

Uvod u znanost o podacima

Uvod u obradu teksta

Prof. dr. sc. Mile Šikić

11. predavanje, 11. siječnja 2022.

ak. god. 2021./2022.

Čime smo se do sada bavili...

What are the most important statistical ideas of the past 50 years?*

Andrew Gelman[†] and Aki Vehtari[‡]

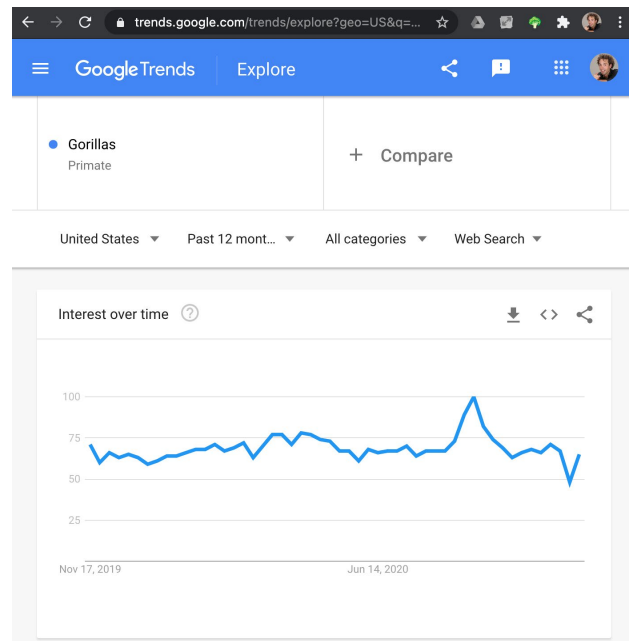
3 June 2021

Abstract

We review the most important statistical ideas of the past half century, which we categorize as: counterfactual causal inference, bootstrapping and simulation-based inference, overparameterized models and regularization, Bayesian multilevel models, generic computation algorithms, adaptive decision analysis, robust inference, and exploratory data analysis. We discuss key contributions in these subfields, how they relate to modern computing and big data, and how they might be developed and extended in future decades. The goal of this article is to provoke thought and discussion regarding the larger themes of research in statistics and data science.

Tekstualni podaci

- Većina suvremenih podataka je **nestrukturirani tekst**
 - Web
 - Društvene mreže
 - Vijesti
- Često, “čisti” skupovi podataka mogu se proizvesti od “prljavih” tekstualnih podataka
 - npr. upiti pretrage su kratki tekstovi; Google Trends vremenska serije za pojmove (npr. [Q36611 Gorilla](#)) su dobiveni agregiranjem svih upita koji se na referiraju na pojam (npr. “gorilla”, “big black Rwandan apes”, “are gorillas humans?”)



Pregled


- 4 tipična zadatka na tekstualnim podacima
 - Dohvaćanje dokumenata
 - Klasifikacija dokumenata
 - Analiza sentimenta
 - Određivanje teme teksta
- Kako reformulirati ove zadake u problem **strojnog učenja**
- Kako pretprocesirati tekst da ga se može obrađivati algoritmima strojnog učenja

Tipičan zadatak 1:dohvat dokumenta

- Dano:
 - Kolekcija dokumenata (**korpus**)
 - Upit (može biti dokument ili kratki niz)
- Zadatak:
 - **Rangirati** sve dokumente u kolekciji prema sličnosti s upitom
- Stari problem (npr. knjižnice)
- Dohvat dokumenta je osnovni zadatak riješen web pretraživačima (“**10 blue links**”)

Dohvat dokumenta



- Direktan pristup: pretraživanje susjedstva (kao u **kNN**)
- Definiranje **funkcije udaljenosti** među dokumentima
- Za dani upit **q**, pronaći **k** dokumenata s **najmanjom udaljenosti** do **q**
- $k=10$, dokumenti sortirani prema udaljenosti, plavi linkovi, oglasi → 
- Teži dio: **izraditi/naučiti funkciju** udaljenosti (i skalirati na Web...)

Tipičan zadatak 2: Klasifikacija dokumenata

- Dano:
 - Dokument **d**
 - Skup klasa (npr. područja: vijesti, sport, tech, glazba, romanca)
- Zadatak:
 - Odlučiti kojoj klasi dokument **d** pripada
- Primjer:
 - Pronaći članke o finalu svjetskog prvenstva u nogometu

Klasifikacija dokumenata



- Nadzirano učenje
- Prikupiti veliku kolekciju dokumenata
- Označiti svaki dokument s pripadajućom klasom
- Predstaviti dokumente kao **vektore značajki**
- Učiti **klasifikator** na osnovu **označenih** dokumenata:
 - kNN, logistička regresija, stabla odluke, slučajne šume, boosted decision trees, neuronske mreže, ...

Tipični zadatak 3: analiza sentimenta

- Dano:
 - Dokument **d** (npr. recenzija produkta)
- Zadatak:
 - “**Sentimentni**” skor koliko je **d** pozitivan/negativan
- Primjer:
 - Zaključiti što ljudi misle o produkta na osnovu samo teksta (bez danih ocjena)
 - Analiza mišljenja kroz povijest; npr. kako se mijenja sklonost prema određenim političarima kroz vrijeme?

Sentimentna analiza



- Nadzirano učenje
 - Regresija
 - Klasifikacija
- Isti postav kao za klasifikaciju dokumenata
 - Označiti skup za učenje s ispravnim skorom
 - Predstaviti dokumente kao vektore značajki
 - Učiti model: kNN, linearna/logistička regresija, ...

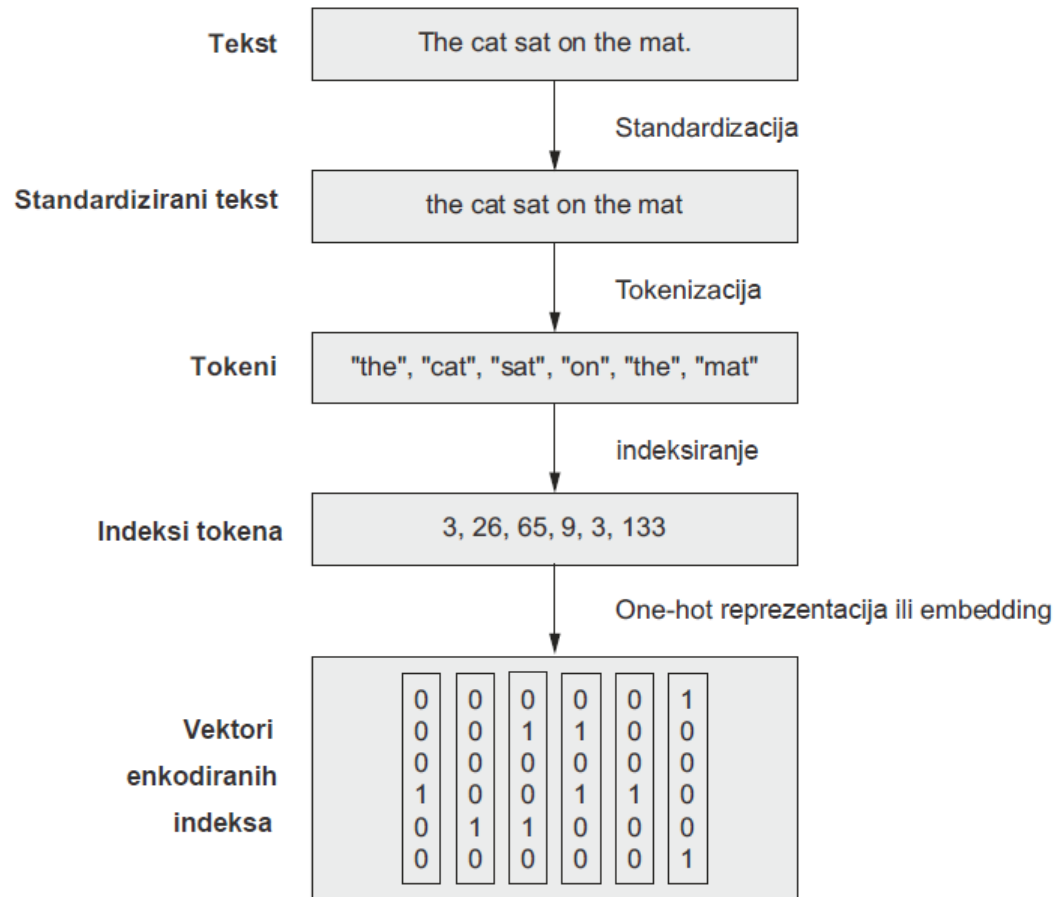
Tipičan zadatak 4: Određivanje teme

- Dano:
 - Neoznačena kolekcija dokumenata
- Zadatak:
 - Odrediti skup dominantnih tema u dokumentima
 - Odrediti za svaki dokument kojoj temi pripada
- Primjer:
 - Određivanje popularnih tema u socijalnim mrežama (npr. Twitter)
 - Utvrđivanje polarizirajućih gledišta oko političkih tema
 - Eksploratorna analiza velike kolekcije dokumenata

Određivanje teme

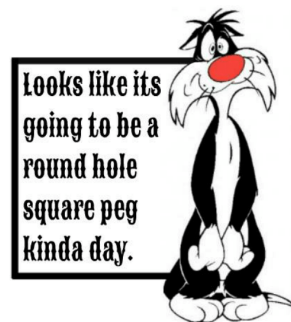


- Grupiranje
- Predstavljanje dokumenata kao vektora značajki
- Pokretanje algoritma grupiranja: hijerarhijskog ili algoritma dodjeljivanja točaka
 - Hijerarhijski: aglomerativan ili razdvajajući
 - Dodjeljivanje točaka: npr. k-means, DBSCAN
- Alternativno: faktORIZACIJA matrica



Vektori značajki

- Gotovo sve ML metode rade s **vektorima značajki**
 - Npr. s prethodnih slajdova: dohvat dokumenata; klasifikacija dokumenata; analiza sentimenta; utvrđivanje teme
- Tekst nije odmah u obliku vektora značajki
 - Varijabilna duljina
 - Čak i za fiksne duljine (npr. tweet...): Pozicije ne odgovaraju smislenim značajkama



Vektori značajki

- Potreba za pretvaranjem proizvoljno dugačkih nizova u vektore fiksne duljine
 - Tradicionalno i dobro isprobano: vreća riječi (**bag of words**)
 - Suvremeno: *učenje* mapiranja nizova u vektore (buzzword: “**text embedding**”)

Bag of words



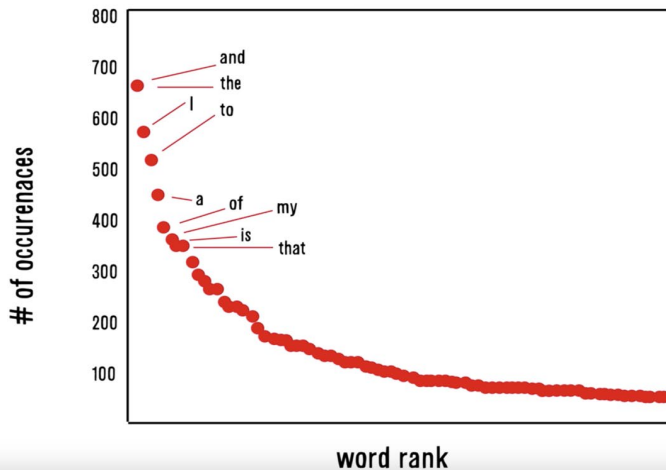
Tom Mitchell (CMU)

- Bag == multiset
 - “**multi-**”: zadržati raznolikost riječi
 - “**-set**”: Ne čuvati poredak
 - Npr. dokument “what you see is what you get”
→ bag of words {get:1, is:1, see:1, what:2, you:2}
- Za imati reprezentaciju fiksne duljine svih dokumenata :
 - Jedan zapis za svaki jedinstvenu riječ u rječniku
 - **Bag-of-word** vektori su visoko dimenzionalni (obično 1e5 or 1e6) i rijetki
 - Npr. za gornji primjer: [0...0 1 0...0 1 0...0 1 0...0 2 0...0 2 0...0]

Dodatan razlog za rijetkost: Zipfov zakon

Poznati zakon potencija

word frequency and rank in *Romeo and Juliet* (linear-linear)



Vjerojatnost pojavljivanje riječi je se obrnuto skalira s
njenim frekvencijskim rangom

$$p(w_i) \propto 1/i \quad (\text{gdje } w_i \text{ je } i\text{-ta najčešća riječ})$$

Bag-of-words matrica

dokumenti



riječi

- Kombinirati dokumente kao retke u matrici
 - Jedan redak po dokumentu
 - Jedan stupac po riječi u rječniku
- Ova matrica je ogromna!
 - Npr. Wikipedia: 5M dokumenata, 2M riječi → 10 bilijuna zapisa
- Korištenje rijetkog zapisa matrice
 - **Trojke**: (doc_idx, word_idx, count)
 - Npr. Wikipedia, pretpostavimo 2000 riječi po članku što iznosi 10 milijardi zapisa različitih od 0 (stane u memoriju)
- S **matričnom reprezentacijom** spremni smo koristiti bilo koji ML model

Jesmo li ?

- U teoriji da
- U praksi: “garbage in, garbage out”
- Biti oprezan pri mapiranju sirovog teksta u bag-of-words matricu!
 - Enkodiranje znakova
 - Određivanje jezika
 - Tokenizacija
 - Uklanjanje zaustavnih riječi
 - Normalizacija riječi
- Podešavanje matrice može voditi u puno bolje performanse
 - Normalizacija/dodjeljivanje težina redaka i/ili stupaca matrice

Vreća trikova za vreću riječi



Enkodiranje znakova

- Mapiranje znakova u oktete
- Old school: [ASCII](#), [Latin-1](#)
- Danas: [Unicode](#) (e.g., UTF-8, UTF-16, UTF-32)
- Npr. , W → 0x57
- Čitanje teksta iz datoteke:
 - Čitanje s **enkodiranjem** korištenim pri zapisu datoteke
 - Posebno važno za [ne-engleske](#) tekstove: č, ć, ž, đ, ...
- Zapisivanje u datoteku: Uvijek koristiti **UTF-8** or **UTF-16**; format!



```
file = codecs.open("temp", "w", "utf-8")
file.write(codecs.BOM_UTF8)
file.close()
```

Utvrdjivanje jezika

- Obično smo zainteresirani za jedan jezik
- Višejezičnost je u porastu (e.g., Twitter, Wikipedia)
- U idealnom slučaju kod jezika je specificiran (npr. zaglavlje HTML; JSON polje u Twitter API rezultatima)
- No, ne uvijek...
- Postoje sjajne biblioteke (npr. [ova](#))
 - Većina je temeljena na **trigramima** (e.g., “eau”, “ghi”, “ijs”, “sch”, “eiß”, “çãø”)
 - Mnogo teže u slučaju krivog **enkodiranja** znakova...

Tokenizacija

- Mapiranje znakova niza u slijed tokena (\approx riječi)
- Npr. “Hello! How are you?” \rightarrow Hello_!_How_are_you_?
- Čini se da je dovoljno samo koristiti bjeline i interpunkcijske znakove
- No, postoji mnogo rubnih slučajeva:
 - “Hello, Mr. President! How are you?! :-)”
 \rightarrow Hello_,_Mr._President_!_How_are_you_?!_:-)
- Bolje je koristiti postojeće biblioteke npr.
 - Python: [spaCy](#), [nltk](#)
 - Definirane na pravilima, deterministične, brze

Tokenizacija

- Optimalna **tokenizacija** je različita za različite jezike (e.g., Swedish “Saint Peter” → “S:t Peter”), ali engleska tokenizacija je obično dovoljna
- Relativno jednostavna **tokenizacija** u engleskom
- Teška u npr. kineskom: nema bjelina između riječi
- Složene riječi, npr. u njemačkom:
 - Napredni modeli mogu razdvojiti “Donaudampfschiffahrtskapitän” into “Donau dampf schiff fahrts kapitän”
 - No ponekad nije moguće znati treba li razdvojiti ili ne...?

Stop riječi

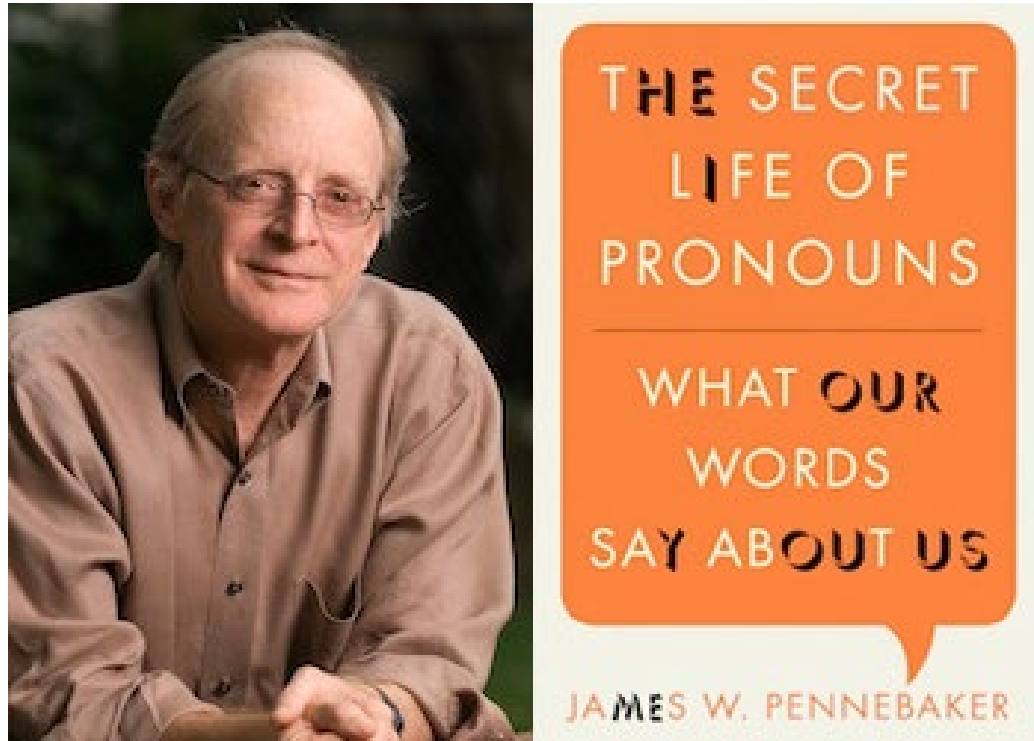
```
1 print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',  
'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she',  
'her', 'hers', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves',  
'that', 'that'll', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be',  
'been', 'ing', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if',  
'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through',  
'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',  
'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there',  
'when', 'where', 'why', 'how', 'all', 'any', 'each', 'every', 'both', 'neither',  
'nor', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',  
'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 's', 't',  
'n', "couldn't", 'didn't', "didn't", 'doesn't', "doesn't", 'hadn't', "hadn't",  
'isn't', 'ma', 'mightn't', "mightn't", 'mustn't', "mustn't", 'needn't', "needn't",  
'weren't', "weren't", 'won't', "won't", 'wouldn't', "wouldn't"]
```

Uklanjanje stop riječi

- Česte, “kratke” riječi nose **malo informacija** za većinu zadataka i “utopiti” informaciju sadržanu u stvarnim sadržajnim riječima
- Npr. “a”, “the”, “is”, “you”, “I”, oznake interpunkcije
- Dostupne mnoge liste stop riječi, no treba biti oprezan!
 - Različiti zadaci zahtijevaju uklanjanje različitih stop riječi
 - Dobra heuristika: **ukloniti riječi** koje se pojavljuju u najmanje **p%** svih tekstova (ali koliko je p...?)
 - Ponekad uklanjanje stop riječi šteti!
 - Identifikacija autora, psihološko modeliranje; **interpunkcija može biti korisna**: npr. “!!!”, “:-)”

Don't throw out the baby with the bathwater!



Normalizacija riječi: pretvaranje u mala slova

- npr, “I love yams. Yams are yummy.”
- Trebaju li “yams” and “Yams” biti različite značajke?
- Jednostavno rješenje: sve pretvoriti u mala slova
- Ali: “I’d rather have an **apple** than an **Apple**.”
- Ručno dodati izuzetke?
- U praksi (osobito za velike skupove podataka), obično je najbolje **NE** pretvarati
- No kada je skup podataka mali, može pomoći zato što je manje rijetkosti

Normalizacija riječi: Pretvaranje u korijen

- Npr “walking”, “walks”, “walked” → “walk”
“business”, “busy” → “busi”
- Obično se koriste heuristike (npr. [Porter stemmer](#))
- Pro: smanjuje rijetkost **bag-of-words** matrici
- Con: odbacuje informaciju
 - E.g., “business” vs. “busy”; “operating” (kao u “op. system”)
- U [engleskom](#) (pogotovo s velikim skupovima podataka) se više [ne koristi](#)
- Koristi se morfološki bogatijim jezicima (npr. hrvatski, njemački, [finski](#))

Socijalne mreže

Stvarni tweet:

“ikr smh he asked fir yo last name so he can add u on fb lololol”

- Prijevod:
 - *"ikr"* means "I know, right?"
 - *"smh"* means "shake my head"
 - *"fb"* means "Facebook", a very common proper noun.
 - *"yo"* is being used as equivalent to *"your"*.
 - *"fir"* is a misspelling or spelling variant of the preposition *for*. (But who knows?!)
- Često: ponavljanje slova/slogova (“yeahhh”, “hahahaha”, “haha”)
- Nezgodno s tradicionalnim NLP alatima...
- Potrebno imati specijalne alate kao što je [TweetNLP](#)

Tokeni vs. n-grami

- Do sada: **bag-of-words** matrica
 - Retci: dokumenti
 - Stupci: tokeni (a.k.a. **unigrami** ili **1-grami**)
- Često, dulje sekvence dolaze zajedno
 - Npr. “United States”, “operating system” – informacija o poretku
- Brute-force pristup: koristiti $n > 1$
 - Npr., sve **bigrame** ($n=2$), ili **trigrame** ($n=3$)
 - Korištenjem svih 5-grama može biti uspješnije od neuronskih (Tablica 1 [ovdje](#))
 - Problem: **kombinatorna eksplozija**

The cat sat on the mat

- Bigram

{"the", "the cat", "cat", "cat sat", "sat", "sat on",
"on", "on the", "the mat", "mat"}

- Trigram

{"the", "the cat", "cat", "cat sat", "the cat sat", "sat",
"sat on", "on", "cat sat on", "on the", "the mat",
"mat", "on the mat"}

Indeksiranje rječnika

- Enkodiranje svakog tokena u brojčanu reprezentaciju (npr. raspršeno adresiranje)
- Često restrikcija na 20000 – 30000 najčešćih riječi
- Specijalni tokeni:
 - “”:0 - maskiranje (nadopunjavanje)
 - “UNK”:1 - izvan rječnika

Reprezentacija grupe riječi

- Poredak u rečenici (**pozicija**)
- **Bag of words** – odbacimo informaciju o poretku (tretiramo kao skup)
- Slijedni modeli (slično vremenskim nizovima):
RNN, Transformers

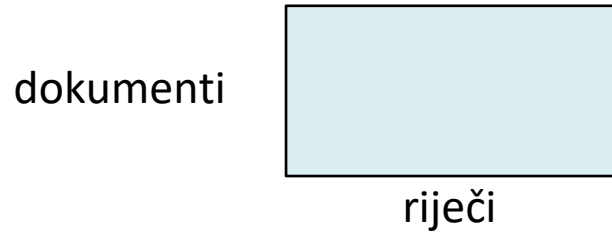
Bag of words

- {"cat", "mat", "on", "sat", "the"}
- Multi-hot enkodiranje (jedan vektor)
- {"the", "the cat", "cat", "cat sat", "sat", "sat on", "on", "on the", "the mat", "mat"} – bolja točnost

Brojanje pojavljivanja

- {"the": 2, "the cat": 1, "cat": 1, "cat sat": 1, "sat": 1, "sat on": 1, "on": 1, "on the": 1, "the mat": 1, "mat": 1}
- Riječi tipa "the", "a" i "is" prečeste -> normalizacija
- Klasična normalizacija -> problem (oduzimanjem srednje vrijednosti gubimo rijetkost)
- Rješenje **TF-IDF** normalizacija

Postprocesiranje BOW matrica



Inverse document frequency

- Nisu sve riječi jednako informativne
- Zbog toga uklanjamo **stop riječi** (“a”, “the”, “is”, ...)
- Osim uklanjanja stop riječi, želimo dati manju težinu na češće riječi
 - Npr. “per” vs. “perceptron”
- Standardan način: **IDF = inverse document frequency**
 - **docfreq(w)**: broj dokumenata koji sadrže riječ w
 - N: ukupan broj dokumenata
 - $\text{idf}(w) = -\log(\text{docfreq}(w) / N) = \log(N) - \log(\text{docfreq}(w))$

TF-IDF matrica

dokumenti



riječi

- $tf(w, d)$: term frequency – frekvencija pojavljivanja riječi w u dokumentu d
 - BOW to uhvati
 - Npr., dokument “what you see is what you get”
→ BOW {get:1, is:1, see:1, what:2, you:2}
- $idf(w)$: inverzna frekvencija pojavljivanja w u dokumentima (izračunato na cijelom korpusu)
- TF-IDF matrica:
 - Zapis u retku d i stupcu w ima vrijednost
 $tf(w, d) * idf(w)$
 - Iznos umnoška stupca w s konstantnim $idf(w)$

Normalizacija redaka TF-IDF matrice

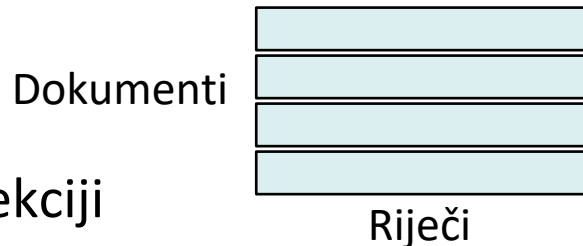
- Dulji dokumenti imaju više unosa različitih od nule
- Ako ih interpretiramo kao vektore, dulji dokumenti imaju dulje vektore
- Može smesti algoritme strojnog učenja
 - Dulji vektori su dalje od kratkih vektora
 - **Skalarni produkt**: slučajni vektor ima veći skalarni produkt s duljim vektorom
- Rješenje: normalizacija vektora dokumenata, npr. redaka TF-IDF matrice
 - **L2-normalizacija**: svi redci imaju euklidsku udaljenost 1 od ishodišta (sve točke leže na jediničnoj sferi)
 - **L1-normalizacija**: suma svih redaka je 1, može se interpretirati kao distribucija

Normalizacija stupca

- IDF-skaliranje se može promatrati kao normalizacija stupca
- Dodatno korištenje standardnih tehnika normalizacije
 - **Min-max** skala
 - Standardizacija: oduzeti srednju vrijednost i podijeliti sa standardnom devijacijom

Tipičan zadatak:dohvat dokumenta

- Metoda najbližih susjeda kao u **kNN**
- Usporediti upitni dokument q sa svim dokumentima (redci **TF-IDF** matrice) u kolekciji
- Rangirati dokumente iz kolekcije u rastućem poretку udaljenosti
- Matrice udaljenosti
 - Obično **kosinusna udaljenost** ($= 1 - \text{kosinusna sličnost}$)
 - Kosinusna sličnost q i $v = \langle q/|q|, v/|v| \rangle$
 - Ako su retci **L2**-normalizirani, jednostavno uzeti skalarni produkt $\langle q, v \rangle$

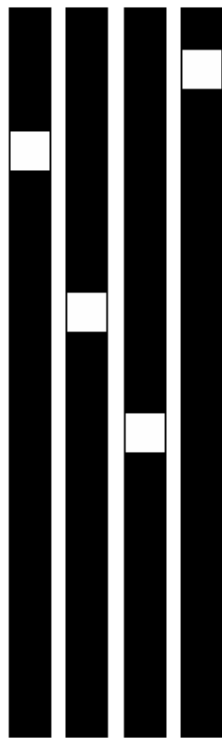


Tipičan zadatak: dohvat dokumenta

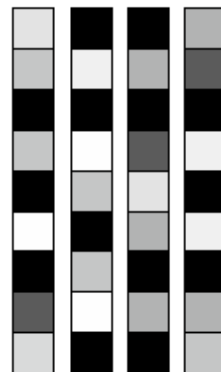
- Ovo je najosnovniji pristup
- Google radi mnogo toga dodatno...
 - Relevantnost neovisna o upitu: PageRank
 - Veća težina **novijim rezultatima**
 - personalizacija, konceptualizacija
 - ...
- Za efikasnost
 - Početi s filtriranjem dokumenata na osnovu prisustva termina iz upita (korištenje efikasnost indeksa cijelog teksta)
 - Znatno sužava skup dokumenata koje treba rangirati

Slijedni modeli - reprezentacija

- One-hot
 - tokeni su međusobno neovisni
 - 20000 dimenzionalan prostor
- Word embeddings
 - Geometrijski odnos između vektor riječi treba odražavati **semantički** odnos
 - 256 – 1024 dimenzije

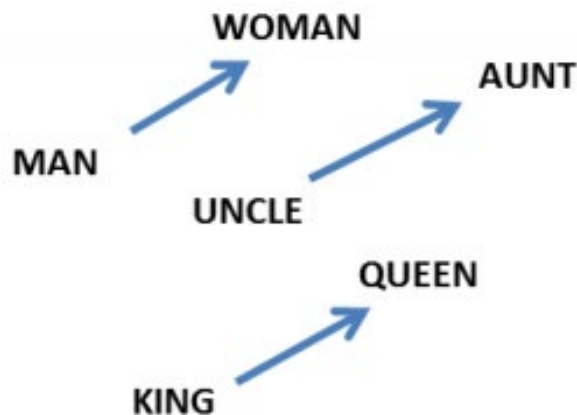


One hot vektori



Word embeddings

Word embedding – jednostavan primjer



$$W(\text{"woman"}) - W(\text{"man"}) \simeq W(\text{"aunt"}) - W(\text{"uncle"})$$

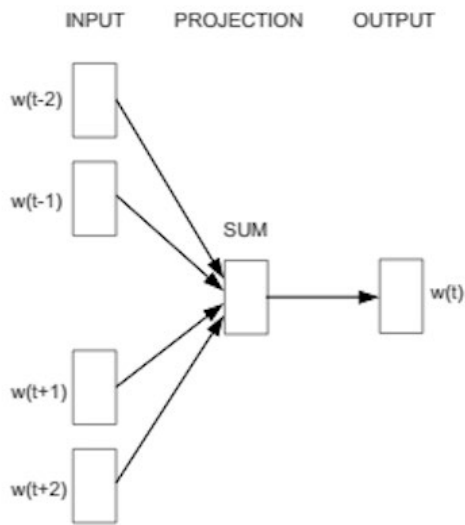
$$W(\text{"woman"}) - W(\text{"man"}) \simeq W(\text{"queen"}) - W(\text{"king"})$$

Kreiranje word embeddinga

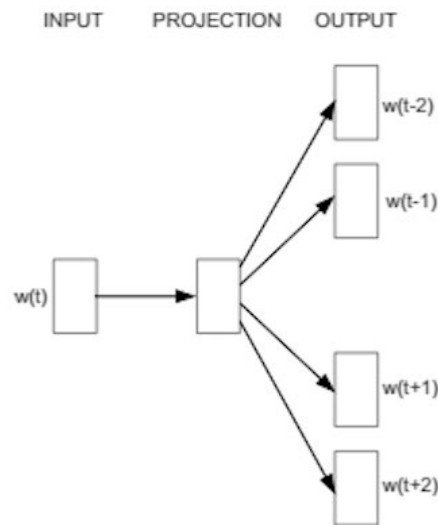
- Učenje zajedno s glavnim zadatkom (npr. klasifikacijom teksta)
- Učitavanje prethodno izračunatih u nekom drugom problemu – kada imamo malo ulaznih podataka

word2vec

- Predviđanje riječi koristeći kontekst
- Dvije verzije: **CBOW** (continuous bag of words) i **Skip-gram**



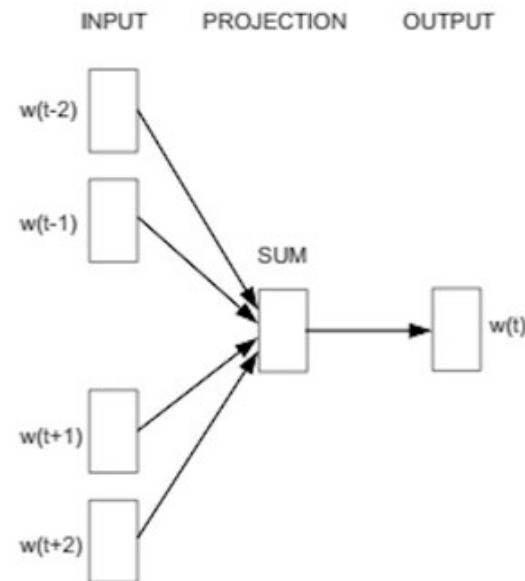
CBOW



Skip-gram

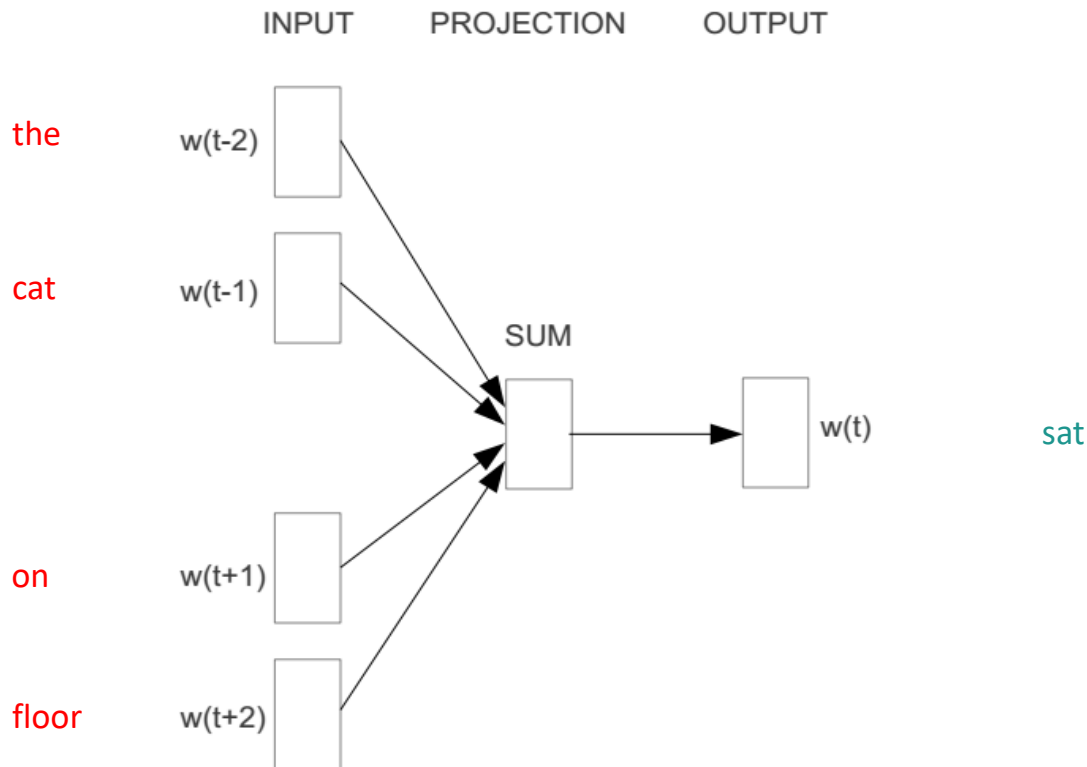
CBOW

- CBOW
 - Uzima vektor **embeddinga** n riječi prije ciljane riječi i n riječi nakon i zbraja ih.
 - Uklanja poredak riječi, no suma vektora je dovoljno smisljena da odredi nedostajuću riječ



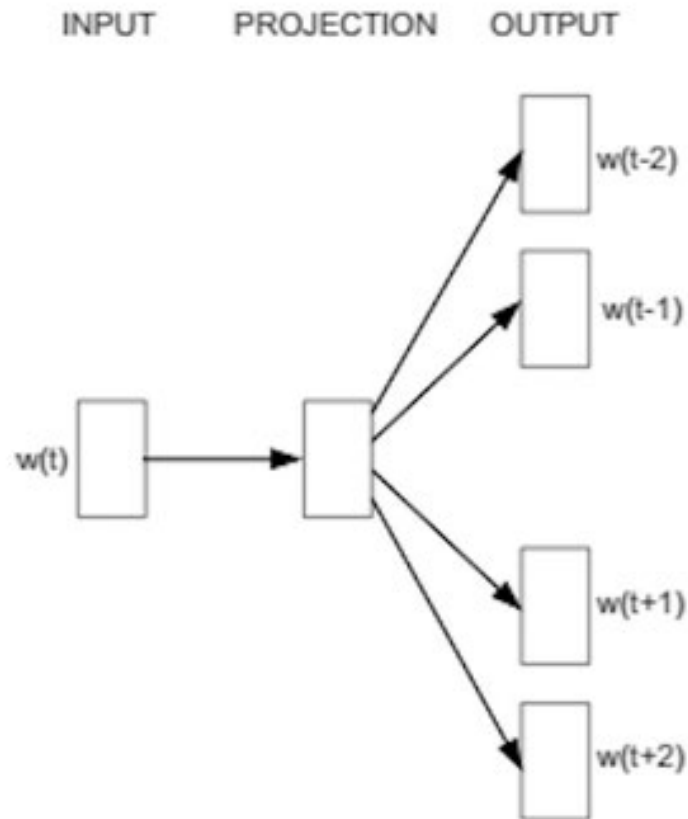
CBOW

Primjer “The cat sat on floor”

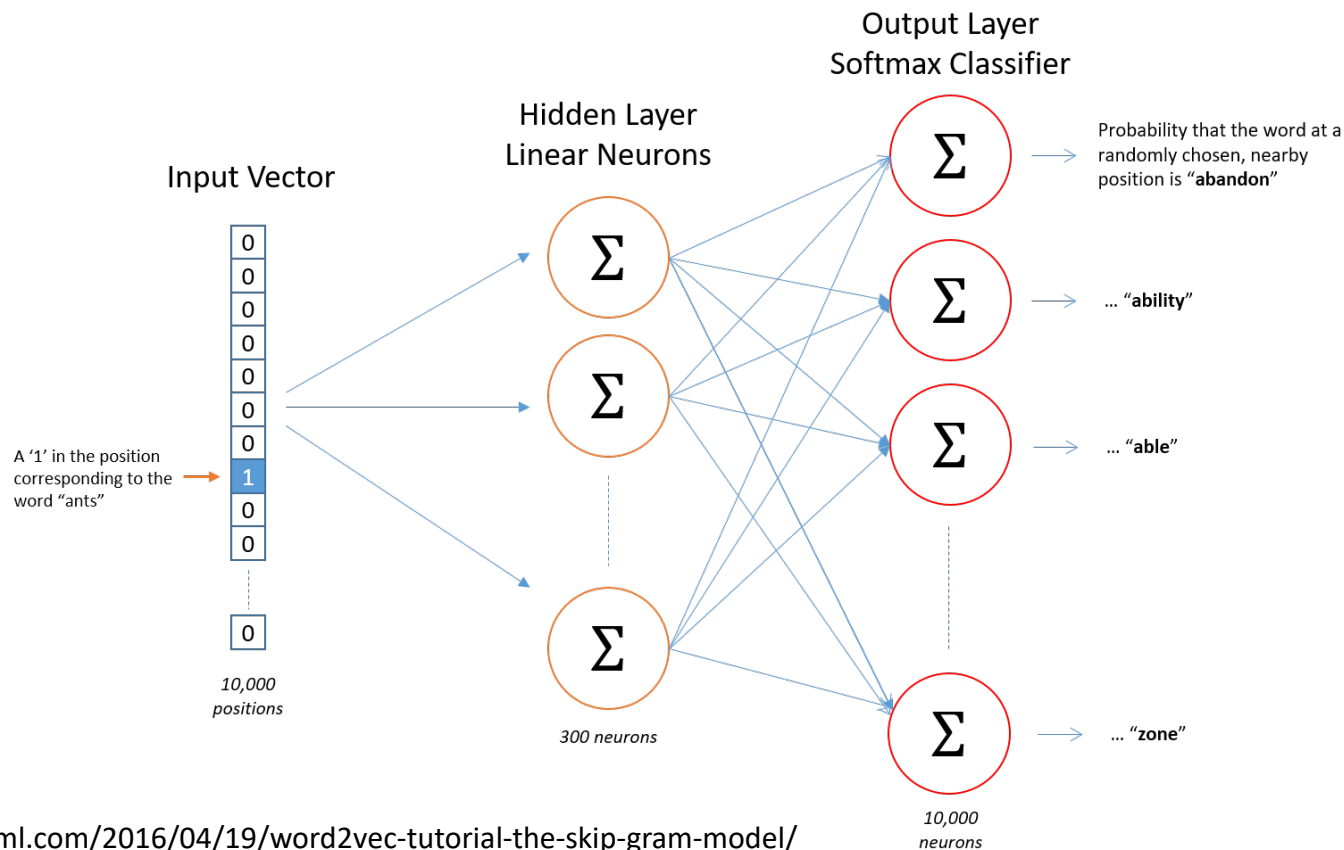


Skip gram

- **Skip gram** – alternativa za CBOW
 - Počinje s **embeddingom** jedne riječi i pokušava predvidjeti riječi koje ju okružuju
 - Lošije definiran problem, no bolje radi u praksi



Skip gram (jedna ulazno/izlazna jedinica)



Skip gram/CBOW intuicija

- Sličan „**kontekst**” vodi sličnom **embeddingu** za dvije riječi.
- Jedan je način da mreža na izlazu daje slične predikcije za te dvije riječi ukoliko su njihovi vektori slični. Ako dvije riječi imaju **sličan kontekst** onda je **mreža motivirana** naučiti slične vektore riječi za te dvije riječi!
- Alternativa:
 - **GloVe** (Global Vector) – temeljena na faktORIZACIJI matrici statistike ko-pojavljivanja riječi
 - **ELMo** (Embeddings from Language Models) - duboka kontekstualna reprezentacija riječi

Od riječi do teksta

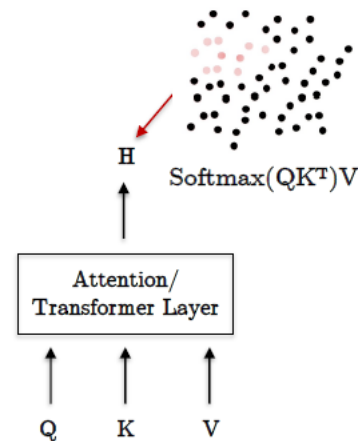
- Kako predstaviti veće jedinice, poput rečenica, odlomaka i dokumenata?
- Tipičan pristup: uzeti sumu/srednju vrijednost vektora riječi
- To je ugrubo što su **BOW** (kada koriste “one-hot” enkodiranje za riječi, npr. vektor s jednom 1, ostalo 0)
- Trenutno istraživanje: **učenje** vektora za veće jedinice
 - [Cr5](#), [sent2vec](#)
 - Konvolucijske neuronske mreže
 - Povratne neuronske mreže, npr. LSTM
 - Transformer-temeljeni modeli, npr., GPT3

Slijedni modeli

- Povratne neuronske mreže
- Transformeri (temeljeni na self-attention mehanizmu)

Context-To-Word reprezentacija

- Od **Word reprezentacija** do **Context-to-Word** reprezentacije
$$H \leftarrow \text{Softmax}(QK^T)V \in \mathbb{R}^{n \times d}$$
- Nova reprezentacija je suma svih ulaznih podataka otežana attention skorovima
- Podskup podataka s **attentionom** različitim od nula kreira kontekst
- **Attention** mehanizam omogućava dinamičku promjenu reprezentacije riječi u skladu s njegovim kontekstom
- **Context-to-Word** je snažna ideja u obradi jezika zbog toga što riječi mogu imati drugačija značenja koja se mogu razjasniti jedino u pojedinom kontekstu
- The vase **broke**. The news **broke**. Sandy **broke** the world record. Sandy **broke** the law. We **broke** even. The burglar **broke** into the house. Etc



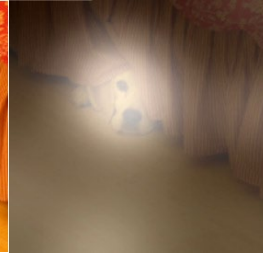
Određivanje naslova slike koristeći attention



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.



Attention princip

- Imitira dohvat **vrijednosti** v_i za **upit** q na osnovu **ključa** k_i u bazi podataka
- $attention(q, \mathbf{k}, \mathbf{v}) = \sum_i sličnost(q, k_i) \times v_i$

Ulazni
slijed

'the', 'train', 'left', 'the', 'station', 'on', 'time'

Token
vektori



	the	train	left	the	station	on	time
the	1.0	0.3	0.1	0.5	0.2	0.1	0.1
train	0.3	1.0	0.6	0.3	0.8	0.1	0.2
left	0.1	0.6	1.0	0.1	0.6	0.1	0.1
the	0.5	0.3	0.1	1.0	0.3	0.1	0.2
station	0.2	0.8	0.6	0.3	1.0	0.2	0.2
on	0.1	0.1	0.1	0.1	0.2	1.0	0.5
time	0.1	0.2	0.1	0.2	0.2	0.5	1.0

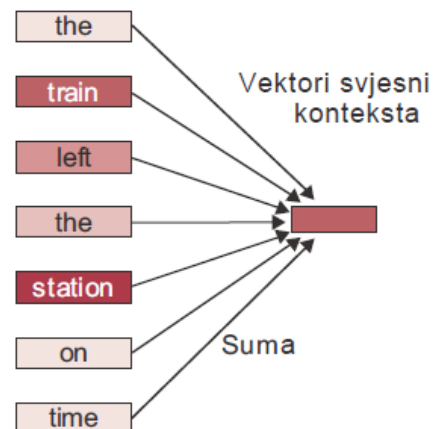
Attention skorovi

Skorovi za
"station"

0.2
0.8
0.6
0.3
1.0
0.2
0.2

Softmax,
skaliranje i
množenje

Težinski
token vektori



Softmax

Formula

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

σ = softmax

\vec{z} = input vector

e^{z_i} = standard exponential function for input vector

K = number of classes in the multi-class classifier

e^{z_j} = standard exponential function for output vector

Prvo kreiramo tri vektora
množenjem ulaznih **embeddinga**
(1x512) x_i
s tri matrice (512x64):

$$q_i = x_i W^Q$$

$$K_i = x_i W^K$$

$$V_i = x_i W^V$$

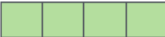
Input

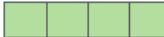
Thinking

Machines

Embedding


512 \rightarrow 4


x_1 

x_2 

Queries

64 \rightarrow 3

q_1 

q_2 



W^Q

Keys

k_1 


k_2 



W^K

Values

v_1 

v_2 



W^V

Računamo skor da
odredimo koliko fokusa
je na druge
dijelove ulaza.

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 ($\sqrt{d_k}$)

Softmax

Thinking

x_1 

q_1 

k_1 

v_1 

$$q_1 \cdot k_1 = 112$$

14

0.88

Machines

x_2 

q_2 

k_2 

v_2 

$$q_1 \cdot k_2 = 96$$

12

0.12

Formula

Q

×

K^T

)

V

softmax

(

$\frac{\quad}{\sqrt{d_k}}$

)

=

Z

$d_k=64$ je dimenzija vektora key

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 ($\sqrt{d_k}$)

Softmax

Softmax

X
Value

Sum

Thinking

Machines

x₁

x₂

q₁

q₂

k₁

k₂

v₁

v₂

q₁ • k₁ = 112

q₁ • k₂ = 96

14

12

0.88

0.12

v₁

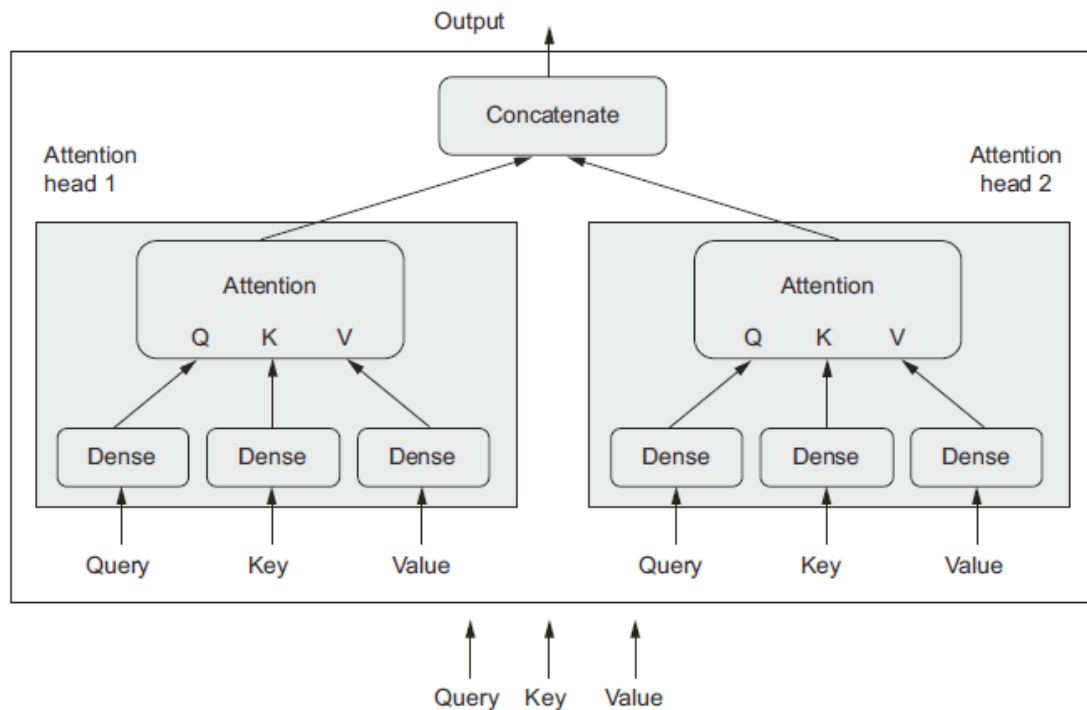
v₂

$z_1 = 0.88v_1 + 0.12v_2$

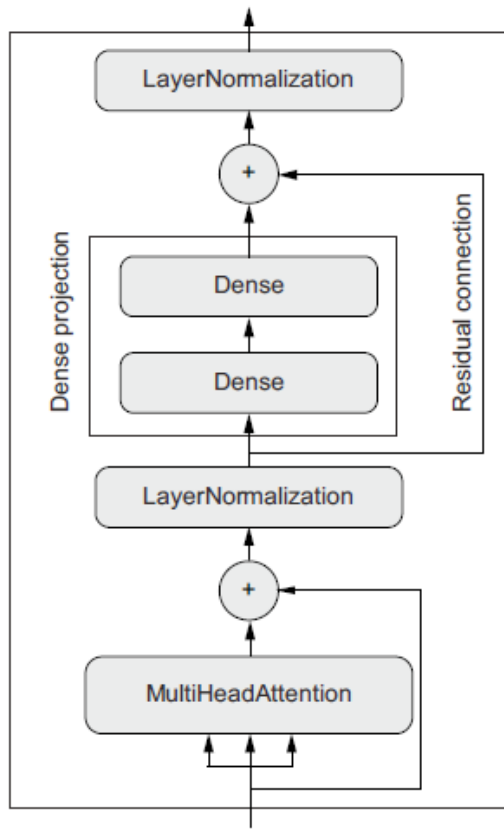
z₁

z₂

Multihead attention



Transformer encoder

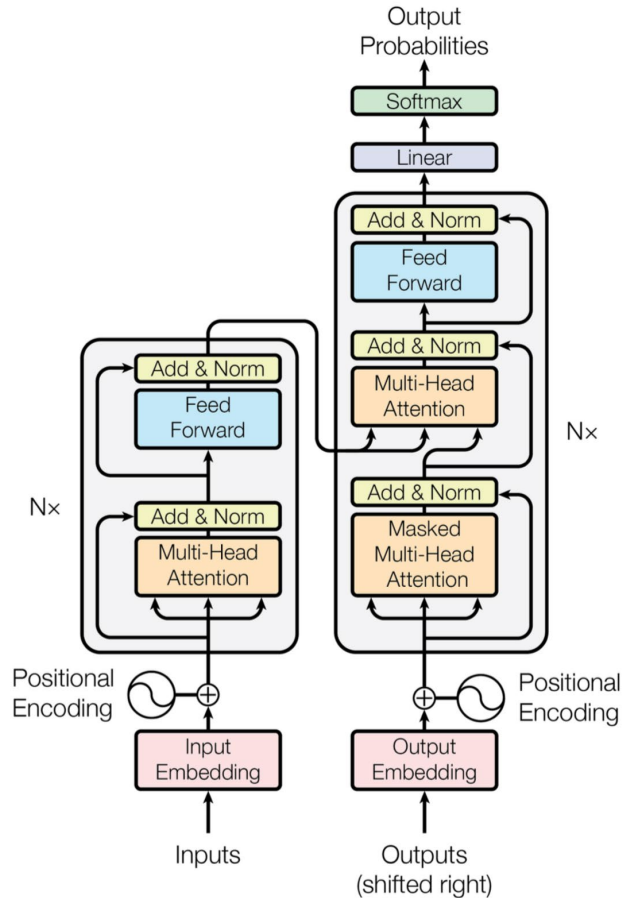


- Koristimo ga za kvalifikaciju teksta
- U kombinaciji s dekomerom koristimo ga prevođenje (nije dio ovoga predmeta)
- Dodavanje dodatnih **dense** projekcija
- **Layer Normalization/Batch Normalization**
Normalizacijski slojevi pomažu da gradijent bolje „teče” za vrijeme backpropagacije
- **Rezidualna veza** – želimo biti sigurni da nismo niti jednu korisnu informaciju u transformacijama

Enkodiranje pozicije

- Self-attention ne vodi računa o poziciji
- Originalan rad „Attention is all you need”
 - Dodavanje word embeddinza vektor kontinuiranih vrijednosti u rasponu $[-1, 1]$ koji varira ciklički ovisno o poziciji (kosinusna funkcija)
 - Ovaj trik pruža način da se jedinstveno opiše bilo koji cijeli broj u veliku rasponu s vektorom malih vrijednosti
- Može se umjesto toga koristiti naučiti **enkodiranje pozicije** (positional embedding)

Transformer



	Model svjestan poretka riječi	Model svjestan konteksta (interakcija među riječima)
Bag-of-unigrams	Ne	Ne
Bag-of-bigrams	Vrlo ograničeno	Ne
RNN	Da	Ne
Self-attention	Ne	Da
Transformer	Da	Da

Pošaljite mi svoje mišljenje o ovom predavanju na:
mile.sikic@fer.hr

- Što (ni) ste naučili na ovom predavanju?
- Što (ni) je bilo dobro objašnjeno?
- O čemu (ne) biste željeli čuti više detalja?
- ...