Distributed Semantic Models for Word Vectors

Correlated Occurrence Analogue to Lexical Semantic (COALS)

Ramaseshan Ramachandran

You shall know a word by the company it keeps - Firth

You shall know a word by the company it keeps - Firth

- A word's meaning is derived from the contexts in which it appears

You shall know a word by the company it keeps - Firth

- A word's meaning is derived from the contexts in which it appears
- We can represent this company in two primary ways:
  - **First-Order:**
    The words a target word appears *with*
  - **Second-Order:**
    The words that appear in **similar contexts** as the target word

FIRST-ORDER CO-OCCURRENCE (SYNTAGMATIC RELATIONS)

- Measures direct co-occurrence of words within a specific context window (weighted/unweighted)

FIRST-ORDER CO-OCCURRENCE (SYNTAGMATIC RELATIONS)

- Measures direct co-occurrence of words within a specific context window (weighted/unweighted)
- Captures **syntagmatic** relationships: words that often appear together in sentences.

FIRST-ORDER CO-OCCURRENCE (SYNTAGMATIC RELATIONS)

- Measures direct co-occurrence of words within a specific context window (weighted/unweighted)
- Captures **syntagmatic** relationships: words that often appear together in sentences.
- **Example:** The words **dog**, barks, and leash have strong **first-order** associations.

FIRST-ORDER CO-OCCURRENCE (SYNTAGMATIC RELATIONS)

- Measures direct co-occurrence of words within a specific context window (weighted/unweighted)
- Captures **syntagmatic** relationships: words that often appear together in sentences.
- **Example:** The words **dog**, barks, and leash have strong **first-order** associations.
- This is typically modeled with a word-context co-occurrence matrix, often denoted as M.

FIRST-ORDER CO-OCCURRENCE (SYNTAGMATIC RELATIONS)

- Measures direct co-occurrence of words within a specific context window (weighted/unweighted)
- Captures **syntagmatic** relationships: words that often appear together in sentences.
- **Example:** The words **dog**, barks, and leash have strong **first-order** associations.
- This is typically modeled with a word-context co-occurrence matrix, often denoted as M.

The association is a direct lookup in the matrix: $\text{assoc}_1(w, c) = M_{wc}$

where w and c are the target and context words, respectively

and M_{wc} is the frequency, correlation, entropy-based or PPMI of w co-occurring with c

SECOND-ORDER CO-OCCURRENCE (PARADIGMATIC RELATIONS)

- Measures the similarity of the **contexts** in which two different words appear

SECOND-ORDER CO-OCCURRENCE (PARADIGMATIC RELATIONS)

- Measures the similarity of the **contexts** in which two different words appear
- Captures **paradigmatic** relationships: words that are semantically similar or substitutable.

SECOND-ORDER CO-OCCURRENCE (PARADIGMATIC RELATIONS)

- Measures the similarity of the **contexts** in which two different words appear
- Captures **paradigmatic** relationships: words that are semantically similar or substitutable.
- **Example:** **dog** and **cat** are similar because they both co-occur with pet, food, vet, and animal

SECOND-ORDER CO-OCCURRENCE (PARADIGMATIC RELATIONS)

- Measures the similarity of the **contexts** in which two different words appear
- Captures **paradigmatic** relationships: words that are semantically similar or substitutable.
- **Example:** **dog** and **cat** are similar because they both co-occur with pet, food, vet, and animal
- It is an indirect relationship: **dog** and **cat** might never appear together, but they appeared in similar contexts

SECOND-ORDER CO-OCCURRENCE (PARADIGMATIC RELATIONS)

- Measures the similarity of the **contexts** in which two different words appear
- Captures **paradigmatic** relationships: words that are semantically similar or substitutable.
- **Example:** **dog** and **cat** are similar because they both co-occur with pet, food, vet, and animal
- It is an indirect relationship: **dog** and **cat** might never appear together, but they appeared in similar contexts

The association is the similarity between two word vectors from the first-order matrix:

$$\text{assoc}_2(w_1, w_2) = \text{sim}(\vec{v}_{w_1}, \vec{v}_{w_2})$$

- where $\vec{v}_{w_1}$ and $\vec{v}_{w_2}$ are the context vectors (rows) for words w_1 and w_2 from the matrix M.
- sim is a similarity function, commonly Cosine Similarity.

ANOTHER EXAMPLE

The view from the top of the mountain was
The view from the summit was
La vue du sommet de la montagne était
Mtazamo wa juu wa mlima huo ulikuwa

awesome/(*impressionnante*, *impressionnant*)
breathtaking
amazing, அற்புதமான/అద్భుతకరమాయ/
stunning/(*superbe*)/అద్భుతమైన
astounding अद्भुत/চমকপ্রদ
astonishing
awe-inspiring
extraordinary
incredible/(*incroyable*)
unbelievable
magnificent शानदार/ಗಂಭೀರಮಾಯ/ભવ્ય
wonderful/(*ajabu*)
spectacular
remarkable/(*yakuvutia*)

Paradigmatic Axis

Syntagmatic Axis

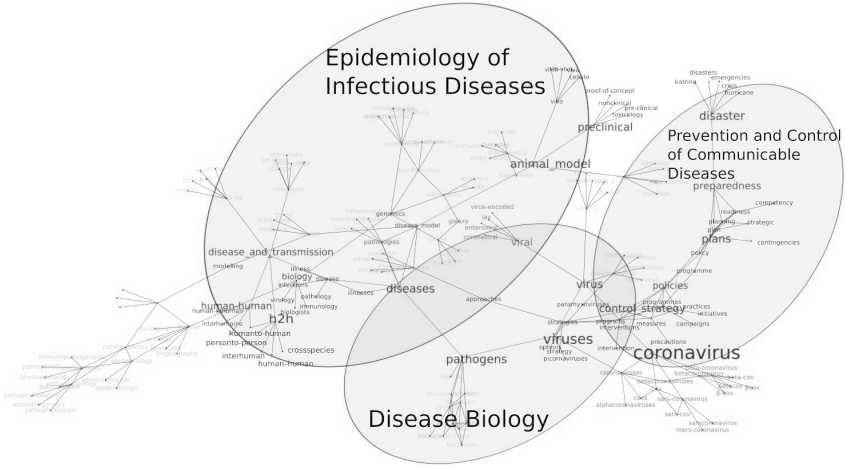
First Order Association

- Pairs of words in common contexts are semantically related
- If a word w_x occurred in several contexts along with w_y, then w_x and w_y are related by the first-order association. w_x and w_y are called first-order associates.
- For any pair of words, w_i and w_j, if the strength of similarity is stronger, then they have a large number of common first-order associates

Second Order Association

- If a word w_y occurred in several contexts along with w_z in which w_x is absent (or occurred in a statistically insignificant number of times), then w_x and w_z are related by the second-order association. w_x and w_z are called second-order associates.
  $\forall w_x, w_y, w_z \in V$, if we have $w_x R w_y$ and $w_y R w_z$, then $w_x R w_z$, treating the association relation R as transitive

MULTI-LEVEL WORD ASSOCIATIONS



SUMMARY

First-Order (Syntagmatic)

- **What:** Words that appear *together*
- **Finds:** Collocations, thematic links
- **Example:** **iphone** $\rightarrow$ apple
- **Model:** Direct word-context counts

Second-Order (Paradigmatic)

- **What:** Words that appear in *similar contexts*.
- **Finds:** Synonyms, analogies
- **Example:** **iphone** $\rightarrow$ microsoft
- **Model:** Similarity of word vectors

Predictive embeddings models (like Word2Vec, GloVe) are powerful as they are optimized to capture second-order (paradigmatic) relationships

SAMPLE CO-OCCURRENCE MATRIX

How much wood would a woodchuck chuck, if a woodchuck could chuck wood? As much wood as a woodchuck would,if a woodchuck could chuck wood.

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?	
a	0	5	9	6	1	10	4	8	18	9	10	0	0	80
as	5	4	2	1	0	0	7	10	3	2	1	0	5	40
chuck	9	2	0	8	0	0	1	9	11	2	4	3	3	52
could	6	1	8	0	0	4	0	6	8	0	2	2	2	39
how	1	0	0	0	0	4	3	0	0	2	0	0	0	10
if	10	0	5	4	0	0	4	3	10	3	0	0	0	39
much	4	7	1	0	4	0	0	10	2	3	0	0	3	34
wood	8	10	9	6	3	0	10	2	8	5	0	4	6	71
woodchuck	18	3	11	0	0	10	2	8	8	10	1	1	1	73
would	9	2	2	0	2	3	5	8	0	5	0	0	0	36
,	10	1	4	2	0	0	0	10	5	0	0	0	0	32
.	0	0	3	2	0	0	4	1	1	0	0	0	0	11
?	0	5	3	2	0	0	3	6	1	0	0	0	0	20
	80	40	57	31	10	31	43	81	75	41	18	10	20	537

NORMALIZATION PROCEDURES

Various normalization procedures[1]

$$\text{Row: } w'_{a,b} = \frac{w_{a,b}}{\sum_j w_{a,j}}$$

$$\text{Column: } w'_{a,b} = \frac{w_{a,b}}{\sum_i w_{i,b}}$$

$$\text{Length: } w'_{a,b} = \frac{w_{a,b}}{\sqrt{\sum_j w_{a,j}^2}}$$

where $w_{a,b}$ is the raw co-occurrence frequency

PEARSON CORRELATION-BASED NORMALIZATION I

$$w'_{a,b} = \frac{T w_{a,b} - \sum_j w_{a,j} \cdot \sum_i w_{i,b}}{\sqrt{\sum_j w_{a,j} \cdot \left(T - \sum_j w_{a,j}\right) \cdot \sum_i w_{i,b} \cdot (T - \sum_i w_{i,b})}}$$

$$T = \sum_i \sum_j w_{i,j}$$

PEARSON CORRELATION-BASED NORMALIZATION II

- $Tw_{a,b}$ – total sum T times raw co-occurrence at (a, b)
- $\sum_j w_{a,j} \cdot \sum_i w_{i,b}$ – expected value at (a, b) under independence
- Difference measures actual vs expected association at (a, b)
- $w'_{a,b}$ shows normalized association between a and b
- Values above 0 indicate positive relationship beyond expectation
- Values below 0 suggest negative association or lesser than expected co-occurrence
- $\sum_j w_{a,j}$ – total for row a
- $(T - \sum_j w_{a,j})$ – data excluding row a
- $(T - \sum_i w_{i,b})$ – data excluding column b
- Scales the numerator, similar to standard deviation terms

PEARSON CORRELATION-BASED NORMALIZATION III

- $w'_{a,b}$ shows normalized association between a and b
- Values above 0 - positive relationship beyond expectation
- Values below 0 - negative association or lesser than expected co-occurrence

PEARSON CORRELATION COEFFICIENT

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?
a	-0.175	-0.019	0.009	0.031	-0.019	0.121	-0.046	-0.059	0.103	0.057	0.213	-0.058	-0.082
as	-0.019	0.028	-0.052	-0.04	-0.039	-0.07	0.099	0.079	-0.053	-0.028	-0.013	-0.039	0.131
chuck	0.022	-0.045	-0.113	0.135	-0.045	-0.081	-0.073	0.02	0.068	-0.047	0.079	0.095	0.035
could	0.004	-0.052	0.09	-0.069	-0.039	0.054	-0.083	0.002	0.053	-0.08	0.028	0.068	0.021
how	-0.019	-0.039	-0.047	-0.034	-0.019	0.202	0.112	-0.058	-0.056	0.064	-0.026	-0.019	-0.027
if	0.084	-0.079	0.02	0.054	-0.039	-0.069	0.023	-0.058	0.094	0.001	-0.052	-0.039	-0.055
much	-0.023	0.13	-0.065	-0.064	0.19	-0.064	-0.077	0.104	-0.061	0.012	-0.048	-0.036	0.07
wood	-0.04	0.099	0.026	0.045	0.068	-0.097	0.087	-0.134	-0.03	-0.009	-0.073	0.109	0.097
woodchuck	0.109	-0.05	0.057	-0.098	-0.055	0.135	-0.077	-0.046	-0.034	0.091	-0.044	-0.014	-0.049
would	0.076	-0.019	-0.044	-0.066	0.073	0.029	0.058	0.053	-0.108	0.063	-0.05	-0.037	-0.053
,	0.116	-0.041	0.015	0.005	-0.035	-0.062	-0.074	0.114	0.012	-0.072	-0.047	-0.035	-0.05
.	-0.061	-0.041	0.078	0.077	-0.02	-0.036	0.151	-0.024	-0.02	-0.042	-0.027	-0.02	-0.028
?	-0.082	0.131	0.028	0.036	-0.027	-0.049	0.051	0.082	-0.051	-0.057	-0.037	-0.027	-0.039

$$r = \frac{T w_{a,b} - \sum_j w_{a,j} \sum_i w_{b,i}}{\sqrt{\sum_j w_{a,j} (T - \sum_j w_{a,j}) \sum_i w_{b,i} (T - \sum_i w_{b,i})}}$$

$$\text{where } T = \sum_i \sum_j w_{i,j} = 537$$

ENTROPY-BASED NORMALIZATION

$$\text{Entropy-based: } w'_{a,b} = \frac{\log(1 + w_{a,b})}{H_a}$$

$$H_a = -\sum_b \left(\frac{w_{a,b}}{\sum_j w_{a,j}} \right) \cdot \log \left(\frac{w_{a,b}}{\sum_j w_{a,j}} \right)$$

1. This correlation measures the distributional similarity between words across the entire vocabulary
2. **High Correlation** - Words tend to co-occur with similar set of words/context
3. Near zero - words are independent
4. Negative correlation - words tend to cooccur with different set of words

ENTROPY-BASED NORMALIZATION - INTERPRETATION

- The numerator applies log-scaling to dampen the effect of high counts
- The denominator (H_a) normalizes by distributional uniformity
- Words concentrated in few contexts (low entropy) are upweighted; those spread evenly (high entropy) are downweighted
- Converts raw association weights for a word into a proper probability distribution - independent of absolute counts or scale.
- Entropy of that distribution reflects how specific the words contextual signal is
  - **Low entropy:** Sharp, focused context (e.g., python $\rightarrow$ programming, language)
  - **High entropy:** Diffuse, broad usage (e.g., thing $\rightarrow$ object, idea, matter)
  - Low entropy $\rightarrow$ word meaning is distinct and discriminative
  - High entropy $\rightarrow$ word usage is generic and ambiguous
- Models often encourage lower entropy attention/association to focus on informative contexts

NORMALIZATION USING ENTROPY

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?
a	0.0	0.834	1.071	0.905	0.322	1.116	0.749	1.022	1.37	1.071	1.116	0.0	0.0
as	0.861	0.774	0.528	0.333	0.0	0.0	1.0	1.153	0.666	0.528	0.333	0.0	0.861
chuck	1.109	0.529	0.0	1.058	0.0	0.0	0.334	1.109	1.197	0.529	0.775	0.668	0.668
could	0.968	0.345	1.093	0.0	0.0	0.801	0.0	0.968	1.093	0.0	0.546	0.546	0.546
how	0.542	0.0	0.0	0.0	0.0	1.258	1.083	0.0	0.0	0.858	0.0	0.0	0.0
if	1.315	0.0	0.983	0.883	0.0	0.0	0.883	0.76	1.315	0.76	0.0	0.0	0.0
much	0.853	1.102	0.367	0.0	0.853	0.0	0.0	1.27	0.582	0.734	0.0	0.0	0.734
wood	0.953	1.04	0.998	0.844	0.601	0.0	1.04	0.476	0.953	0.777	0.0	0.698	0.844
woodchuck	1.425	0.671	1.203	0.0	0.0	1.161	0.532	1.064	1.064	1.161	0.336	0.336	0.336
would	1.201	0.573	0.573	0.0	0.573	0.723	0.934	1.146	0.0	0.934	0.0	0.0	0.0
,	1.539	0.445	1.033	0.705	0.0	0.0	0.0	1.539	1.15	0.0	0.0	0.0	0.0
.	0.0	0.0	0.944	0.748	0.0	0.0	1.096	0.472	0.472	0.0	0.0	0.0	0.0
?	0.0	1.081	0.837	0.663	0.0	0.0	0.837	1.174	0.418	0.0	0.0	0.0	0.0

POINTWISE MUTUAL INFORMATION

$$\mathrm{PMI}(w_i, w_c) = \log_2 \left(\frac{p(w_i, w_c)}{p(w_i)p(w_c)} \right)$$

The range of PMI is $(-\infty, \infty)$. Positive PMI means the words co-occur more often than expected by chance; zero means they are independent; negative means they co-occur less often than chance. Since negative values are unreliable to estimate from finite corpora, we usually keep only the positive part:

$$\mathrm{PPMI}(w_i, w_c) = \max \left(\log \left(\frac{p(w_i, w_c)}{p(w_i)p(w_c)} \right), 0 \right)$$

PPMI - NORMALIZED VALUES

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?
a	0.0	0.0	0.058	0.262	0.0	0.773	0.0	0.0	0.477	0.388	1.316	0.0	0.0
as	0.0	0.295	0.0	0.0	0.0	0.0	0.782	0.505	0.0	0.0	0.0	0.0	1.211
chuck	0.15	0.0	0.0	0.98	0.0	0.0	0.0	0.138	0.415	0.0	0.831	1.131	0.438
could	0.032	0.0	0.659	0.0	0.0	0.575	0.0	0.02	0.384	0.0	0.425	1.013	0.32
how	0.0	0.0	0.0	0.0	0.0	1.936	1.321	0.0	0.0	0.963	0.0	0.0	0.0
if	0.543	0.0	0.189	0.575	0.0	0.0	0.248	0.0	0.608	0.007	0.0	0.0	0.0
much	0.0	1.017	0.0	0.0	1.843	0.0	0.0	0.668	0.0	0.145	0.0	0.0	0.863
wood	0.0	0.637	0.177	0.381	0.819	0.0	0.565	0.0	0.0	0.0	0.0	1.107	0.819
woodchuck	0.504	0.0	0.35	0.0	0.0	0.864	0.0	0.0	0.0	0.585	0.0	0.0	0.0
would	0.518	0.0	0.0	0.0	1.093	0.367	0.551	0.387	0.0	0.598	0.0	0.0	0.0
,	0.741	0.0	0.164	0.079	0.0	0.0	0.0	0.728	0.112	0.0	0.0	0.0	0.0
.	0.0	0.0	0.944	1.147	0.0	0.0	1.513	0.0	0.0	0.0	0.0	0.0	0.0
?	0.0	1.211	0.346	0.549	0.0	0.0	0.628	0.688	0.0	0.0	0.0	0.0	0.0

BIAS IN NORMALIZATION

All normalization procedures have some bias. PMI scores are inflated for rare words. One way to reduce this bias toward low-frequency contexts is to smooth the context probability:

$$\text{PPMI}(w_i, w_c)_b = \max\left(\log\left(\frac{p(w_i, w_c)}{p(w_i)p_b(w_c)}\right), 0\right)$$

$$p_b(c) = \frac{\text{count}(c)^b}{\sum_c \text{count}(c)^b}$$

where $b \approx 0.75$

With $b \approx 0.75$, rare contexts get $p_b(c) > p(c)$, which lowers their PMI and reduces the rare-word bias (the same exponent word2vec uses for negative sampling)

Correlated Occurrence Analogue to Lexical Semantics[1] (COALS) [1]

[1] D. L. T. Rohde, L. M. Gonnerman, and D. C. Plaut. An improved model of semantic similarity based on lexical co-occurrence. Communications of the ACM, 8:627633, 2006.

COALS METHODOLOGY

- Gather co-occurrence counts, typically ignoring closed-class neighbors and using a ramped window of size 4.
- Discard all but the m (14,000, in this case) columns reflecting the most common open-class words.
- Convert counts to word pair correlations - Instead of using the raw frequency score, correlation score is used to analyze the relationship between pair of words
- The correlation coefficient values with this normalization will be in the range of [-1,1]
- Set negative values to 0.
- Take square root of positive values.
- The semantic similarity between two words is given by the correlation of their vectors.
- The matrix constructed using this correlation would be semantic space
- COALS method employs a normalization strategy that largely factors out lexical frequency.
- Columns representing low-frequency words are removed

COALS ALGORITHM

1. Gather co-occurrence counts, typically ignoring closed-class neighbors and using a ramped, size 4 window:

$$1\ 2\ 3\ 4\ \boxed{0}\ 4\ 3\ 2\ 1$$

2. Discard all but the m columns reflecting the most common open-class words
3. Convert counts to word pair correlations, set negative values to 0, and take square roots of the positive ones
4. The semantic similarity between two words is given by the correlation of their vectors

BUILDING CO-OCCURRENCE MATRIX - COALS STEP 1

How much wood would a woodchuck chuck, if a woodchuck could chuck wood? As much wood as a woodchuck would,if a woodchuck could chuck wood.

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?	
a	0	5	9	6	1	10	4	8	18	9	10	0	0	80
as	5	4	2	1	0	0	7	10	3	2	1	0	5	40
chuck	9	2	0	8	0	0	1	9	11	2	4	3	3	52
could	6	1	8	0	0	4	0	6	8	0	2	2	2	39
how	1	0	0	0	0	4	3	0	0	2	0	0	0	10
if	10	0	5	4	0	0	4	3	10	3	0	0	0	39
much	4	7	1	0	4	0	0	10	2	3	0	0	3	34
wood	8	10	9	6	3	0	10	2	8	5	0	4	6	71
woodchuck	18	3	11	0	0	10	2	8	8	10	1	1	1	73
would	9	2	2	0	2	3	5	8	0	5	0	0	0	36
,	10	1	4	2	0	0	0	10	5	0	0	0	0	32
.	0	0	3	2	0	0	4	1	1	0	0	0	0	11
?	0	5	3	2	0	0	3	6	1	0	0	0	0	20
	80	40	57	31	10	31	43	81	75	41	18	10	20	537

COALS-STEP 2

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?
a	-0.175	-0.019	0.009	0.031	-0.019	0.121	-0.046	-0.059	0.103	0.057	0.213	-0.058	-0.082
as	-0.019	0.028	-0.052	-0.04	-0.039	-0.07	0.099	0.079	-0.053	-0.028	-0.013	-0.039	0.131
chuck	0.022	-0.045	-0.113	0.135	-0.045	-0.081	-0.073	0.02	0.068	-0.047	0.079	0.095	0.035
could	0.004	-0.052	0.09	-0.069	-0.039	0.054	-0.083	0.002	0.053	-0.08	0.028	0.068	0.021
how	-0.019	-0.039	-0.047	-0.034	-0.019	0.202	0.112	-0.058	-0.056	0.064	-0.026	-0.019	-0.027
if	0.084	-0.079	0.02	0.054	-0.039	-0.069	0.023	-0.058	0.094	0.001	-0.052	-0.039	-0.055
much	-0.023	0.13	-0.065	-0.064	0.19	-0.064	-0.077	0.104	-0.061	0.012	-0.048	-0.036	0.07
wood	-0.04	0.099	0.026	0.045	0.068	-0.097	0.087	-0.134	-0.03	-0.009	-0.073	0.109	0.097
woodchuck	0.109	-0.05	0.057	-0.098	-0.055	0.135	-0.077	-0.046	-0.034	0.091	-0.044	-0.014	-0.049
would	0.076	-0.019	-0.044	-0.066	0.073	0.029	0.058	0.053	-0.108	0.063	-0.05	-0.037	-0.053
,	0.116	-0.041	0.015	0.005	-0.035	-0.062	-0.074	0.114	0.012	-0.072	-0.047	-0.035	-0.05
.	-0.061	-0.041	0.078	0.077	-0.02	-0.036	0.151	-0.024	-0.02	-0.042	-0.027	-0.02	-0.028
?	-0.082	0.131	0.028	0.036	-0.027	-0.049	0.051	0.082	-0.051	-0.057	-0.037	-0.027	-0.039

$$r = \frac{T w_{a,b} - \sum_j w_{a,j} \sum_i w_{b,i}}{\sqrt{\sum_j w_{a,j} (T - \sum_j w_{a,j}) \sum_i w_{b,i} (T - \sum_i w_{b,i})}}$$

$$\text{where } T = \sum_i \sum_j w_{i,j} = 537$$

COALS-STEP 3

	a	as	chuck	could	how	if	much	wood	woodchuck	would	,	.	?
a	0.0	0.0	0.009	0.031	0.0	0.121	0.0	0.0	0.103	0.057	0.213	0.0	0.0
as	0.0	0.028	0.0	0.0	0.0	0.0	0.099	0.079	0.0	0.0	0.0	0.0	0.131
chuck	0.022	0.0	0.0	0.135	0.0	0.0	0.0	0.02	0.068	0.0	0.079	0.095	0.035
could	0.004	0.0	0.09	0.0	0.0	0.054	0.0	0.002	0.053	0.0	0.028	0.068	0.021
how	0.0	0.0	0.0	0.0	0.0	0.202	0.112	0.0	0.0	0.064	0.0	0.0	0.0
if	0.084	0.0	0.02	0.054	0.0	0.0	0.023	0.0	0.094	0.001	0.0	0.0	0.0
much	0.0	0.13	0.0	0.0	0.19	0.0	0.0	0.104	0.0	0.012	0.0	0.0	0.07
wood	0.0	0.099	0.026	0.045	0.068	0.0	0.087	0.0	0.0	0.0	0.0	0.109	0.097
woodchuck	0.109	0.0	0.057	0.0	0.0	0.135	0.0	0.0	0.0	0.091	0.0	0.0	0.0
would	0.076	0.0	0.0	0.0	0.073	0.029	0.058	0.053	0.0	0.063	0.0	0.0	0.0
,	0.116	0.0	0.015	0.005	0.0	0.0	0.0	0.114	0.012	0.0	0.0	0.0	0.0
.	0.0	0.0	0.078	0.077	0.0	0.0	0.151	0.0	0.0	0.0	0.0	0.0	0.0
?	0.0	0.131	0.028	0.036	0.0	0.0	0.051	0.082	0.0	0.0	0.0	0.0	0.0

COALS RESULTS

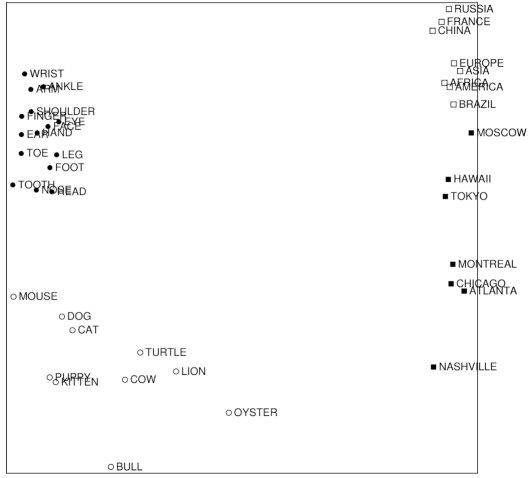
Nearest neighbors and their percent correlation similarities for a set of nouns

	gun	point	mind	monopoly
1)	46.4 handgun	32.4 points	33.5 minds	39.9 monopolies
2)	41.1 firearms	29.2 argument	24.9 consciousness	27.8 monopolistic
3)	41.0 firearm	25.4 question	23.2 thoughts	26.5 corporations
4)	35.3 handguns	22.3 arguments	22.4 senses	25.0 government
5)	35.0 guns	21.5 idea	22.2 subconscious	23.2 ownership
6)	32.7 pistol	20.1 assertion	20.8 thinking	22.2 property
7)	26.3 weapon	19.5 premise	20.6 perception	22.2 capitalism
8)	24.4 rifles	19.3 moot	20.4 emotions	21.8 capitalist
9)	24.2 shotgun	18.9 distinction	20.1 brain	21.6 authority
10)	23.6 weapons	18.7 statement	19.9 psyche	21.3 subsidies

COALS RESULTS - VERBS

	need	buy	play	change
1)	50.4 want	53.5 buying	63.5 playing	56.9 changing
2)	50.2 needed	52.5 sell	55.5 played	55.3 changes
3)	42.1 needing	49.1 bought	47.6 plays	48.9 changed
4)	41.2 needs	41.8 purchase	37.2 players	32.2 adjust
5)	41.1 can	40.3 purchased	35.4 player	30.2 affect
6)	39.5 able	39.7 selling	33.8 game	29.5 modify
7)	36.3 try	38.2 sells	32.3 games	28.3 different
8)	35.4 should	36.3 buys	29.0 listen	27.1 alter
9)	35.3 do	34.0 sale	26.8 playable	25.6 shift
10)	34.7 necessary	31.5 cheap	25.0 beat	25.1 altering

MULTIDIMENSIONAL SCALING



SOME INSIGHTS 1/2

- The majority of the correlations are negative
- Words with negative correlations contribute less to similarity than those with positive correlations
- Closed-class words (147 of them) convey more syntactic than semantic information. Punctuation marks and words such as *she*, *he*, *where*, *after* can be removed from the correlation table

SOME INSIGHTS 2/2

- **Positive Correlation** means that two words often appear together. In other words, their contexts are similar.
- **Zero correlation** means the pair of words are statistically independent. Hence no influence
  - CMI and Engineering_Drawing/blueprint have no inherent or direct relationship in terms or context.
- **Negative correlation** indicates an inverse relationship. For every word pair in this set, the second word in each pair is less likely to appear, and vice versa.
  - When talking about one of the specializations of CMI as the first word in a pair, some words, such as art, literature, philosophy, and religion, are less likely to appear together (although they may appear a few times).

SHOULD NEGATIVE CORRELATIONS BE DROPPED IN WORD EMBEDDINGS?

- Negative correlations show **contrasting or exclusive meaning** between words
- Keeping negatives provides **richer semantic information** for distinguishing opposites
- Some models drop negatives for **simplicity** or require **non-negative inputs**
- Removing negatives can reduce **noise** from unreliable negative values
- Popular embeddings often **capture contrasts implicitly** without explicit negative values
- Decision depends on **model design and goals**:
  - Keep negatives for **expressiveness**
  - Drop negatives for **interpretability or stability**

DENSE VECTORS

- Not sparse
- Shorter than sparse word vectors
- Real-valued and continuous
- Captures fine-grained semantic information
- Can be used to represent the entire sentences or paragraphs
- Word2Vec, GloVe, and FastText use dense embeddings
- Typical sizes are 50, 100, or 300 elements

SINGULAR VALUE DECOMPOSITION

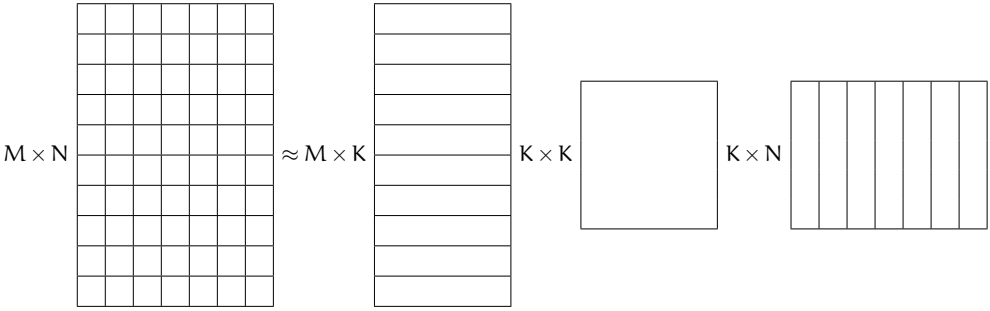
Singular value decomposition is a method to factorize a rectangular/square matrix into three matrices.

$$A = U\Sigma V^T \tag{1}$$

where A is an $M \times N$ matrix

- U is an $M \times K$ matrix
- Σ is a diagonal matrix of size $K \times K$
- V^T is a $K \times N$ matrix
- The columns of U are called the left-singular vectors; they form an orthonormal set
- The rows of V^T (columns of V) are called the right-singular vectors; they also form an orthonormal set
- Σ holds the singular values, arranged in descending order along the diagonal

SVD



SINGULAR VALUES

- Σ is a diagonal matrix with the singular values arranged in descending order
- The first dimension captures the most variance of the original term–document matrix
- Each subsequent dimension captures the next largest share of the remaining variance
- Singular values reflect the major associative patterns in the data, and ignore the smaller, less important influences

SUMMARY OF SVD

- Find a new set of dimensions or attributes that capture the variability of the data
- Identify the strongest pattern in the data
- Most variability is captured by a small fraction of the total set of dimensions
- Patterns among the terms are captured by the left-singular matrix
- Patterns among the documents are captured by the right-singular matrix
- The eigenvector associated with the largest eigenvalue indicates the direction of largest variance[2]
- Truncating to the top K singular values gives the best rank-K approximation of A. This is the basis of Latent Semantic Analysis (LSA)

[2] Pang-Ning Tan et al., "Introduction to Data Mining", 2007

SVD DEMO

NEED FOR BETTER MODELS

- HAL, COALS and their derivatives employ $\vec{v} = \langle A(f(t, b_1)), A(f(t, b_2)), ..., A(f(t, b_n)) \rangle$, where A is an association function
- Is it possible to generalize and *learn* semantic relationships and analogies, rather than count them?
- Understanding the structure of a sentence alone is not enough to capture its semantics
- This motivates prediction-based models (word2vec, GloVe) and, later, contextual models

SUMMARY OF WORD EMBEDDINGS

- Maps words to vectors of real numbers
- Learned from a corpus of text
- Capture the semantic meaning of words
- Used as input to many downstream applications, such as text classification, sentiment analysis, and machine translation, context embedding

SEMANTIC SPACE

- A semantic space model is method of assigning each word in a language to a point in a real finite dimensional vector space. Formally it is a quadruple $\mathbb{A}, \mathbb{B}, \mathbb{S}, \mathbb{M}$[2].
- $\mathbb{B}$ is the set $b_{1...D}$ of basis elements, the dimensionality of the space D. $\mathbb{B}$ can be a set of words.
- The dimensionality of the matrix is k where k will represent k most frequent words (minus the stop words) in a corpus.
- A word vector can be defined as $\vec{v} = \langle A(t, b_1), A(t, b_2), ..., A(t, b_D) \rangle$ where $\mathbb{A}$ is an association function. A is an identity matrix, if raw frequencies are used. t is the target word and b is the basis element[3].
- $\mathbb{S}$ is a similarity measure that maps pairs of vectors onto a continuous valued quantity that represents contextual similarity
- $\mathbb{M}$ is a transformation that takes one semantic space and maps it onto another, for example by reducing its dimensionality
- b and t may not be semantically related, but related distributionally through their lexical association

REFERENCES I

[1] Douglas LT Rohde, Laura M Gonnerman, and David C Plaut. "An improved model of semantic similarity based on lexical co-occurrence". In: *Communications of the ACM* 8.627-633 (2006), p. 116.

[2] Will Lowe. "Towards a Theory of Semantic Space". In: *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, 2001, pp. 576–581.

[3] Sebastian Padó and Mirella Lapata. "Dependency-Based Construction of Semantic Space Models". In: *Computational Linguistics* 33.2 (2007), pp. 161–199. DOI: 10.1162/coli.2007.33.2.161. URL: https://aclanthology.org/J07-2002.