1 w_{i-1}, v_{i+1} = W_1 w_{i+1}, ..., v_{i+c} = W_1 w_{i+c}.$$

3. Arvuta peidetud kihi tunnused:

$$h = f(W_2^1 v_{i-c} + ... + W_2^c v_{i-1} + W_2^{c+1} v_{i+1} + ... + W_2^{2c} v_{i+c}),$$

   kus f on mingisugune mittelineaarne funktsioon (näiteks sigmoidfunktsioon või *tanh*).

4. Leia skoorivektor $z = W_3 h$. (Maatriks W_3 on mõõtmetaga $H \times S$.)

5. Muuda skoorivektor z tõenäosusteks kasutades multinomiaalset logistilist funktsiooni:

$$p(w_i | w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k}) = \frac{e^{z_{I(w_i)}}}{\sum_{m=1}^{S} e^{z_m}}.$$

Tundmatud on maatriksid $W_1, W_2^1, ..., W_2^{2c}, W_3$. Sõnade uue esitusena kasutatakse maatriksi W_1 veerge.

Ajalooliselt töötati neuro-keelemudel välja varem kui CBOW [4]. Kuigi keerukam mudel on küll parem keelemudelina, siis sõnadele õpitud vektoresitused pole tingimata paremad lihtsama mudeliga saadud vektoresitustest [39].

1.2.4 Rekurrentne neurovõrk

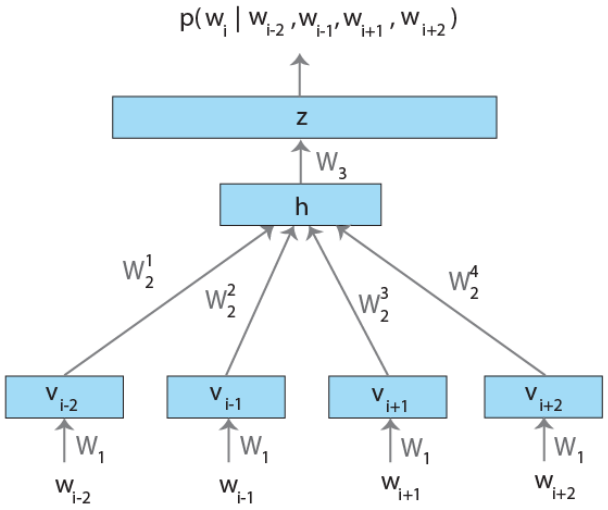
Eelnevate mudelite korral oleme kasutanud fikseeritud pikkusega konteksti (st tegime Markovi eelduse). Keelemudeli ja sõnade vektoresituse õppimisel võib see olla piirav. Rekurrentsel neurovõrgul põhinev keelemudel aga ei kasuta fikseeritud suurusega konteksti (joonis 1.6).

Mudeli algoritmiline kirjeldus on järgmine:

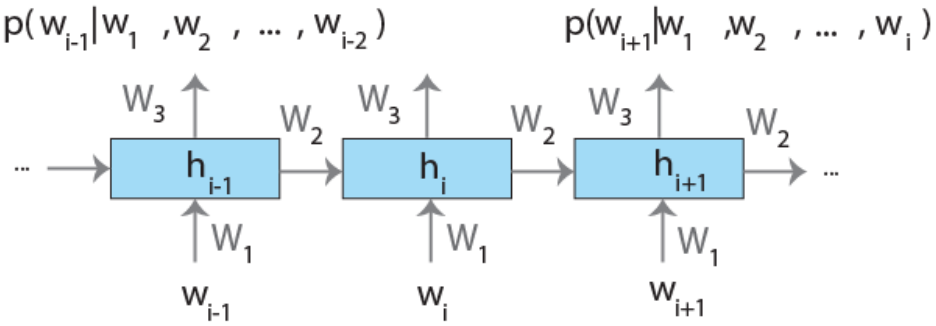
1. Olgu korpuses N sõna $w_1, ..., w_N$, mis on esitatud 1-S kodeeringus.

2. Transformeeri sõnad uude esitusse $K \times S$-mõõtmelise maatriksi W_1 kaudu:

$$v_1 = W_1 w_1, v_2 = W_1 w_2, ..., v_N = W_1 w_N$$



Joonis 1.5 Võrreldes CBOW mudeliga, võtab edasileviga neurovõrgul põhinev keelemudel arvesse ka sõnade järjekorda ning sisaldab mittelineaarset peidetud kihti.



Joonis 1.6 Rekurrentsel neurovõrgul põhinev keelemudel erineb eelnevatest mudelitest selle poolest, et ei kasuta fikseeritud suurusega konteksti.

3. Ajahetkel t arvuta peidetud kihi tunnused kasutades $K \times K$-mõõtmelist maatriksit W_2 järgmiselt:

$$h_t = f(v_t + W_2 h_{t-1}),$$

kus f on mingisugune mittelineaarne funktsioon (näiteks sigmoidfunktsioon või $tanh$).

4. Iga ajahetke t korral leia skoorivektor

$$z_t = W_3 h_t,$$

kus W_3 on $S \times K$-mõtmeline maatriks.

5. Muuda skoorivektor z tõenäosusteks $p(w_i|w_1, w_2, ..., w_{i-1})$ näiteks kasutades multinomiaalset logistilist funktsiooni.

Tundmatud on maatriksid W_1, W_2, W_3. Sõnade vektoresitusena kasutatakse W_1 veerge. Jällegi, kuigi keerukam mudel on parem keelemudelina, siis sõnadele õpitud vektoresitused pole tingimata paremad lihtsama mudeliga saadud vektoresitustest [39].

1.3 Eksperimendid

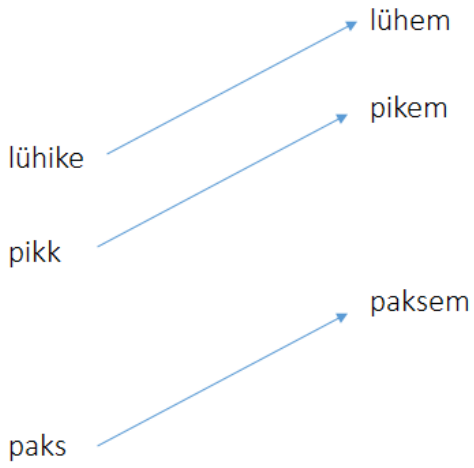
Õppisin word2vec *skip-gram* ja SVD meetodiga 200-mõõtmelise vektoresituse eestikeelsete sõnade jaoks. Rekurrentsel ja klassikalisel neurovõrgul põhinevate keelemudelite käigus õpitavad vektoresitused jätsin võrdlusest kõrvale nende ajalise keerukuse [39] ja halvemate tulemuste pärast ingliskeelsete andmete korral [39]. Lisaks ei ole ma otseselt huvitatud võimalikult heast keelemudelist, vaid sõnavektoritest, mida on kerge konkreetsema ülesande jaoks kohandada. Korpusena kasutasin järgmiste allkorpuste ühendit:

- ajaleht "Postimees"(27.11.1995 - 10.10.2000, 32,9 miljonit sõna);
- ajaleht "Eesti Ekspress"(09.08.1996 - 29.11.2001, 7,2 miljonit sõna);
- ajaleht "Eesti Päevaleht"(18.10.1995 - 31.10.2007, 87,9 miljonit sõna);
- ajaleht "Maaleht"(2001-2004, 4,3 miljonit sõna):
- ajaleht "SL Õhtuleht"(1997-2007, 45.5 miljonit sõna).

Võrdlen neid vektoresitusi analoogiaülesande abil ja visualiseerin õpituid vektoresitusi (1.3.1). Lisaks vaatan, kuidas sõnade vektoresitusi kasutada sõnade tõlkimiseks (1.3.2) ning kuidas võõrkeelsetest vektoresitustest kanda üle tarkust eestikeelsetele sõnavektoritele (1.3.3).

1.3.1 Analoogiaülesanne

Analoogiaülesanne põhineb ideel, et sõna *pikk* erineb sõnast *pikem* samamoodi nagu sõna *lühike* erineb sõnast *lühem* (joonis 1.7). Seda sama seost saab testida küsimusega: "Milline sõna erineb sõnast *lühike* samamoodi nagu sõna *pikem* erineb sõnast *pikk*?". Sellisele küsimusele saab vastata tehes algebralisi tehteid sõnavektoritega. Soovides leida sõna, mis erineb sõnast *lühike* sarnaselt sellega, kuidas erineb sõna *pikem* sõnast *pikk*, võime arvutada vektori $x = v_{\text{pikem}} - v_{\text{pikk}} + v_{\text{lühike}}$ ning saadud vektorile leida lähima vektori (näiteks kasutades koosinuse sarnasust).



Joonis 1.7 Joonisel on näidatud kolme sõnapaari vekoresituste erinevused, mis illustreerivad omadussõna keskvõrde seost.

Seega testides analoogiaseost sõnapaari $a : b$ ning $c : d$ vahel, kus sõna d on teadmata, leian kõigepealt sõnavektorid v_a, v_b, v_c ja arvutan $x = v_b - v_a + v_c$. Saadud sõnavektorit x eeldan olevat parima vastusena analoogiaküsimusele. Kuna täpselt x asukohas ei pruugi olla ühtegi sõna, otsin vektorile x kõige lähimat sõnavektorit x^*. Selleks kasutan koosinuse sarnasust:

$$x^* = v_{\text{argmax}_i \frac{v_i x}{\|v_i\|\|x\|}}.$$

Vastava analoogiaülesande pakkus välja Mikolov et al. [39] [41]. Neist artiklitest võtan eeskuju eestikeelse analoogiaküsimuste andmestiku koostamisel. Andmestikus on neli semantilist kategooriat: rahvus, pealinn, rahaühik ja sugu. Näiteks alamgrupis sugu testitakse seost: sõna *kuningas* on sõnale ?, nagu sõna *mees* on sõnale *naine*. Õigeks vastuseks

loen sõna *kuninganna*. Grammatilistest seostest testin nimisõnade käänamist ja mitmust, tegusõnade pööramist ja omadussõnade keskvõrret. Testin kõiki nimisõnade käändeid. Pööramisel testin kindla kõneviisi olevikku ja lihtminevikku, ainsust ja mitmust, 1., 2. ja 3. isikut. Analoogiaküsimuste näiteid olen toonud tabelis 1.1.

Analoogiaandmestiku koostamisel kasutasin Eesti kirjakeele sagedussõnastikku [16], et valida välja 100 kõige sagedasemat ajalehetekstides leiduvat nimisõna, tegusõna ja omadussõna. Seejärel kasutasin teeki *estnltk*[2], et etteantud sõnalemmast ja morfoloogilisest kategooriast lähtuvalt genereerida uusi sõnavorme (st käändeid ja pöördeid). Semantilised sõnadepaarid valisin välja käsitsi. Iga sõnapaari jaoks valisin juhuslikult välja viis teist sõnapaari, mida kontrollitakse analoogiaülesandel. Andmestikku on võimalik allalaadida veebilehelt www.stat24.ee/est_word_analogy.txt.

Tabel 1.1 Sõnapaaride näited analoogiaandmestikus

Analoogia tüüp	Sõnade paar 1		Sõnade paar 2	
Rahvus	Eesti	eestlane	Gruusia	grusiin
Pealinn	Eesti	Tallinn	Läti	Riia
Rahaühik	Šveits	frank	Jaapan	jeen
Sugu	mees	naine	kuningas	kuninganna
Keskvõrre	hea	parem	pikk	pikem
Nimisõna mitmus	pliiats	pliiatsid	käsi	käed
Nimisõna - omastav	pliiats	pliiatsi	käsi	käe
Nimisõna - osastav	pliiats	pliiatsit	käsi	kätt
...	...	...	...	...
Nimisõna - kaasaütlev	pliiats	pliiatsiga	käsi	käega
Tegusõna - olevik 1. isik ainsus	tantsima	tantsin	laulma	laulan
Tegusõna - olevik 2. isik ainsus	tantsima	tantsid	laulma	laulad
Tegusõna - olevik 3. isik ainsus	tantsima	tantsib	laulma	laulab
...	...	...	...	...
Tegusõna - lihtminevik 2. isik mitmus	tantsima	tantsisite	laulma	laulsite
Tegusõna - lihtminevik 3. isik mitmus	tantsima	tantsisid	laulma	laulsid

Tabelis 1.2 olen toonud analoogiaülesande tulemused. Mudelite täpsuse kategooriate lõikes olen toonud tabelis A.1. Skip-gram mudeli tulemused on paremad võrreldes SVD-l põhinevate mudelite tulemustega. Eriti selgelt on skip-gram mudeli paremus näha süntaktiliste analoogiaküsimuste tulemustes (vastavalt 53.6% ja 5.6%). Samuti on selgelt näha, et SVD-l põhinevad vektoresitused on märgatavalt paremad, kui sõnade koosesinemise maatriksit eelnevalt kaaluda (semantiline täpsus vastavalt 5.4% ja 36.7%).

[2] http://estnltk.github.io/estnltk/

Tabel 1.2 Mudelite tulemused analoogiaandmestikul.

Mudel	Täpsus	
	Semantiline	Süntaktiline
SVD	5.4	1.0
SVD-PPMI	36.7	5.6
skip-gram	48.5	53.6

Kvalitatiivne analüüs

Analoogiaülesandel andis *word2vec* skip-gram mudel parimaid tulemusi. Et saada paremat ettekujutust leitud vektoresitustest, vähendan vektoresituste mõõtmelisust ja visualiseerin neid kahemõõtmelises ruumis.

Olen näidanud 1500 sagedasema sõna vektoresituste projektsiooni kahemõõtmelisse ruumi ja selle lähivaateid (joonis 1.8). On näha, et sarnased sõnad asuvad ruumis lähestikku (näiteks riigid, nädalapäevad, nimed, arvud). See illustreerib, et sarnastel sõnadel on tõesti sarnane vektoresitus.

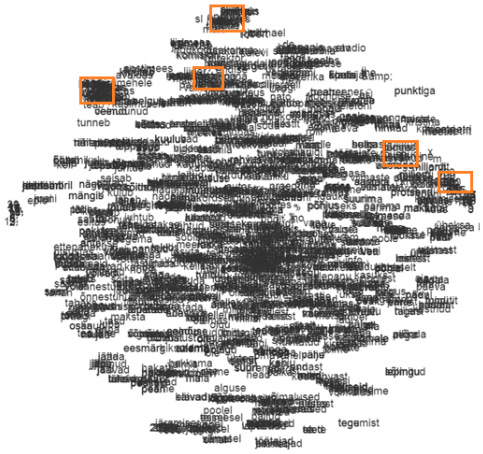
Joonisel 1.9 on näidatud osade omadussõnade vektoresituste kahemõõtmeline PCA projektsioon. See joonis illustreerib, et mudel on võimeline automaatselt leidma sõnade vahel seoseid (tuletan meelde, et mudelile ei anta ette reegleid, mida näiteks keskvõrre tähendab).

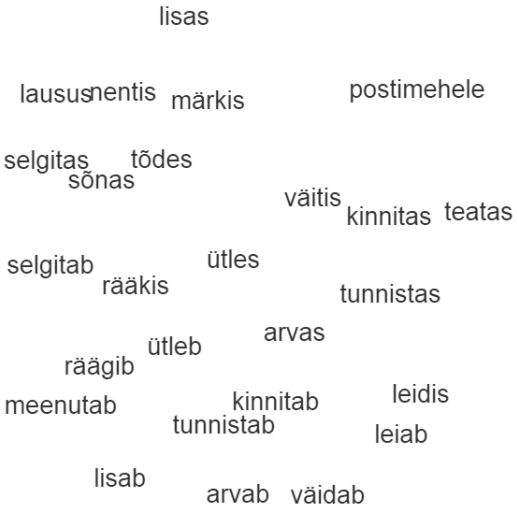
Sarnased seosed nimisõnade ja tegusõnade kõigi testitud kategooriate lõikes on näidatud vastavalt joonisel A.1 ja A.2. Mudeli tulemustetabelis A.1 oli näha, et tegusõnade analoogiaküsimuste täpsus oli parem kui nimisõna käänete analoogiaküsimuste täpsus. See paistab hästi välja ka joonistel A.1 ja A.2, kus on näha, et tegusõnade korral on paralleelseid jooni rohkem.

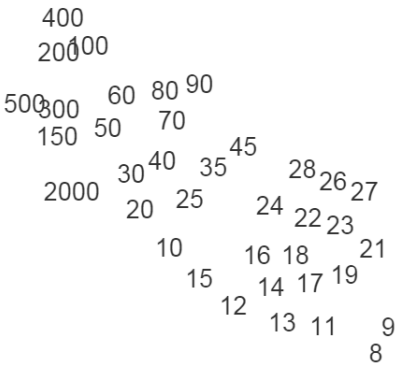
1.3.2 Sõnade tõlkimine

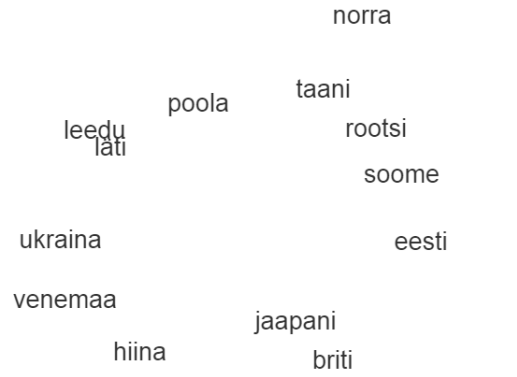
Tüüpilised statistilise masintõlke meetodid põhinevad sõnastikel ja fraasitabelitel. Selles sektsioonis uurin, kuidas sõnavektorite abil automatiseerida sõnastike moodustamist.

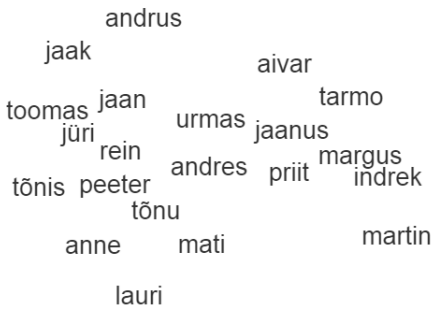
Joonisel 1.10 on näidatud eestikeelsete sõnade *null, üks, kaks, kolm, neli* ja ingliskeelsete vastete *zero, one, two, three, four* peakomponentanalüüsi abil saadud sõnavektorite projektsioonid kahemõõtmelisse ruumi. Jooniselt on näha, et geomeetriline struktuur on mõlemas keeleruumis sarnane (erineb peamiselt skaleerimise ja pööramise võrra) ja võiksime saada keelteruumide vahelist seost kirjeldada lineaarse projektsiooni abil. Seega teades sõnade *null, üks, kaks, kolm, neli* ingliskeelseid tõlkeid saame leida projektsioonimaatriksi, mille abil saab tõlkida teisigi numbreid inglise keelde.



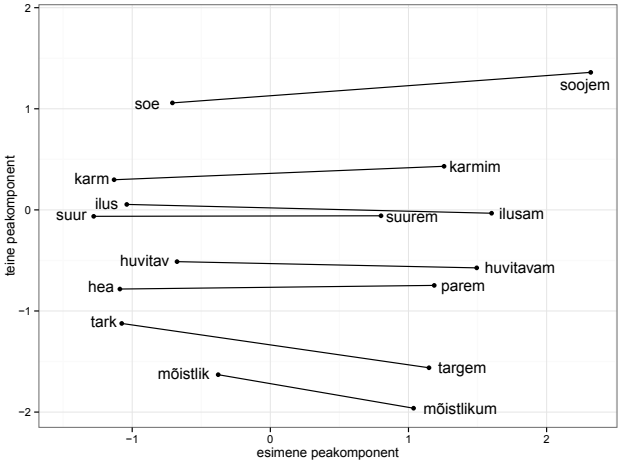




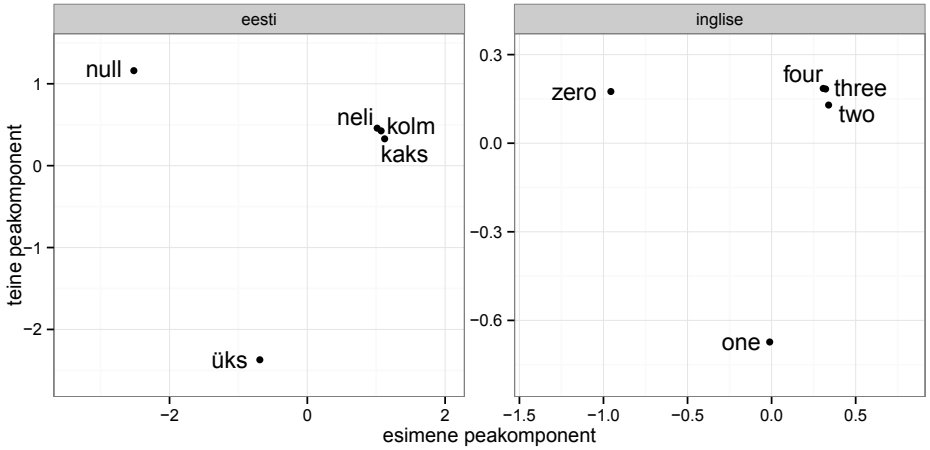




Joonis 1.8 1500 sagedasema sõna skip-gram vektoresituse porjektsioon kahemõõtmelisse ruumi ja lähivaated sellest.



Joonis 1.9 Omadussõnade PCA projektsioon *skip-gram* mudeli sõnavektoritest. Mudel on automaatselt tuvastanud keskvõrde kontseptsiooni.



Joonis 1.10 PCA projektsioonid skip-gram eesti- ja ingliskeelsetest vektoritest. Sõnadevaheline struktuur mõlemas keeleruumis on sarnane.

Olgu meil antud n sõnapaari ja neile vastavad vektoresitused $\{v_i, v_i^*\}_{i=1}^n$, kus $v_i \in \mathbb{R}^{d_1}$ on i-nda sisendsõna vektoresitus ja $v_i^* \in \mathbb{R}^{d_2}$ on sisendsõna tõlke vektoresitus. Tahame leida $d_2 \times d_1$-mõõtmelist maatriksit W nii, et Wv_i lähendaks hästi tõlget v_i^*. Selleks minimiseerime ruutude summat

$$\min_W \sum_{i=1}^{n} \|Wv_i - v_i^*\|^2$$

Kui soovime tõlkida uut sõna, siis projitseerime selle sõna vektoresituse v teise keele vektorruumi arvutades $v^* = Wv$. Seejärel tagastame tõlkeks sõna, mille vektoresitus on kõige lähedasem v-le (selleks võime kasutada koosinuse sarnastust).

Eelkirjeldatud meetodi sõnade tõlkimiseks pakuti välja artiklis [40]. Uurisin selle meetodi headust eestikeelsete sõnade tõlkimisel ingliskeelseteks ja vastupidi. Kasutasin 4000 sagedasemat eestikeelset lemmat ja nende tõlget (saadud *Google Translate* abil), et leida projektsioonimaatriks W. Sageduselt järgmise 1000 sõnade peal hindasin tõlkimise täpsust.

Täpsust hindasin kahel viisil. Esiteks kontrollisin, kas kõige lähim sõna on õige tõlge (P1). Selline viis aga alahindab meetodi täpsust, sest näiteks sünonüümid loetakse valeks tõlkeks. Seega kontrollisin lisaks, kas õige tõlge on viie kõige lähima sõna hulgas (P5). Tabelis 1.3 on näidatud tulemused.

Tabel 1.3 Sõnade tõlkimise täpsus.

	P1	**P5**
Est-Eng	11.8	18.3
Eng-Est	9.4	19.7

Eesti keelest inglise keelde tõlkimise täpsus on 11.8%. Kui uurida, kas õige tõlge on viie lähima sõna hulgas, sain täpsuseks 18.3%. Vastavad täpsused inglise keelest eesti keelde tõlkimisel on 9.4% ja 19.7%.

Sama metoodikat kasutades on saadud hispaania keelest inglise keelde tõlkimisel vastavateks täpsusteks 35% ja 52%.[40] Need täpsused on paremad kui saadud eesti keele eksperimentide tulemused, aga samuti on lingvistiliselt hispaania keel võrreldes eesti keelega inglise keelele sarnasem.

1.3.3 Keeltevaheline siirdeõpe

Ingliskeelseid sõnavektoreid treenitakse korpustel, kus on 100 miljardit sõna [39]. Suurimad korpused eesti keele jaoks sisaldavad aga kõigest 250 miljonit sõna. Suuremad andmemahud muudavad praktikas tihti rakenduste tulemusi paremaks [13], kuid on võimatu, et lähiajal

skaleeruks eestikeelne korpus, säilitades kvaliteedi, 100 miljardi sõna lähedale (võttes arvesse 10 aasta meediaväljaannete teksti hulka, kuluks selleks umbes 4000 aastat). Aga ehk on võimalik suurte andmemahtude peal treenitud ingliskeelsetes sõnavektorites olevat semantilist infot üle kanda eestikeelsetele vektoritele? Semantiline info (näiteks pealinnad, rahaühikud jne) ei sõltu keelest.

Üheks võimaluseks semantilise info kättesaamiseks võõrkeelsetest sõnavektoritest, on kasutada kanoonilist korrelatsioonianalüüsi (CCA) ning projitseerida sõnade inglis- ja eestikeelsed vektoripaarid uude vektorruumi.

Olgu maatriksid $V \in \mathbb{R}^{n \times d_1}, V^* \in \mathbb{R}^{n \times d_2}$ vastavalt eestikeelsete ja ingliskeelsete sõnade esitused. See tähendab, et maatriksi V i-s rida v_i tähistab eestikeelset vektoresitust ning vastava sõna tõlke esitus v_i^* on maatriksi V^* i-ndas reas. CCA leiab projektsioonimaatriksid $U \in \mathbb{R}^{d \times d_1}$ ja $W \in \mathbb{R}^{d \times d_2}$, kus $d = min\{d_1, d_2\}$. Maatriksite U ja W esimene suund u_1 (ehk maatriksi U esimene rida) ning w_1 (ehk maatriks W esimene rida) leitakse selliselt, et kanooniliste tunnuste Vu_1 ja V^*w_1 vaheline korrelatsioon $\rho(Vu_1, V^*w_1)$ oleks maksimaalne. Teine suund (vastavalt u_2 ja w_2) leitakse analoogiliselt, maksimiseerides Vu_2 ja V^*w_2 vahelist korrelatsiooni $\rho(Vu_2, V^*w_2)$, kuid tingimusel, et korrelatsioon eelmiste kanooniliste tunnustega oleks 0 (st $\rho(Vu_1, Vu_2) = \rho(V^*w_1, V^*w_2) = \rho(Vu_1, V^*w_2) = (Vu_2, V^*w_1) = 0$). Seda protsessi korratakse d korda.

Ingliskeelsete sõnavektoritena kasutan 300-dimensionaalseid vektoreid, mis on eeltreenitud *Google News* andmestikul (100 miljardit sõna)[3]. Eestikeelsete sõnavektoritena kasutan 300-dimensionaalseid skip-gram vektoreid. CCA-d rakendan 5000-le sõnale (sõnadeks on valitud 5000 sagedasemat eestikeelset sõna ja nende ingliskeelsed vasted). CCA tulemusena saan projektsioonimaatriksid U ja W. Kasutan maatriksit U, et transformeerida eesikeelsed sõnavektorid uude ruumi. Uusi vektoresitusi testin analoogiaülesandel.

Tabel 1.4 Siirdeõppe tulemused analoogiaülesandel semantiliste kategooriate lõikes.

Analoogia tüüp	**skip-gram**	**skip-gram + siire**
Rahvus	75% (36/48)	79% (38/48)
Pealinn	32% (21/65)	37% (24/65)
Rahaühik	9% (1/11)	0% (0/11)
Sugu	74% (31/42)	76% (32/42)

Siirdeõppe abil leitud vektoresitused on õigesti arvanud mõne analoogia rohkem kategoorites rahvus, pealinn ja sugu (tabel 1.4). Kuigi erinevused pole suured, näitavad tulemused, et võõrkeelsetest vektoritest on võimalik semantilist informatsiooni üle kanda.

[3] https://code.google.com/p/word2vec/

Peatükk 2

Lausetest arusaamine

Kas *statistiline masin* on võimeline lausetest aru saama, kui ta teab sõnade tähendust ja struktuuri? Lausete mõistmise all võib käsitleda oskust teksti kategoriseerida, lühendada, meelsust tuvastada või tõlkida. Selle peatüki läbiva näitena uurin *statistilise masina* oskust teksti meelsust tuvastada.

2.1 Lausete esitamise viisid statistilistes algoritmides

Laused koosnevad sõnadest ning nende esitusi saadakse tüüpiliselt sõnade esitusi kombineerides. Lausete esitamise teeb keeruliseks aga see, et laused on erineva pikkusega. Statistilistes algoritmides on tüüpiliselt aga vaja just fikseeritud pikkusega esitusi. Muuhulgas on lauseid väga palju ja erinevaid. Suures tekstikorpuses on harva ühte lauset mitu korda.

Nende probleemide lahendamiseks tehakse sageli lihtsustav eeldus, et sõnade järjekord lauses ei oma tähtsust ning piisav on sõnade esiemiste arv (2.1.1). See on aga liialt naiivne eeldus. Seega uurin ka lausete saamise viise, kus mudeli abil õpitakse automaatselt lausele sobiv esitus, mis arvestab ka sõnade järjekorda (2.1.2).

2.1.1 Sageduspõhised meetodid

Tüüpiliselt esitatakse lauseid sõnade või sõna mitmikute sagedusloendina (vastavalt *bag-of-words* ja *bag-of-ngrams* esitus). Näiteks olgu meil kaks lauset:

- L1 = Elu on lill, isegi pärast magistriõpinguid!
- L2 = Tudengi elu on lill, aga töömesilase elu?

Lausete põhjal koostatakse sõnastik. Ignoreerides suurt ja väikest algustähte, saame sõnastikuks: [1 - elu; 2 - on; 3 - lill; 4 - isegi; 5 - pärast; 6 - magistriõpinguid; 7 -tudengi; 8 - aga; 9 - töömesilase; 10 - ,; 11 - !; 12 - ?].

Sõnastikus on 12 erinevat sõna ning iga lauset kirjeldatakse 12-elemendilise vektorina. Vektori iga element näitab, mitu korda esines vastav sõna etteantud lauses. Näitelausete vektoresituseks saame:

v_{L_1} = [1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0]

v_{L_2} = [2, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1]

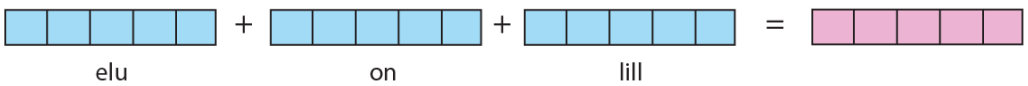
Sageli rakendatakse tekstile ka muud eeltöötlust peale teksti väiksetäheliseks tegemise. Näiteks lemmatiseerimist ehk sõna algvormi leidmist (näiteks sõna *tudengi* lemmatiseerimisel on tulemuseks *tudeng*). See on kasulik vastava esituse mõõtmelisuse vähendamiseks.

Sõna paaride ehk bigrammide esituse korral loetakse kokku tekstis olnud sõnapaaride sagedused. Näiteks mitu korda esines lauses sõnapaar *elu on*, mitu korda *on lill* jne.

Neil esitustel on mitmeid puudusi. Sõnade sagedusloendi esitus eirab sõnade järjekorda ning seega võib erinevatel lausetel olla täpselt sama esitus (kui kasutatakse samu sõnu). Kuigi sõnade mitmikud arvestavad sõnade järjekorda lühikeses kontekstis, siis on vastaval esitusel kõrge dimensionaalsus ning esitus on hõre (st vektoresituses on vähe nullist erinevaid elemente). Lisaks puudub lausetel semantiline tähendus. Näiteks lause *"Koer näris konti"* on võrdsel kaugusel nii lausest *"Kass sõi kala"* kui ka lausest *"Suurim tänaseks loodud sünteetiline teemant kaalub kolm karaati"*.

Sõnavektorite aritmeetiline kombinatsioon

Inspireerituna sõnavektorite distributiivsemantika mudelitest, on mitmed teadlased proovinud hästi töötavaid sõnavektorite mudeleid laiendada lausetele. Kõige lihtsam viis selleks on võtta kõigi lauses olnud sõnade vektoresituste aritmeetiline keskmine.



Joonis 2.1 Lauset võib esitada lauses olevate sõnade vektoresituste aritmeetilise keskmisena.

Olgu meil lause, milles on n sõna. Lauses olevate sõnade vektoresitused olgu $v_1, \ldots, v_n$. Lauset esitame sõnavektorite aritmeetilise keskmisena $\frac{1}{n}\sum_{i=1}^{n} v_i$. Sõnu ei pea ilmtingimata

võrdselt kaaluma, vaid *informatiivsematele* sõnadele võib anda suurema kaalu ja vähemütlevatele sõnadele (näiteks sidesõnadele) väiksema kaalu.

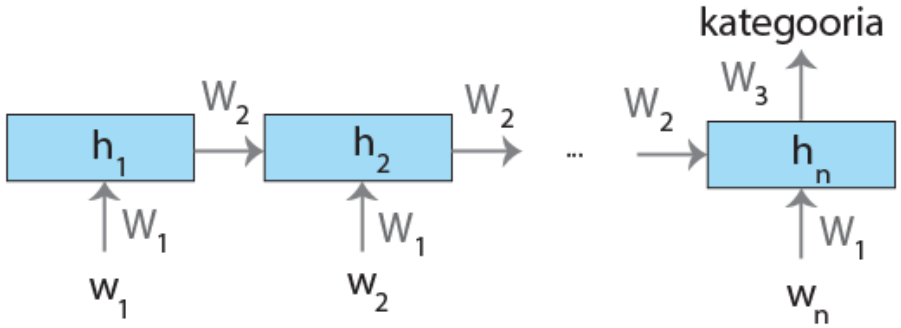
Eelkirjeldatud lause esitus eirab samuti sõnade järjestust, kuid omab rohkem semantilist informatsiooni kui sõnade sagedusloendi esitus.

2.1.2 Mudelipõhised meetodid

Selle asemel, et heuristiliselt sõnavektorite esitusi kombineerida, võime seda lasta teha mudelil. Näiteks saame õppida sellised lauseesitused, mis on kasulikud klassifitseerimisülesandes.

Rekurrentne neurovõrk

Esimeses peatükis tutvustasin, kuidas rekurrentset neurovõrku (RNN) kasutada sõnavektorite leidmiseks (1.2.4). RNNil oli hea omadus, et ta polnud piiratud fikseeritud pikkusega konteksti kasutamisega. Seega on RNNiga mugav modelleerida erineva pikkusega jadasid (näiteks lauseid).



Joonis 2.2 Rekurrentse neurovõrgu viimast kihti h_n võib käsitleda lause esitusena.

Rekurrentne neurovõrk koosneb sisendkihist, rekurrentset peidetud kihist ja väljundkihist. Iga uue sõnaga w_i uuendatakse peidetud kihti h_i, kusjuures h_i hoiab endas informatsiooni ka varasemate sõnade kohta. Lause esitusena võib käsitleda rekurrentse neurovõrgu viimast peidetud kihti h_n (kus n tähistab lause pikkust). Kuigi sellise lause esituse korral on viimastel sõnadel suurem mõju, siis on sellist lause esitust edukalt kasutatud masintõlkes [49].

Konvolutsiooniline neurovõrk

Konvolutsioonilised neurovõrgud töötati algselt välja küll piltide klassifitseerimiseks, kuid neid on edukalt kasutatud ka tekstide sildistamisel [19] [22]. Neurovõrgu sisendiks on

lauses olevate sõnade vektoresitustest moodustatud maatriks, mille põhjal õpitakse lausele uus esitus, mis on kasulik lause klassifitseerimiseks (joonis 2.3).

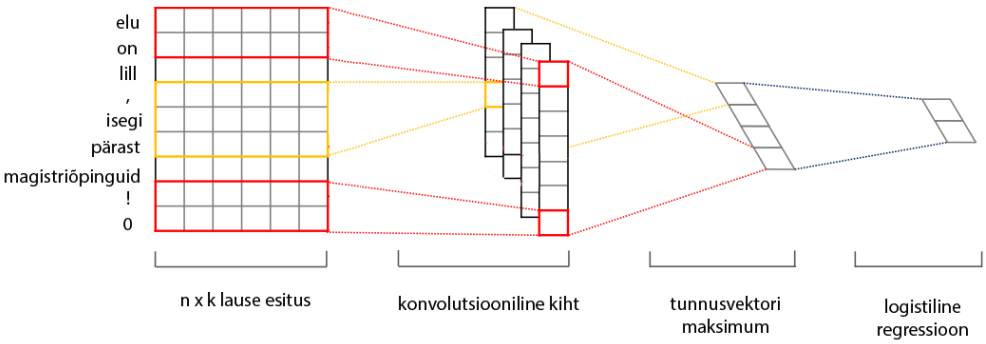
Täpsemalt, olgu lauses n sõna ja $v_i \in \mathbb{R}^k$ olgu i-nda sõna k-mõõtmeline esitus. Tähistame lauset $n \times k$-mõõtmelise maatriksiga $V_{1:n}$, kus i-s rida tähistab i-nda sõna vektoresitust. Lause alamosa, i-ndast sõnast $i+j$-nda sõnani, tähistame $j \times k$ maatriksiga $V_{i:i+j}$.

Konvolutsioonilises kihis leitakse üks tunnus järgnevalt:

1. Filtrit $w \in \mathbb{R}^{h \times k}$ rakendatakse igale aknale, kus on h sõna, st osalausetele $\{V_{1:h}, V_{2:h+1}, ..., V_{n-h+1:n}\}$ ning saadakse $n-h+1$-elemendiline tunnusvektor c.

2. Näiteks $c_i = f(W * V_{i:i+h-1})$, kus f on mittelineaarne funktsioon (näiteks sigmoidi-funktsioon) ja $A * B$ tähistab maatriksite A ja B elemendiviisilist korrutamist ja kõigi liikmete summeerimist. Vajadusel võib lisada vabaliikme.

3. Vektorist c tagastame maksimumi $\max\{c\}$ (*max-over-time pooling*). See aitab meil töötada suvalise pikkusega lausega (vastasel korral oleks pikematel lausetel tunnuseid rohkem).



Joonis 2.3 Konvolutsioonilise neurovõrgu arhitektuur lausete klassifitseerimiseks.

Eelnev kirjeldus näitas, kuidas leida ühte tunnust. Tavaliselt leitakse mitu tunnust ning kasutatakse erinevaid filtrite suuruseid h. Vastavad tunnused ongi lause uueks esituseks. See esitus õpitakse olevat selline, mis on kasulik näiteks klassifitseerimiseks. Et saada erinevatesse klassidesse kuulumise tõenäosusi, kasutatakse multinomiaalset logistilist funktsiooni. Ehk kui õpitakse m erinevat filtrit, saadakse lause uueks esituseks m-elemendile vektor z, kust j-ndasse klassi kuulumise tõenäosused on $\frac{e^{z^T w_j}}{\sum_{l=1}^{L} e^{z^T w_j}}$, kus L on erinevate klasside arv ja w_j on vastava klassiga seotud kaalud.

2.2 Eksperimendid

Võrdlen lausete vektoresituste meetodeid lausete meelsuse tuvastamise ülesandel (2.2.1). Teen katseid, et selgitada välja, kas sõnavektorite algväärtustamine on oluline (2.2.2). Muuhulgas visualiseerin saadud vektoresitusi (2.2.3). Lisaks uurin, kui hästi on võimalik lausete meelsust tuvastada, kui vajalikku siltidega andmekogu pole eesti keele jaoks olemas, kuid kättesaadav on sildistatud ingliskeelne korpus (2.2.4).

2.2.1 Lause meelsuse klassifitseerimine

Lausete vektoresituse võrdlemiseks kasutan Eesti emotsionaalse kõne korpust, et hinnata lausete meelsust (vastavalt viha või rõõm). Korpuses on 306 lauset, mille meelsuseks on viha ning 271 lauset, mille meelsuseks on rõõm (näitelaused on tabelis 2.1). Eesti emotsionaalse kõne korpuse eesmärgiks on olla usaldusväärne andmekogu nii kõnes kui ka kirjas avalduvate emotsioonide uurimiseks [1].

Tabel 2.1 Näitelaused Eesti emotsionaalse kõne korpusest, mille meelsuseks on rõõm või viha.

Kategooria	lause
rõõm	Me usume oma tulevikku.
rõõm	Aga tegelikult on Eesti mees ju tubli.
rõõm	Kui ma Otti esimest korda nägin , siis mulle tundus kohe , et selle inimesega ma abiellun .
rõõm	Mõelgu-mõelgu !
rõõm	Mu süda on nii rahul, kui üldse võib olla.
viha	Kolm-neli aastat tagasi ei huvitanud kedagi , kust raha tuleb .
viha	Kui astun kellelegi varbale , vabandab tema , et jäi mulle ette .
viha	Avalikkusele aeti sellist möga , et oli vastik kuulata .
viha	Ma peseks nõud parema meelega hommikul , aga mees õiendab kogu aeg mu kallal , et olen lohakas ja laisk .
viha	See korrumpeerunud punt ei hakka iialgi seda tegema .

Täpsemalt olen võrrelnud järgmisi mudeleid:

- lemmade sagedusloend, millele on sobitud logistilise regressiooni mudel;
- lemmade ja lemma paaride sagedusloend, millele on sobitud logistilise regressiooni mudel;
- sõnavektorite aritmeetiline kombinatsioon, millele on sobitatud logistilise regressiooni mudel;

- konvolutsioonilisel neurovõrgul põhinev mudel.

Kuna sagedusloendil põhinevate esituste mõõtmelisus on suur ja andmeid on suhteliselt vähe, kasutasin kõigi mudelite juures ka L2 regulariseerimist.

Sõnavektorite esitused on õpitud eelmises peatükis kirjeldatud *word2vec* skip-gram mudeliga ajalehetekstide peal. Konvolutsioonilisel neurovõrgul põhineva mudeli sõnaesitused olen initsialiseeritud *word2vec* vektoritega ja need fikseerinud (st mudeli treenimise käigus need ei muutu). Selle mudeli korral õppisin 50 filitrit nii suurusega $h = 3$, $h = 4$ ja $h = 5$ (seega kokku 150 tunnust). Rekurrentse neurovõrgu jätan siinsest analüüsist välja (seda mudelit uurin järgmises peatükis lausete tõlkimise kontekstis).

Kuna emotsionaalsuse andmestik on suhteliselt väike (kokku 577 lauset), siis kasutan 10-kordset ristvalideerimist, et hinnata mudelite täpsused (tabel 2.2). Peab meeles pidama, et inimesed ei hinda emotsionaalsust alati ühtemoodi ning 100% täpsuse saavutamine pole võimalik.

Tabel 2.2 Meelsuse tuvastamise risvalideerimise tulemused.

Mudel	Keskmine täpsus
lemmade sagedusloend + logistiline regressioon	73.5%
lemmade ja lemma paaride sagedusloend + logistiline regressioon	73.2%
sõnavektorite keskmine + logistiline regressioon	78.2 %
konvolutsiooniline neurovõrk (fikseeritud sõnavektorid)	79.5%

Lemma sagedusloendil põhineva logistilise regressiooni ristvalideerimise keskmine täpsus on 73.5%. Sõnapaaride informatsiooni kasutamine pole tulemust parandanud (täpsus 73.2%). Üllatavalt hea täpsuse annab sõnvektorite keskmistamine (täpsus 78.2%). See on heaks illustratsiooniks siirdeõppe kasulikkusest - 200 miljoni sõnaga korpusest on suudetud tarkust üle kanda ülesandele, kus andmestiku suurus on suhteliselt väike. Konvolutsioonilise neurovõrgu kasutamine parandab tulemust veelgi (täpsus 79.5%). See mudel võttis arvesse sõnade mitmikke. Kuigi sõnapaaride sagedusloendi kasutamine tulemusi ei parandanud, siis konvolutsiooniline võrk oskas ka sõna mitmikes olevat informatsiooni ära kasutada.

2.2.2 Sõnavektorite algväärtustamine

Uurisin edasi, kuidas mõjutab konvolutsioonilise võrgu tulemusi meelsuse tuvastamise ülesandel see, kas sõnavektorid algväärtustada juhuslikult või kasutada algväärtustamisel eelmises peatükis leitud *word2vec* vektoreid (tabel 2.3).

Juhuslike algväärtuste korral tuli 10-kordse ristvalideerimise keskmiseks täpsuseks 72.3%, *word2vec* vektoritega algväärtustamisel oli täpsuseks 79.5%. Parem tulemus *word2vec* vektorite kasutamisel on heaks näiteks siirdeõppe õnnestumisest.

Teisalt ei ole iga ülesande korral suure korpuse peal õpitud sõnade vektoresitused optimaalsed. Näiteks on esimeses peatükis õpitud *word2vec* mudeli korral sõna *hea* vekoresitusele lähimad esitused sõnadel *suurepärane*, *halb*, *ülihea*, *tore*, *superhea*, *vilets*, *kehv*. Seega on lausetel *See on väga hea!* ja *See on väga halb!* sarnane vektoresitus. Meelsuse tuvastamise korral ei ole sellised esitused aga sobilikud (sooviksime, et sõnade *halb* ja *hea* vektoresitused oleksid üksteisest kaugel).

Seega võime küll sõnavektorid algväärtustada *word2vec* tulemustega, aga mudeli treenimise käigus neid muuta. Uurisin selle mõju tulemustele (tabel 2.3). Sõnaesituste fikseerimisel *word2vec* tulemustega, oli konvolutsioonilise neurovõrgu 10-kordse ristvalideerimise täpsuseks 79.5%, lastel sõnaesitustel mudeli treenimise käigus ka muutuda, tuli täpsuseks 80.0%. Seega erinevus antud juhul praktiliselt puudub.

Tabel 2.3 Meelsuse tuvastamise ristvalideerimise tulemused sõnavektorite erinevate algväärtustamiste korral.

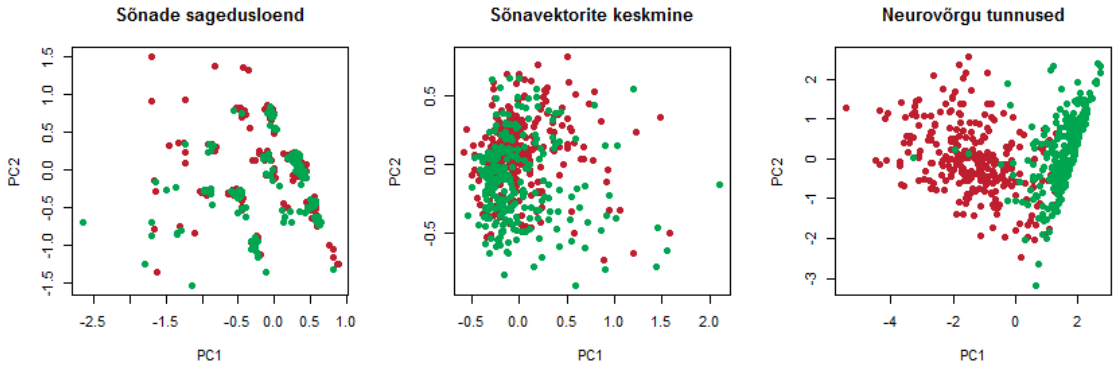
Mudel	Keskmine täpsus
sõnavektorite parameetrite juhuslik algväärtustamine	72.3 %
sõnavektorite parameetrite *word2vec*-iga algväärtustamine ja nende fikseerimine	79.5%
sõnavektorite parameetrite *word2vec*-iga algväärtustamine ja nende vabastamine	80.0%

2.2.3 Kvalitatiivne võrdlus

Võrdlesin sõnade sagedusloendi, sõnavektoritite keskmise ja konvolutsioonilise neurovõrgu õpitud lausete esitusi kvalitatiivselt. Selleks kasutasin peakomponentanalüüsi, et vähendada tunnusmaatriksi mõõtmelisust. Seejärel visualiseerisin kahte esimest peakomponenti (joonis 2.4).

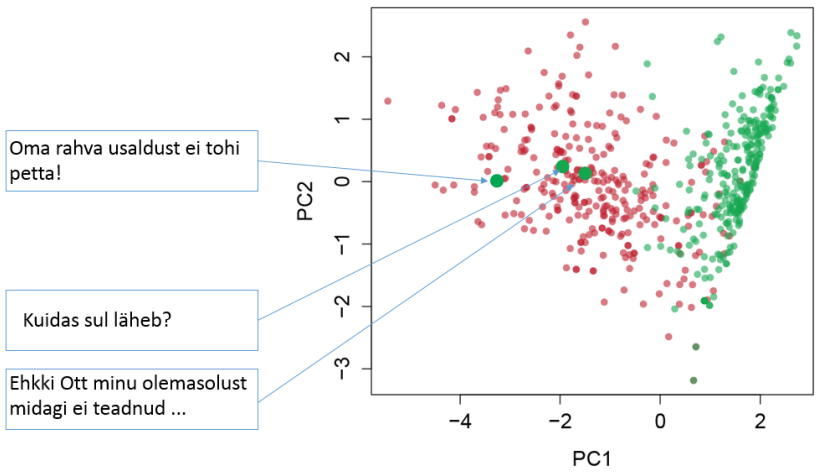
Sõnade sagedusloendi ja sõnavektorite keskmistamisel saadud lausete esituste visualisatsioonil ei eristu viha- ja rõõmulaused. Konvolutsioonilise neurovõrgu õpitud tunnustelt saadud esimese kahe peakomponendi visualisatsioonil eristavad kaks kategooriat selgelt. Seega on konvolutsiooniline võrk õppinud selliseid tunnuseid, mis tulevad kasuks meelsuse eristamisel.

Uurides lähemalt konvolutsioonilise neurovõrgu esimese kahe peakomponendi joonist, on näha, et punasesse punktipilve on eksinud 3 rohelist punkti(joonis 2.5). Nende seas on



Joonis 2.4 Esimese kahe peakomponendi visualisatsioon. Rohelised punktid tähistavad lauseid, mille meelsus on *rõõm*, punased *viha*.

lause *Oma rahva usaldust ei tohi petta!*, mis on andmestikus sildistatud kui *rõõm*, kuid sisult on negatiivne ja sobib tõesti rohkem vihalausete hulka.



Joonis 2.5 Esimese kahe peakomponendi visualisatsioon konvolutsioonilise neurovõrgu õpitud tunnustest. Rohelised punktid tähistavad lauseid, mille meelsuseks on *rõõm*. Kolm rohelist punkti on eksinud punasesse pilve. Nende seas on ka lause *Oma rahva usaldust ei tohi petta!*, mis on andmestikus sildistatud kui *rõõm*, aga mudel arvab olevat meelsusega *viha*.

2.2.4 Keeltevaheline siirdeõpe

Eesti keele korpused on tihti väiksemad kui analoogilised ingliskeelsed korpused. Sageli ei olegi vajalikku eestikeelset andmestikku olemas, kuigi võib leiduda mitu sobivat võõrkeelset andmestikku. Eelmises peatükis uurisin, kuidas kanda ingliskeelsetest sõnavektoritest tarkust üle eestikeelsetele sõnavektoritele kasutades kanoonilist korrelatsioonianalüüsi (1.3.3). Tekkinud inglise-eesti ühisruumi saab kasutada ka lausete klassifitseerimisel. Näiteks kui õppida klassifitseerija ingliskeelsete tekstide peal, saab seda üldistada ka eesti keele peale.

Näitena kasutan ingliskeelset andmestikku, kus on ajalehe artiklite pealkirjad ja nende meelsused [48]. Andmestikus on 250 lauset, mille meelsus on *rõõm* ja 250 lauset, mille meelsus on *viha*. Projitseerin selle andmestikku inglise-eesti sõnavektorite ühisesse ruumi. Projitseerimismaatrksina kasutasin esimeses peatükis kanoonilise korrelatsioonianalüüsi abil leitud maatriksit. Lausete esitusena kasutasin sõnavektorite aritmeetilist keskmist. Neile andmetele sobitasin logistilise regressiooni mudeli, mis prognoosis, kas teksti meelsus on *rõõm* või *viha*.

Seejärel testisin klassifitseerija üldistusvõimet eesti keelse teksti peal. Selleks projitseerisin eesti keelse teksti inglise-eesti ühisruumi. Lause vektoresitusena kasutasin sõnavektorite aritmeetilist keskmist ning eelnevalt leitud logistilise regressiooni mudeli abil prognoosisin meelsust.

Baasmudelina kasutasin *Google Translate*-i, et tõlkida ingliskeelne tekst eestikeelseks. Seejärel kasutasin lause esitusena eestikeelsete sõnaesituste keskmist, millele sobitasin logistilise regressiooni mudeli.

Tabel 2.4 Keeltevahelise siirdeõppe tulemused.

Mudel	Keskmine täpsus
Logistiline regressioon inglise-eesti ühisruumis	61.5 %
Logistiline regressioon tõlgitud sõnadel	64.2 %
Kõige sagedasema klassi ennustamine	50.0 %

Kuigi inglise-eesti vektorruumi põhjal õpitud klassifitseerija tulemused (tabel 2.4) meelsuse tuvastamisel polnud paremad otsetõlke tulemustest (vastavalt 61.5% ja 64.2%), on mitmekeelse ühisruumi õppimine siiski perspektiivikas. Näiteks inglise ühine vektorruum prantsuse ja saksa keelega andis paremaid tulemusi kui naiivne tõlkimine ja seejärel mudeli õppimine [31]. Eesti keele pealgi tuleks katsetada paremaid meetodeid ühise vektorruumi õppimiseks kui kanooniline korrelatsioonianalüüs.

Peatükk 3

Lausete tõlkimine

Olgu meil ingliskeelne lause Eng, mida soovime tõlkida eestikeelseks lauseks Est. See tähendab, et soovime leida just sellise eestikeelse lause, mis maksimiseerib tõenäosuse $p(Est|Eng)$. Kuidas õpetada *statistilist masinat* seda ülesannet lahendama?

Kõigepealt annan sissejuhatuse standardsest lähenemisest statistilisele masintõlkele (3.1). Seejärel kirjeldan, kuidas neurovõrkude abil tõlkida (3.2) ja kuidas automaatselt tõlkimise kvaliteeti hinnata (3.3). Lõpetuseks treenin neurovõrkudel põhineva masintõlkesüsteemi inglise keelest eesti keelde tõlkima (3.4).

3.1 Fraasipõhine statistiline masintõlge

Tüüpiliselt eeldatakse statistilises masintõlkes juhusliku kanali mudelit (*noisy channel model*). Teeseldakse, et eestikeelne lause läks läbi infokanali, millest tuli välja mürane (ingliskeelne) lause. Ülesandeks on tuvastada esialgne sisendlause, mis genereeris mürase (ingliskeelse) lause.

Seega selle asemel, et modelleerida otse $p(Est|Eng)$ ning leida

$$\underset{Est}{\operatorname{argmax}}\, p(Est|Eng),$$

kasutatakse Bayesi valemit, et pöörata ülesanne ümber ning leitakse

$$\underset{Est}{\operatorname{argmax}}\, p(Eng|Est)p(Est),$$

kus tõlkemudel $p(Eng|Est)$ treenitakse paralleeltekstide peal ja keelemudel $p(Est)$ suurel eestikeelsel korpusel. Fikseeritud ingliskeelse lause korral on $p(Eng)$ konstant ning seda pole vaja arvestada.

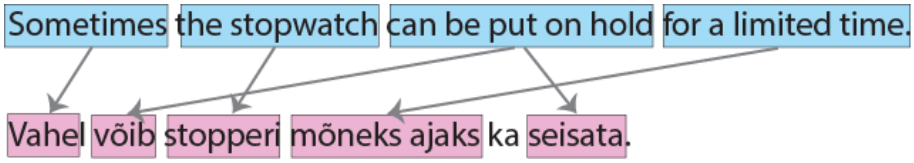
Ülesande ümberpööramiseks on mitu head põhjendust. Esiteks ei pea keelemudeli $p(Est)$ treenimiseks paralleeltekste kasutama, vaid saab kasutada suuri ühekeelseid korpuseid. Teiseks ei pea $p(Eng|Est)$ olema nii täpne kui $p(Est|Eng)$, sest saame kasutada keelemudelit $p(Est)$ grammatiliselt korrektsete lausete valimiseks.

Näiteks oletame, et tõenäosus $p(Eng|Est)$ on suur, kui ingliskeelses lauses Eng olevad sõnad on head tõlked eestikeelses lauses Est olevatele sõnadele ning sõnade järjestus ingliskeelses lauses Eng pole oluline. See ei ole väga hea tõlkemudel, sest fikseeritud ingliskeelse lause *I love bananas* korral saaksid võrdse tõenäosuse laused *Mulle meeldivad banaanid* ja *Banaanid mulle meeldivad*. Keelemudel $P(Est)$ aga annaks esimesele neist suurema tõenäosuse ning kokkuvõttes oleks esimene lause ka parem tõlge.

Eelkirjeldatud statistilise masintõlke mudel vajab kolme komponenti: keelemudelit $p(Est)$, tõlkemudelit $p(Eng|Est)$ ja dekodeerijat, mis tagastab ingliskeelse lause etteandmisel kõige tõenäolisema eestikeelse lause. Keelemudeli $p(Est)$ õppimist kirjeldasin esimeses peatükis.

Tõlkemudeli $p(Eng|Est)$ saamise üheks võimaluseks on tükeldada ingliskeelne lause Eng fraasideks $eng_1, eng_2, ..., eng_n$, eestikeelne lause Est fraasideks $est_1, est_2, ..., est_n$ ning vastavad fraasid joondada (joonis 3.1). Selle põhjal saab koostada fraaside tõlketabeli, kus iga ingliskeelse fraasi eng_i kohta on teada, mitu korda tõlgiti seda mõneks eestikeelseks fraasiks est_j. Tõlketabeli normaliseerimisel saab leida fraasi tõlketõenäosuse $p(eng_i|est_j)$. Sellest saab omakorda leida lause tõlketõenäosuse

$$p(Eng|Est) = \prod_{i=1}^{n} p(eng_i|est_i).$$



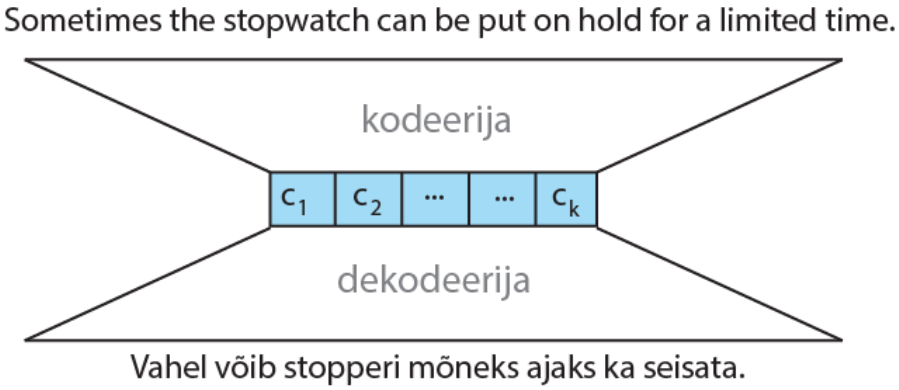
Joonis 3.1 Fraaside tõlketabeli saamiseks tuleb inglise ja eesti keele laused fraasideks tükeldada ning need joondada.

Keele- ja tõlkemudeli olemasolul oskame iga sisend- ja väljundlause korral leida tõenäosust $p(Eng|Est)p(Est)$. Kuid millise lause Est korral on see maksimaalne? Selleks kasutatakse tüüpiliselt algoritmi kiireotsing *beam-search*. Otsingul ei käida läbi kõiki lauseid. See oleks liialt ajakulukas. Tahame käsitleda vaid selliseid lauseid, milles on fraase, mis on võimalikud sisendlause fraaside tõlked. Kõigepealt tõlgitakse üks fraas ning hoitakse meeles k suurima tõenäosusega fraasi. Nende põhjal genereeritakse kahe fraasiga tõlked,

millest jäetakse alles k parimat fraasi. Selliselt jätkatakse kuniks kõik lauses olevad fraasid on tõlgitud.

3.2 Neurovõrkudel põhinev tõlkimine

Eelnevalt nägime, et standardne fraasipõhine masintõlge sisaldab mitut erinevat komponenti (näiteks keelemudel, joondamismudel). Neurovõrke on edukalt kasutatud fraasipõhises tõlkesüsteemis keelemudelina grammatiliselt korrektsete lausete valimisel [46] või lisatunnuste saamiseks tõlketabeli koostamisel [45].



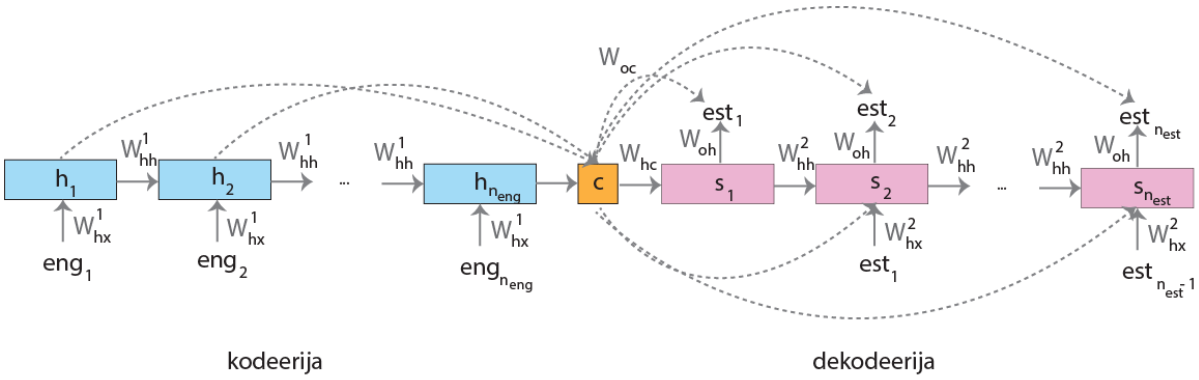
Joonis 3.2 Suvalise pikkusega sisendlause kodeeritakse fikseeritud pikkusega vektoriks. Vastav vektor dekodeeritakse tõlkelauseks.

Hiljuti pakuti välja aga uus neurovõrkudel põhinev lähenemine tõlkimisele, mis on näidanud häid tulemusi nii eraldiseisvate tõlkesüsteemidena kui ka lisakomponentidena fraasipõhisele tõlkimisele [18], [49], [6]. Selle asemel, et häälestada eraldi iga komponenti nagu keelemudel, joondamine, tõlketabel (millede eraldi seadistamine on kaugel optimaalsest), modelleeritakse otse tõenäosust $p(Est|Eng)$. Kõigepealt kodeeritakse suvalise pikkusega sisendkeele lause fikseeritud pikkusega vektoriks (joonis 3.2). Seejärel dekodeeritakse vastav vektor suvalise pikkusega tõlkelauseks. Kodeerija ja dekodeerija treenitakse koos, et maksimiseerida tõese tõlke tõenäosust fikseeritud sisendlause korral. Kui tõlkemudel $p(Est|Eng)$ on õpitud, siis sisendlause Est korral leitakse tõlge Eng otsides lauset, mis maksimiseerib vastava tingliku tõenäosuse.

3.2.1 Kodeerija-dekodeerija süsteem

Kodeerijana ja dekodeerijana võib kasutada rekurrentseid neurovõrke [6]. Üks neist kodeerib sisendlause fikseeritud suurusega vektoriks. Teine rekurrentne neurovõrk dekodeerib vastava

lauseesituse teise keelde.



Joonis 3.3 Rekurrentsetel neurovõrkudel põhineva kodeerija-dekodeerija tõlkesüsteemi arhitektuur.

Kodeerija

Olgu meil sisendlause $Eng = (eng_1, eng_2, ..., eng_{n_{eng}})$, kus eng_i on sisendlause i-nda sõna vektoresitus ning n_{eng} näitab sõnade arvu sisendlauses. Kodeerija, milleks on rekurrentne neurovõrk, loeb sisendlause Eng fikseeritud pikkusega vektoriks $c = q(h_1, ..., h_{n_{eng}})$, kus $h_t = f(eng_t, h_{t-1}) \in \mathbb{R}^n$ on peidetud olek pärast sõna t ning f ja q on mingisugused mittelineaarsed funktsioonid. Näiteks f võib olla sigmoidifunktsioon ning $q(h_1, ..., h_{n_{eng}}) = h_{n_{eng}}$.

Dekodeerija

Dekodeerijaks on teine rekurrentne neurovõrk, mis treenitakse genereerima uut tõlkesõna est_t etteantud lause kontekstivektori c ja eelnevalt genereeritud tõlkesõnade $est_1, ..., est_{t-1}$ korral:

$$p(est_1, ..., est_{n_{est}}) = \prod_{t=1}^{n_{est}} p(est_t | est_1, ..., est_{t-1}, c).$$

Kasutades rekurrentset neurovõrku, modelleeritakse iga tinglikku tõenäosust järgmiselt:

$$p(est_t | est_1, ..., est_{t-1}, c) = g(est_{t-1}, s_t, c),$$

kus g on mingisugune mittelineaarne funktsioon, mis tagastab tõenäosuse ning s_t on vastava neurovõrgu peidetud olek.

Treenimine

Mõlema neurovõrgu parameetrid treenitakse ühiselt, et maksimiseerida järgmist tõepära:

$$\frac{1}{N}\sum_{n=1}^{N}\log p(Est_n|Eng_n),$$

kus N on treeningkorpuse suurus, (Eng_n, Est_n) on paralleelkorpuse vastavalt n-is ingliskeelne ja eestikeelne lause. Parameetriteks on kodeerija neurovõrgu sisendkihti ja peidetud kihti ühendav maatriks W^1_{hx}, rekurrentne maatriks W^1_{hh}. Dekodeerija neurovõrgu vastavad maatriksid on W^2_{hx} ja W^2_{hh}. Lisaks on veel peidetud kihti ja väljundkihti ühendav maatriks W_{ho} ja lause kontekstivektorit c väljundkihiga ühendav maatriksid W_{oc} ja peidetud kihiga ühendav maatriks W_{hc}.

Tõlkimine

Kui mudel on treenitud, saab sisendlauset tõlkida kahte moodi. Üheks võimaluseks on genereerida õpitud mudelist tõlge sõna-sõna haaval. Teiseks mooduseks on otsida tõlget, mis maksimiseerib $p(Est|Eng)$. Selleks võib kasutada näiteks fraasipõhise tõlke sektsioonis kirjeldatud protseduuri kiireotsing (*beam-search*).

Mudeli edasiarendused

Eelnevalt kirjeldatud kodeerija loeb sisendlause Eng sisse alustades sõnast eng_1 ning lõpetades sõnaga $eng_{n_{eng}}$. Selliselt kirjeldab iga peidetud olek h_t eelmisi lauseid, mitte aga järgnevaid. Üheks võimaluseks nii sõna vasak- kui ka parempoolset konteksti arvestada on kasutada kahesuunalist rekurrentset neurovõrku.

Kahesuunaline rekurrentne neurovõrk koosneb kahest rekurrentsest neurovõrgust (edaspidine RNN ja tagurpidine RNN). Edaspidine RNN loeb sisse sisendlause järjekorras $eng_1, ..., eng_{n_{eng}}$ ning leiab peidetud tunnused $h'_1, ..., h'_{n_{eng}}$. Tagurpidine RNN loeb sisendlause sisse tagurpidises järjekorras $eng_{n_{eng}}, ..., eng_1$ ning leiab peidetud tunnused $h''_1, ..., h''_{n_{eng}}$. Sõna t peidetud tunnus h_t saadakse h'_t ja h''_t kokkupanemisel $[h'_t, h''_t]$.

Teiseks standardse rekurrentse neurovõrgu probleemiks on raskus pikki sõltuvusi arvestada. Kuigi peidetud kiht peaks sisaldama informatsiooni kõikidest eelnevatest ajahetkedest

(sõnadest), siis praktikas on keeruline pika jada (lause) korral jada alguse informatsiooni peidetud kihis talletada [43]. Selle lahendamiseks on pakutud välja keerulisemaid peidetud kihi konstruktsioone, nagu LSTM neuronid (*long short-term memory*) [15] või GRU neuronid (*gated recurrent unit*) [7]. Kuna lausete tõlkimisel on pikad sõnadevahelised sõltuvused olulised, siis neurovõrkudel põhinevatel kodeerija-dekodeerija tõlkesüsteemidel kasutatakse-gi vastavaid keerukamaid peidetud kihi neuroneid (artiklis [49] kasutati LSTM neuroneid, artiklis [3] kasutati GRU neuroneid).

3.3 Automaatne tõlkekvaliteedi hindamine

Üks võimalus tõlkimise kvaliteedi hindamiseks on lasta tõlkeeksperdil hinnata kriteeriume nagu teksti ladusus, stiil ja loomulikkus. Inimeste kasutamine tõlkekvaliteedi hindamisel on aga kallis, ajamahukas ja subjektiivne.

Üheks enamkasutatavaks heuristiliseks kriteeriumiks tõlkekvaliteedi automaatseks hindamiseks on näidik BLEU. See arvestab ngrammide arvu, mis kattuvad referentstõlkega. On näidatud, et BLEU tulemused korreleeruvad inimeste poolt antud hinnangutega [42].

BLEU idee seisneb selles, et heal tõlkel on rohkem kattuvusi inimesese antud referentstõlkega (joonis 3.4).

Tõlge 1: Ma olen natuke kahtleval seisukohal selle kompromissi suhtes.

Tõgle 2: Ma ei poolda seda kompromissi.

Referents: Mul on natuke segased tunded selle kompromissi suhtes.

Joonis 3.4 Kaks võimalikku tõlget inglise keelest eesti keelde. Esimene tõlge sisaldab rohkem ühiseid sõnu inimese tehtud referentstõlkega. Seda seost kvantifitseerib näidik BLEU.

BLEU arvestab sõnade, sõnapaaride, kolmikute ja nelikute kattuvust. Näiteks sõnade korral loetakse kokku, mitu sõna masintõlke abil saadud kandidaatlausest esineb referentstõlkes ning see jagatakse kandidaatlauses olevate sõnade arvuga.

Sellisel juhul annaks lause *On on on on on on on on.* joonisel 3.4 oleva näite korral kattuvuseks 100%. Et välistada selliseid patoloogilisi juhtumeid, ei lubata sõna kattuvuseks saada suuremaks referentstõlkes oleva sõna sageduse jagatisega tõlkes olevate sõnade arvuga. Ehk eelmainitud lause korral oleks kattuvuseks $\frac{1}{8}$.

Leidmaks BLEU skoori terve testandmestiku põhjal, leitakse kõigepealt sõna mitmike kattuvuste arv iga lause korral, see summeeritakse ning jagatakse kogu kattuvuste arvuga:

$$p_n = \frac{\sum_{l \in T} \sum_{ngram \in l} \min\{\#ngram, \#_{ref}ngram\}}{\sum_{l' \in T} \sum_{ngram' \in l'} \#ngram'},$$

kus l on lause, T on tõlkelausete kogum, # ja $\#_{ref}$ tähistavad vastavalt ngrammi sagedust tõlkelauses ja referentslauses.

Muuhulgas annab BLEU väiksema skoori lühikestele lausetele. Vastasel korral saaks lause *Mul on* joonisel 3.4 oleva näite korral skooriks 100%. Selle vältimiseks karistatakse lühikesi lauseid arvutades

$$BP = \begin{cases} 1 & \text{kui } c > r \\ e^{1-\frac{r}{c}} & \text{kui } c \leq r \end{cases},$$

kus r on sõnade koguarv kõigis referentstõlgete lausetes ning c on sõnade koguarv kõigis tõlkelausetes. Lõplik BLEU skoor saadakse järgmiselt (aritmeetilise keskmise asemel võetakse geomeetriline keskmine):

$$\text{BLEU} = \text{BP} \times \exp\Big(\frac{1}{N}\sum_{n=1}^{N} \log p_n\Big),$$

kus klassikaliselt on $N = 4$.

3.4 Eksperimendid

Treenin neurovõrkudel põhineva tõlkesüsteemi ingliskeelseid lauseid eesti keelde tõlkima. Inglise ja eesti keele jaoks on mitmeid paralleelkorpuseid, näiteks:

- subtiitrite korpus OPUS OpenSubtitles (4.3 miljonit lauset, 53.4 miljonit ingliskeelset sõna) [51] ;
- Euroopa parlamendi istungite kõnede korpus Europarl (0.7 miljonit lauset, 15.7 miljonit ingliskeelset sõna) [25];
- Euroopa Liidu seadustekstide korpus JRC-Acquis (7.5 miljonit ingliskeelset sõna) [47];
- medikamentidega seotud tekstide korpus EMEA (0.3 miljonit lauset, 8.9 miljonit ingliskeelset sõna) [50].

Kasutatud korpus

Euroopa Liidu seadustekstide korpus JRC-Acquis ja medikamentidega seotud korpus EMEA sisaldavad liiga spetsiifilist teksti, mis ei lähe kokku loomulike piltide kirjeldamisega. Seega õppisin tõlkemudeli subtiitrite korpuse OPUS OpenSubtitles ja Euroopa parlamendi istundite kõnede korpuse Europarl ühendil. Kuigi Europarl korpus on poliitilise sisuga, sisaldab see endas siiski loomulikku kõnet, mis võiks olla abiks piltide kirjeldamisel. Europarl korpuse lisamisel on paralleeltekstides 69.4 miljonit ingliskeelset sõna. Võrdluseks, sarnaste mudelite treenimisel inglise ja prantsuse keele vahel tõlkimiseks, kasutatakse paralleeltekste, kus on 350 miljonit ingliskeelset sõna (350 miljonit hea kvaliteediga lauset on eraldatud 850-miljonilise sõnaga paralleelkorpusest [3]). Mudeli treenimiseks kasutasin lauseid, mis on kuni 50 sõna pikad.

Tõlkemudeli treenimise detailid

Mudeli treenimisel võtan eeskuju artiklist [3]. Sõnastikuna kasutasin 30 000 kõige sagedasemat ingliskeelset ja eestikeelset sõna. Kõik sõnad, mis ei kuulu 30 000 sagedasema sõna hulka, asendasin sümboliga *UNK*. Muud eeltöötlust ma tekstile ei rakendanud (sõnu ei muutnud väiketäheliseks, samuti ei teinud lemmatiseerimist). Mõlema neurovõrgu (nii kodeerija kui ka dekodeerija) peidetud kihi suuruseks oli 1000. Peidetud kihis on kasutatud GRU neuroneid.

Mudeli treenimiseks kasutasin rekurrentsete neurovõrkude paketti *GroundHog* [37], mis põhineb matemaatiliste avaldiste kompileerijal *Theano* [5]. Treenisin mudeli EENet kobararvuti süsteemis kasutades Nvidia Tesla K20 graafikakaarti. Treenimiseks kulus 5 päeva.

3.4.1 Kvalitatiivne analüüs

Tõlkemudeli kvalitatiivseks hindamiseks uurisin tõlketulemusi näidislausetel (tabel 3.1). Kolm esimest näidislauset on piltide kirjeldused, mis on võetud Microsoft COCO andmestikust [35]. Lõppeesmärgina soovingi just selliseid lauseid hästi tõlkida. Võrdluseks olen toonud ka *Google Translate* abil saadud tõlked ning Tartu Ülikooli keeletehnoloogia teadusgrupi masintõlkesüsteemi masintolge.ut.ee tõlked.

Mudel on õppinud edukalt kasutama komasid. Näiteks ingliskeelne lause:

A man is walking down a sidewalk holding an umbrella.

on tõlgitud:

Mees, kes kõnnib tänaval, hoiab vihmavarju käes.

Samuti on mudelil võime lause konteksti mõista ja vastavalt sellele mitmetähenduslikke sõnu tõlkida. Näiteks lause

A woman is playing tennis on a court.

korral on neurovõrkudel põhinev mudel aru saanud lause kontekstist ning tõlkinud sõna *court* väljakuks mitte kohtuks. Fraasipõhised süsteemid *Google Translate* ja masintolge.ut.ee tõlkisid selle kohtuks.

Kolme viimase näitelause sisuks on Kreeka võlakriis. Need laused olen võtnud portaalist *BBC News*. Kuna mudeli treeningkorpuses oli palju poliitlisi tekste, võiks mudel neid lauseid hästi tõlkida. Nii tõesti ka on. Mudel on võimeline tõlkima ka pikki lauseid keeleliselt ilusalt.

Mudeli suureks probleemiks on aga puuduvad sõnad (tabel 3.2). Kuna mudeli sõnastikus on vaid 30 000 erinevat sõna (erineva käände, pöörde ja algustähe suurusega sõnad loetakse erinevateks), on keeruline kogu loomulikku keelt modelleerida. Seega tagastab mudel tihti sümbolit *UNK*, mis tähistab kõiki neid sõnu, mida ei olnud 30 000 kõige sagedasema sõna seas.

Tõlke koostamisel on võimalik mudelit piirata selliselt, et sümboli *UNK* tagastamine pole võimalik. Mõnikord jääb lause sisu suuresti muutumatuks (näiteks *sõitma* asendatakse sõnaga *ratsutama*). Vahel aga muutub lause tähendus täiesti teistsuguseks. Näiteks Microsoft COCO andmestikus olev pildikirjeldus *A large clock tower with a clock on top.* esmane tõlge on *UNK UNK UNK UNK.* Kui sümbolit *UNK* mitte lubada, tagastab mudel tõlkeks *Mul on väga kiire*. Kuigi tähenduslikult muutus lause sisu teiseks, on siiski mõlemad laused kellaaja-teemalised.

Huvitaval kombel tagastab mudel tihti tõlkeks *Ma ei tea.*, kui tõlkimisel mitte lubada kasutada puuduva sõna sümbolit *UNK* (tabel 3.2). Seega on mudel automaatselt õppinud, et lausel *Ma ei tea.* ja mitmetest *UNK*-sümbolitest koosneval lausel on sarnane tähendus.

3.4.2 Kvantitatiivne analüüs

Mudeli tulemuste võrdlemine teiste tõlkesüsteemidega on keeruline. Esiteks pole standardeid, milliseid lauseid kasutatakse treening-, valideerimis- ja testandmetes. Või millist paralleelkorpust üldse kasutada.

Tõlkesüsteemi masintolge.ut.ee tulemusi inglise keelest eesti keelde tõlkimisel pole publitseeritud. Kuigi on teada, et vastav süsteem on treenitud Europarl, JRC-Acquis, OPUS

Tabel 3.1 Tõlkemudelite tõlked näidislausetel. Esimesed kolm näitelauset piltide kirjeldused (andmestikust Microsoft COCO [35]). Viimased kolm näidislauset on Kreeka võlakriisiga seotud laused portaalist BBC News. NT tähistab minu treenitud neurovõrkudel põhinevat mudelit, MT tähistab masintolge.ut.ee tõlget ja GT tähistab *Google Translate* tõlget.

Sisend:	*A man is walking down a sidewalk holding an umbrella.*
NT:	Mees, kes kõnnib tänaval, hoiab vihmavarju käes.
MT:	mees kõnnib sätestatakse kõnnitee holding vihmavarju
GT:	mees kõnnib mööda kõnniteed, kellel vihmavari
Sisend:	*A woman sitting on a couch with a dog.*
NT:	Naine istub diivanil koos koeraga.
MT:	Naist istumas diivanil koera
GT:	naine istub diivanil koer
Sisend:	*A woman is playing tennis on a court.*
NT:	Naine mängib väljakul tennist.
MT:	Naine on tennis kohtu kohta
GT:	naine mängib tennist kohus.
Sisend:	*Nobody believes that the Greek debt is sustainable.*
NT:	Mitte keegi ei usu, et Kreeka võlg on jätkusuutlik.
MT:	keegi ei usu , et kreeka võla jätkusuutlikkus
GT:	Keegi ei usu, et Kreeka võlg on jätkusuutlik.
Sisend:	*Despite the fact that there are differences in perspective, I am absolutely confident that we will soon manage to reach a mutually beneficial agreement, both for Greece and for Europe as a whole.*
NT:	Hoolimata asjaolust, et esineb erinevusi, olen ma täiesti kindel, et me saavutame peagi lahenduse, nii Kreeka kui ka Euroopa jaoks tervikuna.
MT:	hoolimata asjaolust , et leidub erinevusi perspektiivi , ma olen täiesti kindel , et me suudame jõuda peagi vastastikku kasulik leping nii kreeka kui ka kogu euroopa jaoks .
GT:	Vaatamata sellele, et esineb erinevusi perspektiivi, ma olen täiesti kindel, et me varsti õnnestub jõuda vastastikku kasulik leping, nii et Kreeka ja Euroopa tervikuna
Sisend:	*There is a feeling among many here that this is a historic moment, and supporters speak in terms of big ideas and dreams.*
NT:	Paljude seas on tunda, et see on ajalooline hetk, ja toetajad kõnelevad suurte ideede ja unistuste osas.
MT:	seal on tunne paljudest siin , et see on ajalooline hetk ja toetajate rääkida seoses suurte ideede ja unistused .
GT:	On tunne, paljude siin, et see on ajalooline hetk, ja pooldajad nii suuri ideid ja unistusi.

Tabel 3.2 Minu treenitud neurovõrkudel põhineva mudeli sõnastiku suuruseks on 30 000. Seega tagastab mudel tihti puuduva sõna sümbolit *UNK*. Tõlke leidmisel on võimalik *UNK* sümboli genereerimine välistada (tähistatud NT-UNK). Järgnevalt on näidatud tulemused pildikirjeldus lausetel. Huvitaval kombel tagastab mudel tihti tulemuseks *Ma ei tea*.

Sisend:	*A man riding a surfboard on top of a wave.*
NT:	UNK UNK laine peale .
NT-UNK:	Üks mees ratsutab laine otsas .
MT:	mees sõitis surfilauale laine peal
GT:	mees ratsutamine lainelaua peal laine
Sisend:	*A pizza with cheese and toppings on a plate.*
NT:	UNK pitsa ja UNK UNK .
NT-UNK:	Pitsa koos juustuga ja ...
MT:	Pitsa juustuga ja kastme taldriku peal
GT:	pizza juustu ja toppings plaadil
Sisend:	*A large clock tower with a clock on top.*
NT:	UNK UNK UNK UNK .
NT-UNK:	Mul on väga kiire .
MT:	Suur kellatorni kell koos tipus
GT:	suur kellatorn kella peal
Sisend:	*A bicycle is chained to a bicycle rack.*
NT:	UNK on UNK UNK .
NT-UNK:	**Ma ei tea .**
MT:	Aheldatud jalgratta jalgratta Rack
GT:	jalgratta külge aheldatud rattahoidja
Sisend:	*A display case filled with lots of donuts.*
NT:	UNK juhtum täidetud paljude UNK .
NT-UNK:	**Ma ei tea .**
MT:	Kuvar juhul palju sõõrikuid täis
GT:	vitriinkarpi täis palju sõõrikud
Sisend:	*A giraffe standing in a field of grass.*
NT:	UNK UNK UNK ees .
NT-UNK:	**Ma ei tea .**
MT:	kaelkirjak seisab väljal grass
GT:	kaelkirjak seisab rohuväli
Sisend:	*A computer keyboard and mouse on a desk.*
NT:	UNK ja UNK UNK .
NT-UNK:	**Ma ei tea .**
MT:	Arvuti klaviatuur ja hiir laual
GT:	arvuti klaviatuur ja hiir laual

ja muudel korpustel, on raske valida välja kasutamata testandmestikku, mille põhjal tulemusi võrrelda.

Inglise keelest eesti keelde tõlkimise kohta on tehtud kaks eksperimenti, mille tulemused on ka publitseeritud. Koehn et al võrdles 2009. aastal 462 erinevat keeltepaari, nende seas ka inglise keelest eesti keelde tõlkimist. Nad kasutasid paralleelkorpust JRC-Acquis ning said fraasipõhise tõlkesüsteemi korral BLEU skooriks 34.8 [26].

2010. aastal võrdles Khalilov et al fraasipõhise tõlkesüsteemi täpsusi läti, leedu ja eesti keele tõlkimisel inglise keelde ning vastupidi. Nad kasutasid samuti JRC-Acquis korpust, kuid mitte sama treening-, valideerimis- ja testandmestikku. Nemad said inglise keelest eesti keelde tõlkimisel BLEU skooriks 11.84 [21].

Isegi 3 ühikuline tõus BLEU skooris tähendab suurt edasiminekut tõlkesüsteemis. Khalilov et al artiklis pole kommenteeritud, miks nende süsteemi BLEU skoor on märkimisväärselt halvem eelnevalt publitseeritud tulemusest.

Tõenäoliselt jääb BLEU skooride erinevus erinevate treening- ning testvalimite tükelduse taha. Koehn et al filtreerisid kogu JRC-Acquis korpusest välja 12322 lauset, mida oli võimalik joondada 22-s keeles. Tasub tähele panna, et kogu JRC-Acquis andmestikus on üle 1 miljoni lause. Khalilov et al valisid juhuslikult välja 1000 lauset testimiseks, 500 valideerimiseks ja ülejäänusid kasutati treenimiseks.

Fraasipõhise mudeli treenimine paralleelkorpuse Europarl ja Opensubs ühendil pole selle töö prioriteet ning arvatavasti jääks ainult JRC-Acquis korpus liiga väikeseks neurovõrkudel põhineva mudeli treenimiseks. Seega testin töös koostatud mudelit, mis on treenitud Europarl ja Opensubs korpuste ühendil, JRC-Acquis korpuse lausetel.

Sellise testimise tulemused pole otseselt võrreldavad Khalilov et al fraasipõhise mudeli tulemustega, sest ühelt poolt kasutasin mina rohkem andmeid, aga teiselt poolt pole minu kasutatud korpuses seadustekste, mille peal tõlketäpsust testin.

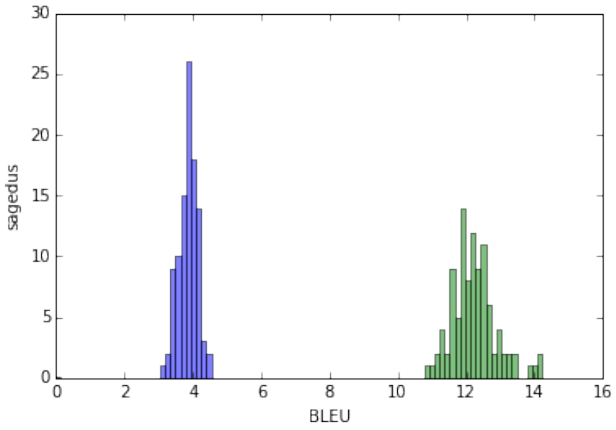
Näiteks Koehn et al artiklis uuriti, kui hästi töötab seadustekstide JRC-Acquis korpusel treenitud tõlkemudel uudisteandmetel. Kui esialgne BLEU skoor saksa keelest inglise keelde tõlkimisel oli 53.6, saadi uudisteandmete peal BLEU skooriks 11.6. Seega alati ei ole tõlkesüsteemid ülekantavad ühe korpuse pealt teisele ja seda peab meeles pidama ka praegu.

Kuna Koehn et al filtreeris esialgset andmestikku märgatavalt ning seda on raske reprodutseerida, võtan sarnaselt Khalilov et al eksperimendile juhuslikult 1000 lauset JRC-Acquis korpusest. Saadud testandmestikul arvutan BLEU skoori. Testandmestiku juhuslikkuse mõjust ettekujutuse saamiseks kordan seda protsessi 100 korda.

Sarnaselt artiklile [3], arvutasin BLEU skoori ka vaid sellistel lausetel, kus kõik sisendlause sõnad on mudeli sõnastikus (st 30 000 kõige sagedasema sõna seas) ja ka referentstõlke

sõnad on sõnastikus. Lisaks keelasin mudelil tagastada tõlkena sümbolit *UNK*. Sellise testandmestiku suuruseks võtsin samuti 1000 lauset ning kordasin protsessi 100 korda.

Keskmisteks BLEU skoorideks tulid vastavalt 3.8 ja 12.2 (joonis 3.5). Tulemustest on selge, et 30 000 sagedasema sõna kasutamine sõnastikuna on piirav ning tuleb leida viise, kuidas neurovõrkudel põhinevaid tõlkesüsteeme skaleerida suurtematele sõnastikele. Muuhulgas annab tulemus tundmuse, et teadaolevate sõnadega lausete peal on neurovõrkudel põhinev tõlkesüsteem vähemalt sama hea kui fraasipõhine tõlkesüsteem.



Joonis 3.5 BLEU skoorid sajal erineval testandmestikul (igaühes neist oli 1000 lauset). Rohelisega on näidatud BLEU skoorid, kui tõlgiti vaid selliseid lauseid, kus kõik sõnad olid tõlkesüsteemi sõnastikus olemas. Sisinega on näidatud BLEU skoorid, kui vastavat filtreerimist ei tehtud. Keskmisteks BLEU skoordieks tulid 3.8 ja 12.2 (standardhälbed olid vastavalt 0.3 ja 0.6)

Peatükk 4

Piltidest arusaamine

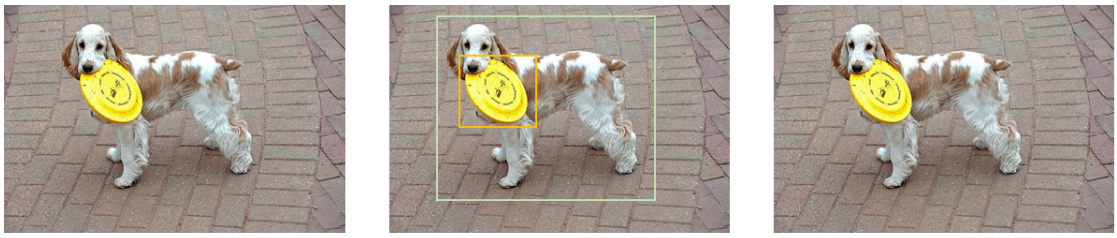
Pilt ütleb rohkem kui tuhat sõna. Aga kas ka *statistilisele masinale*?

Esmalt kirjeldan, kuidas treenida *statistilist masinat*, et pilt oleks väärt vaid ühe sõna - pildil olev tähtsaim objekt või pildi temaatika.

Ühesõnaline kirjeldus on harva piisav. Mida teha juhul, kui pildil on sama suurelt nii kass kui ka koer? Kas tagastada *koer*, *kass* või üldiselt *loomad*? Mõistlik on soovida, et *statistiline masin* leiab üles kõik pildil olevad objektid. Kirjutan, kuidas seda saavutada neurovõrkudega.

Objektide tuvastamine ei tähenda aga pildi mõistmist. Kui *statistiline masin* teab, et joonisel on nii koer kui ka lendav taldrik, kas ta oskab aru saada ka nende omavahelisest asukohast ja suhtest (joonis 4.1)? Kirjutan, kuidas pilti kirjeldada lausega.

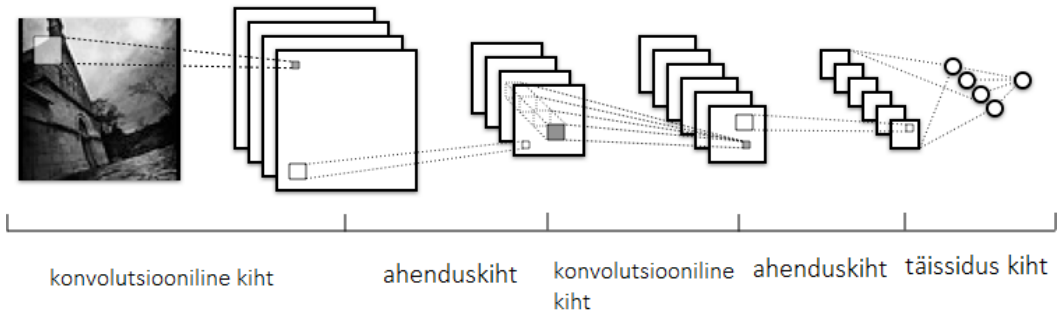
Lõpetuseks olen koostanud mudeli, mis genereerib automaatselt pildile eestikeelse kirjelduse ning näidanud selle mudeli tulemusi.



Joonis 4.1 Pildi kirjeldamine sõna, mitme sõna ja lausega.

4.1 Piltide kirjeldamine ühe sõnaga

Piltide kirjeldamiseks ühe sõnaga, on vaja kategoriseerida pilt mõnda etteantud gruppi. Kõige paremaid tulemusi piltide klassifitseerimisel on andnud konvolutsioonilised neurovõrgud [27].



Joonis 4.2 Tüüpilise konvolutsioonilise neurovõrgu arhitektuur [36].

Tüüpiline konvolutsiooniline neurovõrk koosneb konvolutsioonilistest, ahendus- (*pooling*) ja täissidus (*fully connected*) kihtidest (joonis 4.2). Võrgu sisendiks on mõõtmega $h \times w \times c$ pilt x, kus h on pildi kõrgus, w on pildi laius ning c tähistab (värvi)kanalite arvu. Konvolutsioonilises kihis on k filtrit f_i, igaüks mõõtmetega $h_f \times w_f \times c_f$, kus h_f, w_f, c_f ei ole suuremad vastavalt h-st, w-st ja c-st. Iga filter f_i tagastab tunnustekaardi v_i mõõtmetega $(h - h_f + 1) \times (w - w_f + 1)$, kus $v_i = g(x * f_i)$, kus g on mingisugune mittelineaarne funktsioon (näiteks sigmoidifunktsioon). Vajadusel lisatakse vabaliige $b \in \mathbb{R}$ (st $v_i = g(x * f_i + b)$). Konvolutsiooni operatsiooni $*$ illustreerib joonis 4.3.

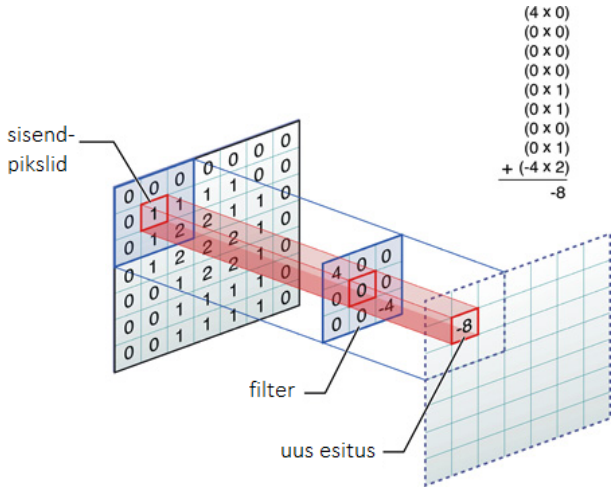
Ahenduskihis valime regiooni mõõtmetega $w_p \times h_p$ üle mille tunnuseid agregeeritakse. Pärast mõõtmete fikseerimist jagatakse ahenduskihi sisend (üksteisega mittekattuvateks) osadeks (igaüks suurusega $w_p \times h_p$) ning agregeeritakse (võetakse maksimum või keskmine), et saada ahendatud tunnustekaart (joonis 4.4).

Viimaseks kihiks on tüüpiliselt multinomiaalne kiht (logistilise funktsiooni üldistus K kategooria jaoks), mis normaliseerib skoorid ja mille väljundit saab interpreteerida kui klassi kuulumise tõenäosust. Täpsemalt, aktivatsioonivektori z korral on j-ndasse klassi kuulumise tõenäosus

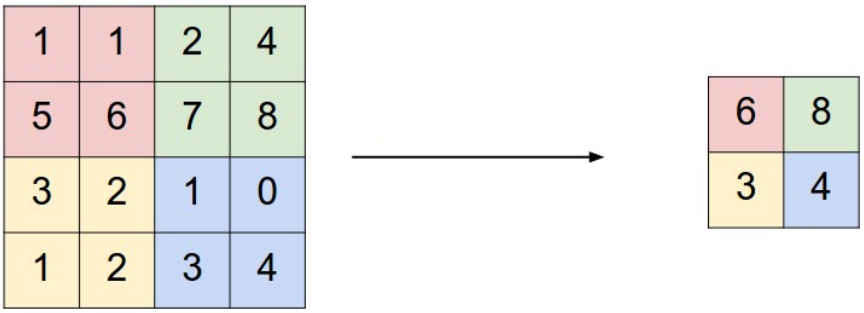
$$\frac{e^{z^T w_j}}{\sum_{k=1}^{K} e^{z^T w_k}},$$

kus w_k on vastava klassiga seotud kaalud.

Konvolutsiooniline kiht on kasulik, sest tunnused, mis on olulised ühes regioonis on tõenäoliselt olulised ka mujal. Näiteks õllepurk võib pildil esineda nii vasakul üleval kui



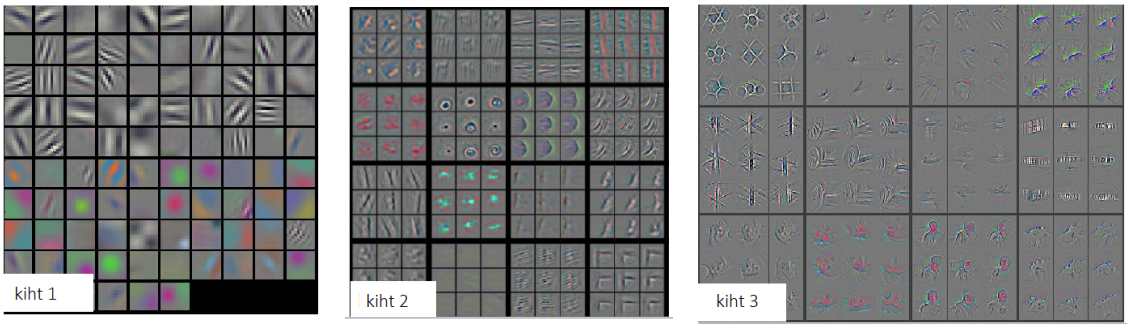
Joonis 4.3 Konvolutsiooni operatsioon asendab piksli väärtuse tema ja ta naabrite kaalutud keskmisega [2].



Joonis 4.4 Tüüpilises konvolutsioonilise neurovõrgu ahendamiskihis agregeeritakse maksimumi võttes (*max pooling*). Joonisel on maksimum võetud üksteisega mittekattuvatest 2×2 regioonidest [33].

ka all paremal ning seega on mõistlik õppida õllepurki tuvastada aitavad tunnused korraga. Konvolutsiooni operatsioon aitab vähendada õpitud parameetrite arvu. Ahendamine aitab saavutada aga nihke-invariantsust. See tähendab, et isegi kui nihutaksime pilti mõnes suunas veidi, on sama ahendatud tunnus ikka aktiivne.

Neurovõrgu igas järgmises kihis õpitakse abstraktsemaid tunnuseid. Loomulikel piltidel õpitud konvolutsioonilise neurovõrgu esimese kihi tunnused aitavad eristada mitmesuguseid ääriseid. Järgmiste kihtide tunnused eristavad juba konkreetsemaid osi (joonis 4.5).



Joonis 4.5 ImageNet andmestikul treenitud konvolutsioonilise neurovõrgu esimese kihi tunnused tuvastavad erinevaid ääriseid. Kõrgema kihi tunnused tajuvad konktreetsemaid objekte (näiteks autoratast) [53].

Konvolutsioonilise neurovõrk õpib pildile uue esituse. Näiteks propageerides pildi pikslid läbi esimese kihi, saame esituse, mis iseloomustab, millised äärised pildil esinevad. Propageerides saadud tunnused sügavamatesse kihtides, saame esituse, mis iseloomustab pilti terviklikumalt (näiteks kas pildil on vuntsid, kõrv, saba, ratas vms).

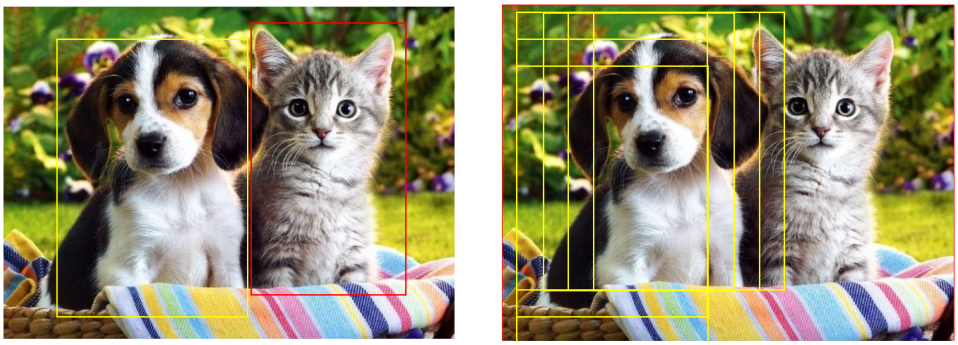
Seega võib treenitud konvolutsioonilist neurovõrku kasutada ka kui tunnuseeraldajat. Selle asemel, et töötada piksli väärtustega, võib pildi propageerida läbi konvolutsioonilise võrgu, mis on treenitud suurel sildistatud andmestikul, eraldada viimaste kihtide tunnused ning kasutada neid tunnuseid mõne muu ülesande lahendamisel. Näiteks ImageNet andmetel õpitud konvolutsioonilise võrgu tunnuseid on edukalt kasutatud nii objektide kategoriseerimisel, asukoha määramisel kui ka piltide otsingul [9], [44].

4.2 Piltide kirjeldamine mitme sõnaga

Ühesõnaline kirjeldus on piisav, kui pildil ongi kujutatud ühte objekti. Tüüpiliselt on aga piltidel objekte rohkem. Näiteks kui kõrvuti on kass ja koer, siis soovime, et algoritm tagastaks kirjelduseks mõlemad. Kuidas aga tuvastada mitut objekti?

Üks võimalus selleks on treenida mitu erinevat binaarset klassifitseerijat. Näiteks üks binaarne klassifitseerija on selline, mis tuvastab, kas pildil on koer või mitte. Teine kassifitseerija võib olla selline, mis tuvastab ainult kasse. Kui vastavad klassifitseerijad on olemas, saame neid kõiki pildi peal rakendada. Tulemuseks on pildi kirjeldus, kus on üks, mitu või mitte ühtegi tuvastatud silti.

Tihti tahame ka teada, kus objekt asub. Kas pildi paremas nurgas, keskel või kuskil mujal? Objekti asukoha määramiseks võime vaadelda erineva pikkuse ja laiusega alampilte ning neid klassifitseerida (joonis 4.6).

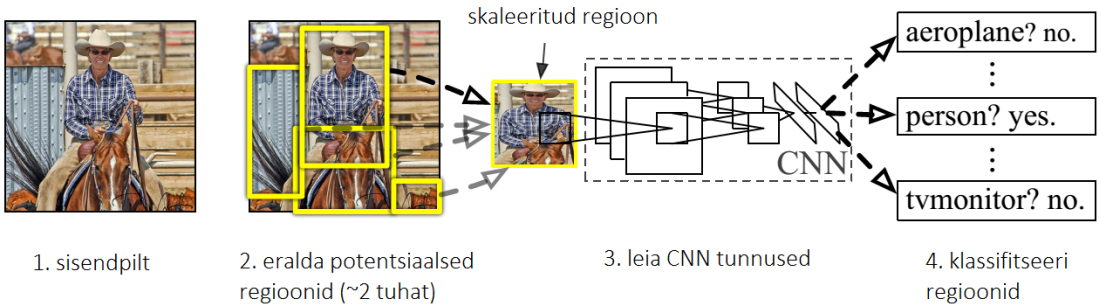


Joonis 4.6 Tihti soovime ka objektide asukohta määrata (vasakpoolne pilt). Üks viis selleks on käia läbi alampilte ning igaühte neist klassifitseerida.

Hiljutine häid tulemusi näidanud meetod mitme objekti tuvastamiseks ja nende asukoha kindlaks tegemiseks on R-CNN (*Regions with CNN features*) [12]. Selle asemel, et vaadelda kõiki võimalikke alampilte, vaadeldakse ligikaudu 2000 regiooni, mis valitakse sõltuvalt pildi statistikutest. Seejärel transformeeritakse regioon sobiva mõõtmetega pildiks, et propageerida see läbi konvolutsioonilise võrgu, et saada regioonile uus esitus. Saadud esitust klassifitseeritakse mitme erineva binaarse klassifitseerijaga (joonis 4.7).

4.3 Piltide kirjeldamine lausega

Automaatne piltide kirjeldamine lausega on väga lähedane tehisnägemise suurele eesmärgile - mõista pilte. Enne 2014. aastat kasutati lauselise kirjelduse saamiseks peamiselt šablooni-põhiseid meetodeid [28] [34]. See tähendab, et esmalt valitakse välja mõistlikud lausete šabloonid, mis täidetakse pildilt tuvastatud objektidega. Näiteks joonise 4.1 korral võiks tüüpiline kirjeldus olla: *Pildil on üks koer ja üks lendav taldrik. Lendav taldrik on pruuni koera lähedal.* Mõne teise pildi korral aga: *Pildil on kaks koera. Pruun koer on musta koera*

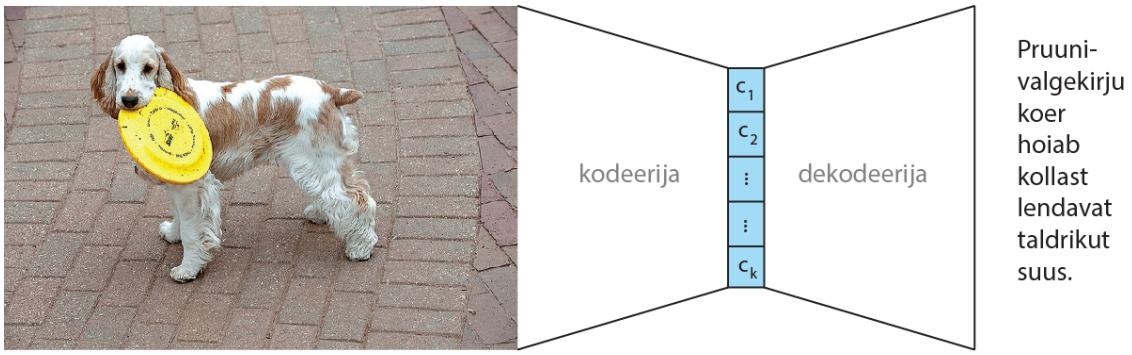


Joonis 4.7 Objektide tuvastamise ja asukoha määramise süsteemi R-CNN ülevaade. Sisendpildilt eraldatakse mitmeid alamregioone, mis propageeritakse läbi eeltreenitud konvolutsioonilise võrgu, mille tulemusena saadakse regioonile uus esitus, mida seejärel klassifitseeritakse.

kõrval. Teine lähenemine, mida kasutati pildi kirjelduse saamiseks, oli sisendpildiga sarnaste piltide kirjelduste kombineerimine [29].

2014. aastal pakuti välja neurovõrkudel põhinev lähenemine pildikirjelduste genereerimiseks. Lühikese ajavahemiku vältel publitseerisid vähemalt kuus erinevat teadusgruppi oma meetodid neurovõrkudel põhineva automaatse kirjelduse saamiseks: [23], [20], [52], [24], [8], [38] ja [11].

Kõik need meetodid põhinevad kodeerija-dekodeerija lähenemisel. Kõigepealt kodeeritakse pilt fikseeritud pikkusega vektoriks. Seejärel genereerib dekodeerija fikseeritud pikkusega vektorist loomulikus keeles lause (joonis 4.8). See on väga sarnane kolmandas peatükis kirjeldatud neurovõrkudel põhinevate tõlkesüsteemidega. Pildi kirjelduse saamisest võibki mõelda, kui pildi *tõlkimisest* eesti keelde.



Joonis 4.8 Pilt kodeeritakse fikseeritud suurusega vektoriks. Seejärel dekodeeritakse vastav vektor lauseks.

Neurovõrkudel põhinevad meetodid erinevad keelemudeli valiku poolest: artiklis [23] kasutati edasileviga neurovõrku, artiklites [38] ja [20] aga rekurrentset neurovõrku. Artiklites [52] ja [8] kasutati tavalise rekurrentse neurovõrgu asemel LSTM neuroneid. Teine suurem erinevus on selles, kas pilditunnuseid arvestatakse vaid rekurrentse neurovõrgu esimeses peidetus kihis [52], [20] või dekodeerija kõigis peidetud olekutes [38].

Vastavad meetodid põhinevad suurtel pildi andmebaasidel, kus piltide juures on kirjeldused (mõnikord kuni viie erineva inimese kirjeldus). Üheks selliseks andmestikuks on Microsoft COCO (näitelaused joonisel 4.9).

Joonis 4.9 Pildi automaatsed kirjeldajad põhinevad suurtel pildi andmebaasidel, kus iga pildi juures on ka tekstiline kirjeldus. Näidatud on kolm pilti koos kirjeldusega andmestikust Microsoft COCO.

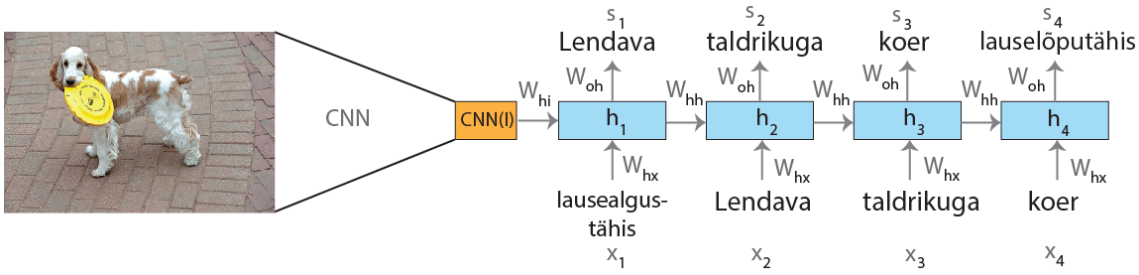
Olgu meil pilt I, millele soovime saada kirjelduse S. See tähendab, et tahame leida lauset $S = (s_1, ..., s_T)$, mis maksimiseerib tõenäosuse $p(S|I)$. Kasutades ketireeglit, saame $p(S|I)$ faktoriseerida korrutiseks

$$\prod_{t=1}^{T} p(s_t|I, s_1, ..., s_{t-1}),$$

kus T on kirjelduse pikkus. Üks võimalik viis $p(s_t|I, s_1, ..., s_{t-1})$ modelleerimiseks on kasutada rekurrentset neurovõrku. Rekurrentne neurovõrk võtab sisendiks pildi I ja jada sõnade esitustest $(x_1, ..., x_T)$. Seejärel arvutatakse peidetud tunnused $(h_1, ..., h_T)$ ning väljundid $(s_1, ..., s_T)$. Vastav rekurrentne seos on:

$$\begin{aligned} b_v &= W_{hi}[CNN(I)], \\ h_t &= f(W_{hx}x_t + W_{hh}h_{t-1} + b_h + b_v), \\ s_t &= g(W_{oh}h_t + b_o), \end{aligned}$$

kus $W_{hi}, W_{hx}, W_{hh}, W_{oh}, b_h, b_o$ on õpitavad parameetrid, $CNN(I)$ tähistab konvolutsioonilise neurovõrgu viimase kihi tunnuseid, f on mõni mittelineaarne funktsioon (näiteks *tanh*) ja g on tüüpilselt multinomiaalne logistiline funktsioon. Ainuke erinevus võrreldes esimeses peatükis kirjeldatud rekurrentsel neurovõrgul põhineva keelemudeliga on lisatunnus b_v.



Joonis 4.10 Rekurrentne võrk võtab sisendiks sõna, eelmise ajahetke konteksti ja pildi ning defineerib lause järgmise sõna jaotuse. Pildi informatsiooni võib kasutada vaid esimesel ajahetkel (nagu sellel joonisel kujutatud).

Mudeli treenimisel seatakse $h_1 = \vec{0}$, x_1 saab väärtuseks spetsiaalse lausealgustähise vektoresituse ja s_1 on soovitud esimene sõna. Seejärel väärtustatakse x_2 selle sama soovitud sõna vektoresitusega ja loodetakse, et s_2 on lähedane kirjelduse teisele sõnale (joonis 4.10).

Et saada pildile I kirjeldust, tuleb kõigepealt propageerida pilt läbi konvolutsioonilise neurovõrgu, et saada esitus $CNN(I)$. Seejärel algväärtustada $h_1 = \vec{0}$ ja anda tunnusele x_1 väärtuseks spetsiaalse lausealgustähise vektoresituse. Seejärel saab leida tõenäosusjaotuse üle sõna s_1, mis võimaldab simuleerida esimese sõna realisatsiooni (või võtta maksimaalse tõenäosusega sõna). See protsess kordub kuniks on simuleeritud lauselõpu märgis.

4.4 Eksperiment: automaatne piltide kirjeldamine eestikeelse lausega

Eesti keele jaoks pole suuri pildiandmebaase koos kirjeldustega (nt analoogilist Microsoft COCO andmestikule [35]). Seega ei ole võimalik eelmises sektsioonis kirjeldatud meetodeid otse rakendada.

Eestikeelse kirjelduse saamiseks kasutan sarnast tehnikat nagu masintõlkes, kui on vaja tõlkida näiteks jaapani keelest eesti keelde, kuid vastav paralleelkorpus puudub. Sel juhul kasutatakse tõlke saamiseks vahekeelt (näiteks inglise keelt). See tähendab, et kõigepealt tõlgitakse jaapani keelest inglise keelde ning seejärel inglise keelest eesti keelde.

Sarnaselt tõlgin mina pildi kõigepealt inglise keelde. Selleks kasutan eeltreenitud mudelit [20]. Seejärel tõlgin ingliskeelse kirjelduse eesti keelde kasutades eelmises peatükis kirjeldatud neurovõrkudel põhinevat tõlkesüsteemi.

Mudel töötab üllatavalt hästi (joonis 4.11). Mudel on võimeline täpselt kirjeldama nii tennisemängijat, hobuseid rannas kui ka vihmavarju hoidmist. Muidugi teeb mudel ka vigu (joonis 4.12). Vahel tulevad vead valest ingliskeelsest kirjeldusest. Näiteks elevandi seljas ratsutamine asendatakse hobuse seljas ratsutamisega. Vahel tulevad vead tõlkimisest. Näiteks pesapallikurikas tõlgitakse pesapalliks. Vahel tekib viga mõlemast allikast.

a cow is standing in the middle of a street

NT: **lehm seisab keset tänavat .**

UT: Lehm seisab keset Street

GT: lehm seisab keset tänavat

a woman holding a pink umbrella in the rain

NT: **Naine hoiab roosat vihmavarju vihma käes .**

UT: Naine, kellel on roosa vihmavarju vihmas

GT: naine, kellel on roosa vihmavari vihm

a group of people sitting around a table with food

NT: **Paljud inimesed istuvad laua ääres koos toiduga .**

UT: rühm inimesi laua ümber istuvad koos toiduga

GT: grupp inimesi istub laua koos toiduga

a woman is playing tennis on a court

NT: **Naine mängib väljakul tennist .**

UT: Naine on tennis kohtu kohta

GT: naine mängib tennist kohus

a man is playing a video game in a room

NT: **Mees mängib ruumis videomänge.**

UT: mees mängib videomäng ruumis

GT: mees mängib videomängu ruumis

a man and a woman standing next to each other

NT: **Mees ja naine seisavad teineteise kõrval .**

UT: mees ja naine üksteise kõrval

GT: Mees ja naine seisab kõrvuti

a group of sheep standing in a field

NT: **Kamp lambaid seisab põllul .**

UT: lammaste rühma alalise valdkonnas

GT: rühma lambad seisab väljal

a dog is standing on a beach near a body of water

NT: **Koer seisab rannas vee lähedal.**

UT: koer seisab lahedal rannas veekogu

GT: koer seisab ranna lähedal veekogu

a man riding a horse on top of a beach

NT: **Mees , kes ratsutab hobusega rannal .**

UT: mees ratsutades lõhkuva metshobuse seljas tipus beach

GT: mees ratsutamine peal rannas

Joonis 4.11 Näiteid genereeritud piltide kirjeldustest. Kaldkirjas on CNN-RNN mudeli ingliskeelne kirjeldus. Rasvaselt on näidatud neurovõrkudel põhineva tõlkemudeli tõlge. UT tähistab masintolge.ut.ee tõlget ning GT tähistab *Google Translate* tõlget.

Joonis 4.12 Näiteid valesti genereeritud kirjeldustest. Vead võivad tekkida valest genereeritud ingliskeelsest kirjeldusest, valest tõlkest või mõlemast.

Kokkuvõte

Töö eesmärgiks oli koostada mudel, mis on võimeline pilte eesti keeles kirjeldama. See eesmärk täideti edukalt. Näiteks oli valminud mudel suuteline tuvastama, et naine hoiab roosat vihmavarju vihma käes või et mees ratsutab hobusega rannal.

Piltide automaatse kirjeldamise tegid võimalikuks väga hiljutised arengud tehisnägemises ja keeletehnoloogias. Kuna vastavad neurovõrkudel põhinevad keeletehnoloogiameetodid olid aga eesti keele jaoks uurimata, ei olnud teekond eesmärgini sirgjooneline. Loomulikus eesti keeles pilte kirjeldava mudelini jõudmiseks tegin muuhulgas järgmist:

- Tüüpilistes keeletehnoloogia algoritmides rakendatakse eestikeelsetele tekstidele eeltöötlust (nagu lemmatiseerimine, morfoloogiline analüüs jne). Näitasin, et suure korpuse peal õpivad distributiivsemantika algoritmid automaatselt grammatilisi kontseptsioone nagu käänded ja pöörded ning ilma eeltöötluseta on võimalik edukalt lausete meelsust tuvastada.

- Näitasin, kuidas suure eestikeelse korpuse peal õppida sõnadele distributiivseid vektoresitusi ning neid edukalt kasutada klassifitseerimisülesandes, kus andmemahud on väikesed.

- Koostasin eestikeelse analoogiaandmestiku, kus on üle 10 000 analoogiaküsimuse ning mille abil saab võrrelda distributiivsemantika algoritme.

- Pakkusin välja meetodi, kuidas suurtelt võõrkeelsetelt korpustelt kanda informatsiooni üle eestikeelsetele rakendustele. Selleks õppisin inglis- ja eestikeelsetele sõnadele ühise vektorruumi. Kasutasin seda ruumi, et õppida klassifitseerija ainult ingliskeelsetel tekstidel ning seejärel kasutada seda eestikeelsete tekstide sildistamisel. Kuigi mina ei suutnud selles osas edukaid tulemusi näidata, on analüüsisuund perspektiivikas ning vajaks edasist uurimist.

- Treenisin uudse tõlkesüsteemi ingliskeelsete lausete tõlkimiseks eesti keelde. Vastav rekurrentsetel neurovõrkudel põhinev tõlkesüsteem tagastab keeleliselt ilusaid tulemusi.

Aga enne kui seda saab kasutada eraldi tõlkesüsteemina, tuleb leida viis, kuidas skaleerida neurovõrku suurematele sõnastikele.

Kirjandus

[1] Altrov, R. and Pajupuu, H. (2012). Estonian emotional speech corpus: theoretical base and implementation. In *4th international workshop on corpora for research on emotion sentiment & social signals (ES3)*, pages 50–53.

[2] Apple Inc. (2015). Performing convolution operations. https://developer.apple.com/library/mac/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html. Külastatud: 2015-05-06.

[3] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[4] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

[5] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX.

[6] Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[7] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[8] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2014). Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*.

[9] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013). Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*.

[10] Elson, J., Douceur, J. R., Howell, J., and Saul, J. (2007). Asirra: a captcha that exploits interest-aligned manual image categorization. In *ACM Conference on Computer and Communications Security*, pages 366–374.

[11] Fang, H., Gupta, S., Iandola, F., Srivastava, R., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J., et al. (2014). From captions to visual concepts and back. *arXiv preprint arXiv:1411.4952*.

[12] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE.

[13] Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12.

[14] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

[15] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[16] Kaalep, H.-J. and Muischnek, K. (2002). *Eesti kirjakeele sagedussõnastik*. Tartu Ülikool.

[17] Kaggle (2014). Create an algorithm to distinguish dogs from cats. https://www.kaggle.com/c/dogs-vs-cats. Külastatud: 2015-04-27.

[18] Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *EMNLP*, pages 1700–1709.

[19] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[20] Karpathy, A. and Fei-Fei, L. (2014). Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*.

[21] Khalilov, M., Pretkalnina, L., Kuvaldina, N., and Pereseina, V. (2010). Smt of latvian, lithuanian and estonian languages: a comparative study. In *Baltic HLT*, pages 117–124.

[22] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[23] Kiros, R., Salakhutdinov, R., and Zemel, R. (2014a). Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603.

[24] Kiros, R., Salakhutdinov, R., and Zemel, R.Š. (2014b). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

[25] Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

[26] Koehn, P., Birch, A., and Steinberger, R. (2009). 462 machine translation systems for europe. *Proceedings of MT Summit XII*, pages 65–72.

[27] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[28] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2011). Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer.

[29] Kuznetsova, P., Ordonez, V., Berg, A. C., Berg, T. L., and Choi, Y. (2012). Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics.

[30] Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

[31] Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., and Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861.

[32] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM.

[33] Li, F.-F. and Karpathy, A. (2015). Cs231n convolutional neural networks for visual recognition. http://cs231n.github.io/convolutional-networks/. Külastatud: 2015-05-06.

[34] Li, S., Kulkarni, G., Berg, T. L., Berg, A. C., and Choi, Y. (2011). Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228. Association for Computational Linguistics.

[35] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer.

[36] LISA lab, U. o. M. (2015a). Convolutional neural networks (lenet). http://deeplearning.net/tutorial/lenet.html. Külastatud: 2015-05-06.

[37] LISA lab, U. o. M. (2015b). Groundhog - library for implementing rnns with theano. https://github.com/lisa-groundhog/GroundHog. Külastatud: 2015-05-03.

[38] Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2014). Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*.

[39] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[40] Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

[41] Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

[42] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[43] Pascanu, R., Mikolov, T., and Bengio, Y. (2012). On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.

[44] Razavian, A.Š., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE.

[45] Schwenk, H. (2012). Continuous space translation models for phrase-based statistical machine translation. In *COLING (Posters)*, pages 1071–1080.

[46] Schwenk, H., Dchelotte, D., and Gauvain, J.-L. (2006). Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 723–730. Association for Computational Linguistics.

[47] Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*.

[48] Strapparava, C. and Mihalcea, R. (2007). Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

[49] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

[50] Tiedemann, J. (2009). News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

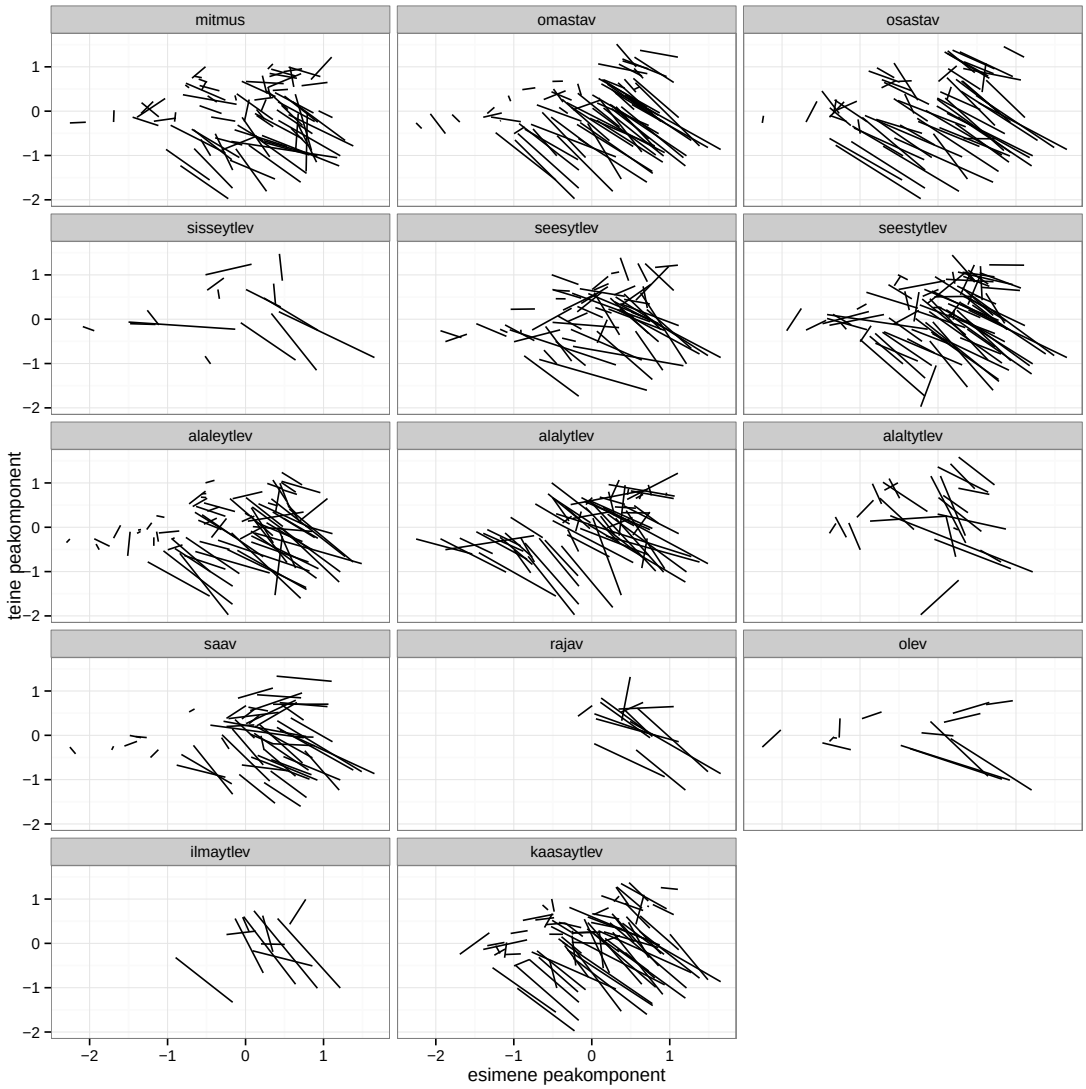
[51] Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *LREC*, pages 2214–2218.

[52] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2014). Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*.

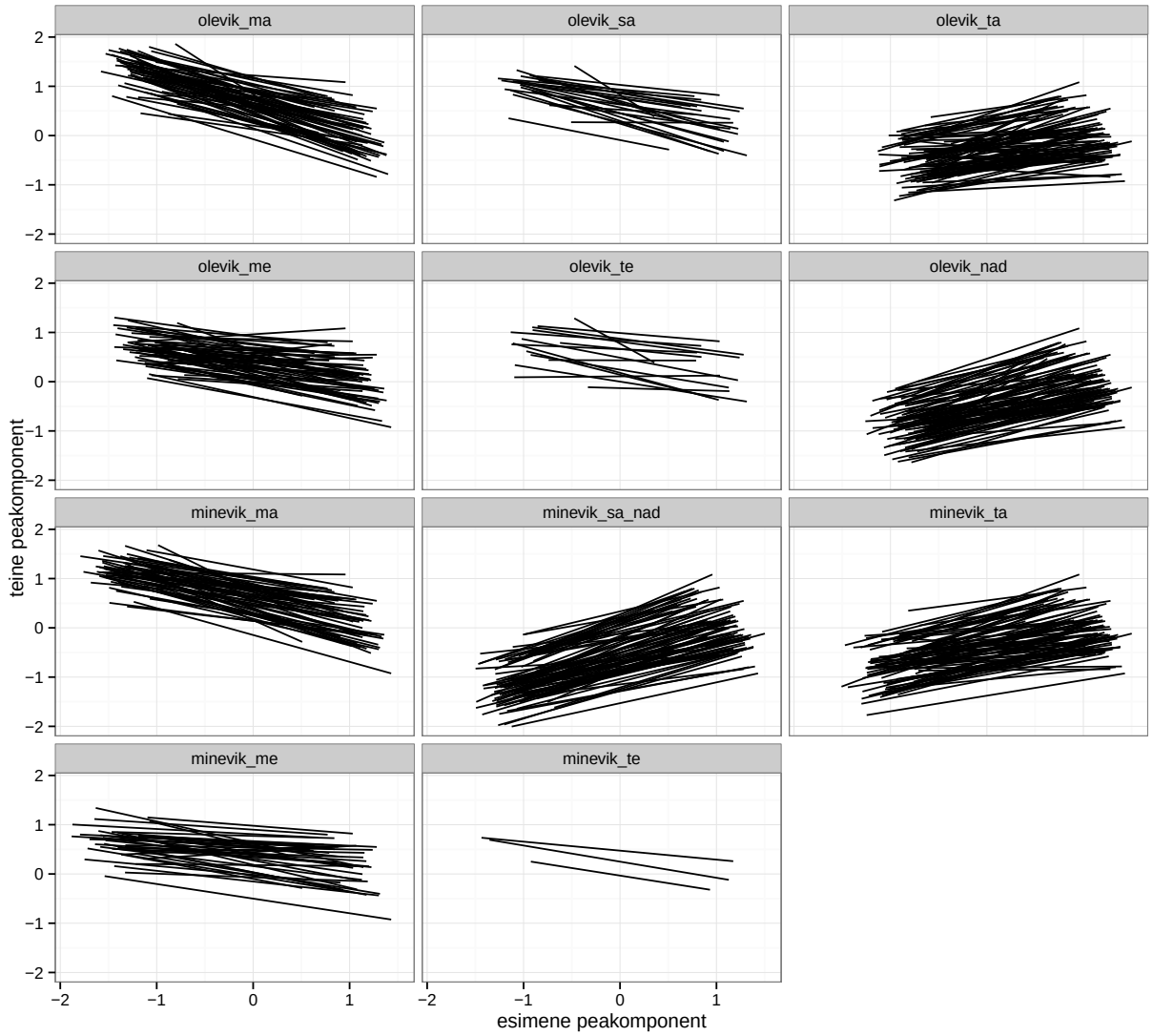
[53] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer.

Lisa A

Sõnavektorite lisatulemused



Joonis A.1 1500 sagedasema sõna seast olen valinud välja nimisõnad ja vastavatele vektoresitustele teinud dimensioonide vähendamist kasutades peakomponentanalüüsi. Joonisel on näidatud kaks esimest peakomponenti. Sõnad on liigendatud käände või mitmuse lõikes. Joon ühendab sõna algvormi ja käände (või mitmuse) vektoresitust.



Joonis A.2 1500 sagedasema sõna seast olen valinud välja tegusõnad ja vastavatele vektoresitustele teinud dimensioonide vähendamist kasutades peakomponentanalüüsi. Joonisel on näidatud kaks esimest peakomponenti. Sõnad on liigendatud tegusõna pöörde lõikes. Joon ühendab tegusõna algvormi (ma-tegevusnime) ja pöörde vektoresitust.

Tabel A.1 Analoogiaülesande tulemused kategooriate lõikes.

Analoogia tüüp	õigete analoogiate arv			kokku teste	täpsus protsentides		
	svd	ppmi	w2v		svd	ppmi	w2v
Rahvus	1	23	36	48	2,08%	47,92%	75,00%
Pealinn	8	11	21	65	12,31%	16,92%	32,31%
Rahaühik	0	0	1	11	0,00%	0,00%	9,09%
Sugu	0	27	31	42	0,00%	64,29%	73,81%
Nimisõna -omastav	1	63	125	360	0,28%	17,50%	34,72%
Nimisõna - osastav	0	41	129	414	0,00%	9,90%	31,16%
Nimisõna - sisseütlev	1	1	2	14	7,14%	7,14%	14,29%
Nimisõna - seesütlev	0	1	15	188	0,00%	0,53%	7,98%
Nimisõna - seesütlev	0	1	15	188	0,00%	0,53%	7,98%
Nimisõna - seestültev	2	2	90	345	0,58%	0,58%	26,09%
Nimisõna - alaleütlev	2	7	107	319	0,63%	2,19%	33,54%
Nimisõna - alalütlev	2	2	52	241	0,83%	0,83%	21,58%
Nimisõna - alaltütlev	0	1	6	32	0,00%	3,13%	18,75%
Nimisõna - saav	1	9	22	140	0,71%	6,43%	15,71%
Nimisõna - rajav	0	0	3	17	0,00%	0,00%	17,65%
Nimisõna - ilmaütlev	0	0	0	8	0,00%	0,00%	0,00%
Nimisõna - kaasaütlev	12	9	104	308	3,90%	2,92%	33,77%
Tegusõna - olevik 1. isik ainsus	6	11	120	191	3,14%	5,76%	62,83%
Tegusõna - olevik 2. isik ainsus	0	3	19	27	0,00%	11,11%	70,37%
Tegusõna - olevik 3. isik ainsus	3	72	274	416	0,72%	17,31%	65,87%
Tegusõna - olevik 1. isik mitmus	15	5	184	233	6,44%	2,15%	78,97%
Tegusõna - olevik 2. isik mitmus	0	0	5	12	0,00%	0,00%	41,67%
Tegusõna - olevik 3. isik mitmus	1	31	349	409	0,24%	7,58%	85,33%
Tegusõna - lihtminevik 1. isik ainsus	2	7	124	165	1,21%	4,24%	75,15%
Tegusõna - lihtminevik 2. isik ainsus	2	3	289	393	0,51%	0,76%	73,54%
Tegusõna - lihtminevik 3. isik ainsus	0	2	227	414	0,00%	0,48%	54,83%
Tegusõna - lihtminevik 1. isik mitmus	2	6	58	75	2,67%	8,00%	77,33%
Tegusõna - lihtminevik 2. isik mitmus	0	0	2	2	0,00%	0,00%	100,00%

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Tanel Pärnamaa (sünnikuupäev 8. september 1991),

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   "Piltide automaatne kirjeldamine eesti keeles - visuaalse ja semantilise ühisesituse õppimine neurovõrkudega",
   mille juhendajad on Leopold Parts ja Sven Laur,

   (a) reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

   (b) üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus 13. mail 2015