Distributional Semantics

Computational Methods for Text Analysis

Alena Pestova

HSE Saint-Petersburg

08.11.2022

The Distributional Hypothesis

Firth 1957

You shall know a word by the company it keeps.

The Distributional Hypothesis

Harris 1954

The fact that, for example, not every adjectives occurs with every noun can be used as a measure of meaning difference. For it is not merely that different members of the one class have different selections of members of the other class with which they are actually found. More than that: if we consider words or morphemes A and B to be more different than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference in meaning correlates with difference in distribution.

In simple words: words that occur in the same contexts tend to have similar meanings.

Example

	Lemma	Lemma	Частотность	Употреблений на миллион	
1	маленький	девочка	668	7,40	•••
2	маленький	ребенок	541	6,00	•••
3	маленький	мальчик	342	3,79	•••
4	маленький	пионер	12	0,13	•••
5	маленький	пионерка	4	0,04	•••
6	долговязый	мальчик	240	2,66	•••
7	долговязый	девочка	3	0,03	•••
8	хороший	мальчик	236	2,62	•••
9	хороший	девочка	234	2,59	•••
10	хороший	ребенок	60	0,67	•••
11	хороший	пионер	18	0,20	•••
12	хороший	пионерка	9	0,10	•••
13	красивый	девочка	136	1,51	•••
14	красивый	мальчик	32	0,35	•••
15	красивый	ребенок	9	0,10	•••
16	рыжий	девочка	129	1,43	•••
17	рыжий	мальчик	42	0,47	•••
18	рыжий	пионер	2	0,02	•••

Why is it useful?

- we can try to use this knowledge for building word vector representations
- they will help us to solve very different tasks

Distributional Semantic Models

- words are represented as real-valued vectors built from their distribution in contexts
- similarity between words is approximated in terms of their geometric distance between vectors
- build as general-purpose semantic models that can be applied to various tasks (resulting vectors are not task specific)
- rely on a co-occurrence matrix

Co-occurence matrix

- Each row in the matrix M represents a word w_i
- Each column in the matrix M represents a context c_j (depending on window size: number of words in the context considered; frequently paragraphs or documents as below)
- Matrix M_{ij} represents the strength of association
- term-document matrices can get very large => need dimensionality reduction
- compute similarity between two words based on row vectors

	d_1	d_2	d_3	d_4	d_5
dog	88	92	11	1	2
lion	57	28	3	0	0
bark	80	62	10	0	1
car	0	1	0	93	97
tire	2	0	2	80	72
drive	0	1	0	90	45

Association and similarity

- Pointwise-mutual information (PMI)
- Cosine similarity/distance between vectors

Dimensionality Reduction

One available method is Singular Value Decomposition (SVD)):

- Factorizes matrix M into three matrices: $M_{m,n} = U_{m,m}\Sigma_{m,n}V_{n,n}^T$
- U and V represent orthogonal matrices and Σ is a diagonal matrix
- we want to select the k top singular values to obtain lower dimensional vectors that account for the maximum proportion of the original variance
- we want to get the best rank-d approximation of M
- computing the truncated projection: $M_{reduced} = U_{m,k}\Sigma_{k,k}V_{n,k}^T$
- now, we have vectors with length less than original vectors

LSA-SVD Example

Matrix M_{mxn}:

	d_1	d_2	d_3	d_4	d_5
dog	88	92	11	1	2
lion	57	28	3	0	0
bark	80	62	10	0	1
car	0	1	0	93	97
tire	2	0	2	80	72
drive	0	1	0	90	45

Matrix U_{mxm} (word-to-concept):

	1	2	3	4	5	6
dog	-0.059	0.73	0.0038	-0.58	-0.26	0.25
lion	-0.023	0.35	0.0069	0.77	-0.45	0.28
bark	-0.042	0.58	-0.0072	0.25	0.59	-0.49
car	-0.67	-0.053	-0.52	-0.034	-0.33	-0.41
tire	-0.54	-0.037	-0.13	0.063	0.51	0.65
drive	-0.49	-0.039	0.85	-0.013	-0.11	-0.16

Matrix Σ_{mxn}

197.6	0	0	0	0
0	**173.9**	0	0	0
0	0	27.78	0	0
0	0	0	20.93	0
0	0	0	0	2.744
0	0	0	0	0

Matrix $V^{\top}_{nxn}$ (document-to-concept):

	d_1	d_2	d_3	d_4	d_5
1	-0.055	-0.05	-0.011	-0.76	-0.64
2	0.75	0.65	0.085	-0.061	-0.043
3	-0.0037	0.015	-0.0097	0.64	-0.77
4	0.66	-0.75	-0.065	0.01	-0.0092
5	-0.022	-0.11	0.99	0.0036	-0.012

LSA-SVD Example

Matrix $M_{reduced}$:

	d_1	d_2	d_3	d_4	d_5
dog	95.85	83.10	10.92	1.12	2.00
lion	45.90	39.79	5.22	-0.26	0.29
bark	76.10	65.98	8.66	0.15	0.97
car	0.37	0.63	0.67	101.18	85.13
tire	1.04	1.15	0.63	81.49	68.57
drive	0.24	0.43	0.49	74.00	62.26

Matrix U_{mxk} (word-to-concept):

	1	2	3	4	5	6
dog	-0.059	0.73	0	0	0	0
lion	-0.023	0.35	0	0	0	0
bark	-0.042	0.58	0	0	0	0
car	-0.67	-0.053	0	0	0	0
tire	-0.54	-0.037	0	0	0	0
drive	-0.49	-0.039	0	0	0	0

Matrix Σ_{kxk}:

197.6	0	0	0	0
0	**173.9**	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Matrix V^T_{nxk} (document-to-concept):

	d_1	d_2	d_3	d_4	d_5
1	-0.055	-0.05	-0.011	-0.76	-0.64
2	0.75	0.65	0.085	-0.061	-0.043
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

LSA-SVD Example

- Cosine similarity between d1 and d3 originally 0.96
- Cosine similarity between d1 and d3 reduced 0.99
- Cosine similarity between lion and car originally 0.0033
- Cosine similarity between lion and car reduced 0.0054
- etc.

Latent Semantic Analysis (LSA)

Words and documents - vectors in the semantic space, dimensions which are "latent"variables.

- co-occurring words are projected onto the same dimensions;
- vector for the document - the weighted sum of the vectors of the words included in it (centroid);
- in semantic space the angle between document vectors may be small, even if there are no common words in the documents.

Disadvantages of LSA-SVD

- Relatively poor quality of obtained representstions
- The complexity of working with a very large and sparse matrix
- Difficulty in adding new words/documents

Word Embeddings

Definition: Real-valued and sub-symbolic representations of words as dense numeric vectors.

- distributed representation of word meanings (not count-based)
- usually learned with neural networks
- specific dimensions of the resulting vectors cannot bet directly mapped to symbolic representation
- models that seek to predict between a center word and context words (predict models)
- key elements of deep learning models

Word Embeddings

Word Embeddings (WE) is a popular framework in NLP that allows to represent meaning of words and phrases as vectors of real numbers (Mikolov et al., 2013).
WE has become a very common in many NLP tasks, such as machine translation, classification, document ranking, sentiment analysis.

Methods to Train Word Embeddings

- word2vec (CBOW and SGNS)
- FastText - extension of Word2Vec, train on subword information (n-grams)
- Clove
- and many others (with almost all other NLP neural networks)

Idea of word2vec

Framework for learning word embeddings; main idea:

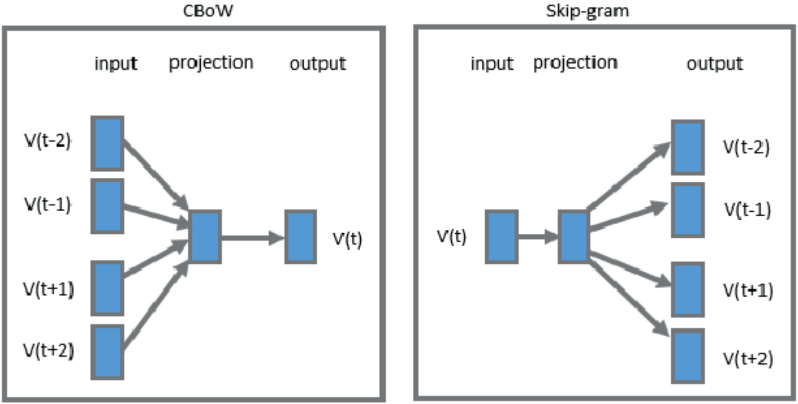
- takes words from a very large corpus of text as input (unsupervised)
- learn a vector representation for each word to predict between every word and its context

Two main algorithms:

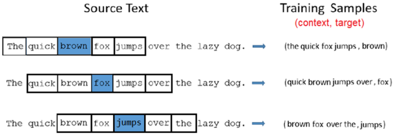
- Continuous Bag of Words (CBOW): predicts center word from the given context (sum of surrounding words vectors)
- Skip-gram (SGNS): predicts context taking the center word as input

Idea of word2vec

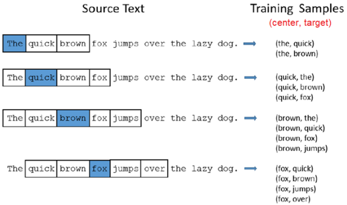
Don't count, predict!



Idea of word2vec



CBOW

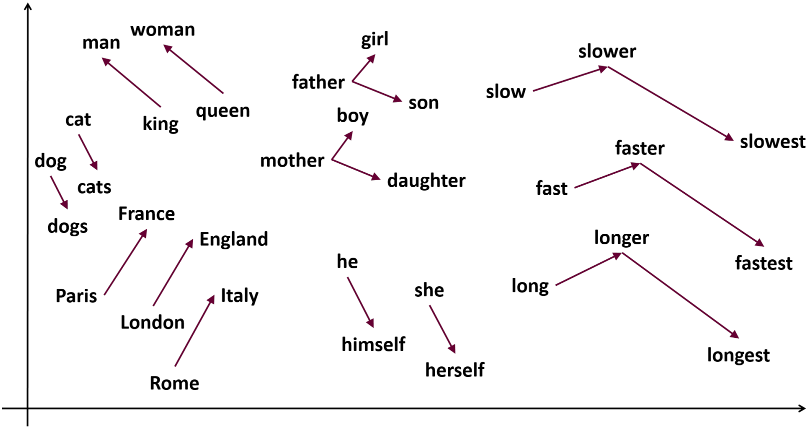


Skipgram

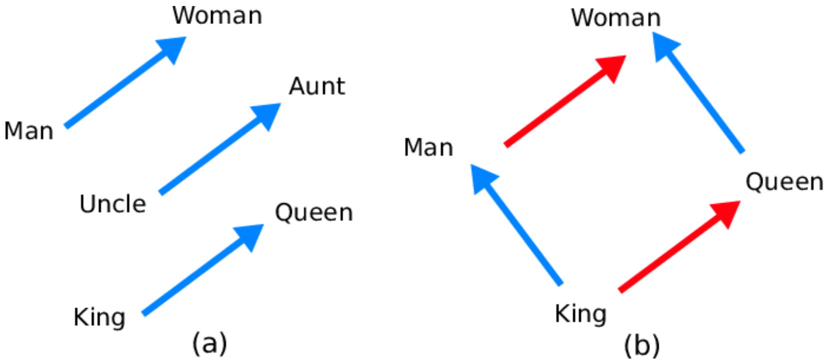
Word2vec hyperparameters

- corpus size / source / type
- text preprocessing
- model type
- context window (size of the window)
- vector dimensions

Semantic relations between words



Semantic relations between words



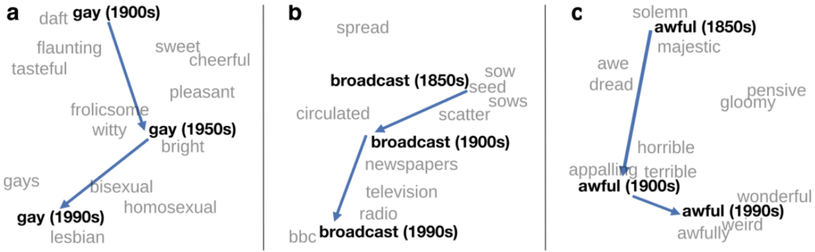
Trained WV for Russian

- Rusvectores https://rusvectores.org/ru/models/
- Navec (from Natasha project) https://github.com/natasha/navec
- Deeppavlov http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html
- and many many others

WE in social science research

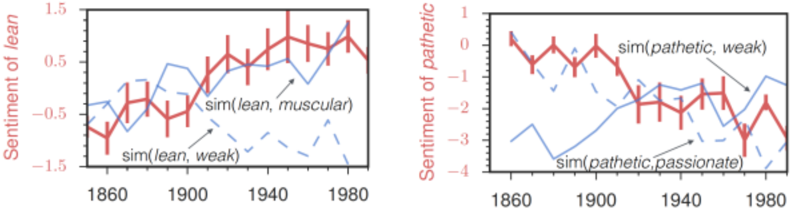
- Train WE on some corpus of interest
- Select some words/sets of words corresponding to some concepts from your research
- Look at the words relations, change of this relation in time

Examples of WE for social science research



Hamilton W. L., Leskovec J., Jurafsky D. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change //Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). – 2016. – C. 1489-1501.

Examples of WE for social science research



Hamilton W. L. et al. Inducing domain-specific sentiment lexicons from unlabeled corpora //Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing. – NIH Public Access, 2016. – T. 2016. – C. 595.

Senetences and documents embeddings

- Often in the task of text analysis, the necessary embeddings are not words in documents, but the documents themselves or their parts
- The easiest way to get it is a weighted sum word embeddings (e.g. with tf-idf as weights) This approach shows good results and is often used in practice, but there are more interesting methods
- If there is a good way to find sentence embedding, vector document can again be obtained by averaging over sentences
- Quality is assessed by the task metric (external criterion) or, for example, searching for the quality of labeled analogues (internal)
- many other methods, for example, doc2vec