Obravnava delno strukturiranih podatkov in vpetja (*embeddings*)

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za upravo
Institut Jožef Stefan, Odsek za tehnologije znanja (E-8)

Maj 2020

Pregled predavanja

Delno strukturirani podatki

Besedila, slike, . . .

Model vreče besed (*bag of words*)

- Slovar terminov
- Frekvenci terminov in dokumentov
- Frekvenca terminov in obratna frekvenca dokumentov (*TFIDF*)

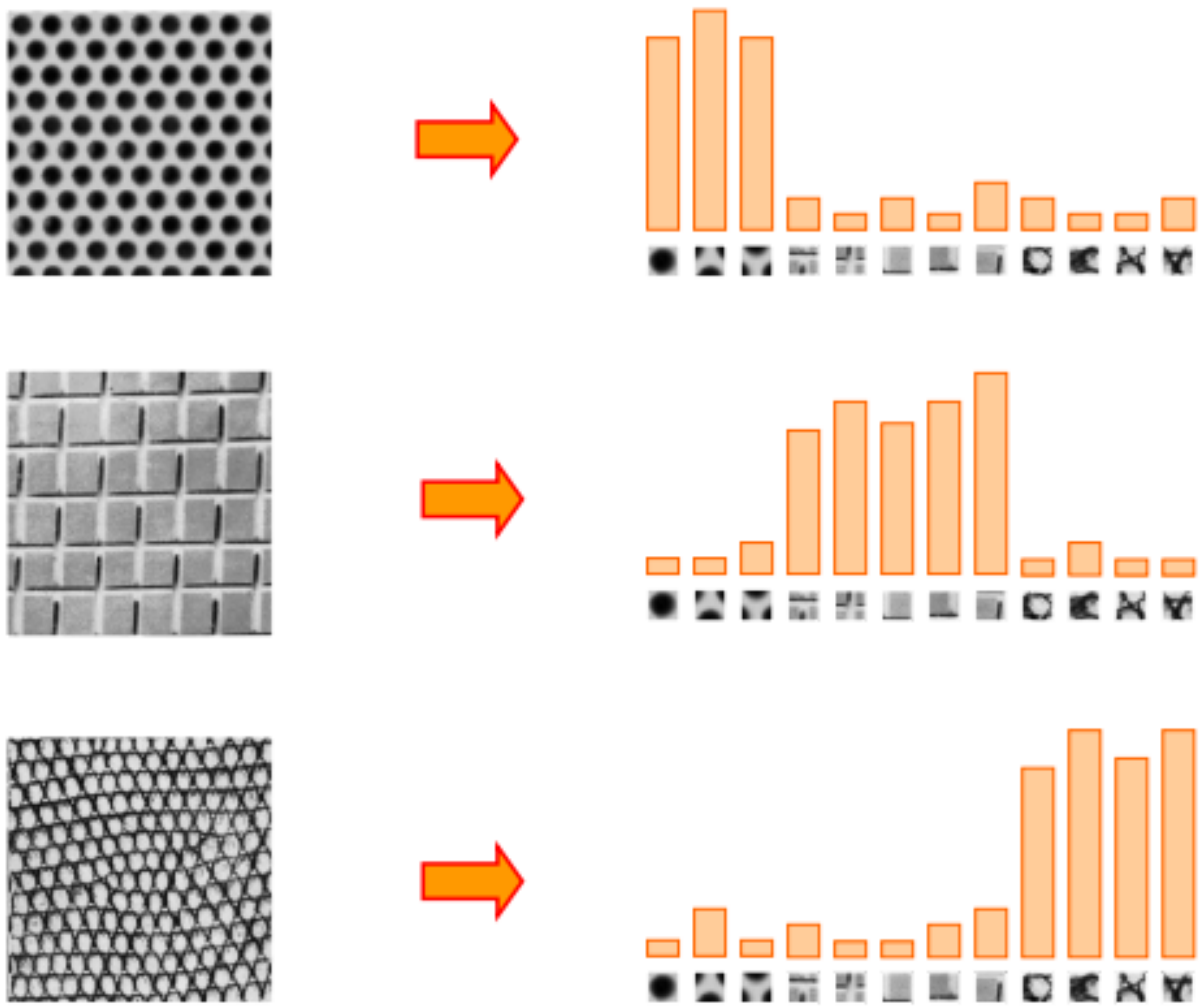
Vpetja (*embeddings*) besedil

- Pomenska podobnost besed
- Vpetja word2vec in doc2vec

Motivacija: klasifikacija besedil

Motivacija: razpoznavanje slik

Osnovna ideja: slovar besed



Formalizacija modela

Tri komponente O, E in f

- Osnovna podatkovna množica O: delno strukturirani podatki
- Slovar E: množica osnovnih podatkovnih elementov
- Funkcija frekvence $f : E \times O \to \mathbb{N}_0$: vrne frekvenco (število) pojavitev $e \in E$ v $o \in O$

$|E|$ spremenljivk za predstavitev delno strukturiranih podatkov iz O

- Vsakemu osnovnemu elementu $e \in E$ priredimo spremenljivko X_e
- Vrednost X_e za podatek $o \in O$ je enaka $f(e, o)$ ali njeni transformaciji

Transformacije f za besedila: frekvenca termina TF

Besedila

- O je množica besedil dokumentov $d \in O$
- E je slovar besed, imenovanih terminov (*terms*) $t \in E$

Frekvenca termina, *term freqency*, $TF(t, d) = tf_{t,d} = f(t, d)$

Imenujemo jo tudi naravna frekvenca besede oziroma termina.

Alternativne mere TF

- Boolova $TF_B(t, d) = I(TF(t, d) > 0)$
- Normalizirana $TF_N(t, d) = TF(t, d)/|d|$, $|d|$ je število besed v d
- Logaritmična $TF_L(t, d) = \log(1 + TF(t, d))$

Transformacije f za besedila: frekvenca v dokumentih DF

Slabost frekvence termina TF

Pogoste so besede brez pomena (prazne besede, *stop words*), ki se pojavijo v vsakem dokumentu iz O, na primer *in*, *ali* ali *biti*. Možni rešitvi:

- Praznih besed ne damo v slovar E
- Upoštevamo frekvenco v dokumentih DF

Frekvenca v dokumentih, *document frequency* DF

$$DF(t) = |\{d \in O : TF(t, d) > 0\}|$$

Transformacije f za besedila: *TFIDF*

Obratna frekvenca v dokumentih, *inverse document freqency IDF*

$$IDF(t) = \log \frac{|O|}{DF(t)}$$

Za prazne besede t je $IDF(t) \approx 0$, saj za njih velja $DF(t) \approx |O|$.

Alternativi meri *IDF*

- Gladka $IDF_G(t) = \log(1 + |O|/DF(t))$
- Verjetnostna $IDF_P(t) = \log(|O|/DF(t) - 1)$

$$TFIDF(t, d) = TF(t, d) \cdot IDF(t)$$

Orodja za pred-obdelavo besedil

Lematizacija

- Pretvorba besede v njeno osnovno obliko
- Na primer: besede *je*, *sem* in *so* imajo osnovno obliko *biti*
- Lematizator je orodje, ki besedi pripiše njeno osnovno obliko
- Lematizator za slovenščino dostopen na lemmatise.ijs.si

Krnenje (*stemming*)

- Poenostavljena in hitra lematizacija, pretvorba besede v njen koren
- Na primer: besedi *delati* in *delovanje* imata skupni koren *delo*
- Krnilnik: orodje, ki besedi pripiše njen koren

Primer obdelave besedil: pet stavkov in lematizacija

Osnovni stavki

- Hotel je večji gostinski obrat, v katerem se dobi prenočišče in hrana.
- Motel je gostinski obrat hotelskega tipa, navadno ob velikih cestah zunaj naselij.
- Bi izbrali hotel ali motel?
- Janez je izbral hotel.
- Metka je najela sobo 215 v motelu Medno.

Lematizirani in pred-obdelani stavki

- hotel biti velik gostinski obrat v kateri se dobiti prenočišče in hrana
- motel biti gostinski obrat hotelski tip navadno ob velik cesta zunaj naselje
- biti izbrati hotel ali motel
- janez biti izbrati hotel
- metka biti najeti soba 215 v motel meden

Primer obdelave besedil: izračun TF_B in IDF

$$E = \{biti, hotel, motel, obrat, hrana\}$$

ID	biti	hotel	hrana	motel	obrat
1	1	1	1	0	1
2	1	0	0	1	1
3	1	1	0	1	0
4	1	1	0	0	0
5	1	0	0	1	0
DF	5	3	1	3	2
$IDF = \log(5/DF)$	0	0.7370	2.3219	0.7370	1.3219

Primer obdelave besedil: izračun *TFIDF*

ID	biti	hotel	hrana	motel	obrat
1	0	0.7370	2.3219	0	1.3219
2	0	0	0	0.7370	1.3219
3	0	0.7370	0	0.7370	0
4	0	0.7370	0	0	0
5	0	0	0	0.7370	0

Kako pridemo do slovarja E?

Za besedila

- Običajno vse besede v dokumentih iz O
- Včasih tudi izbrane besede in njih kombinacije, n-grami
- Osnovne (korenske) oblike za lematizirana (krnenja) besedila

Za slike

- Vizualni slovar: koščki slik, ki jih lahko najdemo v slikah iz O
- Lahko ga dobimo avtomatsko z iskanjem pogostih vzorcev v slikah

Kako implementiramo f?

Za besedila

Preprosto primerjanje besed.

Za slike

Konvolucija: konvolucijske ravni nevronskih mrež pravzaprav avtomatsko odkrivajo vizualni slovar.

Omejitve

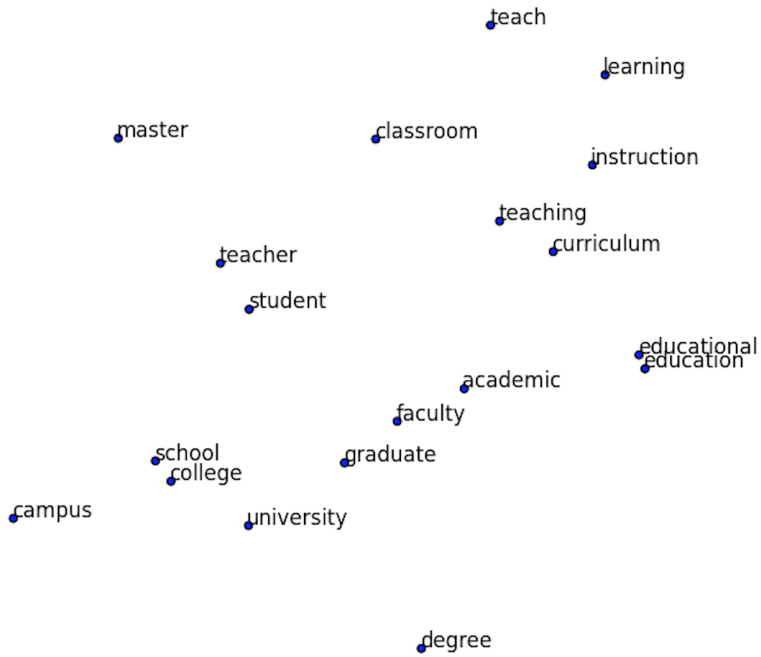
Pomen besed popolnoma nepomemben

Podobne besede iz slovarja predstavljene z različnimi spremenljivkami.
Rešitev: vpetja, drugi del današnjega predavanja.

Veliko število spremenljivk

Rešitev: metode za izbiro in konstrukcijo napovednih spremenljivk, naslednja predavanja.

Osnovna ideja



Pretvorba besed v vektorje realnih števil

Omejitev: Podobni vektorji ustrezajo podobnim besedam.

Porazdelitvena predpostavka (*distributional hypothesis*)

Besede s podobnim pomenom uporabljamo v istem kontekstu.

Kaj je kontekst?

Dva lematizirana stavka

- hotel biti velik gostinski obrat v kateri se dobiti prenočišče in hrana
- motel biti gostinski obrat hotelski tip navadno ob velik cesta zunaj naselje

Dve-besedni kontekst besede *obrat*

- Prvi stavek: *velik*, *gostinski*, *v*, *kateri*
- Drugi stavek: *biti*, *gostinski*, *hotelski*, *tip*

Dva možna napovedna modela

- Napovedne spremenljivke: kontekst, ciljna spremenljivka: beseda
- Napovedna spremenljivka: beseda, ciljne spremenljivke: kontekst

Kodiranje *one-hot*

Uporabljamo ga za kodiranje besed v nevronskih mrežah.

Kodiranje besede e iz slovarja E

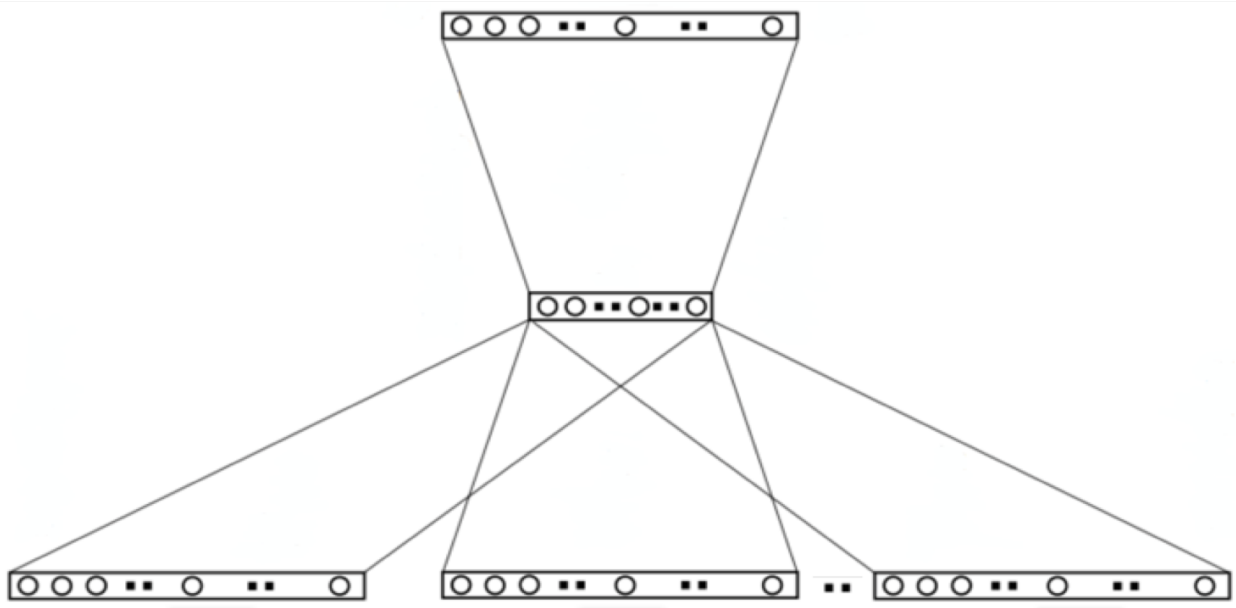
$$x_i = \begin{cases} 1 & ; \quad i = index(e, E) \\ 0 & ; \quad sicer \end{cases}$$

- Vektor x dolžine $|E|$
- $index(e, E)$ je indeks besede e v slovarju E
- Slovar E obravnavamo kot urejeno množico

Model *CBOW*, *Continuous Bag-of-Words*

Struktura nevronske mreže

- Vhodna raven: $2 \cdot k \cdot |E|$ nevronov, m je velikost konteksta, E je slovar
- Skrita raven: m nevronov, m je dolžina predstavitvenih vektorjev
- Izhodna raven: $|E|$ nevronov



CBOW: vhodna in skrita raven

Vhodi za besede iz konteksta $w^{(c-k)}, \ldots, w^{(c-1)}, w^{(c+1)}, \ldots, w^{(c+k)}$

- Vsaka beseda $w^{(i)}$ dobi svojo skupino $|E|$-tih nevronov $x^{(i)}$
- Vse besede so kodirane z *one-hot*

Sinapse in uteži

- Vsaka skupina vhodov $x^{(i)}$ je polno povezana s skrito ravnijo
- Uteži na sinapsah za različne skupine vhodnih nevronov so **enake**
- Matrika uteži W dimenzij $m \times |E|$

CBOW: stanja in izhodi skritih nevronov

$$\boldsymbol{v} = \frac{W\,x^{(c-k)} + \ldots + W\,x^{(c-1)} + W\,x^{(c+1)} + W\,x^{(c+k)}}{2\,k}$$

- Vektor stanj skritih nevronov $\boldsymbol{v}$ je vpetje besede $w^{(c)}$
- Za izhode stanj lahko uporabimo poljubno aktivacijsko funkcijo ϕ
- Običajna je identiteta $\phi(v) = v$

CBOW: izhodna raven in funkcija izgube

Izhodna raven: koda *one-hot* y za centralno besedo $w^{(c)}$

$$\boldsymbol{z} = W'\boldsymbol{v},\ \hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^{|E|}\exp(z_j)}$$

- Skrita raven polno povezana z izhodno ravnijo: $\boldsymbol{z} = W'\boldsymbol{v}$
- Matrika uteži W' je dimenzij $|E| \times m$
- Aktivacijska funkcija *softmax*

Funkcija izgube je prečna entropija (*cross entropy*)

$$L(y, \hat{y}) = -\sum_{j=1}^{|E|} y_j \log \hat{y}_j$$

CBOW: od funkcije izgube do ciljne funkcije

Ker imamo kodiranje *one-hot*, vsota ima le en člen

$$L(y, \hat{y}) = -y_i \log \hat{y}_i = -\log \hat{y}_i$$

i je indeks centralne besede $w^{(c)}$ v slovarju E

Ciljna funkcija je zato enostavna

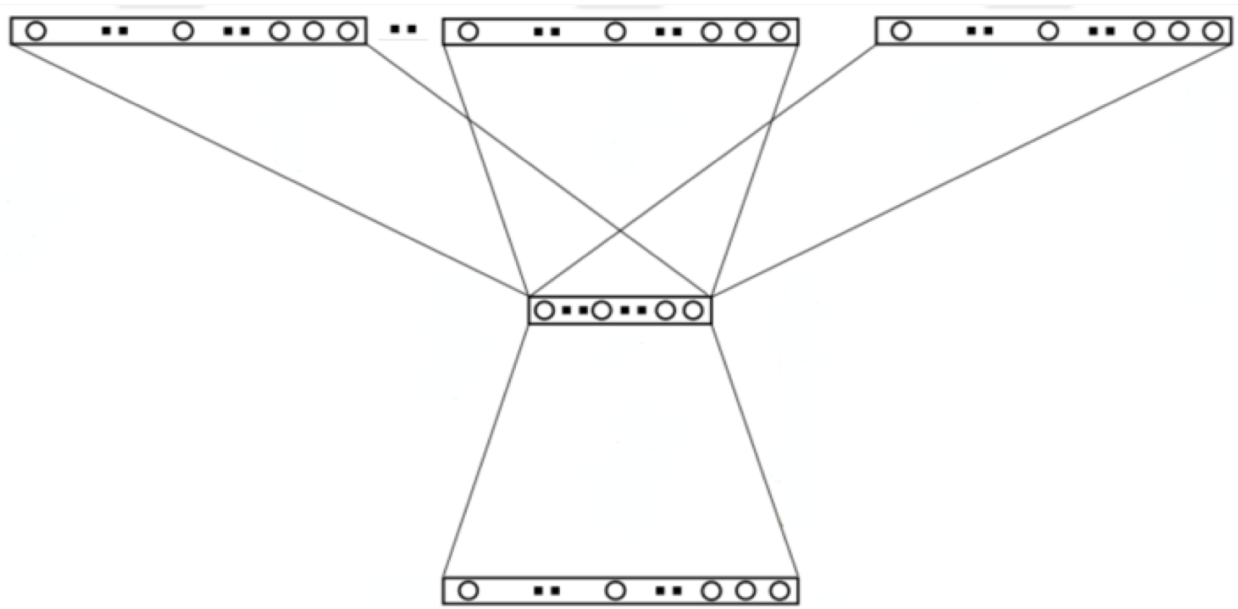
$$Err = -\log \hat{y}_i = -\log \frac{\exp(z_i)}{\sum_{j=1}^{|E|} \exp(z_j)} = -z_i + \log \sum_{j=1}^{|E|} \exp(z_j)$$

Od tukaj naprej vzvratno razširjanje napake.

Model *Skip–Gram*

Struktura nevronske mreže

- Vhodna raven: $|E|$ nevronov
- Skrita raven: m nevronov, m je dolžina predstavitvenih vektorjev
- Izhodna raven: $2 \cdot k \cdot |E|$ nevronov, k je velikost konteksta, E slovar



Skip–Gram: vhodna in skrita raven

Vhodi za centralno besedo $w^{(c)}$

Koda *one-hot* $x^{(c)}$ za centralno besedo.

Sinapse in uteži

- Vhodi $x^{(c)}$ so polno povezani s skrito ravnijo
- Matrika uteži W dimenzij $m \times |E|$

Skip–Gram: stanja in izhodi skritih nevronov

$$\boldsymbol{v} = W\, x^{(c)}$$

- Vektor stanj skritih nevronov $\boldsymbol{v}$ je vpetje besede $w^{(c)}$
- Za izhode stanj lahko uporabimo poljubno aktivacijsko funkcijo ϕ
- Označili bomo $v' = \phi(v)$

Skip–Gram: izhodna raven

Kode *one-hot* $y^{(j)}$ za besede iz konteksta $w^{(j)}$

$$\boldsymbol{z} = W'\boldsymbol{v},\ \hat{y}_i^{(j)} = \frac{\exp(z_i^{(j)})}{\sum_{l=1}^{|E|}\exp(z_l^{(j)})}$$

- Skrita raven polno povezana z izhodno ravnijo: $\boldsymbol{z} = W'\boldsymbol{v}$
- Izhodna raven sestavljena iz skupin izhodnih nevronov $y^{(j)}$ za vsako besedo iz konteksta $w^{(j)}$, $j = c-k, \ldots, c-1, c+1, \ldots, c+k$
- Matrika uteži W' je dimenzij $2 \cdot k \cdot |E| \times m$
- Aktivacijska funkcija *softmax* za vsako skupino izhodov

Skip–Gram: ciljna funkcija

Negativni logaritem verjetja konteksta za podano centralno besedo

$$Err = -\log P(w^{(c-k)}, \ldots, w^{(c-1)}, w^{(c+1)}, \ldots, w^{(c+k)} | w^{(c)})$$

Predpostavka neodvisnosti besed iz konteksta (naïvni Bayes)

$$Err = -\log \prod_{j=-m, j\neq 0}^{k} P(w^{(c+j)} | w^{(c)}) = -\log \prod_{j=-m, j\neq 0}^{k} \hat{y}_{i^{(j)}}^{(c+j)}$$

Indeksi $i^{(j)}$ so indeksi besede iz konteksta $w^{(j)}$ v slovarju E.

Skip–Gram: ciljna funkcija, nadaljevanje

$$
\begin{aligned}
Err &= -\log \prod_{j=-m, j\neq 0}^{k} \hat{y}_{i(j)}^{(c+j)} \\
&= -\log \prod_{j=-m, j\neq 0}^{k} \frac{\exp(z_{i(j)}^{(j)})}{\sum_{l=1}^{|E|} \exp(z_l^{(j)})} \\
&= -\sum_{j=-m, j\neq 0}^{k} z_{i(j)}^{(j)} + \sum_{j=-m, j\neq 0}^{k} \sum_{l=1}^{|E|} \exp(z_l^{(j)})
\end{aligned}
$$

Od tukaj naprej vzvratno razširjanje napake.

Ohranjanje pomena besed

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Ohranjanje razmerij analogije med besedami

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Praktična uporaba za besedilne podatke

1. Naučimo word2vec model iz čim večjega nabora besedil
2. Uporabljamo ga za pretvorbo posameznih besed v vektorje
3. Od vektorjev za besede do vektorja za besedilo
   - Izračunamo povprečje vektorjev za vse besede v besedilu
   - Izračunamo povprečje vektorjev za nekaj izbranih besed, npr. tistih z največjim *TFIDF*
   - Naredimo stik vektorjev za nekaj izbranih besed

Ali pa uporabimo doc2vec, verzijo word2vec za besedila.

Enostavna nadgradnja modela *CBOW*

Nadgradimo vhodno raven nevronov

- Kontekstu dodamo še eno vhodno spremenljivko: ID besedila
- Iz konteksta in ID besedila, napovedujemo centralno besedo

Skriti nevroni

Enako kot pri word2vec, podajo predstavitev besedila z vektorjem realnih števil: podobna besedila predstavljena s podobnimi vektorji.

Učenje mreže za vsako novo besedilo

Spreminjajo se le tiste uteži, ki se nanašajo na ID besedila, ne pa tiste, ki se nanašajo na kontekst.

Algoritmi in implementacije

word2vec (Mikolov in ost 2013)

Implementacija v R github.com/bmschmidt/wordVectors

doc2vec (Le in Mikolov 2014)

Implementacija v R: paket textTinyR