

# Analiza i usporedba aplikacijskih programskih sučelja za prirodni jezik

---

Grobenski, Emilio

Undergraduate thesis / Završni rad

2019

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:211:371194>

*Rights / Prava:* [Attribution-NonCommercial-NoDerivs 3.0 Unported](#)

*Download date / Datum preuzimanja:* **2020-11-29**



*Repository / Repozitorij:*

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Emilio Grobenski**

**ANALIZA I USPOREDBA APLIKACIJSKIH  
PROGRAMSKIH SUČELJA ZA PRIRODNI  
JEZIK**

**ZAVRŠNI RAD**

**Varaždin, 2019.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Emilio Grobensi**

**Matični broj: 44862/16–R**

**Studij: Informacijski sustavi**

**ANALIZA I USPOREDBA APLIKACIJSKIH PROGRAMSKIH**  
**SUČELJA ZA PRIRODNI JEZIK**

**ZAVRŠNI RAD**

**Mentor:**

Prof. dr. sc. Kliček Božidar

**Varaždin, rujan 2019.**

*Emilio Grobenski*

### **Izjava o izvornosti**

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi*

---

## **Sažetak**

Rad se bavi obradom, usporedbom i analizom različitih aplikacijskih programskih sučelja za implementaciju umjetne inteligencije za prirodni jezik. Prikazana je izrada jednostavne aplikacije i korištenje određenog aplikacijskog sučelja pomoću slanja HTTP zahtjeva. Naposljetku su na teškoj zadaći analizirana aplikacijska programska sučelja iz iste domene te izveden zaključak.

**Ključne riječi:** umjetna inteligencija, prirodni jezik, aplikacijska programska sučelja, aplikacija

# Sadržaj

1. Uvod .....	1
2. Metode i tehnike rada .....	2
3. Umjetna inteligencija.....	3
3.1. Obrada prirodnog jezika.....	3
4. Aplikacijka programska sučelja .....	5
4.1. IBM Watson API .....	5
4.2. Google Cloud AI .....	10
4.2.1. Natural Language API .....	10
4.2.2. Translation API.....	11
4.2.3. Text to Speech API i Speech to Text API .....	12
4.3. Microsoft Azure Cognitive Services APIs.....	12
4.3.1. Speech Services .....	13
4.3.2. Speaker Recognition API .....	13
4.3.3. Bing Speech.....	14
4.3.4. Translator Speech API .....	14
4.3.5. Immersive Reader .....	15
4.3.6. Language Understanding .....	15
4.3.7. QnA Maker.....	16
4.3.8. Text Analytics API .....	17
4.3.9. Translator Text.....	18
5. Aplikacija Prevoditelj.....	19
5.1. Programski kod aplikacije.....	20
6. Analiza i usporedba API-a .....	22
7. Zaključak .....	25

Popis literature .....	26
Popis slika .....	27

# 1. Uvod

Ljudi međusobno komuniciraju verbalno i neverbalno, verbalnu komunikaciju ostvaruju riječima na nekom jeziku. Taj jezik nazivamo prirodnim jezikom. Međutim, prirodni jezik ne razumiju računala. Računala funkcioniraju pomoću strojnog jezika. Kako bi omogućili računalima da razumiju prirodni jezik koristimo umjetnu inteligenciju. Zašto nam je važno da računala razumiju prirodni jezik? Zato što računala tada mogu obrađivati podatke na prirodnom jeziku. Ova tehnologija ima široku primjenu u svakodnevnom životu ljudi. Primjerice, umjetna inteligencija za obradu prirodnog jezika uz pomoć strojnog učenja koriste računalni prevoditelji koji prevode tekst ili govor iz jednog jezika u drugi i obratno, aplikacije za prepoznavanje govora tj. upravljanje govorom na mobitelu ili u automobilu. Također, primjenjuje se i za prepoznavanje semantičkih pogrešaka, klasifikaciju pojmova u tekstu, razumijevanje osjećaja teksta i tako dalje što će biti detaljnije obrađeno u nastavku. Za razvoj takvih funkcionalnosti u aplikacijama koristimo aplikacijska programska sučelja za prirodni jezik kojih ima nekolicina. U nastavku će biti obrađena umjetna inteligencija i strojno učenje za lakše razumijevanje funkcioniranja programskih sučelja koja iste tehnologije koriste. Nadalje, biti će obrađena i međusobno uspoređena najpopularnija aplikacijska programska sučelja za prirodni jezik vodećih kompanija na tom području. U nastavku, prikazana će biti implementacija aplikacijskog programskog sučelja i izvedeni testovi kako bi uvidjeli prednosti i nedostatke pojedinih aplikacijskih programskih sučelja za prevođenje te izveden objektivan zaključak.



## 2. Metode i tehnike rada

Tehnologije koje će biti korištene za izradu rada:

- Visual studio 2019 – okruženje za pisanje koda
- Translator Text – aplikacijsko programsko sučelje
- Language Translator demo – demo verzija aplikacije koja koristi Language Translator API
- Translation API demo - demo verzija aplikacije koja koristi Translation API

## 3. Umjetna inteligencija

Umjetna inteligencija je tehnologija koja neživim bićima pridaje ljudsku inteligenciju. Do danas je razvijena tako što ju prepoznamo u računalima. Svojstva umjetne inteligencije su: zaključivanje, razumijevanje prirodnog jezika, učenje, prepoznavanje predmeta itd. Ona se dijeli na slabu i jaku inteligenciju. Pod slabu inteligenciju se smatra da je računalo sposobno odrađivati zadatke u nekoj odvojenoj domeni, npr. samo prepoznavanje predmeta, a jaka inteligencija podrazumijeva sustav koji ima sposobnosti rješavati različite zadatke iz različitih domena bez prisustva čovjeka što znači da će praktični primjer u sklopu ovog završnog rada predstavljati slabu inteligenciju.

### 3.1. Obrada prirodnog jezika

Tehnologija umjetne inteligencije s kojom se postiže razumijevanje prirodnog jezika (eng *Natural language processing* - NLP), kraće NLP, ima svojstva čitanja, razumijevanja i pisanja ljudskog jezika. Računala koja koriste NLP imaju sposobnost komunicirati međusobno ili sa ljudima. Primjer za komunikaciju dva računala putem prirodnog jezika bio bi putem aplikacije Shazam gdje računalo osluškuje prirodni jezik i ako prepozna pjesmu ispisuje naslov te pjesme. Komunikaciju između čovjeka i računala možemo prepoznati u glasovnim naredbama u automobilu ili na pametnom telefonu



Slika 1: Shazam logo (Google Play, nema dat.)

NLP se trenutno koristi u razne svrhe poput izrade tzv. Chatbot-ova, računalnih prevoditelja, programa za ispravljanje gramatičkih i sintaktičkih pogrešaka, programa za filtriranje neželjene pošte itd. Chatbot-ovi u aplikacijama služe kako bi poboljšali korisničko iskustvo tako što pomoću NLP-a automatizirano odgovaraju na pitanja korisnika. Ukoliko aplikacija ima ugrađenu funkcionalnost pretraživanja, pomoću NLP-a mogu se ugraditi

dodatne funkcionalnosti u svrhu poboljšanja korisničkog iskustva. Tako možemo ugraditi sposobnost aplikacije da prilikom unosa svakog slova korisnika ponudi rezultate koje korisnik želi pretražiti. Također možemo ugraditi da aplikacija ponudi rješenje ako prepozna da je korisnik unio neispravan pojam. Kao dobar primjer može nam poslužiti Google pretraživanje koje ima obje funkcionalnosti.

Zbog globalizacije svijeta, kroz godine se javila sve veća potreba za međuljudskom komunikacijom za koju su nerijetko bile prepreke različitosti, poput jezika i kulture. S napretkom tehnologije, uz razvoj umjetne inteligencije razvilo se i strojno prevođenje koje je osmišljeno tako da program prevodi tekst ili glas iz jednog jezika u drugi bez prisustva čovjeka. Ilya Pestov iznosi na web stranici freeCodeCamp kako je sve započelo davne 1933. kada je sovjetski znanstvenik Peter Troyanskii prezentirao na moskovskom Academy of Science stroj za selekciju i ispis riječi kada prevodi s jednog jezika u drugi. Do danas, raznim istraživanjima i razvojem, tehnologija strojnog prevođenja je napredovala do te mjere da određeni programi posjeduju veliku točnost za vodeće svjetske jezike. Nadalje, postavlja se pitanje opstanka prevođenja u budućnosti kao ljudskog posla. Na to pitanje, znanstvenica Zrinka Šimunić u članku „Ljudsko prevođenje vs strojno prevođenje – iz perspektive prevoditelja“ odgovara:

Stručnjaci će ih pokušati razuvjeriti tvrdnjama da je čovjek ključni i nezamjenjiv akter u vrlo kompleksnom procesu prevođenja te da strojno prevođenje ne predstavlja prijetnju budućnosti prevoditeljskog posla jer nema prevođenja bez čovjeka, pa makar on bio sveden na ulogu post-editora strojnih prijevoda. (Šimunić, 2017.)

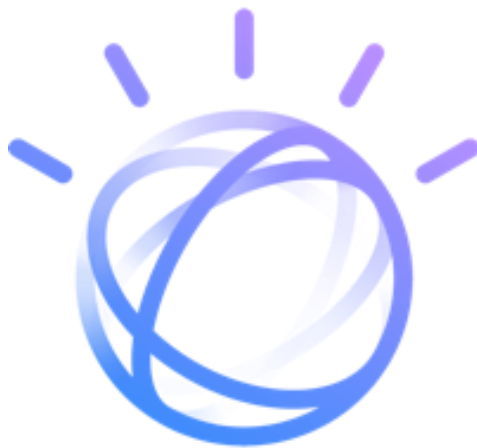
U nastavku će biti prikazani primjeri prijevoda na hrvatski jezik pomoću aplikacijskih programskih sučelja vodećih svjetskih kompanija koje koriste svoje standardne modele podataka, ali važno je napomenuti da se prijevodi mogu poboljšavati ukoliko napravimo specifičan model podataka za naše potrebe. Razvojem strojnog prevođenja razvijene su različite tehnologije prevođenja poput strojnog prevođenja baziranog na pravilima (eng. Rule-Based Machine Translation - RBMT), strojnog učenja baziranog na primjerima (eng. Example-Based Machine Translation - EBMT), statističkog strojnog prevođenja (eng. statistical machine translation - SMT) i najnovije živčano strojno prevođenje (eng. neural machine translation - NMT). RBMT je prvi primjer u praksi strojnog prevođenja koji koristi lingvističke informacije jezika iz kojeg se prevodi i jezika u koji se prevodi uz pomoć gramatike i rječnika. EBMT funkcionira pomoću pronalaska primjera parova za prevođenje između jezika prevođenja. Postoji mogućnost visokokvalitetnih prijevoda ukoliko su slični pronađeni. Isto tako, ukoliko primjeri nisu pronađeni, kvaliteta prijevoda može biti vrlo niska. SMT je dosegao vrhunac u performansama prevođenja, pa je napravljena nova tehnologija bazirana na živčanoj mreži (eng. Neural Network - NN) koja raznim algoritmima i matematičkim modelima pokušava simulirati živčani sustav živih bića. NMT ne pruža samo bolje performanse u kvaliteti

prevođenja, već prijevodi zvuče mnogo čišće, tečnije, više nalik na ljude, nego SMT. Razlog tomu jest što NMT uzima cijeli kontekst rečenice koju prevodi dok SMT uzima samo kontekst nekoliko riječi prije i poslije riječi koju prevodi. Prevođenja govora u govor koriste NMT dok prevođenja iz govora u tekst koriste kombinaciju SMT i NMT tehnologija. Međutim, ako je za odabrani jezik podržana NMT tehnologija u potpunosti onda se koristi samo NMT. Kada nije podržana NMT tehnologija u potpunosti, koristi se engleski jezik između dva jezika koja se prevodi što znači da se iz prvog jezika prevodi u engleski, pa zatim iz engleskog u drugi jezik pomoću NMT i SMT tehnologija.

## 4. Aplikacijska programska sučelja za prirodni jezik

Kako bismo mogli izraditi, do sad, navedene funkcionalnosti koristimo se aplikacijskim programskim sučeljima (eng. *Application programming interface* - API), u nastavku API. API je skup podprograma, komunikacijskih protokola i alata za razvoj softvera. (Application programming interface, 2019.) U suštini, svrha API-a jest u tome da neka tehnologija bude lakše implementirana u više programa, tj. ako ne bismo imali izrađene API-e za određenu tehnologiju svaki put kad bi ju netko želio implementirati morao bi ju zasebno razviti. Dakle, API nam olakšava i ubrzava proces razvoja programa, pa su stoga tvrtke došle na ideju naplaćivanja API-a. Naime, mnogim programerima ili tvrtkama se isplati platiti određenu cijenu za API/API-e kako ne bi morali razvijati svoj, ako određeni API zadovoljava njihove potrebe. Budući da je tržište dosta razvijeno na tom području postoji i mnoštvo besplatnih API-a ovisno o tehnologiji o kojoj je riječ. U nastavku će biti obrađeni trenutno najpopularniji API-i za NLP gdje će se uzimati u obzir mogućnosti koje određeni API nudi, cijena, dokumentacija i opće iskustvo.

### 4.1. IBM Watson API



Slika 2: IBM Watson logo (IBM, nema dat.)

IBM Watson API-u pristupamo preko besplatnog računa koji smo otvorili na IBM Cloud-u. IBM Cloud je platforma s različitim alatima za razvoj koju je razvila tvrtka IBM. Nadogradnjom računa otvaraju se dodatne mogućnosti poput većeg memorijskog prostora na IBM-ovom

serveru i veći izbor alata. U daljnjem opisu fokus će biti prebačen na IBM Watson API koji je samo dio platforme IBM Cloud.

Samo web sučelje preko kojeg pristupamo funkcionalnostima koje nudi IBM Watson je vrlo intuitivno. IBM Watson je podijeljen na API-e za određene funkcionalnosti u NLP-u te je svakom API-u pridodan prigodan naziv kako bi lakše odabrali onaj koji nam je potreban. Svaki od njih sadrži detaljno dokumentaciju i pojmovnik s pojašnjenjima. API-e koje možemo koristiti s besplatnim računom su slijedeći: Watson Assistant, Discovery, Visual Recognition, Natural Language Understanding, Speech to Text, Text to Speech, Natural Language Classifier, Personality Insights, Tone Analyzer, Language Translator, Watson Studio, Knowledge Studio, Machine Learning, Knowledge Catalog i Compare and Comply.

Watson Assistant omogućava izradu chatbot-ova i virtualnih agenata za konverzaciju. Discovery omogućava aplikacijama analizu sadržaja, prepoznavanje obrazaca i detaljan uvid za donošenje boljih odluka. Također koristi upite kako ne bismo morali podatke ručno filtrirati, grafove za jasniji prikaz i potpunu enkripciju (eng. *end to end encryption*). NLP je integriran u API kako bi se lakše koristili njime i izbjegli kompleksnost NLP-a. Nadalje, u dokumentaciji se nalaze informacije o autentifikaciji i pristupu API-u, upravljanju pogreškama, upravljanju podacima, kreiranju i brisanju sjednice itd. Primjeri koda su podržani za programske jezike Curl, C#, Go, Java, Node.js, Python, Ruby i Swift.

Visual Recognition omogućava prepoznavanje sadržaja slike. Ima ugrađen model po kojem prepoznaje objekte, ali taj model možemo mijenjati i prilagoditi potrebama. Ima ugrađen specijaliziran rječnik sa preko 2000 riječi za prepoznavanje hrane i obroka. Može prepoznati lice na fotografiji, odrediti starost i spol osobe. U dokumentaciji se nalaze objašnjenja kako kreirati prilagođen model svojim potrebama, kako izraditi aplikaciju korištenjem ovog API-a, istaknute dobre prakse korištenja, kako spriječiti gubljenje podataka u slučaju nesreće poput nestanka struje u području servera ili koruptiranog tvrdog diska itd.

Natural Language Understanding služi za analizu i izdvajanje metapodataka iz sadržaja. Može analizirati ulazni tekst, kategorije, koncepte, emocije, objekte, ključne riječi, metapodatke i uloge kako bi postigli bolju optimizaciju oglašavanja, prilagodili određenim korisnicima, analizirali glas korisnicima ili rudarili podatke (eng. *data mining*). Također, mogu se kreirati vlastiti modeli za prepoznavanje vlastitih objekata. Trenutno se može koristiti za prepoznavanje engleskog, arapskog, kineskog, nizozemskog, francuskog, njemačkog, talijanskog, japanskog, korejskog, portugalskog, ruskog, španjolskog i švedskog jezika. Za testiranje API-a možemo koristiti odmah dostupan demo koji, za dani tekst, analizira osjećaje, pozitivan ili negativan sadržaj, ključne riječi, entitete, kategorije, koncepte, sintaksu i semantiku. Nadalje, objašnjeno je kako analizirati web stranice, prilagođavati model

prepoznavanja entiteta, veza i kategorija, prepoznati prevladavajući jezik u tekstu itd. Postoje klijentske biblioteke za Android, Java, Go, Node.js, Python, Ruby, C#, Salesforce, Swift i Unity.

Speech to Text pretvara ljudski glas u riječi. Podržava Mobile SDKs koji omogućava prirodnu interakciju na IOS i Android uređajima. Također postoji demo verzija aplikacije koja koristi ovaj API, a omogućuje snimanje i prenošenje audio zapisa za prevođenje u tekst. Trenutno se može koristiti za arapski, portugalski, kineski, engleski, francuski, njemački, japanski, korejski i španjolski jezik. API-u je moguće pristupiti putem HTTP zahtjeva i klijentskih biblioteka. Klijentske biblioteke su dostupne za Android, Java, Go, Node.js, Python, Ruby, C#, Salesforce, Swift i Unity. Podržani formati audio zapisa su: mpeg, alaw, flac,, g729, l16, mp3, mulaw, ogg, wav i webm.

Suprotno, Text to Speech riječi čita, tj. pretvara u zvuk. Dostupan je za engleski, francuski, njemački, talijanski, španjolski, portugalski i japanski jezik, te dodatne mogućnosti poput: TTS customization i Mobile SDKs. TTS customization omogućuje kreiranje vlastitog rječnika s vlastitim riječima što je i objašnjeno na primjerima. TTS customization i Mobile SDKs su trenutno u beta verziji. Postoji i demo verzija koja čita uneseni tekst, tj. prevodi ga u audio zapis. Ovisno o jeziku podržano je više glasova. Podržani audio formati su: flac, l16, mp3, mpeg, mulaw, ogg, wav i webm. Za prevođenje teksta u audio zapis API podržava dva standarda prevođenja International Phonetic Alphabet (IPA) i Symbolic Phonetic Representation (SPR) te je objašnjeno kako ih koristiti

Natural Language Classifier koristi napredan NLP i strojno učenje za raspoređivanje riječi unesenog teksta u određene kategorije. Na primjer, za uneseno pitanje vraća ključne riječi koje najbolje odgovaraju odgovoru. Postoji demo verzija aplikacije koja koristi ovaj API kako bi si korisnici lakše predočili koja je svrha API-a. API podržava i slanje do trideset izreka za klasificiranje od jednom. Podržani jezici su engleski, arapski, francuski, njemački, talijanski, japanski, korejski, portugalski i španjolski. Za implementaciju API-a pomoću klijentskih biblioteka možemo koristiti u Android, Java, Go, Node.js, Python, Ruby, C#, Salesforce, Swift i Unity okruženjima ili i u drugim okruženjima koja podržavaju slanje HTTP zahtjeva .

Personality Insights može se koristiti za analizu podataka sa društvenih mreža kako bi se odredile psihološke osobine korisnika i poboljšala konverzacija. Ovaj API koristi analizu pisanja kako bi se odredila osobnost korisnika preko digitalne komunikacije poput komunikacije putem elektroničke pošte, poruka na socijalnim mrežama, tweet-ova i objava na forumu. Pomoću tih podataka možemo odrediti koje su preferencije određenog korisnika kako bi mu ponudili određene proizvode, usluge ili aktivnosti. API možemo implementirati pomoću slanja HTTP zahtjeva ili klijentskih biblioteka podržanih u Android, Java, Go, Node.js, Python, Ruby, C#, Salesforce, Swift i Unity okruženjima. Formatu teksta koje podržava API su običan,

jednostavan tekst (eng. *plain text*), HTML, JSON i CSV. Podržani jezici su arapski, engleski, japanski, korejski, španjolski, portugalski, francuski, njemački, talijanski i kineski.

Pomoću Tone Analyzer-a možemo implementirati razumijevanje emocija i stila pisanja iz teksta. Ovaj API se može koristiti kako bi dobili informaciju kakav je naš stil pisanja u nekom dokumentu te ga poboljšali, ovisno o tome što s određenim stilom pisanja želimo postići. Također, možemo ga koristiti kako bi saznali kojim stilom pišu naši korisnici i prilagodili se njima ili unaprijedili njihov način izražavanja. API može analizirati običan tekst, JSON ili HTML format te IBM navodi da podržava 128KB sadržaja što je otprilike 1000 rečenica. Situacije u kojima se može koristiti su: osluškivanje društvenih mreža, na primjer, što korisnici kažu o našem proizvodu ili brendu, korištenje personaliziranog marketinga poput slanja opuštajućih poruka ljutitim korisnicima, u chatbot-ovima itd. I ovaj API može se koristiti pomoću HTTP zahtjeva kao i pomoću klijentskih biblioteka. Postoje dva demo primjera aplikacija kako možemo koristiti ovaj API. API je trenutno dostupan samo za engleski i francuski.

Language Translator služi za prevođenje prirodnih jezika. Pomoću njega možemo kreirati vlastite modele za prevođenje. Trenutno podržava arapski, bugarski, katalonski, kineski, hrvatski, češki, danski, nizozemski, engleski, estonski, finski, francuski, njemački, grčki, hebrejski, indijski, mađarski, talijanski, japanski, korejski, norveški, poljski, portugalski, ruski, slovački, slovenski, španjolski, švedski i turski između kojih može prevoditi odmah cijele dokumente, aplikacije i web stranice. Također posjeduje i tzv. Language Identification, Neural Machine Translation, Document Translator i Corpus Customization. Language Identification omogućuje prepoznavanje do 68 jezika. Neural Machine Translation je nova tehnologija koja omogućava veću preciznost prevođenja i veću brzinu pomoću dubokog učenja (eng. *deep learning*). Document Translator je tehnologija u beta verziji koja nudi mogućnost prevođenja dokumenata s očuvanjem tipa i formata dokumenta. Trenutno podržani formati su : MS Office, Open Office, PDF, HTML, JSON, TXT i XML. Corpus Customization je funkcionalnost koja je dostupna samo za nadograđene članove, pa ona neće biti obrađena. Postoji korisna demo verzija aplikacije koja koristi ovaj API koja će biti korištena za testiranje složenijih zadataka u nekim jezičnim domenama.

Watson Studio nudi set alata i okruženje koje koristi strojno učenje i duboko učenje kako bi se ubrzao proces uključivanja umjetne inteligencije u poslovnom svijetu. Može se koristiti uz programske jezike Python, R i Scala. Budući da je okruženje u oblaku na projekt se mogu dodati i ostali sudionici što se dodatno naplaćuje. Postoji mnoštvo korisnih alata za analizu i prikaz podataka.

Knowledge Studio je jezik sa prilagođenim modelima strojnog učenja koji identificiraju objekte i veze u nestrukturiranom tekstu. Omogućava razvoj vlastitih modela na intuitivan način



bez pisanja koda. Kreirani modeli se mogu koristiti u Discovery-u, Watson Natural Language Understanding-u i Watson Explorer-u. Pomoću Knowledge Studio možemo kreirati model pravila prema kojima će se tražiti entiteti, značenja i veze u našim dokumentima. U dokumentaciji je prikazano kako sastaviti tim, kreirati radno okruženje, kreirati model pravila, model strojnog učenja i model kategorija prilagođenih za naše potrebe.

Machine Learning usluga je skup API-a koji se mogu koristiti u bilo kojem programskom jeziku za razvoj aplikacija koje će donositi pametnije odluke, rješavati probleme i unaprijediti korisničko iskustvo. Okviri koje možemo koristiti za strojno učenje su: TensorFlow, Keras, Caffe, PyTorch, Spark MLlib, scikit learn, xgboost i SPSS.

Knowledge Catalog pojednostavljuje upravljanje i kontrolu nad prikupljenim podacima te ih povezuje s ljudima kojima su potrebni. Omogućava nam pristup podacima temeljen na poslovnim pravilima u poduzeću.

Compare and Comply je skup naprednih API-a koji omogućuju brže i bolje razumijevanja dokumenta. To su API-i za konverziju PDF, TIFF, JPEG i Word dokumenata u HTML, razumijevanje tablica, NLP i usporedbu ugovora. Kako bi se poboljšalo iskustvo korištenja svaki korisnik može dodavati povratne informacije. Mogućnosti koje se još mogu koristiti su: identifikacija strana i obveza u ugovorima, automatsko označavanje rečenica u ugovoru prema kategorijama poput trajanje ugovora, privatne informacije, plaćanja itd., usporedba API-a za analizu dokumenata te isticanje razlika u analizama itd.

Naposlijetku možemo zaključiti da IBM Watson nudi mnoštvo API-a za implementaciju NLP-a sa popratnom detaljnom dokumentacijom. Za pomoć pri korištenju API-a postoji blog s člancima, kanal na YouTube-u, Watson webinar s obrađenim mnogim temama i primjeri koda na GitHub-u što dodatno olakšava razvoj i implementaciju funkcionalnosti umjetne inteligencije za prirodni jezik. Također podržana je i podrška korisnicima kao i forma za slanje povratnih informacija.

## 4.2. Google Cloud AI



Slika 3: Google Cloud logo (Google Cloud, nema dat.)

### 4.2.1. Natural Language API

Natural Language API-u pristupamo preko računa na Google Cloud Platform aplikaciji koja ima besplatan probni period od godinu dana. Ovaj API omogućuje programerima korištenje funkcionalnosti umjetne inteligencije za prirodni jezik. Neke od funkcionalnosti su: analiza teksta, analiza objekata, klasifikacija sadržaja itd. Postoji i jednostavna demo aplikacija koja koristi Natural Language API da bismo isprobali funkcionalnost API-a. Aplikacija radi tako da uneseni tekst klasificira pojmove u određene skupine poput lokacije, organizacije, osobe, cijene, događaja itd.

Dokumentacija je vrlo detaljna popraćena vrlo korisnim primjerima. Objašnjeno je kako početi sa radom i kako koristiti klijentske biblioteke. Primjeri su navedeni za početak rada na tri operacijska sustava: Windows, Linux i macOS. Google Cloud Natural Language API možemo koristiti pomoću Google Cloud Client Libraries ili Google Cloud SDK. Google Cloud Client Libraries je skup biblioteka koje možemo koristiti u razvoju programa. Google Cloud SDK je skup alata kojima pristupamo preko komandne linije. Za oba načina korištenja Natural Language API-a objašnjeno je kako ih pripremiti za rad. Funkcionalnosti su podijeljene na analizu osjećaja (eng. *Analyzing sentiment*), analizu entiteta (eng. *Analyzing entities*), analizu sintakse (eng. *Analyzing syntax*), analizu osjećaja entiteta (eng. *Analyzing entity sentiment*), klasifikaciju sadržaja (eng. *Classifying content*).

Analiza osjećaja, za dani tekst, određuje osjećaje koji prevladavaju u tekstu kako bi odredili da li je stav autora pozitivan, negativan ili neutralan. Ova funkcionalnost je podržana

za kineski, engleski, francuski, njemački, talijanski, japanski, korejski, portugalski i španjolski jezik. Dodatna mogućnost je čitanje cijelog dokumenta bez direktnog slanja sadržaja dokumenta ako je prethodno pohranjen na Google Cloud Storage. Uz navedeno, postoji i tutorijal gdje je implementirana funkcionalnost u Python programskom jeziku.

Analiza entiteta u danom tekstu prepoznaje određene objekte (osobe, lokacije, organizacije, događaji itd.) i vraća informacije o njima. Za ovu funkcionalnost podržani su isti jezici kao i za analizu osjećaja uz dodatak ruskog jezika. Kao i kod analize osjećaja, pomoću Google Cloud Storage može se čitati sadržaj cijelog dokumenta.

Analiza sintakse dijeli dani tekst na riječi i rečenice te vraća lingvističke informacije o njima dok ostali API-i za prirodni jezik obično analiziraju sadržaj teksta kako navodi Google. Podržani su isti jezici kao i kod analize entiteta te čitanje cijelog dokumenta prenesenog na Google Cloud Storage.

Analiza osjećaja entiteta je spoj analize entiteta i analize osjećaja koji pokušava odrediti da li je neki entitet u tekstu pozitivan ili negativan. Za svaki pojedinačni entitet u tekstu vraća neki rezultat koji se može interpretirati. U slučaju neutralnog sadržaja interpretira se kao nisko emotivan sadržaj ili miješano emotivan zato što se pozitivan i negativan sadržaj mogu međusobno poništiti. Trenutno su podržani engleski, japanski i španjolski jezici uz čitanje cijelog dokumenta pomoću Google Cloud Storage.

Klasifikacija sadržaja analizira dani dokument i vraća listu kategorija koje odgovaraju tom dokumentu. Ova funkcionalnost je dostupna samo na engleskom jeziku. Također je omogućeno čitanje cijelog dokumenta uz pomoć Google Cloud Storage. Kao i kod analize osjećaja postoji tutorijal za implementaciju u Pythonu.

## **4.2.2. Translation API**

Translation API je API koji služi za prevođenje teksta iz jednog jezika u drugi. Također, postoji mogućnost kreiranja prilagođenog rječnika našim potrebama. API možemo koristiti pomoću slanja HTTP zahtjeva ili korištenjem klijentskih biblioteka. Za oba načina korištenja detaljno je objašnjeno kako pripremiti okruženje za rad te je potkrepljeno primjerima. Programski jezici koji podržavaju klijentske aplikacije za ovaj API su C#, Go, Java, Node.js, PHP, Python i Ruby. Nadalje, na primjeru je objašnjeno kako možemo integrirati u našu web stranice prevođenje sadržaja pomoću ovog API-a. Prevođenje je podržano za preko sto svjetskih jezika što je dosta više od IBM-ovog Language translator API-a. Nadalje, u dokumentaciji se nalaze pojašnjenja kako prevoditi tekst koristeći API, prevođenje teksta korištenjem živčanog strojnog prevođenja (eng. *neural machine translation* - NMT), u daljnjem tekstu NMT, kako istražiti podržane jezike, prepoznati jezik u stringu, koristiti rječnik, prevoditi

uz pomoć slanja serija zahtjeva kod prevođenja većih količina teksta. Također, postoji i korisna demo verzija aplikacije koja koristi Google-ov Translation API kako bismo mogli isprobati prevođenja između podržanih jezika što će nam biti korisno kod usporedbe kvaliteta prevođenja između API-a ostalih kompanija.

### 4.2.3. Text to Speech API i Speech to Text API

Text to Speech API omogućuje funkcionalnost prevođenja teksta u audio zapis, tj. ljudski govor. U svoje aplikacije ga možemo implementirati pomoću klijentskih biblioteka ili slanjem HTTP zahtjeva. Za oba načina potrebno je provesti autentifikaciju što je detaljno objašnjeno u dokumentaciji. Nadalje, objašnjene su i druge mogućnosti poput korištenja različitih profila za reprodukciju zvuka, na primjer, podešavanje kada želimo da se zvuk reproducira preko manjeg zvučnika i slično. Zatim određivanje formata audio zapisa, ispis podržanih glasova za određeni jezik i dekodiranje i enkodiranje audio sadržaja.

Speech to Text API služi za prevođenje audio zapisa u tekstualni. Kao i prethodni, možemo ga implementirati pomoću klijentskih biblioteka ili slanjem HTTP zahtjeva. U dokumentaciji se nalaze objašnjenja za prevođenje audio zapisa s kraćim i dužim vremenskim trajanjem te prevođenje strujanja audio zapisa. Mnoge funkcionalnosti su još u beta verziji poput unaprjeđenja točnosti, dijeljenje sadržaja prema govorniku iz jednog audio zapisa, automatsko prepoznavanje jezika itd.

## 4.3. Microsoft Azure Cognitive Services APIs



Slika 4: Microsoft logo (Microsoft, nema dat.)

Azure Cognitive Services je skup API-a, alata i usluga za pomoć programerima u razvoju aplikacija sa umjetnom inteligencijom. API-i su podijeljeni u slijedeće kategorije: vid (eng. *Vision*), govor (eng. *Speech*), jezik (eng. *Language*), Internet pretraživanje (eng. *Web*

*Search*) i odluka (eng. *Decision*). Naime, biti će obrađeni API-i i samo iz područja govora i jezika jer ostala ne spadaju pod umjetnu inteligenciju za prirodni jezik. U kategoriji govora nalaze se četiri API-a: Speech Services, Speaker Recognition, Bing Speech API, Translator Speech. U kategoriji jezika nalaze se slijedeći API-i: Immersive Reader, Language Understanding, QnA Maker, Text Analytics i Translator Text.

### **4.3.1. Speech Services**

Speech Services podržava prevođenje govora u tekst i obratno i prevođenje govora u govor. Prevođenje govora u tekst podržano je za C#, JavaScript, Python, Linux, Java, C++, Objective-C i Swift programski jezik. Prevođenje teksta u govor podržavaju C# i C++. Prevođenje govora u govor trenutno se može koristiti u Java, C# i C++ programskim jezicima. Jezici koje podržava prevođenje govora u tekst su: arapski, katalonski, danski, njemački, engleski, španjolski, finski, francuski, indijski, talijanski, japanski, korejski, norveški, nizozemski, poljski, portugalski, ruski, švedski, kineski i tajlandski, a među njima nije omogućeno prilagođavanje za katalonski, danski, norveški, poljski, švedski i tajlandski jezik. Prevođenje iz teksta u govor podržavano je na slijedećim jezicima: njemački, engleski, talijanski, kineski, arapski, bugarski, katalonski, češki, danski, grčki, španjolski, finski, francuski, hebrejski, indijski, hrvatski, mađarski, indonezijski, japanski, korejski, malajski, norveški, nizozemski, poljski, portugalski, rumunjski, ruski, slovački, slovenski, švedski, turski i vijetnamski. Prevođenje govora u govor je podržano između slijedećih jezika: arapski, kineski, engleski, francuski, njemački, talijanski, japanski, portugalski, ruski i španjolski. Nadalje, dostupni su primjeri postavljanja, pripreme i korištenja okruženja za API u podržanim programskim jezicima, često postavljana pitanja i korisnička podrška.

### **4.3.2. Speaker Recognition API**

Speaker Recognition API je API u oblaku koji se može podijeliti u dvije kategorije API-a: verifikacija govornika i identifikacija govornika. Verifikacija govornika je nova tehnologija prepoznavanja osobe koja automatski prepoznaje osobu na temelju njezinog glasa ili govora, a koristi se kako bi se povećala sigurnost ili koristilo glasovno upravljanje. Prije korištenja potrebno je odrediti neku izreku koja će se koristiti prilikom verifikacije korisnika ili ostalih mogućnosti. Odabrana izreka se prvotno upisuje, a zatim i izgovara kako bi se snimio glas i izgovor te izreke. U postupku verifikacije uspoređuje se snimljeni glas i izgovorena izreka te se uspoređuje s onom zadanom je li to ta osoba i da li je točno izgovorena izreka.

Identifikacija govornika podrazumijeva automatsku identifikaciju osoba preko njihovog glasa. Kako bi se osoba kasnije mogla identificirati mora proći kroz sličan postupak kao i kod verifikacije govornika, a to znači da osoba treba nešto izgovoriti i upisati što je izgovorila kako

bi dobila svoj jedinstveni glasovni potpis (eng. *voice signature*). Nadalje, kod korištenja identifikacije osobe, pretražuju se glasovni potpisi i prilikom pronalaska vraća se identitet osobe koja je pronađena. U dokumentaciji se još nalaze forma za slanje povratne informacije, primjeri na GitHub-u i upute za korištenje API-a.

### 4.3.3. Bing Speech

Bing Speech je API u oblaku koji programerima omogućuje implementaciju kontrole putem glasovnih naredbi, komunikaciju korisnika i aplikacije pomoću prirodnog jezika, diktiranje itd. Ovaj API je također podijeljen u dvije kategorije: prevođenje govora u tekst i teksta u govor, a Microsoft ga planira pripojiti pod Speech Services gdje smo već naišli na prevođenje govora u tekst i obratno. Naime, prevođenje govora u tekst ili razumijevanje govora programerima omogućuje dva načina implementacije pomoću REST (eng. *Representational State Transfer*) API-a ili klijentskih biblioteka. REST API se koristi pomoću HTTP poziva iz aplikacije na sustav za razumijevanje govora. Klijentske biblioteke treba preuzeti i povezati s našom aplikacijom. One su dostupne za Windows, Linux i macOS platforme i C#, Java, JavaScript, ObjectiveC programske jezike. REST API omogućava prijevod kraćih audio zapisa (do petnaest sekundi), dok klijentske aplikacije podržavaju tu kao i još neke mogućnosti. Na primjer, prevođenje dužeg audio zapisa od petnaest sekundi, strujanje audio zapisa sa privremenim rezultatima po želji korisnika, razumije prevođenje govora u tekst korištenjem Language Understanding Intelligent Service, kraće LUIS. Međutim, obje mogućnosti podržavaju slijedeće funkcionalnosti: Microsoftove napredne tehnologije prepoznavanja govora koje koriste Cortana, Office Dictation, Office Translator i ostali Microsoftovi proizvodi, prepoznavanje u realnom vremenu, optimiziranu tehnologiju za prepoznavanje govora prilikom interakcija, konverzacija i diktiranja. Za interaktivne opcije i diktiranja dostupni su slijedeći jezici: arapski, katalonski, danski, njemački, engleski, španjolski, finski, francuski, indijski, talijanski, japanski, korejski, norveški, nizozemski, poljski, portugalski, ruski, švedski i kineski, a za konverzacije arapski, njemački, engleski, španjolski, francuski, talijanski, japanski, portugalski, ruski i kineski. Za prevođenje teksta u audio zapis, tj. umjetni govor ili sintezu govora koristi se REST API.

### 4.3.4. Translator Speech API

Translator Speech API se može koristiti za dodavanje funkcionalnosti prevođenja govora u realnom vremenu ili prevođenja od kraja do kraja (eng. *end to end*) u aplikacije i alate za koje su nam potrebne bez obzira koju platformu ili programski jezik koristimo. Ovaj API koristi algoritme strojnog učenja i umjetne inteligencije, a možemo ga koristiti za prevođenje govora u govor i govora u tekst. Također, i ovaj API, kao i Bing Speech Microsoft planira svrstati

pod Speech Services. Translator Speech API koristi dvije tehnologije, stariju, statističko strojno prevođenje (eng. *statistical machine translation* - SMT) i novije živčano strojno prevođenje (eng. *neural machine translation* - NMT). SMT je dosegao vrhunac u performansama prevođenja, pa je napravljena nova tehnologija bazirana na živčanoj mreži (eng. *Neural Network* - NN) koja raznim algoritmima i matematičkim modelima pokušava simulirati živčani sustav živih bića. NMT ne pruža samo bolje performanse u kvaliteti prevođenja, već prijevodi zvuče mnogo čišće, tečnije, više nalik na ljude, nego SMT. Razlog tomu jest što NMT uzima cijeli kontekst rečenice koju prevodi dok SMT uzima samo kontekst nekoliko riječi prije i poslije riječi koju prevodi. Najveća unaprjeđenja očituju se kod prevođenja kineskog, japanskog i arapskog jezika. Prevođenja govora u govor koriste NMT dok prevođenja iz govora u tekst koriste kombinaciju SMT i NMT tehnologija. Međutim, ako je za odabrani jezik podržana NMT tehnologija u potpunosti onda se koristi samo NMT. Kada nije podržana NMT tehnologija u potpunosti, koristi se engleski jezik između dva jezika koja se prevodi što znači da se iz prvog jezika prevodi u engleski, pa zatim iz engleskog u drugi jezik pomoću NMT i SMT tehnologija. Jezici podržani za prevođenja iz govora u govor su: arapski, portugalski, kineski, engleski, francuski, njemački, talijanski, japanski, ruski i španjolski. Za lakše korištenje prikazani su primjeri u C#, Java, Node.js i Python programskim jezicima i kodom aplikacije za prevođenje u C#. Od korisnih poveznica tu su primjeri na GitHub-u i Stack Overflow-u kao i često postavljana pitanja.

#### **4.3.5. Immersive Reader**

Immersive Reader je skup alata koji koriste određene tehnike kako bi olakšali čitanje s razumijevanjem ljudima koji uče jezik, ljudima s poteškoćama ili onima koji uče čitati. S ovim alatima možemo svima olakšati čitanje tako što riječi mogu biti popraćene slikama koje na vizualan način opisuju riječ, omogućeno čitanje sadržaja, prevođenje sadržaja u drugi jezik, razdvajanje riječi na slogove, korištenje različitih boja za imenice, glagole, pridjeve itd. Koristi se tako da uključimo API koji će pokrenuti Immersive Reader i odrediti sadržaj koji će se automatski prikazivati pomoću Immersive Reader-a. U dokumentaciji se nalazi primjer pripreme okruženja za rad te primjeri za implementaciju u C#, Swift, Node.js i Python programskim jezicima. Preglednici koji podržavaju Immersive Reader su: Microsoft Edge, Internet Explorer 11, Google Chrome, Mozilla Firefox i Apple Safari.

#### **4.3.6. Language Understanding**

Language Understanding je API u oblaku koji omogućava programerima korištenje strojnog učenja u konverzaciji aplikaciji i korisnika, razumijevanje teksta na prirodnom jeziku i izvlačenje relevantnih i detaljnih informacija. Primjer konverzacije aplikacije i korisnika su chatbot-ovi.

Pri izradi chatbot-a Language Understanding API prima neki tekst, u ovom slučaju, poruku korisnika naše aplikacije preko HTTP zahtjeva za obradu umjetnom inteligencijom te vraća rezultat obrade u JSON formatu. U JSON formatu se nalaze informacije o poslanom upitu, smisao poslanog upita, rezultat te za svaki pojedini objekt u upitu naziv objekta, tip, broj početnog i završnog indeksa objekta. S tim podacima aplikacija može napraviti pametne odluke, na primjer odgovoriti korisniku na upit. Za korištenje Natural Language možemo koristiti već postojeći model podataka koji sadrži već postojeće izreke, svrhe i objekte, a možemo prilagoditi model svojim potrebama. Također možemo koristiti kombinaciju oba modela.

Language Understanding se može koristiti kao REST API sa svakim proizvodom, uslugom i okvirom koji podržava slanje HTTP zahtjeva. Također mogu se koristiti i klijentske biblioteke. Za kreiranje i upravljanje našim aplikacijama možemo koristiti LUIS sustav popraćen raznim tutorialima za koji je potrebno napraviti besplatan račun. U dokumentaciji se nalazi detaljan postupak kako postaviti radnu okolinu, dodati našu aplikaciju na sustav, testirati aplikaciju i Language Understanding API itd. Nadalje, tu su i postupci kako pripremiti aplikaciju za rad korištenjem klijentskih biblioteka u C#, Python, Node.js, Go i Java programskim jezicima.

#### **4.3.7. QnA Maker**

QnA Maker je također API u oblaku koji služi za kreiranje pitanja i odgovora za konverzaciju pomoću naših podataka. QnA Maker možemo koristiti kada imamo statične informacije u bazi znanja. Baza znanja se može prilagoditi našim potrebama sa dokumentima u PDF ili URL formatu. Također, QnA Maker možemo koristiti kada želimo za isti zahtjev ili pitanje dati isti odgovor za više korisnika. Nadalje, možemo ga koristiti za filtriranje statičnih informacija baziranim na metapodacima tako što dodamo metapodatke koji su relevantni za naš sadržaj i važni za naše korisnike. Pomoću ove skupine alata možemo kreirati bazu znanja tako što spremamo pitanja i najbolje odgovore na ta pitanja. Postoje već zadana pitanja i odgovori koje možemo mijenjati, brisati i dodavati druge ovisno o našim potrebama. Možemo čak dodati dokumente poput često postavljana pitanja, priručnik proizvoda, web stranice koje će biti strukturirane u pitanja i odgovore u našoj bazi znanja. Preporuča se čak dodavanje nekih nebitnih informacija u našu bazu znanja koje nemaju veze s našom aplikacijom kako bi komunikacija s korisnikom bila prirodnija. Također, poželjno je dodati sinonime u bazu znanja radi lakšeg pronalaska najboljeg odgovora. Primjerom je prikazano kako se kreira baza znanja QnA Maker web aplikaciji.

Kada smo kreirali bazu znanja moramo je još objaviti kako bi dobili krajnju točku (eng. *endpoint*) koju koristimo u našoj aplikaciji. Kada smo to napravili možemo testirati slanje



zahtjeva za odgovorom iz baze znanja pomoću dvije metode: Postman i cURL. Nadalje, ove funkcionalnosti su objašnjene na primjerima za C#, Go, Java, Node.js i Python korištenjem REST API-a i C#, Node.js i Python korištenjem klijentskih aplikacija.

Prilikom izrade chatbot-a možemo koristiti kombinaciju Language Understanding i QnA Maker tako što prilikom postavljanja pitanja korisnika odredimo svrhu pomoću Language Understanding API-a. Zatim pomoću QnA Maker API-a tražimo odgovor na postavljeno pitanje uz pomoć određene svrhe koju smo odredili pomoću Language Understanding te vraćamo odgovor.

### **4.3.8. Text Analytics API**

Text Analytics API je API u oblaku koji koristi umjetnu inteligenciju za prirodni jezik kako bi za dani tekst mogli odrediti osjećaje koji prevladavaju, ključne riječi, jezik i entitete. Ove funkcionalnosti možemo koristiti pomoću REST API-a ili klijentskih biblioteka u C#, Python, Node.js, Go i Ruby programskim jezicima koji su popraćeni primjerima postavljanja radnog okruženja aplikacije i implementacije funkcionalnosti.

Analiza osjećaja se koristi kako bi iz nekog teksta zaključili da li je pozitivan ili negativan. Na primjer, ako korisnik komentira naš proizvod da li je komentar pozitivan ili negativan. Na temelju danog teksta API vraća rezultat između nula i jedan gdje jedan označava najveću pozitivnost. Podržani jezici za analizu osjećaja su: danski, nizozemski, engleski, finski, francuski, njemački, grčki, talijanski, norveški, poljski, portugalski, ruski, španjolski, švedski i turski.

Prepoznavanje ključnih riječi služi za automatsko i brzo razumijevanje danog teksta. Podržano je za danski, nizozemski, engleski, finski, francuski, njemački, talijanski, japanski, korejski, norveški, poljski, portugalski, ruski, španjolski i švedski jezik.

Prepoznavanje entiteta služi za kategorizaciju riječi u određene kategorije kojima pripadaju, na primjer, ljudi, mjesta, organizacije, datumi, količine, postoci, valute itd. Podržani su svi jezici kao i za prethodne funkcionalnosti uz dodatak arapskog, češkog i kineskog jezika.

Trenutno postoji ograničenje od pet tisuća sto dvadeset znakova po analizi teksta tako da ako dokument za analizu teksta ima više od navedenog broja znakova potrebno ga je podijeliti u manje odlomke. U dokumentaciji su pomoću tutorijala prikazani kako uz pomoć Power BI aplikacije i Text Analytics API-a prikazati podatke, kako uz pomoć Text Analytics API-a analizirati tekst, kako implementirati analizu osjećaja kod strujanja podataka koristeći Azuer DataBricks te kako izraditi web aplikaciju koja koristi Cognitive Services za prevođenje teksta, analizu osjećaja i prevođenje iz teksta u audio signal, tj. govor.

### 4.3.9. Translator Text

Translator Text je API u oblaku koji služi za prevođenje teksta u gotovo realnom vremenu podržan na više od šezdeset jezika. API koristi ranije spomenute NMT i SMT tehnologije. Ovaj API omogućuje prevođenje teksta, prevođenje u drugo pismo, prepoznavanje jezika i određivanje duljine rečenice što je popraćeno primjerima u C#, Java, Python, Node.js i Go programskim jezicima. Tutorijalima je prikazano kako kreirati aplikaciju za prevođenje u C# programskom jeziku uz WPF okvir što je zamjena za .NET okvir i aplikacija za prevođenje teksta, analizu osjećaja i prevođenje teksta u govor u Python programskom jeziku sa Flask okvirom. Nadalje, ovaj API omogućuje prilagođavanje prevođenja pomoću NMT tehnologije u verziji tri API-a što nam može biti od velike koristi ako u svojim dokumentima ili aplikaciji koristimo specifičnu terminologiju. U dokumentaciji se još nalaze općenite informacije poput pretplate za API, prelaska na noviju verziju tri, izbacivanja vulgarnih riječi, dobivanja duljine riječi, ne prevođenja određenih riječi poput naziva tvrtke i slično.

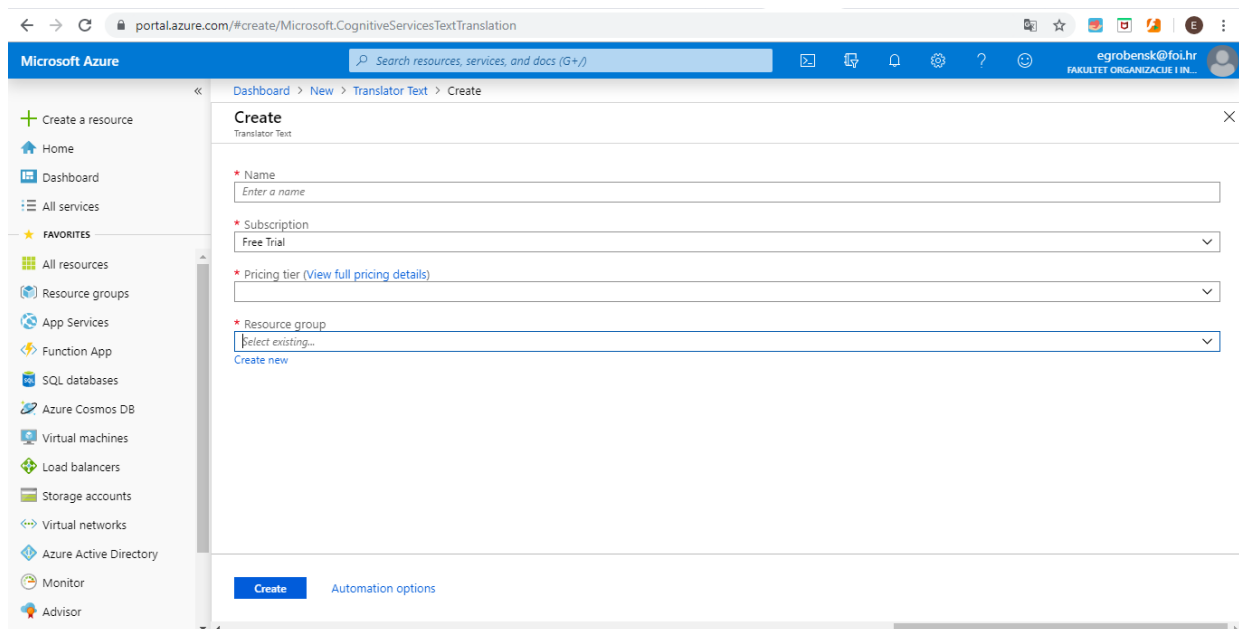
Gotovo svaki od obrađenih API-a nudi formu za slanje povratnih informacija, povezanost s nekim korisnim poveznicama poput GitHub-a ili Stack Overflow-a. Dokumentacije su vrlo detaljne popraćene mnoštvom primjera u više programskih jezika što je velika prednost. Naime, toliko sadržaja u dokumentaciji pomalo skreće fokus s važnih informacija.

U usporedbi sa prethodna dva sustava umjetne inteligencije za prirodni jezik IBM Watson i Google Cloud Natural Language, po mom mišljenju, rekao bih da je Microsoft Azure Cognitive Services APIs donekle spoj oba. Kao što možemo vidjeti, Microsoft Azure Cognitive Services APIs nudi širinu API-a i funkcionalnosti poput IBM Watson sustava, ali također zadržava tu detaljnost dokumentacije popraćene mnoštvom primjera kao Google Cloud Natural Language.

## 5. Aplikacija Prevoditelj

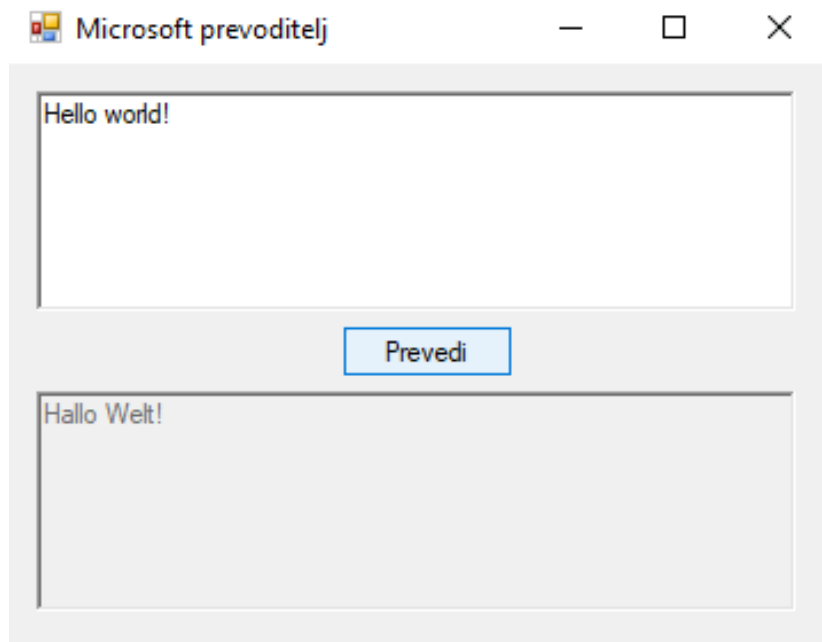
Aplikacija je izrađena pomoću Microsoftovog API-a za prevođenje teksta Translator Text koji je dostupan preko računa na Microsoft Azure sustavu pod kategorijom Cognitive Services. Za aplikaciju sam odabrao programski jezik C# zato što postoje objašnjenja i primjeri u dokumentaciji API-a te sam već imao iskustva sa njime. Budući da za ovaj API ne postoje klijentske biblioteke API-u moramo pristupati preko HTTP zahtjeva.

Kako bismo počeli sa radom prvo smo se morali registrirati na Microsoft Azure sustav. Nakon što smo to napravili potrebno je kreirati resurs kako bismo dobili podatke o ključu i poveznici tzv. endpoint s čime pristupamo API-u.



Slika 5: Kreiranje resursa na Microsoft Azure platformi (Microsoft Azure, nema dat.)

U prvo polje unosimo proizvoljan naziv resursa. Zatim odabiremo pretplatu, za dodatne pogodnosti moguće se pretplatiti na viši nivo računa, ali u ovom slučaju nam to nije potrebno. Nadalje, odabiremo paket koji sadrži određeni broj znakova za prijevod za određenu cijenu. Također, postoji i probni besplatan paket u trajanju od mjesec dana koji podržava prijevod dva milijuna znakova što je sasvim dovoljno za probu. U zadnje polje unosimo grupu resursa ili kreiramo novu ako još nemamo ni jednu. Kada smo kreirali resurs možemo pristupiti mnogim korisnim mogućnostima poput dnevniku aktivnosti gdje možemo vidjeti kada su vršene određene radnje, kontroli pristupa gdje možemo dodati ili ukloniti uloge našeg API-a te postaviti dozvole, pratiti koliko smo znakova iskoristili, pristupiti podršci itd.



Slika 6: Sučelje aplikacije

U prvo polje unosimo tekst koji želimo prevesti i klikom na gumb prevedi prevodimo tekst. Polje u kojem se ispisuje tekst je zaključano te se ne može mijenjati sadržaj, već se u njemu ispisuje isključivo prevedeni tekst. Postavljeni jezik u koji se prevodi je njemački, a tekst koji se prevodi nije potrebno navesti jer API sam može odrediti o kojem se jeziku radi.

## 5.1. Programski kod aplikacije

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Net;
using System.Net.Http;
using System.Xml.Linq;
namespace Microsoft_prevoditelj
{
    public partial class Form1 : Form
    {
        private const string Kljuc = "";
```

```

public static string prijevod;
public Form1()
{
    InitializeComponent();
}
private static readonly HttpClient klijent = new HttpClient
{
    Kljuc } }
    DefaultRequestHeaders = { { "Ocp-Apim-Subscription-Key",
};
public static async Task<string> prevedi(string text, string
jezik)
{
    var enkodirano = WebUtility.UrlEncode(text);
    var url =
"https://api.microsofttranslator.com/V2/Http.svc/Translate?" +
$"to={jezik}&text={enkodirano}";
    var rezultat = await klijent.GetStringAsync(url);
    return XElement.Parse(rezultat).Value;
}
private async void btnPrevedi_Click(object sender, EventArgs
e)
{
    await dohvati(txtSadrzaj.Text);
    txtPrevod.Text = prijevod;
}
public static async Task dohvati(string text)
{
    prijevod = await prevedi(text, "de");
}
}
}

```

Kao što je vidljivo iz koda aplikacije, na početku smo morali dodati C# biblioteke za rad s HTTP zahtjevima i xml formatom zapisa. Naime, u šesnaestoj liniji koda umjesto praznog stringa se nalazi kod ključa API-a koji se dodaje u zaglavlje zahtjeva kako bismo pristupili našem API-u, ali je izbrisan kako mu ne bi pristupali korisnici koji za to nemaju ovlasti. Nadalje, u četrdeset i petoj liniji koda u argumentu funkcije prevedi prosljeđuje se jezik prevođenja. Naime, u ovoj aplikaciji je postavljeno da se prevodi u njemački jezik koji ima oznaku de. Oznake jezika možemo pronaći u dokumentaciji Translator Text API-a, te promjenom tog stringa mijenjamo jezik u koji će API prevoditi unesene riječi.

## 6. Analiza i usporedba API-a na primjeru

Za ovu usporedbu potrebna su nam tri API-a za svako od poduzeća IBM, Google i Microsoft koji posjeduju istu funkcionalnost. Odabrao sam API-e za prevođenje teksta iz jednog jezika u drugi zato što IBM i Google podržavaju demo verziju aplikacije za prevođenje koja koristi njihov API te sam napravio primjer aplikacije koja koristi Microsoft-ov API za prevođenje teksta.

Test sam osmislio tako da u sve aplikacije unesemo isti, od jednostavnijeg prema složenijim tekstovima i prevedemo u isti jezik. Nadalje, usporediti ćemo međusobne prijevode aplikacija i sa službenim prijevodom kako bismo vidjeli koji API je preveo najbliže onom službenom. Odabrani tekstovi će biti prevedeni u hrvatski jezi kako bi čitatelji mogli lakše uvidjeti razlike između prijevoda.

Kao prvi primjer testnog teksta koristiti ćemo tekst novinskog članka na engleskom jeziku koji glasi: „But there are concerns that Apple News and Facebook Instant Articles — which are not paying for the content — will chip away at the notion of a newspaper or magazine website as a standalone brand to be digested whole, rather than broken up into individual stories and scattered across different apps. The services could "dilute publisher brands as destination sites and boost the platform brand", says Gordon Crovitz, a media adviser and former publisher of the Wall Street Journal.“ („Tečaj engleskog“, kolovoz 2015.)

Prema Tečaju engleskog, najbolji prijevod danog teksta bi glasio: „No postoji i zabrinutost da će Apple News i Facebook Instant Articles — koji ne plaćaju sadržaj — ugroziti pojam internetskih novina ili časopisa kao samostalnog brenda koji treba čitati kao cjelinu, a ne razlomljenog na pojedinačne priče i razbacanog na različitim aplikacijama. Takvi bi servisi mogli "oslabiti brend nakladnika kao ciljanog internetskog odredišta i ojačati brend platforme", kaže Gordon Crovitz, medijski savjetnik i nekadašnji nakladnik The Wall Street Journala.“ („Tečaj engleskog“, kolovoz 2015.)

Testovi će se redom provoditi za IBM Translate Text, Google Translation i Microsoft Translator Text API. Dakle, prvi prijevod je prijevod pomoću demo verzije aplikacije koja koristi IBM-ov Translate Text, a glasi: „Međutim, postoje zabrinutosti da Apple News i Facebook Instant Članke-koji ne plaćaju za sadržaj-čip će se raspršiti na ideji lista novina ili časopisa kao samostalnu marku koja će se probaviti cjelinu, umjesto da se razne u pojedinačne priče i raštrkane po različitim aplikacijama. Usluge bi mogle "razrijediti izdavača marke kao odredišne lokacije i pojačati brand platforme", kaže Gordon Crovitz, savjetnik za medije i bivši izdavač Wall Street Journala.“ Kao što možemo vidjeti, prijevod pomoću IBM-ovog Translate Text ima dosta nedostataka i pogrešaka. Prva, i po meni, najveća pogreška je prevođenje naziva u prvaj

rečenici (Articles) i prijevod glagola „chip“ u frazi „chip away“ kao imenice što u potpunosti narušava smisao teksta. U principu, nije pogođen smisao teksta, ali je većina riječi dobro prevedena.

Sljedeći prijevod je pomoću Google-ove demo aplikacije koja koristi Translation API: „Ali postoje zabrinutosti da će Apple News i Facebook Instant Članci - koji ne plaćaju sadržaj - otkinuti pri poimanju novina ili časopisa kao samostalne marke da bi se mogli cijepati u cjelini, a ne raščlaniti na pojedinačne priče i raštrkati. preko različitih aplikacija. Usluge bi mogle "razrijediti brendove izdavača kao odredišne stranice i pojačati platformu", kaže Gordon Crovitz, medijski savjetnik i bivši izdavač Wall Street Journala.“ Ponovo nailazimo na prijevod naziva u prvoj rečenici iste riječi kao i prethodni API te promašeni smisao teksta. Po mom, mišljenju ne možemo zaključiti da je prijevod ni bitno bolji ni bitno lošiji od prijevoda IBM-ovog API-a.

Nadalje, slijedi prijevod u izrađenoj aplikaciji koja koristi Microsoft Translator Text API, glasi: „No, postoji zabrinutost da Apple News i Facebook instant članci – koji ne plaćaju za sadržaj — će čip daleko na pojam novinskog ili časopisa web stranice kao samostalan brand biti probavljeni cjelinu, a ne razlomljena u pojedinačne priče i raspršene po različitim aplikacijama. Usluge bi mogle "razrijediti robne marke izdavača kao odredišne stranice i potaknuti brand platforme", kaže Gordon Crovitz, medijski savjetnik i bivši izdavač Wall Street Journala.“ Kao što možemo vidjeti, ovaj API je preveo dvije riječi u nazivu (Instant Articles) i ponovio pogrešku prevođenja fraze „chip away“ te je nastavak sličan prijevodima prethodnih API-a.

U sva tri prijevoda istog teksta možemo vidjeti da je završetak teksta odvojen zarezima dobro i smisleno preveden iz čega možemo naslutiti da je API-ima lakše prevoditi kraće tekstove ili fraze.

Slijedeći testni tekst će biti stručni članak koji glasi: „The EU should promote a single market for the European medical technology industry, among other things through adequate standardisation in the field of biomedical engineering in combination with the care services industry, ICT and health informatics.“ (Glosbe, nema dat.)

Prema Glosbe web aplikaciji, prijevod bi trebao glasiti: „EU bi trebao promovirati jedinstveno tržište za europsku industriju medicinske tehnologije, između ostalog putem odgovarajuće standardizacije na području biomedicinskog inženjeringa i u kombinaciji s industrijom usluga skrbi, IKT-om i zdravstvenom informatikom.“ (Glosbe, nema dat.)

Redom, prvi prijevod je pomoću demo aplikacije IBM Translate Text: „EU bi trebala promovirati jedinstveno tržište za europsku medicinsku tehnološku industriju, između ostalog kroz primjerenu standardizaciju u području biomedicinskih inženjeringa u kombinaciji s

industrijom zdravstvene skrbi, ICT i zdravstvenim informatičarima.“ U ovom prijevodu je pogođen smisao uz neke pogreške poput prijevoda kratice i pojedinih riječi koje ne narušavaju toliko smisao teksta kao u prethodnom primjeru.

Slijedi prijevod pomoću Google Translation demo aplikacije: „EU bi trebala promicati jedinstveno tržište europske industrije medicinske tehnologije, između ostalog i odgovarajućom standardizacijom u području biomedicinskog inženjerstva u kombinaciji s industrijom skrbi, IKT-om i zdravstvenom informatikom.“ Kao što možemo vidjeti prijevod je gotovo identičan originalnom prijevodu te je pogođen smisao teksta.

Nadalje, slijedi prijevod Microsoftovog API-a Translator Text: „EU bi trebao promicati jedinstveno tržište za europsku industriju medicinskih tehnologija, među ostalim i odgovarajućom normizacijom u području biomedicine u kombinaciji s industrijom usluge skrbi, IKT-om i zdravstvenom informatikom.“ Također i ovaj prijevod je gotovo isti kao i originalni prijevod te je u potpunosti iskoristiv.

Posljednji test je ujedno i najzahtjevniji test, a to je tekst na francuskom jeziku. Tekst koji će se prevoditi je šesta strofa francuske himne Marseljeze.

Tekst na francuskom glasi:

„Amour sacré de la Patrie  
Conduis, soutiens nos bras vengeurs  
Liberté, Liberté chérie,  
Combats avec tes défenseurs !  
Sous nos drapeaux, que la victoire  
Accoure à tes mâles accents,  
Que tes ennemis expirants  
Voient ton triomphe et notre gloire !“  
(„Marsljeza - Wikizvor“, 30. 1. 2017.)

Hrvatski prijevod prema kojemu ćemo uspoređivati prijevode API-a za ovu strofu:

„Ljubavi sveta za Domovinu  
Vodi, ojačaj naše osvetničke mišice!  
Slobodo! Slobodo draga,  
Bori se sa svojim braniteljima!  
Pod naše zastave neka pobjeda



Dotrči na tvoje muške pozive!

Neka tvoji umirući neprijatelji

Vide tvoj trijumf i našu slavu!“

(„Marsljeza - Wikizvor“, 30. 1. 2017.)

Prvi prijevod je pomoću IBM Translate Text demo verzije aplikacije:

„Sveta ljubav od Domovinu

Provodis, podržava naše osvetomaste ruke

Liberté, Liberté Chérie,

Combats avec tes branitelji!

Pod našim zastavama, da je pobjednički

Accoure vašim muškim akcentima,

neka vaši istek neprijatelji

vide svoj trijumf i našu slavu!“

Kao što možemo vidjeti, smisao pjesme nije pogođen da bismo mogli reći da je prijevod zadovoljavajući, neke riječi uopće nisu ni prevedene, već su ostavljene u izvornom obliku.

Slijedeći API je Google Translation čiji prijevod glasi:

„Sveta ljubav Oče domovine,

podrži naše osvetoljubive ruke

Sloboda, Sloboda draga,

Borite se sa svojim braniteljima!

Pod našim zastavama, neka pobjeda

Dođite do vaših muških naglaska,

Neka vaši neprijatelji istjeruju.

Pogledajte vaš trijumf i našu slavu!“

U usporedbi s prethodnim prijevodom možemo vidjeti da je mnogo čitljiviji, sve riječi su prevedene te daleko upotrebljiviji. Posljednji prijevod je preveden pomoću aplikacije Prevoditelj koja koristi Microsoft Translator Text API:

„Sveta ljubav domovine

Vozi, podrži naše osvetoljubljive ruke.

Freedom, Freedom, draga,  
Borite se sa svojim braniteljima!  
Pod našim zastavama, ta pobjeda  
Akosting svoje muške naglaske,  
Neka vaš istekli neprijatelji  
Vidi svoj trijumf i našu slavu!“.

Iz priloženoga možemo vidjeti da prijevod Microsoftovog API-a nije zadovoljio uvjete. Mogli bismo ga smjestiti na drugo mjesto po kvaliteti prijevoda ispred IBM-ovog API-a i iza Google-ovog. Naime, nismo ni mogli očekivati da prijevodi budu identični originalu s obzirom na zahtjevnost teksta.

Prema viđenome, možemo zaključiti da su prevoditelji dovoljno dobri i u potpunosti iskoristivi u kraćim prijevodima. Međutim, dulje prijevode moramo uzeti sa rezervom jer u prvom testu možemo vidjeti kako prijevodi nisu u duhu hrvatskog jezika dok u drugom testu jesu. Nadalje, možemo pretpostaviti da u tome ulogu ima nepopularnost hrvatskog jezika, tj. hrvatski jezik je jezik manjeg govornog područja, pa je stoga slabije podržan. Daljnjim razvojem i unaprjeđenjem API-a možemo očekivati, u bliskoj budućnosti, širu primjenu API-a za prevođenje jezika. Sličan zaključak su

## 7. Zaključak

Umjetna inteligencija za prirodni jezik je nova tehnologija koja je još u stadiju razvoja i koja sigurno ima svoju primjenu u životu privatnih i poslovnih korisnika. Obradom API-a možemo vidjeti trenutnu širinu primjene umjetne inteligencije za prirodni jezik koja će s vremenom sve više rasti. Iako, velike kompanije vode bitku u bržem napretku na području umjetne inteligencije za prirodni jezik kao i na ostalim područjima umjetne inteligencije i strojnog učenja, manja poduzeća bi se također trebala uključiti u osmišljavanje novih mogućnosti umjetne inteligencije te korištenja već postojećih kako bi unaprijedila svoje poslovanje i istaknula se nad konkurencijom. Naime, umjetna inteligencija nije ograničena samo na poslovanja tvrtki, već se može koristiti i u obrazovne svrhe. Tako da bi se svakako obrazovne ustanove trebale uključiti umjetnu inteligenciju u svoje programe, a posebno visoko obrazovne ustanove s područja informacijskih i komunikacijskih znanosti, računarstva i informatike te potaknuti razvoj i osmišljavanje projekata. Nadalje, kad se već odlučimo krenuti u razvoj aplikacije s implementacijom umjetne inteligencije nailazimo na problem koje aplikacijsko sučelje odabrati. Vodeće kompanije uglavnom podržavaju slične funkcionalnosti, ali i besplatne probne periode. Koristeći besplatni probni period možemo isprobati i usporediti koje aplikacijsko programsko sučelje najbolje odgovara našim potrebama. Kao što smo imali prilike vidjeti u testovima, prijevodi mogu, ali i ne moraju biti u potpunosti točni: Budući da su testovi bili osmišljeni tako da se tekst prevodi u hrvatski jezik koji je jezik manjeg govornog područja, pa time i nepopularniji stoga postoji mogućnost da bi prijevodi bili kvalitetniji kada bi se radilo o nekom od vodećih svjetskih jezika. Svakako smatram da prijevode pomoću strojnog prevođenja, još uvijek, moramo uzimati sa rezervom pogotovo kod prevođenja ozbiljnijih dokumenata. Također, smatram da mogu biti vrlo korisni kao pomoć pri prevođenju i lakšem razumijevanju danih tekstova, a prostora za napredak tehnologije strojnog prevođenja i umjetne inteligencije definitivno ne nedostaje.

## Popis literature

- [1] *AI (artificial intelligence)* (2018), preuzeto 1.9.2019. s <https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence>
- [2] Andovar, *Machine translation*, preuzeto 11.9. s <https://www.andovar.com/machine-translation/>
- [3] freeCodeCamp (2018), preuzeto 11.9.2019. s <https://www.freecodecamp.org/news/a-history-of-machine-translation-from-the-cold-war-to-deep-learning-f1d335ce8b5>
- [4] Garbade M. J. (2018) *A Simple Introduction to Natural Language Processing*, preuzeto s <https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32>
- [5] Glosbe (2019), preuzeto 7.9.2019. s <https://hr.glosbe.com/>
- [6] Google (2019), preuzeto 1.9.2019. s <https://cloud.google.com>
- [7] IBM (2019), preuzeto 1.9.2019. s <https://cloud.ibm.com/>
- [8] Microsoft (2019), preuzeto 1.9.2019. s <https://docs.microsoft.com/>
- [9] Šimunić Z. (2017)  
*Ljudsko prevođenje vs strojno prevođenje - iz perspektive prevoditelja* preuzeto 11.9.2019. s <https://www.traducta-prijevođa.hr/single-post/2017/08/22/Ljudsko-prevo%C4%91enje-vs-strojno-prevo%C4%91enje---iz-perspektive-prevoditelja>
- [10] *TečajEngleskog* (2015), preuzeto 7.9.2019. s [http://www.tranexp.hr/TecajEngleskog/TecajEngleskog-broj\\_210.html](http://www.tranexp.hr/TecajEngleskog/TecajEngleskog-broj_210.html)
- [11] *Umjetna inteligencija*, preuzeto 1.9.2019. s <http://www.enciklopedija.hr/natuknica.aspx?ID=63150>
- [12] Wikipedia (2019) *Application programming interface*, preuzeto 1.9.2019. s [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface)
- [13] Wikipedia (2019) *Machine translation*, preuzeto 11.9. s [https://en.wikipedia.org/wiki/Machine\\_translation](https://en.wikipedia.org/wiki/Machine_translation)

[14] Wonderflow (2018) *12 NLP Examples: How Natural Language Processing is Used* preuzeto 1.9.2019. s <https://www.wonderflow.co/natural-language-processing-examples/>

## Popis slika

Slika 1: Shazam logo.....	3
Slika 2: IBM Watson logo .....	5
Slika 3: Google Cloud logo .....	10
Slika 4: Microsoft logo .....	12
Slika 5: Kreiranje resursa na Microsoft Azure platformi.....	19
Slika 6: Sučelje aplikacije .....	20