WORD EMBEDDINGS II: GLOVE AND EXTENSIONS

Natural Language Processing

Sergey Nikolenko

Harbour Space University, Barcelona, Spain

January 15, 2018

GLOVE

GloVe

- GloVe – in a way, a variation of LSA.
- We are trying to approximate the cooccurrence matrix $X \in \mathbb{R}^{V \times V}$.
- Let X_{ij} be how many times word i cooccurs in our corpus with word j, $X_i = \sum_j X_{ij}$. Then

$$p_{ij} = p(j \mid i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}}.$$

  i.e., p_{ij} is the probability of the fact that word j occurs in the context of word i.
- If we tried to approximate the matrix of p_{ij}, it would be almost exactly like LSA.

- But we want to approximate the matrix of *ratios* $\frac{p_{ij}}{p_{kj}}$.
- The values $p_{ij} = p(j \mid i)$ themselves are hard to compare.
- But p_{ik} and p_{jk} *for the same word* k do become comparable.
- Example from a 6 billion token corpus:

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- We train the function

$$F(\mathbf{w}_i, \mathbf{w}_j; \tilde{\mathbf{w}}_k) \approx \frac{p_{ij}}{p_{jk}},$$

  where $\mathbf{w}_i$ and $\mathbf{w}_j$ are word vectors (embeddings) for words i and j in the space $\mathbb{R}^d$, and $\tilde{\mathbf{w}}_k$ are *context vectors* that ensure that we approximate the ratio in the context of k.
- What is F going to be?
- Theoretically, it could be a very complicated function, say, a deep neural network.
- But in reality we want the relations between word vectors to be simple (king−man+woman≈queen).

- So we train a simple function, assuming that

$$F((\mathbf{w}_i - \mathbf{w}_j)^\top \tilde{\mathbf{w}}_k) = \frac{F\left(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k\right)}{F\left(\mathbf{w}_j^\top \tilde{\mathbf{w}}_k\right)} = \frac{p_{ij}}{p_{kj}}.$$

  This makes the model train simple relations between word vectors.
- And one more reasonable assumption: F shouldn't change when we pass from X to $X^\top$ and from $\mathbf{w}$ to $\tilde{\mathbf{w}}$.
- To add this symmetry, we assume that F not only maps $\mathbf{w}_i - \mathbf{w}_j$ to the ratio of probabilities, but in general maps sums of arguments to products of function values:

$$F((\mathbf{w}_i - \mathbf{w}_j)^\top \tilde{\mathbf{w}}_k) = \frac{F\left(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k\right)}{F\left(\mathbf{w}_j^\top \tilde{\mathbf{w}}_k\right)} = \frac{p_{ij}}{p_{jk}}.$$

- What kind of a function is F then?

- F actually has to be an exponent:

$$\mathbf{w}_i^\top \tilde{\mathbf{w}}_k = \log(p_{ik}) = \log(X_{ik}) - \log(X_i).$$

- We can hide $\log(X_i)$ in bias terms $\mathbf{b}_i$, getting a nice symmetric model:

$$\mathbf{w}_i^\top \tilde{\mathbf{w}}_k + b_i + \tilde{b}_k = \log(X_{ik}).$$

- Two problems left:
  - $\log(X_{ik})$ very often diverges because X_{ik} is often zero; generally, X is a very sparse matrix;
  - the model treats all X_{ik} the same, but for rare words the ratio is very random, and for very frequent words it's not very important.
- In GloVe, we solve these problems by training $\mathbf{w}$ and $\tilde{\mathbf{w}}$ via *weighted* sum of squares loss function.

- Thus, the objective function for GloVe will be

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2,$$

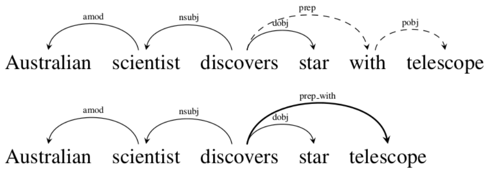
where f is a nondecreasing function with $f(0) = 0$ that doesn't grow too fast, e.g.,

$$f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^\alpha, & \text{if } x < x_{\max}, \\ 1 & \text{otherwise.} \end{cases}$$

- Demo: nearest neighbors, geometric relations.

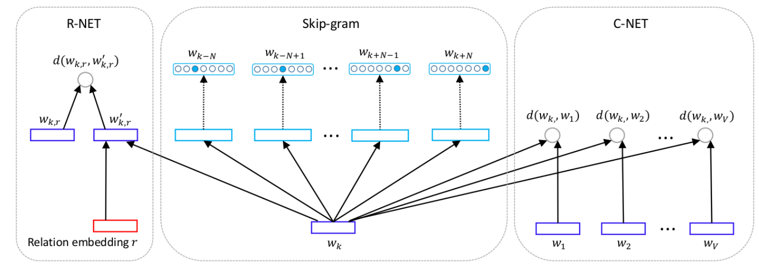
EXTENSIONS

- Some modifications of word embeddings add external information.
- (Levy et al., 2014): use dependency parsing for local context.



WORD	CONTEXTS
australian	scientist/amod^{-1}
scientist	australian/amod, discovers/nsubj^{-1}
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj^{-1}
telescope	discovers/prep_with^{-1}

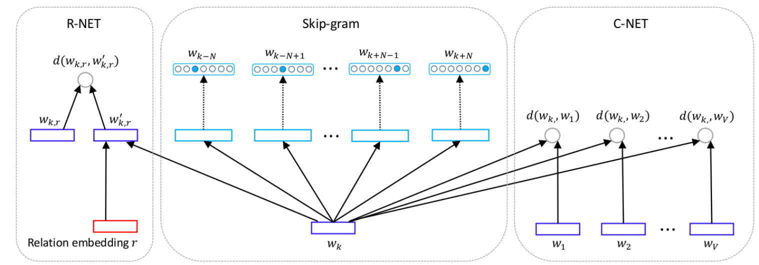
- The RC-NET model (Xu et al. 2014) extends skip-grams with relations (semantic and syntactic) and categorical knowledge (sets of synonyms, domain knowledge etc.).



- We would like to add relations from *Freebase* or similar knowledge bases – how?

- The basic *word2vec* model gets a regularizer for every relation that tries to bring it closer to a linear relation between the vectors, so that, e.g.,

$$\mathbf{w}_{\text{Hinton}} - \mathbf{w}_{\text{Wimbledon}} \approx r_{\text{born at}} \approx \mathbf{w}_{\text{Euler}} - \mathbf{w}_{\text{Basel}}$$



- Another important problem with both word vectors and char-level models: homonyms.
- How do we distinguish different senses of the same word?
  - the model usually just chooses one meaning;
  - e.g., let's check nearest neighbors for words like **converse**, **jaguar**, and other homonyms.
- We have to add *latent* variables for different meaning and infer them from context.

- To train the meanings with latent variables — Bayesian inference with stochastic variational inference (Bartunov et al., 2015).
- Problem: we don't know in advance how many senses a word has.
- Basic idea – set a prior distribution that allows for any number of senses, just with decreasing probabilities.
- Stick-breaking priors on the senses z_w:

$$p(z = k \mid w, \beta) = \beta_{wk} \prod_{r=1}^{k-1} (1 - \beta_{wr}), \quad p(\beta_{wk} \mid \alpha) = \mathrm{Beta}(\beta_{wk} \mid 1, \alpha).$$

- The total likelihood is now

$$p(C, Z, \beta \mid W, \alpha, \theta) =$$
$$= \prod_{w=1}^{V} \prod_{k=1}^{\infty} p(\beta_{wk} \mid \alpha) \prod_{i=1}^{N} p(z_i \mid w_i, \beta) \prod_{j=1}^{N} p(c_{ij} \mid z_i, w_i, \theta).$$

- And we are optimizing

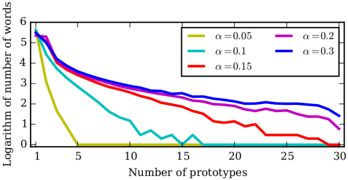
$$p(C \mid W, \alpha, \theta) = \int \sum_{Z} p(C, Z, \beta \mid W, \alpha, \theta) \mathrm{d}\beta.$$

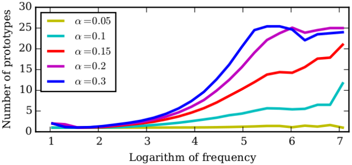
- Hard but possible to optimize – stochastic variational inference.

- Nice results:

ALPHA	"LIGHT" $p(z)$	nearest neighbours	"CORE" $p(z)$	nearest neighbours
Skip-gram	1.00	far-red, emitting	1.00	cores, component, i7
0.05	1.00	far-red, illumination	0.40	corium, cores, sub-critical
			0.60	basic, i7, standards-based
0.075	0.28	armoured, amx-13, kilcrease	0.30	competencies, curriculum
	0.72	bright, sunlight, luminous	0.34	cpu, cores, i7, powerxcell
			0.36	nucleus, backbone
0.1	0.09	tvärbanan, hudson-bergen	0.21	reactor, hydrogen-rich
	0.17	dark, bright, green	0.13	intel, processors
	0.09	4th, dragoons, 2nd	0.27	curricular, competencies
	0.26	radiation, ultraviolet	0.15	downtown, cores, center
	0.28	darkness, shining, shadows	0.24	nucleus, rag-tag, roster
	0.11	self-propelled, armored		

- The hyperparameter α controls how many senses are probable:





THANK YOU!

Thank you for your attention!