

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**PERSONĪGO ZINĀŠANU PĀRVALDĪBAS IESPĒJAS  
AR MOBILĀ TELEFONA LIETOTNES PALĪDZĪBU**

MAGISTRA DARBS

**Autors: Fjodors Pohodņevs**

Stud. apl. Nr.: fp12001

Darba vadītājs: Dr.dat. Uldis Straujums

RĪGA 2018

## ANOTĀCIJA

Pasaule ir ļoti sarežģīti aptverama un izzināma vieta. Ikdienā cilvēki uztver ļoti daudz informācijas, kas netiek organizēta. Personīgo zināšanu pārvaldība palīdz saprast, kas ir svarīgs, pārvietot informāciju un organizēt to, lai tiktu galā ar problēmām, kas izveidojas, prasot risinājumu.

Darbā tiek apskatītas esošās pieejas – modeļi, sistēmas. Lai risinātu šo problēmu, tiek apskatīts veids, kā izstrādāt Android telefonu lietotni, kas izmantojot personības testu, nosaka ar personību saistītos atslēgvārdus, lai iegūt datus no kāda avota un izmanto NLP metodes, lai pakārtotu informāciju, kas palīdzētu iegūt konkrētam indivīdam būtisku informāciju.

**Atslēgas vārdi:** Android, personīgā zināšanu pārvaldība, NLP, MBTI.

## **ABSTRACT**

### *Personal knowledge management usability using mobile phone application*

The world is a very complicated place. Every day people perceive a lot of information that is not organized. Managing your personal knowledge helps you understand what is important, move information and organize it to deal with the challenges.

The thesis deals with existing approaches - models, systems. To address this problem, it is researched on how to design an Android phone app, that could use personality test to determinate personality type and its associated keywords, and use NLP methods to organize information, so that particular individual could acquire relevant information.

**Keywords:** Android, personal knowledge management, NLP, MBTI

## AUTOREFERĀTS

No maģistra darba aprakstītais autors ir paveicis sekojošo:

- Izpētījis literatūru saistībā ar personīgo zināšanu pārvaldību;
- Izpētījis sistēmas un veidus ko var sasaistīts ar personīgo zināšanu pārvaldību;
- Izstrādājis Android lietotni, kurā var izpildīt personības testu un balstoties uz personības tipu iegūt nepieciešamo informāciju;
- Noformējis darbu, vadoties pēc *Maģistra darba izstrādes un aizstāvēšanas metodiskajiem norādījumiem* un atbilst tā norādēm un vadlīnijām.

# SATURS

ANOTĀCIJA.....	2
ABSTRACT .....	3
AUTOREFERĀTS .....	4
SATURS.....	5
APZĪMĒJUMU SARAKSTS.....	7
IEVADS .....	9
1. PERSONĪGO ZINĀŠANU PĀRVALDĪBA .....	12
1.1. Uzkrāšana.....	12
1.2. Organizēšana.....	12
1.3. Dalīšanās.....	13
2. PERSONĪGĀS ZINĀŠANAS PĀRVALDĪBAS ĪSTENOŠANA.....	14
2.1. Izaicinājumi .....	14
2.2. Informācijas uzkrāšana .....	15
2.3. Informācijas organizēšana .....	16
2.4. Informācijas pieprasīšana .....	18
2.5. Informācijas vizualizēšana.....	19
3. Palīg līdzekļi personības zināšanas pārvaldībai .....	21
3.1. Personības tipi.....	21
3.2. News API.....	23
3.3. Google Cloud Natural Language .....	24
4. ANDROID LIETOTNES IZSTRĀDE.....	28
4.1. Izstrādes process .....	28
4.2. Lietotnes darbība.....	28
4.3. Problēmas un risinājumi izstrādājot lietotni .....	40
REZULTĀTI .....	44
SECINĀJUMI .....	45
IZMANTOTĀ LITERATŪRA UN AVOTI.....	46



## APZĪMĒJUMU SARAKSTS

Apzīmējums	Apraksts
PKM	<i>Personal Knowledge Managment</i> Procesi, kas palīdz uzkrāt, organizēt und dalīties ar informāciju
Prāta karte	Karte, kur pa vidu ir ideja, kas sasaistās ar apakšidejām
Android	Mobilo ierīču operētājsistēma
RDF	<i>Resource Description Framework</i> Datu modelis resursu aprakstīšanai
SPARQL	<i>Protocol and RDF Query Language</i> Vaicājumu valoda semantiskā tīmekļa datu avotiem
HTML	<i>Hyper Text Markup Language</i> Hiperteksta iezīmēšanas valoda
HTTP	<i>Hyper Text Transfer Protocol</i> Hiperteksta pārsūtīšanas protokols
NLP	<i>Natural Language Proccesing</i> Zinātne, kas nodarbojas ar dabiskās valodas pārvēršanu par informāciju, ko saprot dators
n-gram	N elementu sekošana secībā no konkrēta teksta vai runas parauga
TF-IDF	<i>term frequency–inverse document frequency</i> Skaitliska statistika, kas paradzēta, lai atspoguļotu cik svarīgs vārds ir dokumentu kolekcijā
PageRank	Algoritms, ko izmanto <i>Google Search</i> , lai novērtētu mājaslapas Google meklētājprogrammas rezultātos

JSON	<p><i>JavaScript Object Notation</i></p> <p>Minimāli lasāms formāts strukturētiem datiem, ko izmanto, lai sūtītu datus starp serveri un lietotni</p>
API	<p><i>Application Programming Interface</i></p> <p>Funkciju un procedūru kopums, kas ļauj izveidot lietotni, kas piekļūst pie datiem no operētājsistēmas, lietotnes vai cita servisa</p>
REST	<p><i>Representational State Transfer</i></p> <p>Programmatūras arhitektūras stils, kas nosaka, kā darbosies HTTP protokola funkcijas</p>
MBTI	<p><i>Myers-Briggs Type Indicator</i></p> <p>Personību teorija</p>

## IEVADS

Cilvēkam ir nepārtraukti sevi jāpilnveido, lai kļūtu par labāku, zinošāku, kreatīvāku personu, kas spēj realizēt savu potenciālu pilnībā, darbam, kā arī vispārīgi - jebkādām situācijām un izaicinājumiem atbilstoši, konkurētspējīgā veidā. Lai uzlabotu sevi, ir nepieciešams apgūt gandrīz bezgalīgu informācijas klāstu, iemācīties paņēmienu, saprast metodes un galvenais – likt lietā to, kas var noderēt nospraustā mērķa realizācijai. Nav svarīgi kādā sfērā – tas varētu būt IT, ēdienu gatavošana, veselība, sports, dejošana, finanses, utt.. Bieži vien tas nav tik vienkārši. Ikdienā mēs uzņemam lielāko daļu informācijas no interneta. Patiesībā tā ir sadrumstalota, stipri fragmentāra informācija, tāda, ko mēdz dēvēt – ziņas un padomi “par visu ko”. Šo informāciju ir grūti sakārtot tā, lai to veiksmīgi pārvērstu par zināšanām, ko var sekmīgi pielietot dzīvē. Būtībā tas ir un paliek “informatīvais troksnis”, ja nerodam veidu kā ar to strādāt, kā atšķelt nevajadzīgo un – kā saglabāt vajadzīgo tā, lai informācija vēlāk būtu atrodama un viegli pieejama.

Bieži vien gadās ielikt informāciju pārlūka grāmatzīmēs vai arī saglabāt saiti uz kādu lapu, kur ir noderīga informācija, vai vismaz norāde uz to, kur meklēt tālāk, ja, piemēram, informācijas ievākšana uz brīdi apturēta. Ar laiku informācija, kas var būt arī ļoti būtiska, sakrājas un ir nepārredzama, tādējādi vairs nav izmantojama un burtiski “krāj putekļus”, kļūstot par savā ziņā nepieejamu un neizmantojamu resursu.

Tādēļ, ka viss ir samests vienā maisā – informācija nav strukturēta, pārskatāma un viegli pieejama, vairs nevar saprast to, kāda saite atbilst kādai sfērai, kas kopīgs, cik katra lieta ir svarīga, utt... Un tā uzkrātais, kā iegūšanai ticis veltīts ne mazums laika, kļūst par bezjēdzīgu informāciju, ko nevar pārvērst par zināšanām. Būtiski ir tas, ka novecojusi informācija pārklāj un sajaucas ar to, kas noteikti ir nepieciešama, bet kļuvusi neaizniedzama. Tā notiek arī tādēļ, ka cilvēkam ir ierobežots laiks, ko katru reizi varētu patērēt tajā brīdī nepieciešamās informācijas atrašanai. Cilvēks, meklējot kaut ko, lai papildinātu un pilnveidotu savas zināšanas, nevar dažkārt atcerēties, to, kas tas bija par faktu, paņēmienu vai metodi, kur nu vēl visu uzreiz atcerēties un noorientēties paša radītajā, nesakārtotajā datu apjomā.

Protams, informāciju joprojām arī iegūstam no grāmatām vai arī ejot uz kādu, profesijai vai interešu sfērai atbilstošu, konferenci. Problēma joprojām paliek aktuāla. Jo atcerēties visu informāciju ir grūti, turklāt, arī rūpīgi veiktajiem pierakstiem ir tendence nozust. Parasti arī labi strukturētajos prezentāciju materiālos, kas tikuši sagatavoti un, piemēram, pēc konferences atsūtīti ē-pastā, vēlāk ieskatās reti. Tā vēl viens svarīgs informācijas fragments nogulstas, kā kārtējie liekie faili, datu nesējā. Būtībā informācija, kas nepieciešama, lai veicinātu mācīšanās procesu, izveido nepārskatāmu slāni, kurā ne tikai grūti orjentēties, bet, kas var kļūt gandrīz vai

par attīstību bremsējošu faktoru, jo cilvēks, piemēram, maldīgi domā, ka spēš vajadzīgajā brīdī atrast nepieciešamo. Taču tajā brīdī, kad šī nepieciešamība rodas, parasti laika periods, kas atvēlēts problēmas atrisināšanai, ir par īsu, lai veiktu laikietilpīgu atcerēšanās un meklēšanas procesu. Tā cilvēks var pieņemt, ka zina risinājumu, jo “kaut kur taču man tā informācija bija”. Tā rezultātā var tik pieņemt absolūti kļūdainus lēmumus. Tā ir vēl viena problēma, ko rada nestrukturēts informācijas apjoms – zināšanu ilūzija. Cilvēkam šķiet, ja jau ir uzkrāta informācija, kas teorētiski ir pieejama, tad jau, pat, ja nav laika atrast precīzi vajadzīgo un atbilstošo, pieņemsim, ka es to zinu, jo ir taču kaut kur tas informācijas fragments... Tas savukārt secīgi noved pie nepareiziem lēmumiem, nevis uz zināšanām balstītiem, risinājumiem.

Mācīšanās, kur izmantota strukturēta, saprotama un pieejama informācija, noved pie zināšanām, ne ilūzijām par to, ko šķietami it kā zinām. Tā ir būtiska lieta, ja spriežam par darbu ar informāciju un problēmām, kas var rasties. Zināšanas palīdz pieņemt atbilstošākus un labākus lēmumus, paveikt uzdevumu ātrāk, atrisināt problēmu veiksmīgāk. Būt pārliecinātiem par sevi, savām prasmēm un iespējām. Maldīgi pieņēmumi, par it kā zināšanām, kas balstītas uz nepieejamu un neizmantojamu informāciju, kuru, piemēram, atceramies tikai fragmentāri, bet dažādu iemeslu dēļ nevaram pārbaudīt, – rada jaunas problēmas, labākajā gadījumā, tikai papildinot haosu, bet sliktākajā – var novest pie nopietnām kļūdām.

Labākais veids kā sevi, savas zināšanas un prasmes uzlabot, ir izmantot labu mācības sistēmu. Tā sistēma ir – personīgo zināšanu pārvaldības sistēma (PKM). PKM ir procesu apkopojums, ko indivīds izmanto, lai uzkrātu, strukturētu un dalītos ar zināšanām[2]. Šī sistēma palīdz indivīdam mācīties. Bet, lai izmantot šādu sistēmu, ir nepieciešami palīgriķi, kuros var organizēt informāciju, lai no tās izveidot zināšanas, un tad ar tām dalīties ar citiem.

Pastāv tāda lieta kā attēla pārākuma efekts[1], tapēc ir labi arī izmantot prāta kartes, kuras sasaita ar konkrētiem pierakstiem, kas tiek saglabāti PKM rīkos, lai vieglāk sakārtot un izprast informāciju, kura radīs zināšanas.

Darbā tiek izpētīti esošie rīki, stratēģijas, ko izmanto PKM darbībai, un veids kā izstrādā Android lietotni, ko var izmantot konkrēta indivīda PKM.

Maģistra darba mērķi ir:

- Apkopot informāciju par personīgo zināšanu pārvaldību;
- Izpētīt sistēmas, ko var izmantot, lai intigrētos ar PKM;
- Balstoties uz iegūto informāciju, izstrādāt Android lietotni.

Darbs sastāv no teorijas apskata un pētnieciskās daļas. Teorijas apskata mērķis ir iegūt informāciju par personīgo zināšanu pārvaldību, esošiem risinājumiem un sistēmām, kas var palīdzēt to realizēšanā. Pētnieciskās daļas mērķis, ir izpētīt un izdarīt secinājumus par to, ko

var un ko nevar izdarīt PKM sfvērā, izstrādājot lietotni, izmantojot personība tipus un NLP metodes, balstoties uz informācijas daudzveidību atkarībā no personības tipa.

Darba iegūtajos rezultātos tiek apzināts, kas ir PKM, kādi ir esošie risinājumi. Kā arī, kā var integrēt citas sistēmas un idejas – šajā gadījumā personības tipus, lai izstrādāt PKM Android lietotni. Veicot Android lietotnes izstrādi tika izdarīti secinājumi, par to, ko var un ko nevar izstrādāt PKM ar Android lietotnes palīdzību. Kā arī tas, ka personības testa jautājumi var būt par daudz un ir nepieciešams pieeja īsākam testam. Un tas, ka paši pa sevi atslēgvārdi nav pietiekoši labi, ja lietoti vieni paši, lai izmantotu datu pieprasīšanai, jo tika iegūti līdzīgi rezultāti starp personību tipiem. Minētā situācija situācija izveidojās ar laiku.

# 1. PERSONĪGO ZINĀŠANU PĀRVALDĪBA

## 1.1. Uzkrāšana

Pastāv diva veida informācija – aktīvā un pasīvā. Aktīvā informācija ir, tas ko mēs paši meklējam, piemēram veicot pieprasījumus Google. Bet pasīvā ir tā, ko mēs saņemam no citiem sūtītājiem, piemēram, Twitter.[5]

Šajā brīdī ir vērtīgi novērtēt to, kāda informācija ir svarīga, lai saglabātu lasīšanai nākotnē, kad tas būs nepieciešams lietotājam. Šajā brīdī ir noderīgi rīki, kas palīdz pārvērst informāciju vēl lasāmāka formātā, piemēram pdf failā, un bez visas tās liekās informācijas, kas parasti ir mājaslapā lasot blogus (piemēram reklāmas), un saglabāt viegla pieejamā vietā (telefonā vai arī makonī). Vizuālās reklāmas, nereti arī ilustrācijas, ja tikai tām nav kādas tekstu papildinošas funkcijas, būtībā ir lieka informācija, kas piesārņo vidi, kā arī pārsātina uztveri. Ir grūti meklēt, piemēram, vienu vajadzīgo teksta rindu, kas burtiski pazūd starp lapas dizaina elementiem un košām reklāmām. Turklāt nevajadzīgas informācijas fragmenti novērš uzmanību un traucē fokusēties uzdevuma veikšanai. [5]

## 1.2. Organizēšana

Zināšanu organizēšanai ir nepieciešama ļoti individuāla pieeja, lai izprastu jēgu no iegūtās informācijas. Un ir atkarīgs no tā, pie kā tiek strādāts, par ko atbildīgs un, kas tiek dotā brīdī mācīts. Organizēšana galvenokārt sastāv no klasifikācijas un papildināšanas. Papildus svarīgāko punktu izcelšanai piezīmēs, vēl ir svarīgi paplašināt noteiktu saturu vai vārdus, lai palīdzētu saprast dziļāku domu. Piemēram, ja ir rinda terminu, kas nav skaidri, tad skaidrojumu var sameklēt internetā, un tad var piezīmēt skaidrojumu.[5]

Pēc informācija apstrādes, tā tiek saglabā konkrētā vietā, un tad to var klasificēt, piemēram izmantojot vārdšķiru etiķešanu, atbilstošā kategorijā. Piezīmes tiek saglabātas vairākās sekcijās, atbilstoši tēmai. [5]

Vēl ir svarīgi regulāri pārskatīt sakārtotos materiālus, lai neizveidojas situācija, ka sarāžotos daudz neatbilstošā informācija. Pārskatot materiālus ir svarīgi izprast, vai atbilstošā piezīme atbilst tēmai, kādu problēmu tā var risināt, un vai ir joprojām atbilstoša. Vai informācija jau nav manījusi, nav novecojusi. [5]

Svarīgi paturēt prātā, ka informācijas vide ir dinamiska, vienmēr mainīga un diezgan nestabila. Piemērs, ir jebkuras datoru programmas, aplikācijas utml. Nepātraukti tiek veikti papildinājumi, uzlabojumi un atjauninājumi. Tas vienmēr ir siasstīts ar to, ka rodas jauna

informācija, tiek pārstrādāta, arī atmesta, jau esošā. Informācijas organizēšana vienmēr būs dinamisks process, kas ir nebeidzams.[5]

### **1.3. Dalīšanās**

Dalīšanās ir ļoti svarīgs PKM posms. Dalīšanās var izpausties kā raksta rakstīšana, publiskā uzstāšanās, stāsta stāstīšana, informācijas fragmentu sūtīšana no vienas ierīces uz otru, no viena indivīda – citam, piemēram, ja kāds cilvēks personīgi lūdz padomu par konkrētu tēmu. Tas piespiež strukturēt un pārorganizēt zināšanas, ko esam apguvuši, lai to attēlotu saprotamā veidā citiem cilvēkiem. Pēc šī informācijas apstrādes posma var teikt ka indivīds ir apguvis zināšanas, jo mēs tās ne tikai uzzinājām, bet arī izpratām un izmantojām.[5]

Jāatceras, ka labi pārvaldīts un strukturēts saturs ir tāda informācija, kas būs vērtīga, viegli izmantojama un veicinās zināšanu apgūšanu ilgtermiņā. Kā arī ar to būs ērti un viegli dalīties ar citiem indivīdiem.

## 2. PERSONĪGĀS ZINĀŠANAS PĀRVALDĪBAS ĪSTENOŠANA

Cilvēki ikdienā uztver ļoti daudz informācijas, protams, ka cilvēks fiziski nespēj visu atcerēties, tāpēc ir nepieciešams izstrādāt sistēmas, kas cilvēkam atvieglos dzīvi. Sistēmas, kas ļautu ērtāk saglabāt informāciju un to pēc tam organizēt vizuāli patīkamā un lasāmā veidā, lai cilvēks šo informāciju varētu izmantot kā zināšanas. Un spētu pēc tam arī ar šīm zināšanām dalīties ar citiem indivīdiem, kā arī – kopumā arī ar plašāku sabiedrību.

### 2.1. Izaicinājumi

Mūsdienās mēs kā cilvēki strauji attīstāmies un lēnām kļūstam par daļu no zināšanu sabiedrības. Bet, lai tādu sabiedrību īstenotu, ir nepieciešami personīgie zināšanu pārvaldības rīki, kas mums palīdzētu. Diemžēl, mūsdienas tādu trūkst, turklāt tie bieži vien ir pārāk specifiski.

Lai izstrādātu labu personīgo zināšanu pārvaldības sistēmu, ir nepieciešams tikt galā ar konkrētiem izaicinājumiem. Personīgo zināšanu pārvaldībā izaicinājumi tiek sadalīti četros līmeņos:

- Saglabāšanas līmenis;
- Sadarbības līmenis;
- kapacitātes attīstības līmenis;
- konceptuālais līmenis. [10]

Saglabāšanas līmenī problēma ir tajā, ka mēs informāciju saglabājam dažādās ierīcēs vai arī saglabājam informāciju galvā. Un ar laiku ierīces sabojājas un atmiņa pazūd. Tādēļ pazūd iespēja atcerēties informācijas atrašanās vietas, kā arī tās saturu mūsu personīgās zināšanu krājumos un arhīvos.

Arī sadarbības līmenī viss nav tik vienkārši kā varētu vēlēties. Pastāv dažāda veida rīki, kas palīdz ar ātru piekļuvi resursu plāniem, progresā ziņojumiem un komandas dalībnieku ieguldījumiem nepabeigtajos darbos. Tomēr trūkst tas, ka zināšanu pārvaldības sistēmās iztrūkst iesaistīto personu personīgo ieguldījumu iznākums. Mērķis ir kopīgi sazināties un kopīgi savākt, iepriekš iegūto, personīgo zināšanu apakškopas.

Kapacitātes attīstības līmenī nepieciešamas fokusēties uz karjeras progresā kultivētajām zināšanām ekonomikā, kā arī to sarežģītajām problēmām daudznazaru telpā. Lai gūtu panākumus šajā līmenī, ir nepieciešama vairāk pārdomāta, apņēmīga un enerģiskāka pieeja intelektuālajam kapitālam. Esošie rīki šobrīd ir vairāk uzņēmumu orientēti, bet prioritāri ir

personalizētās zināšanu bāzes veicināšanai - noderīgi indivīda izglītības un profesionālās karjeras attīstībai.

Konceptuālā līmenī abstraktie modeļi un uzlabotās prakse var gan vadīt, gan maldināt jauno iegūto datu, informācijas un pieredzes integrēšanu personīgās zināšanas ietvaros. Informācijas avotu apjoms visu laiku palielinās. Tādēļ tiek zaudēts laiks meklējot vērtīgo informāciju, jo bieži vien jātiek galā ar lieko informāciju un informācijas kārtošanu, organizēšanu, lai gan šajā laikā varētu fokusēties uz kreatīviem un inovatīviem mērķiem.[10]

## **2.2. Informācijas uzkrāšana**

Lai iegūtu zināšanās, jāsāk ar to, ka ir jāiegūst nepieciešamā informācija, kuru varētu organizēt. Protams, to lietotājs var darīt, ievadot informāciju pats ar roku, rakstot piezīmes, saglabājot bildes, vai arī saglabājot adresi uz konkrētu saiti, kas lietotājam interesēja, bet nebija laika kārtīgi izlasīt, vai arī bija vēlme saglabāt, lai vēlreiz pārlasītu vēlāk. Var ekonomēt laiku, piemēram, saglabāt mājaslapu iezīmēs, bet bieži vien veidojas tāda situācija, ka tiek saglabātas ļoti daudz mājaslapu iezīmēs, līdz to sakrājas simtiem. Iezīmju ir daudz, tā ir nesaorganizētas. Informācijas uzkrājējs pats vairs neko nevar saprast, kāpēc vispār tāda informācija tika saglabāta, un tā arī netika izmantota. Proti, indivīds vairs neorientējas tajā, ko viņš uzkrājis, lai arī, visticamāk tas bijis svarīgs, piemēram, nepieciešams darbam.

Problēma slēpjās tajā, ka cilvēkiem ir maz laika vai arī nav vēlmes un zināšanu kā kaut ko tādu darīt parocīgi izmantošanai nākotnē - proti sistematizējot. Bieži vien, klejojot cauri internetam, cilvēki ikdienu iziet cauri milzīgam daudzumam informācijas. Tāpēc ir vēlams automatizēt informācijas iegūšanas un apkopošanas procesu.

Viens no risinājumiem ir izvilkt konkrēto informāciju, ko tajā brīdī lietotājs apskata. To var darīt izvelkot mājas lapas HTML un vienkāršo teksta saturu, kā arī saglabāt mājaslapas momentuzņēmumu.[12]

Mājaslapā bieži vien ir vienkāršais teksts, kas mums neinteresē. Tie būtu navigācijas elementi, reklāmas, sānjoslas, kājene, galvene – šādi elementi nesatur informāciju, kas mums ir nepieciešama, tāpēc vēlams, lai šī informācija netiktu uzkrāta, bet gan tiktu ignorēta. Izmantojot readability algoritmu, mēs varam tikt vaļā no visas liekas informācijas un iegūt vienkāršo tekstu, kas atbilst mājaslapas saturam.[13]

Tādējādi lietotājam vairs nav jādomā par to, ko viņš vēlas vai nevēlas saglabāt. Indivīdam jāveic vienkārša darbība - jāieslēdz spraudnis, kas klausīsies lietotāja darbības un saglabās svarīgo informāciju, kas bija konkrētajā mājas lapā. Tālāk, jau apstrādātā, no visa lielā atbrīvotā informācija, tiks padota sistēmai, lai tā tiktu organizēta.

Otrs variants ir nodrošināt sistēmai sarakstu ar datu avotiem. Un šie datu avoti būtu: e-pasts, kalendāri, informācijas atrašanās vieta vai arī adresu grāmatiņas. Kā arī nodrošināt marķieri vai paroli, lai sistēma spētu piekļūt šiem datiem. Tādējādi sistēma spētu iegūt visu nepieciešamo informāciju no šiem avotiem.[16]

Šajā gadījumā, lai sistēma spētu uzkrāt informāciju kādā konkrētā vietā, nepieciešams tai sniegt esošus datu avotus. Lai izveidotos datu avoti, nepieciešami dati ar ko strādāt. Tāpēc lietotājam nepieciešams veikt kādu, paša izvēlētu datu, saglabāšanu, lai vēlāk sistēma tos varētu apvienot zem viena avota. Atrašanās vietas vēstures gadījumā nepieciešams tikai tas, lai lietotājs lietotnē tur ieslēgtu ģeo lokācijas funkcionalitāti. Ja tā tiks ieslēgta, tad telefons automātiski uzņems informāciju un saglabās atrašanās vietas vēsturi piesaistītajā kontā, kas ir uz telefona. Gadījumā, ja tiek strādāts ar notikumiem un to laikiem, vēlams, lai lietotājs to atzīmē kalendārā. E-pasti un kontakti ir pašsaprotama lieta, kas lietotājam jau būs. Tāpēc, neskaitot kalendāra ierakstus, un to, ka jatur ģeo lokācijas funkcionalitāte ieslēgta, nav nepieciešama papildus piepūle, lai aprakstītu notikumus sīkās detaļās nākotnei.[16]

### 2.3. Informācijas organizēšana

Šī ir svarīgākā sastāvdaļa personīgās zināšanas sistēmas pārvaldībā. Lai varētu organizēt informāciju un pārvērst to par zināšanām, tad jāizlemj veids kā organizēt šādus informāciju.

Viens no risinājumiem ir izmantot NLP (atšifrē, kas tas ir). Iegūto informāciju apstrādāt ar NLP metodēm:

Šī ir svarīgākā sastāvdaļa personīgās zināšanas sistēmas pārvaldībā. Lai varētu organizēt informāciju un pārvērst to par zināšanām, tad jāizlemj veids kā organizēt šādus informāciju.

Viens no risinājumiem ir izmantot NLP (atšifrē, kas tas ir). Iegūto informāciju apstrādāt ar NLP metodēm[8]:

- Teikumu noteikšana – sadala tekstu teikumos pēc interpunkcijas, piemēram, ja ir kārtas skaitlis, kur parādās punkts, tad to daļu nedala atsevišķi;
- Marķēšana – sadala teikumus marķieros – pa vienam vārdam vai simbolam;
- Vārdu meklēšana – sameklē konkrētus vārdus;
- Vārdšķiras etiķēšana (*tagging*) – atgriež vārdus ar vārdšķiras etiķetes palīdzību;
- Čankošana (*chunking*) – sadala tekstu sintaktiski korelētajos vārdšķiras vārdos;
- Parsēšana – pārveido teikumu parsētā kokā ar vārdšķiras etiķetēm [14, 15, 8].

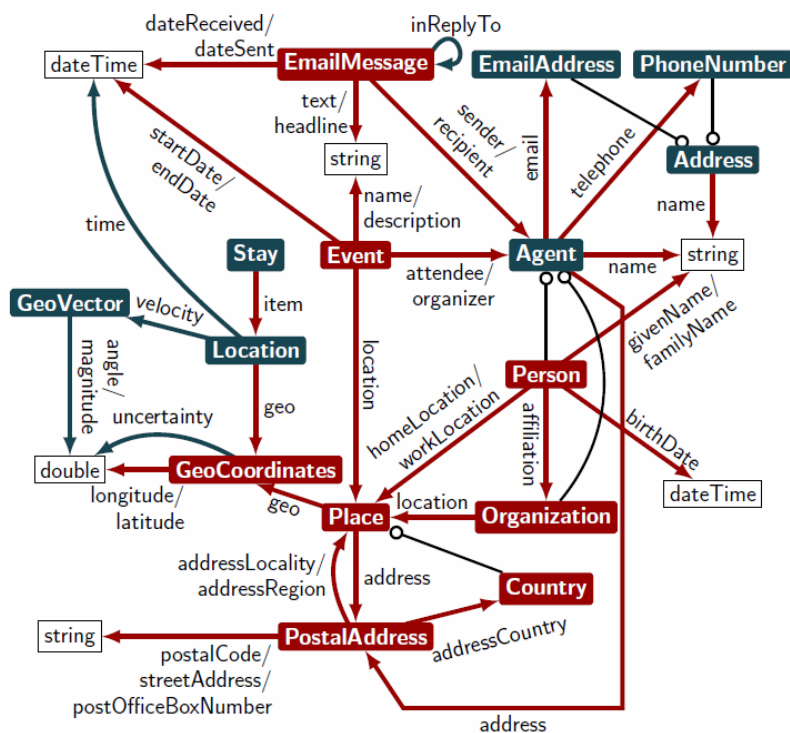
Sadalot informāciju šādos trijos tipos:

- Nosauktās entītijas;
- Atslēgvārdi;
- Attiecības starp entītijām.

Svarīgākais informācijas tips ir atslēgvārdi. Šajā gadījumā tie saturēs informāciju, kurai ir liela vērtība. Tāpēc ir svarīgi izvēlēties pareizu atslēgvārdu. Atslēgvārdiem jābūt īsiem un kodolīgiem, tapēc vēlām limitēt n-grams līdz trīs. Katrs n-grams, kas pārvar TF-IDF punktu minimālo sliekšni, tiek pievienots entītijū sarakstam ar tipu "KEYWORD". Katru reizi, kad tiek apskatīta jauna informācija, tad izmantojot TF-IDF mērīšanu, tiek ietekmēti esošie atslēgvārdi. Pastāv iespēja, ka punkti var samazināties, dēļ kā konkrēts atslēgvārds būs zem minimālo punktu sliekšņā un tiks izmests no entītijū saraksta. Tādējādi entītijū saraksts, lietotājam iegūstot jaunu informāciju, tiek nepārtraukti atjaunots,.

Attiecības starp entītijām tiek izveidotas, ja entītijas vai atslēgvārdi ir bijuši vienā teikumā.[12]

Otrs risinājums ir izmantot RDF zināšanu attēlošanai. Iegūti dati no dažādiem datu avotiem tiek pārvērsti RDF formātā un tiek saglabāti jaunā avotā. Tiek izmantoti speciāli



2.1. att. Thymeflow personīgais datu modelis [16]

adapteri, lai pārvērst informāciju RDF shēmā. Daļu no RDF shēmas Thymeflow sistēmā var redzēt 2.1. attēlā, kur redzams kā savā starpā ir sasaistītas entītijas.[16]

Lai veiksmīgi sasaistīt informāciju, tiek izmantoti bagātinātāju moduļi. Tie uzsāk darbu pēc lejuplādēšanas fāzes. Šie moduļi veic dažāda secinājuma uzdevumus – entītiju regulējumi, notikumu ģeolokācijas un citus zināšanu bagātināšanas uzdevumus. Bagātinātāju moduļi veic darbības, katru reizi, kad ir jaunas izmaiņas. Piemēram, kad tiek ievadīti jauni dati kalendārā, tad ģeokodēšanas bagātinātājs cenšas to atrast, un tad cits bagātinātājs sasaista to ar atrašanās vietas vēsturi.

Lai veiksmīgi sasaistīt datus un izveidot labu zināšanu bāzi, iesakāms izmantot šāda veida bagātinātājus:

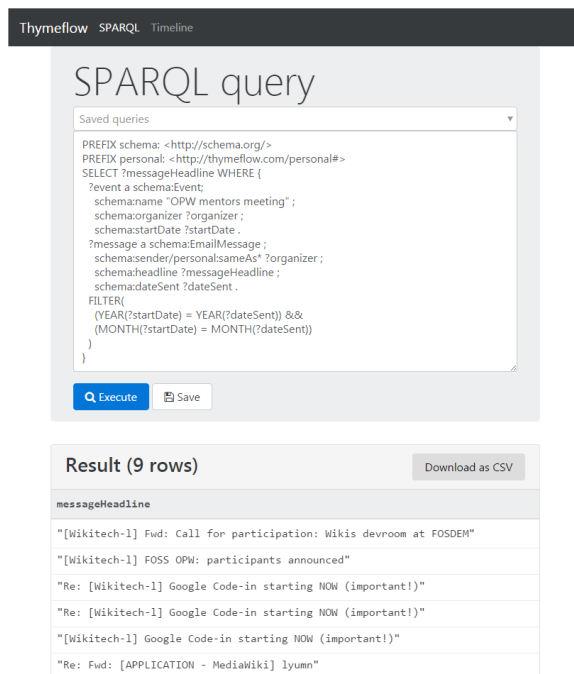
- Bagātinātājs, kurš spēj sameklēt to, ka divi aģenti ir līdzvērtīgi vienam aģentam īstajā pasaulē. Aģenti ir sasaistīti pamatojoties uz līdzīgiem atribūtiem, tādiem kā e-pasta adresei un vārdam. Tādējādi sasaistot e-pastu ar kontaktu.
- Bagātinātājs, kurš izmanto laika pamatotu algoritmu, lai noteikt vietas, kur lietotājs ir ilglaicīgi uzturējies.[11]
- Bagātinātājs, kurš izmanto iepriekšējo bagātinātāja iegūtos kalendāra notikumus un uzturēšanos, lai sasaistīt kopējos laika periodus.
- Bagātinātājs, kurš sasaita katru kalendāra notikumu ar īstām pasaules vietām (ielas adrese, dažāda veida orjentieri, utt.).

Pielietojot šādus bagātinātāju moduļus, rezultātām iegūstam zināšanas. Kuras pēc tam var iegūt veicot pieprasījumus.

## **2.4. Informācijas pieprasīšana**

Parastam lietotājam ir vēlams, lai tiktu veikta abstrakcija, jo lietotājs nezinās, kā pareizi rakstīt SQL vai SPARQL pieprasījumus. Tāpēc šī informācija jāpaslēpj, un jāļauj lietotājam rakstīt parasta teksta pieprasījumu, lai iegūtu kāda veida informāciju.

Ja sistēma izmanto RDF, tad tā var būt diezgan sarežģīta un varētu būt grūti implementēt tāda veida abstrakciju, tāpēc vēlams izmantot SPARQL valodu sistēmā, lai ļautu lietotājam izmantot SPARQL pieprasījumus. 2.2. attēlā var redzēt pieprasījumu, lai iegūtu e-pastus, ko ir izsūtījis organizātors pasākuma notikšanas mēnesī.



The screenshot shows the Thymeflow SPARQL query interface. At the top, there is a header with 'Thymeflow SPARQL Timeline'. Below it, a 'SPARQL query' section contains a text area with the following query:

```
PREFIX schema: <http://schema.org/>
PREFIX personal: <http://thymeflow.com/personal#>
SELECT ?messageHeadline WHERE {
  ?event a schema:Event;
  schema:name "OPW mentors meeting";
  schema:organizer ?organizer;
  schema:startDate ?startDate .
  ?message a schema:EmailMessage;
  schema:sender/personalsameAs* ?organizer;
  schema:headline ?messageHeadline;
  schema:dateSent ?dateSent .
  FILTER(
    (YEAR(?startDate) = YEAR(?dateSent)) &&
    (MONTH(?startDate) = MONTH(?dateSent))
  )
}
```

Below the query text are 'Execute' and 'Save' buttons. Underneath, a 'Result (9 rows)' section is shown with a 'Download as CSV' button. The results are displayed in a table with the column 'messageHeadline':

messageHeadline
"[Wikitech-1] Fwd: Call for participation: Wikis devroom at FOSDEM"
"[Wikitech-1] FOSS OPW: participants announced"
"Re: [Wikitech-1] Google Code-in starting NOW (important!)"
"Re: [Wikitech-1] Google Code-in starting NOW (important!)"
"[Wikitech-1] Google Code-in starting NOW (important!)"
"Re: Fwd: [APPLICATION - MediaWiki] Iyumn"

## 2.2. att. Thymeflow SPARQL pieprasījuma piemērs [16]

Vienīgais, ja rezultātā iegūstam ļoti daudz informācijas, piemēram 100 rezultātus, tad būtu diezgan grūti atrast nepieciešamo lietu, tāpēc vēlams, lai varētu eksportēt vizuāli patīkamā veidā.

## 2.5. Informācijas vizualizēšana

Kad tiek pieprasīta nepieciešamā informācija, tad beidzot ir nepieciešams to parādīt lietotājam, lai lietotājs šo informāciju var viegli apstrādāt un iegūt nepieciešamas zināšanas.

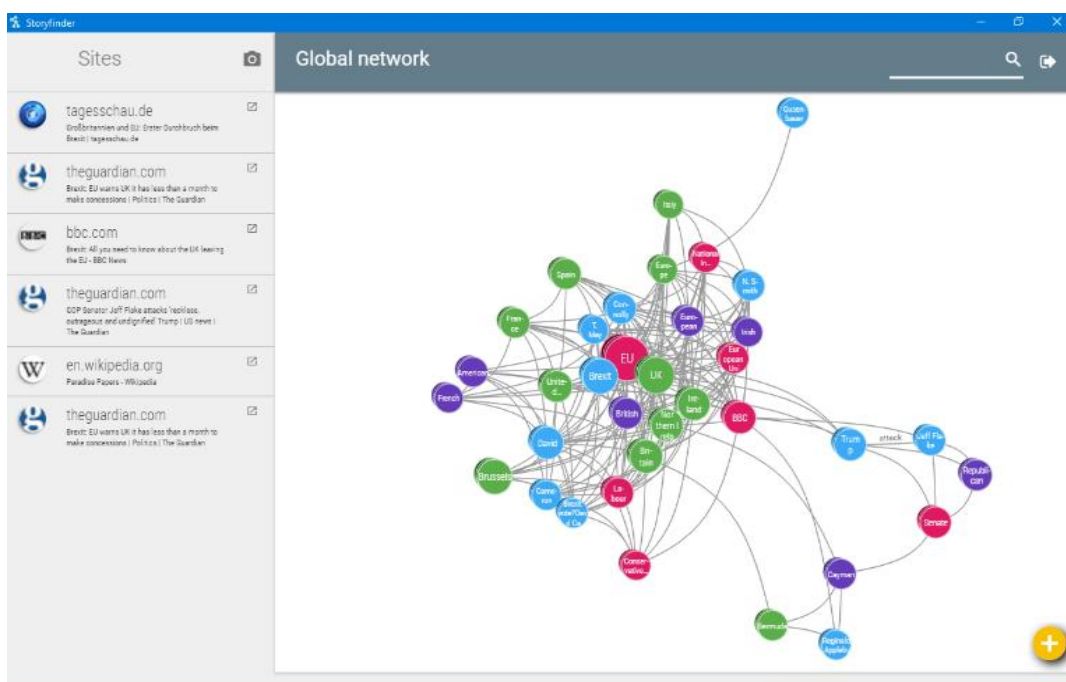
Jāspēj izvairīties no kognitīvās slodzes, kas var rasties lietotājam. Tas notiek, kad rodas informācijas pārslodzes problēmas. Šādas problēmas notiek, kad lietotājas tiek novērsts un pārslogots ar nenozīmīgu informāciju, kas tiek pasniegta konkrētam uzdevumam.

Lai izvairītos no šīm problēmām, ir nepieciešams vadītīs pēc šādām vizuālās analītikas vadlīnijām:

- Dažādas krāsas dažādām entītijām;
- Mezgli ir vizuāli grupēti pēc entītijas vairākkārtējās sastapšanas;

- Mezgla lielums apzīme nozīmīgumu relatīvi entītijai;
- Animācijas sniedz lietotājam atsauksmi uz darbībām;
- Aktīvie elementi ir centrēti un vizuāli stimulēti;
- Mezglu nosaukumi ir pielāgoti to izmēriem.

Mezglu daudzums ir atkarīgs no tā, cik vizuāli vietas ir pieejams. Prioritāte ir tiem mezgliem, kas ir svarīgākie. Svarīgākie mezgli tiek noteikti pēc *PageRank* algoritma. Rezultātā varam redzēt 2.3. attēlā StoryFinder grafa implementāciju.[12]



### 2.3. att. StoryFinder grafs [12]

Attēlā redzams, ka var ļoti veigli uztvert nepieciešamo informāciju, jo var redzēt, kas ar ko ir saistīts, kā arī tiek izmantotas vizuāli patīkamas krāsas.

### 3. Palīgīdzekļi personības zināšanas pārvaldībai

#### 3.1. Personības tipi

Šī darba ietvaros tiek apskatīts, kā var sasaistīt personības tipus un informācijas iegūšanu. Šajā gadījumā der Myers-Briggs tipa indikators (MBTI). Tā ir introspektīva aptauja, kuras mērķis ir saņemt ieskatu tajā, kā konkrēto tipu pārstāvji uztver ārējo pasauli, kā arī savējo iekšējo pasauli. Anketas rezultātā aizpildīšanas rezultātā tiek iegūts konkrēts personības tips.[8]

Atkarībā kāds ir personības tips, varētu iegūt konkrētu informācijas paveidu. Piemēram, ja, ja ir ENFJ, tad iegūst informāciju par sociālo dzīvi no ziņām, bet, ja ir ESTP, tad ziņas par ceļojumiem. Ideja ir tāda - izmantot personības īpašības kā metriku un sasaitīt to ar konkrētu informāciju, ko var iegūt.

Zemāk ir redzami konkrēti personības tipi un atbilstošais informācijas paveids.

3.1. tabula

#### Personību tipi un to atslēgvārdi[9]

Personības tips	Atslēgvārdi
ISTJ	Organized, decisive, follows-through, gets things done, maintains existing framework.
ISFJ	Organized, preserves what works, builds consensus, people skills.
INFJ	Imagination, sense of purpose, creativity, organized, develops others.
INTJ	Strategic, analytical, problem solver, conceptual design, planning.
ISTP	Analytical, problem-solver, technical knowledge and expertise, adaptable, takes risks.

**Personību tipi un to atslēgvārdi[9]**

<b>Personības tips</b>	<b>Īpašības</b>
ISFP	Gentle persuader, craftsman, adaptable, people skills, builds consensus.
INFP	Sense of purpose, creative, people skills, adaptable, gentle persuader.
INTP	Analytical, problem solving at system level, technical knowledge and expertise, adaptable, conceptual design.
ESTP	Entrepreneurial, crisis management, selling and promoting, adaptable, takes risks.
ESFP	Adaptable, enthusiastic, high energy, motivate others, negotiation, consensus building, problem solving.
ENFP	Ideas and possibilities, enthusiasm, high energy, people skills, adaptable, creative.
ENTP	Creative, problem solving at systems level, analytical, technical, entrepreneurial.

**Personību tipi un to atslēgvārdi[9]**

<b>Personības tips</b>	<b>Īpašības</b>
ESTJ	Organize people and tasks, analytical, problem solving at functional level, developing and running processes and procedures, decisive.
ESFJ	Develop effective procedures, decisiveness, maintain and preserve what works, people skills, consensus builder.

Personības tips	Īpašības
ENFJ	Aktīvi sociāls, entuziasts, harmonizējošs, izteiksmīgs, saskarsmē silts, uz iespējām orjentēts, ideālists, empātisks, ar bagātu iztēli, sadarbojas, spēj novērtēt, taktisks.
ENTJ	Vadošs organizētājs, plānotājs, uz vīziju fokusēts, uzsācējs, konceptuāls, stratēģis, sistemātisks, kritisks, loģisks, organizētājs, spēj mierīgi aizstāvēt savu viedokli, ātri un efektīvi pieņem lēmumus, tiecas pēc sasniegumiem un uzlabojumiem.

### 3.2. News API

Lai ieviestu personīgo zināšanu pārvaldi, ir nepieciešami dati. News API nodrošina iespēju iegūt datus par ziņām, no dažādām valstīm. Viss, kas ir nepieciešams – pierēģistrēt savu kontu un iegūt API key.

Izmantojot šo API var iegūt ziņas no visiem avotiem, vai arī no specifiskiem uzņēmumiem dažādās valstīs. Kā arī var iegūt rakstus saistībā ar konkrētu pieprasījumu, piemēram visus rakstus, kur ir minēta Google kompānija, un ir iespējams šos rakstus kārtot konkrēta secībā, piemēram pēc popularitātes. Var arī kārtot no konkrēta līdz konkrētam datumam.

Ļoti parocīgi ir tas, ka tiek iegūti vērtīgi dati, process tikai aprobežots ar to, ka atnāk norāde uz mājas lapas konkrētā raksta adresi.[6]

Piemēram var veikt šādu GET pieprasījumu:

<https://newsapi.org/v2/everything?q=apple&from=2018-05-20&to=2018-05-20&sortBy=popularity&apiKey=af4be50a552f432b825a8f5d30388273>

```
{
  "status": "ok",
  "totalResults": 865,
  "articles": [
    {
      "source": {
        "id": "the-verge",
        "name": "The Verge"
      },
      "author": "Dieter Bohn",
      "title": "Hands-on with the RED Hydrogen One, a wildly ambitious smartphone",
      "description": "Come for the holograms, stay for the modules",
      "url": "https://www.theverge.com/2018/5/19/17372510/red-hydrogen-one-smartphone-hands-on-photos-video",
      "urlToImage": "https://cdn.vox-cdn.com/thumbor/kc9LQVjESsbA8u6M77yR64L8-cE=/0x292:2040x1360/fit-in/1200x630/cdn.vox-cdn.com/uploads/chorus_asset/file/10885253/8A0A7296.jpg",
      "publishedAt": "2018-05-20T01:38:02Z"
    },
    {
      "source": {
        "id": null,
        "name": "Gizmodo.com"
      },
      "author": "AJ Dellinger",
      "title": "App That Allows Parents to Monitor Teens Leaked Thousands of Passwords",
      "description": "TeenSafe, a service used by parents to monitor the online behaviors and phone activity of their children...",
      "url": "https://gizmodo.com/app-that-allows-parents-to-monitor-teens-leaked-thousan-1826180559",
      "urlToImage": "https://i.kinja-img.com/gawker-media/image/upload/s--QhgyIuwr--/c_fill,fl_progressive,g_center,h_900,q_80,w_1600/mgds9npzeujkv7hqato0.png",
      "publishedAt": "2018-05-20T21:02:00Z"
    },
    ....
  ]
}
```

### 3.1. att. Rakstu JSON masīvs, kur katrā ir objekts, kas astur informāciju par rakstu

Veicot iepriekšminēto GET pieprasījumu, atbildē tiek saņemts JSON objekts, kāds tas ir redzams 3.1 attēlā. Ļoti noderīgi ir tas, ka viens no datiem, kas tiek atgriezts, ir raksta apraksts, kuru var izmantot, lai iegūtu nepieciešamo informāciju, izmantojot NLP metodes.

Izpētot šo JSON, var izprast, ka var iegūt diezgan daudz noderīgu informāciju, pirms vispār apmeklēt raksta mājas lapas adresi.

## 3.3. Google Cloud Natural Language

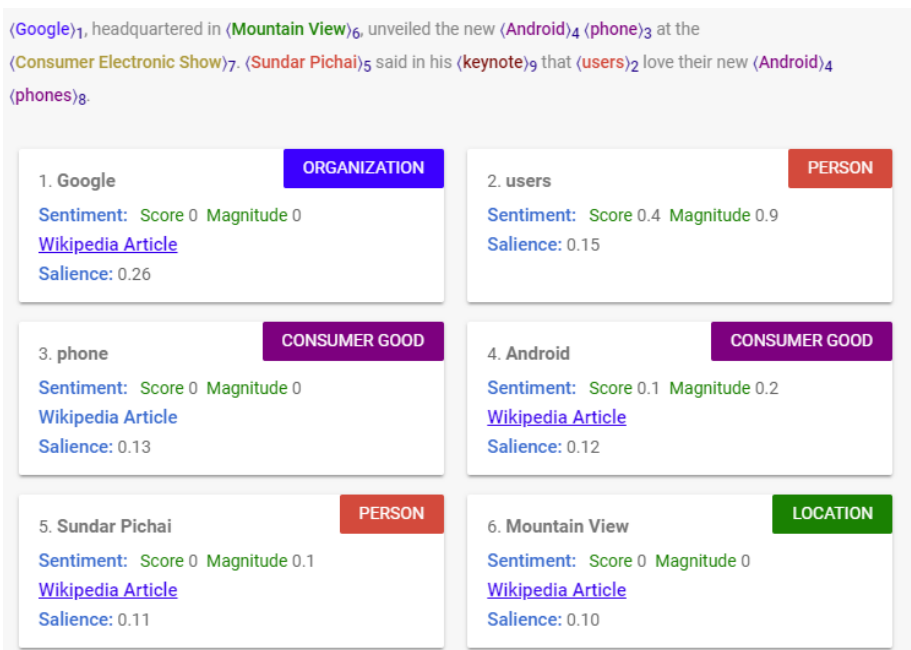
Google Cloud Natural Language ir sistēma, kas palīdz iegūt teksta nozīmi un struktūru. To var izmantot cauri REST API. Sūtot, piemēram, iepriekš izvēlētu, tekstu uz šo sistēmu un iegūstot nepieciešamo informāciju atbildē.

Google Cloud Natural Language API piedāvā šādas iespējas:

- Entītijū iegūšana no teksta;
- Noskaņojuma analīze;
- Sintakse;
- Kategorijas iegūšana.

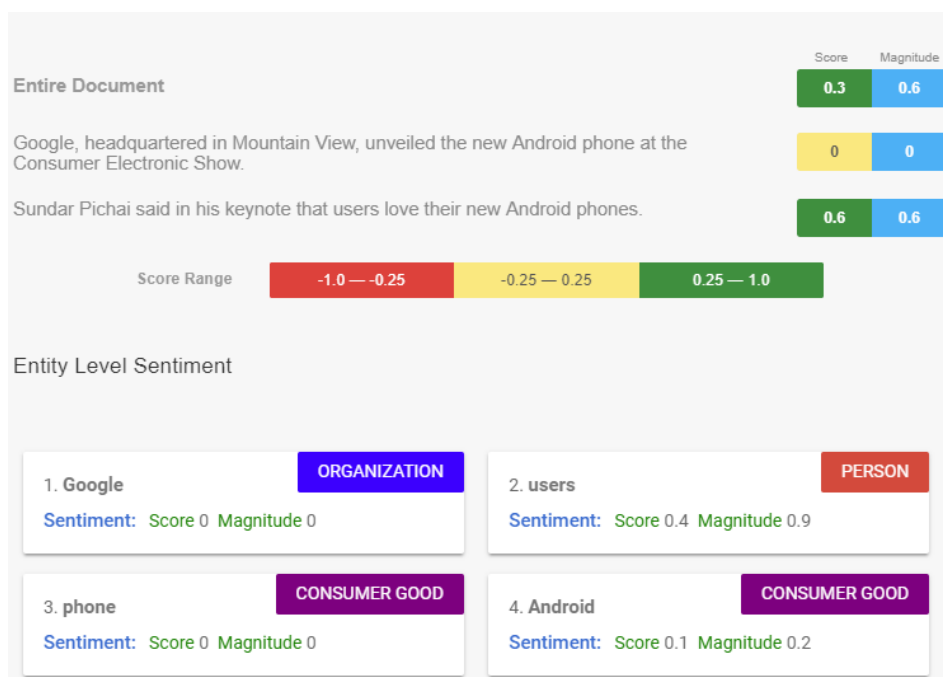
Piemēram, ja tiek padots šāds teksts – “Google, headquartered in Mountain View, unveiled the new Android phone at the Consumer Electronic Show. Sundar Pichai said in his keynote that users love their new Android phones.”, tad ir šādi rezultāti:

Entītijū analīzē pārbauda norādīto tekstu, meklējot pazīstamus vārdus, kas ir korekti lietvārdi kā sabiedrības darbinieku vai orjentiери un atgriež informāciju par šīm entītijām. Attēlā 3.2. var redzēt iegūtos datus.



### 3.2. att. Iegūtās entītijas no teksta

Noskaņojuma analīze pārbauda norādīto tekstu un identificē dominējošo emocionālo viedokli tekstā, it sevišķi, lai noteiktu rakstītāja (autora) attieksmi kā pozitīvu, negatīvu, vai neitrālu. Attēlā 3.3. var redzēt iegūtos datus.

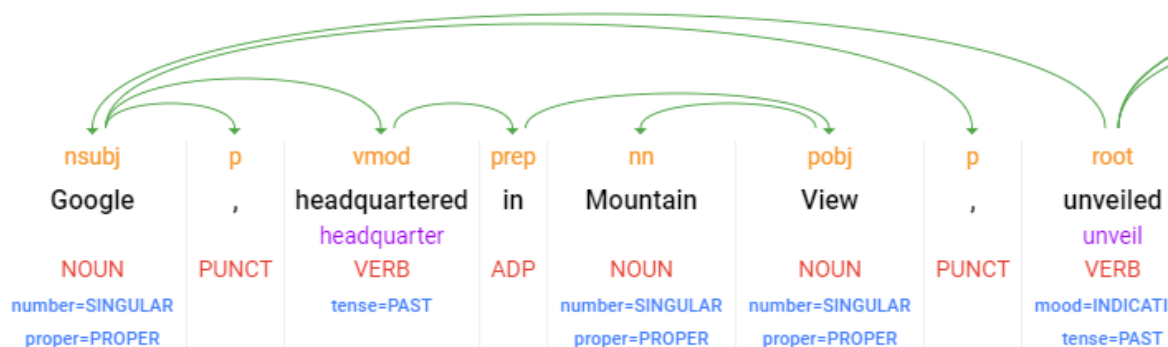


### 3.3. Iegūtais noskaņojums no teksta

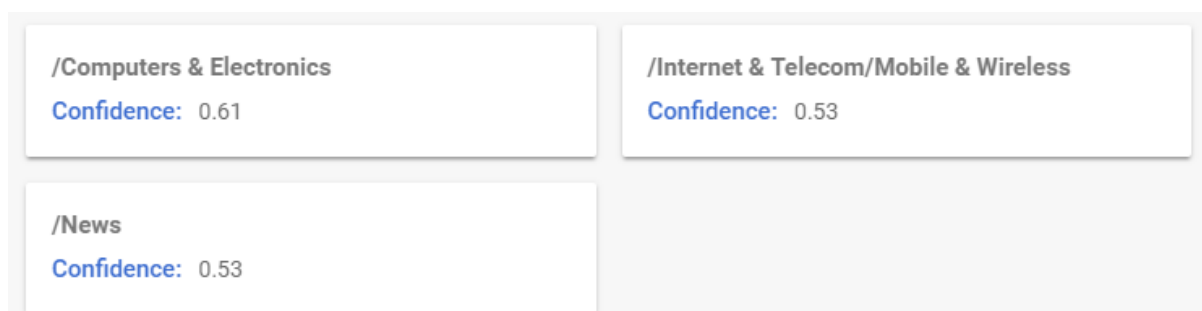
Sintaktiskā analīze iedala attiecīgo tekstu teikumu un vārdšķiru etiķešu kopās, un sniedz valodas informāciju par vārdšķiru etiķetēm. Attēlā 3.4. var redzēt iegūtos datus.

Satura klasifikācija analizē dokumentu un atgriež kategoriju sarakstu, kas attiecas uz dokumentā atrasto tekstu. Attēlā 3.5. var redzēt iegūtos datus.

Dependency 
  Parse Label 
  Part of Speech 
  Lemma 
  Morphology



3.4. att. Iegūtā sintakse no teksta



3.5. att. Iegūtās kategorijas no teksta

Personālās zināšanas pārvaldībai, visvairāk noder Entītiju analīze, kas no teksta atgriež entītijas, kuras var izmantot kā atslēgvārdus. Kā arī var izmantot sintaktisko analīzi, ja nepieciešams sasaistīt atslēgvārdus kopā. Un, ja iespējams ar satura klasifikāciju iegūt kategoriju, tad tā palīdzēs atslēgvārdam piešķirt kontekstu.

## 4. ANDROID LIETOTNES IZSTRĀDE

Nodaļā tiek aplūkoti vairāki soļi, kā tiek izstrādāta Android lietotne un kā notiek tās darbība. Lietotne ļaus cilvēkam aizpildīt personības testu. Apstrādājot testa rezultātus, lietotājam parādīs rezultātu – kāda viņam ir personība un īsu aprakstu. Kā arī tiks attēlots saraksts ar ziņām, ko lietotājs var aplūkot, atkarībā no atslēgvārdiem, kas ir saistīti ar šo personības tipu. Lietotājam spiežot uz rakstiem, rakstu sarakstā, apmeklējot tos, tiek sakrāta informācija no rakstu aprakstiem. Šī informācija tiek apstrādāta, kas savukārt rada jaunus atslēgvārdus, caur kuriem lietotājs var iet uz organizēto datu sadaļu un apskatīt jaunas ziņas saistībā ar jauniem iegūtiem atslēgvārdiem. Un turpinot apmeklēt arī jaunus rakstus, lietotne joprojām apstrādā rakstu aprakstus un ģenērē jaunus atslēgvārdus.

### 4.1. Izstrādes process

Lietotne tika izstrādāta uz Android platformas Java programmēšanas valodā. Tika veikti pieprasījumi uz News API sistēmu, izmantojot Retrofit bibliotēku, lai atvieglotu darbu ar http pieprasījumiem un iegūtu datus par dažāda veida ziņām. Kā arī tika izmantota Google Cloud Natural Language bibliotēka, kas apstrādāja no News API iegūtos datus, lai iegūtu atslēgvārdus, pēc kuriem var veidot jaunus pieprasījumus uz News API sistēmu, lai iegūtu jaunus datus, kuri atkal tiek apstrādāti ar NLP operācijām.

Visas tīkla operācijas darba autors veica paralēlos pavedienos, lai lietotāja saskarnes pavediens netiktu bloķēts.

Lai lietotnē realizētu personīgās zināšanas pārvaldību, tika izmantotas personības tipi un to raksturīgākie, vienlaikus raksturojošie, atslēgu vārdi, kā arī atslēgu vārdi, kas tika iegūti no ziņu avota, pēc dabiskās valodas apstrādes.

### 4.2. Lietotnes darbība

Šajā apakšnodaļā tiek aprakstīta lietotnes darbība un parādīti koda fragmenti.

Lietotnes startēšanas brīdī tiek inicializēti lietotāja saskarnes elementi un tiek ielādēti personības tipa testa jautājumi. Tā kā ir ļoti daudz personības testa jautājumu (86), tad tos ir parocīgāk glabāt atsevišķā failā. Lai varētu ērti apstrādāt datus, fails tiek glabāts JSON formātā. Lai iegūtu informāciju no šī faila, nepieciešams to atvērt un noparsēt datus. Attēlā 4.2. redzamā koda fragmentā tiek ielasīts teksts no faila, izmantojot ievades straumi. Un Attēlā 4.3. redzamā koda fragmentā tiek noparsēti dati. Attēlā 4.1. var redzēt personības testa lietotnes saskarni.

BITE △ 52% 19:21

## Personality Quiz

51. When I am at a meeting and a difference of opinion gets heated, I usually

defend the side I think is most logical.

try to create a harmonious atmosphere and solution.

52. When I get a new project, I usually prefer to

get to it and complete it as quickly as possible.

move as slowly as allowable and explore various approaches.

53. I am more interested in

the actual tasks I am doing day-by-day.

the thoughts I have about my work.

54. When I attend a training session, I usually prefer

NEXT

#### 4.1. att. Personības testa skats

```
public String readFromFile(String fileName, Context context) {
    ...
    try {
        fIn = context.getResources().getAssets()
            .open(fileName);
        isr = new InputStreamReader(fIn);
        input = new BufferedReader(isr);
        String line = "";
        while ((line = input.readLine()) != null) {
            ReturnString.append(line);
        }
    } catch (Exception e) {
        e.getMessage();
    } finally {
        ...
    }
    return ReturnString.toString();
}
```

#### 4.2. att. Datu ielasišana no faila

```

public List<Question> parseQuizJSON(String fileName) {
    String quizString = readFromFile(fileName, getBaseContext());
    Gson gson = new GsonBuilder().create();
    Quiz quiz = gson.fromJson(quizString, Quiz.class);
    return quiz.getQuestionList();
}

```

#### 4.3. att. JSON datu parsēšana, izmantojot Gson bibliotēku

Lietotājam ir jāatbild uz visiem jautājumiem, lai var noteikt personības tipu, un, lai lietotājs varetu doties uz nākošo skatu. Katrā jautājumā ir divas atbildes. Lietotājam obligāti jāatbild uz vienu no tām. Personības testa jautājuma un atbildes formāts redzams 4.4. attēlā.

```

"questions": [{
    "title": "1. My best ideas at work come from",
    "selections": [
        "an interchange of ideas and sharing with others.",
        "quiet thought on my own."
    ]
}

```

#### 4.4. att. Viens no jautājumiem personības noteikšanai[7]

Kad lietotājs atbild uz visiem jautājumiem un nospiež pogu - turpināt. Tad tiek veikti aprēķini. Lai varētu ērtāk aprēķināt rezultātus, personības testa jautājumi tika sadalīti konkrētā secībā. Ik pa 4 jautājumiem secība atkārtojas. Šādā secībā:

- Ekstraverts un introverts;
- Sensorisks un intuitīvs;
- Domājošs un jūtīgs;
- Izprotošs un noteicošs.

Atbildot uz jautājumu, tiek pielikts viens punkts vienā no pāra elementiem, piemēram, ekstravertam vai introvertam. Beigās iegūtie rezultāti (punktos) tiek saskaitīti. Un no katra pāra paņem vienu no personības iezīmes iniciāļam atkarībā, kuram ir vairāk punktu. Piemēram, ja rezultātā vissvairāk punktu ir: ekstraverts, sensorisks, domājošs un izprotošs, tad rezultātā ir ESTJ personības tips. Attēlā 4.5. redzams koda fragments, kur tiek saskaitīti punkti un noskaidrots personības tips.

```

private String personalityType() {
    Score score = new Score();

    for (int i = 0; i < adapter.getQuestionList().size(); i++) {
        int type = (i + 1) % 4;

        int answer = adapter.getQuestionList().get(i).answer;

        switch (type) {
            case 1: // E/I
                if (answer == 0) {
                    score.extravert++;
                } else {
                    score.introvert++;
                }
                break;
            case 2: // S/N
                ...
                break;
            case 3: // T/F
                ...
            case 0: // J/P
                ...
        }

        String type = "";

        type = type + String.valueOf(score.extravert > score.introvert ? "E" : "I");
        type = type + String.valueOf(score.sensor > score.intuitive ? "S" : "N");
        type = type + String.valueOf(score.thinker > score.feeler ? "T" : "F");
        type = type + String.valueOf(score.perceiver > score.judger ? "J" : "P");

        return type;
    }
}

```

#### 4.5. att. Personības testa punktu saskaitīšana un personības noteikšana

Nospiežot pogu - tālāk - lietotājs redz savus iegūtos rezultātus. Tiek parādīts personības tipa nosaukums un īss apraksts. Kā arī ir redzamas divas daļas – ziņas un organizētie dati. To var redzēt 4.6. attēlā koda fragmentā.

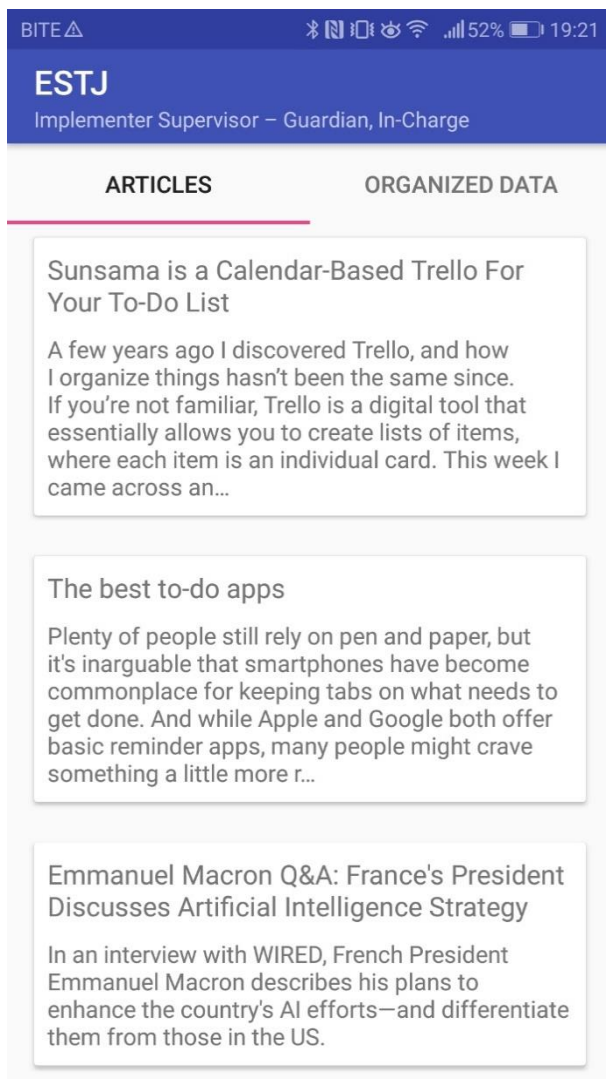
```

"questions": [{
  "title": "1. My best ideas at work come from",
  "selections": [
    "an interchange of ideas and sharing with others.",
    "quiet thought on my own."
  ]
}]

```

#### 4.6. att. Personības testa jautājuma json struktūra

Nospiežot pogu – tālāk – lietotājs redz savu iegūto rezultātus. Tiek parādīts personības tipa nosaukums un īss apraksts. Kā arī ir redzamas divas daļas – ziņas un organizētie dati. To var redzēt 4.7. attēla redzamā saskarnē.



#### 4.7. att. **Ziņu saraksta saskarne**

Kad lietotājs ieiet šajā saskarnē, tad tiek veikts pieprasījums uz News API, lai iegūtu nepieciešamos datus par ziņām. Pirms veikt pieprasījumus, tiek inicializēta bibliotēku, kas palīdzēs to darīt. Šajā gadījumā tā ir Retrofit bibliotēka. Tiek inicializēts Retrofit objekts, kuram tiek padoti HTTP klienta dati, lai var veikt HTTP pieprasījumus, kā arī news REST API url, lai iegūtu datus tieši no news REST API. Līdz ar Retrofit objekta inicializāciju arī tiek padots http inceptors ar kura palīdzību var sekot līdzi izsūtītiem un saņemtiem datiem. Attēlā 4.8. redzams koda fragments.

```

public enum PersonalityType {
    ISTJ("Planner Inspector - Guardian, Chart-the-Course ",
        Arrays.asList("Organized", "decisive", "follows-through", "gets things done", "maintains existing framework")),
    ISTP("Analyzer Operator - Artisan, Chart-the-Course",
        Arrays.asList("Analytical", "problem-solver", "technical knowledge and expertise", "adaptable", " takes risks")),
    ISFJ("Protector Supporter - Guardian, Behind-the-Scenes",
        Arrays.asList("Organized", "preserves what works", "builds consensus", "people skills")),
    ....

    public final String description;
    public final List<String> keywords;

    PersonalityType(String description, List<String> keywords) {
        this.description = description;
        this.keywords = keywords;
    }
}

```

#### 4.8. att. Retrofit objekta inicializācijas metode

Tā kā news API ir nepieciešams pieprasījuma simbolu virkne, tad tiek sūtīts personības tipa atslēgvārdi kā pieprasījums. Katrs personības tips satur vairākus atslēgvārdus. Uz katru atslēgvārdu atsevišķi tiek veikts pieprasījums un saņemti atpakaļ dati. Attēla 4.9. koda fragmentā var redzēt personības tipu atslēgvārdus.

```

private void initRetrofit() {
    OkHttpClient okHttpClient = new OkHttpClient.Builder()
        .addNetworkInterceptor(new StethoInterceptor())
        .build();

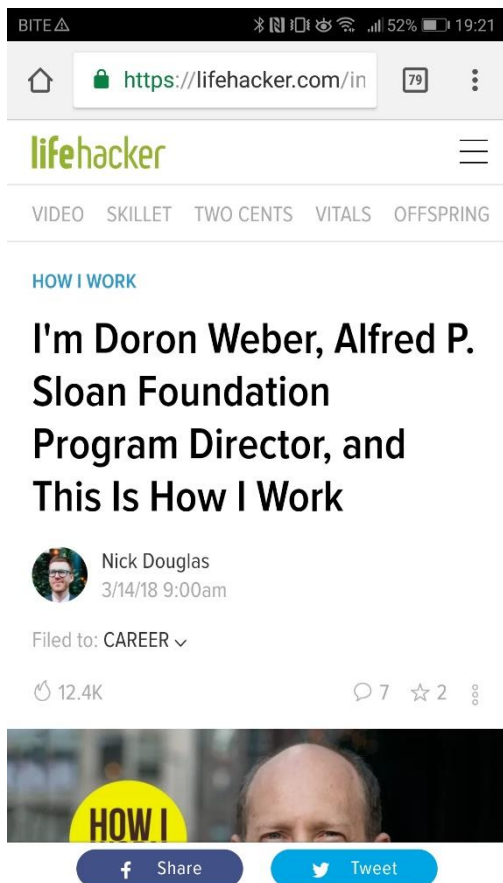
    Retrofit retrofit = new Retrofit.Builder()
        .client(okHttpClient)
        .baseUrl(BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
        .build();

    apiService =
        retrofit.create(NewsApi.class);
}

```

#### 4.9. att. Personības apraksts un atslēgvārdi

Kad lietotājs uzspiež uz kādu no ziņām, tad tiek atvērts ziņu portāls telefona tīmekļa pārlūkprogrammā, kas ir redzams 4.10 attēlā. Un attēlā 5.11 redzams koda fragments.



#### 4.10. att. Ziņu portāla paraugs

```
public void openArticleWebsite(String url) {  
    try {  
        Uri webpage = Uri.parse(url);  
        Intent myIntent = new Intent(Intent.ACTION_VIEW, webpage);  
        startActivity(myIntent);  
    } catch (ActivityNotFoundException e) {  
        e.printStackTrace();  
    }  
}
```

#### 4.11. att. Ziņu portāla atvēršanas metode

Tajā brīdī, kad tiek atvērts ziņu portāls, tiek paņemts ziņu portāla apraksts un ar Google Cloud Natural language bibliotēkas palīdzību tiek apstrādāts, izmantojot NLP operācijas. Tas tiek veikts uz paralēliem pavedieniem, lai nebloķēt lietotāja saskarni, jo teksts tiek padots dokumenta objektam un, tad nosūtīts uz Google Cloud serveri, kur dabiskās valodās sistēma apstrādāt vārdus un atgriež nosauktās entītijas, piemēram organizāciju vai personu vārdus. 4.12. Attēla redzams koda fragments.

```

private void analyzeText(String text, String url) {
    Document document = new Document();
    document.setType("PLAIN_TEXT");
    document.setLanguage("en-US");
    document.setContent(text);

    ...

    AsyncTask.execute(() -> {
        AnnotateTextResponse response;
        try {
            response = naturalLanguageService.documents()
                .annotateText(request).execute();

            final List<Entity> entityList = response.getEntities(),
                saveKeyWords(entityList, url);

        } catch (IOException e) {
            e.printStackTrace();
        }
    });
}

```

#### 4.12. att. Metode, kas apstrādā vārdus ar NLP operācijām

Kad no Googles servera atnāk nosauktās entītijas, tad tās tiek saglabātas uz ierīces, lai var izmantot kā atslēgvārdus. Entītijas tiek saglabātas izmantojot Android failu sistēmu shared preferences, kas atvieglo darbu ar failu sistēmu. NLP entītijām ir vairākas kategorijas:

- Person;
- Location;
- Organization
- Event
- Work of art
- Consumer good
- Unknown
- Other

Katra kategorija tiek saglabāta atsevišķā failā, un piekļuves atslēga failam ir nosaukt pēc entītijas kategorijas nosaukuma, t.i. viens fails personām, cits fails organizācijām, utt... Tā kā failu sistēmā var saglabāt tikai tekstu, tad ir nepieciešams vārdus saglabātā tādā formātā, lai vēlāk nolasot failu var iegūt saprotamu informāciju. Lai to izdarītu entītijas objektus pārveido par json tipa objektu, kuru pārvērš par simbolu virkni un saglabā failā. Attēlā 4.13. redzams koda fragments, kas atbild par entītijas saglabāšanu. Pirms saglabājam datus failu sistēmā, nepieciešams pārliecināties, ka nav duplikātu, jo var gadīties, ka ziņu aprakstos var gadīties entītijas ar tādu pašu nosaukumu, vai arī, ja apskata vienu un to pašu rakstu vairākas reizes, arī

tad var notikt nevajadzīga atkārtšanās. Duplikātus izņemt no saraksta, pārveidojot json objektu par entītiju sarakstus, kuru pārvērš par ImmutableSet objektu. Tā kā ImmutableSet objektā nedrīkst atrasties kopijas, tad automātiski pārveršot sarakstu par šādu objektu, tiek izņemti duplikāti. Tālāk šo objektu pārvērš atpakaļ par json objektu, kuru var jau saglabāt failu sistēmā. Attēlā 4.14 redzams koda fragments, kur tiek izņemti duplikāti. Bet ar to nepietiek. Duplikāti ir objekti. Lai varētu noteikt, kuri no objektiem ir duplikāti, ir nepieciešams Entītiju objektam pievieno speciālas metodes – hashCode() un equals(), ar kuru palīdzību var salīdzināt divus entītiju objektus. 4.15. Attēlā redzamā koda fragmentā, var redzēt šīs metodes.

```
private void saveEntity(Entity entity, Gson gson, SharedPreferences sharedPref, String url) {
    String jsonSaved;
    jsonSaved = sharedPref.getString(entity.getType(), "");

    EntityWrapper entityWrapper = new EntityWrapper(entity.getName(), url, entity.getType());
    String jsonNewEntityToAdd = gson.toJson(entityWrapper);

    JSONArray jsonArrayEntity = new JSONArray();

    try {
        if (jsonSaved.length() != 0) {
            jsonArrayEntity = new JSONArray(jsonSaved);
        }
        jsonArrayEntity.put(new JSONObject(jsonNewEntityToAdd));
    } catch (JSONException e) {
        e.printStackTrace();
    }

    JSONArray jsonArrayEntityNoCopies = removeDuplicates(jsonArrayEntity, gson);

    SharedPreferences.Editor editor = sharedPref.edit();
    editor.putString(entity.getType(), String.valueOf(jsonArrayEntityNoCopies));
    editor.commit();
}
```

#### 4.13. att. Entītiju saglabāšanas metode

```
public JSONArray removeDuplicates(JSONArray jsonArrayEntity, Gson gson) {
    Type type = new TypeToken<List<EntityWrapper>>() {
    }.getType();
    List<EntityWrapper> entityList = gson.fromJson(String.valueOf(jsonArrayEntity), type);

    JsonElement jsonElement =
    gson.toJsonTree(ImmutableSet.copyOf(entityList).asList(), new TypeToken<List<EntityWrapper>>() {
    }.getType());

    return jsonElement.getAsJsonArray();
}
```

#### 4.14. att. Duplikātu izņemšanas metode

```

public class EntityWrapper {
    ...
    public EntityWrapper(String name, String url, String type) {...}

    @Override
    public String toString() {
        return name;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((name == null) ? 0 : name.hashCode());
        return result;
    }

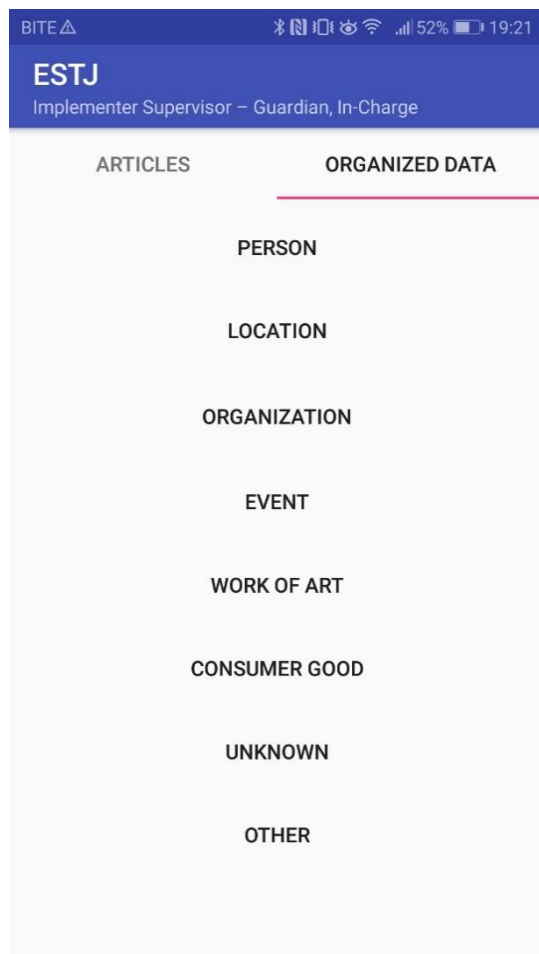
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        EntityWrapper other = (EntityWrapper) obj;
        if (name == null) {
            if (other.name != null)
                return false;
        } else if (!name.equals(other.name))
            return false;
        return true;
    }
}

```

#### 4.15. att. Entīcijas objekta klase

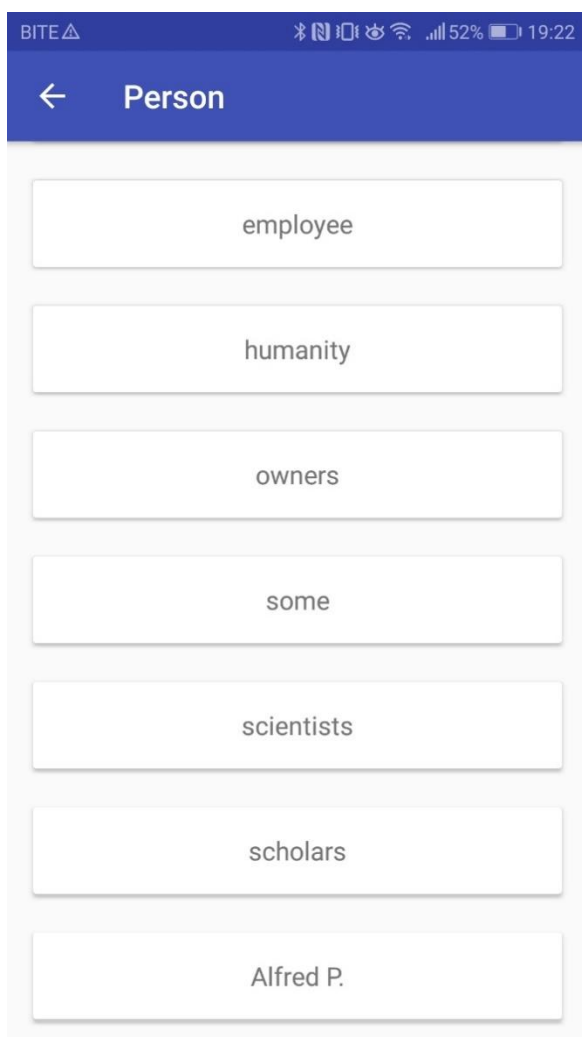
Ejot vairākos ziņu portālos, visu laiku tiek iegūtas jaunas entīcijas no ziņu portāla raksta aprakstiem. Tā kā tie tiek saglabāti uz ierīces. Tos var izmantot organizēto datu sadaļā, kā atslēgvārdus, lai iegūt informāciju par jauniem rakstiem. Lai apskatītu organizētos datus,

lietotājam nepieciešams doties uz organizēto datu saskarnes skatu. Šo skatu var redzēt 4.16. attēlā. Šajā skatā var redzēt visas iespējamās entītiņu kategorijas.



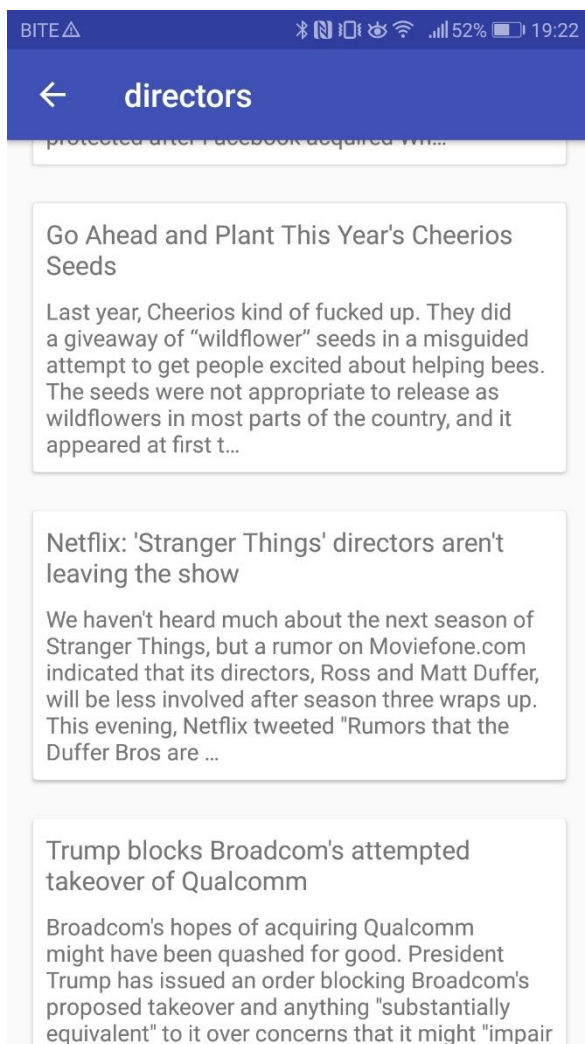
#### 4.16. att. **Organizēto datu lietotnes saskarne**

Atverot kādu no kategorijām, tiek atvērts jauns skats un ielādēts fails ar konkrēto kategoriju, kas sevī ietver atslēgvārdus, kas tika iegūti no entītiņām. Šo skatu var redzēt 4.17. attēlā.



4.17. att. Skats, kur attēloti atslēgvārdi personas kategorijai

Kad lietotājs uzspiež uz kādu no atslēgvārdiem, tad atverās saraksts ar rakstiem, līdzīgā veidā kā saskarnē, kur tiek iegūti raksti, sūtot personību tipu atslēgvārdus, tikai šajā gadījumā tiek sūtīti atslēgvārdi, kas ir iegūti no entītijām, kas ir iegūti apstrādājot rakstu aprakstus, izmantojot NLP operācijas. Šo skatu var redzēt 4.18. attēlā.



#### 4.18. att. **Directors** atslēgvārda iegūtie raksti

Šajā skatā ir tā pati funkcionalitāte, kas ir skatā, kur ir redzami raksti, kas tika iegūti no personības tipa atslēgvārdiem. Tas nozīmē, ka atkal tiek veikts pieprasījums uz news API un Google Cloud, lai iegūtu jaunās ziņas un entītijas, kā arī tiek izdzēsti duplikāti. Tādējādi tagad ejot jaunos rakstos, tiek iegūti jaunie atslēgvārdi, kuru dēļ tiek iegūti jauni raksti.

### **4.3. Problēmas un risinājumi izstrādājot lietotni**

Pildot testu, ja lietotājs koncentrēti un apzinīgi nepiervērš uzmanību jautājumiem, pastāv iespēja, ka var iegūt kļūdainu rezultātu - nepareizo personības tipu. Paredzams, ka lietotājam var kļūt grūti, garlaicīgi, vienkārši radīsies nevēlēšanās to darīt pietiekoši precīzi un kārtīgi, sevišķi uz beigām, jo ir tests liels – satur 86 jautājumus. Jāņem vērā, ka lietotājs var būt ierobežots laikā, steigties visu izdarīt pārāk ātri. Tādējādi arī tiks iegūts neprecīzs rezultāts.

Izmanojot iegūtos jautājumus, tika izveidots personības tests. Šis tests tika vairākas reizes iziets, lai iegūtu dažādus personība tipus un redzētu dažādus rakstus jau pašā darba sākumā. No iegūtajiem atslēgvārdiem, liekot tos lietā, tika iegūti raksti no News API. Pašā sākuma raksti bija diezgan atšķirīgi, atkarība no personības tipa, izņemot gadījumus, ja abiem bija dažas līdzīgas īpašības. Dažādām personībām tika parādīti tie paši raksti, jo atslēgvārdi bija vienādi, piemēram ISTP un INTJ abiem ir atslēgvārds analytical, un tādēļ, ka netika atslēgvārdam pievienots vēl kāds vārds, kas aprakstītu ISTP un INTJ personības tipus, tika iegūti līdzīgi raksti, kas principā bija paredzams rezultāts. Proti, tas, kas šāda situācija noteikti var izveidoties arī atkārtoti, jāņem vērā uzreiz.

Katram atslēgvārdam tika pieprasīti 20 populārākie raksti no visiem iespējamiem avotiem kopumā. Tādēļ, ka tika izmantoti dažādi un daudzveidīgi avoti, dažreiz tika iegūti raksti, kas nebija tieši un precīzi atbilstoši konkrētam personību tipam. Arī tas ir jāņem vērā, jo strādājot ar lielu informācijas apjomu, daļa no tā būs “informācijas troksnis”, kas nebūs vajadzīgais materiāls, bet vienkārši kaut kas, kas atbilst tikai šķietami. Proti – kļūdaini pozitīvs rezultāts.

Izejot cauri pietiekoši daudziem rakstiem ar dažādiem personību tipiem, gala rezultātā bija ļoti līdzīgi raksti, jo tika iegūti līdzīgi atslēgvārdi, šāda situācij izveidojās tādēļ, ka tiem atslēgvārdiem netika piešķirts konteksts saistība ar personības tipu.

Apstrādājot datus ar NLP operācijām tika iegūtas entītijas. Šīs entītijas ne vienmēr bija precīzas. Piemēram tika iegūtas tādas organizācijas (vārdi) kā Microsoft, Google, Samsung, kas ir pareizi, bet tajā pašā laikā tika iegūti arī tādi vārdi kā: party, colleges, Samsung Galaxy S9, kas nav organizācijas. Un, saprotams, ka īsti neatbilst prasītajam.

### **Problēmas un risinājumi izstrādājot lietotni**

Izstrādājot lietotni, darba autoram nācās saskarties ar dažādām problēmām. 5.1. tabulā attēlotas problēmas, kas parādījās izstrādes procesā un kādi bija to risinājumi.

*4.1. tabula*

#### **Problēmas, ar kurām saskārās, veidojot lietotni un to risinājumi**

<b>Problēma</b>	<b>Risinājums</b>
Nepieciešams saglabāt entītijas atmiņā, bet entītijas objektus tieši (pa taisno) nevar saglabāt – tikai primitīvus datus.	Pārveidot entītiju objektu par json objektu, kuru pārveidot par simbolu virkni, kuru var ierakstīt failā.

**Problēmas, ar kurām saskārās, veidojot lietotni un to risinājumi**

<b>Problēma</b>	<b>Risinājums</b>
Google Cloud Natural Language bibliotēkas piedāvāto objektu nevar serializēt un pārveidot par json objektu, jo GSON bibliotēka uztver float par double tipu, tāpēc, ka bibliotēkas objektam nav aiz cipara rakstīts f burtiņš.	Vai arī rakstīt savu parseri, kas var noparsēt šādu objektu, vai arī no bibliotēkas piedāvāta objekta izvilkt nepieciešamos datus, izņemot to, kur ir float tips, ielikta citā objektā un tad pārveidot par json objektu.

**Problēmas, ar kurām saskārās, veidojot lietotni un to risinājumi**

<b>Problēma</b>	<b>Risinājums</b>
Personības testa jautājumu pievienošana kodā izskatās neppāredzami, jo ir ļoti daudz jautājumi.	Jautājumus saglabāt atsevišķā failā JSON formātā.
Kad dati tiek pieprasīti no News API vai Google Cloud Natural Language, tad iestājās kļūda.	Android lietotne avarē, ja tiek veiktas tīkla operācijas uz UI pavediena, tāpēc, lai izvairītos no šīs problēmas, tiek veikti datu pieprasījumi uz cita pavediena un rezultāti tiek rādīti uz UI pavediena.
Kad personības testa jautājumi tiek aizpildīti un jautājumu saraksts ritināts uz leju, pēkšņi pārējie jautājumi jau arī izrādās atbildēti, jo radio button jau ir iezīmēts.	Android saraksta UI elements ir optimizēti apstrādāt, lai varētu ik pa brīdim atkārtoti izmantot iepriekšējo elementu, un, ja iepriekšējā elementā radio button tika jau iezīmēts, tad jaunajā, tas arī notiks. Tāpēc katru reizi, kad tiek attēlots jauns UI elements, jāpārjautā, vai tas jautājums tika jau atbildēts un kāda bija atbilde.

Kā redzams, lielākā daļa problēmu ir saistīta ar datu ielasīšanu un saglabāšanu saistībā ar to, ka dati jāglabā atmiņā, lai pie viņiem varētu piekļūt konkrētā vietā un viņi nepazustu pēc iziešanas no lietotnes. Tāpēc vajag to pareizi serializēt un saglabāt json formātā, lai var ērti piekļūt un saglabāt. Tā kā ir arī dati, kas tiek pieprasīti no interneta, prasot resursu patēriņus, tādēļ tiek bloķēta saskarne, no tā būtu jāizvairās – jāatrisina tas, kā to izdarīt. Kā arī jāpievērš uzmanība Android UI elementiem, lai nav vizuālas kļūmes, kas var rasties elementu optimizācijas dēļ.

### **Personīgās zināšanas pārvaldības problēmas**

- Neprecīzi rezultāti pildot personības testu.

Risinājums: Pārveidot testu, lai tas ir īsāks, vai arī piedāvāt vairākus testus - gan garāku, gan īsāku, bet brīdinot, ka precīzāki rezultāti var būt garākajā testā, ar nosacījumu, ka lietotājs to pacietīgi un kārtīgi, pievēršos uzmanību precizitātei līdz pat beigām, tomēr izpildīs.

- Dažiem personību tipiem līdzīgi raksti pēc testa izpildīšana.

Risinājums: Papildināt personības tipa atslēgvārdu ar vēl kādu vārdu, kas atbilstu konkrētam persobīnas tipam, lai atšķiras no cita personības tipa tā paša atslēgvārda. Šajā gadījumā ir īpaši jāpārdomā tas, kādu pildus atslēgvārdu izvēlēties, ņemot vērā visu testa sistēmu, lai domājot par uzlabošanu, nejauši netiktu ieviesta papildus kļūda.

- Daudziem personību tipiem līdzīgi raksti organizētos datos, ja ir iziets cauri pietiekoši lielam rakstu apjomam.

Risinājums: Iegūtos jaunus atslēgvārdus sasaistīt ar personības tipu kā arī papildināt ar vēl kādiem atslēgvārdiem no iegūta raksta, piemēram, ja tas vārds bija tajā pašā teikumā vai kontekstā, jo var būt viens un tas pats atslēgas vārds no dažādiem rakstiem apzīmējot dažādu kontekstu.

- Neatnāk atbilstoši raksti konkrētam atslēgvārdam atkarība no personības tipa.

Risinājums: Sašaurināt raksta avotus atkarībā no tā kāds personības tips, jo konkrētā brīdī tiek pieprasīti populārākie raksti no pilnīgi visiem avotiem, kas pieejami news API.

- Iegūtās entītijas pēc NLP operācijām ne vienmēr precīzas.

Risinājums: Ņemt vērā precizitātes vērtību, ko atgriež Google Cloud Natural Language sistēma, un noteikt minimālo vērtību, lai ignorētu neprecīzās entītijas.

## REZULTĀTI

Darbā tika veikts pētījums ar mērķi noskaidrot, kā izstrādāt Android lietotni, kas palīdzētu ar personīgo zināšanu pārvaldību.

Izstrādājot maģistra dabu tika apkopoti vairāki resursi par personīgo zināšanu pārvaldību. Tika izpētīts, kas īsti ir personīgo zināšanu pārvaldība.

Lai izprastu, ko var darīt ar PKM un kā to realizēt ar Android palīdzību, tika izpētīti daži esošie risinājumi un veidi, kā to var realizēt.

Tika izprasts, kā lietot News REST API, lai iegūt rakstus un nepieciešamo datus no tiem.

Meklējot veidu kā varētu apstrādāt iegūtos rakstu datus, tika atrast Google Cloud Natural Language API un izpētīts, kā lietot.

Kā arī tika pārlasīts maģistra autora bakalaura darbs, lai izmantot personību tipus Android lietotnes izstrādē. Un tika sameklēti atslēgvārdi, kas saistīti ar katru personības tipu. Kā arī veiksmīgi tika atrasti personība testa jautājumi un veids, kā var rezultātā iegūt konkrētu personības tipu.

Tika izstrādāta Android lietotne Java programmēšanas valodā. Android lietotnē tika implementēta MBTI personības tests. MBTI testa rezultātā tika iegūts personības tips un atslēgvārdi, kurus izmantoja News API, lai iegūt rakstu, no kuriem varētu iegūt raksta aprakstu. Raksta apraksti tika apstrādāti ar Google Cloud Natural Language bibliotēku, lai iegūtu jaunus atslēgu vārdus. Un ar jauniem atslēgvārdiem sadalītiem pa kategorijām, varēja apskatīt jaunus rakstus. Tika izteikti secinājumi par to, cik daudzveidīgus rakstus var iegūt dažādi personības tipi. Izstrādājot lietotni, nācas saskarties ar grūtībām, un tika atrasti un aprakstīti risinājumi šīm problēmām.

## SECINĀJUMI

PKM ir svarīgi ievērot visus posmus – uzkrāšanu, organizēšanu un dalīšanos. Ja kaut kas tiks izlaists, tad netiks iegūti pilnīgi rezultāti.

PKM ir daudz dažādi risinājumi, šajos risinājumus datus var gan pats pievienot, gan arī uzkrājot veicot kādas darbības. Labākie risinājumi ir tie, kas parūpējas, lai informācija ir viegli vizuāli uztverama, piemēram ar grafiem, tādejādi var ātri atrast nepieciešamo informāciju.

Personības tests lietotnē bija pārāk liels, nepieciešams piedāvāt vai nu īsāku testu, vai vairākus, kur var būt arī īsāki, jo lietotājs uz beigām vairs neseko un ir paviršāks.

Pēc Bakalaura darba pienesums maģistra ir tāds, ka ir vieglāk izprast dabisko valodas apstrādi un MBTI personības tipus.

Daudziem personību tipiem ir līdzīgi atslēgvārdi, tapēc tiek redzēti līdzīgi dati. Nepietiek tikai ar pliku atslēgvārdu, vajag viņu sasaistīt vel ar kaut ko, kas ir aktuāls personības tipam, lai ir atšķirīgāki dati. Kā arī ilgāk ejot cauri ziņām dažādi personības tipi iegūst līdzīgus atslēgvārdus, kas paši pa sevi ir vienādi un atkal tiek iegūta līdzīg informācija. Tapēc vajag šos atslēgvārdus sasaistīt ar kontekstu no kurām ziņām tik dabūti ārā. NLP analīzē nevienmēr ir precīza, tapēc var dažreiz parādīties nepareizi atslēgvārdi.

Izstrādājot lietotni, bija problēmas ar JSON parsēšanu, jo esoši objekti no bibliotēkas bija ar problēmatiskiem datiem, un bija problēma starp float un double tipu, tapēc nākotnē vēlams vienmēr sagatavot savu JSON parseri.

Pateicoties Android lietotnes izstrādei, ir skaidrs tas, ka var izstrādāt lietotni, bet smagām operācijām vajag ārējas sistēmas, kurās tiks veikti sarežģīti aprēķini, kā piemēram NLP operācijas. Un, lai ir precīzi rezultāti, vajag daudz datus, tapēc vēlams tos glabāt uz servera. Jo, lai būtu labāka precizitāte, vēlams daudz atslēgvārdus, bet tad vairs nebūs vietas.

Kad bija daudz atslēgvārdi, tad bija grūti sākt orientēties. Tapēc vēlams nākotnē realizēt prāta karti, kas palīdzētu vizuāli redzēt elementus.

## IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] Attēla pārākuma efekts. [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
[https://en.wikipedia.org/wiki/Picture\\_superiority\\_effect?lipi=urn%3Ali%3Apage%3Ad\\_flags%3Ahip3\\_pulse\\_read%3Bh40CboVpTNC1c5GBGVcd%2BQ%3D%3D](https://en.wikipedia.org/wiki/Picture_superiority_effect?lipi=urn%3Ali%3Apage%3Ad_flags%3Ahip3_pulse_read%3Bh40CboVpTNC1c5GBGVcd%2BQ%3D%3D)
- [2] Personīgo zināšanu pārvaldība. [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
[https://en.wikipedia.org/wiki/Personal\\_knowledge\\_management](https://en.wikipedia.org/wiki/Personal_knowledge_management)
- [3] Personīgo zināšanu pārvaldības komponentes. [tiešsaite] – [atsauce: 21.05.2018].  
Pieejams: <http://tom.spiglanin.com/2015/01/i-believe-in-the-importance-of-personal-knowledge-management/>
- [4] Personīgo zināšanu pārvaldības rīki. [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
<https://legacy.ici-berlin.org/library/research-toolkit/personal-knowledge-management-tools/>
- [5] Efektīva personīgo zināšanu pārvaldības sistēmas veidošana. [tiešsaite] – [atsauce: 22.02.2018]. Pieejams: <https://medium.com/@axtonliu/how-to-build-an-efficient-personal-knowledge-management-system-355332ae5991>
- [6] Newst REST API. [tiešsaite] – [atsauce: 21.05.2018]. Pieejams: <https://newsapi.org/>
- [7] Personības testa jauājumi [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
<https://github.com/jamesflorentino/mbti-test/blob/master/list.json>
- [8] . Fjodors Pohodņevs, *Android lietotne personības profila noteikšanai*
- [9] . Personību tipu iezīmes. [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
<http://execglobalnet.com/wp-content/uploads/2014/04/Reference-1.1-Personality-Type-Characteristics.pdf>
- [10] Ulrich Schmitt, *Knowcations – The Quest for a Personal Knowledge Management Solution*
- [11] Kyosuke Nishida, Hiroyuki Toda, Yoshimasa Koike, *Extracting Arbitrary-shaped Stay Regions from Geospatial Trajectories with Outliers and Missing Points*. [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
[https://www.researchgate.net/publication/291342986\\_Extracting\\_Arbitrary-shaped\\_Stay\\_Regions\\_from\\_Geospatial\\_Trajectories\\_with\\_Outliers\\_and\\_Missing\\_Points](https://www.researchgate.net/publication/291342986_Extracting_Arbitrary-shaped_Stay_Regions_from_Geospatial_Trajectories_with_Outliers_and_Missing_Points)
- [12] Steffen Remus, Manuel Kaufmann, Kathrin Ballweg, Tatiana von Landesberger, Chris Biemann, *Storyfinder: Personalized Knowledge Base Construction and Management by Browsing the Web*
- [13] Readability.js bibliotēka [tiešsaite] – [atsauce: 21.05.2018]. Pieejams:  
<https://github.com/mozilla/readability>

- [14] OpenNLP dabiskās valodas apstrādes bibliotēka. [tiešsaiste] – [atsauce: 21.05.2086].  
Pieejams: <https://opennlp.apache.org/>
- [15] OpenNLP dokumentācija. [tiešsaiste] – [atsauce: 21.05.2018]. Pieejams:  
<https://opennlp.apache.org/documentation/1.6.0/manual/opennlp.html>
- [16] David Montoya, Serge Abiteboul, Thomas Pellissier Tanon, Fabian M. Suchanek,  
*Thymeflow, A Personal Knowledge Base with Spatio-temporal Data* [tiešsaiste] – [atsauce:  
21.05.2018]. Pieejams: <http://www.lsv.fr/Publis/PAPERS/PDF/MPAS-cikm16.pdf>

# PIELIKUMI

## 1. pielikums

### JSON dati, kas tiek saņemti no News REST API, lai attēlotu rakstu sarakstu

```
▼ {status: "ok", totalResults: 922,...}
▼ articles: [{source: {id: null, name: "Lifehacker.com"}, author: "Emily Price",...},...]
▼ 0: {source: {id: null, name: "Lifehacker.com"}, author: "Emily Price",...}
  author: "Emily Price"
  description: "A few years ago I discovered Trello, and how I organize things hasn't been the same since. ..."
  publishedAt: "2018-04-08T19:40:00Z"
  ▼ source: {id: null, name: "Lifehacker.com"}
    id: null
    name: "Lifehacker.com"
    title: "Sunsama is a Calendar-Based Trello For Your To-Do List"
    url: "https://lifehacker.com/saunsama-is-a-calendar-based-trello-for-your-to-do-list-1825092309"
    urlToImage: "https://i.kinja-img.com/gawker-media/image/upload/s--Td1ndAMg--/c_fill,fl_progressive,g_center,..."
  ▼ 1: {source: {id: "engadget", name: "Engadget"}, author: "Nathan Ingraham", title: "The best to-do apps",...}
    author: "Nathan Ingraham"
    description: "Plenty of people still rely on pen and paper, but it's inarguable that smartphones have become..."
    publishedAt: "2018-04-17T18:40:00Z"
    ▶ source: {id: "engadget", name: "Engadget"}
      title: "The best to-do apps"
      url: "https://www.engadget.com/2018/04/17/best-to-do-apps/"
      urlToImage: "https://o.aolcdn.com/images/dims?thumbnail=1200%2C630&quality=80&image_uri=https%3A%2F%2Fs.a..."
  ▼ 2: {source: {id: "wired", name: "Wired"}, author: "Nicholas Thompson",...}
    author: "Nicholas Thompson"
    description: "In an interview with WIRED, French President Emmanuel Macron describes his plans to enhance..."
    publishedAt: "2018-03-31T10:00:00Z"
    ▶ source: {id: "wired", name: "Wired"}
      title: "Emmanuel Macron Q&A: France's President Discusses Artificial Intelligence Strategy"
      url: "https://www.wired.com/story/emmanuel-macron-talks-to-wired-about-frances-ai-strategy/"
      urlToImage: "https://media.wired.com/photos/5abd72924fc5f35b3fad572c/191:100/pass/Emmanuel_Macron_140-w.j..."
  ▶ 3: {source: {id: "the-verge", name: "The Verge"}, author: "Jacob Kastrenakes",...}
  ▶ 4: {source: {id: null, name: "Gizmodo.com"}, author: "Matt Novak",...}
  ▶ 5: {source: {id: "mashable", name: "Mashable"}, author: "Mark Kaufman",...}
  ▶ 6: {source: {id: null, name: "Blog.google"}, author: "Sundar PichaiCEO",...}
  ▶ 7: {source: {id: null, name: "Blog.google"}, author: "Alicia CormieKeyword Team",...}
  ▶ 8: {source: {id: null, name: "Themuse.com"}, author: "The Muse",...}
  ▶ 9: {source: {id: null, name: "Monster.com"}, author: "Monster",...}
  ▶ 10: {source: {id: null, name: "Entrepreneur.com"}, author: "Matthew McCreary",...}
  ▶ 11: {source: {id: "business-insider", name: "Business Insider"}, author: "Sara Lopley",...}
  ▶ 12: {source: {id: null, name: "Entrepreneur.com"}, author: "Andrew Yang",...}
  ▶ 13: {source: {id: null, name: "Entrepreneur.com"}, author: "Brian Tracy",...}
  ▶ 14: {source: {id: null, name: "Entrepreneur.com"}, author: "Nina Zipkin",...}
  ▶ 15: {source: {id: null, name: "Entrepreneur.com"}, author: "Ruschelle Khanna",...}
  ▶ 16: {source: {id: null, name: "Smashingmagazine.com"}, author: "Kyle Cassidy",...}
  ▶ 17: {source: {id: null, name: "Smashingmagazine.com"}, author: "Trine Falbe",...}
  ▶ 18: {source: {id: null, name: "Smashingmagazine.com"}, author: null,...}
  ▶ 19: {source: {id: null, name: "Smashingmagazine.com"}, author: null,...}
status: "ok"
totalResults: 922
```

## 2. pielikums

### Entīcija, kas tiek saņemta no Google Cloud Natural Language API

```
0 = {DataMap$Entry@12916} "mentions" -> " size = 1"
1 = {DataMap$Entry@12917} "metadata" -> " size = 0"
2 = {DataMap$Entry@12918} "name" -> "smartphones"
3 = {DataMap$Entry@12919} "salience" -> "0.08075606"
4 = {DataMap$Entry@12920} "type" -> "CONSUMER_GOOD"
```

## JSON dati, kas tiek saglabāti failu sistēmā

```
[
  {
    "name": "Alfred P.",
    "type": "PERSON",
    "url": "https://lifehacker.com/im-doron-weber-alfred-p-sloan-foundation-program-dire-1822771825"
  },
  {
    "name": "Trump",
    "type": "PERSON",
    "url": "https://www.engadget.com/2018/03/12/trump-blocks-broadcom-takeover-of-qualcomm/"
  },
  {
    "name": "Emmanuel Macron",
    "type": "PERSON",
    "url": "https://www.wired.com/story/emmanuel-macron-talks-to-wired-about-frances-ai-strategy/"
  }
]
```

## JSON fails, kas sevi glabā personības testa jautājumus[7]

```
{
  "questions": [
    {
      "title": "1. My best ideas at work come from",
      "selections": [
        "an interchange of ideas and sharing with others.",
        "quiet thought on my own."
      ]
    },
    {
      "title": "2. When I run a meeting, I am usually",
      "selections": [
        "disciplined about following my prepared agenda.",
        "flexible and open to whatever comes up."
      ]
    },
    {
      "title": "3. When handling a subordinate's development, I am most likely to ",
      "selections": [
        "discuss a problem I have with his or her behavior.",
        "dislike telling him or her unpleasant things."
      ]
    },
    {
      "title": "4. I prefer the place I work to be",
      "selections": [
        "structured with clear rules and regulations.",
        "more open-ended and laissez-faire."
      ]
    },
    ...
  ]
}
```

## DOKUMENTĀRĀ LAPA

Maģistra darbs “**Personīgo zināšanu pārvaldības iespējas ar mobilā telefona lietotnes palīdzību**” izstrādāts LU Datorikas fakultātē.

Darba teksta galīgā versija izgatavota **21.05.2018.**

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Fjodors Pohodņevs* \_\_\_\_\_ **21.05.2018.**

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **pieņemrotu / nepieņemrotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: *Dr.dat. Uldis Straujums* \_\_\_\_\_ **21.05.2018.**

Darbs iesniegts **maģistratūras sekretariātā** 21.05.2018.

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: \_\_\_\_\_.

Recenzents: *Dr.dat. Zane Bičevska*

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

\_\_\_\_\_.\_\_\_\_\_.2018 prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_