Distributional Semantics

Lecture 2. Count-based Distributional Models

Daniil Vodolazsky

March 2nd, 2019

Lecture Plan

Lecture Plan

1. First Vector Space Models: Matrix of Word Co-occurrence Counts
2. Measures of Word Co-occurrence
3. Similarity Measures
4. Matrix Factorization. SVD, PCA
5. Count-based Distributional Models Based on Matrix Factorization: LSA, LSI
6. Clustering
7. Pros and Cons of Count-based Models
8. Seminar

First Vector Space Models: Matrix of Word Co-occurrence Counts

Term-document Matrix

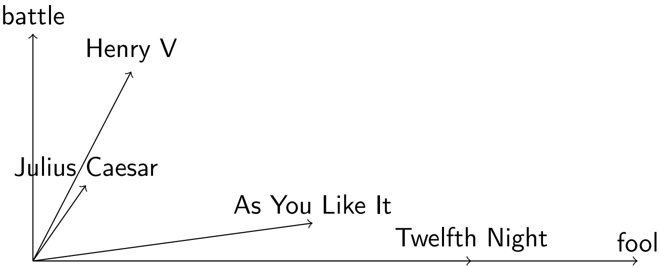
In a **term-document matrix**, each row represents a word in the vocabulary and each column represents a document from some collection of documents.

	As You Like It	**Twelfth Night**	**Julius Caesar**	**Henry V**
battle	1	0	7	13
good	115	81	71	91
fool	37	58	2	5
wit	21	15	2	3

The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

Term-document Matrix

Two terms / documents are **similar**, if their vectors are similar.



Vector space projection on 2-dimensional fool-battle space.

Term-term and Document-document Matrices

However, it is most common to use a different kind of context for the dimensions of a word's vector representation. Rather than the term-document matrix we use the **term-term matrix**, more commonly called the **word-word matrix**, the **word-context matrix** or the **term-context matrix**, in which the columns are labeled by words rather than documents.

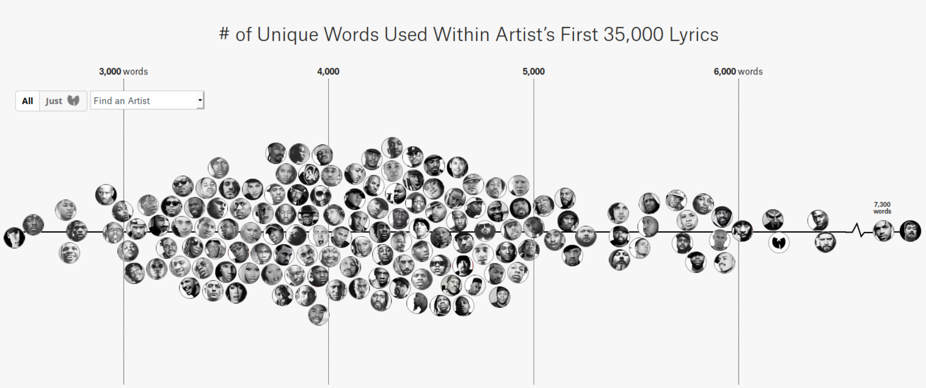
- term-document matrix: $|\mathcal{W}| \times |\mathcal{D}|$. How many times a given word appears in a given document?
- term-term matrix: $|\mathcal{W}| \times |\mathcal{W}|$. How many times two given words appear in the same documents?
- document-document matrix: $|\mathcal{D}| \times |\mathcal{D}|$. How many words appear in two given documents?

Measures of Word Co-occurrence

Binary Matrix

$$W_{w,d} = \operatorname{sgn} \#(w, d).$$

This means that in this case we are interested only in whether a word appeared in a document.



▸ Source

TF-IDF

The **term frequency—inverse document frequency** or **TF-IDF** is a numerical statistics that tries to point out how important is a word in a given document.
The **term frequency (TF)** is proportional to the number of times a given query word *w* is found in the corpora. The simplest and most preferred choice is to use simple raw counts of a word *w* and a document *d*:

$$TF_{w,d} = \#(w, d).$$

An **inverse document frequency (IDF)** (1972) is a factor increasing weights of unfrequent words and decreasing weights of frequent words.
To simplify further formulas, we will denote $\mathcal{D}_{|w} := \{d \in \mathcal{D} : w \in d\}$.
Thus,

$$IDF_w = \log \frac{|\mathcal{D}|}{|\mathcal{D}_{|w}|}.$$

Thus for the elements of TF-IDF matrix we have the following formula:

$$W_{w,d} = TF_{w,d} \cdot IDF_w.$$

PMI. Collocation Extraction

Pointwise mutual information (PMI) (1961) is a measure of how often two random variables x and y occur, compared with what we would expect if they were independent:

$$I(\mathrm{x},\mathrm{y}) = \log_2 \frac{P(\mathrm{x},\mathrm{y})}{P(\mathrm{x})P(\mathrm{y})} = \log_2 \frac{P(\mathrm{x} \mid \mathrm{y})P(\mathrm{y})}{P(\mathrm{x})P(\mathrm{y})} = \log_2 \frac{P(\mathrm{x} \mid \mathrm{y})}{P(\mathrm{x})} = \log_2 \frac{P(\mathrm{y} \mid \mathrm{x})P(\mathrm{x})}{P(\mathrm{x})P(\mathrm{y})} = \log_2 \frac{P(\mathrm{y} \mid \mathrm{x})}{P(\mathrm{y})}.$$

If $\mathrm{x} \perp \mathrm{y}$ then $P(\mathrm{x} \mid \mathrm{y}) = P(\mathrm{x})$ and $P(\mathrm{y} \mid \mathrm{x}) = P(\mathrm{y})$, and thus $I(\mathrm{x},\mathrm{y}) = 0$.

PMI. Collocation Extraction

The **pointwise mutual information (PMI)** (1989) between a target word w and a context word c is then defined as:

$$PMI_{w,c} = \log_2 \frac{P(w,c)}{P(w)P(c)}.$$

The numerator tells us how often we observed the two words together (assuming we compute probability by using the MLE). The denominator tells us how often we would expect the two words to co-occur assuming they each occurred independently. Thus, the ratio gives us an estimate of how much more the two words co-occur than we expect by chance. PMI is a useful tool whenever we need to find words that are strongly associated.

However, it is more common to use **positive PMI (PPMI)** which replaces all negative PMI values with zero:

$$PPMI_{w,c} = \max(PMI_{w,c}, 0) = PMI^{+}_{w,c}.$$

PMI. Collocation Extraction

w	c	$\#(w)$	$\#(c)$	$\#(w,c)$	PMI
puerto	rico	1938	1311	1159	10.0349081703
hong	kong	2438	2694	2205	9.72831972408
los	angeles	3501	2808	2791	9.56067615065
carbon	dioxide	4265	1353	1032	9.09852946116
san	francisco	5237	2477	1779	8.83305176711
ice	hockey	5607	3002	1933	8.6555759741
it	the	283891	3293296	3347	-1.72037278119
are	of	234458	1761436	1019	-2.09254205335
this	the	199882	3293296	1211	-2.38612756961
is	of	565679	1761436	1562	-2.54614706831
a	and	984442	1375396	1457	-2.92239510038
in	and	1187652	1375396	1537	-3.05660070757
of	and	1761436	1375396	1190	-3.70663100173

The results of applying PMI to search collocations.

Similarity Measures

Cosine Similarity

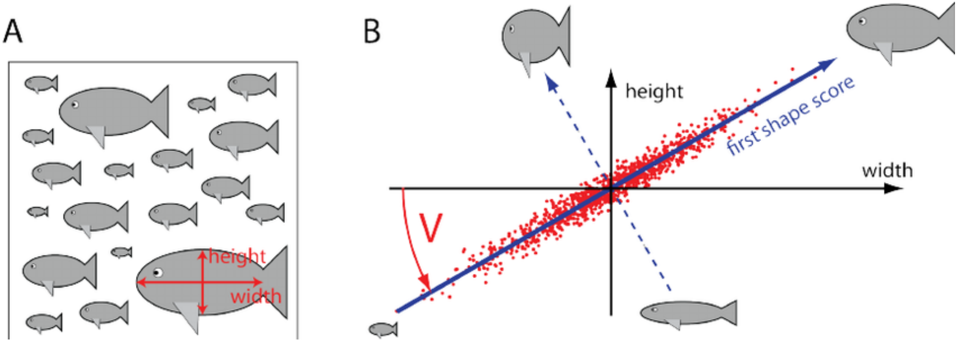
To define similarity between two target words u and v, we need a measure for taking two such vectors and giving a measure of vector similarity. By far the most common similarity metric is the **cosine** of the angle between the vectors:

$$\cos(u, v) = \frac{u \cdot v}{||u|| \cdot ||v||} = \frac{\sum_{i=1}^{N} u_i v_i}{\sqrt{\sum_{i=1}^{N} u_i^2}\sqrt{\sum_{i=1}^{N} v_i^2}}.$$

Matrix Factorization. SVD, PCA

PCA

Principal component analysis (PCA) (1901) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.



SVD

Perhaps the most known and widely used matrix decomposition method is the **singular-value decomposition (SVD)** (1936).

SVD Theorem. Suppose A is a $m \times n$ real-valued matrix. Then there exists a factorization, called a «singular value decomposition» of A, of the form

$$A = U\Sigma V^\top$$

where U is an $m \times m$ orthogonal matrix, Σ is a diagonal $m \times n$ matrix with non-negative real numbers on the diagonal, V is an $n \times n$ orthogonal matrix.

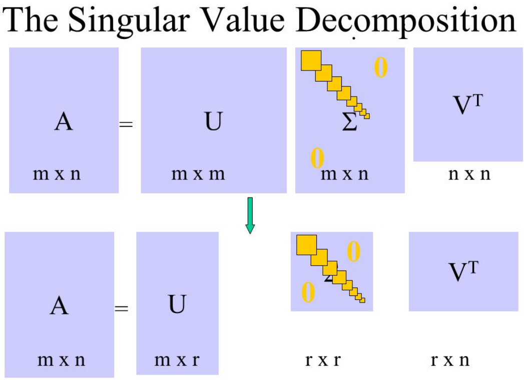
The diagonal entries σ_i of Σ are known as the **singular values** of A.

SVD

Some practical applications need to solve the problem of approximating a matrix A with another matrix $\tilde{A}$, which has a specific rank r. It turns out that the solution is given by the SVD of A, namely

$$\tilde{A} = U\tilde{\Sigma}V^{\top}$$

where $\tilde{\Sigma}$ is the same matrix as Σ except that it contains only the r largest singular values.



Count-based Distributional Models Based on Matrix Factorization: LSA, LSI

LSA

Latent semantic analysis (LSA) (1970s) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations. The underlying idea is that the aggregate of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other.

LSA's Ability to Model Human Conceptual Knowledge

How well does LSA actually work as a representational model and measure of human verbal concepts? Its performance has been assessed more or less rigorously in several ways. LSA was assessed as

1. a predictor of query-document topic similarity judgments.
2. a simulation of agreed upon word-word relations and of human vocabulary test synonym judgments.
3. a simulation of human choices on subject-matter multiple choice tests.
4. a predictor of text coherence and resulting comprehension.
5. a simulation of word-word and passage-word relations found in lexical priming experiments.
6. a predictor of subjective ratings of text properties, i.e. grades assigned to essays.
7. a predictor of appropriate matches of instructional text to learners.
8. LSA has been used with good results to mimic synonym, antonym, singular-plural and compound-component word relations, aspects of some classical word sorting studies, to simulate aspects of imputed human representation of single digits, and, in pilot studies, to replicate semantic categorical clusterings of words found in certain neuropsychological deficits.

LSI

J. R. Anderson (1990) has called attention to the analogy between **information retrieval** and human semantic memory processes. One way of expressing their commonality is to think of a searcher as having in mind a certain meaning, which he or she expresses in words, and the system as trying to find a text with the same meaning.

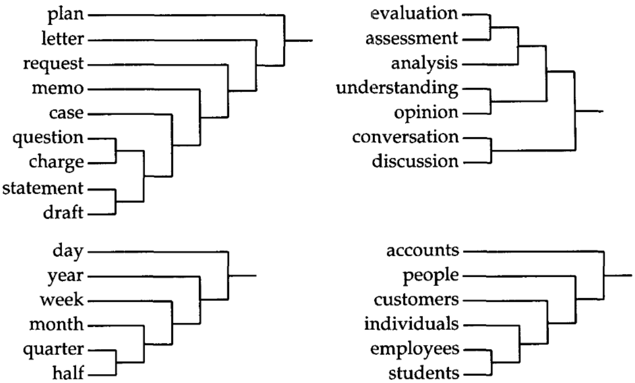
Latent semantic indexing (**LSI** (1990); LSA's alias in this application) does this better than systems that depend on literal matches between terms in queries and documents.

The document database is first represented as a term-document matrix and subjected to SVD, and each word and document is represented as a reduced dimensionality vector, usually with 50–400 dimensions. A **query** is represented as a "pseudo-document" a weighted average of the vectors of the words it contains.

Clustering

Brown Clustering

A **clustering**, or cluster analysis, is the process of creating sets of similar elements.
The **Brown clustering** algorithm (1992) groups elements of a vocabulary $\mathcal{W}$ into k clusters. The main idea of Brown's model is that the words from one cluster often appear after the words from other clusters. Developing of this idea leads to hierarchical clustering.



Pros and Cons of Count-based Models

Pros and Cons of Count-based Models

Advantages

- They have been used and explored for decades. Hence, results found are for most of them very well known and accurate. The support for these algorithms covers an impressive range going from linear algebra to information theory and statistics.
- As soon as sets and corpora are small enough, for a given task, accuracy, efficiency and speed are going to be very high.

Drawbacks

- The order of entities in a sentence does not matter: «A cat is chasing a dog» is equivalent to «A dog is chasing a cat».
- Linked with the idea of «no grammar structure» encoded, issues about ambiguous words can be raised.
- It may appear that some words do not belong to the training data set and then are considered as unknown (out-of-vocabulary or OOV words). In such case, one variant is to recompute all matrices, but it will possibly require a lot of time.

Questions?

Seminar

A Simple Information Retrieval System

1. How to preprocess data
2. gensim usage
3. Bag-of-words search engine
4. TF-IDF model
5. Doing SVD / LSA with your own hands
6. LSI
7. Homework

https://www.kaggle.com/s231644/a-simple-information-retrieval-system

▸ Link on a notebook