Analiza sentimenta

(angl. Sentiment Analysis)

Analiza sentimenta

Različna imena

- Analiza sentimenta (angl. *Sentiment Analaysis*)
- Pridobivanje mnenja (angl. *Opinion extraction*)
- Rudarjenje mnenj (angl. *Opinion mining*)
- Rudarjenje sentimenta (angl. *Sentiment mining*)
- Analiza subjektivnosti (angl. *Subjectivity analysis*)

Analiza sentimenta

- Je kritika filma pozitivna ali negativna?
- Kakšno mnenje imajo študenti na FERI o Androidu
- Kakšno je zaupanje potrošnikov (narašča/pada)
- Kakšno je mnenje ljudi o politikih
- Napoved rezultatov volitev

Tipologija čustev (Scherer)

- Emocije (jeza, žalost, veselje, strah, sram, ponos, vznesenost)
- Razpoloženje (veselo, mračno, razdražljivo, brezvoljno, depresivno)
- Medosebna stališča (prijaznost, spogledljivost, oddaljenost, hladnost, toplost)
- Odnos (naklonjenost, ljubezen, sovraštvo, spoštovanje)
- Osebnostne lastnosti (živčnost, nestrpnost, nepremišljenost, čemerenost, sovražnost, ljubosumnost)

Analiza sentimenta

- Analiza sentimenta je odkrivanje odnosov
- Kakšno mnenje imajo študenti na FERI o Androidu
- Kakšno je zaupanje potrošnikov (narašča/pada)
- Kakšno je mnenje ljudi o politikih
- Napoved rezultatov volitev

Analiza sentimenta

Analiza sentimenta je odkrivanje **odnosov**.

- Nosilec (izvor)
- Cilj (aspekt)
  - cena tiskalnika, enostavnost uporabe tiskalnika itd.
- Tipi aspektov
  - Množica tipov (naklonjenost, ljubezen, sovraštvo, spoštovanje)
  - Obtežena polarnost (pozitivno, negativno, nevtralno)
- Besedilo, ki vsebuje opis odnosa
  - Stavek ali celoten dokument

Analiza sentimenta

- Enostavnejša naloga
  - Ali je odnos v besedilu pozitiven ali negativen
- Zahtevnejša naloga
  - Rangirati odnos v besedilu (npr. od 1 do 10)
- Naprednejša naloga
  - Odkrivanje vira, cilja, kompleksnih tipov odnosov

Osnovni algoritem analize sentimenta

- *BoPang and Lillian Lee. 2004.A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.*
- Klasifikacija sentimenta s pomočjo filmskih kritik
- Ugotavljanje polarnosti
  - Ali je kritika filma (IMDB) pozitivna ali negativna
- Podatki: `http://www.cs.cornell.edu/people/pabo/movie-review-data`

Osnovni algoritem analize sentimenta

- Prilagodila sta ga Pang in Lee
- Leksikalna analiza
- Ugotavljanje značilk
- Klasifikacija z uporabo različnih klasifikatorjev
  - Naïve Bayes
  - MaxEnt (Max Entropy)
  - SVM (Support Vector Machine)

Leksikalna analiza

- Operacije nad datotekami HTML in XML
- Twitter (uporabniška imena, označevanja)
- Ohraniti velike črke
- Telefonske številke, datumi
- Čustveni simboli (angl. *Emotions*)

Ugotavljanje značilk za analizo sentimenta

- Kako ugotovimo negacije?
  - Ta film mi ni všeč
  - Ta filmi mi je zelo všeč
- Katere besede uporabiti?
  - Samo pridevnike
  - Vse besede (izkaže se, da uporaba vseh besed daje boljše rezultate)
- Metoda TF-IDF (angl. term frequency–inverse document frequency) in predstavitev s pomočjo vektorjev

Negacija

- Das, Sanjiv and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings of the Asia Pacific Finance Association Annual Conference (APFA).

  Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?

  Sentiment Classification using Machine Learning Techniques. EMNLP-2002,

  79-86.
- Dodajanje besede NOT_ vsaki besedi med negacijo in ločilom
  - didn't like this movie, but I
  - didn't NOT_like NOT_this NOT_movie but I

Naïve Bayes (ponovitev)

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in \text{položaj}} P(x_i|c_j)$$

$$\hat{P}(c_j) = \frac{\text{številoDokumentov}(C = c_j)}{\text{številoDokumentov}}$$

$$\hat{P}(b_i|c_j) = \frac{\text{število}(b_i, c_j)}{\sum_{b \in V} \text{število}(b, c_j)}$$

Binariziran Multinominal Naïve Bayes

Učenje

- Pojavitev besed je pomembnejša kot njihove frekvence
- Iz učnega korpusa se izlušči slovar (angl. *Vocabulary*)
- Izračunajo se $P(c_j)$
  - dok_j = vsi dokumenti razredac_j
  - $P(c_j) = \frac{|dok_j|}{|\text{število vseh dokumentov}|}$
- Izračunamo verjetnost $P(b_k|c_j)$
  - $besedilo_j$ = dokument, ki vsebuje vse dokumente dok_j
  - Za vsako besedo b_k iz slovarja
    - n_k = število pojavitev besede b_k v $besedilo_j$
    - $P(b_k|c_j) = \frac{n_k+\alpha}{n+\alpha|slovar|}$
    - n - število leksikalnih simbolov v razredu c_j

Binariziran Multinominal Naïve Bayes

Učenje

- Pojavitev besed je pomembnejša kot njihove frekvence
- Iz učnega korpusa se izlušči slovar (angl. *Vocabulary*)
- Izračunajo se $P(c_j)$
  - dok_j = vsi dokumenti razredac_j
  - $P(c_j) = \frac{|dok_j|}{|\text{število vseh dokumentov}|}$
- Izračunamo verjetnost $P(b_k|c_j)$
  - Odstranimo vse podvojene besede znotraj *dok*
    - Za vsako besedo b v dok_j ohranimo eno instanco besede b
  - $besedilo_j$ = dokument, ki vsebuje vse dokumente dok_j
  - Za vsako besedo b_k iz slovarja
    - n_k = število pojavitev besede b_k v $besedilo_j$
    - $P(b_k|c_j) = \frac{n_k+\alpha}{n+\alpha|slovar|}$
    - n - število leksikalnih simbolov v razredu c_j

Binariziran Multinominal Naïve Bayes

Klasifikacija

- Odstranimo vse podvojene besede v testnem dokumentu d
- Uporabimo enačbo za Multinominal Naïve Bayes

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in \text{položaj}} P(x_i|c_j)$$

Klasifikatorji

- Klasifikatorja MaxEnt in SVM ponavadi dajeta boljše rezultate kot Naïve Bayes
- Problemi
  - Včasih je težko iz besedila izluščiti značilke
    Če berete ta oglas samo zato ker je avto znamke Ferrari, si morate sliko avtomobila zalepiti na steno.
  - Včasih značilke kažejo na nasprotno klasifikacijo besedila
    Ta film bi lahko bil dober, nastopajo odlični igralci in tudi zgodba je odlična.

Leksikon sentimenta

- Kateri pomen sentimenta imajo besede
- Leksikoni
  - The General Inquirer (`http://www.wjh.harvard.edu/~inquirer`)
    Prosto dostopen za raziskovanje
  - Linguistic Inquiry and Word Count (`http://www.liwc.net/`)
    Cena $30 oz. $90
  - MPQA Subjectivity Cues Lexicon
    (`http://www.cs.pitt.edu/mpqa/subj_lexicon.html`)
    GNU GPL
  - Bing Liu Opinion Lexicon
    (`http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`)
  - SentiWordNet
    (`http://sentiwordnet.isti.cnr.it/`)
    Vsem WordNet sinsetom je dodeljena stopnja pozitivnosti, negativnosti in nevtralnosti

Leksikoni sentimenta

- Želimo zgraditi lasten leksikon sentimenta
- Delno nadzorovano učenje
  - Imamo na voljo malo količino informacij (nekaj označenih primerov ali ročno zgrajenih vzorcev)

Intuicija za identifikacijo besedne polarizacije

- Intuicija (Hatzivassiloglou in McKeown)
  - Pridevniška povezanost z besedo "and" ima enako polarizacijo (corrupt and brutal)
  - Pridevniška povezanost z besedo "but" nima enako polarizacijo (fair but brutal)
- 1. korak: ročno označimo "semensko" množico pridevnikov
- 2. korak: razširimo semensko množico s povezanimi pridevniki
  - kako pogosto se pojavljajo v povezavi z besedo "and" in besedo "but"
- 3. korak: nadzorovano klasificiranje glede na "podobnost polarnosti"
- 4. korak: združevanje besed v dve množici (pozitivno/negativno)

Algoritem ki temelji na polariteti fraz

- *Turney (2002): Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*
- Glede na kritike, pridobivanje fraznega leksikona
- Učenje polaritete fraz
- Rangiranje kritik, glede na povprečnost polaritete fraz

Algoritem Turney

- Pridobivanje dvo-besednih fraz skupaj s pridevniki
  - Uporabimo samodejno oblikoslovno označevanje (angl. *part of speech tagging*)
- Kako merimo polariteto fraz v katerih se pojavi beseda "odlično"?
- Kako merimo polariteto fraz v katerih se pojavi beseda "slabo"?

Pointwise Mutual Information

- Medsebojne informacije (angl. *Mutual information*) med dvema naključnima spremenljivkama X in Y

$$I(X,Y) = \sum_x \sum_y P(x,y) \cdot log_2 \frac{P(x,y)}{P(x) \cdot P(y)}$$

- Točkovna medsebojna informacija (angl. *Pointwise mutual information*)
  - Kako pogosto se pojavita dogodka x in y skupaj v primerjavi z njihovo neodvisno pojavitvijo

$$PMI(X,Y) = log_2 \frac{P(x,y)}{P(x) \cdot P(y)}$$

Pointwise Mutual Information

- Točkovna medsebojna informacija (angl. *Pointwise mutual information*)
  - Kako pogosto se pojavita dogodka x in y skupaj v primerjavi z njihovo neodvisno pojavitvijo

$$PMI(X, Y) = log_2 \frac{P(x, y)}{P(x) \cdot P(y)}$$

- PMI med dvema besedama
  - Kako pogosto se pojavita besedi x in y skupaj v primerjavi z njihovo neodvisno pojavitvijo

$$PMI(beseda_1, beseda_2) = log_2 \frac{P(beseda_1, beseda_2)}{P(beseda_1) \cdot P(beseda_2)}$$

Kako ocenimo PMI

- Lahko uporabimo spletni iskalnik
  - P(*beseda*) ocenimo s številom zadetkov $\frac{zadetki(beseda)}{N}$
  - P($beseda_1$,$beseda_2$) ocenimo s številom zadetkov $\frac{zadetki(beseda_1 \ \ BLIZU \ \ beseda_2)}{N^2}$
  - Primer (Google): apple AROUND(4) iphone
  - N - $zadetki(beseda_1) + zadetki(beseda_2)$

$$PMI(beseda_1, beseda_2) = log_2 \frac{\frac{1}{N^2} \cdot zadetki(beseda_1 \ \ BLIZU \ \ beseda_2)}{\frac{1}{N} \cdot zadetki(beseda_1) \cdot \frac{1}{N} \cdot zadetki(beseda_2)}$$

Fraza v relaciji z besedama *dobro* in *slabo*

Ali se fraza pojavlja pogosteje z besedo *dobro* ali z besedo *slabo*?

$$Polariteta(fraza) = PMI(fraza, dobro) - PMI(fraza, slabo) =$$

$$= log_2 \frac{zadetki(fraza\ \ BLIZU\ \ dobro)}{zadetki(fraza) \cdot zadetki(dobro)} - log_2 \frac{zadetki(fraza\ \ BLIZU\ \ slabo)}{zadetki(fraza) \cdot zadetki(slabo)}$$

$$= log_2 \frac{zadetki(fraza\ \ BLIZU\ \ dobro)}{zadetki(fraza) \cdot zadetki(dobro)} \frac{zadetki(fraza) \cdot zadetki(slabo)}{zadetki(fraza\ \ BLIZU\ \ slabo)}$$

$$= log_2 \frac{zadetki(fraza\ \ BLIZU\ \ dobro) \cdot zadetki(slabo)}{zadetki(dobro) \cdot zadetki(fraza\ \ BLIZU\ \ slabo)}$$

Algoritem Turney

- Uspešnejši od osnovnega algoritma
- Uporablja fraze namesto besed
- Uči se na domensko specifičnih informacijah
- Na podoben način lahko učimo tudi WordNet

Analiza sentimenta

- Naprednejše metode (ugotavljanje aspektov, atributov in cilja sentimenta)
- V osnovi je modelirana kot klasifikacija ali regresija
- Značilnosti:
  - Negacija je pomembna
  - Uporaba vseh besed s pomočjo Naïve Bayes daje boljše rezultate
  - Iskanje podmnožic besed pomaga v določenih nalogah
    - Ročno zgrajeni leksikoni polarnosti
    - Uporaba semen in delno-nadzorovano učenje za izgradnjo leksikona

Viri

- Dan Jurafsky, Chris Manning, Natural Language Processing http://web.stanford.edu/∼jurafsky/NLPCourseraSlides.html
- Park K., Hong J., and Kim W., A Methodology Combining Cosine Similarity with Classifier for Text Classification. Applied Artificial Intelligence, 2020. 34(5): 396-411.

Diskriminatorni klasifikatorji

(angl. Discriminative classifiers)

Uvod

- Do sedaj smo obravnavali “generativne modele”
  - Jezikovni modeli, Naïve Bayes
- Bolj uporabni so diskriminatorni (pogojni) modeli
  - So bolj točni
  - Omogočajo enostavno vključitev jezikovno pomembnih značilk
  - Omogočajo samodejno izgradnjo jezikovno neodvisnih modelov

Generativni in diskriminatorni modeli

- Klasifikatorji se v fazi sklepanja odločajo glede na verjetnost $P(c|d)$
- Generativni modeli
  - Uporaba pogojne gostote $P(d|c)$ skupaj s priorno (predhodno) verjetnostjo $P(c)$
  - Uporaba Bayesovega pravila za izračun posteriorne verjetnosti $P(c|d) = \frac{P(d|c)P(c)}{P(d)}$
  - n-gram modeli, klasifikator Naïve Bayes, skriti Markovi modeli,verjetnostne kontekstno proste gramatike, modeli v strojnem prevajanju
- Diskriminatorni modeli
  - Neposreden izračun verjetnosti $P(c|d)$
  - Maksimiramo pogojno verjetnost
  - Težje narediti
  - Izboljša zmogljivost klasifikatorja - Naïve Bayes $73,6 \Rightarrow 76,1$ (2,5%)
  - Logistična regresija, Perceptron, SVM, ...

Značilke besedila za diskriminatorni model

- Značilka besedila f predstavlja elementarne vidike povezanosti videnega besedila d z razredom c.
- Značilko predstavimo s pomočjo funkcije, ki vrne realno vrednost na določenem intervalu: $f : CxD \rightarrow \mathbb{R}$.
- Primeri:
  - $f_1(c, d) \equiv [c = \text{MESTO} \wedge b_{i-1} = \text{"v"} \wedge \ velikaZačetnica(b_i)]$
  - $f_2(c, d) \equiv [c = \text{DRŽAVA} \wedge b_{i-2} = \text{"glavno"} \wedge b_{i-1} = \text{"mesto"} \wedge velikaZačetnica(b_i) \wedge b_{i+1} = \text{"je"}]$
  - Glavno mesto Slovenije je Ljubljana.
  - Potujem v Pariz.
- Model vsaki značilki dodeli utež, ki je lahko pozitivna (značilka je pravilna) ali negativna (značilka ni pravilna).

Značilke besedila za diskriminatorni model

- Empirično štetje (pričakovanje) značilk:
  empirični $E(f_i) = \sum_{(c,d) \in observed(C,D)} f_i(c,d)$
- Model za pričakovanje značilk
  $E(f_i) = \sum_{(c,d) \in observed(C,D)} P(c,d) f_i(c,d)$
- Splošna predstavitev značilk
  $f_i(c,d) \equiv [\Phi(d) \wedge c = c_j] \rightarrow [vrednost\ 0\ ali\ 1]$
  - Funkcija $\Phi(d)$ vrne logično vrednost
  - Kateremu razredu pripadajo podatki $c = c_j$

Linearni klasifikator, ki temelji na značilkah

Linearni klasifikator v fazi sklepanja

- Linearna funkcija za množico značilk (f_i) in razred c
- Vsaki značilki f_i dodaj utež λ_i
- Obravnavamo vse razrede za podatek d
- Za par (c, d), značilke glasujejo z uporabo uteži: $glasovanje(c) = \sum \lambda_i f_i(c, d)$
- Izbere se razred z največjo vrednostjo glasovanja: $c = \arg\max_{c \in C} \sum \lambda_i f_i(c, d)$

Linearni klasifikator, ki temelji na značilkah

- Kako nastaviti uteži?
- Perceptorn: poišče trenutno napačno obravnavane primere in spremeni uteži v smeri pravilne klasifikacije
- Metode na osnovi oddaljenosti (Support Vector Machines)

Linearni klasifikator, ki temelji na značilkah

- Eksponentni (loglinearni,maxent,logistični,Gibbs) model
- Funkcija *exp* spremeni rezultat v pozitivno vrednost (uteži imajo lahko negativne vrednosti)
- Verjetnostni model normaliziramo glede na linearno kombinacijo $\sum \lambda_i f_i(c, d)$:

$$P(c|d, \lambda) = \frac{exp \sum_i \lambda_i f_i(c, d)}{\sum_{c'} exp \sum_i \lambda_i f_i(c', d)}$$

- Uteži λ_i izberemo tako, da maksimiramo pogojno verjetnost podatkov na podanem modelu

Primer

- $P(\text{MESTO}|\text{v Maribor}) = \frac{e^{0,8}}{e^{0,8}+e^{0,6}} = 0,55$
- $P(\text{DRŽAVA}|\text{v Maribor}) = \frac{e^{0,6}}{e^{0,8}+e^{0,6}} = 0,45$
- $f_1(c,d) \equiv [c = \text{MESTO} \wedge b_{i-1} = \text{"v"} \wedge \ velikaZačetnica(b_i)] => \sum \lambda_1 f_1(c,d) = 0,8$
- $f_2(c,d) \equiv [c = \text{DRŽAVA} \wedge b_{i-2} = \text{"glavno"} \wedge b_{i-1} = \text{"mesto"} \wedge velikaZačetnica(b_i) \wedge b_{i+1} = \text{"je"}] => \sum \lambda_2 f_2(c,d) = 0,6$

Izgradnja modela Maxent

- Definiramo značilke
  - Besede
  - Beseda, ki vsebuje števila
  - Besede z določenimi končnicami (npr. “ing”, “s”)
- Značilke označimo z unikatnimi oznakami
  - Vsak podatek je lahko povezan z več značilkami $\Phi(d)$
  - Vsaka značilka vrne realno vrednost $f_i(c, d) \equiv [\Phi(d) \wedge c = c_j]$
- Značilke se dodajajo v času razvoja modela
  - Model preizkusimo na razvojni množici
  - Poskusimo dodati oz. popraviti obstoječe značilke
  - Ta postopek iterativno ponavljamo

Eksponentni verjetnostni model

Za podan model izberemo take vrednosti parametrov, da maksimirajo pogojno verjetnost modela.

$$\log P(C|D,\lambda) = \sum_{(c,d)\in(C,D)} \log P(c|d,\lambda) =$$

$$\sum_{(c,d)\in(C,D)} \log \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c',d)} =$$

$$\sum_{(c,d)\in(C,D)} \log \exp \sum_i \lambda_i f_i(c,d) - \sum_{(c,d)\in(C,D)} \log \sum_{c'} \exp \sum_i \lambda_i f_i(c',d) =$$

$$N(\lambda) - M(\lambda)$$

Odvod števca

$$\frac{\partial N(\lambda)}{\partial \lambda_i} = \frac{\partial \sum_{(c,d)\in(C,D)} \log \exp \sum_i \lambda_i f_i(c,d)}{\partial \lambda_i} =$$

$$\frac{\partial \sum_{(c,d)\in(C,D)} \sum_i \lambda_i f_i(c,d)}{\partial \lambda_i} =$$

$$\sum_{(c,d)\in(C,D)} \frac{\partial \sum_i \lambda_i f_i(c,d)}{\partial \lambda_i} = \sum_{(c,d)\in(C,D)} f_i(c,d) =$$

empirični *števec*(f_i, C)

Odvod imenovalca

$log'(x) = \frac{1}{x}$

$$\frac{\partial M(\lambda)}{\partial \lambda_i} = \frac{\partial \sum_{(c,d)\in(C,D)} \log \sum_{c'} \exp \sum_i \lambda_i f_i(c', d)}{\partial \lambda_i}$$

$(exp(x) * f(x))' = exp(x) * f(x)'$

$$= \sum_{(c,d)\in(C,D)} \frac{1}{\sum_{c''} \exp \sum_i \lambda_i f_i(c'', d)} \frac{\partial \sum_{c'} \exp \sum_i \lambda_i f_i(c', d)}{\partial \lambda_i}$$

$$= \sum_{(c,d)\in(C,D)} \frac{1}{\sum_{c''} \exp \sum_i \lambda_i f_i(c'', d)} \sum_{c'} \frac{\exp \sum_i \lambda_i f_i(c', d)}{1} \frac{\partial \sum_i \lambda_i f_i(c', d)}{\partial \lambda_i}$$

$$= \sum_{(c,d)\in(C,D)} \sum_{c'} \frac{\exp \sum_i \lambda_i f_i(c', d)}{\sum_{c''} \exp \sum_i \lambda_i f_i(c'', d)} \frac{\partial \sum_i \lambda_i f_i(c', d)}{\partial \lambda_i}$$

$$= \sum_{(c,d)\in(C,D)} \sum_{c'} P(c'|d, \lambda) f_i(c', d) = \textbf{napovedan } \check{s}tevec(f_i, \lambda)$$

Odvod

- $$\frac{\partial \log P(C|D,\lambda)}{\partial \lambda_i} =$$

  empirični *števec*(f_i, C) − **napovedan** *števec*(f_i, λ)

- Optimalni parametri so tisti, pri katerih za vsako značilko velja, da je **napovedano pričakovanje** enako **empiričnemu pričakovanju**.
- Te modele imenujemo tudi modeli maksimalne entropije (angl. maximum entropy model). Razlog temu je, da moramo najti model z maksimalno entropijo oz. zadostiti naslednje omejitve:

$$E_p(f_j) = E_{\tilde{p}}(f_j), \forall j$$

Optimalni parametri

- Želimo izbrati parametre $\lambda_1, \lambda_2, \lambda_3, ...$, ki maksimirajo pogojno logaritemsko verjetnost učne množice

$$CLogVer(D) = \sum_{1}^{n} \log P(c_i|d_i)$$

- Da bi to lahko naredili moramo znati izračunati funkcijsko vrednost in parcialne odvode (gradient)
- Poiščemo optimalne parameter s pomočjo določene optimizacijske metode

Viri

- Dan Jurafsky, Chris Manning, Natural Language Processing `http://web.stanford.edu/~jurafsky/NLPCourseraSlides.html`
- Asist. dr. Igor Locatelli, mag. farm., Logistična regresija, 2014
- Prof. Jurij F. Tasič, Osnove Linearnih klasifikacijskih modelov, 2013

Strojno prevajanje

Vsebina

- Strojno prevajanje
- Leksikalna dvoumnost
- Različni vrstni red besed
- Skladenjske razlike
- Ločljivost samostalnikov
- Klasični pristopi za prevajanje s pravili
- Statistično strojno prevajanje (angl. *Statistical machine translation*)

Strojno prevajanje

- *Machine Translation* (MT)
- Prevajanje je zahtevno in ustvarjalno dejanje.
- Postopek, pri katerem računalniški program analizira besedilo in brez posredovanja človeka ustvari ciljno besedilo.
- Sistemi za strojno prevajanje vključujejo:
  - eno ali večjezične leksikone,
  - programe za morfološko analizo in sintezo,
  - programe za sintaktično analizo in sintezo,
  - programe za razreševanje večpomenskosti,
  - programe za prepoznavanje večbesednih semantičnih enot,
  - itd.

Strojno prevajanje

- Strojno prevajanje lahko v nekaterih primerih prevajalcu olajša delo ali pa ga celo popolnoma nadomesti:
  - grob prevod, ki ga kasneje pregleda in popravi prevajalec,
  - osnutek, ki služi kot pomoč pri prevajanju,
  - določene besedilne vrste, pri katerih je izrazje močno omejeno (vremenska napoved, navodila za uporabo, računalniški programi, zdravniška poročila, itd).
- Strojni prevajalniki so sposobni prepoznati kontekst, frazeme in idiome v izvirnem jeziku ter ustvariti koheziven in razumljiv prevod.
- Strojni prevodi uradnih listin in pravnih aktov so tako razumljivejši in pravilnejši kot pri govorjenem jeziku.

Različen vrstni red besed

- angleški vrstni red: subject - verb - object
- Primer:
  book the flight -> rezerviraj
  read the book -> knjiga

Leksikalna dvoumnost

- V slovenščini ne obstaja točno določeno zaporedje stavčnih členov, velja pa t.i. členitev po aktualnosti: najprej v stavku nastopajo stavčni členi, ki nosijo že znano informacijo, na koncu pa tisti, ki povedo prejemniku nek nov podatek.
- slovenski vrstni red: besedni vrstni red se prilagaja poudarjeni besedi
- Primer:
  - Vrstni red besed v stavku seveda obstaja.
  - Vrstni red besed seveda v stavku obstaja.
  - Seveda vrstni red besed v stavku obstaja.
  - V stavku seveda obstaja vrstni red besed.
  - Seveda obstaja v stavku vrstni red besed.
  - Seveda obstaja vrstni red besed v stavku.

Različen vrstni red besed

- Nobena od teh možnosti ni napačna, kako sodijo v kontekst, je pa drugo.
- Še najmanj zaznamovana je prva poved.
- V drugi povedi poudarjamo samoumevnost obstanka vrstnega reda besed (z besedo seveda).
- V tretji želimo poudariti, da obstaja.
- V četrti poudarjamo vrstni red besed.
- V peti in šesti pa poudarek velja tako na vrstnem redu besed, kot temu, da obstaja.

Skladenjske razlike

- Tisto, kar hočemo poudariti, damo ponavadi na konec, včasih pa na začetek.
- Primeri:
  - Spoznali smo tri fante med vojaki, ki so pravkar imeli počitek. Kdo je imel počitek? Trije fantje ali vojaki?
  - Lepa dekleta ljubijo barabe. Kdo koga ljubi? Lepa dekleta barabe ali barabe lepa dekleta?
- Primeri pri prevajanju (Google translate in popravljena imena):
  - Petra ljubi Marka -> Petra loves Marko.
  - Marka ljubi Petra -> Marko loves Petra.

Ločljivost samostalnikov

The computer outputs the data, **it** is fast. Računalnik izpiše podatke, **je** hiter.

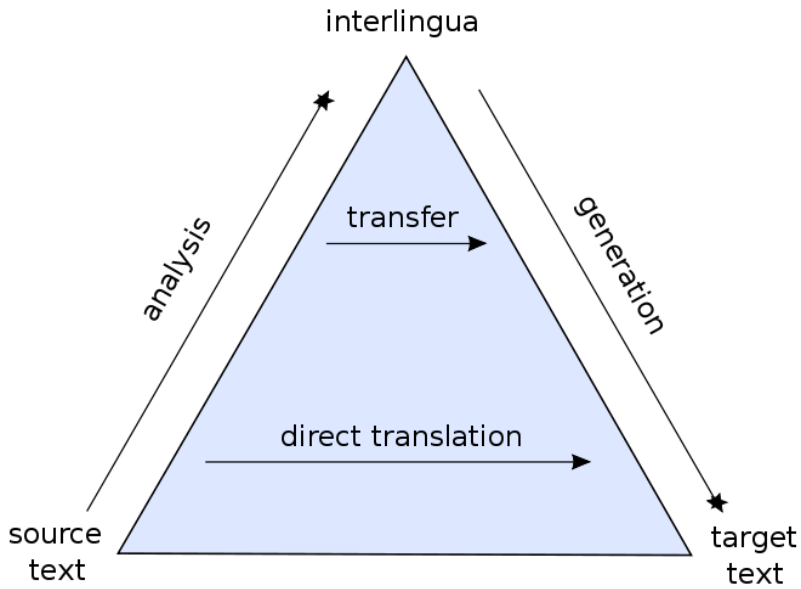
- **it** -> the computer ali the data?
- Vidi se, da je mišljeno za računalnik, zato se prevede kot "je".

The computer outputs the data, **it** is stored in ASCII. Računalnik izpiše podatke, **so** shranjeni v ASCII.

- **it** -> the computer ali the data?
- Vidi se, da je mišljeno za podatke, zato se prevede kot ”so“.

Klasični pristopi za prevajanje s pravili

- Neposredno strojno prevajanje (angl. *Dictionary-Based method*)
- Transferno strojno prevajanje (angl. *Transfer-Based method*)
- Medjezikovno strojno prevajanje (angl. *Interlingua-Based method*)



Neposredno strojno prevajanje

- Prevaja se besedo po besedo.
- Zelo malo analize izvornega besedila.
  - Brez skladenjske ali semantične analize.
- Zanaša se na dvojezični slovar.
  - Za vsako besedo v izvornem jeziku, slovar določa množico pravil za prevod te besede.
- Primer:
  - how much -> if (prejšnjaBeseda == "how") return "koliko";
  - as much -> if (prejšnjaBeseda == "as") return "toliko";

Neposredno strojno prevajanje

Slabosti:

- Pomanjkanje analize izvornega besedila.
  - Težko oz. nemogoče je zajeti dolge prerazporeditve, ker nimamo nobenega skladenjskega znanja.
- Primer:
  - angleščina: Sources said that IBM bought Lotus yesterday.
  - japonščina: Sources yesterday IBM Lotus bought that said.
- Besede so prevedene brez znanja o ločevanju.
- Primer:
  - They said that I like ice-cream.
  - They like that ice-cream.

Transferno strojno prevajanje

3 faze prevajanja:

- Analiza
  - Analiziranje izvornega stavka, npr. sintaktično (skladenjsko) drevo
- Prenos
  - Preoblikujemo razčlenitveno drevo izvornega stavka v razčlenitveno drevo ciljnega stavka tako, da uporabimo množico pravil, ampak ta pravila so zgrajena iz razčlenitvenega drevesa izvornega stavka.
- Ustvarjanje
  - S pomočjo razčlenitvenega drevesa ciljnega stavka ustvarimo ciljni stavek.

Medjezikovno strojno prevajanje

2 fazi prevajanja:

- Analiza
  - Analiziramo izvorni stavek v predstavitev njegovega pomena, pri čemer upamo na to, da je neodvisna od jezika (angl. *language-independed representation*).
- Ustvarjanje
  - Preoblikujemo predstavitev pomena v izhodni stavek.
- Če hočemo zgraditi sistem za prevajanje, ki prevaja med N jeziki, moramo razviti N sistemov za analizo in ustvarjanje.

Koncepti pri medjezikovnem prevajanju

- Nemščina ima dve besedi za steno -> notranja in zunanja stena
- Japonščina ima dve besedi za brata -> starejši in mlajši brat
- Španščina ima dve besedi za nogo -> človeška in živalska noga
- Vsak jezik ima svoje koncepte, zato je potrebno biti previden pri grajenju sistema za prevajanje -> zaradi tega ta pristop ni enostaven

Statistično strojno prevajanje

- *Statistical Machine Translation* (SMT)
- Je vrsta strojnega prevajanja, ki temelji na večji količini vzporednih besedil, iz katerih se s statističnimi algoritmi izračunavajo verjetnosti za posamezne jezikovne enote.
- Osnovna ideja je uporaba vzporednega korpusa za učenje sistema za prevajanje.

"..one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When we look at an article in Russian, we could says that it is written in English, but it has been coded in some strange symbols. We will now proceed to decode."(Warren Waver, 1949)

Dekodirnik (ali iskalni algoritem)

- Ideja: uporaba šumnega kanala za prevajanje
- Cilj: prevajalni sistem iz slovenščine v angleščino
- Ima 2 modela:
  - Jezikovni model $P(e)$
  - Prevajalni model $P(s|e)$
- Pravilo Bayes:

$$P(e|s) = \frac{P(e,s)}{P(s)} = \frac{P(e) \cdot P(s|e)}{P(s)}$$

- Iskalni algoritem:

$$\textit{dekodirnik} = \arg\max_e P(e|s) = \arg\max_e P(e) \cdot P(s|e)$$

Prevajalni model

- "backwards"model -> če prevajamo iz angleščine v slovenščino želimo izračunati $P(s|e)$, v bistvu pa računamo $P(e|s)$ oz. verjetnost, da je slovensko besedilo prevod angleškega besedila.
- Učimo ga na vzporednem korpusu.
- Jezikovni model lahko dopolni pomanjkljivosti prevajalnega modela.

Primer

Prevajamo naslednjo poved iz slovenščine v angleščino z uporabo prevajalnega modela:

Lačen sem. $P(slo|ang)$
$P(\text{Lačen sem}|\text{What hunger have}) = 0,000014$
$P(\text{Lačen sem}|\text{Hungry I am}) = 0,000001$
$P(\text{Lačen sem}|\text{I am hungry}) = 0,0000015$
$P(\text{Lačen sem}|\mathbf{Have\ I\ hunger}) = \mathbf{0,00002}$

Primer

Prevajamo naslednjo poved iz slovenščine v angleščino z uporabo jezikovnega in prevajalnega modela:

Lačen sem.
$P(ang) \cdot P(slo|ang)$
$P(\text{Lačen sem}|\text{What hunger have}) = 0,000001 \cdot 0,000014 = 1,4e^{-11}$
$P(\text{Lačen sem}|\text{Hungry I am}) = 0,0000014 \cdot 0,000001 = 1,4e^{-12}$
$P(\text{Lačen sem}|\textbf{I am hungry}) = 0,0001 \cdot 0,0000015 = \mathbf{1,5e^{-10}}$
$P(\text{Lačen sem}|\text{Have I hunger}) = 0,00000098 \cdot 0,00002 = 1,96e^{-11}$

Prevajalni model IBM

- Modela IBM 1 in 2 (vse skupaj jih je že 5)
  - Prva generacija statističnega strojnega prevajanja.
- Modeli, ki temeljijo na frazah
  - Druga generacija statističnega strojnega prevajanja,
  - boljši kot modeli IBM,
  - te modele je uporabljal prevajalnik Google.

Model IBM 1

- Se več ne uporablja za prevajanje, ampak za poravnavo.
- Poravnava a nam pove iz katere angleške besede so nastale slovenske besede.
  e = the dog barks
  s = pes laja
  $l = 3,\ m = 2$
- Angleški stavek ima l besed $(e_1, ..., e_l)$, medtem ko ima slovenski stavek m besed $(s_1, ..., s_m)$.
- Formalni zapis: $a \in \{a_1, a_2, ..., a_m\}$, kjer je $a_j \in \{0, 1, ..., l\}$
  Za vsako slovensko besedo določimo angleško besedo
- Število vseh možnih poravnav je $(l + 1)^m$.

Primer 1

$\mathrm{e} = \mathrm{the}_1\ \mathrm{dog}_2\ \mathrm{barks}_3$

$\mathrm{s} = \mathrm{pes}_1\ \mathrm{laja}_2$

$a \in \{a_1, a_2\}$, kjer je $a_1 = 2, a_2 = 3$.

$\mathrm{s} = \mathrm{pes}_1\ \mathrm{laja}_2$

$\mathrm{e} = \mathrm{the}_1\ \mathrm{dog}_2\ \mathrm{barks}_3$

$a \in \{a_1, a_2, a_3\}$, kjer je $a_1 = 0, a_2 = 1, a_3 = 2$.

Primer 2

$e = \text{and}_1\ \text{the}_2\ \text{program}_3\ \text{has}_4\ \text{been}_5\ \text{implemented}_6$

$s = \text{in}_1\ \text{program}_2\ \text{je}_3\ \text{bil}_4\ \text{implementiran}_5$

$l = 6,\ \ m = 5$

- Pravilna poravnava je $\{1, 3, 4, 5, 6\}$
  - Prva beseda v slovenščini je poravnana s prvo besedo v angleščini, druga beseda s tretjo besedo, itd.
- Napačna poravnava je $\{1, 1, 1, 1, 1\}$
  - Vsaka beseda v slovenščini je poravnana s prvo besedo v angleščini.

Verjetnost poravnave

- Verjetnost za poravnavo je $P(s, a|e, m)$, kjer je s slovenski stavek, a je poravnava, e je angleški stavek in m je število besed v slovenščini
- Razgradimo na dva verjetnostna modela:
  - $P(a|e, m)$: porazdelitev vseh možnih poravnav; $(l + 1)^m$ vrednosti za a
  - $P(s|a, e, m)$: pogojuje na poravnavo, angleški stavek in število besed v slovenščini
  - A je množica vseh poravnav

$$P(s, a|e, m) = P(a|e, m) \cdot P(s|a, e, m)$$

$$P(s|e, m) = \sum_{a \in A} P(a|e, m) \cdot P(s|a, e, m)$$

Verjetnost poravnave

- Ko imamo model $P(s, a|e, m)$, izračunamo za vsako poravnavo verjetnost:

$$P(a|s, e, m) = \frac{P(s, a|e, m)}{\sum_{a \in A} P(s, a|e, m)}$$

- Za podan par s, e lahko izračunamo najverjetnejšo poravnavo:

$$a^* = \arg\max_a P(a|s, e, m)$$

Modeli

$P(a|e,m) = \frac{1}{(l+1)^m}$

- vsaka poravnava je enako verjetna: slovenska beseda se lahko poravna z vsako besedo iz angleške besede
- t - ocenjena verjetnost poravnave besed (angl. translation parameter)

$P(s|a,e,m) = \prod_{j=1}^{m} t(s_j|e_{a_j})$

$e = the_1\, dog_2\, barks_3$

$s = pes_{s_1}\, laja_{s_2}$

$P(\text{ pes laja } | \{2,3\}, \text{ the dog barks }, 2) = t(pes|dog) \cdot t(laja|barks)$

Primer 1

e = the dog barks
s = pes laja

- Koliko je verjetnost $P(s|a, e, m)$ za zgornji primer?

$t(pes|the) = 0,3 \quad t(laja|the) = 0,4$
$t(pes|dog) = 0,8 \quad t(laja|dog) = 0,3$
$t(pes|barks) = 0,1 \quad t(laja|barks) = 0,7$

$$P(s|a, e, m) = t(pes|dog) \cdot t(laja|barks) = 0,8 \cdot 0,7 = 0,56$$

$$P(s, a|e, m) = P(a|e, m) \cdot P(s|a, e, m) = \frac{1}{(l+1)^m} \cdot \prod_{j=1}^{m} t(s_j|e_{a_j})$$

Primer 2

e = and the program has been implemented
s = in program je bil implementiran

$l = 6,\ m = 5$
$a = \{1, 3, 4, 5, 6\}$

Primer 2

$t(in|and) = 0,8 \quad t(in|the) = 0,1 \quad t(in|program) = 0,2$
$t(in|has) = 0,4 \quad t(in|been) = 0,3 \quad t(in|implemented) = 0,2$
$t(program|and) = 0,3 \quad t(program|the) = 0,1$
$t(program|program) = 0,7 \quad t(program|has) = 0,5$
$t(program|been) = 0,3 \quad t(program|implemented) = 0,4$
$t(je|and) = 0,5 \quad t(je|the) = 0,2 \quad t(je|program) = 0,3$
$t(je|has) = 0,9 \quad t(je|been) = 0,1 \quad t(je|implemented) = 0,4$
$t(bil|and) = 0,2 \quad t(bil|the) = 0,1 \quad t(bil|program) = 0,3$
$t(bil|has) = 0,6 \quad t(bil|been) = 0,8 \quad t(bil|implemented) = 0,4$
$t(implementiran|and) = 0,5 \quad t(implementiran|the) = 0,2$
$t(implementiran|program) = 0,3 \quad t(implementiran|has) = 0,4$
$t(implementiran|been) = 0,1 \quad t(implementiran|implemented) = 0,9$

Primer 2

$t(in|and) = 0,8$
$t(program|program) = 0,7$
$t(je|has) = 0,9$
$t(bil|been) = 0,8$
$t(implementiran|implemented) = 0,9$

Rešitev:

$P(s|a,e,m) = t(in|and) \cdot t(program|program) \cdot t(je|has) \cdot t(bil|been) \cdot t(implementiran|implemented) = 0,36288$

$$P(s,a|e,m) = P(a|e,m) \cdot P(s|a,e,m) = \frac{1}{(l+1)^m} \cdot \prod_{j=1}^{m} t(s_j|e_{a_j})$$

Model IBM 2

- Vpeljuje parametre poravnave in popačenja
  $q(i|j, l, m)$ = verjetnost, da je j-ta slovenska beseda povezana z i-to angleško besedo. Pri tem sta dollžini angleškega in slovenskaga stavka l in m.
- Definiramo:
  $p(a|e, m) = \prod_{j=1}^{m} q(a_j|j, l, m)$; kjer je $a = \{a_1, ..., a_m\}$
- Dobimo:
  $p(s, a|e, m) = \prod_{i=1}^{m} q(a_j|j, l, m)t(s_j, e_{a_j})$

Primer

e = And the program has been implemented
s = In program je bil implementiran
a = $\{1, 3, 4, 5, 6\}$

$p(a|e, 5) =$
$q(1|1, 6, 5) \cdot q(3|2, 6, 5) \cdot q(4|3, 6, 5) \cdot q(5|4, 6, 5) \cdot q(6|5, 6, 5)$

$p(s|a, e, 5) = t(In|And) \cdot t(program|program) \cdot t(je|has) \cdot$
$t(bil|been) \cdot t(implementiran|implemented)$

Končni model

- 1. korak:
  $p(a|e,m) = \prod_{j=1}^{m} q(a|j,l,m)$
- 2. korak: Uporabimo verjetnosti besed
  $p(s|a,e,m) = \prod_{j=1}^{m} t(s_j|e_{a_j})$
- Končni model:
  $p(s,a|e,m) = p(a|e,m) \cdot p(s|a,e,m) =$
  $\prod_{j=1}^{m} q(a_j|j,l,m)t(s_j|e_{a_j})$

Poravnava

- Ko imamo parametre q in t, lahko zelo enostavno določimo najbolj verjetno poravnavo.
- Za par: $e_1, e_2, ..., e_m$ in $s_1, s_2, ..., s_l$ lahko izračunamo:
  $a_j = arg\ max_{a \in \{0...l\}} q(a|j, l, m) * t(s_j|e_{a_j});\ j = 1...m$

Primer

e = NULL And the program has been implemented
s = In program je bil implementiran

Izračun poravnave za besedo “je”:
$NULL : q(0|3,6,5) \cdot t(je|NULL)$
$And : q(1|3,6,5) \cdot t(je|And)$
$the : q(2|3,6,5) \cdot t(je|the)$
...
$implemented : q(6|3,6,5) \cdot t(je|implemented)$

- q - verjetnost pozicije
- t - verjetnost besede
- vrednosti q in t dobimo na osnovi učnega korpusa

Model

- Učna množica ($\{e^{(1)}, s^{(1)}, a^{(1)}\}$, $\{e^{(2)}, s^{(2)}, a^{(2)}\}$, ...):
  $e^{(100)}$ = And the program has been implemented
  $s^{(100)}$ = In program je bil implementiran
  $a^{(100)} = \{1, 3, 4, 5, 6\}$
- $t_{MLE}(s|e) = \frac{\text{število}(e,s)}{\text{število}(e)}$
- $q_{MLE}(j, i|l, m) = \frac{\text{število}(j|i,m,n)}{\text{število}(i,l,m)}$

Viri

- Michael Collins, Natural Language Processing
  https://class.coursera.org/nlangp-001
  https://archive.org/details/academictorrents_f99e7184fca947ee8f77901679e171fcadbf82e7
- Transfer-based MT
  https://en.wikipedia.org/wiki/Transfer-based_machine_translation

Ekstrakcija informacij in prepoznavanje imenskih entitet

(angl. Information Extraction and Named Entity Recognition)

Ekstrakcija informacij

- Začela se je razvijati ob pojavitvi sistemov za razpoznavanje imenskih entitet (leta 1970).
- Ekstrahirani podatki omogočajo nove načine poizvedovanja, organizacije, analize in predstavitve podatkov (biomedicinska domena).
- Zbiranje informacij iz različnih delov besedila
- Najti in razumeti določene dele besedila
- Cilj ekstrakcije informacij je:
  - Pridobiti strukturirane podatke iz nestrukturiranih ali pol-strukturiranih podatkovnih virov.
  - Postaviti informacije v natančno pomensko obliko, ki omogoča računalniškim algoritmom nadaljnja sklepanja.

Ekstrakcija informacij

- Naloga sistemi za ekstrakcijo informacij je, da ekstrahirajo jasne dejanske informacije (kdo, kaj, komu, kdaj, kje).
- Primer
  - Iz besedila ugotovi, kdo je predavatelj, kaj predava, kje pradava, kdaj predava.
  - predavatej(“Janez Novak”, “Slovenščina”,“Maribor”,“Ponedeljek”,“Poletni semester”)

Preprosta ekstrakcija informacij

- Obstaja v različni programski opremi.
- Odjemalec elektronske pošte, na osnovi vsebine ponuja določene aktivnost. Ko razpozna datum, nam ponudi ustvarjanje dogodka.
- Pri spletnem iskanju, imamo ponujene informacije glede na vsebino povpraševanja (npr. kje se nahaja iskano mesto).
- Večina preprostih ekstrakcij informacij uporablja regularne izraze.

Prepoznavanje imenskih entitet

- Zelo pomembno opravilo pri ekstrakciji informacij je prepoznavanje imenskih entitet.
- Potrebno je **poiskati** in **klasificirati** imena znotraj besedila.
  - S pomočjo leksikalne analize ugotovimo imena.
  - Vsa imena klasificiramo npr.: ime človeka, organizacije, lokacije itd.
- Presenečenja te volitve vsaj pri vrhu niso prinesle sprememb. **SDS** je, kot so napovedovale tudi vse javnomnenjske raziskave, nesporna zmagovalka, saj ji je pripadel vsak četrti glas. V novem, osmem sklicu državnega zbora bo imela 25 poslancev. Drugouvrščena, **Stranka Marjana Šarca**, jih bo imela 13. **SD** in **SMC** sta dosegli skoraj identični rezultat, zastopalo ju bo po deset poslancev, **Levica** jih bo imela devet. Sledijo **Nova Slovenija** s sedmimi poslanci, **Stranka Alenke Bratušek** in **DeSUS** s po petimi ter **Slovenska nacionalna stranka** s štirimi. (vir: 24ur.com, 4. 6. 2018)

Prepoznavanje imenskih entitet

- Uporabnost
  - Indeksiranje in povezovanje imenskih entitet (npr. povezave spletnih strani).
  - Določanje na koga ali kaj se nanaša analiza sentimenta.
  - Relacije pri ekstrakciji informacij so povezane z imeni entitet.
  - Sistemi vprašanj in odgovorov, pogosto uporabljajo poimenovanje entitet.

Ocenjevanje prepoznavanja imenskih entitet

- Uporaba kontingenčne tabele (tp,tn,fp,fn)
- Preciznost (angl. *Precision*)
- Priklic (angl. *Recall*)
- Mera F1.

Primer prepoznavanja imenskih entitet

- Miha (oseba)
- Maribor (mesto)
- Merkator (podjetje)
- Nova **Ljubljanska banka** (podjetje)
  Problem ugotavljanja mej, ki določata imensko entiteto.
  Rezultate je delno pravilen.
  Ni nujno, da gre za imensko entiteto.

Metode za prepoznavanje imenskih entitet

- Ročno zapisani regularni izrazi.
  - Uporabno v primeru lepo strukturiranih spetnih straneh.
  - Ponavadi uporabno za nekatere omejene splošne entitete v nestrukturiranih besedilih (npr. datumi in telefonske številke).
  - Pomagamo si lahko s pomočjo:
    - oblikoslovnega označevanja besedil (angl. part-of-speech tagging),
    - sintaksičnega razpornavanja (identifikacija fraz) in
    - semantične klasifikacije besed (npr. s pomočjo orodja WordNet).

Metode za prepoznavanje imenskih entitet

- Uporaba klasifikatorjev.
  - Generativne in diskriminatorne metode.
  - Klasifikacija besed v dva razreda: “za ekstrakcijo” in “ni za ekstrakcijo”.
  - V določenih preprostih domenah dosegajo zavidljive rezultate.
  - Primer: ugotavljanje spremembe elektronskega naslova.
- Sekvenčni modeli.
  - Učenje:
    - Učni dokumenti,
    - vsak leksikalni simbol ima oznako “imenska entiteta” oz. “ostalo”,
    - načrtovanje značilk in
    - učenje sekvenčnih klasifikatorjev.
  - Testiranje:
    - Testna množica,
    - klasifikacija in
    - ocenjevanje kvalitete klasifikacije.

Sekvenčni modeli

	IO kodiranje	IOB kodiranje
Miha	OSEBA	B-OSEBA
je	DRUGO	DRUGO
Marku	OSEBA	B-OSEBA
pokazal	DRUGO	DRUGO
nov	DRUGO	DRUGO
program	DRUGO	DRUGO
Janeza	OSEBA	**B-OSEBA**
Novaka	OSEBA	**I-OSEBA**
	c+1 oznak	2c+1 oznak
Časovna zahtevnost	manjša	večja
Uporabnejše	DA	NE

B - Začetek imenske entitete

I - Nadaljevanje imenske entitete

Značilke za sekvenčno označevanje

- Besede
  - Trenutna beseda (naučen slovar)
  - Prejšnja/naslednja beseda (kontekst)
- Drugi načini klasifikacije
  - Oblikoslovno označevanje
- Kontekst oznak
  - Prejšnja in naslednja oznaka

Viri

- Dan Jurafsky, Chris Manning, Natural Language Processing `http://web.stanford.edu/~jurafsky/NLPCourseraSlides.html`
- Slavnko Žitnik, Iterativno pridobivanje semantičnih podatkov iz nestrukturiranih besednih virov, doktorska disertacija, 2014.

Globoko učenje - uvod

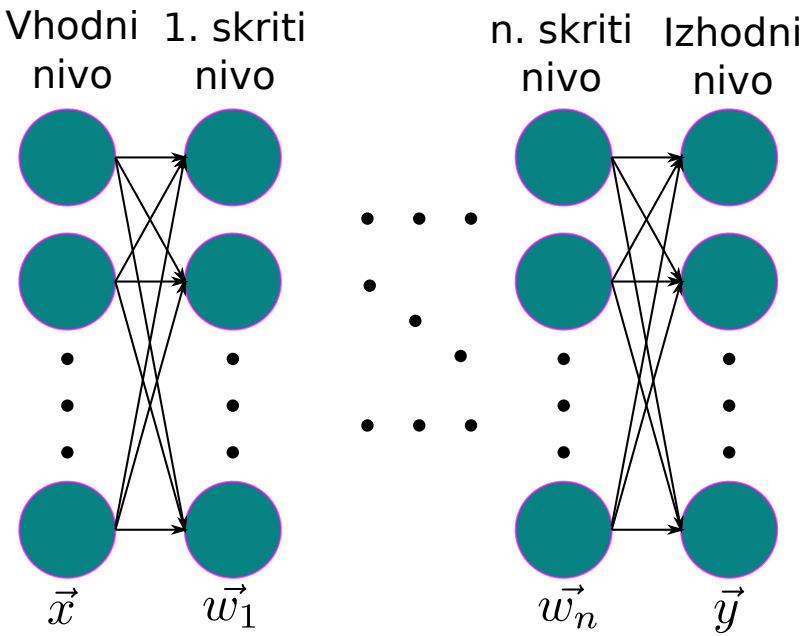
(angl. Deep Learning)

Kaj je globoko učenje

- Je področje strojnega učenja
- Večina metod strojnega učenja deluje dobro.
  - Razlog temu je človeško zasnovana predstavitev in določanje vhodnih značilk.
  - Globoko učenje ima večplastno predstavitev in od tod tudi prihaja njeno ime.
  - Pri izboljševanju metod se ljudje dosti naučijo o določenem problemu in hkrati pomagajo programom, da bolje opravljajo svojo nalogo.
- Na strojno učenje lahko gledamo kot na optimizacijo uteži, da dosežemo najboljšo možno predikcijo.

Kaj je globoko učenje

- Samodejno se poskuša ugotoviti dobre značilke oz. predstavitve (angl. feature learning ali representation learning)
- Algoritmi globokega učenja poskušajo "naučiti" večplastne predstavitve, da dobijo primerne izhode.
- Učenje temelji na vhodnih vzorcih $\vec{\mathbf{x}}$. To so lahko besede, značilke, zvok itd.



Zakaj uporabljati globoko učenje

- Ročno določene značilke so ponavadi preveč specifične, nepopolne in za njihov razvoj in validacijo potrebujemo ogromno časa.
- Določanje značilk s pomočjo strojnega učenja je možno prilagoditi določeni domeni. Ta postopek je relativno hiter.
- Globoko učenje omogoča zelo fleksibilno oz. skoraj univerzalno ogrodje za predstavitev sveta. To vključuje tako vizualni svet kot tudi jezikovne informacije.
- Globoko učenje lahko uporablja nenadzorovano (na osnovi neoznačenega besedila) kot tudi nadzorovano učenje (s pomočjo označenega besedila).

Zakaj uporabljati globoko učenje

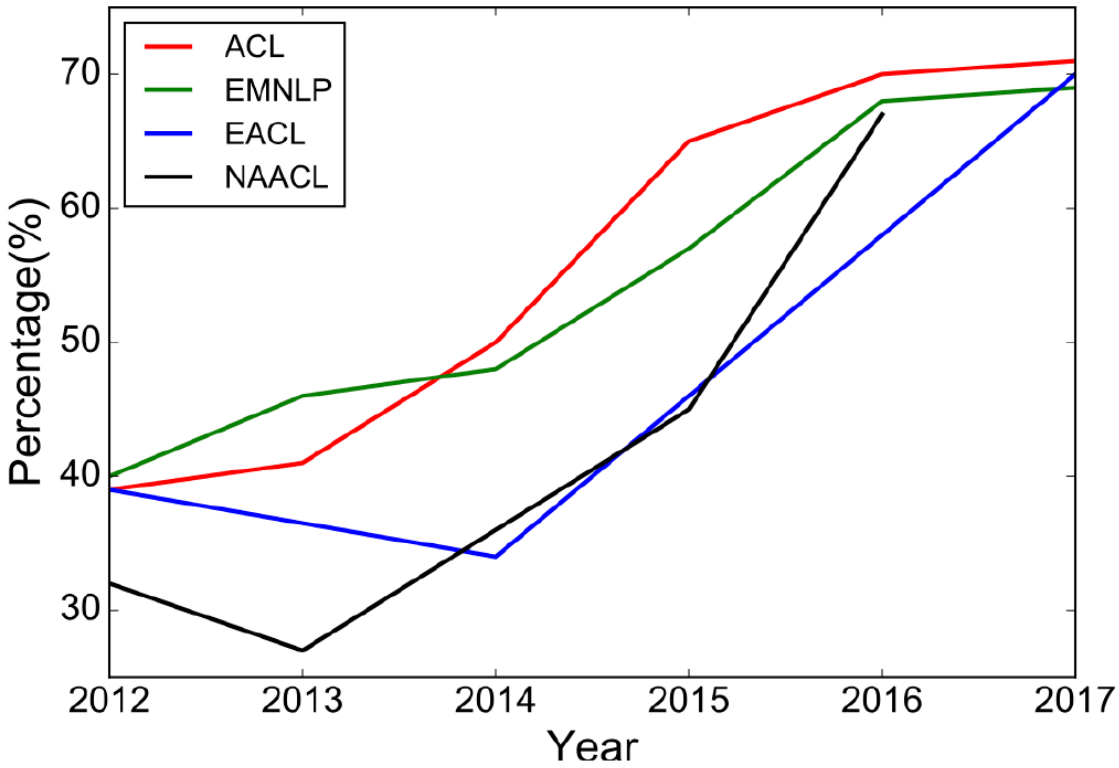
- Približno leta 2010 je globoko učenje začelo premagovati ostale metode strojnega učenja.
- S pomočjo globokega učenja je računalnik premagal svetovnega prvaka v igri go leta 2016.
- Kaj je pripomoglo k razvoju globokega učenja?
  - Velike količine podatkov.
  - Hitrejši in večjedrni procesorji. To vključuje tako centralno procesno enoto kot tudi grafično procesno enoto.
  - Novi modeli, algoritmi in ideje
    - Boljše in bolj fleksibilno učenje vmesnih plasti.
    - Učinkovit združen sistem učenja.
    - Učinkovite metode učenja, ki omogočajo boljši prenos informacij med konteksti kakor tudi med domenami.
  - Izboljšane zmogljivosti pri govoru, vidu in tudi pri jezikovnih tehnologijah.

Globoko učenje in govor

- Prvi preboj globokega učenja se je zgodil pri razpoznavanju govora s pomočjo velikih učnih množic.
  G. E. Dahl, D. Yu, L. Deng in A. Acero, Čontext-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition,"v IEEE Transactions on Audio, Speech, and Language Processing, letn. 20, št. 1, str. 30-42, 2012.
- Na področju računalniškega vida so prvi preboj naredili v naslednjem članku:
  Alex Krizhevsky, Ilya Sutskever in Geoffrey Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". Neural Information Processing Systems, 25, 2012, doi=10.1145/3065386.

Globoko učenje in jezikovne tehnologije

Število objav, ki obravnavajo globoko učenje in jezikovne tehnologije.



Vir: Tom Young, Devamanyu Hazarika, Soujanya Poria in Erik Cambria,

Recent Trends in Deep Learning Based Natural Language Processing, arXiv: 1708.02709

Globoko učenje in jezikovne tehnologije

- Združuje ideje in cilje jezikovnih tehnologij z globokim učenjem.
- Cilj je rešiti probleme jezikovnih tehnologij.
- Globoko učenje je aplicirano na različne
  - **nivoje** jezikovnih tehnologij: govor, besede, sintaksa, semantika itd.
  - **orodja** jezikovnih tehnologij: označevanje besedila, razpoznavanje entitet itd.
  - **aplikacije** jezikovnih tehnologij: strojno prevajanje, analiza sentimenta, sistemi vprašanj in odgovorov.

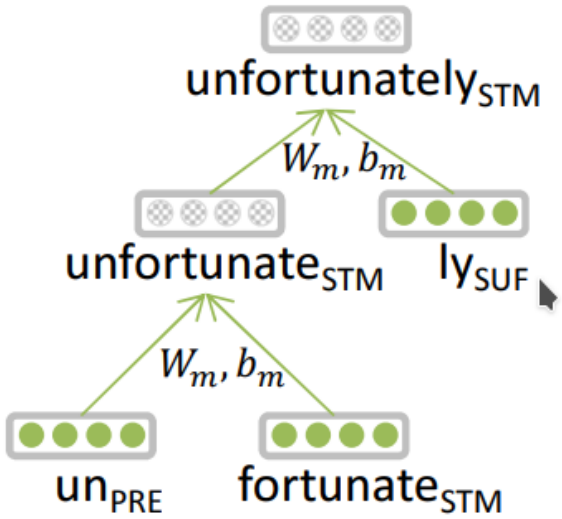
Predstavitev pomena besede - vektor

- V jezikovnih tehnologija se vse začne z besedami.
- Za predstavitev pomena besed uporabimo vektorje.
- Vektorji so velikih dimenzij (minimalno 25 dimenzij).
- Besedo postavimo v n-dimenzionalni prostor.
- Besede s podobnim pomenom so združene v grozde.
- Besede, ki se nahajajo v okolici besede **žaba**:
  Rosnica, Sekulja, Zelena žaba

Predstavitev nivojev jezikovnih tehnologij

- Morfologija
- Tradicionalno: besede so sestavljene iz morfemov oz. iz manjših besednih enot s samostojnim semantičnim pomenom.
- Globoko učenje
  - Vsak morfem je vektor.

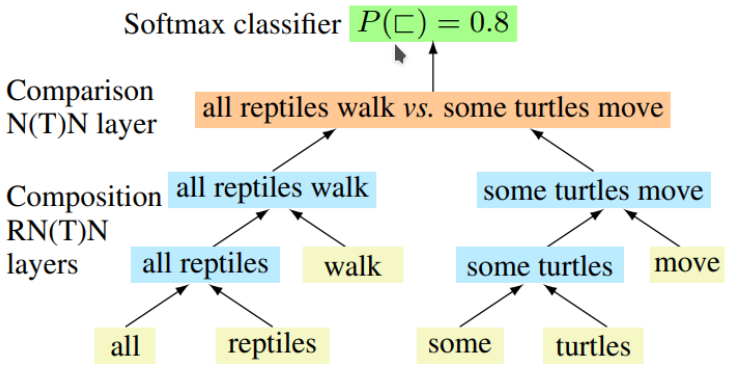
    Thang Luong, Richard Socher in Christopher Manning, Better Word Representations with Recursive Neural Networks for Morphology, Proceedings of the Seventeenth Conference on Computational Natural Language Learning, str. 104-113, 2013
- Nevronske mreže lahko natančno določijo struktura stavkov, ki vključuje tudi razlago.



Thang Luong in ostali, 2013

Predstavitev nivojev jezikovnih tehnologij

- Sintaksa
- Tradicionalen: Lambda račun
  - Natančno načrtovane funkcije.
  - Vhod v funkcijo je neka druga funkcija.
  - Ni notacij o podobnosti in nejasnosti jezika.
- Globoko učenje:
  - Vsaka beseda, vsaka fraza in vsak logični izraz je vektor.
  - Nevronska mreža združuje dva vektorja v enega.



Samuel R. Bowman, Christopher Potts in Christopher D. Manning, Recursive Neural Networks Can Learn Logical Semantics, Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, str. 12-21, 2015

Aplikacije globokega učenja

- Analiza sentimenta
  Lahko se uporabi enak model globokega učenja kot je bil uporabljen za morfologijo, sintakso in logično semantiko
- Sistem vprašanj in odgovorov
  Lahko se uporabi struktura globokega učenja in dejstva so lahko shranjena v vektorjih
- Ustvarjanje odgovorov
  Je aplikacija zmogljivih in splošnih jezikovnih modelov, ki temeljijo na rekurentnih nevronskih mrežah.
- Strojno prevajanje
  Izvorna poved se preslika v vektor, nato se ustvari izhodna poved.

  [Sutskever in ostali 2014, Bahdanau in ostali, 2014, Luong in Manning, 2016]

  Nekateri jeziki v prevajalniku Google.
- **Zaključek:** Vektorji na vseh nivojih predstavitev!

Vir

Christopher Manning in Richard Socher, Natural Language Processing with Deep Learning, CS224N/Ling284

Vektorji besed

(angl. Word Vectors)

Kako predstavimo pomen besed?

- Pomen (SSKJ):
  - Kar beseda vsebuje glede na označevani pojem, predmet.
  - Poudarja bistvene, tipične lastnosti česa, kot jih določa prilastek.
  - Pozitivne lastnosti, značilnosti česa.
  - Izraža nepotrebnost česa.
- Splošen način lingvističnega razmišljanja o pomenu:
  - Označevalec $<=>$ označeno (ideja o zadevi) $=$ denotacija

Kako dobimo uporaben pomen v računalniku?

- Uporaba taksonomij kot je npr. WordNet.
  - Samostalnik
    - mesojedec je hipernim psa
    - pes je hiponim mesojedca
    - volk je kohiponim psa in pes je kohiponim volka
    - stavba je holonim okna
    - okno je meronim stavbe
  - Glagol
    - potovati je hipernim glagola gibati se
    - šepetati je troponim glagola govoriti
    - glagol spati vsebuje glagol smrčati; prvi je pogoj za drugega
    - kohiponim: glagola šepetati in kričati, ki imata skupen hipernim – glagol govoriti
  - Pridevnik
    - hišni prag – pridevnik hišni, ki izvira iz samostalnika hiša
    - deležniki: pojoča deklica

Diskretna predstavitev besed

- Predstavlja dober vir.
- Ima probleme s podrobnostmi. Npr. **strokovnjak** je na seznamu sinonimov besede **dobro**. To drži le v določenih kontekstih.
- Manjkajo novi pomeni besed (nemogoče vzdrževati).
- Subjektivno.
- Potreben je človeški trud za izdelavo in vzdrževanje.
- Ni možno določiti natančne podobnosti.

Diskretna predstavitev besed

- Velika večina metod tradicionalnih jezikovnih tehnologij obravnava besede kot simbole: soba, avto, tek.
- V vektorskem prostoru bi jih tako predstavili z vektorji, katerih ene komponenta ima vrednost 1 in vse ostale komponente vrednost 0. To predstavitev imenujemo "**one hot**" kodiranje.
  - soba $= \{0, 0, 0, 0, 0, 1, 0, 0, 0\}$
  - avto $= \{0, 0, 1, 0, 0, 0, 0, 0, 0\}$
  - tek $= \{0, 0, 0, 1, 0, 0, 0, 0, 0\}$
- Dimenzija vektorjev bi morali biti enaka velikosti slovarja.
- Dva vektorja sta med seboj pravokotna. To pomeni, da je težko določiti podobnost med dvema vektorjema oz. besedama.
  - Lahko bi uporabili WordNet ampak na tak način imamo problem nepopolnosti.
  - **Poskusimo zakodirati podobnost v sam vektor.**

Predstavitev besed s pmočjo konteksta

- Distribucijska semantika: **Pomen besede je podan z besedami, ki se pogosto nahajajo v njihovi bližini.**
- Ena od najuspešnejših idej sodobnih metod jezikovnih tehnologij.
- Ko se beseda pojavi v besedilu, njen konteks predstavlja množica besed v njeni okolici (znotraj fiksne velikosti okna).
- Za izgradnjo predstavitve besede *b* uporabimo več kontekstov.
- Konteksti za predstavitev besede **kolo**:
  - Popoldne so zaplesali **kolo**.
  - 36 prvenstveno **kolo** je bilo ključno za Maribor.
  - Zamenjal je **kolo** avtomobila.
  - Sedel je na **kolo** in se odpeljal.
  - Kupil je novo motorno **kolo**.

Vektorji besed

- Angleški izrazi: word vectors, word embeddings, word representations.
- Zgradili bomo vektor za vsako besedo.
- Vektor bo omogočal izbiro podobnih vektorjev besed, ki se pojavljajo v podobnih kontekstih.
- Besede so porazdeljene po predstavitvi (angl. distributed representation).

Ogrodje za učenje vektorjev besed (word2vec)

- Enostaven in hiter model.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, in Jeff Dean, "Distributed Representations of Words and Phrases and their Compositionality", v Advances in Neural Information Processing Systems 26, str. 3111–3119, 2013.
- Veliki učni korpus.
- Vsaka beseda v slovarju fiksne dolžine je predstavljena s pomočjo vektorja.
- Sprehodimo se skozi besedilo, kjer je osrednja beseda c in besede konteksta so o.
- Uporabi se podobnost vektorjev c in o, da se izračuna verjetnost o za podani c in obratno.
- Nadaljuje se s prilagajanjem vektorjev, tako da se maksimira izračunane verjetnosti.

Ogrodje za učenje vektorjev besed (word2vec)

- Primer okna (velikosti 2) in izračun verjetnosti:
- Centralna besede b_t

Kako	se	**novi**	koronavirus	prenaša
$P(b_{t-2}\|b_t)$	$P(b_{t-1}\|b_t)$	b_t	$P(b_{t+1}\|b_t)$	$P(b_{t+2}\|b_t)$

se	novi	**koronavirus**	prenaša	med
$P(b_{t-2}\|b_t)$	$P(b_{t-1}\|b_t)$	b_t	$P(b_{t+1}\|b_t)$	$P(b_{t+2}\|b_t)$

Ogrodje za učenje vektorjev besed (word2vec)

- za vsak položaj $t = 1, ..., T$ napovemo okoliške besede s pomočjo okna določene velikosti m in podane centralne besede w_j.

$$Verjetnost = L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(b_{t+j} | b_t; \theta)$$

- θ - spremeljivke, ki jih je potrebno optimizirati.
- Ocenitvena funkcija (angl. objective, cost, loss function) $J(\theta)$:

$$J(\theta) = -\frac{1}{T} \log(L(\theta)) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log \ P(b_{t+j} | b_t; \theta)$$

- Minimizacija ocenitvene funkcije <=> maksimizacija natančnosti predikcije.

Ogrodje za učenje vektorjev besed (word2vec)

- Želimi minimizirati naslednjo ocenitveno funkcijo:

$$J(\theta) = -\frac{1}{T}\log(L(\theta)) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\leq j\leq m \\ j\neq 0}} \log\ P(b_{t+j}|b_t;\theta)$$

- Kako izračunati $P(b_{t+j}|b_t;\theta)$?
- Uporabili bomo dva vektorja za vsako besedo.
  - v_b ko je b centralna beseda.
  - u_b ko je b beseda iz konteksta.
- Za centralno besedo c in besedo o iz konteksta dobimo:

$$P(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{b\in s}\exp\left(u_b^T v_c\right)}$$

Ogrodje za učenje vektorjev besed (word2vec)

- Primer okna in računanja verjetnosti $P(b_{t+j}|b_t)$
- koronavirus = korona...
- $P(u_{se}|v_{korona...}) = P(se|korona...; u_{se}, v_{korona}, ..., \theta)$

se novi **korona...** prenaša med

$P(u_{se}|v_{korona...})P(u_{novi}|v_{korona...})$ b_t $P(u_{\text{prenaša}}|v_{korona...})P(u_{med}|v_{korona...})$

Ogrodje za učenje vektorjev besed (word2vec)

$$P(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{b \in s} \exp\left(u_b^T v_c\right)}$$

- Skalarni produkt dveh vektorjev določa podobnost dveh besed: $u_o^T v_c = \sum_{i=1}^{n} u_{o,i} \cdot v_{c,i}$
- Večja vrednost skalarnega produkta pomeni večjo verjetnost.
- Normalizacija skozi celoten slovar: $\sum_{b \in s} exp(u_b^T v_c)$
- To je primer *softmax* funkcije $\mathbb{R}^n \rightarrow (0,1)^n$

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} = p_i$$

- S pomočjo *softmax* funkcije preslikamo vrednosti x_i v verjetnostno porazdelitev p_i.
  - *max* - povečuje verjetnost za večje x_i.
  - *soft* - nekaj verjetnosti dodeli tudi malim x_i
  - Pogosto uporabljeno pri strojnem učenju.

Učenje modela s pomočjo optimizacije

- Izračun gradienta za vse vektorje.
- θ predstavlja vse parametre v modelu.
- V našem primeru imamo d-dimenzionalne vektorje in V besed.

$$\theta = \begin{bmatrix} v_{kolo} \\ v_{avto} \\ \dots \\ v_{sonce} \\ u_{kolo} \\ u_{avto} \\ \dots \\ u_{sonce} \end{bmatrix} \in \mathbb{R}^{2dV}$$

- Vsak beseda ima dva vektorja.
- Te parametre optimiziramo tako, da se premikamo v smeri gradienta oz. uporabimo algoritem Gradientni spust.

Gradient

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\leq j\leq m\\ j\neq 0}} \log\ P(b_{t+j}|b_t;\theta)$$

$$\frac{\partial}{\partial v_c}\log\frac{\exp\left(u_o^T v_c\right)}{\sum_{b\in s}\exp\left(u_b^T v_c\right)} =$$

$$\frac{\partial}{\partial v_c}\log\ \exp\left(u_o^T v_c\right) - \frac{\partial}{\partial v_c}\log\sum_{b\in s}\exp\left(u_b^T v_c\right)$$

Gradient

$$\frac{\partial}{\partial v_c} u_o^T v_c = u_o; \frac{\partial}{\partial (v_c)_1} u_o^T v_c = u_{o_1} v_{c_1} + u_{o_2} v_{c_2} + \ldots + u_{o_d} v_{c_d} = u_{o_1}$$

$$\frac{\partial}{\partial v_c} \log \sum_{b \in s} \exp\left(u_b^T v_c\right) = \frac{1}{\sum_{b \in s} \exp\left(u_b^T v_c\right)} \cdot \frac{\partial}{\partial v_c} \sum_{x \in s} \exp\left(u_x^T v_c\right) =$$

$$\frac{1}{\sum_{b \in s} \exp\left(u_b^T v_c\right)} \cdot \sum_{x \in s} \frac{\partial}{\partial v_c} \exp\left(u_x^T v_c\right) =$$

$$\frac{1}{\sum_{b \in s} \exp\left(u_b^T v_c\right)} \cdot \sum_{x \in s} \left(\exp\left(u_x^T v_c\right) \cdot \frac{\partial}{\partial v_c} u_x^T v_c\right) =$$

$$\frac{1}{\sum_{b \in s} \exp\left(u_b^T v_c\right)} \cdot \sum_{x \in s} \left(\exp\left(u_x^T v_c\right) \cdot u_x\right)$$

Pravilo verige: $y = f(u); \quad u = g(x); \quad y = f(g(x)); \rightarrow \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}u} \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}f(u)}{\mathrm{d}u} \frac{\mathrm{d}g(x)}{\mathrm{d}x}$

$\log'(x) = \frac{1}{x}; \quad \exp'(x) = \exp(x); f(x) = x \rightarrow f(x)' = 1$

Gradient

$$\frac{\partial}{\partial v_c} \log(P(o|c)) = u_o - \frac{\sum_{x \in s} \left(\exp\left(u_x^T v_c \right) \cdot u_x \right)}{\sum_{b \in s} \exp\left(u_b^T v_c \right)} =$$

$$u_o - \sum_{x \in s} \left(\frac{\exp\left(u_x^T v_c \right)}{\sum_{b \in s} \exp\left(u_b^T v_c \right)} \cdot u_x \right) =$$

$$u_o - \sum_{x \in s} \left(p(b|c) \cdot u_x \right)$$

- $\frac{\partial}{\partial v_c} \log(P(o|c))$ - smer v večdimenzionalnem prostoru
- u_o - opazovana predstavitev
- $\sum_{x \in s} p(b|c) \cdot u_x$ - model oz. pričakovana predstavitev

Gradientni spust

- Matrična notacija: $\theta^{new} = \theta^{old} - \alpha \nabla_\theta J(\theta)$
- Notacija za določen parameter: $\theta_j^{new} = \theta_j^{old} - \alpha \frac{\partial}{\partial \theta_j^{old}} J(\theta)$
- α - velikost koraka oz. stopnja učenja.
- Problem: $J(\theta)$ je funkcija vseh oken v korpusu in je časovno zelo zahtevna.
- Rešitev: Uporaba algoritma Stohastični gradientni spust. Iterativno vzorčimo okna in jih posodabljamo po vsaki epohi.
- Zaradi strojne opreme, ki se lahko izvaja paralelno, vzorčimo okna velikosti 32, 64 itd.

Stohastnični gradientni spust

- Iterativno računamo gradiente za vsako okno.
- V vsakem oknu imamo največ 2m+1 besed.
- $\nabla_\theta J_t(\theta)$ je zelo redek.

$$\nabla_\theta J_t(\theta) = \begin{bmatrix} 0 \\ \dots \\ \nabla_{b_{kolo}} \\ \dots \\ 0 \\ \nabla_{b_{avto}} \\ \dots \\ \nabla_{b_{sonce}} \end{bmatrix} \in \mathbb{R}^{2dV}$$

Stohastnični gradientni spust

- Posodabljanje vektorjev besed, ki se dejansko pojavljajo.
- Rešitev: uporaba operatorjev za posodabljanje redkih matrik ali uporaba preslikave vektorjev besed (angl. hash).
- V primeru ogromne količine besed in distribuiranega sistema, si ne moremo privoščiti razpošiljanje ogromnih posodobitev!

Ogrodje za učenje vektorjev besed (word2vec)

- Dva tipa modelov
  - Skip-grams (SG)
    Napove besede konteksta (neodvisno od položaja) na osnovi centralne besede. **Model ki smo ga predstavili.**
  - Continuous Bag of Words (CBOW)
    Napove centralno besedo glede na (vrečo besed) besede konteksta.
- Učinkovitost učenja.
  - Negativno vzorčennje.
  - Naïve softmax - je enostavna in časovno zahtevna metodo učenja.

Skip-gram model z negativnim vzorčenjem

- Normalizacija je časovno zahtevna.
  $P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{b \in s} exp(u_b^T v_c)}$
- Uporaba binarne logistične regresije za pravi par centralne besede in besede v njenem kontekstnem oknu v primerjavi z več šumnimi pari kjer centralno besedo povežemo z naključno besedo.
- Članek: "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al. 2013)
  - Ocenitvena funkcija: $J(\theta) = \frac{1}{T} \sum_{t=1}^{T} J_t(\theta)$

$$J(\theta) = log\ \sigma(b_o^T\ b_c) + \sum_{i=1}^{k} E_{j \sim P(b)}[log\ \sigma(-b_j^T\ b_c)] \qquad (1)$$

  - Sigmoidna funkcija $\sigma(x) = \frac{1}{1+e^{-x}}$

Skip-gram model z negativnim vzorčenjem

- Notacija v našem kontekstu:
  $J_{negativno\_vzorčenje}(\mathbf{o}, \mathbf{b_c}, \mathbf{U}) =$
  $-log(\sigma(b_o^T\ b_c)) - \sum_{k=1}^{K} log(\sigma(-b_k^T\ b_c))$
- Vzamemo k negativnih vzorcev (uporaba verjetnosti besed)
- Maksimiramo verjetnost za realne besede konteksta.
- Minimiziramo verjetnosti za naključno izbrane besede.
- Verjetnost vzorčenja besed: $P(b) = \frac{U(b)^{\frac{3}{4}}}{Z}$
  - $U(b)$ - porazdelitev unigramov
  - Z - normalizacija
  - $\frac{3}{4}$ - manj frekventne besede bodo pogosteje izbrane.

Matrika sopojavljanja - *Co-occurrence matrix*

Nenavadno je iti skozi celoten korpus večkrat. Zakaj preprosto ne uporabimo statistike o tem, katere besede se pojavljajo ena blizu druge?

- Okno velikosti 1 (ponavadi se uporablja velikost od 5 do 10)
- Simetrična

```
Globoko učenje me zanima.  Jezikovne tehnologije so zanimive.  Globoko
učenje in jezikvone tehnologije so uporabne.
```

števec	globoko	učenje	me	zanima	jezikovne	tehnologije	so	zanimive	in	uporabne
globoko	0	2	0	0	0	0	0	0	0	0
učenje	2	0	1	0	0	0	0	0	1	0
me	0	1	0	1	0	0	0	0	0	0
zanima	0	0	1	0	0	0	0	0	0	0
jezikovne	0	0	0	0	0	2	0	0	1	0
tehnologije	0	0	0	0	2	0	2	0	0	0
so	0	0	0	0	0	2	0	1	0	1
zanimive	0	0	0	0	0	0	1	0	0	0
in	0	1	0	0	1	0	0	0	0	0
uporabne	0	0	0	0	0	0	1	0	0	0

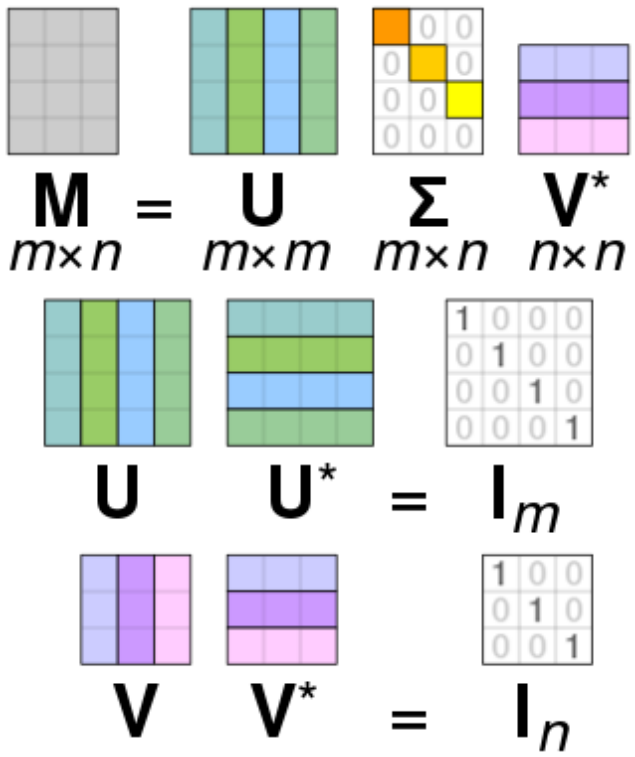
Matrika sopojavljanja - gradnja

- Uporaba manjšega okna
  - Podobno word2vec
  - Uporablja okno v okolici besede (lokalnost)
  - Zajema sintaksične in semantične informacije
- Okno je velikosti odstavka ali celotnega dokumenta
  - Podaja splošno tematiko
  - Analiza sentimenta

Matrika sopojavljanja - značilnosti

- Enostavno določanje vektorjev
  - Z večanjem slovrja se večja dimenzija vektorjev
  - Velika dimenzija zahteva dosti pomnilnika (čeprav so redki).
  - Medeli klsifikacije morajo obravnavati redkost (manjša robustnost).
- Manj dimenzionalni vektorji
  - Uporabnejši
  - Shranimo pomembnejše informacije v vektroje manjših dimenzij - gosti vektorji (angl. dense vectors)
  - Ponavadi imajo od 25 do 1000 dimenzij (podobno kot word2vec)
  - Dekompozicija singularnih vrednosti (angl. singular value decomposition)

Dekompozicija singularnih vrednosti



Vir: wikimedia.org

Dekompozicija singularnih vrednosti

- Na preprostih števcih ne daje dobre rezultate.
- Skaliranje števcev lahko pomaga.
  - Problem: Besede, ki nimajo semantičnega pomena (npr. the, he, has) so zelo frekventne (velik vpliv sintakse).
  - Rešitev
    - Uporaba funkcije *log* nad števci
    - Uporaba manj frekventnih besed
    - Ignoriranje besed, ki nimajo semantičnega pomena
- V skaliranih vektorjih se pojavijo vzorci semantike.
  - Drive $\rightarrow$ Driver
  - Teach $\rightarrow$ Teacher

Neposredna predikcija in predikcija na osnovi štetja

- Na osnovi štetja (dekompozicijska metoda)
  - Hitro učenje.
  - Učinkovita uporaba statistik.
  - Uporabno za zajemanje podobnosti besed.
  - Nesorazmeren pomen dodan velikim številom.
- Neposredna predikcija (skip gram)
  - Skalirajo se z velikostjo korpusa.
  - Neučinkovita uporaba statistik.
  - Omogočajo izboljšave drugih nalog jezikovnih tehnologij.
  - Lahko zajamejo kompleksne vzorce med besedami in ne samo njihovo podobnost.

Kodiranje pomena v razlikah vektorjev

Razmerje verjetnosti sopojavljanja lahko kodira pomen komponent

	x = trdno	x = plinasto	x = voda	x = naključno
$P(x\|led)$	velika	mala	velika	male
$P(x\|para)$	mala	velika	velika	mala
$\frac{P(x\|led)}{P(x\|para)}$	velika	mala	∼1	∼1

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Kodiranje pomena v razlikah vektorjev

Razmerje verjetnosti sopojavljanja lahko kodira pomen komponent

	x = trdno	x = plinasto	x = voda	x = naključno
$P(x\|led)$	$1{,}9\text{x}10^{-4}$	$6{,}6\text{x}10^{-5}$	$3{,}0\text{x}10^{-3}$	$1{,}7\text{x}10^{-5}$
$P(x\|para)$	$2{,}2\text{x}10^{-5}$	$7{,}8\text{x}10^{-4}$	$2{,}2\text{x}10^{-3}$	$1{,}8\text{x}10^{-5}$
$\frac{P(x\|led)}{P(x\|para)}$	8,9	$8{,}5\text{x}10^{-2}$	1,36	0,96

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Kodiranje pomena v razlikah vektorjev

- Kako določimo razmerja verjetnosti sopojavljanja za linearne pomenske komponente v prostoru vektorjev besed?
- Bilinearni model: $w_i \cdot w_j = logP(i|j)$
- Razlika vektorjev: $w_x \cdot (w_a - w_b) = log\frac{P(x|a)}{P(x|b)}$

Združitev obeh pristopov

GloVe (Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.)

Model log-bilinear: $w_i \cdot w_j = logP(i|j)$

Razlike vektorjev: $w_x \cdot (w_a - w_b) = log\frac{P(x|a)}{P(x|b)}$

Funkcija izgube: $J = \sum_{i,j=1}^{V} f(X_{ij})(w_u^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2$

- Hitro učenje
- Možnost velikih korpisov
- Dobri rezultati tudi v primeru malih korpusov in malih dimenzij vektorjev

Kako dobri so vektorji besd?

- Notranje ovrednotenje
  - Ovrednotenje s pomočjo določene notranje naloge
  - Hiter izračun
  - Pomaga razumeti sistem
  - Ne vemo kako uporabni so vektorji, dokler jih ne preizkusimo na realni nalogi
- Zunanje ovrednotenje
  - Ovrednotenje na realni nalogi
  - Časovno zahtevno
  - Ne vemo ali je mogoče problem v sistemu vektorjev ali v katerem drugem podsistemu

Notranje ovrednotenje

- Kako dobro kosinusna razdalja po seštevanju vektorjev zajema semantična in sintaksična vpršanja.

  $d = \arg\max_i \frac{(x_b - x_a + x_c)^T x_i}{||x_b - x_a + x_c||}$

  `moški <=> ženska :: krelj <=> kraljica`

- Kaj pa če relacije niso linearne?

Klasifikacija razlik med vektorji besed

- Kompleksen model
- Uporabimo nevronsko mrežo
- “Učimo” parametre nevronske mreže in “položaje” vektorjev
- Uporabimo `embedding layer`

Viri

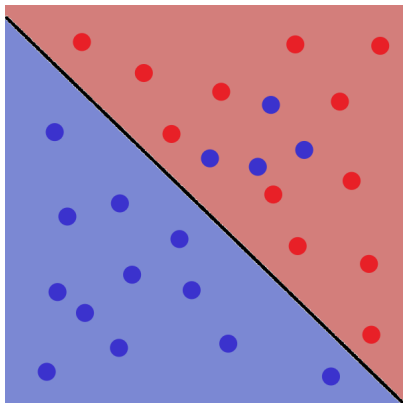
- Christopher Manning in Richard Socher, Natural Language Processing with Deep Learning, CS224N/Ling284
- Rohde in ostali, An Improved Model of Semantic Similarity Base on Lexical Co-Occurrence, 2005
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

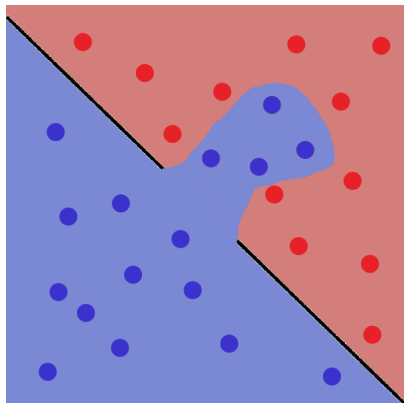
Nevronske mreže

(angl. Neural Networks)

Klasifikator *softmax* in nevronske mreže

- Klasifikatorji *softmax*: $P(y|x) = \frac{\exp(W_y.x)}{\sum_{c=1}^{C} \exp(W_c.x)}$
  - Naučeni parametri W
  - Klasifikator določi linearno odločitveno mejo
- Nevronske mreže
  - V fazi učenja se določajo W in predstavitev oz. porazdelitev besed.
  - Besede so predstavljene z *one-hot* vektorji, ki se preslikajo v vmesni sloj vektorskega prostora. V ta namen se uporabi *embedding* nivo.
  - Uporabimo globoke nevronske mreže, ki naše podatke oz. vektorje preoblikuje več krat. To omogoči ne-linearno klasifikacijo.





Klasifikator *softmax*

Vsebuje tri korake:

- Za vsak razred y izračunamo skalarni produkt: $W_y.x = \sum_{i=1}^{d} W_{yi}x_i = f_y$
- Uporabimo funkcijo *softmax*, da dobimo normalizirano verjetnost: $P(y|x) = \frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)}$
- Izberemo y z največjo verjetnostjo.

Za vsak učni primer (x,y) si želomo maksimirati verjetnost pravilnega razreda y oz. minimizirati negativno *log* verjetnost:

$$-\log P(y|x) = -\log\left(\frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)}\right)$$

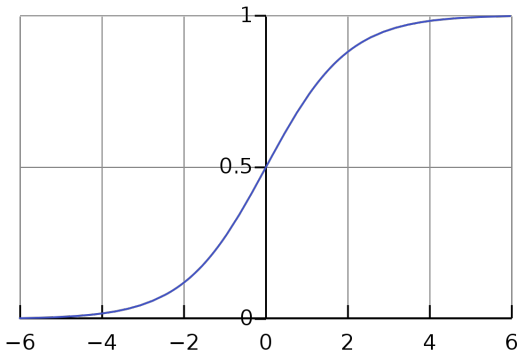
Nevron

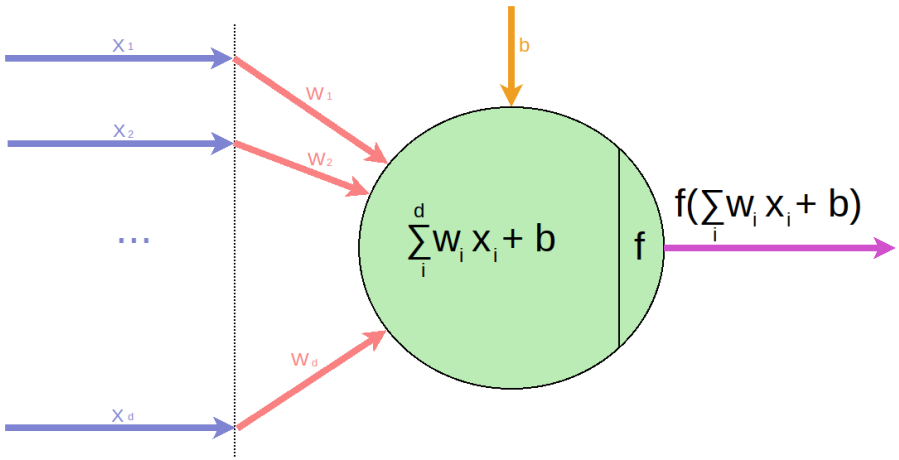
Delovanje binarne logistične regresije je podobno delovanju nevrona.

- f - nelinearna aktivacijska funkcija

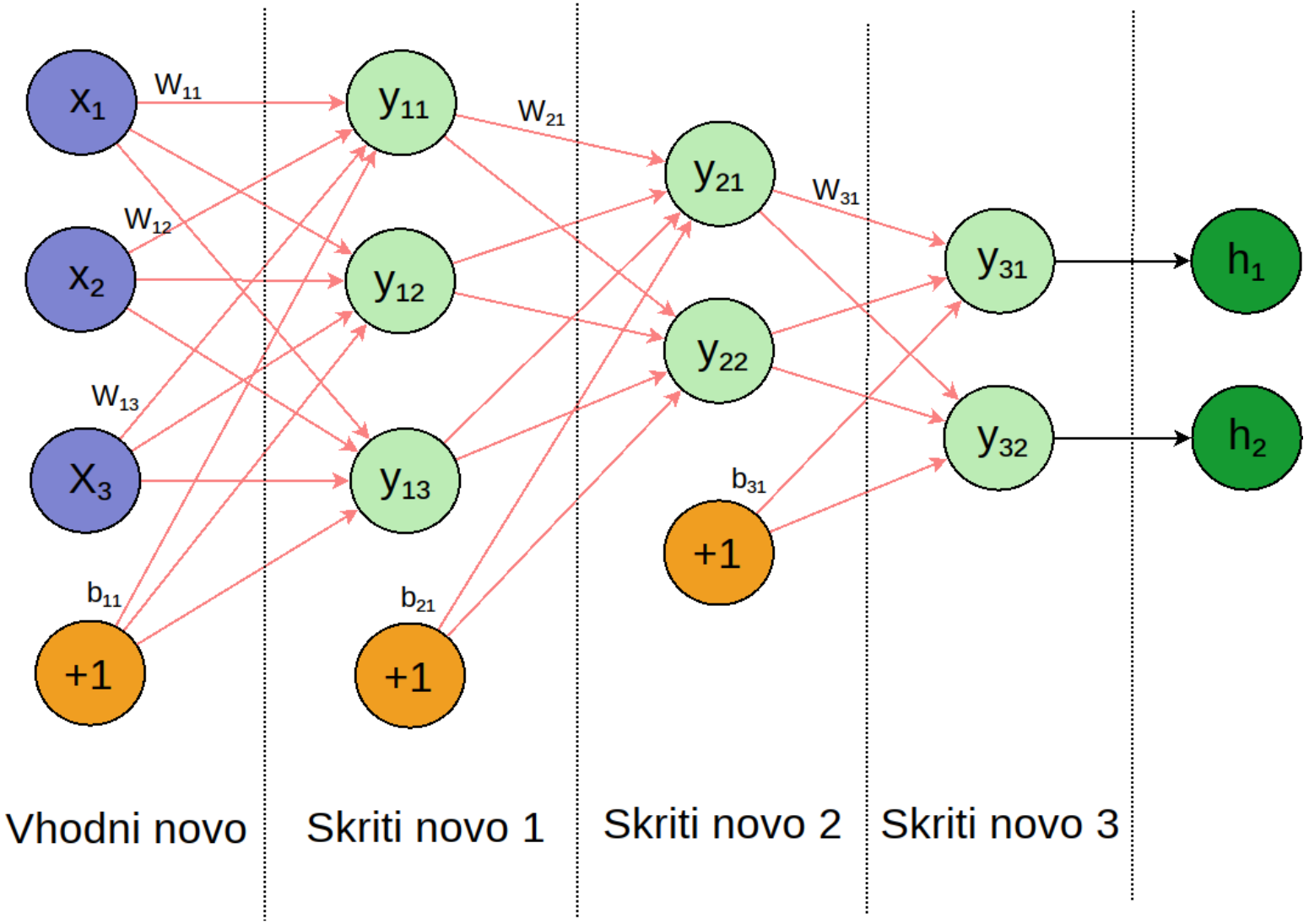
  Sigmoidna: $f(z) = \frac{1}{1+e^{-z}}$

- w_i - uteži
- b - pristranskost
- x_i - vhod





Nevronska mreža



Matrična notacija

- Izhodne vrednosti nevronov
  - $y_{11} = f(W_{11}x_1 + W_{12}x_2 + W_{12}x_3 + b_{11})$
  - $y_{12} = f(W_{21}x_1 + W_{22}x_2 + W_{22}x_3 + b_{12})$
  - itd.
- Matrična notacija
  - $\mathbf{z} = \mathbf{W}\mathbf{x} + b$
  - $\mathbf{y} = f(\mathbf{z})$
- Aktivacijska funkcija se aplicira na vsak element vektorja
  - $\mathbf{y} = f(\{z_1, z_2, z_3\}) = \{f(z_1), f(z_2), f(z_3)\}$

Gradient - ponovitev

- Funkcija z enim vhodom in enim izhodom $f(x) = x^3$
  - Gradient je enak odvodu $f'(x) = \frac{df}{dx} = 3x^2$
- Funkcija z n vhodi in enim izhodom $f(\mathbf{x}) = f(x_1, x_2, ..., x_n)$
  - Gradient je vektor parcialnih odvodov $\frac{\partial f}{\partial \mathbf{x}} = \{\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n}\}$
- Funkcija z n vhodi in m izhodi

  $\mathbf{f}(\mathbf{x}) = \{f_1(x_1, x_2, ..., x_n), f_2(x_1, x_2, ..., x_n), ..., f_m(x_1, x_2, ..., x_n)\}$
  - Gradient je Jakobijeva matrika, ki vsebuje $m \cdot n$ parcialnih odvodov

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \qquad \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)_{i,j} = \frac{\partial f_i}{\partial x_j}$$

Pravilo verige - ponovitev

- Sestavljena funkcija z eno spremenljivko (množenje odvodov)
  - $z = 3y; \quad y = x^2$
  - $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} = (3)(2x) = 6x$
- Funkcije z več spremenljivkami (množenje Jakobijevih matrik)
  - $\mathbf{h} = f(\mathbf{z}); \quad \mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$
  - $\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \frac{\partial \mathbf{h}}{\partial \mathbf{z}}\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \ldots$
  - Jakobijeva matrika za elementarno aktivacijsko funkcijo

$$\left(\frac{\partial \mathbf{h}}{\partial \mathbf{z}}\right)_{i,j} = \frac{\partial h_i}{\partial z_j} = \frac{\partial}{\partial z_j} f(z_i) = \begin{cases} f'(z_i) & \text{če } i == j \\ 0 & \textbf{drugače} \end{cases}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \begin{bmatrix} f'(z_1) & & 0 \\ & \ddots & \\ 0 & & f'(z_n) \end{bmatrix}$$

- Jakobijeve matrike

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right) = \mathbf{W}; \quad \frac{\partial}{\partial \mathbf{b}}\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right) = \mathbf{I}; \quad \frac{\partial}{\partial \mathbf{u}}\left(\mathbf{u}^T\mathbf{h}\right) = \mathbf{h}^T$$

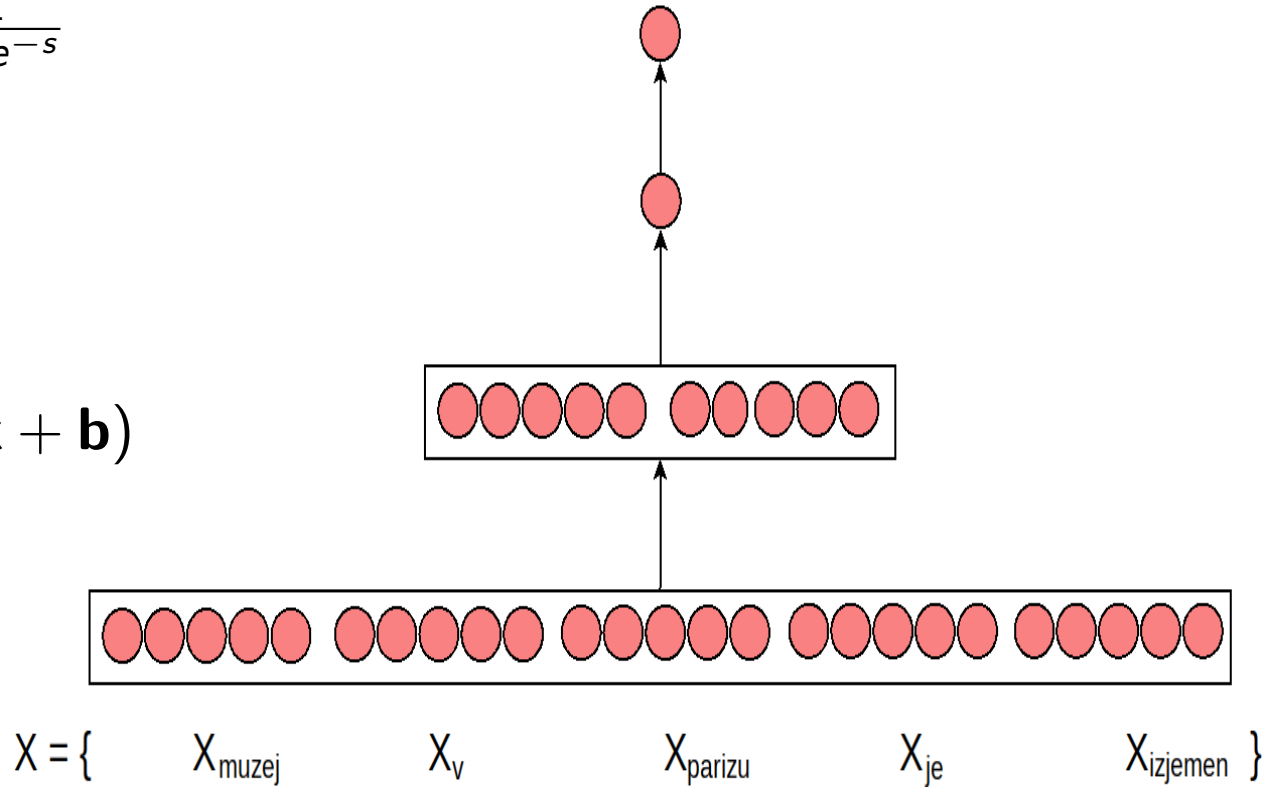
Nevronska mreža

$$\sigma(s) = \frac{1}{1+e^{-s}}$$

$$s = \mathbf{u}^{\mathbf{T}}\mathbf{h}$$

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b})$$

x - vhod



Gradient

Zanima nas gradient izgube (J_t), vendar bomo zaradi poenostavitve izračunali gradient rezultata s oz. $\frac{\partial s}{\partial \mathbf{b}}$.

$$\sigma(s) = \frac{1}{1+e^{-s}}; \quad s = \mathbf{u}^\mathbf{T}\mathbf{h}; \quad \mathbf{h} = f(z); \quad \mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

- Uporabimo pravilo verige: $\frac{\partial s}{\partial \mathbf{b}} = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{b}}$
- Uporabimo Jakobijeve matrike:

$$\frac{\partial s}{\partial \mathbf{h}} = \frac{\partial}{\partial \mathbf{h}} (\mathbf{u}^\mathbf{T}\mathbf{h}) = \mathbf{u}^\mathbf{T}$$
$$\frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \frac{\partial}{\partial \mathbf{z}} (f(z)) = \text{diag}(f'(\mathbf{z}))$$
$$\frac{\partial \mathbf{z}}{\partial \mathbf{b}} = \frac{\partial}{\partial \mathbf{b}} (\mathbf{W}\mathbf{x} + \mathbf{b}) = \mathbf{I}$$
$$\frac{\partial s}{\partial b} = \mathbf{u}^\mathbf{T} \text{ diag}(f'(\mathbf{z})) \mathbf{I} = \mathbf{u}^\mathbf{T} \odot f'(\mathbf{z})$$

Gradient

Izračunajmo $\frac{\partial \mathbf{s}}{\partial \mathbf{W}}$.

- Pravilo verige: $\frac{\partial s}{\partial \mathbf{w}} = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{w}}$
- Uporabimo že izračunane vrednosti:

  $\frac{\partial s}{\partial \mathbf{b}} = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{b}} = \delta \frac{\partial \mathbf{z}}{\partial \mathbf{b}}$

  $\frac{\partial s}{\partial \mathbf{W}} = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}} = \delta \frac{\partial \mathbf{z}}{\partial \mathbf{W}}$
- Signal napake: $\delta = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \mathbf{u}^{\mathbf{T}} \odot f'(\mathbf{z})$
- Gradient: $\frac{\partial s}{\partial \mathbf{W}} = \delta \frac{\partial \mathbf{z}}{\partial \mathbf{W}} = \delta \frac{\partial}{\partial \mathbf{W}} (\mathbf{W}\mathbf{x} + \mathbf{b}) = \delta^T \mathbf{x}^T$
- Transponirana vektorja: $[n \times m] = [n \times 1][1 \times m]$

Vir

Christopher Manning in Richard Socher, Natural Language Processing with Deep Learning, CS224N/Ling284

Primeri nalog

Drugi kolokvij

- Poglavja
  - Analiza sentimenta
  - Diskriminatorni klasifikatorji v jezikovnih tehnologijah
  - Strojno prevajanje
  - Ekstrakcija informacij in prepoznavanje imenskih entitet
  - Globoko učenje v jezikovnih tehnologijah
  - Vektorji besed

Primeri nalog

Glede na podane značilke iz kratkih komentarjev filmov, ki so tudi označeni z žanrom kateremu pripadajo, določite kateremu žanru pripdajo podane 6. značilke.

1. zabava, par, ljubezen: **komedija**
2. hitro, besno: **akcija**
3. par, leti, hitro, zabavno: **komedija**
4. besno, streljanje, streljanje: **akcija**
5. leti, hitro, streljaj: **akcija**
6. hitro, par, streljaj, leti: **?**

Za izračun najverjetnejšega razreda uporabite klasifikator Naïve Bayes in uporabite glajenje add-1.

Primeri nalog

$P(\text{značilke}|\text{razred}) = P(\text{razreda}) \prod_{x \in \text{značilke}} P(x)$

$P(\text{značilke}|\text{komedija}) = P(\text{komedija}) \cdot P(\text{hitro}|\text{komedija}) \cdot$
$P(\text{par}|\text{komedija}) \cdot P(\text{streljaj}|\text{komedija}) \cdot$
$P(\text{leti}|\text{komedija}) = \frac{2}{5} \cdot \frac{1+1}{7+9} \cdot \frac{2+1}{7+9} \cdot \frac{0+1}{7+9} \cdot \frac{1+1}{7+9} =$
$\frac{2}{5} \cdot \frac{2}{16} \cdot \frac{3}{16} \cdot \frac{1}{16} \cdot \frac{2}{16} = 7,32e-05$

$P(\text{značilke}|\text{akcija}) =$
$P(\text{akcija}) \cdot P(\text{hitro}|\text{akcija}) \cdot P(\text{par}|\text{akcija}) \cdot P(\text{streljaj}|\text{akcija}) \cdot$
$P(\text{leti}|\text{akcija}) = \frac{3}{5} \cdot \frac{2+1}{8+9} \cdot \frac{0+1}{8+9} \cdot \frac{1+1}{8+9} \cdot \frac{1+1}{8+9} =$
$\frac{3}{5} \cdot \frac{3}{17} \cdot \frac{1}{17} \cdot \frac{2}{17} \cdot \frac{2}{17} = 8,62e-05$

6. hitro, par, streljaj, leti: **akcija** $(8,62e-05 > 7,32e-05)$

Primeri nalog

S pomočjo binariziranega Multinominal Naïve Bayes-a in glajenja add-1 določite razred podane povedi na osnovi podatkov v tabeli.

dok.	"dober"	"slabo"	"odlični"	razred
d1	6	0	6	poz.
d2	0	2	4	poz.
d3	2	6	0	neg.
d4	2	10	4	neg.
d5	0	4	0	neg.

Poved: Dober, dober zaplet, odlični igralci, se pa slabo konča.

Primeri nalog

$$P(P|poz.) =$$
$$P(poz.) \cdot P(dober|poz.) \cdot P(odlični|poz.) \cdot P(slabo|poz.) =$$
$$\frac{2}{5} \cdot \frac{1+1}{18+3} \cdot \frac{1+1}{18+3} \cdot \frac{1+1}{18+3} = \frac{2}{5} \cdot \frac{2}{21} \cdot \frac{2}{21} \cdot \frac{2}{21} = 0,00034$$
$$P(P|neg.) =$$
$$P(neg.) \cdot P(dober|neg.) \cdot P(odlični|neg.) \cdot P(slabo|neg.) =$$
$$\frac{3}{5} \cdot \frac{1+1}{28+3} \cdot \frac{1+1}{28+3} \cdot \frac{1+1}{28+3} = \frac{3}{5} \cdot \frac{2}{31} \cdot \frac{2}{31} \cdot \frac{2}{31} = 0,00016$$

Poved: Dober, dober zaplet, odlični igralci, se pa slabo konča.
poz. $\mathbf{(}0,00034 > 0,00016\mathbf{)}$

Primeri nalog

Na osnovi podanih podatkov in modela IBM 1, izračunajte katera poravnava je verjetnejša.

$a_1 = \{2, 3, 4, 5\}$
$a_2 = \{1, 3, 4, 5\}$

e = the coronavirus has been defeated
s = koronavirus je bil premagan

$l = 5,\ m = 4$

Primeri nalog

$t(in|the) = 0,13 \quad t(in|coronavirus) = 0,12$
$t(in|has) = 0,31 \quad t(in|been) = 0,32 \quad t(in|defeated) = 0,12$
$t(koronavirus|the) = 0,11$
$t(koronavirus|coronavirus) = 0,9 \quad t(koronavirus|has) = 0,15$
$t(koronavirus|been) = 0,13 \quad t(koronavirus|defeated) = 0,14$
$t(je|the) = 0,21 \quad t(je|coronavirus) = 0,13$
$t(je|has) = 0,75 \quad t(je|been) = 0,3 \quad t(je|defeated) = 0,14$
$t(bil|the) = 0,2 \quad t(bil|coronavirus) = 0,13$
$t(bil|has) = 0,63 \quad t(bil|been) = 0,82 \quad t(bil|defeated) = 0,24$
$t(premagan|the) = 0,12$
$t(premagan|coronavirus) = 0,13 \quad t(premagan|has) = 0,14$
$t(premagan|been) = 0,11 \quad t(premagan|defeated) = 0,89$

Primeri nalog

e = the coronavirus has been defeated
s = koronavirus je bil premagan

$l = 5, \ m = 4$
$a_1 = \{2, 3, 4, 5\}; \quad a_2 = \{1, 3, 4, 5\}$

$t(koronavirus|coronavirus) = 0,9$
$t(koronavirus|the) = 0,11; \quad t(je|has) = 0,75;$
$t(bil|been) = 0,82; \quad t(premagan|defeated) = 0,89$

$P(s|a_1, e, m) = t(koronavirus|coronavirus) \cdot t(je|has) \cdot t(bil|been) \cdot$
$t(premagan|defeated) = 0.49$
$P(s|a_2, e, m) = t(koronavirus|the) \cdot t(je|has) \cdot t(bil|been) \cdot$
$t(premagan|defeated) = 0.060$

Verjetnejša je poravnava a_1.