Jezični model hrvatske Wikipedije

Juričić, Antonio

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski studij računarstva

Završni rad

Jezični model hrvatske Wikipedije

Rijeka, srpanj 2022.

Antonio Juričić
0069088108

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski studij računarstva

Završni rad

Jezični model hrvatske Wikipedije

Mentor: prof.dr.sc. Ivo Ipšić

Rijeka, srpanj 2022.

Antonio Juričić
0069088108

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za računarstvo**
Predmet: **Programiranje II**
Grana: **2.09.04 umjetna inteligencija**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Antonio Juričić (0069088108)**
Studij: Preddiplomski sveučilišni studij računarstva

Zadatak: **Jezični model hrvatske Wikipedije**

Opis zadatka:

Opišite postupak gradnje jezičnog modela zasnovanog na stohastičkim modelima pojavljivanja pojedinih riječi. Na temelju statistike pojavljivanja pojedinih riječi i kombinacija više različitih riječi u tekstovima hrvatske Wikipedije ocijenite vjerojatnosti pojavljivanja nizova od dvije tri i četiri riječi, te ocijenite kompleksnost jezičnog modela. Korištenjem slobodnog alata "SRI Language Modelling Toolkit" izgradite jezični model hrvatske Wikipedije, te usporedite rezultate procjene vjerojatnosti nizova hrvatskih riječi.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor: Prof. dr. sc. Ivo Ipšić

Predsjednik povjerenstva za završni ispit: Prof. dr. sc. Kristijan Lenac

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, srpanj 2022.

_____
Antonio Juričić

Zahvala

Zahvaljujem mentoru prof. dr. sc. Ivi Ipšiću na smjernicama i pomoći tijekom izrade ovog završnog rada. Zahvaljujem i svojoj obitelji na podršci tijekom studiranja.

Sadržaj

Popis slika

Popis tablica

Poglavlje 1

Uvod

Jezično modeliranje jedan je od ključnih koncepata unutar polja obrade prirodnog jezika, grane računarstva koja principima umjetne inteligencije vrši procesiranje i analizu ljudskog jezika.[1]

Jezični modeli u današnje vrijeme koriste se u razne svrhe i imaju raznu primjenu u svakodnevnom životu. Unutar alata za strojno prevođenje, jezični se model koristi za slaganje točnijih i smislenijih prijevoda. Programi za obradu teksta prepoznavaju pravopisne i gramatičke pogreške te predviđaju ispravne izraze upravo na temelju informacija dobivenih jezičnim modeliranjem. U sve naprednijim sustavima za prepoznavanje govora, jezični modeli igraju ključnu ulogu pri prepoznavanju nerazumljivo izgovorenih riječi analiziranjem konteksta unutar kojeg je riječ izgovorena.

Jezični modeli grade se na korpusima koji se mogu sastojati od različitih književnih i znanstvenih djela, članaka, izvještaja. Jedno od većih mjesta na internetu koje sadrže tekstove na hrvatskom jeziku jest hrvatska Wikipedija.

1.1 Zadatak završnog rada

Zadatak završnog rada bio je opisati postupak izgradnje stohastičkog jezičnog modela te izgraditi jezični model na temelju korpusa hrvatske Wikipedije pomoću alata "SRI Language Modeling Toolkit". Koristeći jezični model bilo je potrebno odrediti vjerojatnost pojavljivanja različitih nizova riječi i ocijeniti kompleksnost samog modela.

Poglavlje 2

Stohastički jezični model

2.1 Definicija

Jezični model označava distribuciju vjerojatnosti pojavljivanja riječi i njihovih kombinacija unutar nekog jezičnog konteksta. Za razliku od deterministčkih modela koji su izgrađeni na temelju predefiniranih pravila, stohastičko jezično modeliranje oslanja se na probabilističku prirodu slučajnog pojavljivanja skupova riječi unutar jezika i računanja njihove vjerojatnosti iz zadanog skupa podataka. Zadaća jezičnog modela jest prepoznati strukture i redundancije koje se pojavljuju unutar jezika i na taj način stvoriti kriterije po kojima se riječi mogu međusobno slagati i tvoriti smislene rečenice.

Koristeći formulu uvjetne vjerojatnosti, vjerojatnost pojavljivanja skupa riječi zapisujemo kao skup vjerojatnosti pojavljivanja svih prethodnih riječi od kojih se sastoji:[2]

$$P(w_1^n) = P(w_1)P(w_2|w_1)...P(w_n|w_1^{n-1}) = \prod_{i=1}^{n} P(w_i|w_1^{i-1}) \tag{2.1}$$

gdje je:

- w_1^n - niz n riječi
- w_i - riječ na i-tom mjestu

- w_1^{i-1} - vjerojatnost prijašnjih $i-1$ riječi

Problem kod ovakvog načina računanja vjerojatnosti pojavljivanja riječi javlja se kod dugih nizova riječi. Zbog same prirode jezika, što je niz riječi dulji, veća je vjerojatnost da se navedeni niz riječi neće pojaviti unutar korpusa jezičnog modela te bi bilo nemoguće točno odrediti vjerojatnost za taj niz. Kao rješenje tom problemu javljaju se različite vrste jezičnih modela, od kojih je jedan N-gramski jezični model.

2.2 N-gram model

Najjednostavniji i najčešći tip stohastičkog jezičnog modela jest N-gram jezični model. On počiva na takozvanoj Markovljevoj pretpostavci koja govori da se vjerojatnost novog (budućeg) stanja može aproksimirati uzimajući u obzir samo mali broj prethodnih stanja. U kontekstu jezičnog modeliranja to bi značilo da se vjerojatnost pojavljivanja riječi može aproskimirati uzimajući u obzir vjerojatnost pojavljivanja samo N - 1 prethodnih riječi. Kod bigram modela, za određivanje vjerojatnosti pojavljivanja trenutne riječi u obzir se uzima samo jedna prethodna riječ. Kod trigram modela, promatraju se dvije prethodne riječi. Općenito vrijedi:

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-N+1}^{n-1}) \tag{2.2}$$

gdje je:

- $P(w_n|w_1^{n-1})$ - uvjetna vjerojatnost pojavljivanja riječi na poziciji n
- $P(w_n|w_{n-N+1}^{n-1})$ - vjerojatnost pojavljivanja riječi w_n s obzirom na prethodnih $n-N+1$ riječi

Na temelju navedene pretpostavke moguće je pretpostaviti vjerojatnost nad cijelim nizom riječi promatranjem isključivo N prethodnih riječi, za bigram:

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1}) \tag{2.3}$$

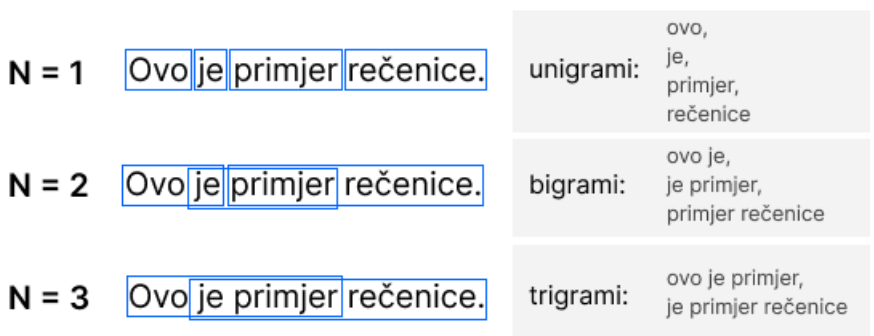
odnosno promatranjem trigrama:

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-2}, w_{k-1}) \tag{2.4}$$

gdje je:

- $P(w_1^n)$ - vjerojatnost niza n riječi
- w_k - riječ na poziciji k

Na slici (2.1) prikazan je primjer razdvajanja rečenice na N-grame te su ispisani unigrami, bigrami i trigrami nastali takvim razdvajanjem.



Slika 2.1 Primjer razdvajanja rečenice na N-grame

2.2.1 Računanje vjerojatnosti

Način na koji se računa vjerojatnost N-grama naziva se metoda maksimalne vjerojatnosti (engl. *Maximum Likelihood Estimation* - MLE). Za parametre modela N-grama MLE procjena dobiva se normalizacijom njihove frekvencije unutar korpusa. Takva procjena uvijek iznosi između 0 i 1. Općenita jednadžba za računanje MLE procjene glasi:[3]

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \tag{2.5}$$

gdje je:

- $P(w_n|W_{n-N+1}^{n-1})$ - vjerojatnost pojave riječi w_n,
- $C(w_{n-N+1}^{n-1}w_n)$ - frekvencija pojave riječi w_n kojoj prethodi niz riječi,
- $C(w_{n-N+1}^{n-1})$ - frekvencija pojave niza koji prethodi riječi (prefiksa)

2.2.2 Zaglađivanje

Budući da korpus jezičnog modela ne mora sadržavati sve riječi koje nudi vokabular određenog jezika, pri procjeni vjerojatnosti testnog seta unutar ispravnog niza riječi može doći do pojave riječ s kojom se jezični model prethodno nije susreo, te se cijelom nizu pridružuje neispravna vjerojatnost nula. Kako bi se takve anomalije izbjegle, koriste se metode zaglađivanja (engl. *smoothing*). One omogućuju ujednačajniju raspodjelu vjerojatnosti među različitim N-gramima i pomažu pri procjeni N-grama s kojima se model prethodno nije susreo.

Metoda zaglađivanja koja inkrementira svaki N-gram unutar modela i tako uklanja mogućnost pojave nule pri procjeni testnog seta naziva se Laplaceovo zaglađivanje. To je najjednostavnija metoda zaglađivanja, no u modernim jezičnim modelima ne nudi dovoljnu točnost upravo zbog nedostatka prilagodljivosti različitim situacijama, stoga se u praksi koriste neke druge metode. Slična metoda je i add-k zaglađivanje, koja umjesto jedinice dodaje proizvoljnu težinu k svim N-gramima.

Sofisticiranije metode zaglađivanja primjenjuju ustručavanje (engl. *backoff*) i interpolaciju za kombiniranje N-grama različitog reda kako bi se dobila točnija vjerojatnost nepostojećeg N-grama. Ustručavanje podrazumijeva korištenje nižih redova N-grama ukoliko ne postoji N-gram traženog reda. Tako na primjer ako ne postoji određeni trigram, metoda koristi bigram, ako ni on ne postoji koristi unigram. Kod interpolacije za dobivanje vjerojatnosti uzimaju se vrijednosti svih redova N-grama, pridodaje im se određena težina te se kombiniraju (interpoliraju).

Neke od najčešće korištenih takvih metoda su Good-Turing, Kneser-Ney i Witten-Bell metoda. Zadana metoda zaglađivanja u alatu SRILM toolkit korištenog u ovom radu jest Good-Turing metoda zaglađivanja.[4]

2.3 Evaluacija modela

Kvaliteta i učinkovitost jezičnog modela određuje se na dva načina. Prvi način, ujedno i najpouzdaniji, jest testiranje modela nakon integracije u konkretnoj aplikaciji i naziva se ekstrinzični tip evaluacije. Međutim, ovakav je tip evaluacije često kompleksan, skup i vremenski iscrpan. Drugi način procjene modela jest intrinzična evaluacija i ona se oslanja na različite metrike kojima je moguće jednostavnije i brže izraziti i usporediti učinkovitost pojedinih jezičnih modela.

Intrinzična evaluacija jezičnog modela zahtjeva postojanje testnog seta, skupa podataka nad kojim će model testirati svoju učinkovitost. Važno je da jezični model bude neovisan od testnog seta, to jest da nema uvid u njegov vokabular i strukturu.

2.3.1 Perpleksnost

Perpleksnost (engl. *perplexity*) u području jezičnog modeliranja predstavlja glavno mjerilo kojim se određuje sposobnost jezičnog modela za procjenu vjerojatnosti nekog testnog seta.[3] Označava se oznakom PP i računa se kao inverz vjerojatnosti testnog skupa normaliziranog brojem riječi. Za testni skup definiran s $W = w_1 w_2 ... w_N$, perleksnost skupa je:

$$PP(W) = P(w_1 w_2 ... w_N)^{-\frac{1}{N}} = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 w_2 ... w_{i-1})}} \tag{2.6}$$

Stoga, kod primjerice bigram modela, perpleksnost skupa W jednaka je:

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}} \tag{2.7}$$

Iz jednadžbi 2.6 i 2.7 može se zaključiti da će perpleksnost nekog niza biti manja, što je veća njegova vjerojatnost. Iz toga proizlazi da minimiziranje perpleksnosti nad nekim testnim setom dovodi do intrinzičnog poboljšanja kvalitete jezičnog modela.

Iako intrinzična evaluacija na temelju perpleksnosti ne uzima u obzir specifičnu namjenu jezičnog modela, korisna je u svrhu usporedbe različitih jezičnih modela, te u pravilu korelira s krajnjom (ekstrinzičnom) kvalitetom proizvoda nastalog na temelju korištenja tih modela.[3]

Poglavlje 3

Konfiguracija i korišteni alati

Prilikom izgradnje jezičnog modela korišteni su dolje navedeni alati i servisi. Zbog bolje kompatibilnosti s nekim od alata korišten je Ubuntu operacijski sustav unutar Oracle VM VirtualBox okruženja. Sustav sadrži 16GB radne memorije, Intel i5-1135G7 procesor, Nvidia GeForce MX350 grafičku karticu te Kingston NVMe SSD disk.

3.1 SRI Language Modeling Toolkit

SRI Language Modeling Toolkit jest skup alata sastavljenih od C++ knjižnica, izvršnih datoteka i pomoćnih skripti namjenjenih izgradnji i testiranju različitih statističkih jezičnih modela. Razvija ga SRI Speech Technology and Research Laboratory od 1995. godine. Nastao je iz potrebe za efikasnim, fleksibilnim i nadogradivim programskim rješenjem u području jezičnog modeliranja. Alat je besplatan za korištenje u nekomercijalne svrhe i često se koristi u modelima za prepoznavanje govora, strojno prevođenje i statističko označavanje.

Sa službene internetske stranice preuzeta je verzija alata 1.7.3 pod *Research Community* licencom koja omogućuje besplatno korištenje alata u nekomercijalne svrhe.

3.1.1 C++

C++ je programski jezik srednje razine i opće namjene kojeg je 1985. u sklopu Bell Labs nezavisnog laboratorija stvorio danski programer Bjarne Stroustrup kao proširenje C programskom jeziku. Prvotno se zvao "C sa klasama" zbog svoje podrške za objektno orijentirano programiranje. C++ je kompajlirani jezik te je poznat po svojim performansama, učinkovitosti i fleksibilnosti. Zbog toga se često koristi kod programiranja sustava s ograničenim resursima, ugradbenih sustava, servera, video igara, čak i svemirskih sondi. Jezik je od 1998. godine normiran ISO standardom.

3.2 Wikipedija

Wikipedija je najveća i najčitanija višejezična besplatna internetska enciklopedija i jedna od najposjećenijih stranica na internetu. Pokrenuli su je Jimmy Wales i Larry Sanger 2001. godine. Danas ju vodi neprofitna organizacija zaklada Wikimedija. Koristi wiki uređivački sustav i njen sadržaj može uređivati svatko, a provjeru ispravnosti i ispravljanje sadržaja provodi zajednica volontera. Sveukupno Wikipedija postoji na 326 različita svjetska jezika te sadrži više od 15 milijuna članaka, od kojih je 5 milijuna na engleskom jeziku. [5]

Hvaljena je zbog obujma informacija koje sadrži i njihovoj dostupnosti, kojom omogućuje prikupljanje i širenje znanja u široj populaciji. Međutim, često je i na meti kritika zbog upitne točnosti i kvalitete tih informacija, nepotkrijepljenih stavova te sustavne pristranosti u određenim područjima. Posljednjih godina se i u tim aspektima poboljšala te prima generalno pozitivne kritike.

Hrvatska Wikipedija objavljena je u veljači 2003. godine, a u trenutku pisanja ovog rada sadrži više od 200 tisuća članaka, koje sveukupno sadrže više od 60 milijuna riječi. [6]

3.3 Python

Python je interpretirani programski jezik visoke razine stvoren 1991. godine. Dopušta objektno orijentirano, strukturno te aspektno orijentirano programiranje. Njegova fleksibilnost, jednostavnost i sveobuhvatnost čini ga jednim od najpopularnijih programskih jezika današnjice.

Česta kritika Pythonu je njegova brzina, smatra se sporijim jezikom upravo zbog toga što je interpretiran, a ne kompjaliran kao na primjer C++ ili C. Međutim, zbog velikog broja knjižnica i alata napisanih za njega, vrlo je koristan u polju prirodne obrade jezika.

Pythonov sustav za upravljanje knjižnicama naziva se *pip*. On omogućava jednostavno dodavanje i uklanjanje potrebnih knjižnica, a pristupa im kroz *Python Package Index* (PyPI).

3.3.1 WikiExtractor

WikiExtractor je programski alat namjenjen ekstrakciji i pročišćivanju teksta iz Wikipedijinih *dump* datoteka. Napisan je u Pythonu, a kao ulazni parametar prima *dump* datoteku s nastavkom `.xml.bz2`. Tijela članaka zapisuje unutar `<doc></doc>` oznaka i sprema kao tekstualne datoteke specificirane veličine. Instalira se naredbom:

```
pip install wikiextractor
```

3.3.2 Natural Language Toolkit

Natural Language Toolkit ili skraćeno NLTK je skup Python knjižnica i izvršnih programa namjenjenih statističkom i simboličkom jezičnom modeliranju. Izradili su ga Steven Bird i Edward Loper na sveučilištu u Pennsylavniji 2001. godine. Koristi se za obradu tekstualnih podataka kod pripreme korpusa za jezično modeliranje. Instalacija se provodi naredbom:

```
pip install nltk
```

Poglavlje 4

Priprema jezičnog korpusa

Prvi korak pri gradnji jezičnog modela jest prikupljanje i obrada podataka na kojima će model biti treniran. Takvi tekstualni podaci nazivaju se jezični korpus i mogu se sastojati od različitih književnih i znanstvenih djela, članaka, izvještaja ili dokumenata na određenom jeziku. Jezični korpus mora biti u obliku u kojem će ga alat za jezično modeliranje moći pravilno obraditi. Korpus se najčešće mora nalaziti unutar jedne tekstualne datoteke, sadržavati jednu rečenicu u svakom redu te biti oslobođen svih interpunkcijskih znakova i brojeva.

4.1 Preuzimanje članaka Wikipedije

U ovom radu kao jezični korpus korišteni su članci hrvatske Wikipedije. Na internetskoj stranici `https://dumps.wikimedia.org/hrwiki` dostupne su preslike cjelokupnog sadržaja hrvatske Wikipedije u obliku pogodnom za preuzimanje i obradu. Preslike se ažuriraju jednom do dvaput mjesečno, a osim samih članaka, omogućeno je preuzimanje slika, baza podataka i meta podataka. Preuzimanja su odovojena ovisno o jeziku sadržaja. Za potrebe rada preuzeta je datoteka

`hrwiki-latest-pages-articles-multistream.xml.bz2`

koja sadrži tekst svih članaka hrvatske Wikipedije u XML formatu komprimirane bz2 standardom. Veličina preuzete komprimirane datoteke iznosila je 306.1 MB.

4.2 Ekstrakcija teksta

Nakon preuzimanja datoteke potrebno ju je iz komprimiranog XML formata pretvoriti u tekstualni (txt) format. To je učinjeno koristeći Python skriptu WikiExtractor. Na slici (4.1) je prikazana naredba kojom je alat pozvan unutar naredbene linije.

```
$ python3 WikiExtractor.py ../hrwiki-latest-pages-articles-multistream.xml.bz2 --processes 8 -q -o Wiki
```

Slika 4.1 Korištenje WikiExtractora

Pri pozivanju alata dodan je parametar `--processes` koji omogućava višenitno izvršavanje i ubrzava izvođenje programa, te je postavljen na 8 zbog broja jezgara procesora. Zatim je dodan parametar `-q` koji uklanja prikaz napretka, te parametar `-o` za specificiranje izlaza programa.

Pokretanjem naredbe, WikiExtractor obrađuje komprimiranu datoteku, pretvara ju u tekst te se podaci zapisuju u mapu proizvoljnog naziva, u ovom slučaju Wiki. Unutar mape nastaju podmape, od kojih svaka sadržava 100 tekstualnih datoteka s nazivima od `wiki_00.txt` do `wiki_99.txt`. Pri završetku izvođenja nastaju mape s nazivima AA, AB, AC i AD, a sveukupno sadržavaju 318 tekstualnih datoteka. Za treniranje modela korištene su prve tri navedene mape, a kao kontrolni testni set uzeta je posljednja mapa AD koja sadrži 18 tekstualnih datoteka, koje sadrže približno 5% ukupnog teksta.

4.3 Pročišćivanje i formatiranje teksta

Nakon što je dobiven sadržaj Wikipedije u obliku tekstualnih datoteka, potrebno je podatke pročistiti, ukloniti nepotrebne znakove i brojeve, razdvojiti rečenice u posebne redove i sve rečenice upisati u jednu txt datoteku. Napisana je Python skripta *clean.py* koja vrši navedene funkcije.

Na početku skripte učitane su potrebne knjižnice (slika 4.2). Knjižnica `string` dio je Pythonove standardne knjižnice i služi za manipulaciju znakovnih nizova. Knjiž-

nica `re` također je ugrađena knjižnica koja omogućuje korištenje regularnih izraza (engl. *regular expression* - RegEx). Regularni izraz je niz znakova kojim se stvara uzorak pretraživanja. Iz modula NLTK opisanog u potpoglavlju 3.3.2 učitana je metoda `sent_tokenize` koja služi rastavljanju teksta u rečenice. Naposljetku su učitane knjižnice `os` i `fnmatch` koje pomažu u pretraživanju datoteka unutar mape.

```
import string
import re
from nltk.tokenize import sent_tokenize
import os
import fnmatch
```

Slika 4.2 Uvoz korištenih modula

Nakon toga, definirane su funkcije za rad s datotekama (slika 4.3). Funkcije koriste ugrađene metode `open()`, `read()`, `write()` i `close()`. Funkcija `load_file()` otvora datoteke i vraća njihov sadržaj, a funkcija `save_file()` iterira kroz rečenice te ih zapisuje u zasebne redove nove tekstualne datoteke.

```
# Učitavanje sadržaja datoteke
def load_file(filename):
    file = open(filename, 'r')
    text = file.read()
    file.close()
    return text

# Spremanje teksta u datoteku
def save_file(lines, filename):
    file = open(filename, 'a')
    for line in lines:
        file.write("%s\n" % line)
    file.close()
```

Slika 4.3 Funkcije za rad s datotekama

U glavnu funkciju (slika 4.4) ulazi tekst učitane datoteke te se obrađuje. Prvo se

iz teksta uklanjaju sve HTML oznake postavljene od strane WikiExtractor-a i one koje su već postojale u člancima. Zatim se pomoću NLTK metode tekst segmentira na rečenice te nastaje niz rečenica. Nadalje, svaka se rečenica rastavlja na riječi (tokene) i ukoliko je rečenica kraća od 4 riječi zanemaruje se. Razlog tome jest to što rečenice kraće od 4 riječi najčešće nastaju kao posljedica pogrešnog rastavljanja na rečenice kod prethodnog koraka, stoga ih je najbolje ukloniti. Rečenicama koje zadovoljavaju provjeru uklanja se sva interpunkcija, uklanjaju se svi brojevi i ostali znakovi te se sva slova postavljaju kao mala. Formatirane rečenice dodaju se novoj listi te se pri završetku izvođenja vraćaju u glavni program.

```
# Pročišćavanje i formatiranje teksta
def prepare_corpus(text):
    # Uklanjanje HTML oznaka
    CLEANR = re.compile('<.*?>')
    text = re.sub(CLEANR, '', text)
    # Razdvajanje na rečenice
    sentences = sent_tokenize(text)
    # Inicijalizacija prazne liste
    clean = []
    for sentence in sentences:
        # Razdvajanje rečenice na riječi
        tokens = sentence.split()
        # Provjera duljine rečenice
        if (len(tokens) <= 4): continue
        # Uklanjanje interpunkcija
        table = str.maketrans('', '', string.punctuation)
        tokens = [w.translate(table) for w in tokens]
        # Uklanjanje svih znakova koji nisu slova
        tokens = [word for word in tokens if word.isalpha()]
        # Pretvaranje u mala slova
        tokens = [word.lower() for word in tokens]
        # Dodavanje rečenice u novu listu
        sentence = ' '.join(tokens)
        clean.append(sentence)
    return clean
```

Slika 4.4 Funkcija za pročišćivanje i formatiranje teksta

U glavnom dijelu skripte (slika 4.5) izvodi se *for* petlja koja prolazi mapom Wiki i dohvaća nazive svih datoteka. Svaka se datoteka otvara, obrađuje i njen sadržaj se zapisuje u datoteku naziva *corpus.txt*. Pri završetku izvođenja petlje datoteka *corpus.txt* predstavlja pripremljen jezični korpus i spremna je za korištenje pri izgradnji modela. Na slici (4.6) vidljiv je isječak pripremljenog korpusa, a u tablici (4.1) vidljiva je detaljnija statistika nastalog jezičnog korpusa.

```
for path,dirs,files in os.walk('Wiki'):
    for file in files:
        if fnmatch.fnmatch(file, '*'):
            fullname = os.path.join(path, file)
            text = load_file(fullname)
            corpus = prepare_corpus(text)
            save_file(corpus, 'corpus.txt')
```

Slika 4.5 Poziv glavnih funkcija

Tablica 4.1 Detalji jezičnog korpusa

Broj riječi	Broj rečenica	Veličina (MB)
2 431 052	40 524 406	259

```
tada dolazi i svjetski rat radi kojeg se tvornica prenamijenjuje za proizvodnju granata i vojnih vozila
nakon rata došlo je do pada monarhije a s njim je došla toliko priželjkivana češka nezavisnost
međutim bila je jako loša gospodarska situacija
propala su stara tržišta i trebalo je krenut iznova
ni domaće tržište nije blistalo
zavladalo je siromaštvo i zbog nestašice goriva automobili su bili jako rijedak prizor na cesti
do tvrtka preživljava proizvodnjom motornih plugova ali te godine stvari kreću na bolje
s novim gospodarskim zamahom rađaju se i novi modeli poput tatra praga alfa tip te njegove izvedenice i
laurinklement je očajno trebao novitete da ojača i postane sposoban slijediti svjetske automobilističke trendove
tada se pojavljuje konzorcij škoda koji uzima laurinklement pod svoje okrilje
nakon godina proizvodnje kamiona blindiranih vozila avio motora i slično počinje njihov ulazak u svijet automobila
tada su potpisali ugovor s hispanosuizom za proizvodnju i prodaju modela
škoda počinje iskorištavati laurinklement i u kratkom vremenu na tržištu se nalazi cijela paleta osobnih vozila s potpisom škode
nakon toga zbiva se spajanja te poduzeće laurinklement biva trajno izbrisano iz trgovačkog registra
godine i pojavljuju se škoda i s i cilindara
nakon toga izlazi i model s velikim osmerocilindričnim motorom obujma litara koji se razvijao u najvećoj tajnosti
sve je išlo dobro dok vlada nije povećala porez na aute što je imalo katastrofalne posljedice
prodaja novih automobila je bila prepolovljena a i veliki dio postojećih automobila povučen je iz prometa
zbog toga škoda konstruira mali auto koji je predstavljen godine pod imenom škoda standard
pokretao ga je četverocilindraš snage konja
zatim izlazi popular koji je bio lakši i jeftiniji a samim tim i popularniji kao što mu i ime govori
osim populara ponudu nadopunjuje i jedan rapid te luksuzni superb koji simbolizira novi uspon škode te početak značajnog prodora na svjetsko tržište
sredinom tridesetih škoda počinje eksperimentirati s novim oblicima i aerodinamikom
iz toga nastaje model s motorom obujma ccm i zračnim hlađenjem koje je smješten u stražnjem dijelu
taj model po mnogim detaljima podsjeća na kasnije nastalu vw bubu
početkom drugog svjetskog rata škoda doživljava katastrofalno bombardiranje
no vrlo brzo se uspješno oporavila
međutim završetkom rata nova politička situacija otežava nastavak prijeratne proizvodnje tako da škoda započinje najprije s proizvodnjom lakih kamiona
ubrzo izlazi redizajnirani superb na istoj platformi kao starija verzija
kasnije izlazi model koji vraća škodi nekadašnju slavu na domaćem i na stranom tržištu
```

Slika 4.6 Isječak pripremljenog korpusa

Poglavlje 5

Izgradnja jezičnog modela

5.1 Prebrojavanje N-grama

Nakon što je korpus pripremljen za jezično modeliranje, sljedeći korak jest prebrojavanje N-grama i izgradnja modela. Naredba `ngram-count` unutar SRILM toolkit-a služi upravo za manipulaciju i prebrojavanje N-grama te određivanje jezičnog modela.[7] Naredba iščitava tekstualnu datoteku, provodi izgradnju modela te ga zapisuje u datoteku posebnog formata. Korištena naredba vidljiva je na slici (5.1).

```
ngram-count -text corpus.txt -order 4 -lm model.bo
```

Slika 5.1 Naredba za izgradnju jezičnog modela

Naredba se sastoji od sljedećih dijelova:

- **ngram-count** je metoda kojom se generira i aproksimira jezični model, prebrojava različite N-grame i bilježi njihove vjerojatnosti
- **-text** je parametar kojim se zadaje tekstualna datoteka nad kojom se trenira model

- **-order** zadaje maksimalnu duljinu N-grama koji će se uzeti u obzir, u ovom slučaju maksimalna duljina je 4, tj. kvadrigram
- **-lm** pridružuje naziv datoteke koja će sadržavati jezični model

5.2 Format modela

Nakon završetka izvođenja operacije nastaje datoteka *model* koja sadržava izgrađeni jezični model, ukupne veličine 600.4 MB.

Format u kojem SRILM toolkit zapisuje N-gram jezični model naziva se ARPA ili Doug Paul format. Prema navedenom formatu, u zaglavlju datoteke zapisan je broj pronađenih jedinstvenih N-grama ovisno o njihovoj duljini:

```
    \data\
ngram 1=1179882
ngram 2=12768734
ngram 3=3592454
ngram 4=2440956
```

U nastavku datoteke nalazi se popis pojedinačnih instanci N-grama, grupiranih po duljini. Svaka grupa počinje oznakon `\N-gram`, gdje je N red te skupine N-grama. Svaki N-gram započinje brojem koji predstavlja njegovu vjerojatnost, zapisanu u obliku logaritma s bazom 10. Stoga u primjeru bigrama:

```
-0.6596466 igrom slučaja 0.02577335
```

vjerojatnost pojavljivanja kombinacije riječi igrom i slučaja iznosi $10^{-0.6596466} = 0.219$. Broj koji se pojavljuje na kraju određenih N-grama predstavlja logaritamsku težinu ustručavanja (engl. *backoff*) pojedinih N-grama.[8]

SRILM toolkit tijekom izgradnje modela na početak svake rečenice dodaje simbol `<s>`, a na kraj rečenice `</s>`. Koristeći ove simbole omogućeno je zapisivanje vjerojatnosti za pojavljivanje riječi na početku ili kraju rečenice. Na isječku izgrađenog modela na slici (5.2) vidljivi su primjeri bigrama koji sadrže navedeni simbol za početak rečenice.

```
-6.372521       <s> prometovao  -0.1225692
-5.455183       <s> prometuje   -0.1198264
-6.981715       <s> prometuju
-5.631443       <s> promicala   -0.617424
-6.372521       <s> promicale   -0.201791
-6.090884       <s> promicali   -0.2986099
-6.981715       <s> promican
-6.981715       <s> promicani
-5.455183       <s> promicanje
-6.090884       <s> promicanjem -0.1430394
-6.981715       <s> promicanju
-4.877947       <s> promicao    -0.9224701
-6.090884       <s> promicatelj
-6.981715       <s> promicateljem
-6.981715       <s> promicati
-5.707317       <s> promidžba   -0.2324237
-5.707317       <s> promidžbena 0.03805817
-6.372521       <s> promidžbeni
-6.981715       <s> promijena
-6.090884       <s> promijene
-6.372521       <s> promijeni   -0.2485416
-5.178977       <s> promijenila -0.6742473
-6.981715       <s> promijenile
-5.330245       <s> promijenili -0.9860175
-5.803234       <s> promijenilo -0.1835754
-5.455183       <s> promijenimo -0.3384155
-4.756213       <s> promijenio  -0.7847333
-6.372521       <s> promijenit  1.048827
-6.981715       <s> promijenite
-6.090884       <s> promijeniti
-6.372521       <s> promijenivši
```

Slika 5.2 Isječak bigrama unutar izgrađenog modela

5.3 Usporedba s ostalim modelima

Jedan od najvećih N-gram jezičnih modela jest Google-ov N-gram model. Korpus nad kojim je izgrađen sadrži više od jednog trilijuna riječi i preko 95 milijardi rečenica prikupljenih sa javno dostupnih internetskih stranica. Koristio se u svrhu statističkog strojnog prevođenja, ispravka pravopisnih pogrešaka, prepoznavanja govora, i sl. [9]

Najveći hrvatski jezični N-gram model naziva se Hascheck (Hašek), što je skraćenica za Hrvatski akademski spelling checker. On se od 1994. godine pojavljuje u raznim servisima za pravopisnu provjeru hrvatskih tekstova te je kao takav jedna od prvih internetskih usluga u Hrvatskoj. Hašekov korpus sadrži više od 10 milijardi riječi, s rječnikom od preko 2 milijuna različitih riječi. Trenutno služi kao jezgra alatu za pravopisnu provjeru na adresi `https://ispravi.me/`.[10], [11]

Ukoliko usporedimo broj N-grama nastalih tijekom izgradnje jezičnog modela hrvatske Wikipedije i ostala dva navedena N-gram modela, vidljivo je da ovaj model sadrži usporedivo manji broj N-grama po svim razinama.

U usporedbi s Google-ovim engleskim modelom koji sadrži preko tri milijarde instanci N-grama, naš model sadrži samo nešto manje od 20 milijuna, što je približno 0.5% od prethodno navedenog modela. U usporedbi s Hascheck modelom, ovaj model sadrži 0.7% broja njegovih instanci. Također, dok ostalim modelima povećanjem razine N-grama raste i broj instanci, ovaj model sadrži najveći broj bigrama, zatim trigrama i kvadrigrama, a najmanje unigrama.

Tablica 5.1 Usporedba broja instanci različitih modela ovisno o redu N-grama

Red N-grama	Google-ov model	Hascheck	Model hrv. Wikipedije
1-gram	13 588 391	5 227 896	1 179 882
2-gram	314 843 401	195 683 360	12 768 734
3-gram	977 069 902	638 624 784	3 592 454
4-gram	1 313 818 354	929 492 670	2 440 956
Ukupno	3 795 790 711	2 725 794 871	19 982 026

Poglavlje 6

Evaluacija jezičnog modela

Za evaluaciju modela u alatu SRI Language Modeling Toolkit koristi se naredba `ngram`. Ona služi za vršenje više različitih operacija nad jezičnim modelom, a osim računanja perpleksnosti, neke od njih su ocjenjivanje vjerojatnosti rečenica, interpolacije modela i generiranje rečenica pomoću N-gram modela.[12] Na slici (6.1) je vidljiva navedena naredba kojom se unutar terminala provodi evaluacija nad zadanim testnim setom.

```
ngram -order 4 -lm model -ppl testset.txt
```

Slika 6.1 Naredba za evaluaciju modela

Pojašnjenje pojedinih parametara:

- **-order** parametar određuje najveći red N-grama koji će se koristiti pri evaluaciji
- **-lm** definira naziv datoteke unutar koje se nalazi model u ARPA formatu
- **-ppl** pokreće evaluaciju nad rečenicama unutar datoteke koja je pridružena parametru

6.1 Kontrolni testni set

Dio teksta koji je izdvojen pri stvaranju korpusa i koji nije korišten pri treniranju modela, uzet je kao kontrolni testni set za provjeru perpleksnosti modela. Tekst je pročišćen i spremljen u datoteku naziva `testni-test-kontrola.txt` i naredbom `ngram` provedena je evaulacija modela nad testnim setom. Rezultat evaluacije prikazan je na slici (6.2).

```
file testset.txt: 136016 sentences, 2334830 words, 54109 OOVs
0 zeroprobs, logprob= -7081948 ppl= 851.8752 ppl1= 1273.901
```

Slika 6.2 Perpleksnost kontrolnog testnog seta

Izlaz govori da datoteka sadrži 136016 rečenica i 2334830 riječi, od kojih su 54109 riječi izvan vokabulara (OOV - *out-of-vocabulary*), to jest pojavljuju se u testnom setu, ali se ne nalaze unutar korpusa. Zahvaljujući metodama zaglađivanja, nema pronađenih vjerojatnosti s vrijednošću nula (*zeroprobs*). Logaritamska vjerojatnost seta iznosi -7081948, a perpleksnost (ppl) je 851,88. Vrijednost ppl1 predstavlja perpleksnost bez uzimanja u obzir oznaka za kraj rečenice te iznosi 1273,9.

Za usporedbu, također je provedena evaluacija modela nad samim korpusom. U tom slučaju, perpleksnost je iznosila niskih 137,5. To govori da model daje visoku vjerojatnost pojavljivanja kombinacijama riječi koje se nalaze u korpusu, što je za pretpostaviti budući da je sam model nastao na temelju danog korpusa.

6.2 Evaluacija nad nizovima hrvatskih riječi

Za određivanje učinkovitosti i točnosti modela pri određivanju vjerojatnosti hrvatskih riječi u konkretnom kontekstu, korištene su transkripcije vremenskih prognoza državnog hidrometeorološkog zavoda. Preuzeto je 10 rečenica različite duljine, stila i složenosti te su formatirane i smještene u datoteku `testni-set-vrijeme.txt`. Ponovno je provedena evaluacija metodom `ngram` nad datotekom (slika 6.3).

Na izlazu je vidljivo kako ovih 10 rečenica sadrži ukupno 104 riječi te je samo jedna od njih riječ izvan vokabulara modela. Također ne postoje *zeroprobs* vrijednost, a logaritamska vjerojatnost za testni skup iznosi -363,64. Perpleksnost je veća nego za kontrolni skup i iznosi 1652,24.

```
file testni-set-vrijeme.txt: 10 sentences, 104 words, 1 OOVs
0 zeroprobs, logprob= -363.6422 ppl= 1652.239 ppl1= 3392.4
```

Slika 6.3 Perpleksnost testnih rečenica vremenske prognoze

Dodavanjem parametra `-debug 1` metodi `ngram` alat ispisuje detaljne vjerojatnosti za svaku pojedinu rečenicu. Za rečenicu *Na jadranu će biometeorološki uvjeti i dalje biti povoljni*, na slici (6.4) vidljivo je da je riječ *biometeorološki* izvan vokabulara modela i stoga je zamjenjena oznakom `<unk>` (*unknown*).

```
na jadranu će biometeorološki uvjeti i dalje biti povoljni
    p( na | <s> )   = [2gram] 0.02107007 [ -1.676334 ]
    p( jadranu | na ...)    = [3gram] 0.0002733255 [ -3.56332 ]
    p( će | jadranu ...)    = [1gram] 0.0001487476 [ -3.82755 ]
    p( <unk> | će ...)  = [OOV] 0 [ -inf ]
    p( uvjeti | <unk> ...)  = [1gram] 2.984491e-05 [ -4.52513 ]
    p( i | uvjeti ...)  = [2gram] 0.0343214 [ -1.464435 ]
    p( dalje | i ...)   = [2gram] 0.002961943 [ -2.528423 ]
    p( biti | dalje ...)    = [3gram] 0.003787879 [ -2.421604 ]
    p( povoljni | biti ...) = [2gram] 8.23192e-06 [ -5.084499 ]
    p( </s> | povoljni ...) = [2gram] 0.08910889 [ -1.050079 ]
1 sentences, 9 words, 1 OOVs
0 zeroprobs, logprob= -26.14137 ppl= 802.781 ppl1= 1852.131
```

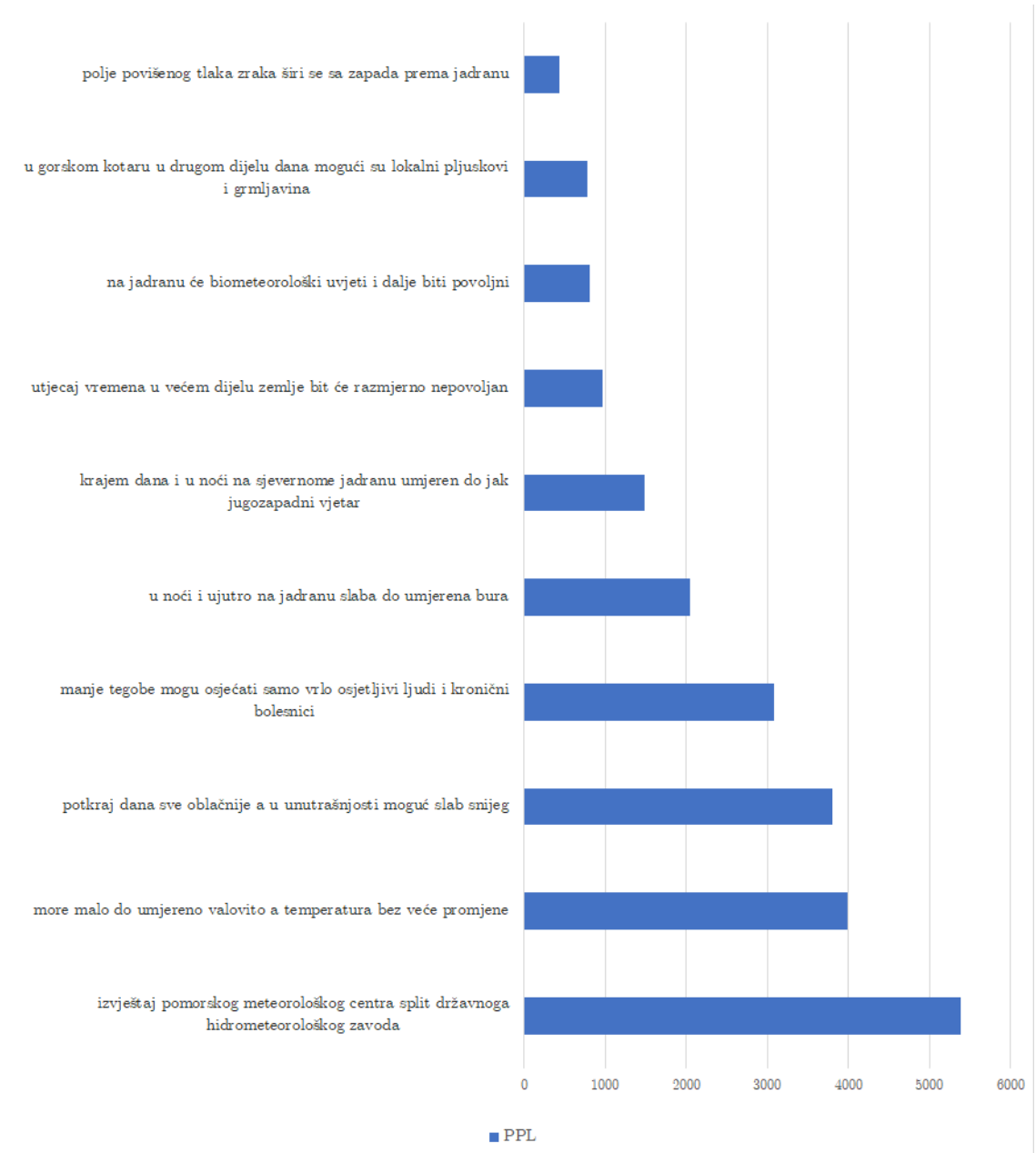
Slika 6.4 Detaljni prikaz vjerojatnosti za rečenicu testnog seta

6.3 Usporedba rezultata

Dobiveni rezultati perpleksnosti za pojedine nizove riječi transkripcija vremenske prognoze zapisani su uzlazno i prikazani na grafu na slici 6.5. Najniža perpleksnost

predstavlja najveću vjerojatnost koju model daje nizu riječi, a visoka perpleksnost predstavlja malu vjerojatnost pojave niza riječi.

Na grafu je primjerice vidljivo da rečenica *polje povišenog tlaka zraka širi se sa zapada prema jadranu* ima najmanju perpleksnost, to jest najveću vjerojatnost, a rečenica *izvještaj pomorskog meteorološkog centra split državnog hidrometeorološkog zavoda* najveću perpleksnost, to jest najmanju vjerojatnost. Iako su obje rečenice pravopisno i gramatički ispravne, razlog manje perpleksnosti prve rečenice je taj što je vjerojatnost bigrama *tlaka zraka* i *širi se* puno veća i češće se pojavljuje u korpusu Wikipedije nego primjerice *split državnog* i *pomorskog meteorološkog*. Moguće je također zaključiti da model favorizira nizove riječi povezane korištenjem više veznika i prijedloga, što je vidljivo u rečenici *u gorskom kotaru u drugom dijelu dana mogući su lokalni pljuskovi i grmljavina*, koja također ima vrlo nisku perpleksnost.



Slika 6.5 Usporedba perpleksnosti pojedinih rečenica vremenske prognoze

6.4 Prepoznavanje pogrešaka u tekstu

Iz transkripcija vremenske prognoze preuzeto je još nekoliko nizova riječi te su im dodane različite pravopisne i gramatičke pogreške. Ispravnim i pogrešnim nizovima određena je perpleksnost korištenjem modela Wikipedije. U tablici 6.1 prikazani su rezultati evaluacije.

Tablica 6.1 Primjeri ispravnih i pogrešnih nizova riječi i njihova perpleksnost

Niz riječi	Perpleksnost
tegobe bi mogli imati	1054.78
tegobe **bih** mogli imati	**4264.39**
najviša temperatura	154.08
najviša **temperature**	**2143.98**
u gorskom kotaru i lici	26.83
u gorskom kotaru i **licu**	**174.58**
na cijelom jadranu	98.12
na **cjelom** jadranu	**5884.74**
vjetar slab u noći	739.61
vjetar slab u **noči**	**4168.70**

U svakom redu tablice nalaze se po dva niza riječi gdje je na vrhu ispravan niz, a ispod njega niz s pogreškom. Pogreška je podebljana te je podebljana perpleksnost niza s pogreškom u svakom redu. Na početku je vidljivo kako je model ispravno odredio ispravan oblik glagola biti, odredivši pogrešnom obliku 4 puta veću perpleksnost nego ispravnom. Nadalje, niz s točnim padežom riječi *temperatura* ima 15 puta veću vjerojatnost od pogrešnog. U skupu riječi *u gorskom kotaru i lici*, model je iz konteksta utvrdio da je riječ *licu* manje vjerojatna od riječi *lici*. Model je također s velikom uspješnošću odredio točno korištenje skupova *ije* i *je*, te slova *č* i *ć*.

6.5 Generiranje rečenica

Naposljetku, jedna od funkcija SRILM toolkita jest i slaganje rečenica na temelju izgrađenog jezičnog modela. Korištenjem `-gen n` parametra u metodi `ngram` moguće je generirati `n` rečenica na temelju istreniranog modela. Rečenice su nasumične duljine, te je moguće odrediti kojom će rječju počinjati.

Na temelju modela hrvatske Wikipedije kreirano je nekoliko rečenica, a neke od izdvojenih su:

- "lipovec je naselje u slovenskoj općini metliki"
- "u prizemlju su zarobljeni u kini"
- "uradak je ponovno oživio pod vodstvom cara trajana"
- "bez novca pavlinima na otoku korčuli četiri je dana slavio uz cetinu"
- "u hrvatskoj postoji pet ljudi grozdova naseljeno danas u općoj populaciji"

Poglavlje 7

Zaključak

Cilj ovog završnog rada bio je izgraditi N-gram jezični model na temelju članaka hrvatske Wikipedije, ocijeniti njegovu kompleksnost te njime provesti evaluaciju nad nizovima hrvatskih riječi. Za tu primjenu korišten je C++ alat SRI Language Modeling Toolkit.

Kroz rad su opisani stohastički jezični modeli te njihova primjena u svakodnevnom životu. Koristeći Python pripremljen je korpus sačinjen od cjelokupnog sadržaja hrvatske Wikipedije te je model izgrađen metodama SRILM toolkit-a. Naposlijetku je provedena evaluacija modela nad nizovima hrvatskih riječi preuzetih iz transkripcija vremenskih prognoza.

Model se pokazao manjim od ostalih javno dostupnih N-gramskih jezičnih modela, međutim evaluacijom nad nizovima riječi pokazao se dovoljno kompleksnim za učinkovito raspoznavanje pravilno složenih rečenica, prepoznavanje pravopisnih i gramatičkih pogrešaka, te generiranje dosljednih rečenica.

Bibliografija

[1] S. Šuman, "Pregled metoda obrade prirodnih jezika i strojnog prevođenja," *Zbornik Veleučilišta u Rijeci*, vol. 9, br. 1, str. 371-384, Rijeka, 2021.

[2] C. Mandery, "Distributed n-gram language models: Application of large models to automatic speech recognition," Karlsruhe Institute of Technology, Karlsruhe, 2011.

[3] D. Jurafsky and J. Martin, "Speech and language processing," Pearson Prentice Hall, New Jersey, 2000.

[4] A. Stolcke, "Srilm — an extensible language modeling toolkit," *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, vol. 2, pp. 901–904, Menlo Park, 2004.

[5] Wikipedia. , s Interneta, https://en.wikipedia.org/wiki/Wikipedia , 4. lipnja 2022.

[6] Wikipedija: Statistika. , s Interneta, https://hr.wikipedia.org/wiki/Posebno:Statistika , 4. lipnja 2022.

[7] A. Stolcke, J. Zheng, and T. Alumae. Srilm toolkit manual pages - ngram-count. , s Interneta, http://www.speech.sri.com/projects/srilm/manpages/ngram.1.html , 8. lipnja 2022.

[8] A. Stolcke. Srilm toolkit manual pages - ngram-format. , s Interneta, http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html , 8. lipnja 2022.

[9] A. Franz and T. Brants. All our n-gram are belong to you. , s Interneta, https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html , 18. lipnja 2022.

[10] S. Dembitz. Strojna obrada hrvatskog jezika. , s Interneta, https://www.matica.hr/kolo/539/strojna-obrada-hrvatskog-jezika-maarski-doprinosi-27748/ , 18. lipnja 2022.

[11] S. Dembitz. O usluzi ispravi.me. , s Interneta, https://ispravi.me/info/ , 18. lipnja 2022.

[12] A. Stolcke, J. Zheng, and T. Alumae. Srilm toolkit manual pages - ngram. , s Interneta, http://www.speech.sri.com/projects/srilm/manpages/ngram.1.html , 8. lipnja 2022.

Sažetak

Tema završnog rada jest izgradnja N-gram jezičnog modela na temelju korpusa hrvatske Wikipedije. U prvom dijelu rada razrađeno je statističko jezično modeliranje. Zatim je provedeno preuzimanje i ekstrakcija teksta iz članaka hrvatske Wikipedije. Naposljetku je model izgrađen alatom SRILM toolkit i njime je provedena evaluacija različitih skupova hrvatskih riječi.

Ključne riječi **— jezično modeliranje, Wikipedija, SRILM toolkit, N-gram**

Abstract

The topic of this thesis is building a N-gram language model based on a corpus extracted from the Croatian Wikipedia. The first part of the paper describes statistical language modeling. After that it covers the gathering and extraction of text from articles of Croatian Wikipedia. Finally, the model is built using SRILM toolkit and is evaluated based on sequences of Croatian words.

Keywords **— language modeling, Wikipedia, SRILM toolkit, N-gram**