TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Herman Petrov 221583IABM

SUURTE KEELEMUDELITE RAKENDAMINE VÕTMESÕNADE TUVASTAMISEKS EESTIKEELSETEST TEKSTIDEST

Magistritöö

Juhendaja: Ahti Lohk
PhD

Kaasjuhendaja: Kais Allkivi-Metsoja
MA

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Herman Petrov

Kuupäev: 10.05.2024

Annotatsioon

Suurte keelemudelite rakendamine võtmesõnade tuvastamiseks eestikeelsetest tekstidest

Magistritöö peamine eesmärk on selgitada välja, kui tõhusalt on võimalik eestikeelsetest tekstidest võtmesõnu ekstraheerida, kasutades eesti keelt toetavaid suuri keelemudeleid ja KeyBERT-i teeki. Seejuures tehakse kindlaks, milliste keelemudelite ja seadistustega annab KeyBERT kõige täpsemaid tulemusi võrreldes inimmärgendatud võtmesõnadega. Töös rakendatakse KeyBERT-iga seitset mitmekeelset ja kahte eestikeelset transformeri arhitektuuril põhinevat keelemudelit. Võrdlusalusena kasutatakse kahte statistilist võtmesõnatuvastuse meetodit *simple maths* ja TextRank.

Erinevate meetodite tulemuslikkust hinnatakse Eesti Rahvusringhäälingu raadiosaadete transkriptsioonide alusel, mille on võtmesõnastanud kaks Tallinna Ülikooli filoloogiharidusega töötajat. Lähtutakse tekstide F1-skoori aritmeetilisest keskmisest. Kuigi märgendatud testmaterjal sisaldab valdavalt üksikuid võtmesõnu, katsetatakse ka KeyBERT-i võtmefraaside otsingu funktsiooni. Kahe- ja kolmesõnaliste võtmefraaside olulisust hinnatakse selle põhjal, kui sageli need sisaldavad filoloogide valitud võtmesõnu.

Analüüsi tulemusena leiti, et KeyBERT-i abil võtmesõnade ekstraheerimiseks on kõige täpsem kasutada lause-transformeritel põhinevaid keelemudeleid, mis on seadistatud rangelt sarnastele leitud võtmesõnadele keskenduma, kasutades suhteliselt väikest mitmekesisusfaktorit 0.2.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 43 leheküljel, 3 peatükki, 12 joonist ja 12 tabelit.

Abstract

Applying Large Language Models for Keyword Detection from Estonian Texts

The main objective of the master's thesis is to evaluate the efficiency of keyword extraction from Estonian texts using large language models that support Estonian language and the KeyBERT library. It is determined which language models and KeyBERT configurations work best, providing most accurate results compared with human-annotated keywords. During the study, nine different transformer models are applied in KeyBERT, including seven multilingual models and two language-specific models for Estonian. The KeyBERT method is compared with two statistical keyword extraction methods, simple maths and TextRank.

The results are assessed based on radio transcripts of the Estonian Public Broadcasting, annotated by two philologists from Tallinn University. Mean F1-score is calculated to evaluate keyword extraction accuracy. Additionally, KeyBERT's keyphrase function is tested, although the annotated test material comprises lists of single keywords. Two- and three-word keyphrases are considered relevant if they contain a human-annotated keyword. Conclusions are reached regarding KeyBERT settings and choice of a language model optimal for each scenario – keyword and keyphrase extraction from Estonian texts.

Based on the analysis, it is concluded that for extracting keywords using KeyBERT, it is most accurate to use sentence-transformer based language models configured to focus strictly on similar existing keywords with a relatively low diversity factor of 0.2.

The thesis is written in Estonian and contains text across 43 pages, three chapters, and 12 figures and 12 tables.

Lühendite ja mõistete sõnastik

Keelekorpus	Struktureeritud tekstide või kõne kogum, mida kasutatakse keele teaduslikuks uurimiseks.
Lemma	Sõna põhi- ehk algvorm.
Lemmatiseerimine	Protsess, kus sõnad teisendatakse nende algvormide kujule.
Märgendamine	Protsess, kus tekstilõigule on lisatud täiendavad sildid või märgendid, nagu sõnaliigid, lause struktuur või semantiline info.
MMR	*Maximal Marginal Relevance* ehk maksimaalne marginaalne seoes on meetod, mida kasutatakse andmete järjestamiseks nii, et maksimaalselt varieerida väljundit, vähendades samal ajal dubleerimist.
Mitmekesisuse faktor	Kaal, mis määrab, kui palju eelistatakse mitmekesiseid tulemusi võrreldes sarnastega informatsiooni otsingus.
N-gramm	Sõnade või sümbolite järjestus, mis koosneb n üksusest. Näiteks unigramm koosneb ühest sõnast, bigramm koosneb kahest sõnast, trigramm kolmest.
Nimiolemid	Nimed, kohanimed või muud olulised üksused tekstis, mida saab automaatselt tuvastada ja eraldada, näiteks inimeste nimed, kohad, organisatsioonid.
Osakaal	Protsentuaalne näitaja, mis kirjeldab, kui suur osa tervikust mingil komponendil on.
Semantiline sarnasus	Mõõt, mis kirjeldab kahe või enama sõna või lause tähenduslikku sarnasust.
Sihtkorpus	Kasutaja vaadeldav korpus, millest sõnu ja termineid eraldatakse.
Sõna vektoresitus	Tihedad numbrilised vektorid, mis hõlmavad iga sõna semantilist tähendust.

Transformer-arhitektuuri keelemudelid	Keelemudelid peamiselt keele analüüsimiseks, tuues välja semantilise ja süntaktilise sarnasusi tekstides.
Transformer-mudel	Närvivõrgu arhitektuur, mis õpib järjestikuste andmete konteksti ning genereerib selle põhjal uusi andmeid.
Transformer-mudeli kodeerija	Transformer-mudeli osa, mis võtab sisendina tekstilise andmevoo, töödeldes seda mitme kihi kaudu, et luua kontekstuaalseid esindusi sisendist.
Transkribeerimine	Protsess, mille käigus konverteeritakse kõne või audiosalvesti kirjalikuks vormiks.
Võrdluskorpus	Keelekorpus, mida kasutatakse võrdlusalusena sihtkorpusest võtmesõnade eraldamiseks.
Võtmefraas	Oluline fraas või lause, mis annab edasi dokumenti või tekstilõiku kõige iseloomustavama info.
Võtmesõnad	Sõnad, mis kirjeldavad teksti või dokumendi peamist sisu.
Võtmesõnade ekstraheerimine	Oluliste ja sisukate sõnade või fraaside leidmist ning eraldamist tekstist.

Sisukord

1. Sissejuhatus 11
1.1. Probleem 12
1.2. Käsitlusala ja eesmärk 12
1.3. Tööprotsess 13
1.4. Töö ülevaade 13
2. Teoreetiline raamistik 15
2.1. Võtmesõnade ekstraheerimine 15
2.1.1. Statistik *simple maths* 17
2.1.2. TextRank 18
2.2. Transformer-mudel 19
2.2.1. Transformer-mudeli üldine arhitektuur 20
2.2.2. Transformer-mudeli kodeerija 23
2.3. Keelemudelid 27
2.4. KeyBERT 29
3. Andmestik ja meetodite rakendamine 32
3.1. Andmestik: transkribeeritud raadiosaated 32
3.2. Märgendatud andmestik 33
3.3. Võrdlusandmete arvutamine 34
3.3.1. Võrdlusandmestiku variandid 35
3.3.2. TextRanki meetodi rakendamine 36
3.3.3. *Simple maths*'i meetodi rakendamine 36
3.3.4. KeyBERT-i meetodi rakendamine 37
3.3.5. KeyBERT-i andmestiku korrastamine 39
3.4. Üksikvõtmesõnade täpsuse hindamine 39
3.5. Üksikute võtmesõnade põhjal võtmefraaside hindamine 41
4. Tulemused 43
4.1. Võtmesõnade hindamine F1-skoori alusel 43
4.2. Võtmesõnade ja -fraaside hindamine märgendatud võtmesõnade esindatuse järgi 45
4.3. Tulemuste järeldused 52
5. Kokkuvõte 54

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks 59

Jooniste loetelu

Joonis 1. Transformer-mudeli arhitektuur [20] 21
Joonis 2. KeyBERT-i töövoog 30
Joonis 3. KeyBERT-i filtrite seadistus 31
Joonis 4. Vektorifiltri seadistus 31
Joonis 5. Sõnade arv valitud transkribeeritud tekstides 32
Joonis 6. Meetodite võrdlus teksti sõnakuju ja märgendusandmete põhjal 35
Joonis 7. TextRanki skript 36
Joonis 8. Simple maths'i skripti osa 37
Joonis 9. Võtmesõnaotsingu variandid KeyBERT-is 38
Joonis 10. KeyBERT-i seadistused ja filtrid 39
Joonis 11. KeyBERT-i andmestiku töötlus võtmefraaside asjakohasuse hindamiseks .. 41
Joonis 12. KeyBERT-i võtmesõnade ja -fraasidega kattuvate märgendatud võtmesõnade osakaalu leidmine 42

Tabelite loetelu

Tabel 1. Keelemudelite kirjeldus 27
Tabel 2. Kolme ekstraheerimismeetodi keskmised F1-skoorid 43
Tabel 3. Parima keskmise F1-skooriga mudelid koos mitmekesisuse faktoriga 44
Tabel 4. Parima keskmise F1-skooriga mitmekesisused 45
Tabel 5. Parima keskmise F1-skooriga keelemudelid 45
Tabel 6. Märgendatud sõnade keskmine esindatus 200, 50 ja 10 võtmesõnakandidaadi seas 46
Tabel 7. KeyBERT-i parimad märgendatud võtmesõnade esindatuse osakaalud 47
Tabel 8. KeyBERT-i parimad mudelid, mille tulemused kattusid märgendusega kõige rohkem 48
Tabel 9. KeyBERT-i parimad mitmekesisuse seadistused 49
Tabel 10. Parima keskmise kattuvusega mitmekesisuse faktorid kõiki mudeleid arvestades 50
Tabel 11. Parima keskmise kattuvusega mudelid kõiki mitmekesisuse faktoreid arvestades 51
Tabel 12. Parimad mudeli ja mitmekesisuse faktori kombinatsioonid kõikide n-grammide ja sõnakujude lõikes 52

1. Sissejuhatus

Suur osa digitaalselt talletatud andmestikest koosneb tekstiandmetest. Üks lahendus andmetest ülevaate saamiseks on võtmesõnade automaatne ekstraheerimine. Võtmesõnade kasutamine on kasulik muuhulgas nii klientide tagasiside põhjal toodete analüüsimiseks kui ka meditsiinis, võimaldades kiiremini tuvastada olulisi sümptomeid, aidates säästa ajalist ressurssi ning luua kokkuvõtlikuma ülevaate kriitilisematest kommentaaridest [1].

Võtmesõnade ekstraheerimise meetodite laienemisega paralleelselt on viimase kümnendi jooksul laienenud ka keelemudelite kasutusalad ning on hakatud kasutama võtmesõnade leidmiseks. Loomuliku keele töötluses (ingl *natural language processing*) kasutatakse transformer arhitektuuri keelemudeleid peamiselt keele analüüsimiseks, tuues välja semantilise ja süntaktilise sarnasusi tekstides [2].

Tänu keelemudelite arengule on tänapäeval kättesaadavad mitte üksnes ingliskeelsed mudelid, vaid ka mitmekeelsed, sealhulgas eesti keelele kohandatud mudelid nagu EstBERT ja Est-RoBERTa. Eraldi võtmesõnade funktsionaalsust ei ole mudelitesse integreeritud, mistõttu on keelemudelite populaarsuse kasvuga seotud arenevaks suunaks võtmesõnade ekstraheerimiseks keelemudelite rakendamine.

Seni on olemas Maarten Grootendorsti loodud avatud lähtekoodiga võtmesõnaekstraheerija KeyBERT, mis kasutab keelemudelite vektoresitust võtmesõnade ekstraheerimiseks [3]. Hiljuti avaldatud teaduslikus artiklis võrreldi KeyBERT-i 11 eri ekstraheerimismeetodiga, kus võtmesõsande ekstraheerimise täpsust võrreldi mahuka ingliskeelsete teadusartiklite kogu peal [4]. Võrdluses saavutas KeyBERT keskmise F1-skoori alusel kõige paremaid tulemusi, millest on tingitud motiiv uurida KeyBERT-i kasutust eestikeelsete tekstidega.

1.1. Probleem

Seni pole teada, millised keelemudelid sobivad kõige paremini KeyBERT-i kasutamiseks eesti keele tekstidega. Kuigi Hugging Face'i keskkonnas on 479 eesti keelt toetavat keelemudelit, puuduvad teadaolevad uuringud, mis hindaksid KeyBERT-i täpsust eesti keele puhul. Samuti ei ole selge, milline peaks olema KeyBERT-i seadistus, et see töötaks efektiivselt eesti keelt toetavate keelemudelitega.

1.2. Käsitlusala ja eesmärk

Magistritöö on seotud Eesti Rahvusringhäälingu (ERR) arhiivi semantilise otsingu väljatöötamisega, millega tegeleb Tallinna Ülikooli Balti filmi, meedia ja kunstide instituudi ning digitehnoloogiate instituudi ühine töörühm[1].

Võtmesõnade tuvastus võimaldab määrata ajakirjandustekstide teemasid, nende muutumist ajas ja seoseid erinevate nimeolemitega (avaliku elu tegelased, organisatsioonid ja tegevuskohad). Nii on töö tulemusena parimaks osutunud meetodit kavas edaspidi rakendada ERR-i arhiivipäringu prototüübis.

Magistritöö peamine **eesmärk** on selgitada välja, kui tõhusalt on võimalik leida eestikeelsetest tekstidest võtmesõnu, kasutades selleks kohandatud eesti keelt toetavaid keelemudeleid ja KeyBERT-i teeki.

Magistritöö alaeesmärgid keskenduvad järgmistele aspektidele: 1) teha kindlaks, millised KeyBERT-i konfiguratsioonid on optimaalsed võtmesõnade tuvastamiseks eesti keelt toetavate keelemudelitega, mis on valitud enim kasuatud keelemudelite hulgast[2]; 2) võrrelda KeyBERT-i genereeritud tulemuste täpsust juba kasutusel olevate meetoditega, statistikal põhineva *simple maths*'i ja graafil põhineva TextRankiga.

Meetodite tulemuslikkust hinnatakse ERR-i raadioarhiivi transkribeeritud tekstide alusel võrdluses filoloogide märgendatud võtmesõnadega ning lähtudes iga teksti F1-skooride

[1] Tööd on rahastatud HTM-i programmist *Eesti keel ja kultuur digiajastul* (projekt *EKKD119 Automaatse keeletöötluse rakendamine ERRi arhiivis: olemitevaheliste seoste mudeldamine linkandmetel põhinevas arhiivipäringus*)

[2] https://huggingface.co/models?library=transformers&language=et&sort=trending

aritmeetilise keskmisest, mis arvestab automaatse võtmesõnatuvastuse saagist ja täpsust. Magistritöö eksperimentaalne osa keskendub ühesõnalistele võtmesõnadele. Kuna KeyBERT võimaldab aga leida ka võtmefraase, siis on autor lisaks vaadelnud, kui sageli sisaldavad KeyBERT-iga eraldatud kahe- ja kolmesõnalised võtmefraasid inimmärgendatud võtmesõnu.

1.3. Tööprotsess

Magistritöö praktilises osas rakendatakse KeyBERT-i ja erinevate keelemudelite kombinatsioone võtmesõnade leidmiseks 180 ERR-i raadioarhiivi kultuurisaate transkriptsioonist. Nendes tekstides on käsitsi võtmesõnu märgendanud kaks Tallinna Ülikooli filoloogiaharidusega spetsialisti. Võrdlemiseks on valitud üheksa keelemudelit. Kasutatud on nelja mitmekeelset lause-transformerit LaBSE, e5, MiniLM ja MPNet ning viit sõnavektoritel põhinevat transformer-tüüpi mudelit, sealhulgas kolm mitmekeelset mudelit mBERT, XLM-RoBERTa ja DistilBERT ning kaks ainult eesti keele jaoks kohandatud mudelit EstBERT ja Est-RoBERTa.

Töö eesmärkide saavutamiseks võrreldakse KeyBERT-i leitud võtmesõnu filoloogide märgitud võtmesõnadega ning analüüsitakse tulemusi võrdluses kahe varasema võtmesõnade leidmise meetodiga *simple maths*, mis on osutunud korpuslingvistikas kasutatavate statistiliste meetodite hulgas üheks tõhusamaks, ja TextRank, mis on laialdaselt kasutatud graafipõhine võtmesõnade ekstraheerimise meetod [5][6].

1.4. Töö ülevaade

Magistritöö on struktureeritud kolme peatükki. Teine peatükk “Teoreetiline raamistik” annab ülevaate võtmesõnade ekstraheerimise protsessidest statistiliste meetoditega nagu *simple maths* ja TextRank. Lisaks tutvustatakse transformer-tüüpi keelemudeli üldist arhitektuuri ja selle kodeerija struktuuri. Käsitletakse töös kasutatud keelemudeleid ja kirjeldatakse KeyBERT-i võtmesõnaotsingu meetodit. Kolmas peatükk “Andmestik ja meetodite rakendamine” kirjeldab ERR-i transkribeeritud tekstide kasutamist võtmesõnade ekstraheerimiseks ja märgendajate koostatud võtmesõnade andmestikku. Peatükk lõpeb võrdlusaluseksolevate meetodite *simple maths*, TextRank ja KeyBERT rakendamisega ERR-i tekstidel ning puudutab selle tulemusena saadud võtmesõnade

andmestiku haldamist. Neljandas peatükis keskendutakse võtmesõnade ekstraheerimise tulemustele ja nende tõlgendamisele. Võrreldakse KeyBERT-i ja erinevate keelemudelite kombineerimisel ning statistiliste meetoditega leitud võtmesõnade kattuvust inimmärgendajate valitud võtmesõnadega. Kirjeldatakse võtmefraaside oletatavat asjakohasust ühesõnaliste võtmesõnadega kattumise alusel. Järelduste osas antakse hinnang võrreldud meetodite sobivusele eestikeelsete tekstide võtmesõnastamisel ning tehakse ettepanekud kindla lahenduse kasutamiseks ja tulemuste täpsustamiseks järeltöötluse abil.

2. Teoreetiline raamistik

Siinses peatükis käsitletakse mõisteid ja meetodeid, mida kasutatakse edaspidi ka töö praktilises osas. Eristatud on mõisteid ja meetodeid, mis on seotud võtmesõnadega ning transformer-arhitektuuril põhinevate keelemudelite ja KeyBERT-iga.

Peatükk määratleb võtmesõnade mõiste ning tutvustab meetodeid, mida on nende tuvastamiseks seni kasutatud ja mida kasutatakse ka käesolevas magistritöös. Edaspidi kirjeldatakse süvaõppes kasutatavat transformer-arhitektuuri koos selle alusel treenitud keelemudelitega ja KeyBERT-i ülesehitust. Transformereid kirjeldav osa toetub Hugging Face'i keeletehnoloogiakursuse õppedokumentatsioonile [7] ja KeyBERT-i kasutusjuhendile [8]. Välja tuuakse transformer-mudeli ja lause-transformeri üldine ülesehitus, keskendudes kodeerimise funktsionaalsusele. Tutvustatakse eesti keelel töötavaid keelemudeleid, mis on valitud KeyBERT-iga kasutamise jaoks, pidades silmas Hugging Face'i trende.

2.1. Võtmesõnade ekstraheerimine

Võtmesõnad on terminid, mida kasutatakse teksti või tekstikogumi iseloomustamiseks, pakkudes olulist teavet selle sisu kohta [9]. Neid võib määrata kas kvalitatiivselt, tuginedes subjektiivsele hinnangule, või kvantitatiivselt, kasutades mitmesuguseid statistilisi meetodeid. Võtmesõnade ekstraheerimine on protsess, kus eraldatakse tekstis leiduvad sõnad või fraasid, mis esindavad teksti sisu olulist aspekti [10]. Eesmärk on tuvastada need sõnad või fraasid, mis annavad kiire ülevaate peamistest teemadest või mõistetest. Võtmesõnade ekstraheerimine saab toimuda nii käsitsi, kui tekstile määrab võtmesõnad inimene, kui ka automaatselt. Tihti neid võimalusi kombineeritakse ning leitakse automaatselt tuvastatud võtmesõnade kattuvus inimese märgendatud sõnadega [11]. Antud lähenemist nimetatakse ka kuldstandardiks (ingl *gold standard*) automaatse võtmesõnaotsingu täpsuse hindamisel [12].

Võtmesõnade ekstraheerimise meetodid võib jaotada juhendatud (ingl *supervised*) ning juhendamata (ingl *unsupervised*) õppe meetoditeks. Juhendatud võtmesõnatuvastus tugineb märgendatud andmestike kasutamisele, kus võtmesõnad on eelnevalt inimeste poolt tekstides identifitseeritud. Selle süsteemi aluseks on õpitud mudel, mis on võimeline

tuvastama võtmesõnu uutes tekstides, lähtudes varasematest näidetest [12]. Mudelit treenitakse võtmesõna tunnuseid ära tundma ning neile toetudes ennustama potentsiaalseid võtmesõnu. Juhendatud meetodite hulka kuuluvad nii-nimetud traditsioonilised ja närvivõrgupõhised meetodid [13].

Traditsioonilised juhendatud võtmesõnade ekstraheerimise tehnikad hõlmavad erinevaid statistilisi meetodeid, mis põhinevad sõnade sagedusel, asukohal ja koosesinemisel tekstis. Need meetodid kasutavad sageli lineaarseid klassifikaatoreid nagu logistiline regressioon, mis on treenitud ära tundma, kas sõna on võtmesõna või mitte. Neid tehnikaid iseloomustab sõltuvus kvaliteetsetest treeningandmetest ning võimekus kohanduda konkreetse kasutusjuhtumiga. Saadud tulemused sõltuvad otseselt õppemudeli kvaliteedist ning sellest, kui täpselt õppeandmestik tegelikke kasutusjuhtumeid kajastab. [14].

Närvivõrgu põhistest meetoditest kasutatakse süvaõpe mudeleid, nagu CNN ehk konvolutsioonilised närvivõrgud (ingl *convolutional neural network*) ja RNN ehk korduvad närvivõrgud (ingl *recurrent neural network*). Närvivõrgupõhised mudelid õpivad tuvastama võtmesõnu, identifitseerides tekstis olulised mustrid ja seosed, mis aitavad eristada olulist informatsiooni mürast. Treenimisprotsessi käigus optimeeritakse mudeli kaalud selliselt, et võimalikult täpselt ennustada etteantud õppeandmete põhjal võtmesõnu, kasutades selleks suurt hulka näiteid (näiteks märgendatud tekstid, kus võtmesõnad on eelnevalt määratud) [15].

Vastupidiselt sellele ei põhine juhendamata võtmesõnade ekstraheerimine eelmärgendatud andmetel. See lähenemine kasutab statistilisi, graafi-, sõnavektori- ja keelemudelipõhiseid meetodeid. Statistilised juhendamata meetodid võtmesõnade tuvastamiseks toetuvad mõõdikutele nagu sõna suhteline sagedus tekstis (ingl *term frequency* – TF) ja sõna tüüpilisus teistes tekstides (ingl *inverse document frequency* – IDF). Need põhinevad eeldusel, et olulisemad sõnad ilmnevad tekstis sagedamini, kuid mitte liiga paljudes teistes dokumentides, mis samasse korpusesse kuuluvad. See lihtne lähenemine sobib hästi suurte andmekogumitega töötamisel, kuid võib mõnikord esile tõsta vähem olulisi või juhuslikke sõnu [16]. Juhendamata statistiliste meetodite hulka kuuluvad ka meetodid, mis kasutavad tekstis või keelekorpuses võtmesõnade tuvastamiseks eraldi võrdluskorpust. Näiteks rakendatakse lihtsat sõnasageduste

suhtarvu, log-tõepära funktsiooni ja hii-ruut testi, mis kõik põhinevad sõna suhtelisel sagedusel vaadeldavas tekstis ja võrdluskorpuses.

Graafipõhised tehnikad nagu TextRank loovad sõnadest graafilise esituse, kus sõnad on sõlmed ja nendevahelised semantilised või leksikaalsed seosed on servad. Algoritmid nagu PageRank või selle variandid arvutavad iga sõlme tähtsuse, määrates kõrgema skoori kesksematele ja mõjukamatele sõnadele. Graafipõhised meetodid on kasulikud, kuna need on võimelised tuvastama tekstis keskseid teemasid [5].

Sõnavektorite (ingl *embedding*) algoritmid nagu Word2Vec, FastText või GloVe pakuvad rikkalikku semantilist informatsiooni, kuna neil põhinevad mudelid õpivad sõnade tähendust kontekstist. Sõnavektoreid kasutades saab luua ruumilisi esitusi, kus semantiliselt sarnased sõnad asetsevad üksteisele lähedal. Need meetodid võimaldavad avastada peidetud tähendusseoseid ja tuvastada võtmesõnu, mis esindavad teksti keskset sisu efektiivsemalt kui lihtsad sagedusel põhinevad lähenemised [17].

Eeltreenitud keelemudelite, nagu BERT ja selle edasiarendused, kasutamine võtmesõnade tuvastamisel toob kaasa võime mõista lausete ja lõikude sügavamat tähendust. Näiteks KeyBERT-i lähenemine kasutab BERT-i semantilisi võimeid, et tuvastada konteksti põhjal olulised võtmesõnad, lähtudes loodud sõnavektorite ja dokumendivektorite vahelise seose arvutamisel [18].

2.1.1. Statistik *simple maths*

Korpuslingvistika meetoditest on valitud KeyBERT-iga võrdlemiseks meetod *simple maths*, mis on andnud autori eelnevas uuringus võtmesõnade ekstraheerimisel võrdlemisi täpseid tulemusi. Võrdluses teiste meetoditega, nagu hii-ruut test ja log-suhe (ingl *log-ratio*), ei olnud *simple maths* nii tundlik kirjaviga sisaldavate, valesti lemmatiseeritud või muul põhjusel võrdlustekstidest puuduvate sõnade suhtes, tuues neid olulisemate võtmesõnade hulgas vähem esile [6]. *Simple maths* on statistiline mõõdik, mida kasutatakse Sketch Engine'i platvormil võtmesõnade leidmiseks kahe erineva korpuse võrdluses [19]. Selle meetodi kasutamiseks on vajalik sihtkorpuse (ingl *focus corpus*) ja võrdluskorpuse (ingl *reference corpus*) olemasolu. *Simple maths* on juhendamata meetod, mis arvestab sõnade esinemissagedust miljoni sõna kohta kahes nimetatud korpuses. See arvutatakse valemiga:

$$SM = \sum \frac{fpm_{focus} + n}{fpm_{ref} + n},$$

kus fpm_{focus} tähendab sõna suhtelist sagedust fookuskorpuses, fpm_{ref} viitab analoogselt sõna suhtelisele sagedusele võrdluskorpuses ning n on konstant, mis võimaldab välistada nulliga jagamise, kui sõna võrdluskorpusest puudub [20].

Sõna esinemissagedus sihtkorpuses arvutatakse järgmiselt:

$$fpm_{focus} = (a \cdot 1000000) \div (a + c),$$

kus a on otsitava sõna absoluutsagedus sihtkorpuses ning c on kõigi ülejäänud sõnade sageduste summa. Analoogselt sihtkorpusega on võrdluskorpuse jaoks valem:

$$fpm_{ref} = (b \cdot 1000000) \div (b + d),$$

kus b on otsitava sõna absoluutsagedus võrdluskorpuses ning d tähistab kõigi ülejäänud sõnade summaarset sagedust [21].

Oluline on märkida, et tähised a ja b viitavad samale sõnale siht- ja võrdluskorpuses. *Simple maths*'i kasutamine eeldab kasutajalt võrdluskorpuse olemasolu, magistritöös kasutatakse 2021. aasta eesti keele ühendkorpust [22].

2.1.2. TextRank

TextRank on juhendamata võtmesõnade ekstraheerimise meetod, mis kuulub graafipõhiste meetodite hulka [23]. See järjestusalgoritm on inspireeritud Google'i PageRanki algoritmi põhimõtetest, mis määravad veebilehtede tähtsuse nende omavaheliste linkide alusel [24]. TextRanki rakendusviis seisneb tekstist graafi loomises, kus tipudena on esindatud sõnad või fraasid ning servadena nende vahelised seosed [25].

TextRank kasutab iteratiivset protsessi, kus iga sõlme skoor arvutatakse uuesti, lähtudes tema naabrite skooridest. Praktikas lähtutakse graafist $G = (V, E)$, kus V on sõlmpunktide sõnade hulk ja E on sõnadevahelisi seoseid väljendavate servade hulk. Iga serv (i, j) graafis esindab kahte sõna i ja j, mis esinevad tekstis üksteise läheduses. Seejärel

kaalutakse servi nende esinemissageduse põhjal tekstis. Mida sagedamini kaks sõna koos esinevad, seda tugevam on nendevaheline seos. Iteratsioonide käigus arvutatakse kaalutletud skoor igale sõlmele valemiga:

$$S(V_i) = (1 - d) + d \cdot \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j),$$

kus $S(V_i)$ tähistab lause skoori ja d on summutustegur, mis jääb vahemikku 0 kuni 1, et vältida skooride liigset kasvu. $In(V_i)$ on tippude hulk, mis viivad tippu i (sisenevad servad). $Out(Vj)$ on tippude hulk, kuhu tippu j viivad (väljuvad servad). Andmestikku töödeldakse seni, kuni skoorid jäävad alla etteantud lävendi või kuni on tehtud maksimaalne lubatud arv iteratsioone [23]. Graafi koostamisel on olemas akna parameeter, millega filtreeritakse sõnu vastavalt servade ja tippude ühendustele. Näiteks kui parameeter on 4, siis vaadatakse 3 sõna suhtes paremale ja vasakule jäävat sõna, mis on vahetult selle kõrval [25]. Meetodi abil saavad kõik sõnad vastava skoori ning andmestik reastatakse ümber kõrgema skoori alusel. Autor valis TextRanki korpuslingvistilise meetodi kõrvale teiseks võrdlusaluseks, kuna selle meetodi puhul ei sõltu kasutaja võrdluskorpuse olemasolust. Magistritöös on valitud akna parameetriks 2, ehk seoste arvutamisel arvestatakse ühe eelneva ja järgneva sõnaga.

2.2. Transformer-mudel

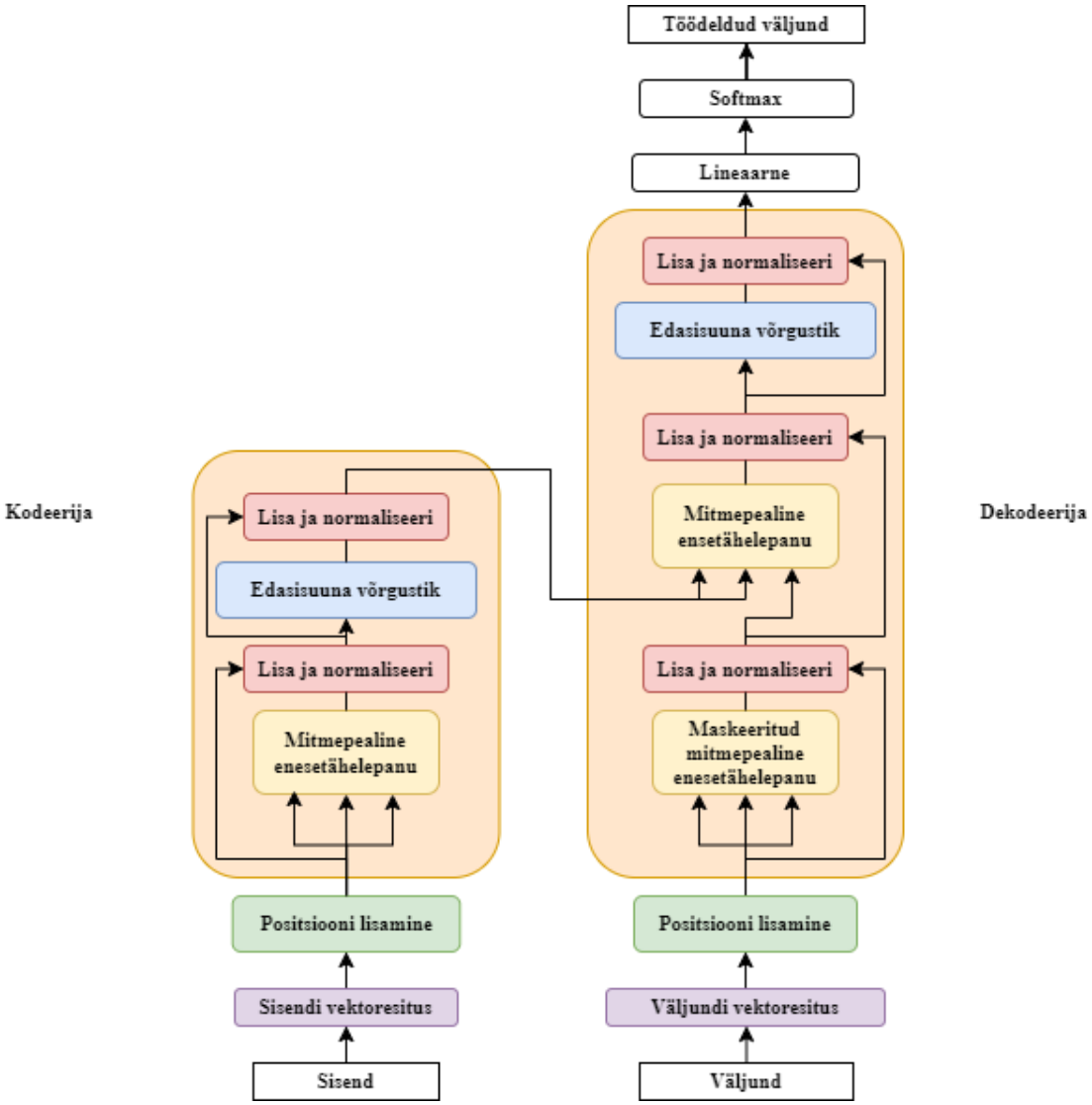
Transformer-mudeleid tutvustas esimesena Google Brain[3] meeskonna 2017. aasta artikkel "Attention Is All You Need". Need on süvaõpe arhitektuurid, mis loodi selleks, et lahendada CNN ja RNN mudelite arhitektuurilisi kitsaskohti [20]. CNN on tavaliselt kasutatud piltide ja visuaalse sisu töötlemisel tunnuste eraldamiseks ja klassifitseerimiseks, samas kui RNN sobib ajaliselt korreleerunud andmete, nagu tekst ja kõne, järjestikuseks analüüsimiseks ja modelleerimiseks. Transformerite peamine eelis on see, et need suudavad töödelda kogu sisendjada korraga, mitte ainult järjestikuselt nagu CNN ja RNN. Selline lähenemine võimaldab transformeritel tõhusamalt tuvastada pikki sõltuvusi tekstides, mis on üliloluline näiteks masintõlkes. Kuigi tänapäeval on transformerid suurte keelemudelite peamine alus, kasutatakse neid mudeleid ka pildi- ja helitöötluses [26].

[3] https://research.google.com/teams/brain/?ref=harveynick.com

Masintõlke valdkonnas on transformerid näidanud, et suudavad keerukaid grammatilisi struktuure ja idioome tõlkida paremini kui RNN-põhised mudelid, mis pikema teksti puhul võivad kaotada konteksti. Lisaks on transformerid osutunud tõhusamaks kui CNN teksti genereerimise ülesannetes, kuna need ei piirdu üksnes lokaalse teabe analüüsimisega, vaid suudavad arvestada laiemaid kontekstuaalseid mõjusid. Transformerite arendamine keskendus peamiselt keeletöötluse valdkonna tõhususe ja tulemuslikkuse parandamisele [20].

2.2.1. Transformer-mudeli üldine arhitektuur

Transformerid kasutavad sisendi töötlemisel mittejärjestikust lähenemist, rakendades enesetähelepanu mehhanismi (ingl *self-attention*). Need mehhanismid hindavad kõigi sisendite osade tähtsust üheaegselt, mis võimaldab iga osa paralleelset töötlemist treeningprotsessi ajal. Enesetähelepanu protsessis antakse igale sõnale tähelepanukiht, mis juhib fookust teatud sõnadele vastavalt mudeli õpitud struktuurile. Järgnevalt kirjeldatakse transformer-mudeli (Joonis 1) arhitektuuri põhikomponente: kodeerija ja dekodeerija.



Joonis 1. Transformer-mudeli arhitektuur [20]

Transformer-mudelitel võib olla kodeerija (ingl *encoder*), dekodeerija (ingl *decoder*) ja kodeerija-dekodeerija arhitektuur. Kodeerija tüüpi mudelid keskenduvad sisendi analüüsimisele ja semantiliste seoste tuvastamisele, dekodeerija keskendub rohkem väljundi töötlemisele, näiteks teksti genereerimisele ja lausete lõpetamisele, ning kodeerija-dekodeerija tüüpi arhitektuurid keskenduvad sisendi kodeeringu põhjal väljundi väljatöötamisele, näiteks tõlkimise ülesannetele.

Kodeerija töö algab sisendi teksti sõnestamisega ning sõnede vektoresitusega, mis muudab sisendi esmalt fikseeritud suurusega vektoriteks, milles rakendatakse igale sisendelemendile positsioonilist kodeeringut. Kodeerija kiht (ingl *layer*) sisaldab mitmeid alamkihte, sealhulgas enesetähelepanu mehhanismi alusel mitmepealist

tähelepanumehhanismi (ingl *multihead attention*), mis võimaldab mudelil paralleelselt keskenduda sisendite erinevatele osadele. See aitab mõista sõnadevahelisi seoseid sügavamalt. Peale tähelepanu kihi lisamist järgneb lisamise ja normaliseerimise etapp (indl *Add & Norm*), mis ühendab tähelepanu kihi algse sisendiga, millele järgneb kogu andmestiku kihi normaliseerimine. Lõpuks on positsioonipõhine pärisuuna võrgu kiht (ingl *Feed Forward Network*) – ühendatud närvivõrk, mis kohandab eelmise kihi mudeli poolt tuvastatud keerukate seostega, mida lisatakse eelnevalt saadud kihile juurde ning uuesti normaliseeritakse. Kokkuvõtteks saab sisendtekst mitmekihilise semantilise kirjelduse [20].

Dekodeerija erinevus seisneb selles, et sellega luuakse väljundeid. Näiteks, lähtudes lause lõpu genereerimise protsessist, on lause sisend sõnestatud ning sõneloendile on lisatud lause algust tähistav sõne. Saadud sõned viiakse vektoresitusele ning lisatakse positsioonilised kodeeringud. Järgnevalt algab maskeeritud mitmepealine tähelepanu (ingl *Masked Multihead Attention*), mille eesmärgiks on tagada mudeli keskendumine tekstis ainult neile sõnadele, mis on antud sisendina. See protsess takistab mudelil näha või kaaluda sõnu, mis pole veel tekstist sisse loetud. Sellist tehnikat kasutatakse näiteks lausete genereerimisel, kus iga uus sõna peab põhinema ainult eelnevalt loodud tekstil, mitte tulevasel tekstil, mida mudel ei tohiks veel teada. See aitab luua järjepidevaid ja loogilisi lauseid, vältides olukorda, kus mudel kasutab informatsiooni, mida tegelikus suhtlusolukorras veel olemas ei oleks. Sellele järgnevad mitmepealine tähelepanu ja pärisuuna võrk koos nendevahelise lisamise ja normaliseerimise etapiga. Viimaks läbib andmevoog lineaarse ja *softmax*-kihi, mis teisendavad dekodeerija viimase kihi väljundi lõplikuks sõnaks, mida esitatakse töödeldud väljundina. Kuna dekodeerija on autoregressiivne ehk protsessis luuakse üks sõna korraga, siis korratakse antud töövoogu alates sõnestamisest nii kaua, kuni sõnade jada on valmis ja lisatud on lause lõppu tähistav sõne.

Kodeerija ja dekodeerija töötavad koos, et teksti kodeerida vektoresituseks ja seejärel dekodeerida see tagasi tekstiks. Kodeerija ülesanne on teksti teisendamine vektoresituse kujule, mis sisaldab konteksti ja semantilist informatsiooni. Dekodeerija alustab protsessi, lisades esmalt lause algust tähistava sõne ning kasutades seejärel mitmepealist tähelepanu, et töödelda ja lisada kodeerijalt saadud vektorkihid ning genereerida järgmine

sõna vastavalt vastu võetud informatsioonile. See protsess jätkub, kuni kõik sõnad on kodeeritud, ning lõpuks lisatakse ka lauselõpu sõne [20].

Transformerite mudeli kodeerija ja dekodeerija keskmes on tähelepanu mehhanism, mis võimaldab mudelil hinnata sõnadevahelisi seoseid, arvestades kogu sisendjada konteksti. See genereerib igale sõnale kolm põhivektorit: päringu, võtme ja väärtuse vektori. Tähelepanu kaalud arvutatakse päringu ja võtme vektorite skalaarkorrutiste abil ja normaliseeritakse *softmax*-funktsiooniga, mis tagab, et väljundis oleksid olulisemad sõnad.

Mitmepealine enesetähelepanu, mis jagab tähelepanu mitmeks peaks, suurendab mudeli võimet paralleelselt hinnata sisendi erinevaid aspekte ja õppida keerukaid sõltuvusi. Sellisel transformer-struktuuril põhinevad paljud keelemudelid, sealhulgas mudelid, mis teostavad kodeerimist, dekodeerimist või mõlemat. Näiteks BERT ja selle derivaadid nagu mBERT, XLM-RoBERTa ja LaBSE kuuluvad kodeerimisfunktsionaalsusega transformer-mudelite hulka, mida käesolev magistritöö käsitleb KeyBERT-i katsetes.

2.2.2. Transformer-mudeli kodeerija

KeyBERT-i kasutamise eelduseks on kodeerija tüüpi mudelid, mis analüüsivad sisendandmestiku. Kodeerimisprotsessi paremaks mõistmiseks võtame näiteks lause: *Mina olen sportlane*. Eeltöötluse käigus jaotatakse lause sõnad eraldi sõnedeks (ingl *token*): ["$Mina$","$olen$","$sportlane$"] [27].

Seejärel leitakse iga sõna koordinadid keelemudeli sõnavarast, näiteks võivad indeksid olla [001,010,100]. Need indeksid konverteeritakse seejärel maatriksiks X, kus iga indeks vastab kindlale koordinadile vektoresitusele. Erinevatel keelemudelitel võib vektorite suurus olla erinev, näiteks BERT-mudelil on see 768. See tähendab, et ühe sõna asukoha kirjeldamiseks kasutatakse 768 dimensiooni, mis on tuletatud treeningandmestiku põhjal [28]. Iga vektoresitus sisaldab informatsiooni sõna kontekstuaalse tähenduse ja sõnadevaheliste seoste kohta.

Järgnevalt lisatakse iga sõna vektorile positsiooniline kodeering (ingl *positional embedding*), mis sõltub sõna asukohast lauses. See saadakse siinus- ja koosinuskõverate kombinatsioonil põhinevate valemitega:

$$PE_{(pos,2i)} = sin\left(pos \div 10000^{2i \div d_{model}}\right),$$

$$PE_{(pos,2i+1)} = cos(pos \div 10000^{2i \div d_{model}}),$$

kus pos on igale sõnale lauses määratud asukoht, indeks i tähistab dimensiooni vektoris ning d_{model} mudeli dimensioonide arvu, mis on olenevalt keelemudelist erinev. Näiteks BERT-mudelitel on 512 dimensiooni ehk nii palju sõnu saab korraga töödelda. Indeksitel $2i$ ja $2i + 1$ rakendatakse vastavalt siinuse ja koosinuse funktsioone. Siinuse ja koosinuse vaheldumine aitab hoida algsete sõnade positsioonilise informatsiooni ühtlaselt jaotatuna kogu vektori ulatuses, vältides teabe kontsentreerumist mõnesse kindlasse dimensiooni ning tagades, et informatsioon on mudeli töötlemise igal tasandil kättesaadav [20].

Järgnevalt leitakse mitmepealine tähelepanu. Esmalt luuakse kolm õpitavat kaalumaatriksit (ingl *Weight*) W^Q päringute (ingl *Query*), W^K võtmete (ingl *Key*) ning W^V väärtuste (ingl *Value*) jaoks. Maatriksid on genereeritud juhuslikult ning alguspunktiks treenimisprotsessile.

Protsessi nimetatakse mitmepealiseks, kuna see hõlmab mitme enesetähelepanu mehhanismi paralleelset töötlemist. Kaalumaatriksite rakendamisel transformeeritakse sisendmaatriks X kolmeks maatriksiks:

$$Q = X \cdot W^Q,$$

kus Q tähistab päringu maatriksit, X on sisendandmestiku maatriks ning W^Q tähendab kaalutud päringu maatriksit.

$$K = X \cdot W^K,$$

kus K tähendab võtmete maatriksit, X on sisendandmestiku maatriks ning W^K tähendab kaalutud võtmete maatriksit.

$$V = X \cdot W^V$$

kus V tähendab võtmete maatriksit, X on sisendandmestiku maatriks ning W^V tähendab kaalutud väärtuste maatriksit [20]. Kus Q, K, V on uued sõnade kirjeldavad maatriksid, mis esindavad vastavalt päringuid, võtmeid ja väärtusi iga tähelepanupea jaoks. Andmestikust saadud maatriksite põhjal arvutatakse skaleeritud tähelepanuskoor valemiga:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V,$$

kus *softmax* on skooride normaliseerimise meetod, d_k on võtmete dimensiooni väärtus ning K^T on transponeeritud võtmete maatriks, kus T tähistab transponeerimist (ingl *transponate*). Antud jagamise valem aitab normaliseerida algsete maatriksite tähelepanuskoori, vähendades seeläbi suuremate dimensioonide liigset mõju ja säilitades andmete stabiilsuse. Selline lähenemine on oluline, et tagada tähelepanumehhanismi täpsus suurtes andmemahtudes, kus võtmete hulk ja dimensioonid võivad varieeruda [20].

Saadud erinevad tähelepanupead ühendatakse üheks tähelepanu kihiks, kus iga pea väljundvektorid liidetakse suureks maatriksiks valemiga:

$$Y = Concat(head_1, \ldots, head_h) \cdot W^O,$$

kus $Concat$ viitab peade liitmisele, $head$ tähistab eelnevalt mainitud päringumaatriksi väljundvektoreid ning kaalumatriksis W^O tähistab O operatsiooni astet (ingl *operation*). Siinjuures on W^O juhuslik õpitav parameeter, mille eesmärk on kohandada ja optimeerida kombineeritud andmestikku ning treenida mudeli võrku. Mudeli iga pea i väljund on sellesse protsessi integreeritud, tagades, et kõikide peade informatsioon panustab ühtlaselt lõppväljundi kihti Y .

Algse sisendi X ja tähelepanu kihi väljundi Y summana saadakse jääkühendus X' (ingl *residual connection*), mis aitab säilitada varasemat informatsiooni õppimisprotsessis [20]. Peale jääkühenduse saamist läbib kiht normaliseerimise. Kihtnormaliseerimiseks (ingl *layer normalization*) kasutatakse järgmist valemit:

$$LayerNorm(X') = \gamma(\frac{X' - \mu}{\sigma}) + \beta,$$

kus μ ja σ on keskmine (ingl *mean*) ja standardhälve (ingl *standard deviation*), mis on arvutatud vektori X' komponentidest, ning γ nihe (ingl *bias*) ja β skaala (ingl *scale*) on keelemudelites määratud õpitavad parameetrid, mida kasutatakse normaliseerimiseks [20]. Igal keelemudelil on enda erinev nihe ja skaala väärtus olenevalt keelemudeli treeningandmestikust. Selle eesmärgiks on parandada mudeli kohandumist sisseloetud andmetega, kui need treeningmaterjaliga sarnanevad [29].

Peale kihi normaliseerimist läbitakse positsioonipõhised pärisuuna võrgud (FNN, ingl *position-wise feed-forward Networks*) mis, lisavad mudelile arvutuslikku keerukust. Sellest tingitult ei õpi mudel lihtsalt lineaarsest andmestikust, vaid uuritakse ka komplekssemaid sõnadevahelisi seoseid. Transformeri FFN struktuur koosneb kahest lineaarsest kihist, mille vahel on aktiveerimisfunktsioon ReLU (ingl *Rectified Linear Unit*) . FFN rakendab järgnevat valemit:

$$FFN(X') = max(0, X'W_1 + b_1)W_2 + b_2,$$

kus 0 on ReLu aktiiveerimisfunktsioon, W on kaalumaatriks ja b tähistab nihkeid. Esimene kiht, kasutades kaalumaatriksit W_1 ja vahendavat vektorit b_1, rakendab sisendile lineaarset transformatsiooni. Seejärel rakendatakse ReLU funktsiooni, mis elimineerib kõik negatiivsed väärtused, aidates seeläbi kaasa mudeli robustsusele ja vähendades võimalikku üleõppimist. Teine kiht, kasutades samuti kaalumaatriksit W_2 ja vahendavat vektorit b_2, teostab teise lineaarse transformatsiooni ning genereerib lõpliku väljundi iga positsiooni jaoks. See väljund on keerukam ja rikkalikum kui algne sisend X, mis aitab mudelil keelelisest kontekstist paremini aru saada ja sellele reageerida [20]. Pärisuuna võrguga luuakse jääkühendus eelneva normaliseeritud kihi maatriksiga X' ning normaliseeritakse uus jääkühenduse kiht. Antud protsessi lõpuks on kõigil algsetel sõnadel olemas 6 kirjeldavat kihti, mis iseloomustavad sõnade rolli lauses. Transformer-mudelite kodeerija kihtide arv võib olla ka suurem, näiteks BERT-i mudelil on 12 kihti.

2.3. Keelemudelid

Hugging Face'i[4] masinõppe andmebaasis on registreeritud 276 000 transformer-tüüpi keelemudelit, millest 479 on kohandatud eesti keele toetamiseks. Siinseks tööks on valitud üheksa mudelit, mis on arhitektuurilt kodeerija tüüpi keelemudelid (Tabel 1). Nendest neli on lause-transformerid, millele on algselt üles ehitatud ka KeyBERT. Sobivad eesti keelt toetavad keelemudelid filtreeriti välja Hugging Face'i keskkonna trendikuse hinnangute alusel, lähtudes ka KeyBERT-i autori soovitatud[5] MTEB[6] (*Massive Text Embedding Benchmark*) edetabelist.

Tabel 1. Keelemudelite kirjeldus

Mudel	Allikad	Versioon	Mudelitüüp	Treening andmestik	Parameetreid	Keelte arv
mBERT	[28]	bert-base-multilingual-cased	BERT	Wikipedia artiklid	179 mln	104
DistilBERT	[30]	distilbert-base-multilingual-cased	DistilBERT	Wikipedia artiklid	135 mln	104
EstBERT	[31]	EstBERT	BERT	Eesti ühendkorpus 2017	124 mln	1: eesti
XLM-RoBERTa	[32]	xlm-roberta-base	XLM-RoBERTa	Common Crawl tekstid	279 mln	94
Est-RoBERTa	[33]	est-roberta	CamemBErT	Ekspress Meedia artiklid	116 mln	1: eesti
mLaBSE	[34]	LaBSE	BERT	Wikipedia artiklid, CommonCrawl tekstid	471 mln	110
MiniLM	[35]	paraphrase-multilingual-MiniLM-L12-v2	BERT	Wikipedia artiklid	118 mln	50
MPNet	[35]	paraphrase-multilingual-mpnet-base-v2	XLM-RoBERTa	Wikipedia artiklid	278 mln	50
e5	[36]	multilingual-e5-large-instruct	XLM-RoBERTa	Wikipedia artiklid, Mitmekeelsed Common Crawl uudised, NLLB,Reddit,S2ORC,Stackexchange,xP3,SBERT andmestik	560 mln	94

[4] https://huggingface.co/models?library=transformers&language=et&sort=trending
[5] https://github.com/MaartenGr/KeyBERT/issues/220#issuecomment-2044935436
[6] https://huggingface.co/spaces/mteb/leaderboard

Valik hõlmab kolme trendipõhist mitmekeelset transformer-mudelit, kus eesti keel on toetatud: DistilBERT, BERT ja XLM-RoBERTa. Lisaks on kaasatud kaks vaid eesti keele jaoks kohandatud mudelit: EstBERT ja Est-RoBERTa. KeyBERT-i esmavaliku[7] alusel on valitud ka mitmekeelsed lause-transformerid (ingl *sentence transformers*): LaBSE, e5, MPNet ja MiniLM.

Lause-transformerid on spetsiaalselt kohandatud töötlema teksti lause ja lõigu tasemel erinevalt traditsioonilistest transformer-mudelitest, mis vajavad eraldi täiendavat töötlemist, nagu sõnavektorite kokkuliitmine, et luua terve dokumendi vektoresitus. Nii võimaldavad lause-transformerid luua sidusaid ja semantiliselt rikkamaid esitusi, mis on optimeeritud suuremate tekstiüksuste mõistmiseks. Selleks kasutatakse koolitusmeetodeid nagu *Siamese* ja *triplet-loss*, mis õpetavad mudelit tuvastama ja hindama tekstiüksuste vahelist semantilist sarnasust, tagades, et sarnased tekstid on esitatud vektorruumis lähedal [34].

Arhitektuuri alusel võib välja tuua neli peamist mudelitüüpi: BERT, DistilBERT, XLM-RoBERTa ja CamemBERT. Erinevalt traditsioonilisest transformer-arhitektuurist, mis kasutab nii kodeerijat kui ka dekodeerijat, hõlmab BERT (*Bidirectional Encoder Representations from Transformers*) ainult kodeerijat. BERT-i iseloomustab selle kahesuunaline (ingl *bidirectional*) sõnade esitusviis, mis eristab seda tavapärasest transformer-mudelist. Oluliseks tunnuseks on eeltreeningute andmestikud, mida kasutatakse mudeli eeltreenimisel. BERT-i mudelid põhinevad peamiselt kahel eeltreenimise meetodil: sõnade maskeerimine, mis hõlmab lünkteksti täitmist peidetud sõna ennustades, ja järgmise lause ennustamine (NPS, ingl *next sentence prediction*). DistilBERT, mille nimi tuleneb andmete destilleerimise protsessist, on väiksem BERT-põhine õpilasmudel (ingl *student model*), mis õpib suuremalt õpetajamudelilt (ingl *teacher model*) jäljendama selle väljundtõenäosusi, et säilitada tõhusust vähendatud parameetrite arvuga. Erinevalt BERT-ist, mis on 12-kihiline, koosneb DistilBERT kuuest kihist ega sisalda NSP komponenti. XLM-RoBERTa (*Extensible Markup Language Robustly Optimized BERT Pretraining Approach*) sarnaneb struktuurilt DistilBERTile, kuna ka see ei kasuta NSP komponenti. Erinevus seisneb dünaamilises maskeerimises, mis toimub nii eeltreenimisel kui ka maskeerimiskihtide treenimise ajal. CamemBERT,

[7] https://maartengr.github.io/KeyBERT/guides/embeddings.html

mis on nimetatud Camembert'i juustu järgi, on sarnane XLM-RoBERTaga, kuid erineb selle poolest, et algmudel on spetsiaalselt treenitud prantsuskeelsete tekstidega.

Üheks peamiseks eristavaks teguriks on mudelite parameetrite hulk, mis mõjutab, kuidas mudelid töötlevad tekstiandmeid vektorkihtide kaupa ning kuivõrd sügav ja laialdane on konteksti mõistmine. Parameetrite arv sõltub mudeli arhitektuurist, sealhulgas mudeli vektoresituse suurusest, positsiooniesitusest ja kihtide normaliseerimise kordustest. Üheks selliseks mudeliks on e5, milles on (eriti suur hulk parameetreid), ning märkimisväärne on ka treeningandmete kogus võrreldes teiste keelemudelitega, mis on põhiliselt treenitud tekstidega Wikipediast ja veebisorimise andmete repositooriumist Common Crawl.

Mudelite versioonipõhised erinevused ilmnevad peamiselt nende spetsiifilistes omadustes, nagu keeleline mitmekeelsus, suur- ja väiketähtede eristamine ning treeningjuhised. Mitmekeelsed mudelid toetavad mitme keele töötlemist, *cased*-mudelid teevad vahet suur- ja väiketähtedel andmestikus ning *instruct*-mudelid on spetsiaalselt treenitud juhiste ja ülesannete tõlgendamiseks.

2.4. KeyBERT

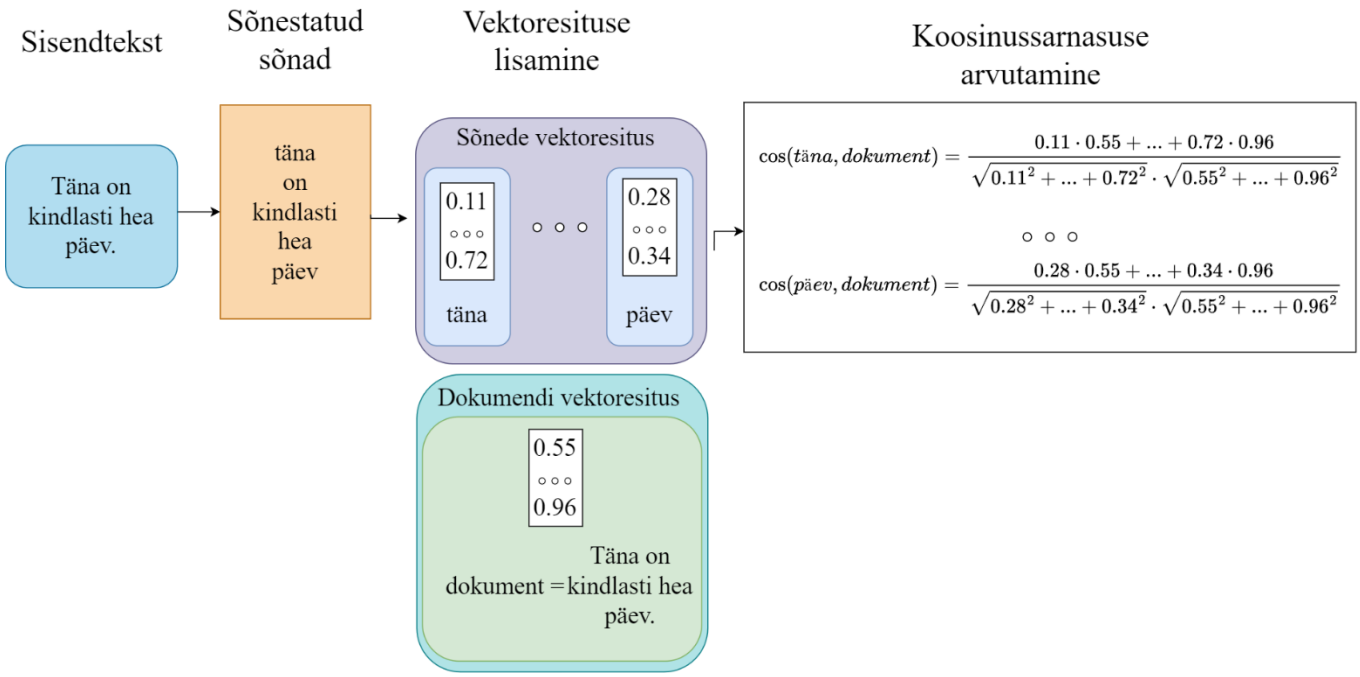
KeyBERT on Pythoni teek[8] juhendamata võtmesõnade ekstraheerimiseks, mille lõi 2020. aastal Maarten Grootendorst [3]. Teegi loomise peamine põhjus oli täita lünk võtmesõnade ekstraheerimise teekide vallas, keskendudes transformer-keelemudelite kasutamisele. Teegi erilisus seisneb minimaalsest meetodist võtmesõnade ekstraheerimiseks, mis kasutab vaid keelemudelitest saadud sõnede vektoresituste andmeid. KeyBERT põhineb koosinussarnasuse valemil:

$$cos(\theta) = \frac{A \cdot B}{\left||A|\right| \cdot \left||B|\right|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}},$$

kus A on sõne vektoresitus ja B on sõne kogu dokumendi vektoresitus.

[8] https://www.python.org/

Joonis 2 demonstreerib töövoogu, kui dokument koosneb lausest *Täna on kindlasti hea päev* Esmalt läbib lause sõnestamise. Sõnestatud andmestikule lisatakse vektoresitus, andes igale sõnele konteksti. Seejärel viiakse kogu dokument vektoresitusele, tuues esile dokumendi kontekst. Saadud sõne ja dokumendi vektoresituste põhjal arvutatakse koosinussarnasus, millega sõne saab võtmesusskoori.



Joonis 2. KeyBERT-i töövoog

Tehnilisest kirjeldusest töötab KeyBERT vastavalt määratud keelemudelile ning sellest sõltub suuresti võtmesõnatuvastuse täpsus ehk olenevalt keelest peab olema valitud ka sobiv keelt toetav mudel. Vaikimisi kasutatakse KeyBERT-is inglise keele jaoks mõeldud MiniLM[9] lause-transformerit. Teiste lause-transformerite kasutamiseks tuleb KeyBERT-is eelnevalt ära määrata keelemudel. Tavaliste transformerimudelite jaoks on vaja kasutada Flairi[10] teeki, mis võimaldab koostada KeyBERT-is vajamineva dokumendi vektoresituse.

Peale keelemudeli määramist on võimalik rakendada erinevaid võtmesõnaotsingu filtri seadeid, millega saab kohandada otsitavaid võtmesõnu (Joonis 3). Töö skoobi piiritlemiseks lähtuti põhiliselt hiljutise võtmesõnade ekstraheerimise võrdlusuuringu kohaselt MMR filtrist ehk maksimaalset marginaali olulisusest (ingl *maximal marginal*

[9] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[10] https://github.com/flairNLP/flair

relevance) [8]. Käesolevas lõputöös rakendatud filtrite hulgas on MMR filter use_mmr, mis on määratud *diversity* ehk mitmekesisuse faktori valitud väärtusega, mis jääb vahemikku 0 kuni 1. 0 tähendab, et leitud võtmesõnade omavaheline sarnasus on väga suur (madal mitmekesisus), ja 1 tähendab, et leitud võtmesõnad on üksteisest väga erinevad (kõrge mitmekesisus). Väljundi kandidaatide arvu määramiseks kasutati parameetreid nr_candidates ja top_n, mille põhjal saadi kätte 200 kõige suurema võtmesuseskooriga tulemust, ning sõne vektoresituse filtrit vectorizer, millega saab detailsemalt kohandada kasutatava sõnade valiku, mille hulgast otsitakse võtmesõnu.

```
keywords_batch = [
    model.extract_keywords(
        doc,
        use_mmr=True,
        diversity=diversity,
        vectorizer=vectorizer,
        nr_candidates=200,
        top_n=200,
    )
    for doc in batch_texts
]
```

Joonis 3. KeyBERT-i filtrite seadistus

Filtrile vectoriser (Joonis 4) lisati sõnede eraldamiseks filter tokenizer, millega saab kohandada seda, missuguseid sümboleid arvestada sõna osana. Näiteks loeti sõna koosseisu sidekriipsud nimedes nagu *Põhja-Ameerika*. Samuti saab lisada väljundile filtri ngram_range, kus n viitab sellele, kui mitmest sõnast võib väljund koosneda. Eksperimendi mõttes seadistati KeyBERT kuvama ühesõnalisi võtmesõnu (unigrammid) ning kahe- ja kolmesõnalisi võtmefraase (bigrammid, trigrammid). Viimase kahe filtrina lisatistoppsõnade ehk soovimatute sõnade eemaldamine ning ka võimalus määrata, kas sõnad on väiketähestatud.

```
vectorizer = CountVectorizer(
    tokenizer=custom_tokenizer,
    ngram_range=ngram_range,
    stop_words=stopwords,
    lowercase=lowercase,
)
```

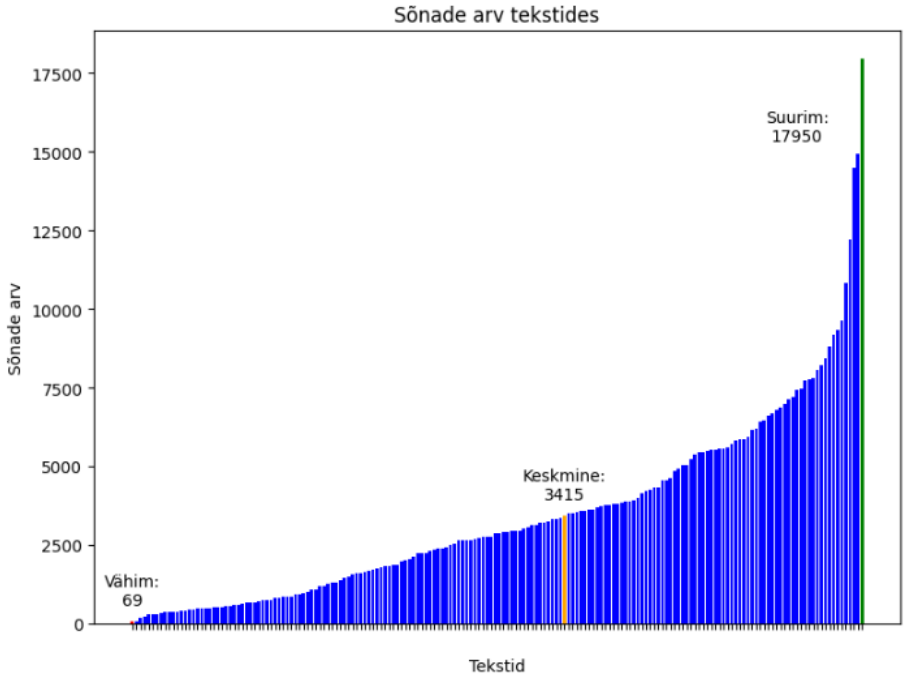
Joonis 4. Vektorifiltri seadistus

3. Andmestik ja meetodite rakendamine

Peatükis tutvustatakse andmestikku, millest võtmesõnu ekstraheeriti, ning kust pärinevad märgendatud võtmesõnadega failid. Seejärel selgitatakse võtmesõnade ekstraheerimiseks rakendatud kolme meetodi töövoogu. Lõpuks käsitletakse, kuidas oli lõppandmestik ettevalmistatud ning milliste meetoditega hinnatud.

3.1. Andmestik: transkribeeritud raadiosaated

KeyBERT-i katsetuste jaoks oli kasutusel Tallinna Ülikooli projekti EKKD119 uurimismaterjal: Eesti Rahvusringhäälingu 44 000 kultuuriteemalist raadiosaadet aastatest 2003 kuni 2021 [37]. Tekstid on automaatselt transkribeeritud TalTechis arendatud kõnetuvastuse tehnoloogiaga [38]. Iga saate tekst on talletatud eraldi tekstifailina. Katsetusteks valiti juhuslikult 180 faili 44 000 hulgast. Valitud tekstide keskmine sõnade arv oli 3415 kuni 4000, mis vastab umbes neljale A4-teksti leheküljele. Suurima faili sõnade arv ulatus aga 17 950-ni ja lühim transkriptsioon sisaldas vaid 69 sõna (Joonis 5).



Joonis 5. Sõnade arv valitud transkribeeritud tekstides

Kõnetuvastusest saadud transkriptsiooni sõnade korrektsust antud töös ei hinnatud. Manuaalse ülevaatuse käigus märkas autor mõningaid transkriptsiooni vigu, mis seisnesid näiteks sõnade vales kokkukirjutamises või nimede vales kirjapildis. Kui sama sõna on erinevalt transkribeeritud, võib see mõjutada võtmesõnade määramist Standardsemate tekstide puhul võiksid eri meetoditega saadud tulemused mõneti erineda.

3.2. Märgendatud andmestik

Selleks, et võrrelda automaatse võtmesõnade ekstraheerimise tulemusi inimlugejate valitutega, paluti kahel Tallinna Ülikooli märgendajal tuvastada 180 saatetekstis võtmesõnad. Märgendajatele anti juhis valida kuni kümme sõna, mis kirjeldavad kõige täpsemini teksti sisu, ning pikemate komplekssemate tekstide puhul oli lubatud välja tuua rohkem kui kümme sõna. Eelistatud olid nimisõnad ja nimisõnafraasid, sealhulgas nimed. Võtmesõnu esitati algvormis, peamiselt ainsuse nimetavas käändes. Märgendajad järgisid nimede puhul tekstis esinevat täispikka kuju (nt eesnimi ja perekonnanimi). Vigaselt transkribeeritud sõnu võtmesõnadeks ei valitud, v.a kui korrektne sõnakuju tekstis puudus – siis lisati võtmesõna järele sulgudes parandus. Mõlemad märgendajad on toonud välja kõigi failide kohta võtmesõnu ja mõningaid kahesõnalisi fraase, mis ei ole järjestatud olulisuse järjekorras. Edaspidi viidatakse tekstis kahele märgendajale kui M1 ja M2.

Kahe märgendaja tulemuste võrdluses tuvastati, et ühel märgendajal M1 olid võtmesõnad enamasti väiketähtedega, samas kui teine märgendaja M2 kirjutas nimesid suurte algustähtedega. Andmestiku ühtlustamiseks otsustati kõik võtmesõnad väiketähestada. Kuna siinse töö põhieesmärgiks oli tuvastada ainult üksikvõtmesõnu, siis jaotati võtmefraasid osadeks, muutes need ühesõnalisteks võtmesõnadeks.

Olulised erinevused ilmnesid märgendajate leitud sõnade koguarvus: 180 failist vaid 33 faili puhul oli võtmesõnade arv ühesugune. Tulemuste kattuvuse võrdlemiseks kasutati Jaccardi sarnasuse indeksit, mis võtab arvesse ainult ühiseid elemente jagatuna elementide ühendhulgaga. Jaccardi indeksi valem on:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|},$$

kus A tähistab märgendaja M1 ja B märgendaja M2 võtmesõnade hulka. Indeksi väärtus võib olla vahemikus 0 kuni 1, kus 1 tähistab täielikku kattuvust [39]. Kõigi tekstide indeksite aritmeetiline keskmine on 0.27, mis viitab märgendusversioonide küllaltki suurele erinevusele. Lisaks Jaccardi indeksile kasutati võtmesõnaloendite sarnasuse mõõtmiseks kattuvuse koefitsienti (ingl *overlap coefficient*), mis erineb Jaccardi valemist, jagades ühisosa väiksema kogumi elementide arvuga. Kattuvuse koefitsiendi valem esitatakse kujul:

$$OC(A,B) = \frac{|A \cap B|}{min(|A|,|B|)},$$

kus A ja B tähistavad analoogselt Jaccardi indeksiga M1 ja M2 võtmesõnu [40]. Võrreldes Jaccardi indeksiga näitas kattuvuse koefitsient suuremat sarnasust, olles keskmiselt 0.43, kuigi kattuvus jäi endiselt alla poole.

Mõõdikute tulemused erinevad, sest Jaccardi indeks mõõdab ühiste elementide proportsiooni kõigi elementide suhtes (liidetakse kõik M1 ja M2 välja toodud võtmesõnad), samas kui kattuvuse koefitsient arvestab väiksemas kogumis sisalduvate ühiste elementide osakaaluga. Automaatse võtmesõnaotsingu tulemusi võrreldi eraldi märgendajate M1 ja M2 valitud sõnadega.

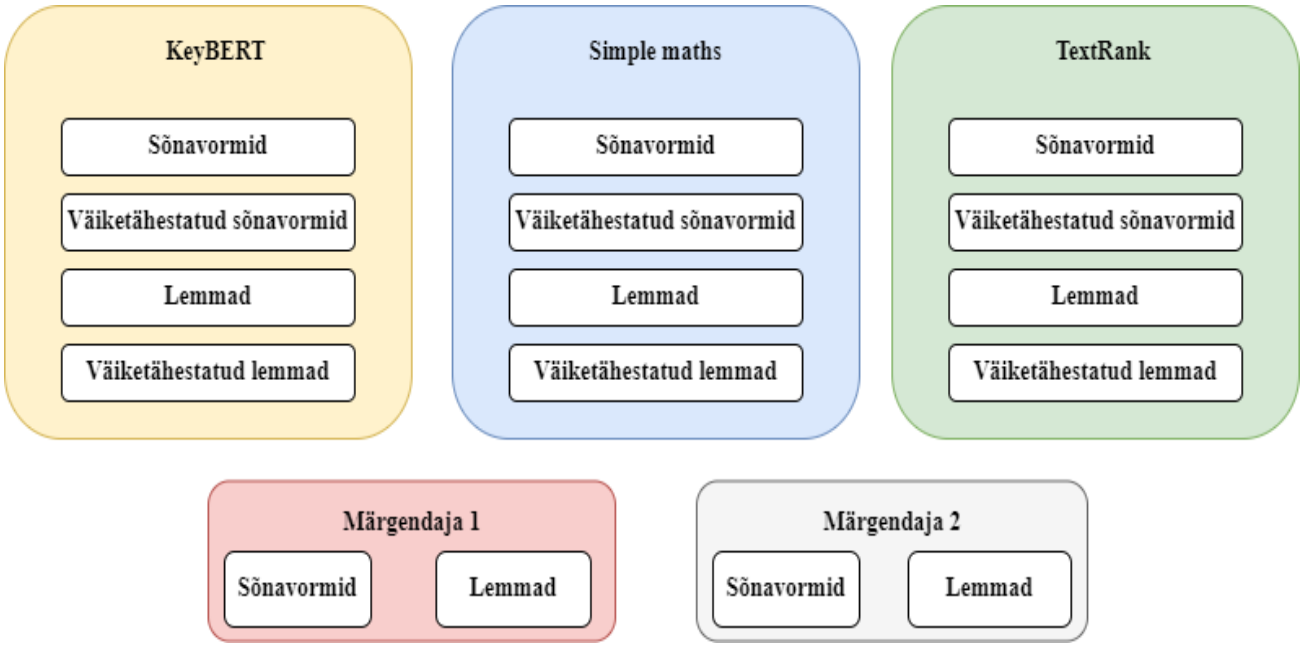
3.3. Võrdlusandmete arvutamine

ERR-i transkriptsioonidest võtmesõnade ekstraheerimiseks kasutati Pythoni versiooni 3.8 Jupyter Lab[11] keskkonnas, kus rakendati kõiki kolme ekstraheerimismeetodit: KeyBERT, *simple maths* ja TextRank. Iga meetodi jaoks loodi eraldi skriptid ning kogu kasutatud lähtekood on lisatud GitHub-i[12] keskkonda. Esmalt laeti alla transkribeeritud testandmestik *txt*-formaadis ning statistiku *simple maths* arvutuste jaoks valmistati ette ühendkorpuse sõnaloend [vt pt 2.1.1], mis koosnes väiketähestamata sõnavormidest ja algvormidest ehk lemmadest [41]. Stoppsõnade (nt *ja*, *et*, *aga*) vältimiseks kasutas autor DataDOI[13] lehelt allalaetud sõnavormide ja lemmade kujul stoppsõnade loendeid [42].

[11] https://jupyter.org/install
[12] https://github.com/hermanpetrov/KeyBERT-Estonian-setup
[13] https://datadoi.ee/handle/33/78

3.3.1. Võrdlusandmestiku variandid

Võttes arvesse keelemudelite versioonide erinevusi (suur- ja väiketähti eristavad *cased*-tüüpi mudelid) ning lähtudes ühendkorpuse algandmestikust, mis sisaldab mitte-väiketähestatud sõnu ja nende lemmasid, katsetas autor võtmesõnade ekstraheerimist nii väiketähestatult kui ka väiketähestamata sõnavormide ja lemmade kujul. Enne võtmesõnameetodite rakendamist loodi kõigist 180 tekstist Stanza[14] teegi abil lemmatiseeritud variant ning vastavalt kaks väiketähestatud varianti. Kuna saadud tulemused erinesid, siis oli eesmärk näha, kuidas võivad erinevad sõnakujud mõjutada võtmesõnade tuvastamise täpsust.



Joonis 6. Meetodite võrdlus teksti sõnakuju ja märgendusandmete põhjal

Seega seadistati võtmesõnade ekstraheerimise skriptid leidma iga meetodiga võtmesõnu tekstist neljal erineval kujul (Joonis 6). Märgendajate määratud võtmesõnu käsitleti sõnavormidena ja andmestikust loodi lemmatiseeritud variant. Valdavalt olid need võtmesõnad juba lemmade kujul, aga vahel kasutati mitmust (nt *rahvariided*) ja võtmefraasi esimene sõna võis olla käändes (nt *eurole üleminek*). Nii automaatselt saadud kui ka märgendajate võtmesõnade lemmatiseerimine Stanza abil võimaldas sõnu algvormistada samadel alustel. Ka väiketähestamata tekstist leitud võtmesõnad viidi lõpuks M1 ja M2-ga võrdlemiseks väiketähelisele kujule.

[14] https://stanfordnlp.github.io/stanza/

3.3.2. TextRanki meetodi rakendamine

Kuna TextRank lähtub PageRanki arhitektuurist, loodi meetodi skriptis[15] esmalt graafi ehitamise funktsioon build_graph, milles sõnadevaheliste seoste arvutamise aken on 2, st et arvestatakse kõrvutiasetsevate sõnade koosesinemisega (Joonis 7).

```
def build_graph(words):
    gr = nx.Graph()
    gr.add_nodes_from(set(words))
    window_size = 2
    for i in range(len(words) - window_size + 1):
        window = words[i : i + window_size]
        for j in range(1, len(window)):
            for k in range(j):
                gr.add_edge(window[k], window[j])
    return gr


def extract_keywords(graph):
    ranks = nx.pagerank(graph)
    ranked_keywords = sorted(((word, ranks[word]) for word in ranks
        ), reverse=True)
    return ranked_keywords[:180]
```

Joonis 7. TextRanki skript

Seejärel kasutatakse seda graafi sõnade tähtsuse hindamiseks, mis põhineb sõnade seostel tekstis. Saadud parimad võtmesõnad tagastatakse *CSV*-formaadis koos võtmesuse skooriga. Tulemuste failid jaotati analüüsis arvesse võetud nelja sõnakuju alusel eri kaustadesse.

3.3.3. *Simple maths*'i meetodi rakendamine

Simple maths'i rakendamiseks loodi esmalt iga teksti kohta eraldi sõnade sagedusloend. Ühtlasi leiti ka sõnade sagedused võrdluskorpuses. Saadud andmete põhjal loodi iga teksti jaoks *CSV*-formaadis tabel, mis sisaldas sõnu koos esinemissagedustega sihttekstis ja võrdluskorpuses, ning samas toodi välja nii teksti kui ka võrdluskorpuse kõikide sõna sageduste summa.

[15] https://github.com/hermanpetrov/KeyBERT-Estonian-setup/blob/main/Masters%20Project%20and%20Analysis/3_TextRank.ipynb

Tekstide kohta loodud tabelite andmeid kasutati skriptis[16] *simple maths*'i valemil põhinevas funktsioonis calculate_simpleMathsScore (Joonis 8), mis arvutas igale sõnale võtmesusskoori. Saadud tulemuste põhjal loodi uued võtmesuse järgi reastatud võtmesõnade failid ning sorteeriti sisendteksti sõnakujule vastavasse kausta. Väiketähestatud ja suurtähti sisaldava versiooni erinevus seisnes selles, kas sama sõna suure ja väikse algustähega variandi sagedus on kokku liidetud või mitte, näiteks *Eesti* ja *eesti* võivad olla erinevad sõnad, kuid väiketähestatud sõnaloendites on vaid *eesti*.

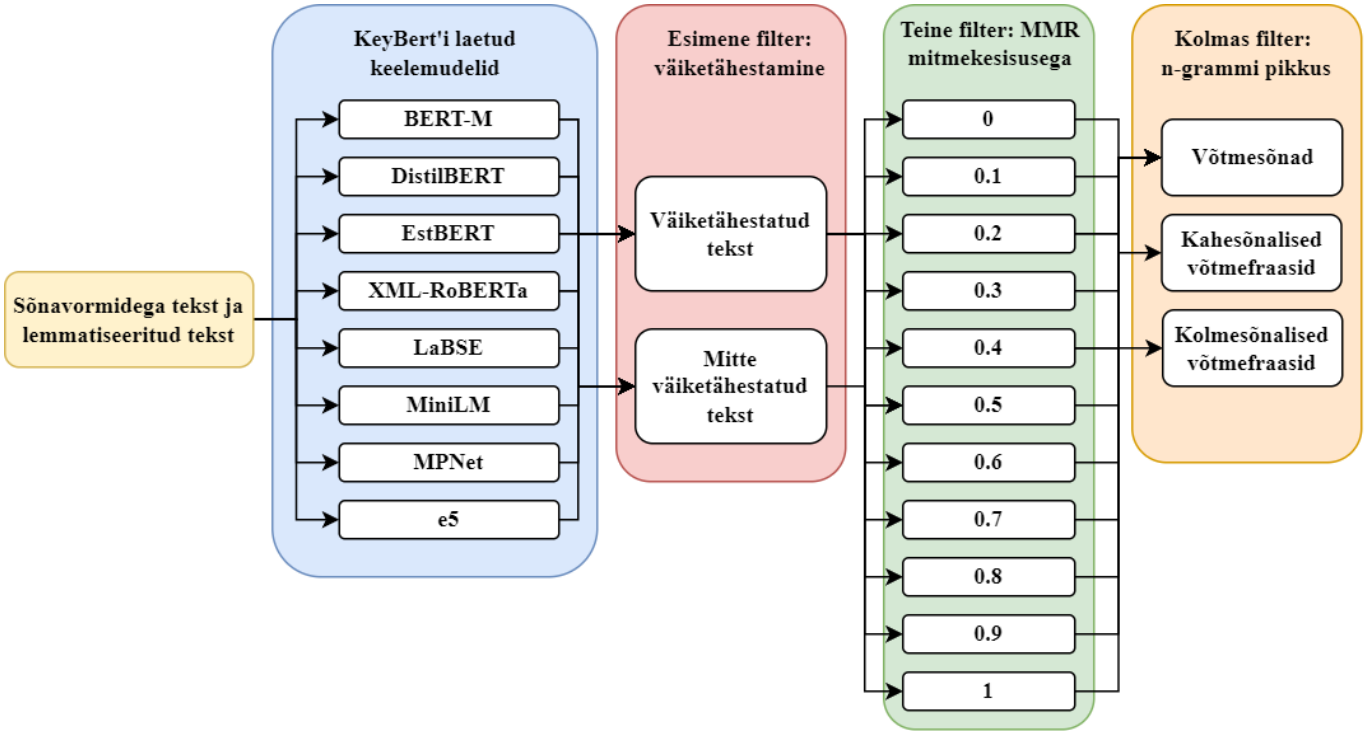
```
def calculate_simpleMathsScore(df):

    df["rfc_count"] = df["rfc_count"].replace(0, 1)

    df["fc_per_million_hits"] = (df["fc_count"] * 1000000) /
        df["fcTotalCount"]
    df["rfc_per_million_hits"] = (df["rfc_count"] * 1000000) /
        df["rfcTotalCount"]

    df["simpleMathsScore"] = (df["fc_per_million_hits"] + 1) / (
        df["rfc_per_million_hits"] + 1
    )
```

Joonis 8. *Simple maths*'i skripti osa

3.3.4. KeyBERT-i meetodi rakendamine

KeyBERT-i parima seadistuse väljaselgitamiseks pidi iga teksti neljal erineval kujul analüüsima kõigi keelemudelitega, kusjuures iga keelemudelit kasutati erineva MMR-i mitmekesisusfaktoriga. Pärast üksikute võtmesõnade leidmist korrati protsessi kahe- ja kolmesõnaliste võtmefraasidega (Joonis 9).

[16] https://github.com/hermanpetrov/KeyBERT-Estonian-setup/blob/main/Masters%20Project%20and%20Analysis/2_SimpleMathsDataCreator.ipynb



Joonis 9. Võtmesõnaotsingu variandid KeyBERT-is

KeyBERT-i rakendamiseks loodi skript, mis loeb esmalt sisse teksti kas sõnavormide või lemmatiseeritud kujul. Seejärel valitakse järjekorras üks üheksast keelemudelist, millele rakendatakse spetsiifilisi otsingu- ja vektorparameetreid. Võtmesõnade otsingu parameetrites kasutatakse järjest 11 mitmekesisusfaktorit ning lisatakse vektorparameetrid, mis täpsustavad, kas otsida võtmesõnu väiketähestatud tekstist või mitte. Viimaks määratakse vektorparameetrite põhjal võtmesõnade või võtmefraaside otsing (vt Joonis 10). Läbides kõik variatsioonid, on ühel tekstifailil iga sõnakuju kohta 297 variatsiooni, mis tähendab, et ühe teksti kohta genereeritakse kokku 1188 *CSV*-faili võtmesõnade ja nende skooridega. 180 teksti põhjal loodi niisiis 213 840 faili 200 olulisema võtmesõnaga.

```
base_folders = {
    'raw_text': 'models/raw_text_data',
    'raw_text_lemma': 'models/raw_text_lemma_data',
}
lcf_folders = {
    'raw_text': 'models/raw_text_data_LCF',
    'raw_text_lemma': 'models/raw_text_lemma_data_LCF'
}
models_info = {
    'LaBSE': ('sentence_transformer', 'sentence-transformers/LaBSE'),
    'multi_e5': ('sentence_transformer', 'intfloat/multilingual-e5-large-instruct'),
    'MiniLM_multi': ('sentence_transformer', 'sentence-transformers/paraphrase-multilingual-mpnet-base-v2'),
    'MiniLM-L12_multi': ('sentence_transformer', 'sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2'),
    'distilbertMulti': ('flair_transformer', 'distilbert/distilbert-base-multilingual-cased'),
    'bertMulti': ('flair_transformer', 'google-bert/bert-base-multilingual-cased'),
    'xlm-roberta': ('flair_transformer', 'FacebookAI/xlm-roberta-base'),
    'EstBERT': ('flair_transformer', 'tartuNLP/EstBERT'),
    'est-roberta': ('flair_transformer', 'EMBEDDIA/est-roberta')
}
ngram_ranges = [(1, 1),(2,2),(3,3)]
diversities = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]
```

Joonis 10. KeyBERT-i seadistused ja filtrid

3.3.5. KeyBERT-i andmestiku korrastamine

Võtmesõnade hindamiseks struktureeriti andmed kokkuvõtlikumaks. Võtmesõnade ekstraheerimise tulemused organiseeriti hierarhiliselt kolmel tasemel. Esiteks grupeeriti need nelja sõnakuju järgi. Teiseks jaotati iga sõnakujuga saadud andmed kolme rühma: ühesõnalised võtmesõnad ning KeyBERT-iga tuvastatud kahe ja kolme sõnaga võtmefraasid. Kolmandaks liigitati iga alajaotus veel 11 erineva mitmekesisuse faktori järgi. Iga mitmekesisuse faktori alamkaustas oli vastavalt tekstifaili nimele loodud kokkuvõttev *CSV*-formaadis tabel, kus iga mudeli veerus toodi välja eri faktorite korral leitud võtmesõnad või võtmefraasid. Antud andmestiku struktuur oli aluseks võtmesõnade täpsuse hindamisel.

3.4. Üksikvõtmesõnade täpsuse hindamine

Üksikute võtmesõnade hindamiseks arvutati kolme meetodi (KeyBERT, *simple maths* ja TextRank) F1-skoorid. Mõõdiku rakendamiseks loodi skript, mille põhjal valiti täpselt sama kogus parimad võtmesõnu, mille olid märgendajad iga teksti kohta välja toonud, ning hinnati eraldi nende kattuvust nii märgendaja M1 kui ka M2-ga. Näiteks kui märgendaja on esile toonud 12 võtmesõna, siis vaadatakse ka iga meetodi pakutud 12

suurima skooriga sõna. Automaatsete meetodite võtmesõnad oli hindamiseks väiketähestatud.

Mõõdikut F1-skoor kasutatakse loomuliku keele töötluses klassifikatsioonimudelite hindamiseks ning nende hulka kuuluvad ka võtmesõnade ekstraheerimismeetodid, mis võimaldavad välja selgitada, kas sõna kuulub teatud hulga olulisemate võtmesõnade hulka. Klassifitseerimise peamine eesmärk on saavutada hea tasakaal täpsuse ja saagise vahel, eriti olukordades, kus andmeklassid on ebavõrdselt esindatud. Mõõdiku rakendamisel leitakse esmalt täpsus (ingl *precision*):

$$precision = \frac{TP}{TP+FP},$$

kus TP (ingl *true positive*) tähistab tõeselt tuvastatud võtmesõnu ehk automaatse meetodiga leitud olulisi võtmesõnu, mis kattuvad märgendajate võtmesõnadega. FP (ingl *false positive*) ehk valepositiivsed ennustused on juhtumid, kus automaatselt leitud võtmesõna ei ole märgendajate võtmesõnade hulgas. Täpsus näitab, kui suur osa võtmesõnadest on õigesti pakutud.

Teiseks arvutatakse F1-skoori leidmiseks saagis (ingl *recall*):

$$recall = \frac{TP}{TP+FN},$$

kus TP tähistab taas õigesti määratud võtmesõnu ja FN (ingl *false negative*) ehk valenegatiivsed ennustused on märgendajate valitud võtmesõnad, mida meetod olulisemate võtmesõnade hulka ei liigitanud. Saagis näitab, kui suure osa märgendajate võtmesõnadest automaatne meetod tuvastas.

F1-skoor näitab täpsuse ja saagise vahelist harmoonilist keskmist, kus skoor jääb vahemiku 0 kuni 1 [14]. Valem on järgmine:

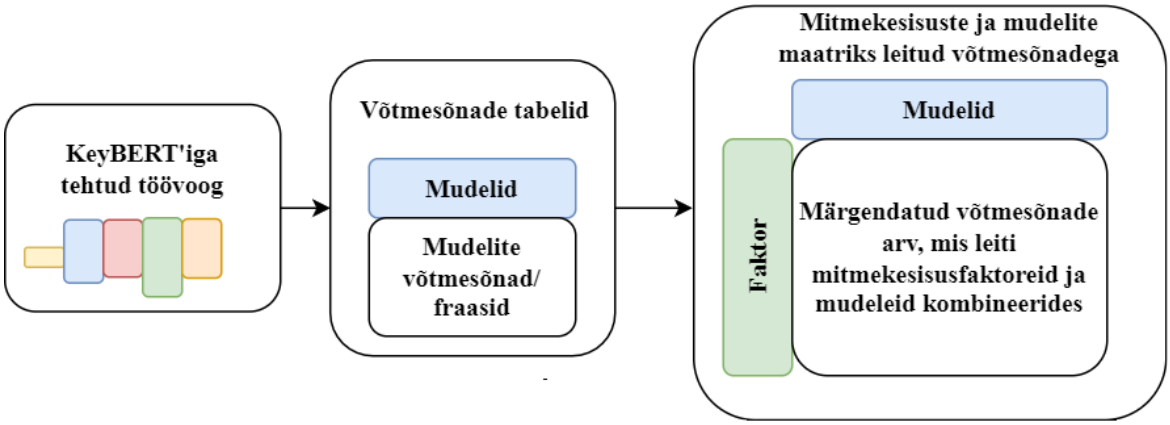
$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}.$$

Antud mõõdikut rakendati 180 tekstile eraldi ning arvutati F1-skooride aritmeetiline keskmine, mida F1-skoori kontekstis nimetatakse ka makrokeskmiseks. Saadud makrokeskmiste põhjal tehti kindlaks, millised meetodid saavutasid erineva sõnakujuga

tekste analüüsides parimaid tulemusi. Hindamise käigus vaadeldi KeyBERT-i puhul täpsemalt, milline mitmekesisusfaktor, keelemudel ja nende kombinatsioon annab suurima keskmise F1-skoori. Lisaks arvutati iga kombinatsiooni jaoks F1-skooride keskmine nelja sõnakuju lõikes.

3.5. Üksikute võtmesõnade põhjal võtmefraaside hindamine

KeyBERT-i võtmefraaside meetodi hindamisel võrreldi kahesõnalisi ja kolmesõnalisi võtmefraase märgendajate võtmesõnadega nii, et võeti arvesse üksikute võtmesõnade esindatust võtmefraasides. Seejuures aga ei kasutatud F1-skoori, kuna võtmefraaside ja võtmesõnade vahelist täpsust ja saagist pole võimalik mõõta. Sellest tingitult loodi võtmesõnade tabelite põhjal mitmekesisusfaktorite ja mudelitevaheline maatriks (Joonis 11). Maatriksi koostamiseks otsiti esmalt iga teksti tabelitest M1 ja M2-ga kattuvaid võtmesõnu. Võtmefraasides leiduvat märgendaja võtmesõna arvestati vaid üks kord ka siis, kui see esines mitmes fraasis.

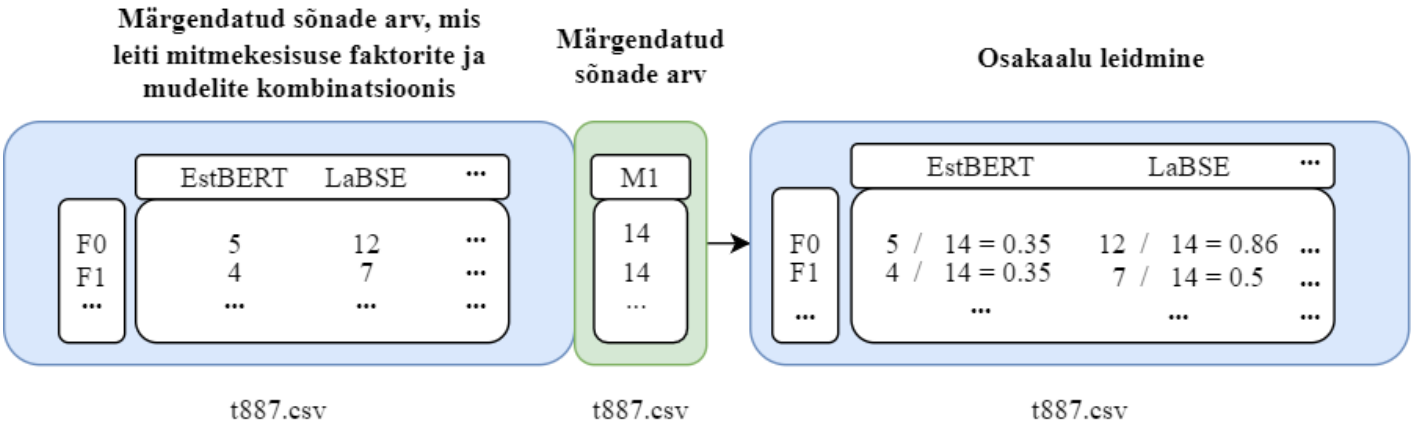


Joonis 11. KeyBERT-i andmestiku töötlus võtmefraaside asjakohasuse hindamiseks

Loodud maatriksite alusel toodi välja mitmekesisuse faktori ja mudeli kombinatsioonid koos KeyBERT-i leitud võtmefraasides esinevate M1 või M2 võtmesõnade arvuga. Iga teksti kohta loodud maatriksid olid *CSV*-formaadis ja hierarhiliselt jaotatud kaustadesse kolmel tasemel: esmalt märgendaja M1 või M2, teiseks nelja sõnakuju järgi ning seejärel n-grammi tüübi järgi. Kahe- ja kolmesõnaliste võtmefraaside kõrval on võrdluseks välja toodud ka märgendajate võtmesõnade esindatus KeyBERT-i tuvastatud üksikute võtmesõnade ehk unigrammide hulgas.

Võtmefraaside hindamiseks kasutati iga teksti puhul maatriksis leiduvaidmitmekesisusfaktori ja mudeli kombinatsiooni tulemusi. Võtmefraasides

sisalduvate märgendatud võtmesõnade arv jagati kõigi märgendatud sõnade arvuga. Selline jagatis arvutati eraldi nii märgendajaga M1 kui ka M2 ning nende tulemuste aritmeetiline keskmine määras parima mitmekesisusfaktori ja mudeli kombinatsiooni üksikute võtmesõnade, kahesõnaliste ja kolmesõnaliste võtmefraaside leidmiseks kõigi sõnakujude lõikes (Joonis 12).



Joonis 12. KeyBERT-i võtmesõnade ja -fraasidega kattuvate märgendatud võtmesõnade osakaalu leidmine

Hindamismeetodi abil oli võimalik uurida ka üksikvõtmesõnade kattuvust märgendatud testandmetega. Kui F1-skoori arvutades võeti arvesse vaid väikest hulka võtmesõnu vastavalt märgendaja valitud sõnade arvule, siis nüüd vaadeldi kattuvust esimese 200, 50 ja 10 võtmesõnade ja -fraaside kandidaadiga. Ühesõnaliste tulemuste võrdlus märgendusega viidi sellisel moel läbi ka meetoditega *simple maths* ja TextRank.

Kõigi tekstide põhjal arvutati keskmine leitud võtmesõnade osakaal. Seejärel toodi välja keelemudelid ja mitmekesisusfaktorid, millega KeyBERT saavutas parimaid tulemusi. Selleks arvutati esmalt iga faktori ja mudeliga eraldi keskmine kattuvate võtmesõnade osakaal uni-, bi- ja trigrammide ning nelja erineva sõnakuju lõikes. Viimaks leiti parimad keelemudeli ja mitmekesisusfaktori kombinatsioonid, mille puhul vaadeldi ka kolme n-grammi pikkuse alusel saadud tulemuste üldist keskväärtust.

4. Tulemused

Antud peatükk keskendub meetodite tulemuste võrdlevale analüüsile. Peatükis võrreldakse KeyBERT-i, *simple maths*'i ja TextRanki tulemusi ning tuuakse välja KeyBERT-i mitmekesisuse faktorid ja mudelid, millega leiti kõikide sõnakujude lõikes märgendusega kõige lähedasemaid võtmesõnu ning kahe- ja kolmesõnalisi võtmefraase. Kõiki saadud tulemusi kokkuvõttev fail koos koondtabeliga on lisatud GitHubi[17].

4.1. Võtmesõnade hindamine F1-skoori alusel

Esimeses analüüsis hinnati kolme meetodi keskmist F1-skoori üksikute võtmesõnade leidmisel (Tabel 2). KeyBERT-i puhul valiti võrdluseks parima mitmekesisuse faktori ja mudeli kombinatsioon iga sõnakuju (väiketähestatud ja -tähestamata sõnavormid ja lemmad) lõikes. Tulemuste põhjal oli KeyBERT-il madalaim keskmine F1-skoor. Sõnakujude paremusjärjestuses oli alati esikohal TextRank, millele järgnesid *simple maths* ja KeyBERT. Olulisena tuli välja, et väiketähestamata lemmade kattuvus märgendajatega oli võrreldes teiste sõnakujudega suurem. Kõigi kasutatud meetoditega oli näha, et lemmade puhul on keskmine F1-skoor kõrgeim, samas kui väiketähestatud sõnavormidega saadi madalaim keskmine F1-skoor. Lisaks ilmneb, et väiketähestatud sõnakuju avaldab KeyBERT-i tulemustele negatiivset mõju. Kuigi TextRanki parim keskmine F1-skoor ületab 18 protsenti juurde, on siiski automaatmeetodite kattuvus märgendajatega väiksem kui märgendajate omavaheline kattuvus.

Tabel 2. Kolme ekstraheerimismeetodi keskmised F1-skoorid

Märgendaja	Meetod	Väiketähestatud sõnavormid	Sõnavormid	Väiketähestatud lemmad	Lemmad
M1	KeyBERT	6.16%	8.18%	10.41%	10.86%
	Simple Maths	7.58%	6.90%	11.69%	12.10%
	TextRank	**11.33%**	**13.14%**	**16.61%**	**16.83%**
M2	KeyBERT	6.06%	7.16%	10.99%	11.26%
	Simple Maths	8.07%	7.42%	14.95%	15.06%
	TextRank	**12.66%**	**11.63%**	**18.53%**	**18.63%**

[17] https://github.com/hermanpetrov/KeyBERT-Estonian-setup/blob/main/Masters%20Project%20and%20Analysis/analytical_data/Complete_results_F1%26AVG.xlsx

Järgnevalt vaadeldi täpsemalt tulemusi, mis saadi erinevate mitmekesisuse faktori (M.faktor) ja mudeli kombinatsioonidega (Tabel 3). Mitmekesisuse faktori eristamiseks määrati faktoritele värvid: kõige sarnasemate võtmesõnade puhul kasutati faktoriga 0 rohelist värvi, väiksema võtmesõnade sarnasusega faktori 1 puhul aga punast.

Keskmise F1-skoori analüüsist selgus, et kõikide sõnakujude lõikes andsid parimaid tulemusi lause-transformerid. Kahe erineva sõnavormi puhul oli märgendajatega suurim kattuvus LaBSE mudelil. Ka väiketähestatud lemmade puhul olid märgendajaga M1 sarnaseim LaBSE-ga saadud tulemus, tavalemmade puhul oli kattuvus suurem MiniLM-iga. Märgendajaga M2 kattus mõlema lemmakuju korral MiniLM.

Mitmekesisuse faktori osas ei ole tulemused väga ühtlased, kuid suuremalt jaolt jäid mitmekesisuse faktorid alla 0.5, välja arvatud märgendajaga M1 väiketähestamata sõnavormide puhul, kui parima tulemuse andis faktor 1.

Tabel 3. Parima keskmise F1-skooriga mudelid koos mitmekesisuse faktoriga

Sõnatüüp	Märgendaja	M. faktor	Mudel	Keskmine F1
Väiketähestatud sõnavormid	M1	0.4	LaBSE	6.16%
	M2	0	LaBSE	6.06%
Sõnavormid	M1	1	LaBSE	8.18%
	M2	0	LaBSE	7.16%
Väiketähestatud lemmad	M1	0.3	LaBSE	10.42%
	M2	0	MiniLM	10.99%
Lemmad	M1	0	MiniLM	10.86%
	M2	0.5	MiniLM	11.32%

Kui vaadata KeyBERT-i parimaid keskmisi F1-skoore, mis saadud kõigi mudelitega erinevate mitmekesisuse faktorite lõikes (Üld.Kesk.F1), siis oli neil juhtudel faktor tihti 0 lähedane, mis tähendab, et eelistati võtmesõnu, mis olid omavahel sarnasemad (Tabel 4). Parima keskmise F1-skoori andnud mitmekesisuse faktorite väärtused jäid vahemikku

0 kuni 0.4, kus kõige sagedamini kasutati mitmekesisuse faktorit 0.2. Varieerumine oli suurem märgendajaga (Märg.) M1, kus mitmekesisuse faktor oli väiketähestatud sõnavormide ja lemmade puhul vastavalt 0.4 ja 0.3.

Tabel 4. Parima keskmise F1-skooriga mitmekesisused

	Väiketähestatud sõnavormid		**Sõnavormid**		**Väiketähestatud lemmad**		**Lemmad**	
Märg.	**M. faktor**	**Üld.Kesk. F1**	**M. faktor**	**Üld.Kesk. F1**	**M. faktor**	**Üld.Kesk. F1**	**M. faktor**	**Üld.Kesk. F1**
M1	0.4	3.31%	0.2	3.76%	0.3	6.13%	0.2	6.39%
M2	0	3.59%	0.2	4.00%	0.2	6.52%	0.2	6.78%

Keskmised F1-skoorid, mis saadud eri mudelitega kõiki mitmekesisuse faktoreid kasutades, tõid esile kaks parimate tulemustega keelemudelit, millest üks paistis silma sõnavormide ja teine lemmade analüüsi korral (Tabel 5). LaBSE mudelil olid mõlema märgendajaga võrdluses parimad tulemused sõnavormidega, samas kui lemmadega oli parim keskmine F1-skoor MiniLM-i mudelil. Taas võib näha, et mitte väiketähestatud lemmade alusel leitud võtmesõnad olid märgendajatega võrreldes parima skooriga.

Tabel 5. Parima keskmise F1-skooriga keelemudelid

	Väiketähestatud sõnavormid		**Sõnavormid**		**Väiketähestatud lemmad**		**Lemmad**	
Märg.	**Mudel**	**Üld.Kesk. F1**	**Mudel**	**Üld.Kesk. F1**	**Mudel**	**Üld.Kesk. F1**	**Mudel**	**Üld.Kesk. F1**
M1	LaBSE	5.91%	LaBSE	7.69%	MiniLM	9.45%	MiniLM	10.18%
M2	LaBSE	5.46%	LaBSE	6.57%	MiniLM	9.99%	MiniLM	10.53%

4.2. Võtmesõnade ja -fraaside hindamine märgendatud võtmesõnade esindatuse järgi

Enne võtmefraaside hindamist kasutati keskmise leitud võtmesõnade osakaalu hindamise meetodit üksikute võtmesõnade hindamiseks (Tabel 6). Saadud tulemuste põhjal ilmnes järjepidevalt, et TextRank oli parem kui teised kaks meetodit kõigi kandidaatide arvude (K.A.) ja sõnakujude lõikes. Kõikide meetodite puhul saavutati 200 esimese võtmesõnaga arvestades kõige paremad tulemused ning kandidaatide arvu järkjärgulisel vähendamisel

oli märgata kattuvate võtmesõnade keskmise osakaalu kahekordset või kolmekordset langust. Samuti on märgatav, et lemmatiseeritud sõnakuju aitab suurema tõenäosusega märgendajate võtmesõnu leida.

Tabel 6. Märgendatud sõnade keskmine esindatus 200, 50 ja 10 võtmesõnakandidaadi seas

K.A.	Märg.	Meetod	Väiketähestatud sõnavorm	Sõnavorm	Väiketähestatud lemmad	Lemmad
200	M1	Keybert	35.21%	36.46%	52.34%	52.14%
		Simple Maths	32.61%	32.71%	52.23%	53.01%
		TextRank	**43.41%**	**43.98%**	**66.22%**	**66.32%**
	M2	Keybert	34.51%	34.73%	50.08%	50.79%
		Simple Maths	34.32%	34.66%	54.55%	55.10%
		TextRank	**47.17%**	**47.46%**	**71.18%**	**71.45%**
50	M1	Keybert	16.31%	17.82%	24.51%	23.71%
		Simple Maths	16.25%	16.00%	24.22%	25.41%
		TextRank	**23.21%**	**26.60%**	**36.79%**	**37.31%**
	M2	Keybert	16.01%	17.63%	24.62%	25.44%
		Simple Maths	16.78%	16.40%	29.01%	30.10%
		TextRank	**26.16%**	**26.60%**	**42.72%**	**43.08%**
10	M1	Keybert	5.32%	7.11%	8.96%	9.15%
		Simple Maths	6.39%	6.00%	10.22%	10.72%
		TextRank	**10.39%**	**11.06%**	**14.22%**	**14.44%**
	M2	Keybert	5.05%	5.94%	9.15%	9.42%
		Simple Maths	7.78%	7.17%	13.89%	14.33%
		TextRank	**12.28%**	**13.11%**	**16.83%**	**17.06%**

Kasutades 200, 50 ja 10 kandidaati erinevate n-grammidega, analüüsiti esmalt kõige suurema keskmise osakaaluga mitmekesisuse faktori ja mudelite kombinatsioonil põhinevaid väärtusi (Tabel 7). Kombinatsioonide osakaaluprotsent oli kõige kõrgem, kui

kasutati kolme sõnalisi võtmefraase (trigramme). Võrreldes tavaliste ühe sõnaliste võtmesõnadega (unigrammidega) olid kolme sõnaliste võtmefraaside tulemused kaks korda parem kõigi kandidaatarvude lõikes. Kahe- ja kolme sõnaliste fraaside võrdluses oli kõigi kandidaatide ja sõnavormide lõikes maksimaalne erinevus umbes nelja kuni kümne protsendi vahel.

Tabel 7. KeyBERT-i parimad märgendatud võtmesõnade esindatuse osakaalud

K.A.	Märg.	n-gramm	Väiketähestatud sõnavorm	Sõnavorm	Väiketähestatud lemmad	Lemmad
200	M1	unigramm	35.21%	36.46%	52.34%	52.14%
		bigramm	47.55%	51.26%	70.92%	75.56%
		trigramm	**54.47%**	**56.71%**	**79.36%**	**81.20%**
	M2	unigramm	34.51%	34.73%	50.08%	50.79%
		bigramm	48.33%	49.25%	68.82%	72.14%
		trigramm	**54.50%**	**55.02%**	**76.89%**	**77.52%**
50	M1	unigramm	16.31%	17.82%	24.51%	23.71%
		bigramm	23.51%	29.23%	37.03%	43.14%
		trigramm	**31.93%**	**36.64%**	**46.64%**	**51.40%**
	M2	unigramm	16.01%	17.63%	24.62%	25.44%
		bigramm	25.76%	28.97%	38.81%	42.42%
		trigramm	**33.07%**	**35.43%**	**47.75%**	**50.50%**
10	M1	unigramm	5.33%	7.12%	8.97%	9.15%
		bigramm	9.43%	12.90%	15.78%	19.02%
		trigramm	**15.02%**	**17.68%**	**22.23%**	**24.78%**
	M2	unigramm	5.05%	5.94%	9.15%	9.43%
		bigramm	10.16%	13.17%	17.03%	19.17%
		trigramm	**14.85%**	**16.90%**	**22.50%**	**23.62%**

Järgnevalt analüüsiti 7. tabeli parimate keskmiste osakaaludega väärtuste mudeleid (Tabel 8). Peamised kolm mudelit, millel olid parimad keskmised osakaalud, olid e5, LaBSE, MiniLM ja EstBERT. Unigrammide tulemuste analüüsist ilmnes, sarnaselt tabeli 3 tulemustega, et LaBSE ja MiniLM lause-transformereid parimat keskmise oskaaluga võtmesõnade leidmisel. Võtmefraaside puhul oli märkimisväärseks tulemuseks 200 ja 50 kandidaadi korral oli väiketähestatud sõnavormidega EstBERTi suur keskmine osakaal, kuid üldiselt oli kõigi sõnavormide lõikes ülekaalus e5 mudel. 200, 50 võtmesõna

kandidaatidega on näha, et e5 on tihti võtmefraaside parima keskmise osakaaluga, kuid 10 võtmesõna kandidaadi põhjal oli selgemini näha, et e5 on parima keskmise osakaaluga keelemudel võtmefraaside puhul, samas kui üksikute võtmesõnade puhul oli LaBSE, parimate tulemustega.

Tabel 8. KeyBERT-i parimad mudelid, mille tulemused kattusid märgendusega kõige rohkem

K.A.	Märg.	n-gramm	Väiketähestatud sõnavorm	Sõnavorm	Väiketähestatud lemmad	Lemmad
200	M1	unigramm	LaBSE	LaBSE	LaBSE	LaBSE
		bigramm	EstBERT	e5	e5	e5
		trigramm	LaBSE	e5	LaBSE	e5
	M2	unigramm	LaBSE	LaBSE	LaBSE	MiniLM
		bigramm	EstBERT	e5	LaBSE	e5
		trigramm	EstBERT	e5	LaBSE	LaBSE
50	M1	unigramm	LaBSE	LaBSE	LaBSE	MiniLM
		bigramm	EstBERT	e5	e5	e5
		trigramm	e5	e5	e5	e5
	M2	unigramm	LaBSE	LaBSE	MiniLM	MiniLM
		bigramm	e5	e5	e5	e5
		trigramm	EstBERT	e5	e5	e5
10	M1	unigramm	LaBSE	LaBSE	LaBSE	LaBSE
		bigramm	e5	e5	e5	e5
		trigramm	e5	e5	e5	e5
	M2	unigramm	LaBSE	LaBSE	LaBSE	MiniLM
		bigramm	e5	e5	e5	e5
		trigramm	e5	e5	e5	e5

Lisaks keelemudelitele kontrolliti järgnevalt parimate keskmiste osakaalude valikute mitmekesisuse faktoreid (Tabel 9). Unigrammide analüüsist on näha peamiselt nullile lähedasi tulemused olid parima keskmise osakaaluga. Unigrammide tulemused jäävad 0.2-0.3 faktori juures, välja arvatud M1 puhul, kus kahe sõnavormi kujul 10 võtmesõna kandidaadiga tulemus oli lähedal 1-le. Võtmefraaside analüüsil olid faktori arvud keskmiselt üle 0.5, vahemikus sagedamini 0.6-0.8 vahemikus.

Tabel 9. KeyBERT-i parimad mitmekesisuse seadistused

K.A.	Märg.	n-gramm	Väiketähestatud sõnavorm	Sõnavorm	Väiketähestatud lemmad	Lemmad
200	M1	unigramm	0	0.2	0.2	0.2
		bigramm	0.3	0.6	0.7	0.6
		trigramm	0.8	0.5	0.8	0.6
	M2	unigramm	0	0	0.1	0.3
		bigramm	0	0.5	0.7	0.7
		trigramm	0.4	0.5	0.8	0.7
50	M1	unigramm	0	0	0.1	0.1
		bigramm	0.5	0.6	0.6	0.8
		trigramm	0.6	0.6	0.6	0.7
	M2	unigramm	0.1	0.1	0.2	0.1
		bigramm	0.6	0.6	0.6	0.7
		trigramm	0.5	0.6	0.6	0.7
10	M1	unigramm	0.7	1	0.3	0.3
		bigramm	0.6	0.8	0.7	0.8
		trigramm	0.7	0.7	0.7	0.7
	M2	unigramm	0	0	0.1	0.2
		bigramm	0.6	0.7	0.7	0.6
		trigramm	0.6	0.7	0.6	0.7

Parimate üldkeskmiste osakaaludega (Üld.Kesk.Osak) mitmekesisuse faktori põhjal oli korduvalt näha, et unigrammidega saadud tulemused on üldiselt täpsemad väiksema faktorinumbriga, välja arvatud M1 märgendaja sõnavormi ja lemma sõnakujudega 200 võtmesõna kandidaadi juures (Tabel 10). Samuti on parimate mudelite keskmiste osakaaludega mudelite mitmekesisuse faktorite tabelist (Tabel 9) näha, et võtmefraaside parimad üldkeskmised osakaalud kõigi kandidaatide ja kahe sõnavormi kuju jaoks jäävad 0,5-0,7 vahemikku ning kahe lemma kuju kasutamisel 0,6-0,7 faktori juurde.

Tabel 10. Parima keskmise kattuvusega mitmekesisuse faktorid kõiki mudeleid arvestades

			Väiketähestatud sõnavorm		Sõnavorm		Väiketähestatud lemmad		Lemmad	
K.A.	**Märg.**	**n-gramm**	**M. faktor**	**Üld.Kesk. Osak.**	**M. faktor**	**Üld.Kesk. Osak.**	**M. faktor**	**Üld.Kesk. Osak.**	**M. faktor**	**Üld.Kesk. Osak.**
200	M1	unigramm	0	24.84%	0.7	22.77%	0.3	42.69%	0.5	44.01%
		bigramm	0.6	40.63%	0.5	37.10%	0.6	63.85%	0.6	66.01%
		trigramm	0.5	48.47%	0.5	43.76%	0.7	72.65%	0.8	74.19%
	M2	unigramm	0	25.75%	0	22.73%	0.2	41.83%	0.3	42.65%
		bigramm	0.4	40.90%	0.5	36.85%	0.6	63.81%	0.6	65.52%
		trigramm	0.5	48.73%	0.5	43.51%	0.8	72.54%	0.8	73.38%
50	M1	unigramm	0	9.65%	0.1	8.75%	0.2	16.71%	0.2	16.85%
		bigramm	0.5	18.24%	0.5	16.79%	0.7	30.74%	0.6	31.76%
		trigramm	0.6	23.93%	0.5	21.62%	0.8	39.25%	0.8	40.19%
	M2	unigramm	0	10.43%	0	9.50%	0.2	17.26%	0.2	17.70%
		bigramm	0.5	19.34%	0.5	17.45%	0.6	32.28%	0.6	33.17%
		trigramm	0.6	25.06%	0.6	22.05%	0.7	40.57%	0.7	41.33%
10	M1	unigramm	0.4	2.79%	0.2	2.65%	0	5.23%	0.1	5.43%
		bigramm	0.6	6.01%	0.6	5.81%	0.6	10.90%	0.6	11.69%
		trigramm	0.6	8.70%	0.6	8.16%	0.7	14.78%	0.7	15.61%
	M2	unigramm	0.2	2.96%	0.2	2.69%	0.1	5.59%	0.1	5.70%
		bigramm	0.6	6.55%	0.6	6.12%	0.6	11.50%	0.6	12.32%
		trigramm	0.6	9.44%	0.6	8.38%	0.6	15.76%	0.7	16.05%

Üldkeskmise osakaalu mudelite tulemuste hulgas oli korduvalt näha, et LaBSE ja MiniLM keelemudelid olid unigrammide puhul paremad. Siiski on märgata, et MiniLM on eelistatud lemma sõnavormide jaoks (Tabel 11). Võtmefraase analüüsides on selge, et e5 keelemudel on ülekaalus ja sobib üldkeskmiste tulemuste põhjal kõige paremini märgendajate võtmesõnade leidmiseks. Erandina oli aga kolmesõnaliste võtmefraasidega 200 ja 50 kandidaadi korral väiketähestatud lemmadega märgendajaga M2, keelemudeliga est-roberta, üldkeskmine tulemus parem.

Tabel 11. Parima keskmise kattuvusega mudelid kõiki mitmekesisuse faktoreid arvestades

			Väiketähestatud sõnavorm		Sõnavorm		Väiketähestatud lemmad		Lemmad	
K.A.	**Märg.**	**n-gramm**	**Mudel**	**Üld.Kesk. Osak.**	**Mudel**	**Üld.Kesk. Osak.**	**Mudel**	**Üld.Kesk. Osak.**	**Mudel**	**Üld.Kesk. Osak.**
200	M1	unigramm	LaBSE	29.20%	LaBSE	32.33%	LaBSE	46.86%	LaBSE	49.81%
		bigramm	e5	43.87%	MiniLM	48.03%	e5	65.73%	e5	69.84%
		trigramm	e5	50.95%	MiniLM	53.62%	e5	74.03%	e5	76.12%
	M2	unigramm	LaBSE	26.31%	MiniLM	29.18%	MiniLM	44.50%	MiniLM	45.78%
		bigramm	e5	43.60%	MiniLM	46.64%	e5	65.40%	e5	68.37%
		trigramm	e5	50.00%	MiniLM	52.00%	est-roberta	74.11%	e5	74.34%
50	M1	unigramm	LaBSE	14.38%	LaBSE	15.97%	LaBSE	22.21%	MiniLM	22.42%
		bigramm	e5	22.04%	MiniLM	26.69%	e5	34.18%	e5	38.81%
		trigramm	e5	29.21%	MiniLM	33.78%	e5	42.80%	e5	46.89%
	M2	unigramm	LaBSE	13.02%	LaBSE	14.64%	MiniLM	22.17%	MiniLM	23.39%
		bigramm	e5	23.34%	MiniLM	26.68%	e5	36.02%	e5	39.49%
		trigramm	e5	29.62%	MiniLM	32.14%	est-roberta	45.46%	e5	46.19%
10	M1	unigramm	LaBSE	5.03%	LaBSE	6.70%	LaBSE	8.15%	MiniLM	8.73%
		bigramm	e5	8.81%	MiniLM	12.02%	e5	14.84%	e5	17.08%
		trigramm	e5	12.99%	MiniLM	16.19%	e5	20.35%	e5	22.33%
	M2	unigramm	LaBSE	4.60%	LaBSE	5.54%	MiniLM	8.58%	MiniLM	8.99%
		bigramm	e5	9.46%	MiniLM	12.08%	e5	15.99%	e5	17.78%
		trigramm	e5	13.38%	MiniLM	15.56%	e5	20.96%	e5	21.67%

Viimaks analüüsiti, millised üldkeskmise osakaaluga mitmekesisuse faktori ja mudeli kombinatsioonid olid parimad kõigi n-grammide sõnavormide lõikes (Tabel 12). Parimad 200 ja 50 kandidaadi väiketähestatud sõnavormide kombinatsioonid koosnesid peamiselt madala mitmekesisuse faktoriga EstBERT mudelitest. Ainus LaBSE mudel koos mitmekesisuse faktoriga 0,6 esines ainult 200 kandidaadi väiketähestatud lemma kujul. Valdav enamus moodustas siiski pea kõigi n-grammide puhul suure ülekaalu, kus

mitmekesisuse faktor oli 0,6-0,7 ning kombineeritud e5 mudeliga, mis jällegi tõi esile e5 paremate hulka kuulumise.

Tabel 12. Parimad mudeli ja mitmekesisuse faktori kombinatsioonid kõikide n-grammide ja sõnakujude lõikes

		Väiketähestatud sõnavorm			Sõnavorm			Väiketähestatud lemmad			Lemmad		
K.A.	**Märg.**	**Mudel**	**M. faktor**	**Üld.Kesk. Osak.**	**Mudel**	**M. faktor**	**Üld.Kesk. Osak.**	**Mudel**	**M. faktor**	**Üld.Kesk. Osak.**	**Mudel**	**M. faktor**	**Üld.Kesk. Osak.**
200	M1	EstBERT	0.2	42.57%	e5	0.6	46.74%	LaBSE	0.6	64.70%	e5	0.6	69.60%
	M2	EstBERT	0	43.78%	e5	0.5	44.43%	e5	0.6	62.21%	e5	0.6	64.62%
50	M1	EstBERT	0.5	21.42%	e5	0.6	25.86%	e5	0.7	32.94%	e5	0.6	37.72%
	M2	EstBERT	0.3	23.12%	e5	0.6	25.36%	e5	0.6	34.44%	e5	0.6	37.76%
10	M1	e5	0.7	9.18%	e5	0.7	11.72%	e5	0.7	14.59%	e5	0.7	17.00%
	M2	e5	0.6	9.40%	e5	0.7	11.61%	e5	0.6	15.61%	e5	0.7	17.01%

4.3. Tulemuste järeldused

Analüüsitud tulemuste põhjal on selge, et KeyBERT ei ületa eestikeelsetest tekstidest võtmesõnade ekstraheerimises lävendit, jäädes TextRanki ja *simple maths*'ile alla nii keskmise F1-skoori kui ka märgendatud võtmesõnade keskmise esindatuse osas. Võrreldes märgendatud sõnu 200, 50 ja 10 esimese võtmesõnakandidaadiga, olid KeyBERT-i tulemused endiselt madalamad kui TextRanki ja *simple maths*'i omad. Tulemused näitasid siiski, et tavapäraste, väiketähestamata lemmade kujule viidud tekstidest leiti enim inimmärgendatud võtmesõnu ja selliseid sõnu sisaldavaid võtmefraase.

KeyBERT-i mitmekesisuse faktorite analüüsimisel selgus, et üksikvõtmesõnade otsingul tuleks eelistada nullile lähedasemaid faktoreid, keskmiselt väärtusega 0.2 juures, mille puhul valitakse omavahel sarnasemaid võtmesõnu. Võtmefraaside puhul tasuks aga eelistada faktoreid, mis on lähedasemad 1-ga, täpsemalt vahemikus 0.6 kuni 0.7, mis tähendab, et leitakse üksteisest semantiliselt erinevamaid fraase.

Mudelite analüüsi tulemusena selgus, et parimaid tulemusi saavutavad LaBSE, MiniLM ja e5. Üksikvõtmesõnade puhul on eelistatav LaBSE ja MiniLM-i mudelite kasutamine.

MiniLM mudeliga saadi parimaid tulemusi mitteväiketähestatud lemmade puhul. e5 mudeli eelis ilmneb erinevaid n-grammi pikkusi arvesse võttes, st nii üksikvõtmesõnade kui ka kahe- ja kolmesõnaliste fraaside otsingul.

Mitmekesisuse faktori ja mudelite kombinatsiooni analüüs näitas, et parimad F1-skoorid saadi sõnavormide puhul LaBSE mudeliga, mille mitmekesisuse faktor oli madal, 0.2. Lemmade puhul andis parima tulemuse MiniLM mudel, samuti 0.2 mitmekesisuse faktoriga. Fraaside puhul oli tõhusaim e5 mudeli kasutamine 0.6–0.7 mitmekesisuse faktoriga. Kõikide n-grammide puhul on parim tulemus e5 mudelil 0.7 mitmekesisuse faktoriga, olenemata sõnakujust.

Nelja sõnakuju võrdlus näitab, et KeyBERT jääb täpsuses maha, kuid tulemus sõltub tugevalt valitud üheksast keelemudelist. Antud katset võib pidada edukaks, kuna see annab teavet, millised keelemudelid on sobivamad KeyBERT-i kasutamiseks. Üldiselt on näha, et parima tulemuse saavutavad lause-transformerid (eelkõige LaBSE, MiniLM ja fraaside puhul e5). Üksikvõtmesõnade seadistuses on sobivaim mitmekesisuse faktor 0.2, kuid edasise arenduse ja uuringute puhul on võtmefraaside tuvastamisel soovitatav lähtuda mitmekesisuse faktorist 0.6–0.7. Kombinatsioone analüüsides on e5 mudel 0.7 mitmekesisuse faktoriga lemmade puhul parima efektiivsusega. ERR-i prototüübi puhul võiks kasutada kas TextRanki meetodit või KeyBERT-i eelistades ekstraheerida võtmesõnu väiketähestamata lemmade kujul, kasutades mudelit MiniLM (võtmefraaside puhul e5). See, et automaatse võtmesõnatuvastuse tulemused osutusid märgendusega kõige sarnasemaks lemmatiseeritud teksti korral, on ootuspärane, lähtudes testmaterjali märgendusest (enamik võtmesõnu on algvormi kujul). Huvitav on sisendteksti väiketähestamise negatiivne mõju võtmesõnastamise tulemustele, mis näitab, et osa võtmesõnaotsinguks vajalikku keelelist infot läheb sel moel kaduma.

Autor näeb edasiarenduse eesmärgina KeyBERT-i täpsuse katsetamist kahesõnaliste võtmefraaside ekstraheerimisel, kasutades märgendajata algsetes failides märgendatud võtmefraase. Võtmesõnade ekstraheerimise edasiarendusena võiks katsetada võtmesõnade otsingut, rakendades sõnaliigi filtreid – näiteks teostada võtmesõnade otsing ainult nimisõnade hulgas, et näha, kas sõnade filtreerimine toob kaasa täpsemaid tulemusi.

5. Kokkuvõte

Magistritöö peamine eesmärk oli välja selgitada, kui tõhusalt on võimalik leida eestikeelsetest tekstidest võtmesõnu, kasutades KeyBERT-i meetodit üheksa erineva keelemudeli ja erinevate seadistustega. Töö alameesmärk oli uurida, millised olid KeyBERT-i optimaalsed konfiguratsioonid ja kui täpselt suudab KeyBERT leida võtmesõnu võrreldes graafipõhise TextRanki ja statistilise *simple maths*'i meetodiga. Viimaseks uuriti KeyBERT-iga tuvastatud võtmefraaside asjakohasust selle põhjal, kuivõrd need sisaldavad märgendatud võtmesõnu.

Uuringus rakendati võtmesõnade ekstraheerimise meetodeid 180 transkribeeritud ERR-i raadiosaate tekstile ning hinnati tulemuste täpsust võrreldes märgendajate määratud võtmesõnadega. Üksikvõtmesõnade ja võtmefraaside tuvastamisel kasutati nelja erinevat sõnakuju (tekst võis olla sõnavormide või lemmade kujul, väiketähestatud või mitte) ning võtmefraasidena vaadeldi kahe- ja kolmesõnalisi üksusi. Tulemusema selgus, et KeyBERT ei ületa lävendmeetodeid TextRank ja *simple maths*. Eri seadistuste ja keelemudelitega saadud tulemuste analüüsis selgus, et parimad mudelid võtmesõnade otsinguks on lause-transformerid: võtmesõnade jaoks LaBSE ja MiniLM ning võtmefraaside jaoks e5. Seadistustest on parem kasutada KeyBERT-iga võtmesõnade otsinguks mitmekesisuse faktorit 0.2 ning võtmefraaside puhul faktorit 0.6–0.7.

Töö väljundid on järgmised:

- Kolme võtmesõnade ekstraheerija võrdlus, mis põhineb F1-skooril ja inimmärgendatud võtmesõnade esindatuse osakaalul;
- KeyBERT-i kohandatud seadistuste uurimine eestikeelsete tekstide analüüsimise jaoks;
- KeyBERT-i efektiivsuse uurimine, kasutades erinevaid keelemudeleid eestikeelsete tekstide analüüsimiseks;
- KeyBERT-iga leitud võtmefraaside analüüs, lähtudes nende üksikvõtmesõnade sisaldusest;

- Parimate transformer-keelemudelite valimine võtmesõnade ekstraheerimiseks eestikeelsetest tekstidest koos sobivaimate konfiguratsioonidega;
- Katsetuslik parimate transformer-keelemudelite valimine leidmaks eestikeelsetest tekstidest võtmefraase, mis sisaldavad olulisi võtmesõnu.

Kasutatud kirjandus

[1] E. Ford, J. A. Carroll, H. E. Smith, D. Scott, ja J. A. Cassell, „Extracting information from the text of electronic medical records to improve case detection: a systematic review“, *J. Am. Med. Inform. Assoc.*, kd 23, nr 5, lk 1007–1015, sept 2016, doi: 10.1093/jamia/ocv180.

[2] W. X. Zhao *et al.*, „A Survey of Large Language Models“. arXiv, 24. november 2023. Vaadatud: 23. aprill 2024. [Online]. Available at: http://arxiv.org/abs/2303.18223

[3] M. Grootendorst, „MaartenGr/KeyBERT: BibTeX“. Zenodo, 25. jaanuar 2021. doi: 10.5281/zenodo.4461265.

[4] M. Nadim, D. Akopian, ja A. Matamoros, „A Comparative Assessment of Unsupervised Keyword Extraction Tools“, *IEEE Access*, kd 11, lk 144778–144798, 2023, doi: 10.1109/ACCESS.2023.3344032.

[5] M. Zhang, X. Li, S. Yue, ja L. Yang, „An Empirical Study of TextRank for Keyword Extraction“, *IEEE Access*, kd 8, lk 178849–178858, 2020, doi: 10.1109/ACCESS.2020.3027567.

[6] H. Petrov, „Rakendus võtmesõnade leidmiseks eestikeelsetest tekstidest“, *Tallinna Ülikooli Üliõpilaste 20212022 Õppeaasta Parimate Tead. Kogum.*, lk 158–161, 2023.

[7] „Introduction - Hugging Face NLP Course“. Vaadatud: 23. aprill 2024. [Online]. Available at: https://huggingface.co/learn/nlp-course/chapter1/1

[8] „Quickstart - KeyBERT“. Vaadatud: 23. aprill 2024. [Online]. Available at: https://maartengr.github.io/KeyBERT/guides/quickstart.html

[9] N. Firoozeh, A. Nazarenko, F. Alizon, ja B. Daille, „Keyword extraction: Issues and methods“, *Nat. Lang. Eng.*, kd 26, nr 3, lk 259–291, mai 2020, doi: 10.1017/S1351324919000457.

[10] S. Siddiqi ja A. Sharan, „Keyword and Keyphrase Extraction Techniques: A Literature Review“, *Int. J. Comput. Appl.*, kd 109, nr 2, lk 18–23, jaan 2015, doi: 10.5120/19161-0607.

[11] T. Nomoto, „Keyword Extraction: A Modern Perspective“, *SN Comput. Sci.*, kd 4, nr 1, lk 92, dets 2022, doi: 10.1007/s42979-022-01481-7.

[12] A. Hulth, „Improved automatic keyword extraction given more linguistic knowledge“, *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, Not Known: Association for Computational Linguistics, 2003, lk 216–223. doi: 10.3115/1119355.1119383.

[13] E. Papagiannopoulou ja G. Tsoumakas, „A review of keyphrase extraction“, *WIREs Data Min. Knowl. Discov.*, kd 10, nr 2, lk e1339, märts 2020, doi: 10.1002/widm.1339.

[14] A. Ushio, F. Liberatore, ja J. Camacho-Collados, „Back to the Basics: A Quantitative Analysis of Statistical and Graph-Based Term Weighting Schemes for Keyword Extraction“, 2021, doi: 10.48550/ARXIV.2104.08028.

[15] Z. Nasar, S. W. Jaffry, ja M. K. Malik, „Textual keyword extraction and summarization: State-of-the-art“, *Inf. Process. Manag.*, kd 56, nr 6, lk 102088, nov 2019, doi: 10.1016/j.ipm.2019.102088.

[16] S. Qaiser ja R. Ali, „Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents“, *Int. J. Comput. Appl.*, kd 181, nr 1, lk 25–29, juuli 2018, doi: 10.5120/ijca2018917395.

[17] A. Onan, „Two-Stage Topic Extraction Model for Bibliometric Data Analysis Based on Word Embeddings and Clustering“, *IEEE Access*, kd 7, lk 145614–

145633, 2019, doi: 10.1109/ACCESS.2019.2945911.
[18] B. Issa, M. B. Jasser, H. N. Chua, ja M. Hamzah, „A Comparative Study on Embedding Models for Keyword Extraction Using KeyBERT Method“, *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia: IEEE, okt 2023, lk 40–45. doi: 10.1109/ICSET59111.2023.10295108.
[19] A. Kilgarriff *et al.*, „The Sketch Engine: ten years on“, *Lexicography*, kd 1, nr 1, lk 7–36, juuli 2014, doi: 10.1007/s40607-014-0009-9.
[20] A. Vaswani *et al.*, „Attention Is All You Need“, 2017, doi: 10.48550/ARXIV.1706.03762.
[21] „Simple maths with keywords and terms | Sketch Engine“. Vaadatud: 29. aprill 2024. [Online]. Available at: https://www.sketchengine.eu/documentation/simple-maths/
[22] I. Hein ja J. Kallas, „Eesti keele ühendkorpuse 2021 lemmade ja sõnavormide sagedusloendid.“, apr 2022, doi: 10.15155/3-00-0000-0000-0000-08D1FL.
[23] R. Mihalcea ja P. Tarau, „TextRank: Bringing Order into Texts“, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, D. Lin ja D. Wu, Toim, Barcelona, Spain: Association for Computational Linguistics, juuli 2004, lk 404–411. Vaadatud: 23. aprill 2024. [Online]. Available at: https://aclanthology.org/W04-3252
[24] L. Page, S. Brin, R. Motwani, ja T. Winograd, „The PageRank Citation Ranking : Bringing Order to the Web“, esitatud The Web Conference, nov 1999. Vaadatud: 23. aprill 2024. [Online]. Available at: https://www.semanticscholar.org/paper/The-PageRank-Citation-Ranking-%3A-Bringing-Order-to-Page-Brin/eb82d3035849cd23578096462ba419b53198a556
[25] F. Barrios, F. López, L. Argerich, ja R. Wachenchauzer, „Variations of the Similarity Function of TextRank for Automated Summarization“. arXiv, 10. veebruar 2016. doi: 10.48550/arXiv.1602.03606.
[26] T. Lin, Y. Wang, X. Liu, ja X. Qiu, „A survey of transformers“, *AI Open*, kd 3, lk 111–132, jaan 2022, doi: 10.1016/j.aiopen.2022.10.001.
[27] „Tokenizers - Hugging Face NLP Course“. Vaadatud: 25. aprill 2024. [Online]. Available at: https://huggingface.co/learn/nlp-course/chapter2/4
[28] J. Devlin, M.-W. Chang, K. Lee, ja K. Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“, *CoRR*, kd abs/1810.04805, 2018, [Online]. Available at: http://arxiv.org/abs/1810.04805
[29] A. S. Eesa ja W. Kh. Arabo, „A Normalization Methods for Backpropagation: A Comparative Study“, *Sci. J. Univ. Zakho*, kd 5, nr 4, lk 319, dets 2017, doi: 10.25271/2017.5.4.381.
[30] V. Sanh, L. Debut, J. Chaumond, ja T. Wolf, „DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter“, 2019, doi: 10.48550/ARXIV.1910.01108.
[31] H. Tanvir, C. Kittask, S. Eiche, ja K. Sirts, „EstBERT: A Pretrained Language-Specific BERT for Estonian“, *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, S. Dobnik ja L. Øvrelid, Toim, Reykjavik, Iceland (Online): Linköping University Electronic Press, Sweden, mai 2021, lk 11–19. Vaadatud: 29. aprill 2024. [Online]. Available at: https://aclanthology.org/2021.nodalida-main.2
[32] A. Conneau *et al.*, „Unsupervised Cross-lingual Representation Learning at Scale“, 2019, doi: 10.48550/ARXIV.1911.02116.
[33] M. Ulčar *et al.*, „Evaluation of contextual embeddings on less-resourced

languages“, 2021, doi: 10.48550/ARXIV.2107.10614.
[34] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, ja W. Wang, „Language-agnostic BERT Sentence Embedding“, 2020, doi: 10.48550/ARXIV.2007.01852.
[35] N. Reimers ja I. Gurevych, „Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks“. arXiv, 27. august 2019. doi: 10.48550/arXiv.1908.10084.
[36] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, ja F. Wei, „Multilingual E5 Text Embeddings: A Technical Report“, 2024, doi: 10.48550/ARXIV.2402.05672.
[37] A. Kõnno ja K. Allkivi-Metsoja, „The role of metadata in interpreting media data as cultural data: The ERR case“, 2023, Vaadatud: 30. aprill 2024. [Online]. Available at: https://www.etis.ee/Portal/Publications/Display/0565174d-fed6-46df-bd15-b0457a35fae0
[38] A. Olev ja T. Alumäe, „Estonian Speech Recognition and Transcription Editing Service“, *Balt. J. Mod. Comput.*, kd 10, nr 3, 2022, doi: 10.22364/bjmc.2022.10.3.14.
[39] G. I. Ivchenko ja S. A. Honov, „On the jaccard similarity test“, *J. Math. Sci.*, kd 88, nr 6, lk 789–794, märts 1998, doi: 10.1007/BF02365362.
[40] L. R. Lawlor, „Overlap, Similarity, and Competition Coefficients“, *Ecology*, kd 61, nr 2, lk 245–251, apr 1980, doi: 10.2307/1935181.
[41] „Eesti keele käsiraamat“. Vaadatud: 1. mai 2024. [Online]. Available at: https://www.eki.ee/books/ekk09/index.php?p=6&p1=2
[42] K. Uiboaed, „Eesti keele stoppsõnad / Estonian stop words“, apr 2018, doi: 10.15155/RE-48.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[18]

Mina, Herman Petrov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Suurte keelemudelite rakendamine võtmesõnade tuvastamiseks eestikeelsetest tekstidest", mille juhendaja on Ahti Lohk ja kaasjuhendaja Kais Allkivi-Metsoja
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

10.05.2024

18 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.