KARELIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma

Joni Silmäri

LUONNOLLISTA TEKSTIÄ KÄSITTELEVIEN APIEN VERTAILU SEKÄ NIIDEN HYÖDYNTÄMINEN OHJELMISTOROBOTIIKASSA

Opinnäytetyö
Toukokuu 2017

Karelia AMMATTIKORKEAKOULU	**OPINNÄYTETYÖ** **Toukokuu 2017** **Tietotekniikka** Karjalankatu 3 80200 JOENSUU

Tekijä(t)
Joni Silmäri

Nimeke
Luonnollista tekstiä käsittelevien apien vertailu sekä niiden hyödyntäminen ohjelmistorobotiikassa

Toimeksiantaja
CGI Suomi Oy

Opinnäytetyö sisältää vertailun kahden luonnollista tekstiä käsittelevän apin välillä sekä esimerkkisovellukset apien hyödyntämisestä ohjelmistorobotiikassa. Vertailussa olevat luonnollista tekstiä käsittelevät apit olivat helposti saatavilla ja niiden käyttö oli maksutonta koko opinnäytetyön ajan.

Apeja vertaillaan käytöstä aiheutuvien kustannusten ja niiden ominaisuuksien sekä niistä saatavan analyysin perusteella. Vertailu on suoritettu ohjelmistoroboteilla. Ohjelmistorobotit on tehty UiPath–sovelluksella, ja toteutuksessa on käytetty Microsoftin Cognitive Text Analytics -apia sekä IBM Watson Natural Language Understanding -apia. Lyhyt kuvaus ohjelmistorobotista, joka on toteutettu Microsoftin apilla. Ohjelmistorobotille annetaan tiedosto, joka sisältää listan dokumentteja, joihin pitäisi lisätä metadataa. Ohjelmistorobotti käy dokumenttien sisällön läpi yksitellen ja lähettää kunkin dokumentin analysoitavaksi Microsoft Cognitive Text Analytics -apille. Microsoftilta saadaan vastaus, joka sisältää listan avainsanoja, joita tekstistä löytyy. Vastauksena saadut avainsanat lisätään dokumentin metatietoihin ja dokumentti tallennettaan. Dokumentteja analysoidaan niin kauan, kuin läpikäymättömiä dokumentteja on jäljellä.

Testauksen aikana ilmeni se, että IBM:n -api palauttaa huomattavasti laadukkaampaa ja täsmällisempää analyysia kuin Microsoftin -api. Molempien apien kustannukset ovat hyvin samankaltaiset ja molemmille löytyy erinomaisia käyttötarkoituksia. Kumpikaan testatuista apeista ei ollut täydellinen ja varsinkin suomen kielen tuen puuttuminen oli suuri pettymys, mutta tästäkin huolimatta molemmille apeille löytyy varmasti käyttöä.

Kieli	Sivuja 35
suomi	Liitteet 1
	Liitesivumäärä 2

Asiasanat

Luonnollinen tekstinkäsittely, API, Ohjelmistorobotiikka, UiPath

Karelia UNIVERSITY OF APPLIED SCIENCES	**THESIS** **May 2017** **Degree Programme in Information Technology** Karjalankatu 3 80200 JOENSUU FINLAND

Author (s)
Joni Silmäri

Title
Comparison between natural language API's and how to make use of them in software robotics

Commissioned by CGI Suomi

This thesis includes comparison between two natural language API's and two example projects that include examples of how you can make use of them in software robotics. Natural language API's that are compared in this thesis were free and it was easy to gain access to them.

API's were compared by their cost, features and by analyze they produced. Comparison was done with software robots. Software robots were made with UiPath. API's that the software robots used were Microsoft Cognitive Text Analytics and IBM Watson Natural Language Understanding. The main point of comparison is to find out which API is the best at analyzing natural language.
Small brief of my example project that was made with Microsoft's API. At the beginning, we give software robot a file that contains a list of documents that don't have any metadata. Software robot starts with the first document and sends it contents to Microsoft Cognitive Text Analytics API. After that the software robot gets a response from Microsoft with bunch of keywords from the text that it just sent. The software robot proceeds to pick ten of those keywords and adds them to the documents tags and saves the document. Software robot continues this until all documents have been tagged.

During testing It was obvious, that IBM's API did return a lot more better analyze and it was a lot more customizable than Microsoft's API. Neither of the API's is perfect and worst part for both, was lack of support for Finnish language. Cost of both API's is similar and they both are great API's in general.

Language	Pages 35 Appendices 1 Pages of Appendices 2
Finnish	

Keywords

Natural language, API, Software Robotics, UiPath

Sisältö

Lyhenteet ja termit

API Application programming interface, ohjelmointirajapinta. Joukko määritelmiä, joiden avulla eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja eli keskustella keskenään. [13.]

CGI CGI on kansainvälisen yrityksen nimi ja se on lyhenne sanoista Consultants to Government and Industry. [17.]

JSON Java Script Object Notation on yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen. [12.]

HTTP Hypertext Transfer Protocol on protokolla, jota selaimet ja www-palvelimet käyttävät tiedonsiirtoon. Protokolla perustuu siihen, että asiakasohjelma avaa TCP-yhteyden palvelimelle ja lähettää pyynnön. Palvelin vastaa lähettämällä sopivan vastauksen, tavallisimmin HTML-sivun tai binääridataa kuten kuvia, ohjelmia tai ääntä. [11.]

1 Johdanto

Tässä dokumentissa kuvaillaan opinnäytetyön aikana tehtyä vertailua luonnollisten tekstinkäsittely -apien välillä, ja kuvaillaan konseptityönä tehtyjen ohjelmistorobottien työvaiheita aina suunnittelusta loppuviimeistelyihin. Opinnäytetyötä ohjasi Joni Ranta ja Petri Laitinen oli tarkastajana. Opinnäytetyö suoritettiin osana tutkintoa.

1.1 Opinnäytetyön toimeksiantaja

Opinnäytetyön toimeksiantaja oli CGI Suomi Oy. CGI on perustettu vuonna 1976 ja sen perustajat olivat Serge Godin sekä André Imbeau. CGI:n pääkonttori sijaitsee Kanadassa, Montrealissa. CGI tarjoaa palveluja it:n ja liiketoimintaprosessien kehittämisen tueksi. CGI:n palveluksessa on noin 70 000 työntekijää sadoissa toimipisteissä Pohjois- ja Etelä-Amerikassa, Euroopassa sekä Aasian ja Tyynenmeren alueilla. CGI:llä on suomessa 17 toimipistettä, niissä työskentelee yhteensä 3300 työntekijää. [17.]

1.2 Opinnäytetyön idea

Idea opinnäytetyöhön tuli työharjoittelussa tehdyn konseptityön aikana. Konseptityön aihe oli mielenkiintoinen ja kysyin esimieheltäni voisinko tehdä aiheesta opinnäytetyön. Esimieheni mielestä aihe oli hyvä ja aloimme miettiä mahdollisia tulokulmia opinnäytetyön suhteen. Päädyimme tekemään vertailua markkinoilta löytyvien apien välillä. Seuraavaksi vuorossa oli opinnäytetyön suunnitelman tekeminen. Opinnäytetyön suunnitelma hyväksyttiin koulun puolelta ja opinnäytetyön tekeminen alkoi.

1.3 Opinnäytetyön tavoitteet

Opinnäytetyön tavoitteena oli saada selville, mikä olisi paras mahdollinen luonnollista tekstiä käsittelevä api ohjelmistorobotiikan sovelluksiin. Tavoitteeseen pääseminen edellyttäisi kahden ohjelmistorobotin valmiiksi saamisen ja niiden vertailun.

2 Kehitysohjelman ja apien valinta

Opinnäytetyöhön tutustuminen alkoi kartoittamalla netistä löytyviä luonnollista tekstiä käsitteleviä apeja. [9]

Kehitysohjelman valinta oli nopea prosessi ja ohjelmaksi valikoitui UiPath. Valinnan perusteena oli se, että ohjelmasta tulisi löytyä jonkin näköistä tutustumismateriaalia sekä ohjelman tulisi olla ilmainen, lisäksi eduksi katsottiin Windows käyttöjärjestelmän tuki.

Tutustuminen UiPathiin alkoi videoiden katselulla ja ilmaisen studion latauksella. Harjoitusten tekemisen jälkeen omasin tarvittavat perustiedot UiPathin käyttöön ja tutustuminen luonnollista tekstiä käsitteleviin apeihin alkoi. [7;8]

Luonnollista tekstiä käsitteleviä apeja löytyy netistä useampiakin, mutta valitut apit olivat Microsoftin Cognitive Services Text Analytics -api sekä IBM Watson Natural Language Understanding -api.

Apin valintakriteerinä oli se, että apin tulisi analysoida avainsanoja annetusta tekstistä sekä sen tulisi olla ilmainen. Googlen Cloud Natural Language -api:a ei valittu sen vuoksi, että sen käyttö olisi edellyttänyt pankkikortin tunnusten antamista ja api vaikutti verrokkeja heikommalta.

Microsoftilta löytyy hyvä dokumentti apin käytöstä ja siitä mitä kaikkea apilla on mahdollista tehdä. [2.]

IBM:ltä löytyy hyvin laaja dokumentaatio apin käytöstä. [5.] Dokumentaatiossa näytetään monia erilaisia tapoja apin hyödyntämiseen ja apin ominaisuuksia kuvailtiin hyvin laajasti.

2.1 Luonnollista tekstiä käsittelevät apit

Opinnäytetyössä vertailtiin kahta erilaista luonnollista tekstiä käsittelevää apia. Vertailussa olivat Microsoft Cognitive Services Text Analytics -api ja IBM Watson Natural Language Understanding -api.

Molempien apien toiminta on samantapainen: käyttäjä syöttää apille tekstiä ja määrittää, miten teksti tulisi analysoida. Esimerkiksi api voisi analysoida tekstissä ilmenevät avainsanat ja tekstin sävyn. Tekstin analysoinnin jälkeen apilta saadaan vastaus, joka pitää sisällään tekstin kannalta merkittäviä avainsanoja sekä sen, onko teksti positiivista vai negatiivista. Tämän jälkeen käyttäjä voi hyödyntää saatua dataa omien tarpeidensa mukaan.

Opinnäytetyöhön liittyy myös kaksi ohjelmistorobottia, joiden avulla havainnollistetaan, miten luonnollista tekstiä käsitteleviä apeja voi hyödyntää ohjelmistorobotiikassa.

2.2 UiPath

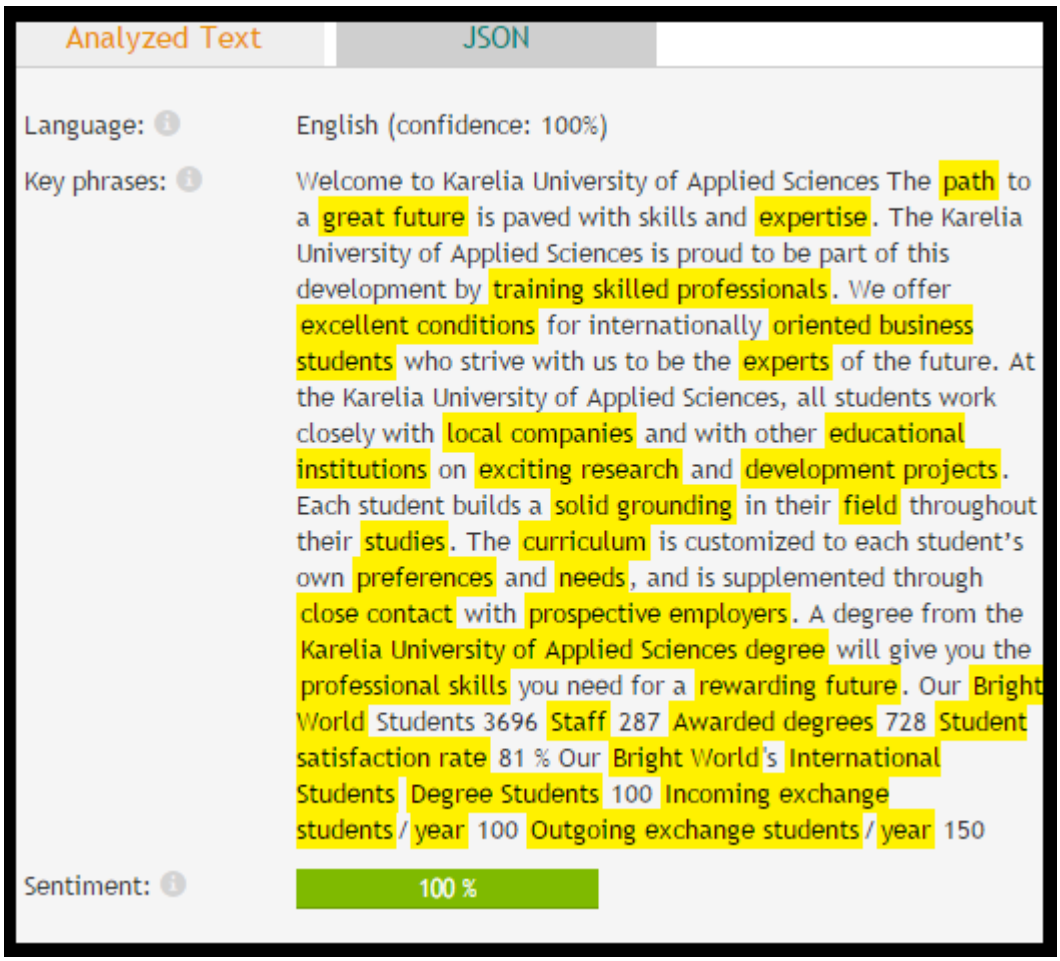
UiPath on ohjelmistorobottien kehittämistä varten tehty ohjelma. UiPathilla automatisoidaan erilaisia tietoteknisiä prosesseja, joko kokonaisuudessaan tai osittain. Automatisointia voidaan, tehdä hyvinkin helposti tehtäviin, jotka ovat sääntöpohjaisia ja puolestaan tehtävät, jotka vaativat luonnollista päättelyä ovat huomattavasti vaikeampi automatisoida. UiPathin ohjelmointikieli on VisualBasic, mutta halutessaan käyttäjä voi luoda omia kirjastoja, joiden avulla ohjelmointi onnistuu esimerkiksi C# -kielellä. [20.]

UiPathista tulee hyvin nopeasti mieleen Microsoft Visio ensi silmäyksellä, mutta todellisuudessa se on paljon muutakin.

Jos UiPath kiinnostaa, siitä on saatavilla ilmainen community –versio sekä heidän nettisivuillaan on mahdollisuus suorittaa ohjelmistorobotiikan perusteista sertifikaatti. Sertifikaatti pitää sisällään laadukkaasti tehtyjä videoita sekä käytännön esimerkkejä sisältäviä harjoitustehtäviä. [7;8]

3 Microsoft Cognitive Services Text Analytics -api

Microsoft Cognitive Services Text Analytics -api on Microsoftin kehittämä ohjelmoinitirajapinta, joka mahdollistaa järjestelemättömän tekstin analysoimisen, tekstin kirjoituskielen tunnistamisen, tekstin sävyn sekä avainsanojen tunnistamisen. Cognitive Servicesin Text Analyticsin on lyhennetty tämän kappaleen luvuissa CSTA -lyhenteellä sisällysluettelon selkeyttämisen vuoksi. Kuvassa 1 on nähtävissä esimerkki analyysistä, jota saadaan Microsoftin apilta. [1.]



Kuva 1. Microsoftilta saatava analyysi [1.]

3.1 Microsoft CSTA -api:n ominaisuudet

Tässä luvussa käydään läpi Microsoft Cognitive Services Text Analytics -apin ominaisuudet, niiden mahdolliset rajoitukset sekä sen miten kutakin ominaisuutta voisi hyödyntää.

Sentiment analysis

Sentiment analysis kertoo lukijalle, onko teksti positiivista vai negatiivista. Tekstin sävystä tulee numeerinen arvo nollan ja yhden väliltä. Nolla tarkoittaa sitä, että annettu teksti on negatiivista ja yksi tarkoittaa sitä, että teksti on hyvin positiivista. Tuetut kielet tälle ominaisuudelle ovat englanti, ranska, espanja ja portugali. Ominaisuus on tarkoitettu asiakaspalautteen positiivisuuden ja negatiivisuuden analysointia varten. [1.]

Kuvassa 1 on nähtävissä esimerkki ominaisuuden toiminnasta, vihreän palkin täyttymismäärä kuvaa arvoa nollan ja yhden väliltä.

Key phrase extraction

Key Phrases palauttaa tekstissä ilmeneviä avainsanoja. Avainsanoja palautetaan tietty määrä ja niitä ei järjestellä mitenkään. Microsoftilta löytyvästä dokumentaatiosta ei käy ilmi millä logiikalla palautuneet avainsanat on valittu. Tuetut kielet key phrases extractionille ovat englanti, saksa, espanja ja japani. Key phrase extraction ominaisuus on rajoitettu siten, että käyttäjä pystyy lähettämään analysointipyynnön dokumentille, joka on maksimissaan 5000 merkkiä pitkä. Ominaisuutta voidaan hyödyntää monilla eri tavoin: esimerkiksi metadatan lisäyksellä dokumentteihin tai vaikkapa tägien lisäyksellä nettisivuille. [1.]

Kuvassa 1 on esimerkki palautuneista avainsanoista.

Topic detection

Topic detection tunnistaa dokumentin sisällön ja palauttaa sen perusteella yhden avainsanan ja antaa sille numeerisen arvon. Topic detectionin huono puoli on siinä, että käyttäjän tulee antaa vähintään 100 dokumenttia tai muuta tekstilähdettä kerralla, muuten kutsua ei voida tehdä. Topic detection on tarkoitettu lähinnä asiakaspalautteen analysoimista varten. [1.]

Language detection

Language detection tunnistaa kielen millä teksti on kirjoitettu. Ominaisuus palauttaa numeerisen arvon nollan ja yhden väliltä. Jos palautunut arvo on lähellä yhtä se tarkoittaa sitä, että kieli on tunnettu lähes 100%:n varmuudella. Ominaisuus tukee 120 eri kieltä. Ominaisuutta voitaisiin käyttää erilaisten kääntöohjelmien apuna. [1.]

3.2 Microsoft CSTA -api:n käytöstä aiheutuvat kustannukset

Palvelu on hinnoiteltu kohtuullisesti, sillä 100000 -kutsua kuukautta kohden on jo hyvin massiivinen määrä esimerkiksi dokumentteja ja hinta tuolle määrälle olisi 150 dollaria kuukaudessa. Pienessä mittakaavassa käytöstä ei aiheudu mitään kustannuksia, sillä jos kuukauden aikana kertyy enintään 5000 -kutsua palvelu on maksuton. [18.]

Palvelun on hinnoiteltu taulukon 1 mukaisesti.

Taulukko 1. Microsoft Cognitive Services api:n hinnoittelu

Kutsujen määrä (/kk)	**Hinta ($)**
5 000	0
100 000	150
500 000	500
2 500 000	1250
10 000 000	2500

3.3 Microsoft CSTA -api:n käyttöönottaminen

Microsoft Cognitive Services -apin käyttöönottaminen vaatii Microsoft tunnuksia, jos niitä ei löydy niin ne täytyy luoda. Palvelun voi ottaa käyttöön alla olevasta linkistä.

https://www.microsoft.com/cognitive-services/en-US/sign-up?ReturnUrl=/cognitive-services/en-us/subscriptions?productId=%2fproducts%2f56f3d5f6eda56503ecb1ab78

Kirjautumisen jälkeen käyttäjä valitsee Text Analytics –palvelun. Palvelusta saatava avain kannattaa kopioida, sillä sitä tullaan tarvitsemaan myöhemmin http-pyyntöjen headerosiossa.

3.4 Microsoft CSTA -api:n toiminnan kuvaus

Microsoft Cognitive Services -api:in lähetetään http-pyyntöjä, ja sieltä saadaan vastauksena haluttua analyysia json -muodossa.

Pyynnön headerosiossa täytyy antaa henkilökohtainen apiavain, joka pitää kirjaa lähetetyistä analysointi pyynnöistä. Tekstin bodyosiossa määritetään teksti, joka tulisi analysoida sekä, miten teksti tulisi analysoida. Esimerkiksi tahdotaanko dokumentista tietää avainsanat.

Vastauksena saadaan halutulla tavalla analysoitua dataa JSON -muodossa, esimerkiksi tekstistä poimitut avainsanat.

Toimintaa on kuvattu huomattavasti laajemmin esimerkkisovelluksessa, ja se on nähtävissä tämän dokumentin viidennessä luvussa.

4 IBM Watson Natural Language Understanding -api

IBM Watson Natural Language Understanding -api on IBM:n kehittämä ohjelmointirajapinta, joka tunnistaa sille lähetetystä tekstistä liudan erilaisia asioita, ja antaa numeerisen arvon ilmestymisen tärkeyden mukaisesti. IBM Watson Natural Language Understanding -api:n tuetut kielet ovat, englanti, ranska, saksa, italia, portugali, venäjä, espanja, ruotsi. [4.]

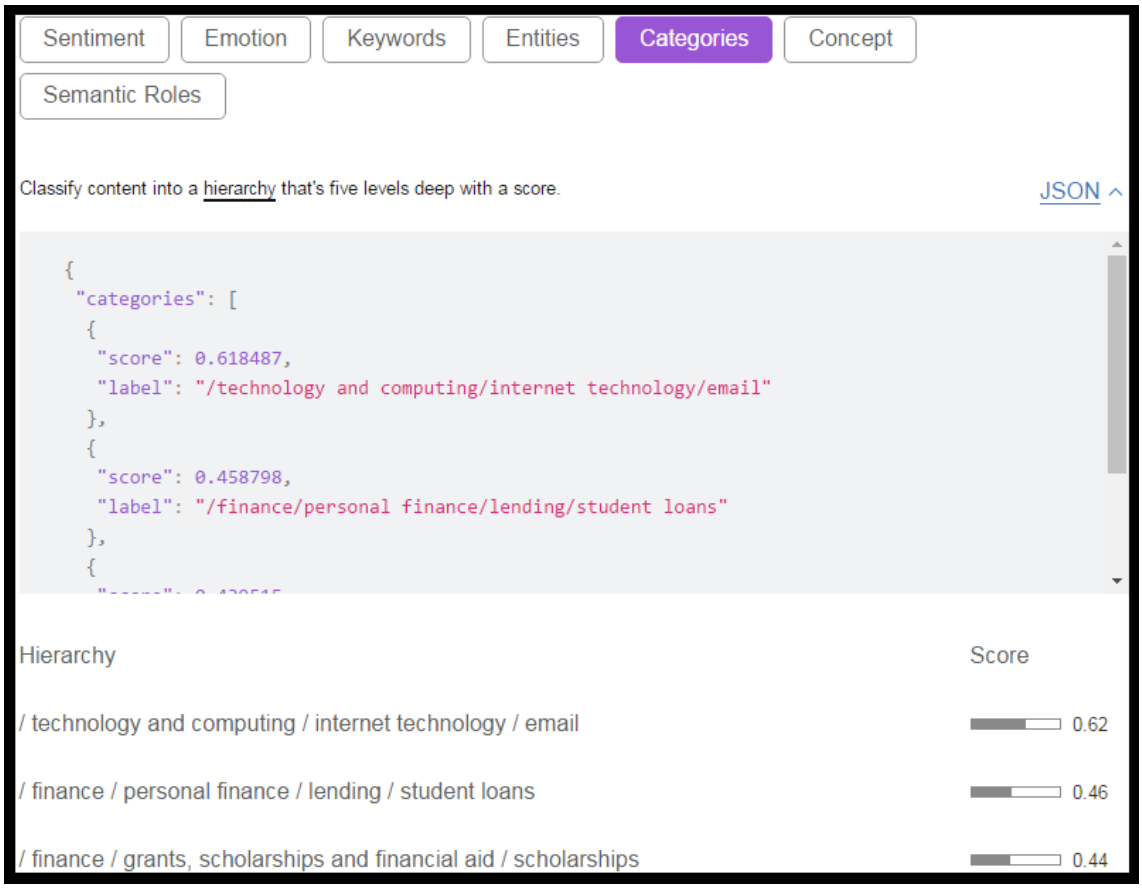
Natural Language Understandingin on lyhennetty tämän kappaleen luvuissa NLU -lyhenteellä sisällysluettelon muotoilun vuoksi.

4.1 IBM Watson NLU -api:n ominaisuudet

Ominaisuudet käydään läpi yksitellen ja niitä kuvaillaan mahdollisimman tarkasti, lyhyen ja ytimekkään selityksen sekä kuvien avulla.

Categories

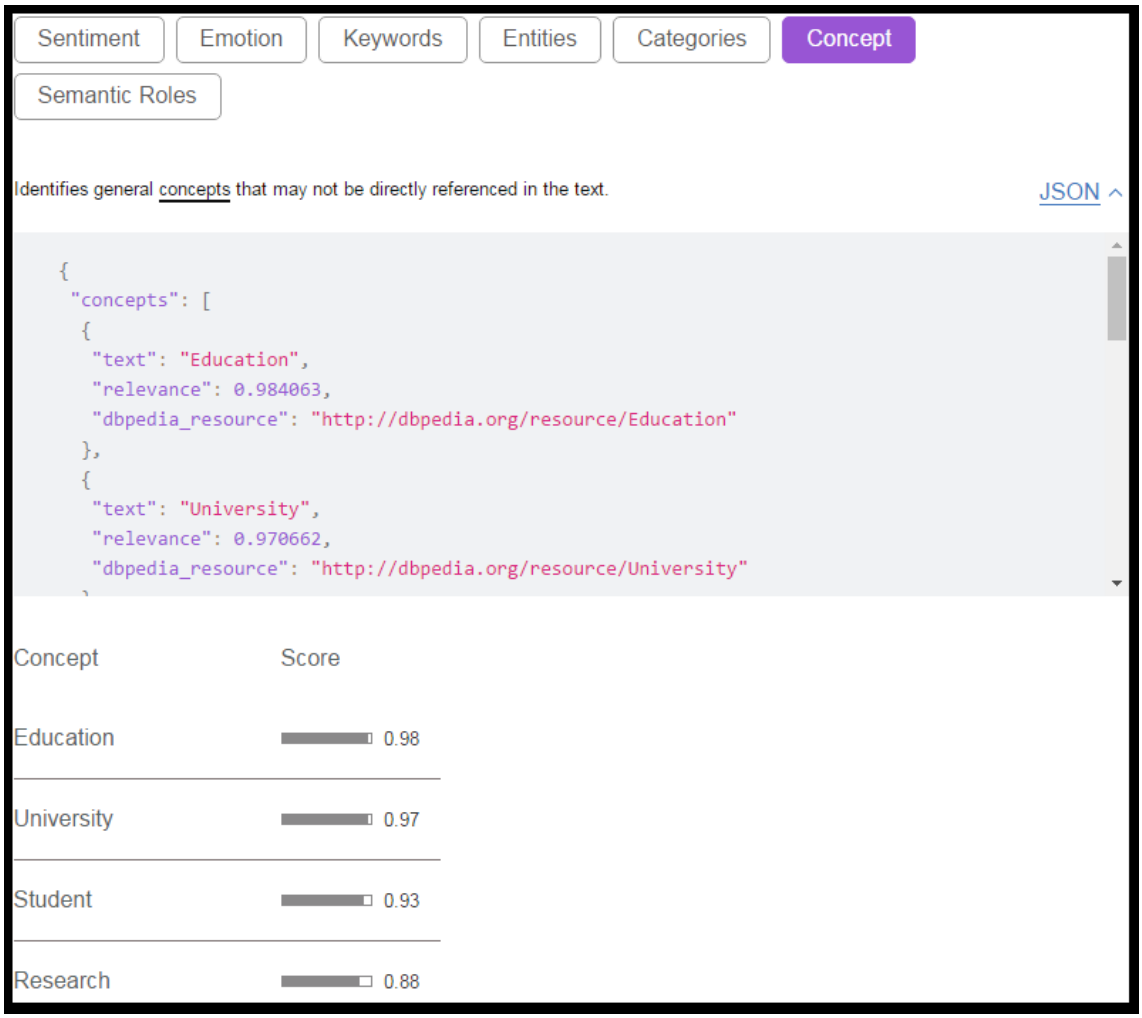
Categories listaa annetussa tekstissä ilmenevät kategoriat, ja antaa kategorioille numeerisen arvon. Mitä lähempänä numeerinen arvo on yhtä, sitä varmemmin tekstistä on tunnistettu jokin kategoria. Kuvassa 2 on vastauksena saatu analyysi categories ominaisuudella. [4;6]



Kuva 2. Categories –analyysi annetusta tekstistä [4.]

Concepts

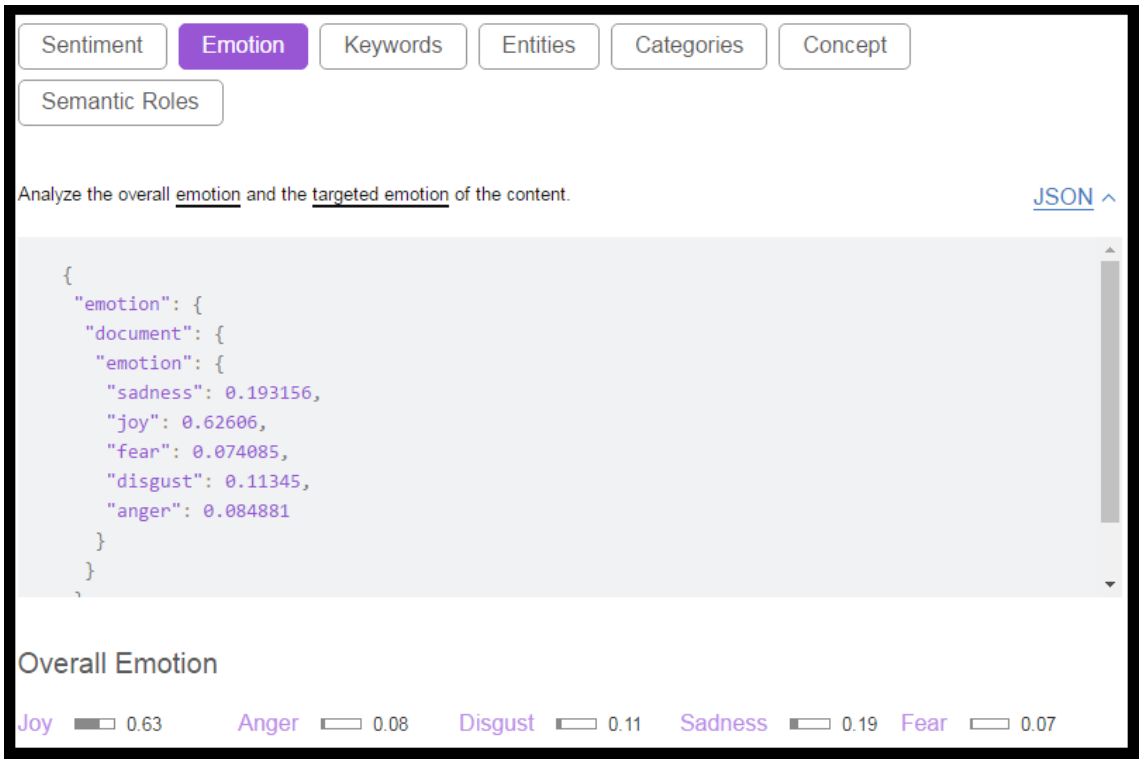
Concepts tunnistaa käsitteitä, joita tekstissä ei välttämättä ilmoiteta suoraan, mutta mihin teksti kuitenkin liittyy olennaisesti. Löydetyt konseptit palautetaan numeerisina arvoina nollasta yhteen ja mitä lähempänä arvo on yhtä, sitä varmemmin konsepti on tunnistettu. Kuvassa 3 on vastauksena saatu analyysi concepts ominaisuudella. [4;6]



Kuva 3. Concept –analyysi annetusta tekstistä [4.]

Emotion

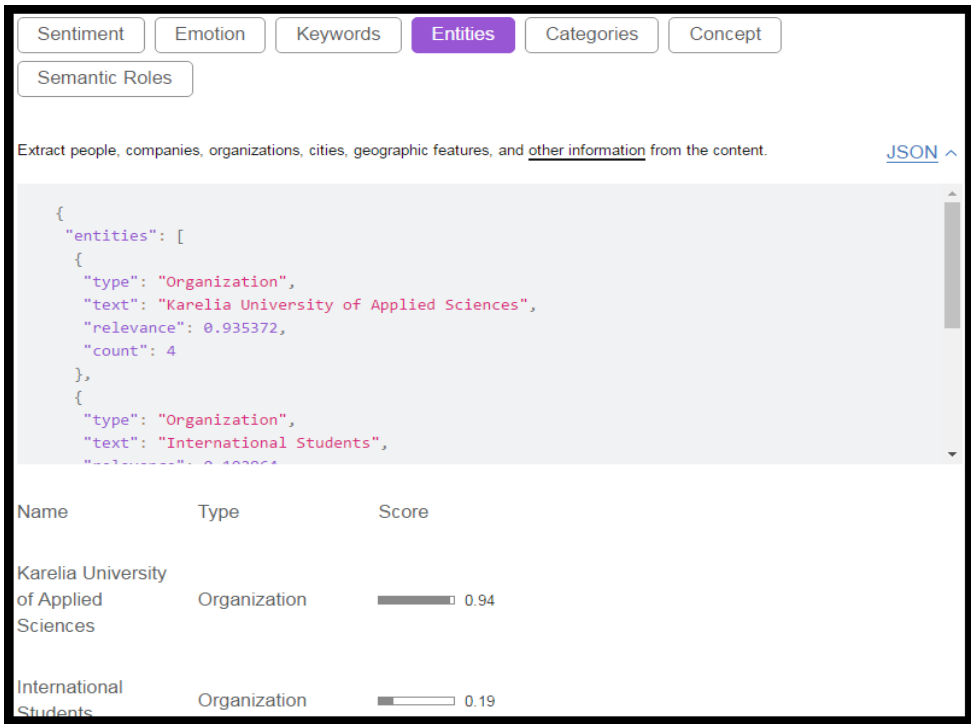
Emotion analysoi tekstissä tai tietyissä sanoissa esiintyvät tunnetilat. Emotion palauttaa numeerisen arvon tiettyjen sanojen tai koko tekstin tunnetilasta. Ominaisuus määrittää miten todennäköisesti tekstissä esiintyy iloa, vihaa, inhottavuutta, surua ja pelkoa. Kuvassa 4 on vastauksena saatu analyysi emotion ominaisuudella. [4;6]



Kuva 4. Emotion -analyysi annetusta tekstistä [4.]

Entities

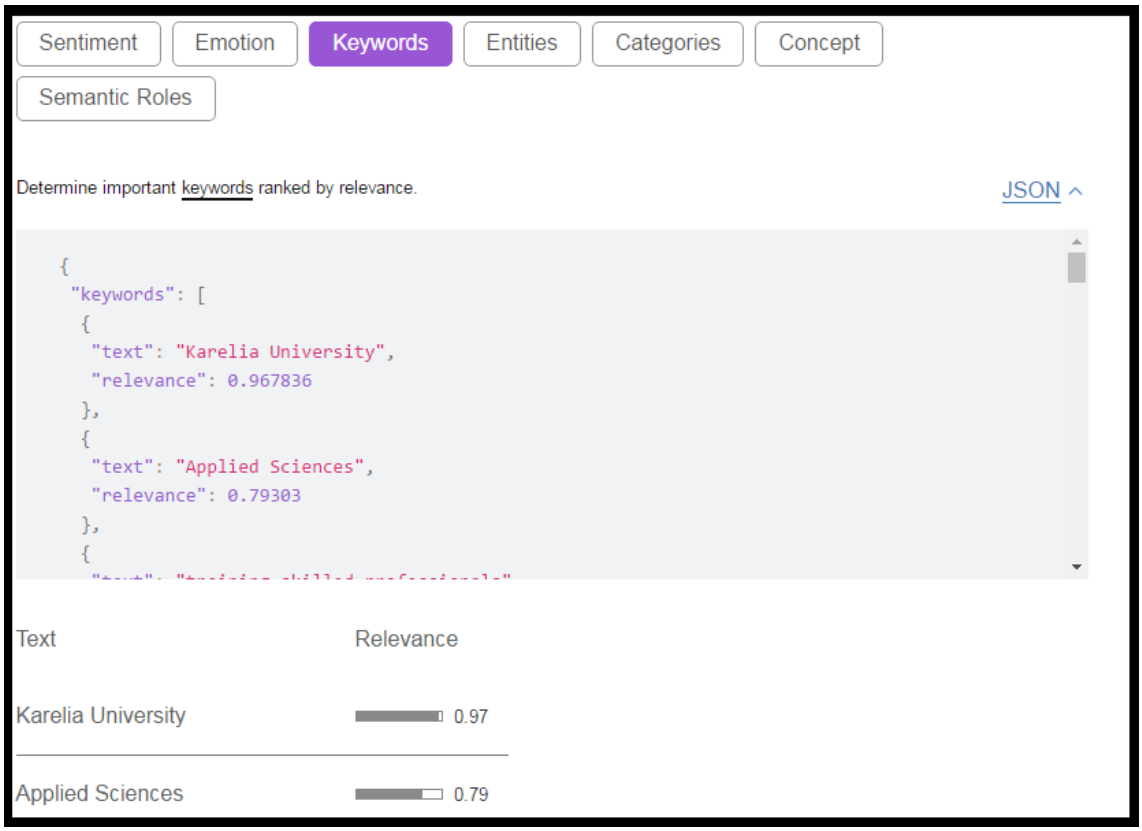
Entities listaa dokumentissa esiintyvät ihmiset, paikat, tapahtumat ja sijainnit. Ominaisuus antaa löydöksille numeerisen arvon, löydön varmuuden mukaisesti. Kuvassa 5 on esimerkki analyysi entities ominaisuudella saatavasta vastauksesta. [4;6]



Kuva 5. Entities -analyysi annetusta tekstistä [4.]

Keywords

Keywords listaa tekstin merkittävimmät avainsanat. Ominaisuus järjestelee esiintyneet avainsanat numeeriseen paremmuusjärjestykseen, ja palauttaa avainsanoja halutun lukumäärän. Kuvassa 6 on esimerkki analyysi keywords ominaisuudella saatavasta vastauksesta. [4;6]



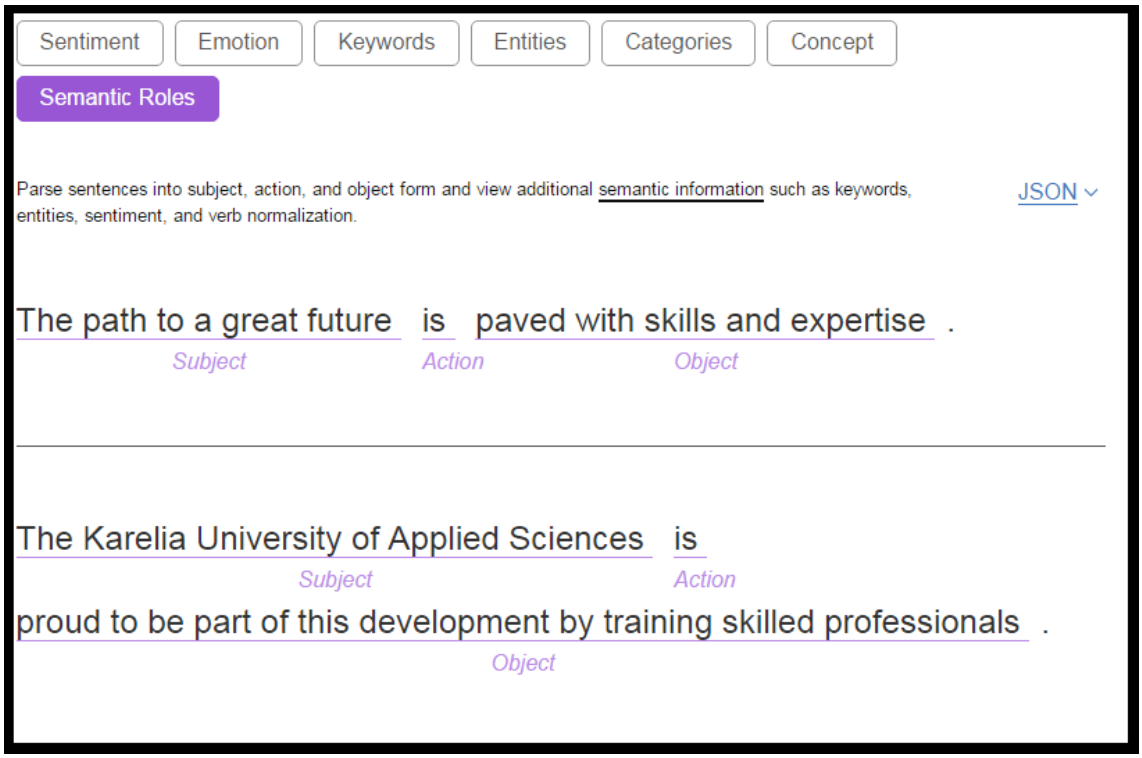
Kuva 6. Keywords -analyysi annetusta tekstistä [4.]

Metadata

Metadata ominaisuudelle täytyy antaa HTML-tiedosto tai URL-osoite parametrina. Ominaisuus palauttaa vastauksena web-sivun tekijän, sivun otsikon ja sivun julkaisupäivämäärän. [6.]

Semantic Roles

Semantic Roles määrittää yksittäisestä lauseesta kohteen tai henkilön, toimintaan tai tekemiseen sekä siihen mitä tai missä tämä tapahtui. Kuvassa 7 on esimerkki analyysi semantic roles ominaisuudella saatavasta vastauksesta. [4;6]



Kuva 7. Semantic Roles -analyysi annetusta tekstistä [4.]

Sentiment

Sentiment ominaisuus tunnistaa tekstin positiivisuuden tai negatiivisuuden, ja antaa sille numeerisen arvon miinus yhden ja yhden väliltä. Kuvassa 8 on esimerkki analyysi sentiment ominaisuudella saatavasta vastauksesta. [4;6]

Kuva 8. Sentiment -analyysi annetusta tekstistä [4.]

4.2 IBM Watson NLU -api:n käytöstä aiheutuvat kustannukset

IBM Watson Natural Language Understanding -api:n käytöstä ei tule kustannuksia, jos kutsuja tehdään enintään 1000 päivässä. Tämän määrän ylittyessä palvelu alkaa maksaa kutsu kohtaisesti taulukon 2 mukaisesti. [19.]

Taulukko 2. IBM Watson Natural Language Understanding -api:n hinnoittelu [19.]

Kutsujen määrä (kk)	**Hinta ($)**
1 - 250 000	0.003
250 001 - 5 000 000	0.001
5 000 000+	0.00002

4.3 IBM Watson NLU -api:n käyttöönottaminen

IBM Watson Natural Language Understanding -api otetaan käyttöön siten, että käyttäjä rekisteröityy alla olevassa osoitteessa.

https://console.ng.bluemix.net/registration/?target=/catalog/services/natural-language-understanding/

Rekisteröitymisen jälkeen käyttäjä käy aktivoimassa Natural Language Understandingin apin käyttöönsä.

Aktivoimisen jälkeen käyttäjän kannattaa ottaa api:n käyttäjänimi sekä salasana ylös, sillä niitä tullaan käyttämään cURL kutsujen yhteydessä.

4.4 IBM Watson NLU -api:n toiminnan kuvaus

IBM Watson Natural Language Understanding palveluun lähetetään cURL pyyntöjä, ja palvelusta saadaan vastauksena analyysia JSON –muodossa. Kuvassa 9 on selkeytetty komentoa jakamalla se osiin, lisäksi jokaisen osan tarkoitus on selitetty kuvan alapuolella. [10;16.]

```
1. curl -X POST -H "Content-Type: application/json"
2. -u "käyttäjänimi":"salasana"
3. -d @parameters.json
4. "https://gateway.watsonplatform.net/natural-language-understanding/api/v1/analyze?ve
5. -o output.json
```

Kuva 9. cURL komento millä pyyntö tehdään.

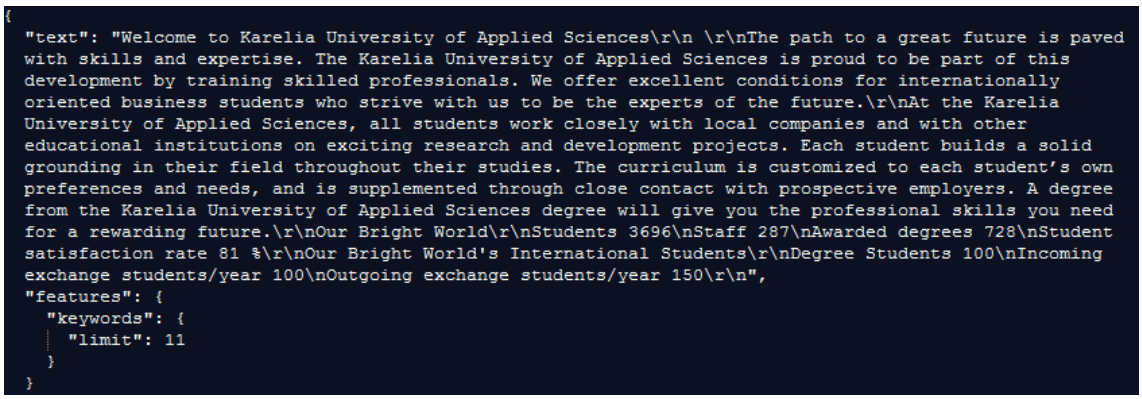
1. Määritetään käytettävä protokolla sekä lähetyksen sisältötyyppi.

2. Määritetään henkilökohtaiset käyttäjätunnukset sekä salasana. Kummatkin saa bluemixistä apin käyttöönoton yhteydessä.

3. Liittenä lähetettävä parameters.json –tiedostosta (kuva 10). Tiedostossa määritellään analysoitavaksi lähetettävä teksti, ja se miten teksti tulisi analysoida.

4. Osoite minne pyyntö lähetetään.

5. Määritetään tiedosto, minne vastuksena saatu analyysi tallennetaan.



Kuva 10. Esimerkki parameters tiedoston sisällöstä

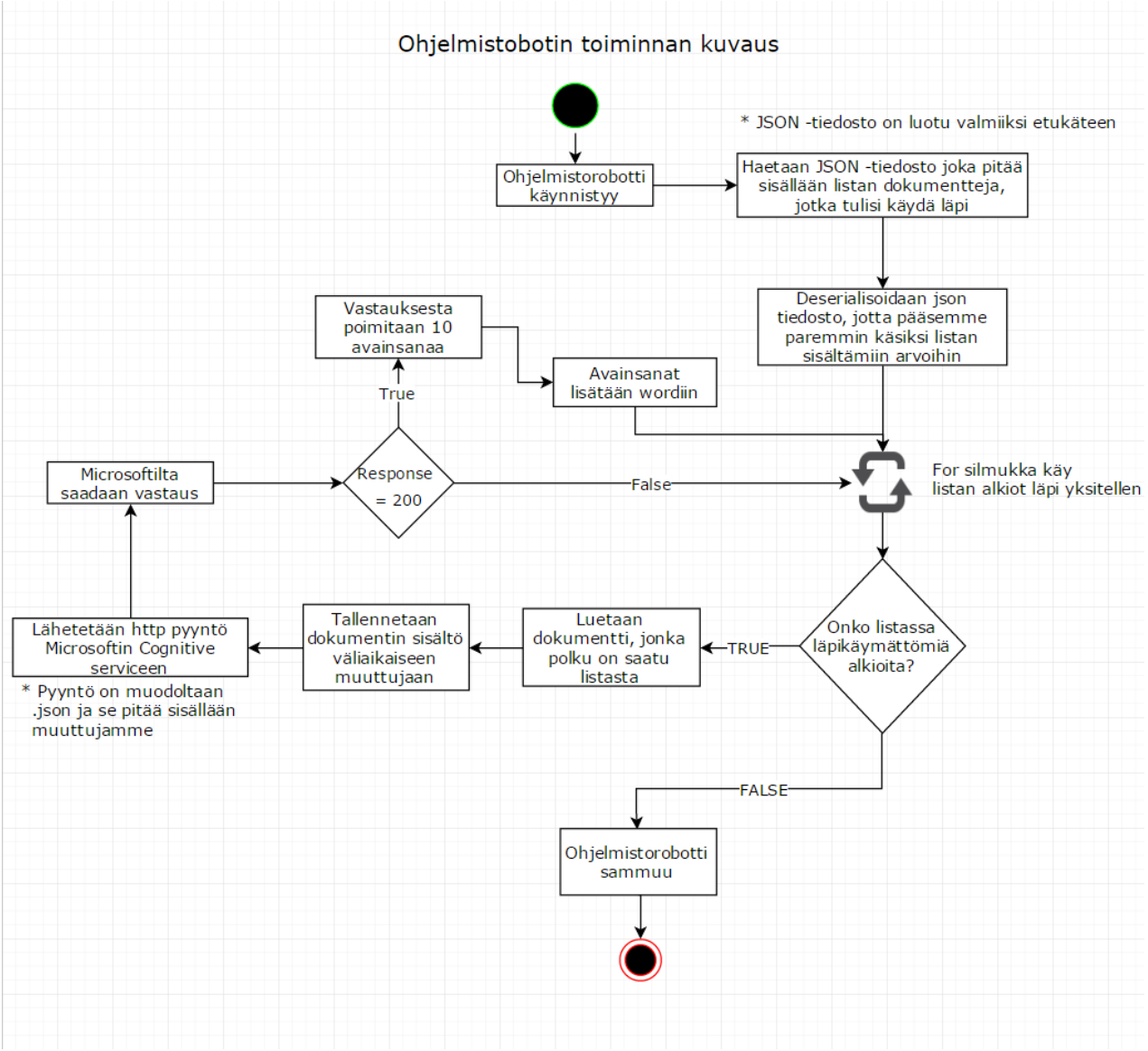
5 Microsoftin apia käyttävän UiPath -sovelluksen toteutus

Tässä luvussa kuvaillaan Microsoft Cognitive Services Text Analytics -api:a käyttävän UiPath -sovelluksen toteutusta vaiheittain, erilaisten kaavioiden sekä kuvien avulla.

Ohjelmistorobotin toiminnan kuvaus kokonaisuudessaan

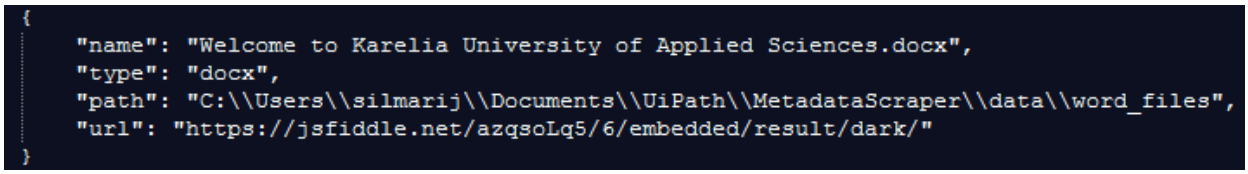
Kokonaisuudessaan kyseessä on hyvin laaja toimintaketju ja sitä olisi hyvin haastava kuvailla tämän dokumentin lukijalle ilman, että kokonaisuutta pilkkoisi pienemmiksi osiksi. Ohjelmistorobotin toimintaa on kuvailtu kaaviossa 1.

Kaavio 1. Microsoftin -api:a käyttävän ohjelmistorobotin toiminnan kuvaus



Dokumenttien listaus

Ennen kuin ohjelmistorobotti voi aloitta työt, se tarvitsee listan dokumenteista, jotka tulisi analysoida. Tämän vaiheen olisi voinut toteuttaa automaattisesti, mutta se olisi vaatinut huomattavasti enemmän aikaa. Ajan säästämisen vuoksi ohjelmistorobotille annetaan ennalta luotu JSON –tiedosto, joka pitää sisällään JSON -listan. Jokainen listan alkio (kuva 11) pitää sisällään seuraavat muuttujat: tiedoston nimi, tyyppi, sijainti ja url-osoite.



```
{
    "name": "Welcome to Karelia University of Applied Sciences.docx",
    "type": "docx",
    "path": "C:\\Users\\silmarij\\Documents\\UiPath\\MetadataScraper\\data\\word_files",
    "url": "https://jsfiddle.net/azqsoLq5/6/embedded/result/dark/"
}
```


Kuva 11. JSON-listan alkio kaikkine muuttujineen

Listattujen dokumenttien sisällön hakeminen

Ohjelmistorobotti aloittaa työt sillä, että se alkaa käydä JSON –listan alkioita läpi yksitellen. Ohjelmistorobotti lukee tiedostoja sijainnin ja nimen mukaisella logiikalla. Kun tiedosto on avattu, sen sisältö kopioidaan textToAnalyze muuttujaan. TextToAnalyze muuttujasta pitää korvata lainausmerkit heittomerkeillä, sillä JSON -kutsua ei voida tehdä, jos se pitää sisällään lainausmerkkejä. Muuttujan pituus täytyy pakottaa 5000 -merkkiin sillä Microsoftin Cognitive Services Text Analytics -api ei pysty analysoimaan sitä suurempaa tekstimäärää.

Dokumentin sisällön lähetystä varten tarvitaan yksi muuttuja lisää ja tässä tapauksessa se on Temp (kuva 12) niminen. Muuttuja pitää sisällään JSON muotoista tekstiä sekä textToAnalyze muuttujan. Muuttujasta löytyy määritys tekstin kielelle sekä dokumentin yksilöivä id tunniste.

```
"{" +
    """documents"": [" +
        "{"+
            """language"": ""en"","+
            """id"":"""+analyzeIdentifier.ToString+""","+
            """text"":""" + textToAnalyze.Replace("""","'")+"""+
        "}"+
    "]"+
"}"
```

Kuva 12. Temp -muuttuja kokonaisuudessaan

Dokumentin sisällön lähetys

Dokumentin sisältö lähetetään HTTP POST pyynnöllä. Postin headerosioon täytyy lisätä henkilökohtainen tunnus, joka pitää kirjaa tehtyjen pyyntöjen lukumäärästä. Postin Body –osioon lisätään edellisessä luvussa mainittu Temp -muuttuja (kuva 12). Lisäksi Postista saatava vastaus tallennettaan response -muuttujaksi ja postin status tallennetaan resStatus -muuttujaksi.

Vastauksena saadun JSON -tiedoston läpikäynti

Mikäli resStatus -muuttujan arvo on 200 tarkoittaa se sitä, että pyyntö onnistui. Onnistuneen pyynnön jälkeen ohjelmistorobotti alkaa käydä läpi response -muuttujaa ja muuttujasta poimitaan kymmenen avainsanaa. Avainsanat tallennetaan väliaikaiseen muuttujaan keyWords. Avainsanojen lisäyksen olisi voitu aloittaa jo tässä vaiheessa, mutta ohjelmasta tulisi huomattavasti raskaampi, jos jokainen sana pitäisi lisätä yksitellen.

Avainsanojen lisäys dokumenttiin

Kun kymmenen avainsanaa on tallennettu keyWords -muuttujaan, ohjelmistorobotti avaa word –tiedoston sille annetun JSON –listassa määritetyn osoitteen mukaisesti ja asettaa keyWords muuttujan arvon tunnisteet kenttään. Tämän jälkeen word -dokumentti tallennetaan samalla nimellä ja ohjelmistorobotti siirtyy seuraavaan dokumenttiin. Kun kaikki dokumentit on käyty läpi ohjelmistorobotti lopettaa prosessin suorittamisen.

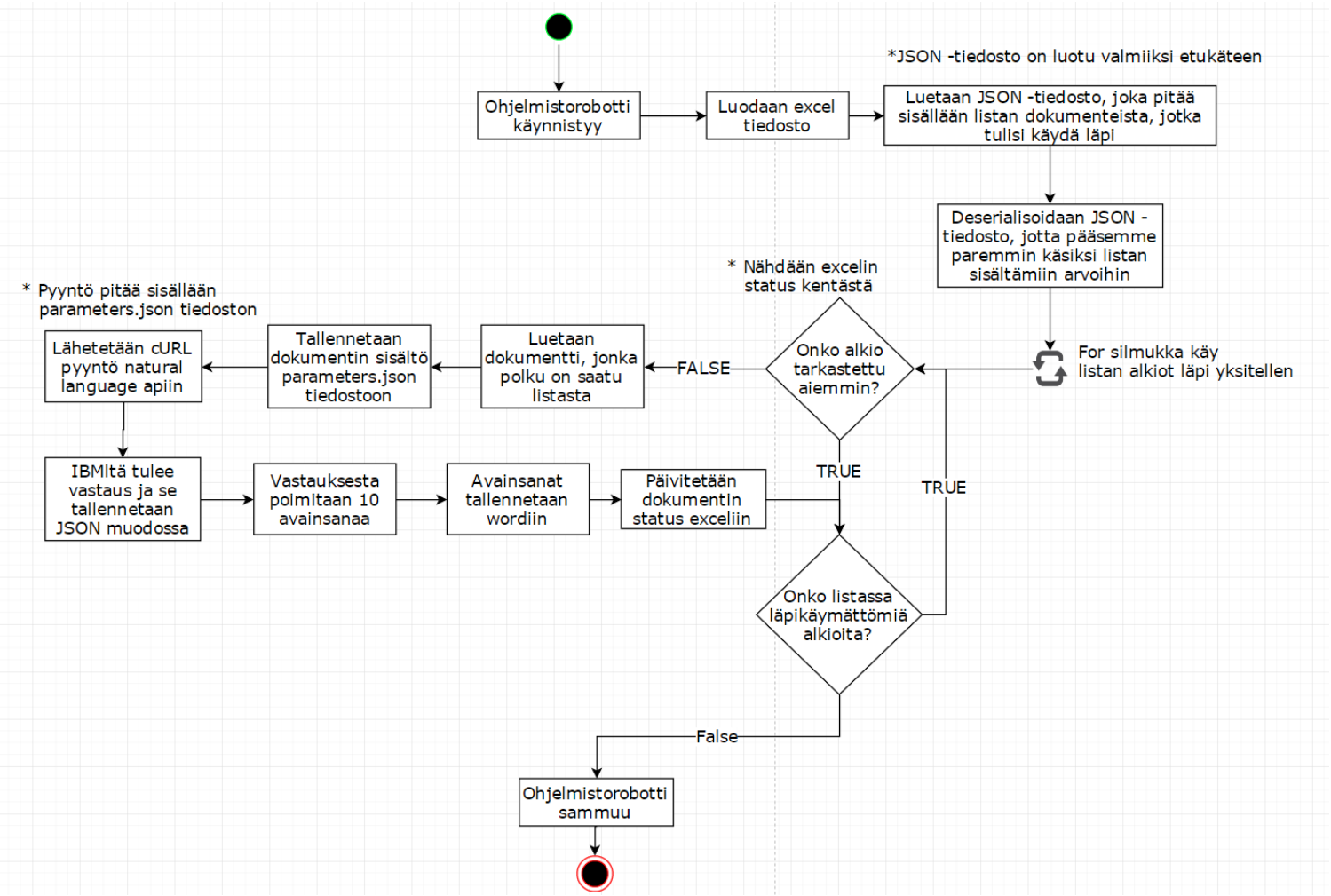
6 IBM:n apia käyttävän UiPath -sovelluksen toteutus

Tässä luvussa kuvaillaan IBM Watson Natural Language Understanding -api;a käyttävän UiPath sovelluksen toteutusta vaiheittain, erilaisten kaavioiden sekä kuvien avulla.

Ohjelmistorobotin toiminnan kuvaus kokonaisuudessaan

Kokonaisuudessaan kyseessä on hyvin laaja toimintaketju ja sitä olisi hyvin haastava kuvailla tämän dokumentin lukijalle ilman, että kokonaisuutta pilkkoisi pienemmiksi osiksi. Ohjelmistorobotin toimintaa on kuvailtu mahdollisimman selvästi kaaviossa 2.

Kaavio 2. IBM:n apia käyttävän ohjelmistorobotin toiminnan kuvaus



Excelin luonti

Ohjelmistorobotti luo Excel tiedoston, johon listataan tekstistä poimitut avainsanat, tiedoston nimi, tiedoston polku sekä status siitä onko tiedosto käyty läpi. Statuksen tarkistus on erittäin tärkeätä, sillä virhetilanteissa robotti tietää välittömästi mitkä dokumenteista on tarkastettu ja näin ollen aikaa säästyy, ja mahdollisesti myös rahaa, sillä koko prosessia ei tarvitse aloittaa alusta.

Dokumenttien listaus

Ennen kuin ohjelmistorobotti voi aloitta työt se tarvitsee listan dokumenteista, jotka tulisi analysoida. Tämän vaiheen olisi voinut toteuttaa automaattisesti, mutta se olisi vaatinut enemmän aikaa. Ajan säästämisen vuoksi ohjelmistorobotille annetaan JSON –tiedoston, joka pitää sisällään JSON -listan (kuva 11). Jokainen listan alkio pitää sisällään seuraavat muuttujat: tiedoston nimi, tyyppi, sijainti ja url -osoite.

Dokumentin statuksen tarkastelu

Ohjelmistorobotti tarkastaa onko dokumentti käyty läpi jo aiemmin. Tarkistusta varten on luotu status niminen kenttä aiemmin luotuun Exceliin. Jos dokumenttia ei ole käyty läpi ohjelmistorobotti jatkaa prosessia eteenpäin kaavion 2 mukaisesti. Jos dokumentti on käyty läpi, siirrytään seuraavaan dokumenttiin listassa, tai vaihtoehtoisesti jos dokumentteja ei ole jäljellä ohjelmistorobotti sulkeutuu.

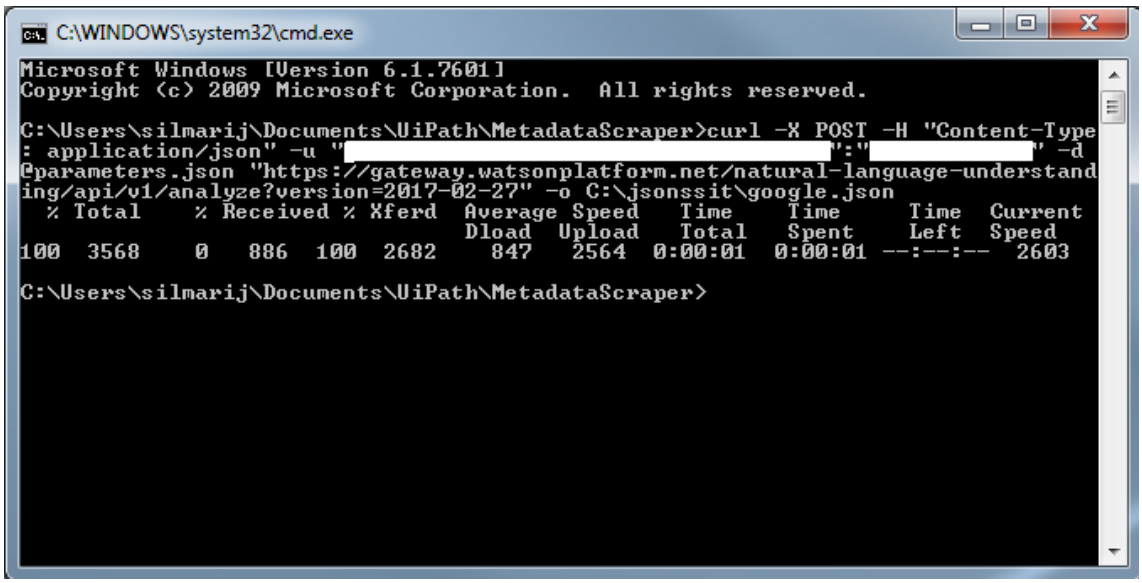
Sisällön hakeminen listatuista dokumenteista

Ohjelmistorobotti alkaa käydä JSON –listan alkioita läpi yksitellen. Ohjelmistorobotti lukee tiedostoja sijainnin ja nimen mukaisella logiikalla. Kun tiedosto on

avattu, sen sisältö kopioidaan ja teksti tallennetaan muuttujaan textToJSON. TextToJSON -muuttujasta pitää korvata lainausmerkit heittomerkeillä, sillä ne hankaloittaisivat JSON –tiedoston muodostamista huomattavasti. Merkkien korvauksen jälkeen vuorossa on uuden JSON –tiedoston luonti. Tiedoston nimeksi on annettu parameters.json ja tiedosto pitää sisällään juuri luomamme muuttujan TextToJSON sekä tiedot siitä millaista analyysia tahdomme. Ohjelmistorobotti analysoi tekstistä avainsanat ja palauttaa niitä kymmenen kappaletta.

Dokumentin sisällön lähetys

Pyyntö lähetetään cURL -komennolla komentoriviltä. Kuvassa 13 on esimerkki lähetettävästä pyynnöstä. Komento on jaettu osiin luvussa 4.4 ja sieltä voi käydä lukemassa mitä kaikkea komento pitää sisällään. Lisäksi komennon lähetyksen jälkeen saadaan kuvan mukainen vastaus, joka pitää sisällään perustietoja lähetyksestä.



Kuva 13. Esimerkki cURL pyynnöstä sekä vastauksesta

Vastauksena saadun JSON –tiedoston läpikäynti

Onnistuneen cURL pyynnön jälkeen vastauksena saadaan output.json –tiedosto. Ohjelmistorobotti lukee palautetut avainsanat output.json -tiedostosta. Tiedoston luvun jälkeen ohjelmistorobotti tekee väliaikaisen Temp -muuttujan, johon tallennetaan kaikkien avainsanojen arvo siten, että ensimmäisen avainsanan jälkeen tulee pilkku, välilyönti, ja seuraava avainsana.

Avainsanat, tiedoston nimi sekä tiedoston sijainti lisätään yksitellen Excel -tiedostoon, edellä mainittujen lisäksi myös dokumentin status päivitetään Exceliin.

Avainsanojen lisäys dokumenttiin

Kun kymmenen avainsanaa on tallennettu muuttujaan keyWords, ohjelmistorobotti avaa Word –tiedoston sille annetun JSON –listassa määritetyn osoitteen mukaisesti ja asettaa Temp muuttujan arvon tunnisteet kenttään. Tämän jälkeen Word dokumentti tallennetaan samalla nimellä ja ohjelmistorobotti siirtyy seuraavaan dokumenttiin. Dokumenttien loppuessa ohjelmistorobotti sulkeutuu.

7 Tulokset

7.1 Luonnollista tekstiä käsittelevien apien ominaisuuksien vertailu

Liitteen 1 taulukoissa 3-8 on nähtävissä palautuneet avainsanat molemmilta ohjelmointi rajapinnoilta. Tuloksista on nähtävissä se, että Microsoftin api ei välttämättä palauta avainsanoja kokonaisuudessaan ja joissakin tapauksista niistä jää osia pois. Esimerkiksi taulukosta 3 on nähtävissä, että 3D Printers palauttaa avainsanan D Printers.

Tästä voidaan päätellä hyvinkin nopeasti, että palautettuihin avainsanoihin ei voi luottaa täydellisesti, ja ehkäpä senkin vuoksi laajempikin testaus olisi voinut olla paikallaan.

Lisäksi Microsoftin api on huomattavasti rajoittuneempi ominaisuuksien suhteen sekä logiikka millä avainsanoja palautetaan, on hyvin erilainen. IBM:n api palauttaa avainsanoja numeerisessa järjestyksessä, mutta Microsoftilta tulee vain sanoja takaisin ilman mitään selitystä. [1;4]

Hinnoittelun suhteen molemmat apit ovat hyvin samankaltaisia, mutta Microsoftin ehdoton heikkous on 5000 –kirjaimen rajoitus analysoitavaksi lähetettävän tekstin mitassa. [18;19]

Omasta mielestäni IBM:n ohjelmointirajapinta on kiistattomasti parempi, ainakin suurempien tekstimäärien analysoimiseen. Microsoftin api ei kykene analysoimaan suuria tekstimääriä lainkaan, ja ominaisuuksiensa puolesta Microsoftin api on huomattavasti verrokkiaan rajoittuneempi.

7.2 UiPath sovellusten tulokset

Sovellusten kehityksen tuloksena UiPath on ohjelmana huomattavasti paremmin hallinnassa. Jälkimmäinen IBM:n rajapinnalla luotu sovellus valmistui huomattavasti nopeammin kuin Microsoftin rajapinnalla luotu sovellus. Tämä johtuu siitä, että lähes kaikki työ oli jo valmiiksi tehtynä. Ainoa ongelma johon ei ollut ratkaisua oli cURL pyynnön toteutus.

Ahkeran tiedonkeruun ja testailun jälkeen cURL pyynnön toteutus onnistui siten, että kutsut lähetetään komentoriviltä. Tämän seurauksena ohjelma on hieman Microsoftin apia hitaampi, mutta IBM:n api on huomattavasti monipuolisempi erilaisten analysointi ominaisuuksien vuoksi.

Yhteenvetona olen tyytyväinen siihen, että sain ohjelmistorobotit valmiiksi molemmilla ohjelmointirajapinnoilla.

8 Pohdintaa

Järjestelemättömän tekstin ymmärtäminen ja analysoiminen on nostanut suosiotaan viime vuosien aikana. Ratkaisuja kehitetään kovaa tahtia, mutta täydelliseen lopputulokseen ei tulla pääsemään vielä pitkään aikaan, sillä uusia ongelmia tulee aina vastaan.

Esimerkiksi slangisanastojen sekä monien eri kielten analysoinnin toteuttaminen luotettavasti tulee kestämään vielä jonkin aikaa. Lisäksi opinnäytetyön tuloksissa (taulukot 3-8) on nähtävissä ongelmia, joita palautetuissa avainsanoissa ilmenee.

Ongelmista huolimatta markkinoilta löytyy jo muutama hyvä api, enää tarvitsee keksiä miten niitä voisi hyödyntää mahdollisimman hyvin.

Mahdollisia tapoja hyödyntää apeja ovat esimerkiksi asiakaspalautteen- ja asiakaskritiikin analysointi, tiedostojen metadatan rikastamisen automatisointi sekä esimerkiksi tägien lisäys nettisivuille sivun sisällön perusteella.

9 Yleisesti opinnäytetyöstä

Opinnäytetyö oli prosessina hyvin mielekästä puuhaa aina suunnittelusta raportointiin asti. Ammatillinen osaamiseni loikkasi eteenpäin huomattavasti ja sain uusia ajatuksia mahdollisesta uravallinastani.

Ohjelmistorobotiikka vaikuttaa varsin miellyttävältä puuhalta ja se voisi olla työni koulun jälkeen. Ohjelmistorobotiikka kiehtoo lähinnä sen vuoksi että se on monipuolista työtä. Jokainen projekti on jo lähtökohtaisestikkin hieman erilainen. Ohjelmistorobottien kehityksen aikana joutuu ratkomaan erilaisia ongelmia ja loogi-nen ajattelu on suuressa roolissa ohjelmistorobotin kehityksen aikana.

Lähteet

1. Microsoft Cognitive Services. 2017. Apin esittely.
https://www.microsoft.com/cognitive-services/en-us/text-analytics-api
18.04.2017

2. Microsoft Cognitive Services. 2017. Apin dokumentaatio.
https://docs.microsoft.com/fi-fi/azure/cognitive-services/Text-Analytics/overview
18.04.2017

3. Microsoft Cognitive Services. 2017. Apin viittaukset. https://westus.dev.cognitive.microsoft.com/docs/services/TextAnalytics.V2.0/operations/56f30ceeeda5650db055a3c7
18.04.2017

4. IBM Natural Language Understanding. 2017. Apin esittely.
https://natural-language-understanding-demo.mybluemix.net/
18.04.2017

5. IBM Natural Language Understanding. 2017. Apin dokumentaatio
https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/ 18.04.2017

6. IBM Natural Language Understanding. 2017. Apin viittaukset.
https://www.ibm.com/watson/developercloud/natural-language-understanding/api/v1/
18.04.2017

7. UiPath. 2017. Community version hankinta.
https://www.uipath.com/
18.04.2017

8. Youtube. 2017. UiPathin videotutorialeihin tutustuminen.
https://www.youtube.com/user/UiPath/videos?shelf_id=0&sort=dd&view=0
18.04.2017

9. Wikipedia. 2017. Natural language understanding.
https://en.wikipedia.org/wiki/Natural_language_understanding
18.04.2017

10. Wikipedia. 2017. Curl.
https://en.wikipedia.org/wiki/CURL
18.04.2017

11. Wikipedia. 2017. HTTP.
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
18.04.2017

12. Wikipedia. 2017. JSON.
https://en.wikipedia.org/wiki/JSON
18.04.2017

13. Wikipedia. 2017. Application programming interface.
https://en.wikipedia.org/wiki/Application_programming_interface
18.04.2017

14. Curl. 2017. Curliin tutustuminen.
https://curl.haxx.se/
18.04.2017

15. Curl. 2017. Latauslinkki curliin.
https://curl.haxx.se/download.html
18.04.2017

16. Curl. 2017. Curlin manuaali.
https://curl.haxx.se/docs/manpage.html
18.04.2017

17. CGI. 2017. CGI:n tarina.
https://www.cgi.fi/historia-suomessa
16.05.2017

18. Microsoft Cognitive Services. 2017. Apin hinnoittelu.
https://azure.microsoft.com/en-us/pricing/details/cognitive-services/text-analytics/
16.05.2017

19. IBM. 2017. Apin hinnoittelu.
https://www.ibm.com/watson/developercloud/natural-language-understanding.html
16.05.2017

20. UiPath. 2017. UiPathin esittely.
https://www.uipath.com/guides/introduction
17.05.2017

Taulukko 3. Ensimmäisestä dokumentista analysoidut avainsanat

KeyWords - Microsoft	KeyWords - IBM
D printers	tactile graphics
D objects	3d printing
D tactile graphics	Push/Pull tool
D models	Click Add File
D Printing Instructions	3D tactile graphics
D scanning	3D object files
Select tool	replicable tactile graphics
D object files	MakerBot 3D printer
Select File	click download
Rectangle tool	Open SketchUp

Taulukko 4. Toisesta dokumentista analysoidut avainsanat

KeyWords - Microsoft	KeyWords - IBM
Google Search	Google
garage	search engine Backrub
Google story	World Wide Web
California	Google story
Sergey Brin	ping pong table
relentless search	Sun co-founder Andy
company dogs	suburban Menlo Park
search engine Backrub	Clunky desktop computers
incorporated team	Silicon Valley investors
Larry Page	newly incorporated team

Taulukko 5. Kolmannesta dokumentista analysoidut avainsanat

KeyWords - Microsoft	KeyWords - IBM
input text	sentiment analysis
text data	key phrase extraction
unstructured text	text analytics
piece of text	main talking points
Text Analytics API	input text
Text Analytics Documentation	text analytics services
key phrase extraction of English text	Text Analytics API
human written text	Text Analytics service
Azure ML Text Analytics service	increasingly popular field
suite of text analytics services	positive sentiment

Taulukko 6. Neljännestä dokumentista analysoidut avainsanat

KeyWords - Microsoft	KeyWords - IBM
NHL history	NHL All-Star Games
national star	Teemu Ilmari Selänne
Finnish National Team	Greatest NHL Players
National Hockey League	NHL Entry Draft
Anaheim Ducks	highest scoring Finn
Star Weekend	Anaheim Ducks
Star Teams	Finnish Flash
goals	numerous team scoring
numerous team scoring records	ice hockey winger
highest scoring Finn	National Hockey League

Taulukko 7. Viidennestä dokumentista analysoidut avainsanat

KeyWords - Microsoft	KeyWords - IBM
Twitch Creative	Twitch Partner Program
Twitch community	Twitch Plays game
Twitch Plays game	Twitch community
Twitch channel	Twitch channel
video game culture	video game
Twitch based	video game culture
video games	social video platform
major video game publisher	daily active users
video game media sites	video game purchase
game genre	new social network

Talukko 8. Kuudennesta dokumentista analysoidut avainsanat

KeyWords - Microsoft	KeyWords - IBM
Degree Students	Karelia University
International Students	Applied Sciences
Incoming exchange students	training skilled professionals
Outgoing exchange students	Bright World
oriented business students	Outgoing exchange students/year
Karelia University of Applied Sciences degree	Incoming exchange students/year
Student satisfaction rate	Student satisfaction rate
great future	Applied Sciences degree
educational institutions	great future
development projects	solid grounding