Multilingual Word Embeddings

Daniel Zeman, Rudolf Rosa

April 20, 2023

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Words Are Sparse

- Brown clustering
  - Peter F. Brown, Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra, Jenifer C. Lai **(1992):** Class-based n-gram models of natural language. In: *Computational Linguistics* 18 (4)
- Short motivation (Jurafsky and Martin):
  - Never seen bigram *to Shanghai*, estimate its probability
  - Known bigrams *to London, to Beijing, to Denver*
  - Known word *Shanghai* but not in this context
  - Can we figure out that names of cities form one class of words?

Brown Clustering

- Start: each word its own class (cluster)
- Repeat: merge two clusters into one
  - Selection: minimize loss in **mutual information (MI)**
- Stop: if desired number of classes (task-dependent)

- MI: How does event A decrease entropy of event B?
- $MI(A, B) = H(B) - H(B|A) = H(A) - H(A|B)$
  - $H(B|A) = -\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} P(a_i, b_j) \log_2 P(b_j|a_i)$
- For an arbitrary bigram (w_{i-1}, w_i):
  - A: word type of w_{i-1} is **in cluster** a
  - B: word type of w_i is **in cluster** b

Example Clusters (from Brown et al., 1992)

- Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays
- June March July April January December October November September August
- people guys folks fellows CEOs chaps doubters commies unfortunates blokes
- down backwards ashore sideways southward northward overboard aloft downwards adrift
- water gas coal liquid acid sand carbon steam shale iron
- great big vast sudden mere sheer gigantic lifelong scant colossal
- man woman boy girl lawyer doctor guy farmer teacher citizen
- American Indian European Japanese German African Catholic Israeli Italian Arab
- pressure temperature permeability density porosity stress velocity viscosity gravity
- mother wife father son husband brother daughter sister boss uncle
- machine device controller processor CPU printer spindle subsystem compiler plotter
- John George James Bob Robert Paul William Jim David Mike
- anyone someone anybody somebody
- feet miles pounds degrees inches barrels tons acres meters bytes
- director chief professor commissioner commander treasurer founder superintendent dean

Example Clusters (Czech PDT dev data)

- než nebo 1993 produkce zhruba 1992 soudu 1994 1991 smlouvu
- a či S miliónů 9. Tento (Tato přičemž anebo
- to tu navíc člověk ta rozpočet vhodné rada nabídka taky
- nám mu mi ho jim svým stal jí existuje dá
- pak dnes stále proto nyní dokonce často tam přitom snad
- však totiž ovšem sice prý jich podařilo poslanec pochopitelně patrně
- již ještě už zde letos stát většinou dále situace firma
- pouze jen asi především právě zejména zcela vůbec přímo něco
- také tedy například zatím vždy opět zájem rovněž skutečně vlastně
- budou bude lze má může mají musí chce mohou nebude
- měl měla mělo mohl měly měli mohla mohly mohli mohlo
- jsme jsem jste bych dlouho divadla americký uvádí ať policisté
- by bychom Aby
- být mít právo moci muset hrát platit hledat pomoci dát
- není byla byl bylo nejsou platí nebyl nebylo podmínky nabízí

Neural Networks and Language Modeling

- Input: **index vectors** ("one-hot vectors"). Vocabulary size $|V|$ is the number of dimensions. The dimension corresponding to the represented word has value 1, all other dimensions have 0.
- **Embedding:** map the index vector into a vector with a lower, fixed number of dimensions (features, latent factors). We get an **embedding matrix:** rows correspond to word types, columns correspond to latent features (dimensions). The values in the cells get updated during training of the network; but we remember the index of the row for each word, so we can locate its updated vector.

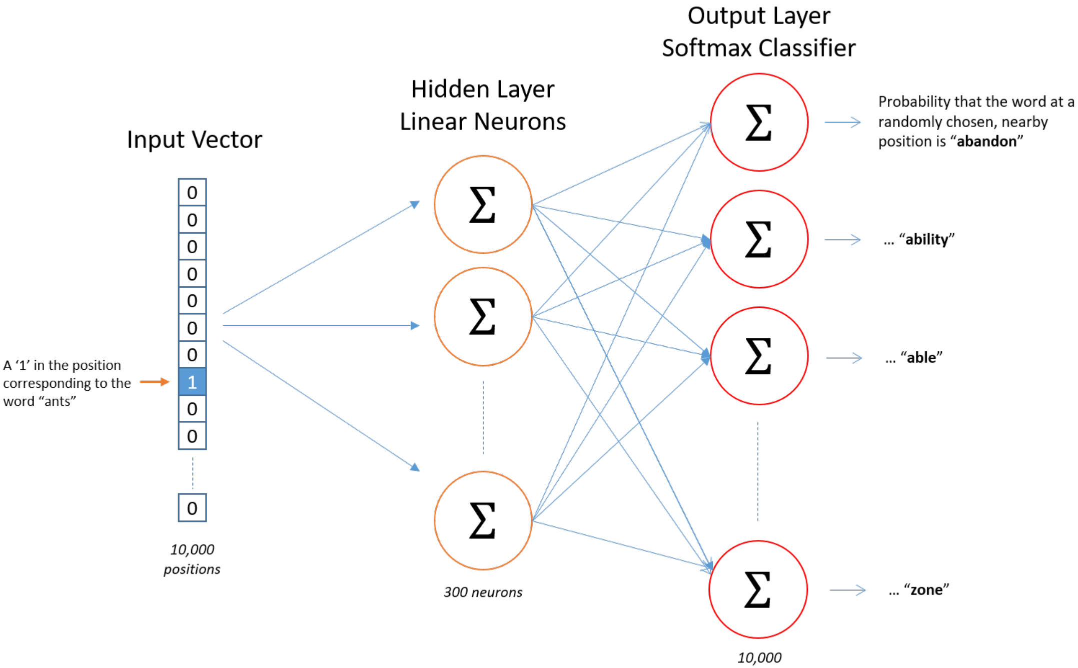
Neural Networks and Language Modeling

- Same goal: probability of a word in the neighborhood
- New setting: 20 years after Brown, neural networks on the rise
  - (Actually, Bengio et al. described a neural LM already in 2003. But it took another decade until a tractable variant appeared: word2vec.)

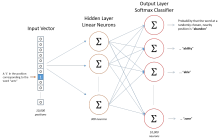
- Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean (2013). *Distributed Representations of Words and Phrases and their Compositionality*

- Skip-gram: probability that word y appears within a window of fixed size around word x

- *Images on following slides credit McCormick, Chris (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model. Retrieved from* `http://www.mccormickml.com`

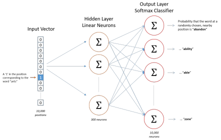

Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
300 neurons
10,000
Probability that the word at a randomly chosen, nearby position is "**abandon**"
... "**ability**"
... "**able**"
... "**zone**"

Skip-gram Neural Network



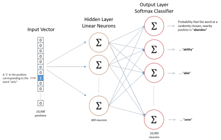
- Hidden layer = embedding layer = embedding matrix
- 300 neurons
  - Each neuron has its own set of weights for each input word
  - $\Rightarrow$ huge matrix 300 neurons $\times$ 10,000 words
  - Input word activates its own weight in each neuron

Skip-gram Neural Network



- Hidden layer = embedding layer = embedding matrix
- 300 neurons
  - Each neuron has its own set of weights for each input word
  - $\Rightarrow$ huge matrix 300 neurons $\times$ 10,000 words
  - Input word activates its own weight in each neuron
  - Another view:
    - Input word represented as one-hot vector (10,000 dimensions)
    - Embedding matrix 10,000 $\times$ 300 dimensions
    - Multiply input vector with embedding matrix
    - $\Rightarrow$ embedding vector for the word (300 dimensions)

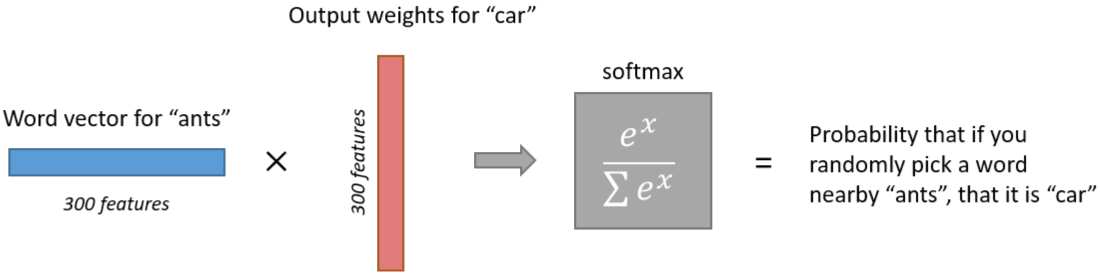
Skip-gram Neural Network



- Hidden layer = embedding layer = embedding matrix
- 300 neurons
  - Each neuron has its own set of weights for each input word
  - $\Rightarrow$ huge matrix 300 neurons $\times$ 10,000 words
  - Input word activates its own weight in each neuron

  - Output layer: 10,000 neurons
    - Each has a vector of 300 weights
    - Multiply word embedding vector with the weight vector
    - Normalize using softmax
    - $\Rightarrow$ probability of word y in the window!

Skip-gram Output Layer



The softmax function takes a vector of K real numbers and returns a normalized vector interpretable as a probability distribution:
$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$ for $i = 1, \ldots, K$ and $\mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$

Skip-gram Training

- Take a word x from the corpus
- Randomly pick a word y from the window around x

Skip-gram Training

- Take a word x from the corpus
- Randomly pick a word y from the window around x
- Input: one-hot vector for x

Skip-gram Training

- Take a word x from the corpus
- Randomly pick a word y from the window around x
- Input: one-hot vector for x
- Output: one-hot vector for y
  - As if we expected the network to allocate all probability to y now

Skip-gram Training

- Take a word x from the corpus
- Randomly pick a word y from the window around x
- Input: one-hot vector for x
- Output: one-hot vector for y
  - As if we expected the network to allocate all probability to y now
- Compare with the actual output given current weights

Skip-gram Training

- Take a word x from the corpus
- Randomly pick a word y from the window around x
- Input: one-hot vector for x
- Output: one-hot vector for y
    - As if we expected the network to allocate all probability to y now
- Compare with the actual output given current weights
- Loss function, gradient descent, back propagation $\Rightarrow$
- $\Rightarrow$ update weights!

Collect Embeddings

- Trained neural network for skip-gram prediction
- Take the embedding layer, discard the rest

- For each word: a vector of 300 dimensions (features)
- **“Similar” words have similar vectors!** (cf. Brown clusters)

Collect Embeddings

- Trained neural network for skip-gram prediction
- Take the embedding layer, discard the rest

- For each word: a vector of 300 dimensions (features)
- **“Similar” words have similar vectors!** (cf. Brown clusters)
- Cosine similarity: angle θ between vectors, regardless their magnitude. Same orientation $\Rightarrow$ similarity 1; orthogonal $\Rightarrow$ similarity 0

Given two vectors of attributes, A and B, the cosine similarity, $\cos()$, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

where A_i and B_i are components of vector A and B respectively.

Multilingual Word Representations

- Both clusters and embeddings are like partial delexicalization
- Unsupervised – they don't need annotated data!
- They rely on context of other words

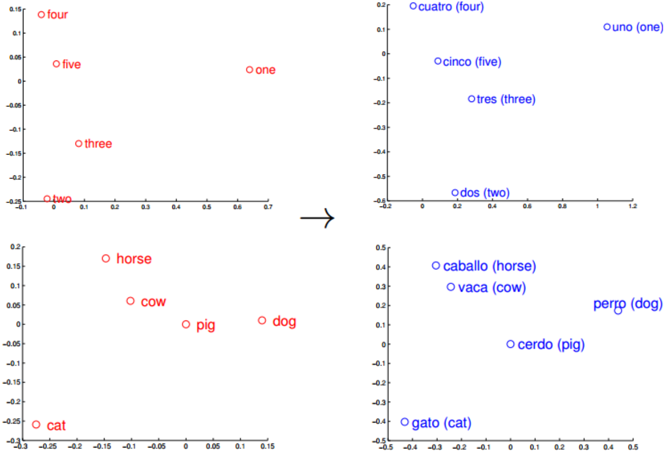
- Unfortunately do not work across languages:
  - You don't see a Czech word surrounded by English very often!

Brown Clusters Trained on Czech and English PUD

- 0000 of through Natural Yves captured comedy globálního independent opinion reduced
- 00010 with about over before around without since record better conquered down
- 001010 is like until began told found brought helps opened sent announced changed
- 0100 his their this her other these North both some New October each South April
- 0101 the its several our European my any your financial natural religious St. bad exact
- 01100 time up people years year city state government world day war own power area
- 10000 aby led včetně followed played pokud prezident aniž directed however
- 10001 že který které ale která co když jak kteří kde což kdy zatímco zda protože proč
- 100111 je byl bylo byla i jsou byly bude má mnoho už není například jsme každý mají
- 10100 roce letech době oblasti případě této druhé důsledku hlavní den jedné roli té
- 10110 na z s ve o k pro za od mezi ze proti pod u de svého nad nové kolem bez několik
- 111100 The In A But Na On During Po After However As At By Avšak With And

Aligning Vector Spaces

- Tomáš Mikolov, Quoc V. Le, Ilya Sutskever (2013). *Exploiting Similarities among Languages for Machine Translation*



Aligning Vector Spaces

- Large monolingual data in two languages
- Train monolingual word embeddings in both languages
- Take a small bilingual dictionary
- Transform (rotate & scale) source vector space to match the target space as closely as possible
  - For known translation pairs (x, z), transformed vector of x should be close to vector of z
  - $\{x_i, z_i\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^{d_1}$, and $z_i \in \mathbb{R}^{d_2}$
  - Find transformation matrix $W \in \mathbb{R}^{d_1 \times d_2}$ such that distance of transformed vectors from their translations is minimal:
$$\min_{W} \sum_{i=1}^{n} ||Wx_i - z_i||^2$$
  - Use neural network (gradient descent) to find the matrix!

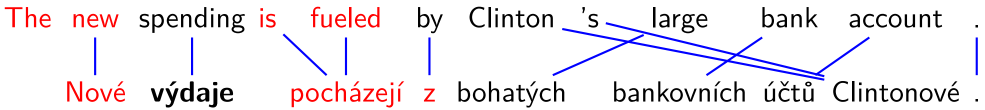
Spanish-English Example Translations

- emociones → **emotions,** emotion, feelings
- protegida → wetland, undevelopable, **protected**
- imperio → dictatorship, imperialism, tyranny (correct: empire)
- determinante → crucial, key, important (correct: determinant)
- preparada → **prepared,** ready, prepare
- millas → kilometers, kilometres, **miles**
- hablamos → talking, talked, **talk**
- destacaron → **highlighted,** emphasized, emphasised

English-Czech Examples of Ambiguities

English	Computed Translation	Dictionary
said	řekl *(said)*	uvedený *(listed)*
will	může *(can)*	vůle *(testament)*
did	udělal *(did)*	ano *(yes)*
hit	zasáhl *(hit)*	hit
must	musí *(must)*	mošt *(cider)*
current	stávající *(current)*	proud *(stream)*
shot	vystřelil *(shot)*	shot
minutes	minut *(minutes)*	zápis *(record)*
latest	nejnovější *(newest)*	poslední *(last)*
blacks	černoši *(black people)*	černá *(black color)*
hub	centrum *(center)*	hub
minus	minus *(minus)*	bez *(without)*
retiring	odejde *(will leave)*	uzavřený *(closed)*
grown	pěstuje *(grows)*	dospělý *(adult)*

Skip-gram across Word Alignment



Skip-gram across Word Alignment

The new spending is fueled by Clinton 's large bank account .

Nové výdaje **pocházejí** z bohatých bankovních účtů Clintonové .

Skip-gram across Word Alignment

The new spending is fueled by Clinton 's large bank account .

Nové výdaje pocházejí **z** bohatých bankovních účtů Clintonové .

Skip-gram across Word Alignment

The new spending is fueled by Clinton 's large bank account .

Nové výdaje pocházejí z **bohatých** bankovních účtů Clintonové .

Skip-gram across Word Alignment

The new spending is fueled by Clinton 's large bank account .

Nové výdaje pocházejí z bohatých **bankovních** účtů Clintonové .

Skip-gram across Word Alignment

The new spending is fueled by Clinton 's large bank account .

Nové výdaje pocházejí z bohatých bankovních **účtů** Clintonové .

Skip-gram across Word Alignment

The new spending is fueled by Clinton 's large bank account .

Nové výdaje pocházejí z bohatých bankovních účtů **Clintonové** .

Artificial Code Switching

At random positions, jump across word alignment to the other language.

The new spending is fueled by Clinton 's large bank account .

Nové výdaje pocházejí z bohatých bankovních účtů Clintonové .

The Nové výdaje is fueled by bohatých bank account Clintonové .

Unsupervised Multilingual Word Embeddings

- Multilingual (as opposed to bilingual)
- Unsupervised: no bilingual dictionary or parallel corpus
- Input: Monolingual embeddings for many languages

- Xilun Chen, Claire Cardie (2018). *Unsupervised Multilingual Word Embeddings*

- Adversarial training
  - Train a network that says whether a vector is likely to come from embeddings of language l_i
  - Train a converter from other languages to an "interlingua" vector space and then to space of l_i
  - Try to make the converter so good that the classifier network cannot recognize converted vectors from domestic ones
- Pseudo-supervised second step:
  - After approximate alignment, we trust translations of high-frequency words
  - $\Rightarrow$ Generate a bilingual dictionary for each language pair
  - $\Rightarrow$ Use it in a "supervised" scenario to align the two languages better