things people have said about word embeddings

an illustrated guide

Fei-Tzin Lee

Columbia University

Spring 2019

Overview

So, what are word embeddings, anyway?

- **Distributed:** dense, low-dimensional representations of words
- Generally, though not always, derived from **distributional information** (i.e., word co-occurrences)

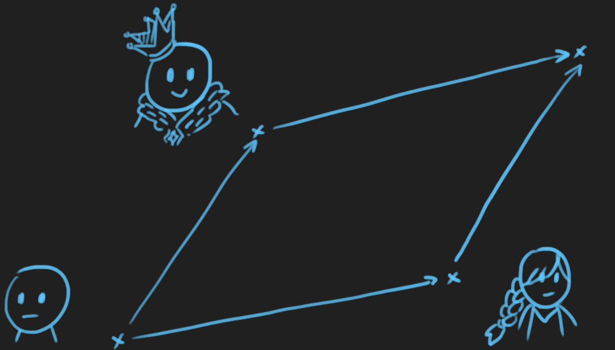
Overview

And why are they useful?

- More computationally efficient than one-hot vectors
- Word embeddings capture semantic information!

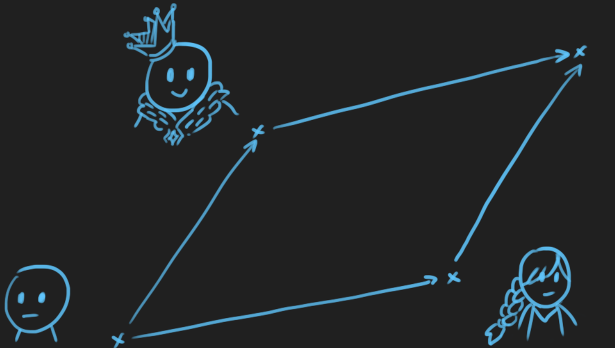
Linear analogies

man : king :: woman : ??



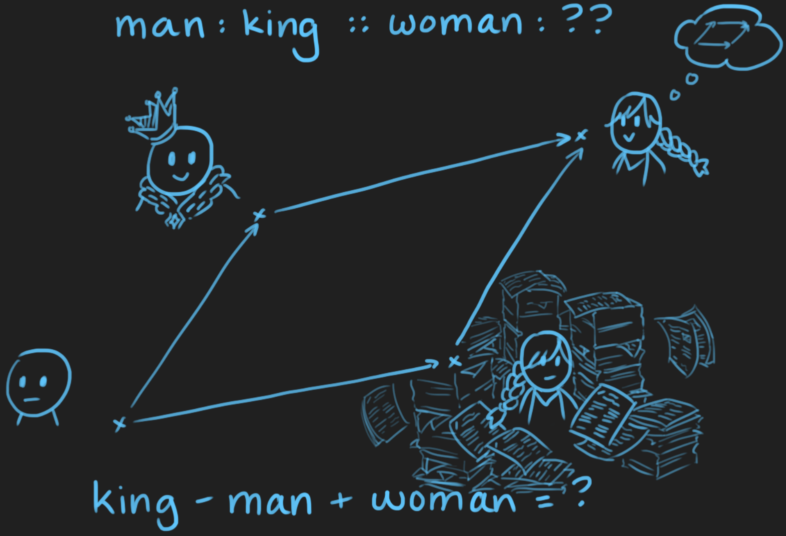
Linear analogies

man : king :: woman : ??



king - man + woman = ?

Linear analogies



This all sounds great; what *don't* we know?

Actually, a lot:

- What does the distribution of word embeddings in space look like?
- Why do such low-dimensional embeddings work so well?
- Why does the vector-addition analogy trick work?
- What do word embeddings even learn? Why do distributional word embeddings work at all? What are co-occurrences really telling us?

Recent work has attempted to address some of these, but many questions still remain.

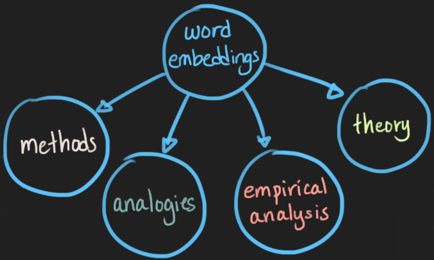
Problems

- Theoretical work is often based on questionable assumptions
- Effects of parameter, algorithm and data choices are not well-known
- Difficult to design new algorithms without fully understanding old ones!

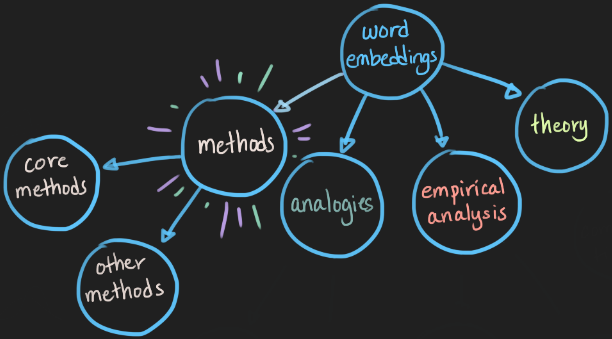
Outline

1. How do we embed words?
2. Linear analogies in embedding space
3. What can we observe about our embeddings?
4. Why are our embeddings like this?
5. Conclusions

Overview


word embeddings
methods
analogies
empirical analysis
theory

Overview

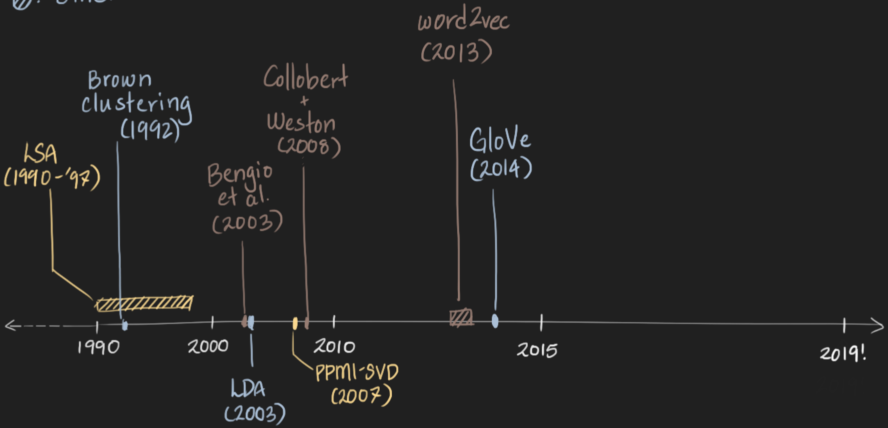

word embeddings
methods
theory
core methods
other methods
analogies
empirical analysis

A(n incomplete) timeline

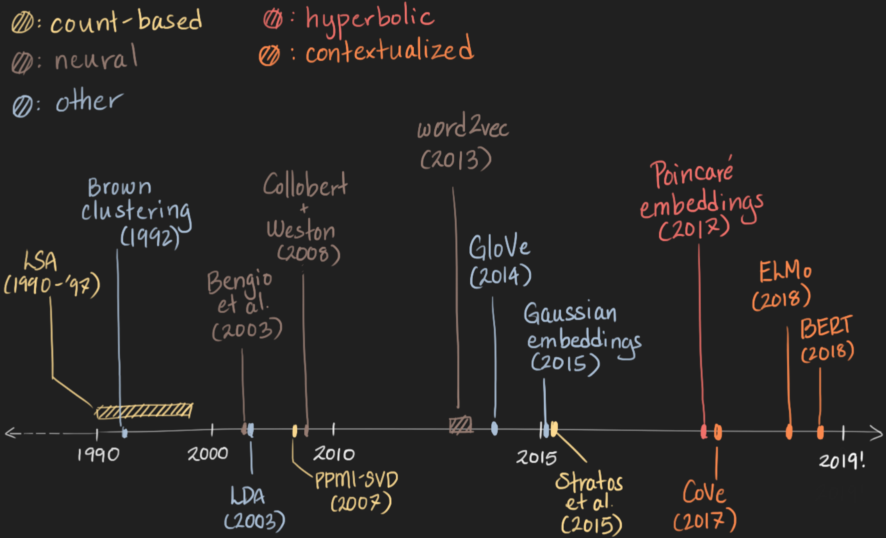
▨: count-based

▨: neural

▨: other



A(n incomplete) timeline



In this talk:

- Core embedding methods
    - Matrix-based methods
    - word2vec
    - GloVe
    - Contextualized embeddings: ELMo and BERT
- Expansions?
    - Atypical contexts (dependency-based embeddings)
    - Association-based embeddings
    - Distribution-based embeddings
    - Hyperbolic embeddings

Non-contextualized word embeddings

- **Basic problem:** given a corpus in which words co-occur with "contexts", find low-dimensional representations for words that encode information about the contexts they occur with.
- **How?**
  - Matrix methods: perform a decomposition of some version of the co-occurrence matrix
  - word2vec: maximize the probability of observed word-context co-occurrences in a sliding window over the corpus
  - GloVe: model the *ratios* of co-occurrence probabilities

Counting co-occurrences

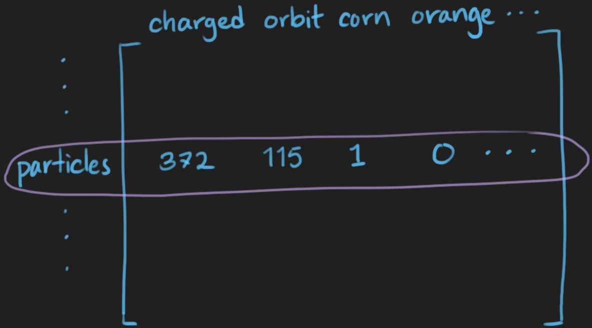
Humans are made of charged particles, known as corn,

Counting co-occurrences

Humans are made of charged particles, known as corn,

Counting co-occurrences

Humans are made of charged particles, known as corn,



Matrix-based methods

- **Given:** a co-occurrence matrix M; M_{ij} measures co-occurrence between word i and context j
- **Goal:** find word vectors w_i, context vectors c_j satisfying $\langle w_i, c_j \rangle = M_{ij}$
- **How?** Typically, use SVD: $M = U\Sigma V^T$; take $W = U\Sigma$, $C = V$ (for dimensionality reduction, truncate to top k singular values)
- LSA - contexts are documents
- PPMI-SVD (Bullinaria and Levy (2007)) - contexts are words
- More recently, Stratos et al. (2016) proposed using CCA rather than matrix factorization

word2vec

Extended from the original paper from earlier that year, Mikolov et al. (2013) presented *skip-gram with negative sampling* (SGNS) as an alternative to hierarchical softmax

- **Given:** a corpus of co-occurrences $D =$ a sequence of pairs (w_i, c_j)
- **Goal:** find w_i, c_j maximizing the log-likelihood of the corpus under the assumption $p(w_i, c_j) = \sigma(\langle w_i, c_j \rangle)$
- **But?** Trivial solution - set all word and context vectors equal. To address this, draw 'noise' context words from the unigram distribution* to use as *negative samples*

*Terms and conditions may apply.

GloVe

GloVe (2015) proposed using *global* co-occurrence information in a similar way, to preserve relative advantages of both matrix-based and predictive methods

- **Idea:** for words i, j and context word k, the ratio p_{ik}/p_{jk} tells us whether word k is more related to i or to j, or equally (un)related to both
- Use a global objective $J = \sum_{i,j=1}^{|V|} f(M_{ij})(w_i^T c_j + b_i + b_j' - \log M_{ij})^2$

word2vec and matrix factorization

Levy and Goldberg (2014) demonstrated that objectives for word2vec SGNS and matrix factorization are very similar:

- SGNS uses the global objective
$$J = \sum_{(w,c)\in D} \#(w,c)(\log \sigma(\langle w, c\rangle) + k\mathbb{E}_{c_N \sim P_D}[\log \sigma(\langle -w, c_N\rangle)])$$
- If all word-context pairs are independent, this reconstructs
$$[M_{ij}] : M_{ij} = \langle w_i, c_j\rangle = \log\left(\frac{\#(w_i, c_j)|D|}{\#(w_i)\#(c_j)}\right) - \log k$$

word2vec and GloVe

GloVe (2015) provided an additional link between word2vec and the GloVe objective:

- The (plain) skipgram objective minimizes weighted cross-entropy between modelled and empirical co-occurrence probabilities:
  $$J = -\sum_i \Big(\sum_k M_{ik}\Big) P_{ij} \log \left(\frac{\exp(\langle w_i, c_j \rangle)}{\sum_k \exp(\langle w_i, c_k \rangle)} \right)$$
- GloVe does the same but with squared error:
  $$J = \sum_{i,j} \Big(\sum_k M_{ik}\Big)(X_{ij} - \exp(\langle w_i, c_j \rangle))^2$$

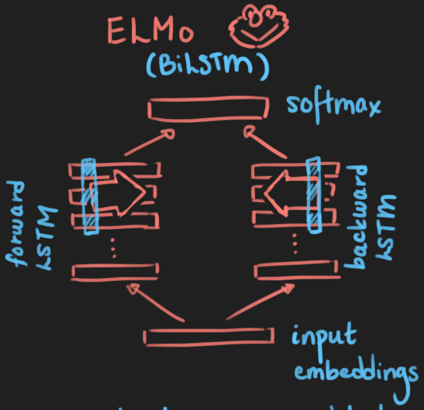
Non-contextualized embeddings - conclusion

Different approaches, but fairly similar, and all based on the same premise - we can represent a word well by encoding its co-occurrence information

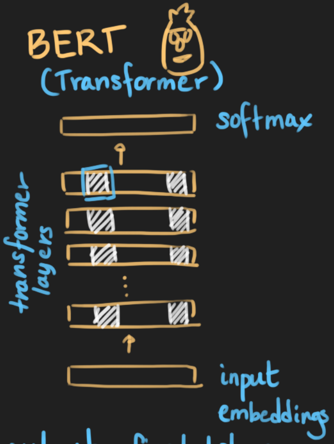
Contextualized word embeddings

- **Idea:** not every word has a single meaning applicable to every context!
- Contextualized word embeddings output a vector for each word conditioned on the context surrounding it
    - ELMo (2018): multilayer representations from a BiLSTM
    - BERT (2018): transformer-based representations

ELMo vs BERT



output is a weighted average over layers



output = final token embedding + segment + position

Does this mean word2vec is obsolete now?

Well... maybe not. ELMo and BERT aren't as well-understood; for analysis purposes, using the old methods may be easier.

Furthermore, non-contextualized methods seem like a better jumping-off point for exploring other assumptions:

- Does linear-context co-occurrence really tell us everything we want?
- Is a single point in Euclidean space really the best way to represent a word?

Moving towards more interpretable data

- Levy and Goldberg (2014) use dependency-parse context windows instead of linear context
    - Dependency-based embeddings capture functional rather than topical similarities

Moving towards more interpretable data

- Levy and Goldberg (2014) use dependency-parse context windows instead of linear context
    - Dependency-based embeddings capture functional rather than topical similarities
- De Deyne et al. (2016) propose embeddings trained directly from word association data rather than co-occurrence counts
    - Idea: corpus co-occurrences are a noisy signal of word associations anyway
    - Unsurprisingly, representations derived from associations perform better on similarity tasks

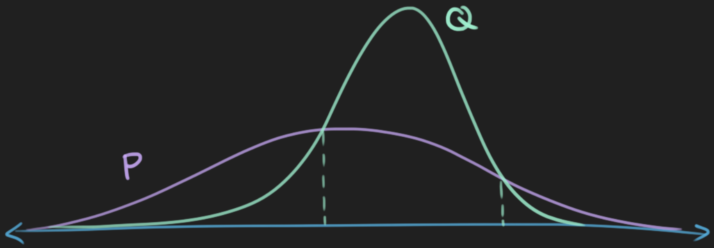
Distribution-based embeddings

Vilnis and McCallum (2015) propose distribution-based embeddings to capture uncertainty inherent in words, as well as to express asymmetric relations.

- Instead of maximizing “probability” of positive co-occurrence pairs, maximize a different kind of energy function
- Symmetric: “inner product” between Gaussians $(\int_{x \in \mathbb{R}^n} \mathcal{N}(x; \mu_i, \Sigma_i)\mathcal{N}(x; \mu_j, \Sigma_j) = \mathcal{N}(0; \mu_i - \mu_j, \Sigma_i + \Sigma_j))$
- Asymmetric: KL divergence (allows expression of entailment)

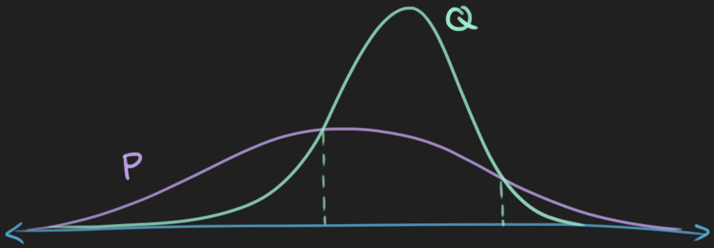
Distribution-based embeddings

$$D_{KL}(Q||P) = \int q(x) \log \frac{q(x)}{p(x)} dx$$



Distribution-based embeddings

$$D_{KL}(Q||P) = \int q(x) \log q(x) - q(x) \log p(x) \, dx$$



Hyperbolic space for hierarchical meaning

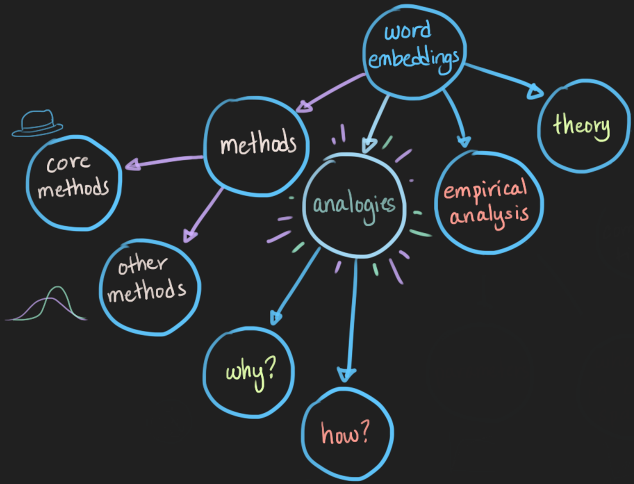
Euclidean space cannot embed hierarchies! Nickel and Kiela (2017) propose using *hyperbolic* space instead, which is capable of embedding arbitrary trees

- Evaluated on WordNet hyper/hyponym relation inference, performs drastically better than Euclidean embeddings

In conclusion?

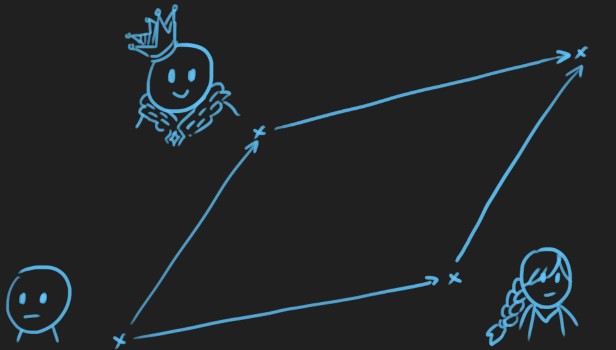
- There are some cool alternatives to plain old word embeddings!
- ...but people don't really use them.

Overview


word embeddings
methods
core methods
other methods
analogies
why?
how?
empirical analysis
theory

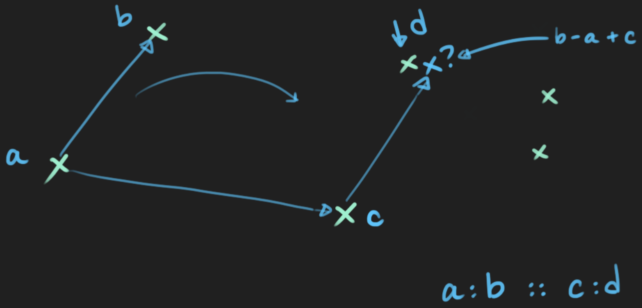
So, about those analogies...

man : king :: woman : ??

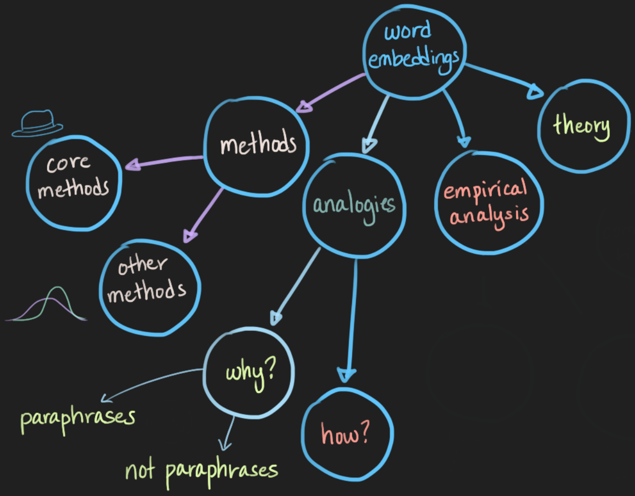


king - man + woman = ?

So, about those analogies...



Overview


word embeddings
methods
analogies
empirical analysis
theory
core methods
other methods
why?
how?
paraphrases
not paraphrases

Paraphrase-based composition

Gittens et al. (2017)

- **Idea:** a word c is a paraphrase for a set of words C if the probability distribution of co-occurrence with other words is identical
- If words obey a uniform distribution, linear composition holds

Allen and Hospedales (2019)

- Extend the previous notion to multiple sets of words
- Define *word transformations* characterized by adding words to a set

Other explanations

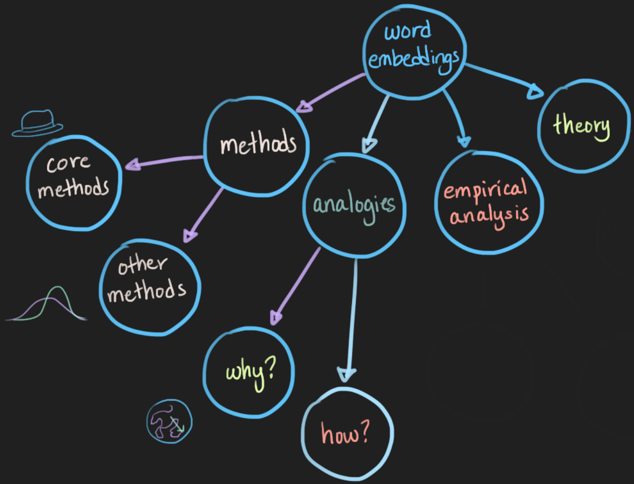
Arora et al. (2016)

- Under the assumptions of their model for language, linear analogies hold

Ethayarajh et al. (2018)

- Rather than making modeling assumptions, just look at PMI-based quantities directly
- Constant *co-occurrence shifted PMI* ($\mathrm{PMI}(x, y) + \log p(x, y)$) acts as a linear relation

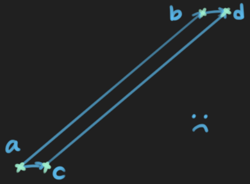
Overview


word embeddings
methods
theory
core methods
analogies
empirical analysis
other methods
why?
how?

...so why *doesn't* it work?

Turns out only some kinds of linear analogies can consistently be solved with "non-cheating" versions of the parallelogram trick :(

- Rogers et al. (2017) demonstrate that the effectiveness of the parallelogram trick depends on how similar the target is to the other words



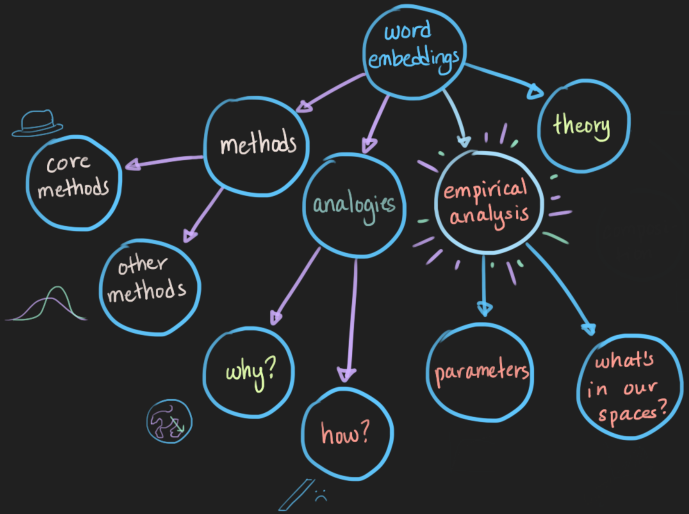
- Finley et al. (2017) propose a comparative baseline: take the nearest neighbor of each the two adjacent vertices ('b' and 'c') and pick the one closer to the true target

In conclusion...

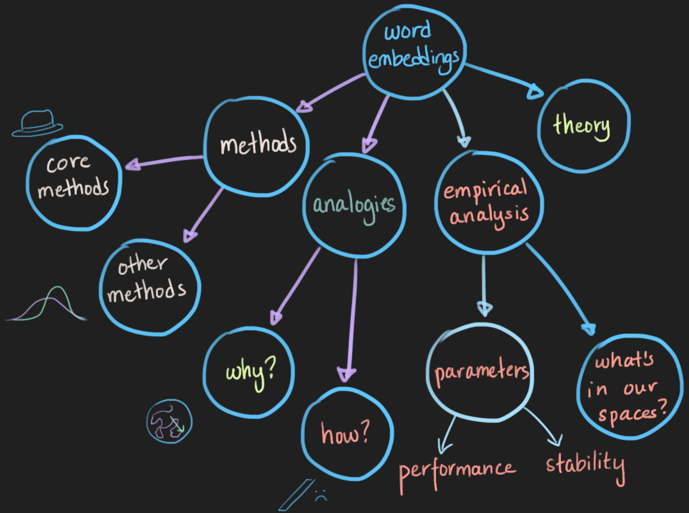
It's important not to jump the gun.

Before investigating an exciting phenomenon, make sure it actually works!

Overview


word embeddings
methods
analogies
empirical analysis
theory
core methods
other methods
why?
how?
parameters
what's in our spaces?

How does the training process affect the final result?



Hyperparameter effects on performance

Levy, Goldberg and Dagan (2015)

- Rebuttal to *Don't count, predict!*
- Changing the exponent of the singular value matrix closes the gap
- Negative samples help for SGNS but not other methods

Österlund et al. (2015)

- Raising Σ to an exponent normalizes principal components
- Removing the largest principal components helps!

Melamud et al. (2016)

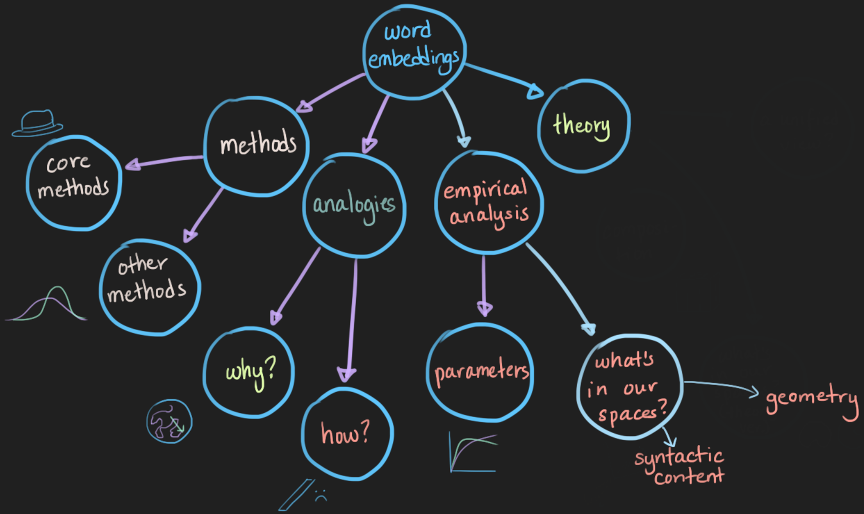
- Increasing dimensionality helps only up to a point
- ...but after that point, concatenating embeddings of different context types and windows helps further!

Hyperparameter effects on the entire space

Wendlandt et al. (2018)

- Word order and POS matter most
- Frequency doesn't matter very much!
- GloVe is the most stable overall

What do we actually get in our word embedding spaces?



Syntax and semantics

Andreas and Klein (2014)

- Embeddings contain redundant information with constituency parsers

Mitchell and Steedman (2015)

- Compare syntactic and semantic relations
- Conclusion: syntactic and semantic information are approximately orthogonal!

Hewitt and Manning (2019)

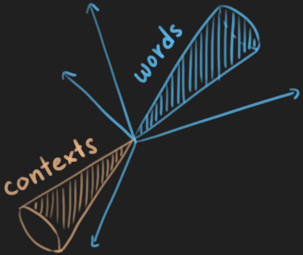
- Idea: can we experimentally identify syntactic information?
- Find a linear transformation such that embeddings in the resulting subspace represent positions in a parse tree*

*Remarkably, this is not a plothole.

The geometry of SGNS

Main point: SGNS word vectors end up in a narrow cone, diametrically opposite the narrow cone of context vectors!

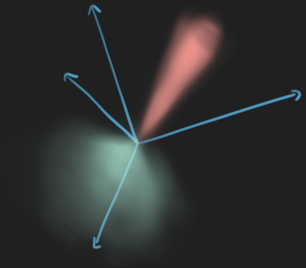
- Effect becomes more pronounced with more negative samples
- **Why?**



The geometry of SGNS

Main point: SGNS word vectors end up in a narrow cone, diametrically opposite the narrow cone of context vectors!

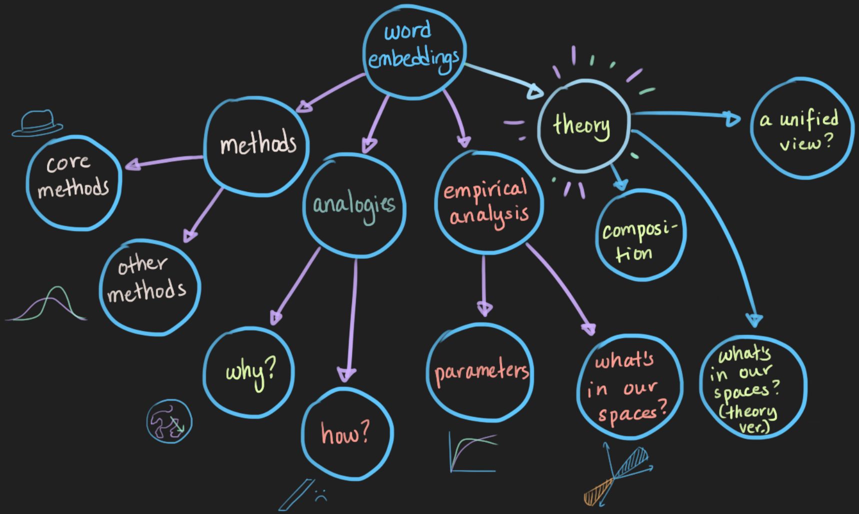
- Effect becomes more pronounced with more negative samples
- **Why?**



Conclusion?

- Algorithm choice is less significant than parameter tuning
- There is more to understand beyond downstream performance and linear analogies!

Overview


word embeddings
methods
core methods
other methods
analogies
why?
how?
empirical analysis
parameters
what's in our spaces?
theory
composi-tion
a unified view?
what's in our spaces? (theory ver.)

Beyond linear composition

- Gittens et al. (2017) do mention an alternate method of composition, but do not test it (actually, no empirical validation at all!)
- Frandsen and Ge (2019) propose using tensor-based composition of word embeddings, rather than linear
    - Tensor decomposition on three-way correlations gives a core tensor that yields corrections to additive composition

What's in our embeddings? (theory edition)

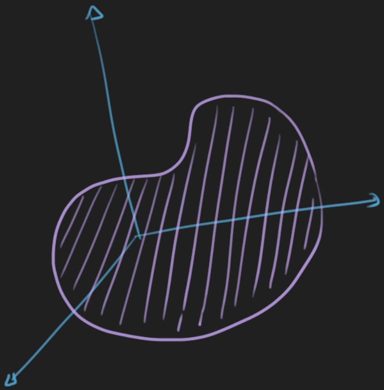
- Yin and Shen (2018) explain optimal dimension with the minimization of a unitary-invariant distance function between spaces
- Alvarez-Melis and Jaakkola (2018) approach the difference between spaces in a slightly different fashion
- Arora et al. (2018) suggest that observed embeddings for polysemous words are linear compositions of true embeddings for each of the word's senses

Dimension and distance

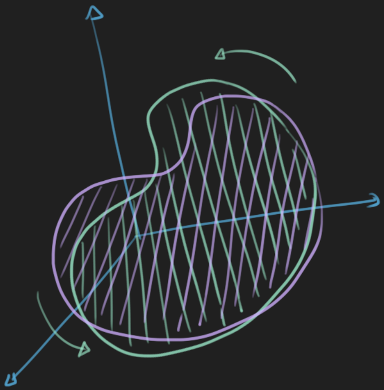
Yin and Shen (2018)

- Goal: predict the optimal dimension by minimizing distance to an ‘oracle’ embedding
- Metric: pairwise inner-product loss
- Perform bias-variance decomposition with noise assumptions to find minimal-distance embedding

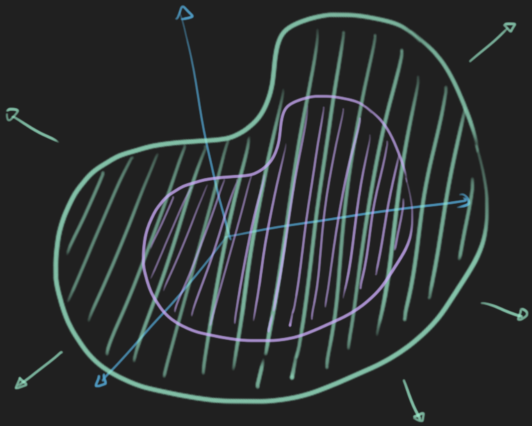
Distance between spaces



Distance between spaces



Distance between spaces



Dimension and distance

Yin and Shen (2018)

- Goal: predict the optimal dimension by minimizing distance to an ‘oracle’ embedding
- Metric: pairwise inner-product loss
- Perform bias-variance decomposition with noise assumptions to find minimal-distance embedding

Dimension and distance

Yin and Shen (2018)

- Goal: predict the optimal dimension by minimizing distance to an ‘oracle’ embedding
- Metric: pairwise inner-product loss
- Perform bias-variance decomposition with noise assumptions to find minimal-distance embedding

Alvarez-Melis and Jaakkola (2018)

- Optimal-transport based distances between spaces
- To avoid rotation and scale issues, use *distance between distances*

Polysemy in non-contextualized word embeddings

Arora et al. (2018)

- Under the same random walk model as Arora et al. (2016), they demonstrate that polysemous words will receive embeddings that are a linear combination of true sense embeddings
- True embeddings can be recovered with sparse coding!

Polysemy in non-contextualized word embeddings

Arora et al. (2018)

- Under the same random walk model as Arora et al. (2016), they demonstrate that polysemous words will receive embeddings that are a linear combination of true sense embeddings
- True embeddings can be recovered with sparse coding!

What are these embedding methods doing?

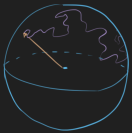
Arora et al. (2016)

- Under a particular model of language, word embedding methods will recover true hidden vectors for words

What are these embedding methods doing?

Arora et al. (2016)

- Under a particular model of language, word embedding methods will recover true hidden vectors for words
- A hidden topic vector performs a random walk over the unit sphere; at each step an observed word is drawn



What are these embedding methods doing?

Arora et al. (2016)

- Under a particular model of language, word embedding methods will recover true hidden vectors for words
- A hidden topic vector performs a random walk over the unit sphere; at each step an observed word is drawn

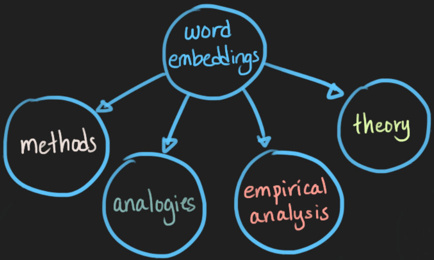
Hashimoto et al. (2016)

- Idea: word embeddings are actually doing manifold embedding over an underlying semantic metric space

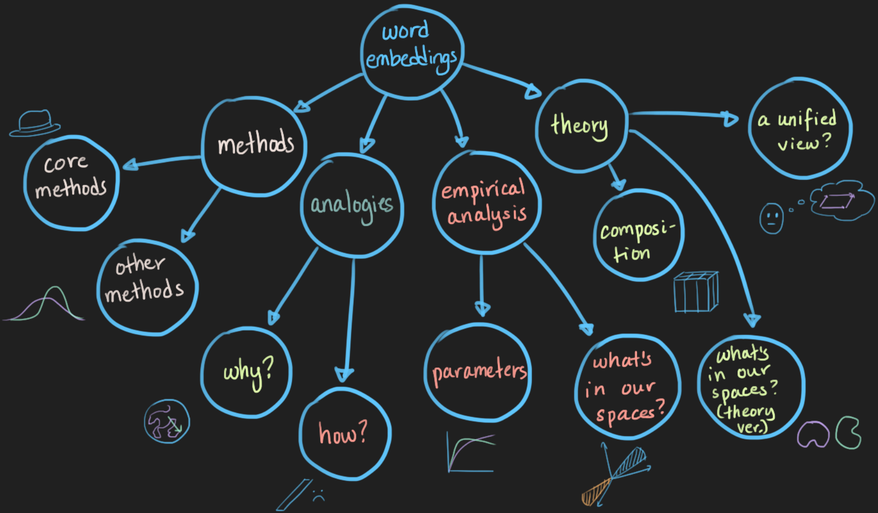
Conclusions?

- It seems like it’s somewhat difficult to match real behavior of language!
- Perhaps better not to make assumptions than to make unrealistic ones

Overview


word embeddings
methods
analogies
empirical analysis
theory

Overview


word embeddings
methods
analogies
empirical analysis
theory
a unified view?
core methods
other methods
composition
why?
how?
parameters
what's in our spaces?
what's in our spaces? (theory ver.)

Overall conclusions?

- Important for theoretical work to be grounded in existing empirical literature
- On the other hand, theory can motivate and explain experimental results
- There are still many questions remaining!
  - Are words *really* distributed on a semantic manifold?
  - What does dimension tell us? Does intrinsic dimension differ between languages?
  - Does embedding word-association data directly give us more interpretable dimensions?
  - Generally, what other implicit assumptions can we call into question?

Thanks!

:) Questions?

Back-links

- Overview
- Methods
    - Non-contextualized
    - Contextualized
    - Extensions
- Analogies
- Empirical analysis
- Theory
- Conclusions

Forward links

1. Data
2. Methods
3. Empirical analysis
4. Theory
5. Extra

Evaluation sets

Dataset	Type	Subcategories	Questions	
RG	similarity	n/a	65	19
synonym	n/a	80	1997	
Wordsim353	similarity	similarity and relatedness	353	
BLESS	concept-relation	co-hyponymy; hypernymy; meronymy; attribute and event	??	
MSR	analogies	syntactic	8,000	
Google	analogies	semantic and syntactic (morphological)	19,544	
BATS	analogies	semantic and morphological	99,200	

Algorithms

Paper	LSA	SVD	word2vec	GloVe	Other
Levy et al. (2015)		x	x	x	

Methods

1. word2vec
2. LSA
3. PPMI-SVD
4. GloVe
5. ELMo
6. BERT
7. Dependency-based embeddings
8. Association-based embeddings
9. Gaussian embeddings
10. Hyperbolic embeddings

word2vec(2013)

Original methods: skipgram and CBOW; both classification tasks

- Skip-gram: classify context words given a target word vector
- CBOW: classify target word given (averaged? summed?) context word vectors

word2vec (2013)

Skip-gram objective: maximize $\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$

How to define $p(w_{t+j}|w_t)$? Originally: use softmax

$$p(w|t) = \frac{\exp(c_w^T v_t)}{\sum_{w'=1}^{W}\exp(c_{w'}^T v_t)}$$

Problem: this is usually very expensive

SGNS: instead, use sigmoid + negative samples

$$\log\sigma(c_w^T v_t) + \sum_{i=1}^{k}\mathbb{E}_{w_i\sim P_n(w)}[\log\sigma(-c_{w_i}^T v_t)]$$

word2vec (2013)

- $J = \sum_{(w,c)\in D} \#(w,c)(\log\sigma(w,c) + k\mathbb{E}_{c_N \sim P_D}[\log\sigma(w,c_N)])$
- In practice, negative sample contexts c_N are drawn from an exponent of the empirical unigram distribution, $P_D(c) = \left(\frac{\#(c)}{\sum c'}\right)^{3/4}$

LSA

- **Given:** a word-document co-occurrence matrix M
- Perform rank-k SVD on M; decompose into $W = U\Sigma$, $C = V$

Neural Word Embedding as Implicit Matrix Factorization (Levy and Goldberg, 2014)

SGNS

- Goal: maximize probability of observed pair occurrence and negative pair non-occurrence
- $P(D = 1|w, c) = \sigma(w \cdot c) = \frac{1}{1 + e^{-w \cdot c}}$
- $P(D = 0|w, c) = 1 - P(D = 1|w, c)$
- For a single (w, c) pair, we have the objective

$$\log \sigma(w \cdot c) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-w \cdot c_N)]$$

Neural Word Embedding as Implicit Matrix Factorization (Levy and Goldberg, 2014)

If we have sufficient dimensionality that (w, c) pairs are independent, the above gives us the global objective

$$l = \sum_{(w,c)\in D} \log \sigma(w \cdot c) + k\mathbb{E}_{c_N \sim P_D}[\log \sigma(-w \cdot c_N)]$$

$$= \sum_{w\in V_w} \sum_{c\in V_c} \#(w, c)(\log \sigma(w \cdot c) + k\mathbb{E}_{c_N \sim P_D}[\log \sigma(-w \cdot c_N)]).$$

Writing out the expectation,

$$\mathbb{E}_{c_N \sim P_D}[\log \sigma(-w \cdot c_N)]) = \sum_{c_N\in V_c} \frac{\#(c_N)}{|D|} \log \sigma(-w \cdot c)$$

GloVe (2014)

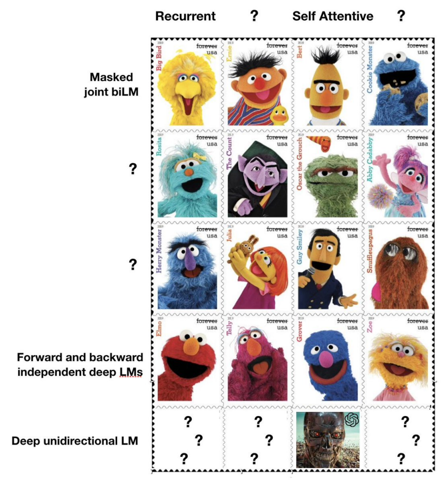
Idea: want to capture *ratios* of co-occurrence probabilities. Want a model $F(w_x, w_y, c_z) = \frac{P_{xz}}{P_{yz}}$ with a few nice properties:

1. Ideally, it should encode these ratios into vector offsets: $F(w_x, w_y, c_z) = G(w_x - w_y, c_z) = \frac{P_{xz}}{P_{yz}}$
2. To keep things simple, the arguments should interact only via dot product: $G(w_x - w_y, c_z) = H((w_x - w_y)^T c_z) = \frac{P_{xz}}{P_{yz}}$
3. Co-occurrence is symmetric, so we should be able to swap out word and context vectors, or word vectors for each other
4.

GloVe objective

- Final loss function: $J = \sum_{i,j=1}^{|V|} f(M_{ij})(w_i^T c_j + b_i + b_j' - logM_{ij})^2$

Contextualized word embeddings



Credit: Yoav Goldberg (`https://twitter.com/yoavgo/status/1106572683016368128`)

ELMo (2018)

- Bidirectional LSTM (separate forwards and backwards components)
- Learned embeddings are weighted averages of intermediate representations from every layer
- Higher level states encode context-dependent meaning (e.g. can be used for WSD) whereas lower-level states model syntax (e.g. can be used for POS tagging)

BERT (2018)

- Transformer architecture
- To avoid issues of peeking at words while still using bidirectional layers, randomly mask a certain percentage of words through all layers
- Next-sentence classification

Dependency-Based Word Embeddings (Levy and Goldberg, 2014)

Idea: replace linear context in SGNS with *dependency-based* context windows.

- Contexts take the form (word, label) for modifiers and (word, label^{-1}) for the head

Dependency-Based Word Embeddings (Levy and Goldberg, 2014)

Experiments

- Qualitative evaluation - nearest neighbors
- Quantitative evaluation - WordSim353; Chiarello et al. 1990. Goal: rank semantically similar (functionally similar) words above semantically related (topically similar) using cosine similarity.

Word Representations via Gaussian Embedding (Vilnis and McCallum, 2015)

Goal: learn parameters θ such that an 'energy function' $E_\theta(x, y)$ scores observed pairs x, y higher than unobserved.

- Training: Backpropagate under max-margin loss:

$$L_m(w, c_p, c_n) = max(0, m - E(w, c_p) + E(w, c_n))$$

- Two energy functions:
  - Symmetric: continuous inner product
  - Asymmetric: KL divergence

Word Representations via Gaussian Embedding (Vilnis and McCallum, 2015)

Evaluation

- Specificity/uncertainty (qualitative)
- Entailment
- Word similarity

Predicting human similarity judgments with distributional models: The value of word associations (De Deyne et al., 2016)

Corpus data

- Text corpora: OpenSubtitle (English, 1970-2016); Corpus of Contemporary English; Global Web-Based English corpus (British, American, Canadian and Australian subsets); SimpleWiki
- Preprocessing: lowercasing; stopwords and words occurring <300 times discarded
- Total: 65,632 unique word types after pruning; 2.16 billion tokens prior to pruning (doesn't say how many after)

Predicting human similarity judgments with distributional models: The value of word associations (De Deyne et al., 2016)

Association data

- Setup: each participant is given 15-20 cue words and asked to respond with three (ranked) associations for each cue
- $>$85,000 participants; 82% native speakers
- Total: 10,021 cue words with at least 300 responses

Predicting human similarity judgments with distributional models: The value of word associations (De Deyne et al., 2016)

Embeddings

- Corpus-based embeddings: raw PPMI vector? (unclear); CBOW; pretrained embeddings
- Association-based embeddings: 'spreading activation' over a random-walk graph

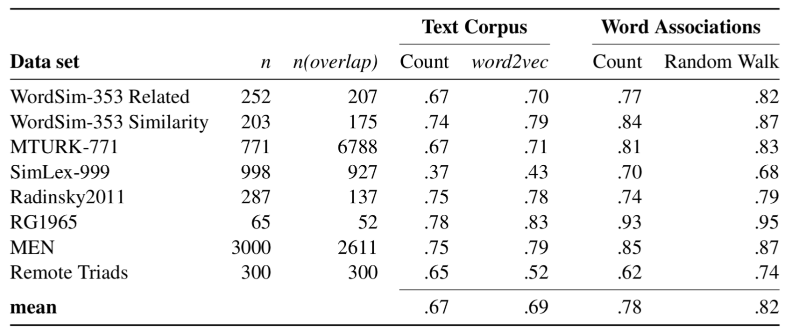
Predicting human similarity judgments with distributional models: The value of word associations (De Deyne et al., 2016)

Evaluation

- Similarity tasks: WordSim353 (similarity subset); SimLex999
- Relatedness tasks: WordSim353 (relatedness subset); MEN (Bruni et al., 2012); MTurk dataset (Radinsky et al., 2011); RG1965 (Rubenstein and Goodenough, 1965); MTURK-771 (Halawi et al., 2012)
- 'Remote triads', a novel task: pick the most related pair from a set of three nouns roughly the same in concreteness and frequency (100 triads evaluated by 40 native speakers)
- Association-based embeddings do better than distributional
- No extrinsic evaluations

Predicting human similarity judgments with distributional models: The value of word associations (De Deyne et al., 2016)

Results



			Text Corpus		**Word Associations**	
Data set	*n*	*n(overlap)*	Count	*word2vec*	Count	Random Walk
WordSim-353 Related	252	207	.67	.70	.77	.82
WordSim-353 Similarity	203	175	.74	.79	.84	.87
MTURK-771	771	6788	.67	.71	.81	.83
SimLex-999	998	927	.37	.43	.70	.68
Radinsky2011	287	137	.75	.78	.74	.79
RG1965	65	52	.78	.83	.93	.95
MEN	3000	2611	.75	.79	.85	.87
Remote Triads	300	300	.65	.52	.62	.74
mean			.67	.69	.78	.82

Poincaré Embeddings for Learning Continuous Hierarchical Representations (Nickel and Kiela, 2017)

Distance on the Poincaré ball: $\operatorname{arcosh}\left(1 + 2\frac{||u - v||^2}{(1 - ||u||^2)(1 - ||v||^2)}\right)$

Training

- Riemannian gradient descent: rescale Euclidean gradient to match hyperbolic distance; project new point back onto the manifold
- Loss function: depends on the problem, but roughly speaking, want to constrain similar words to be close in embedding space

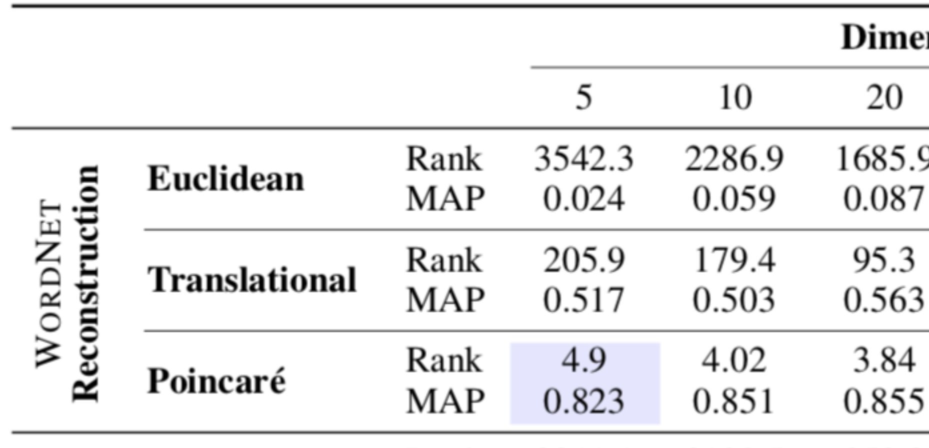
Poincaré Embeddings for Learning Continuous Hierarchical Representations (Nickel and Kiela, 2017)

Evaluation - WordNet

- Three types of embedding (Euclidean-Euclidean, Euclidean-translational, Poincaré); two tasks
- Reconstruction: embed (the closure of) the entire WordNet noun hierarchy, then reconstruct it from the embedding
- Link prediction: split into train, validation and test sets; use train set to learn embeddings and predict links on test set
- Loss function: $L = \sum_{(u,v)\in D} \log \frac{e^{-d(u,v)}}{\sum_{v'\in N(u)} e^{-d(u,v')}}$
- Evaluation: mean rank of observed (u, v) pair among negative observations (u, v')

Poincaré Embeddings for Learning Continuous Hierarchical Representations (Nickel and Kiela, 2017)

Evaluation - WordNet



			Dime		
			5	10	20
WORDNET Reconstruction	**Euclidean**	Rank	3542.3	2286.9	1685.9
		MAP	0.024	0.059	0.087
	Translational	Rank	205.9	179.4	95.3
		MAP	0.517	0.503	0.563
	Poincaré	Rank	4.9	4.02	3.84
		MAP	0.823	0.851	0.855

Poincaré Embeddings for Learning Continuous Hierarchical Representations (Nickel and Kiela, 2017)

Evaluation - HyperLex

Basically just use the WordNet-trained embeddings to predict graded entailment: score$(u, v) = -(1 + \alpha(||v|| - ||u||))d(u, v)$. Poincaré embeddings correlate much better with true ranking than all other methods evaluated.

Empirical analysis

1. Rogers et al.
2. Finley et al.
3. Improving distributional similarity
4. Caron p-transform
5. Context types and dimension
6. Geometry

Frame Title

What Analogies Reveal about Word Vectors and their Compositionality (Finley, Farmer and Pakhomov, 2017)

- Baseline: nearest neighbors of adjacent vertices + reciprocal rank
- Three types of stable relations: named entities; inflectional morphology; gender relations

Improving Word Embeddings with Lessons Learned from Distributional Similarity (Levy, Goldberg and Dagan, 2015)

Three broad classes of parameters

- Preprocessing
    - Dynamic context window
    - Subsampling frequent words
    - Rare word deletion
- Association metric
    - Shifted PMI (negative sampling)
    - Context distribution smoothinng
- Postprocessing
    - Adding context vectors
    - Eigenvalue weighting (e.g. $\sqrt{\Sigma}$)
    - Normalization

Improving Word Embeddings with Lessons Learned from Distributional Similarity (Levy, Goldberg and Dagan, 2015)

The Role of Context Types and Dimensionality in Learning Word Embeddings (Melamud, McClosky, Patwardhan and Bansal, 2016)

-

Factors Influencing the Surprising Instability of Word Embeddings (Wendlandt, Kummerfeld and Mihalcea, 2018)

Stability Percent overlap of a word's k nearest neighbors across two embedding spaces.

Corpora

- Five domains from NYT (U.S., New York and Region, Business, Arts, Sports)
- All five NYT domains combined (121k sentences, 24k word types)
- Europarl (2.3M sentences, 44k word types)

Factors Influencing the Surprising Instability of Word Embeddings (Wendlandt, Kummerfeld and Mihalcea, 2018)

Approach: use ridge regression to model influence of potential factors on stability

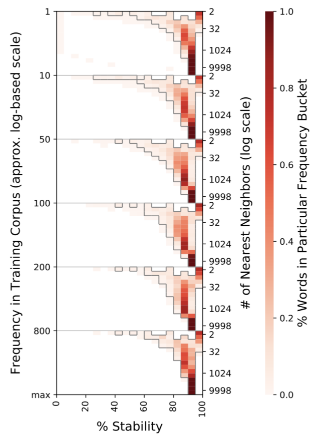
- Primary and secondary POS
- Number of syllables (zero if not present in dictionary)
- Higher raw frequency (between the two spaces); lower raw frequency; absolute difference in raw frequency
- Corpus vocabulary size; percent overlap between vocabularies; domain of each corpus; whether domains are the same
- First appearance in corpus A and in corpus B (as percent of number of sentences)
- Algorithm used (w2v, GloVe or PPMI-SVD); embedding dimension
- Frequency is minorly but not extremely predictive
- Higher stability correlates with slightly higher word similarity; when vectors are modified during training for POS tagging it seems to learn to compensate

Factors Influencing the Surprising Instability of Word Embeddings (Wendlandt, Kummerfeld and Mihalcea, 2018)

Results

- Most important: higher first appearance, lower first appearance
- Very important: POS (numerals, verbs and determiners most stable; punctuation, adpositions and particles least stable)
- Stability within domain is higher than across domain
- GloVe is much more stable than w2v or PPMI

Factors Influencing the Surprising Instability of Word Embeddings (Wendlandt, Kummerfeld and Mihalcea, 2018)



How much syntax do word embeddings encode?

Idea: want to determine whether word embeddings are useful for statistical constituency parsers

Test three ways in which word embeddings might help

-

A Structural Probe for Finding Syntax in Word Representations (Hewitt and Manning, 2019)

- Contextualized embeddings can do WSD
- Perhaps they represent syntax trees in some way?
- In fact, can learn a linear transformation taking the embeddings of a sentence to positions in a subspace that encode tree position via squared L_2.

The strange geometry of skip-gram with negative sampling (Mimno and Thompson, 2017)

Main results

- SGNS word vectors mainly point in the same direction
- SGNS context vectors form a noisy mirror of the word vectors
- The average inner product between word vectors and the mean word vector increases with more negative samples

The strange geometry of skip-gram with negative sampling (Mimno and Thompson, 2017)

Figure 1: SGNS word vectors and their context vectors projected using PCA (left) and t-SNE (right). t-SNE provides a more readable layout, but masks the divergence between word and context vectors.

The strange geometry of skip-gram with negative sampling (Mimno and Thompson, 2017)

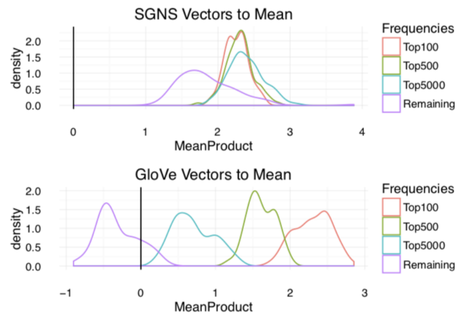
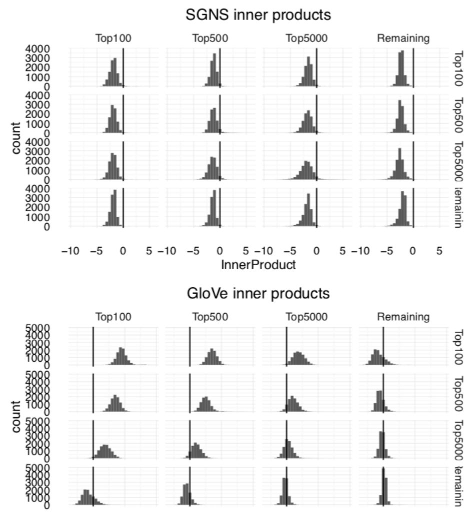


Figure 2: SGNS-trained vectors mostly point in the same direction, towards a mean vector $\hat{\boldsymbol{w}}$.

The strange geometry of skip-gram with negative sampling (Mimno and Thompson, 2017)



Theory

1. A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)
2. Gittens et al. (2017)
3. Ethayarajh et al. (2018)
4. Allen and Hospedales (2019)
5. Frandsen and Ge (2019)
6. Yin and Shen (2018)
7. Alvarez-Melis and Jaakkola (2018)
8. Arora et al. (2018)
9. Stratos et al. (2015)
10. Hashimoto et al. (2016)

Gittens et al. (2017)

Problem setup

- Idea: a set of context words $C = \{c_1, ..., c_m\}$ has the same meaning as a single word c if, for all other words w, $p(w|c_1, ..., c_m) = p(w|c)$.
- However, typically no word c will exactly satisfy this, so approximate the best paraphrase of C as $\text{argmin}_{c \in V} D_{KL}(p(\cdot|C)||p(\cdot|c))$
- Problems:
  - not clear how to define $p(\cdot|C)$
  - minimizing KL-divergence is difficult in general

Gittens et al. (2017)

Assumptions

1. $\forall c \exists Z_c \ni \forall w, p(w|c) = \frac{1}{Z_c} \exp(\langle u_c, v_w \rangle)$
2. $\forall C = \{c_1, ..., c_m\} \exists Z_c \ni \forall w, p(w|C) = \frac{1}{Z_C} p(w)^{1-m} \prod_{i=1}^{m} p(w|c_i)$

Understanding Composition of Word Embeddings via Tensor Decomposition (Frandsen and Ge, 2019)

Based on an extension of Arora et al.'s random-walk model, but with one modification: at every timestep, there is a chance to generate either a single word or a "syntactic word-pair".

On the Dimensionality of Word Embedding (Yin and Shen, 2018)

Problem: determine the optimal dimension for word embeddings from a theoretical perspective.

Approach: Define a "loss function" (distance metric) between embedding spaces; use the distance between an 'oracle embedding' E and actual learned embeddings $\hat{E}$ (on noisy data) at each dimension to determine the best choice

On the Dimensionality of Word Embedding (Yin and Shen, 2018)

Details

- When we use $W = U$, $C = V$, the PIP loss between E and $\hat{E}$ becomes $||PIP(E) - PIP(\hat{E})||^2 = d - k + 2||\hat{E}^T E^{\perp}||^2$

Gromov-Wasserstein Alignment of Word Embedding Spaces (Alvarez-Melis and Jaakkola, 2018)

Goal: find a mapping (‘alignment’) between word embedding spaces

- Use optimal transport mapping between distributions of pairwise distances in each space
- Overall distance between spaces is rotation- and scale-invariant

Linear Algebraic Structure of Word Senses, with Applications to Polysemy (Arora, Li, Liang, Ma and Risteski, 2018)

- Polysemous words have embeddings that are linear compositions of true sense embeddings
- Sense embeddings can be recovered via sparse coding

Word Embeddings from Decompositions of Count Matrices (Stratos et al., 2015)

A Latent Variable Approach to PMI (Arora et al., 2016)

Idea: specify a model for language generation; identify closed-form expressions for co-occurrence probabilities; use properties of model to analyze existing embedding algorithms.

A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

The random walk model

- Assumption: words and topics have fixed true (hidden) vectors
- Words are drawn i.i.d. from $s \cdot \hat{v}$, the product of the spherical Gaussian distribution with a random scalar (assume s has constant expectation τ and constant upper bound κ)
- A latent "discourse vector" performs a random walk over the unit sphere; a word vector v is drawn at each timestep with probability proportional to $e^{v \cdot c}$
- Assume the random walk has stationary distribution uniform over the unit sphere, and step size is bounded by $\epsilon_2/\sqrt{d}$

A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

Main results

- Under the model assumptions, there is some constant Z such that the probability that the 'partition function' $Z_c = \sum_w \exp(v_w \cdot c)$ diverges from Z by a factor of more than $1 \pm \epsilon_z$ is less than δ for appropriately defined ϵ_z, δ
- Using the previous, we can write

$$\log p(w, w') = ||v_w + v_{w'}||^2/2d - 2\log Z \pm \epsilon$$

$$\log p(w) = ||v_w||^2/2d - \log Z \pm \epsilon$$

$$PMI(w, w') = \langle v_w, v_{w'} \rangle/d \pm O(\epsilon)$$

for window size 2; the above are shifted by $\gamma = \log(\frac{q(q-1)}{2})$ for general window size q.

A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

Training objective

If we assume that co-occurrence probability is approximately distributed according to a multinomial distribution, then MLE word vectors optimize

$$\min_{\{v_w\},C} \sum_{w,w'} X_{w,w'}(\log(X_{w,w'}) - ||v_w + v_{w'}||^2 - C)^2$$

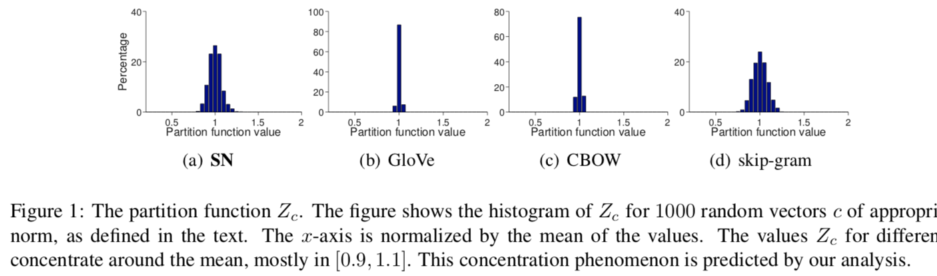
A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

The random walk model

- Assumption: words and topics have fixed true (hidden) vectors
- A latent "discourse vector" performs a random walk over the unit sphere; a word vector is drawn at each timestep

A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

Experimental validation



(a) **SN** (b) GloVe (c) CBOW (d) skip-gram

Figure 1: The partition function Z_c. The figure shows the histogram of Z_c for 1000 random vectors c of appropri norm, as defined in the text. The x-axis is normalized by the mean of the values. The values Z_c for differen concentrate around the mean, mostly in $[0.9, 1.1]$. This concentration phenomenon is predicted by our analysis.

A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

Experimental validation?

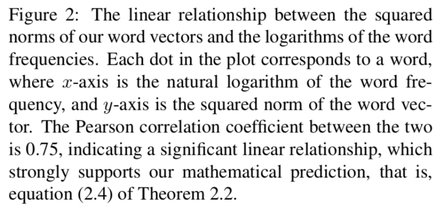


Figure 2: The linear relationship between the squared norms of our word vectors and the logarithms of the word frequencies. Each dot in the plot corresponds to a word, where x-axis is the natural logarithm of the word frequency, and y-axis is the squared norm of the word vector. The Pearson correlation coefficient between the two is 0.75, indicating a significant linear relationship, which strongly supports our mathematical prediction, that is, equation (2.4) of Theorem 2.2.

Evaluation :(

A Latent Variable Model for PMI-based Word Embeddings (Arora et al., 2016)

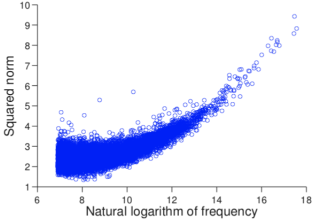


Figure 2: The linear relationship between the squared norms of our word vectors and the logarithms of the word frequencies. Each dot in the plot corresponds to a word, where x-axis is the natural logarithm of the word frequency, and y-axis is the squared norm of the word vector. The Pearson correlation coefficient between the two is 0.75, indicating a significant linear relationship, which strongly supports our mathematical prediction, that is, equation (2.4) of Theorem 2.2.

Word Embeddings as Metric Recovery in Semantic Spaces (Hashimoto et al., 2016)

Extra

TODO: add LSA (1990); Bullinaria and Levy (2007)