Word Embeddings

Jim Glass / MIT 6.806-6.864 / Spring 2021

1

Review: Distributional Semantics

- Word vector representations capture the "distributional hypothesis"
  - Words that occur in similar contexts tend to have the same meaning
- Words and contexts
  - Count how often word i appears in context j
  - This results in a very large matrix of size $|V| * |C|$

	Context 1	Context 2	Context 3	…
table	1	1	0	
chair	1	0	0	
dream	0	0	1	
coffee	0	1	0	

- Vector Space Models such as latent semantic analysis (LSA) factorize this matrix with singular value decomposition (SVD)

2

The Neural Word Embeddings Story

- Laying the groundwork:
  - Learning representations by back-propagation (Rumelhart et al., 1986)
  - A neural probabilistic language model (Bengio et al., 2003)
  - NLP (almost) from Scratch (Collobert et al., 2011)
- The rise of neural word embeddings:
  - WORD2VEC (Mikolov et al., 2013)
  - GloVe (Pennington et al., 2014)
  - FastText (Bojanowski et al., 2017)

3

n-gram Language Models

- For $W = \{w_1, \cdots, w_K\}$, n-gram LMs use chain rule to predict $p(W)$

$$p(W) = \prod_{i=1}^{K} p(w_i \,|w_1, \ldots, w_{i-1}) = \prod_{i=1}^{K} p(\, w_i | \phi(w_i))$$

  - where $\phi(w_i) = \{w_1, \ldots, w_{i-1}\}$ is the history for w_i
- In n-gram models, the previous $n - 1$ words are used to represent the history: $\phi(w_i) = \{w_{i-(n-1)}, \ldots, w_{i-1}\}$
- Estimates are based on counts in training data, e.g., trigram:

$$P(w_i | w_{i-2} w_{i-1}) \approx f(w_i | w_{i-2} w_{i-1}) = \frac{c(w_{i-2} w_{i-1} w_i)}{c(w_{i-2} w_{i-1})}$$

- Smoothing and discounting used for zero counts in training data

4

Quantifying LM Performance

- One LM is often considered better than another if predicts an N word *test* corpus $\mathcal{W}$ with a higher probability $\hat{p}(\mathcal{W})$
- Comparisons are usually based on *negative log likelihood*

$$NLL = -\frac{1}{N}\log\hat{p}\,(\mathcal{W}) = -\frac{1}{N}\sum_i \log\hat{p}\big(w_i|\phi(w_i)\big)$$

- For large N, NLL is a measure of language uncertainty (entropy)
- A more intuitive measure of complexity is the *perplexity*

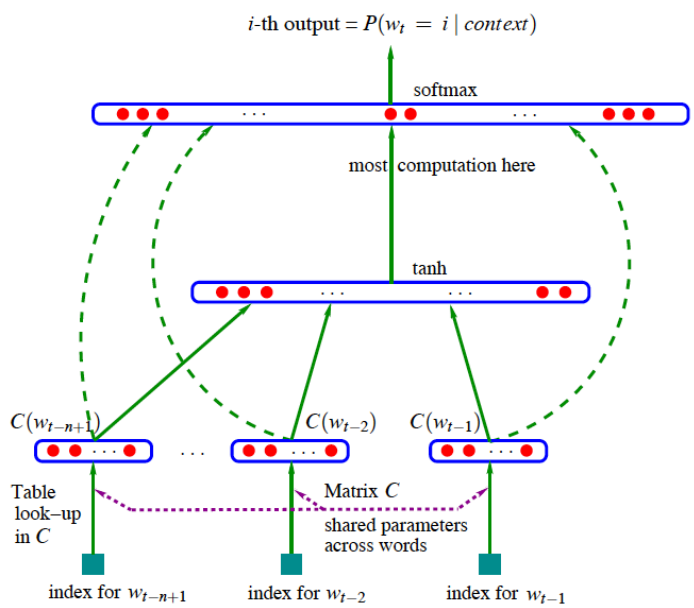
$$PPL = e^{NLL}$$

- PPL is often interpreted as an average branching factor
  - e.g., a uniform LM will have PPL equal to vocabulary size

5

A Neural Probabilistic Language Model

Yoshua Bengio
Réjean Ducharme
Pascal Vincent
Christian Jauvin
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal, Montréal, Québec, Canada



Journal of Machine Learning Research 3 (2003) 1137–1155

6

Neural Language Models

- Motivated by shortcomings of classic count-based n-grams
- Maximize corpus likelihood by estimating next word probability

$$p\big(w_i|\phi(w_i)\big) = softmax\,(\boldsymbol{y})_i = \frac{e^{y_{w_i}}}{\sum_{j=1}^{V} e^{y_j}}$$

  where y_i is the pre-softmax network output for word w_i
- <u>Represent words as low-dimensional distributed vectors!</u>
- Neural network parameters are learned on a training corpus
  - Use cross-entropy loss, SGD and back-propagation

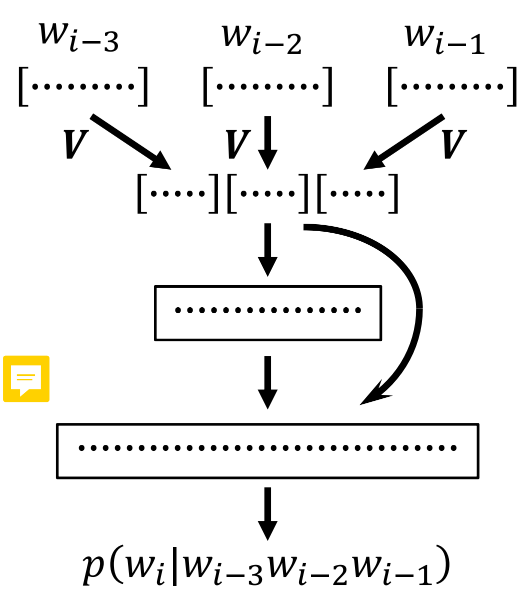
$$L(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T} \log p\big(w_t|\phi(w_t)\big)$$

7

An Early Neural n-gram (Bengio et al., 2003)

- Associate a <u>distributed vector per word</u>
- Express the joint probability function of word sequences in terms of the vectors
- Simultaneously learn word vectors and parameters of the probability function

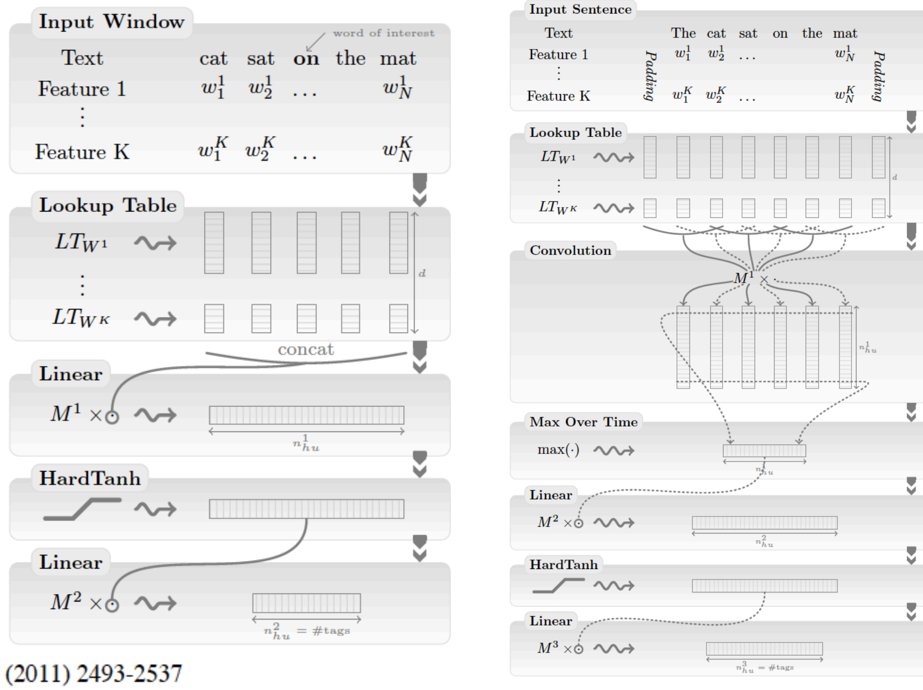
- Implemented as feed-forward network
- Shared vector mapping, $\boldsymbol{V}$, for all words
- First layer concatenated context vectors
- Perplexity improvements on Brown and AP News corpora over best n-grams



8

Natural Language Processing (Almost) from Scratch

Ronan Collobert*
Jason Weston†
Léon Bottou‡
Michael Karlen
Koray Kavukcuoglu§
Pavel Kuksa¶
NEC Laboratories America
4 Independence Way
Princeton, NJ 08540



Journal of Machine Learning Research 12 (2011) 2493-2537

9

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

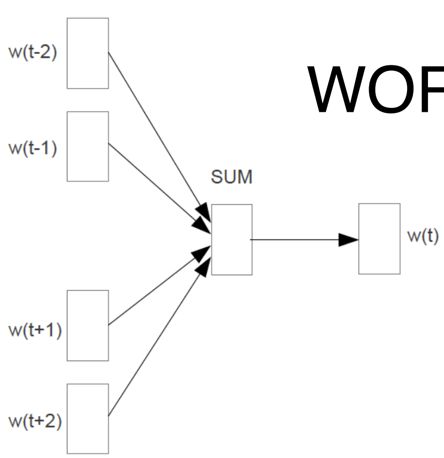
Distributed Representations of Words and Phrases and their Compositionality

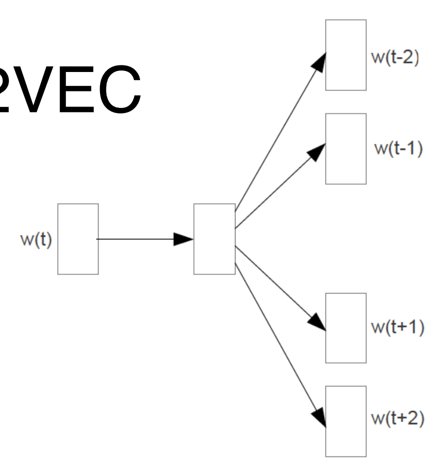
Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

WORD2VEC





ICLR 2013 | **CBOW** | **Skip-gram** | NeurIPS 2013

10

Neural Word Embeddings: WORD2VEC

- A neural framework for learning vector representations of words
- Based on predicting neighboring words in a local context
- Probability based on similarity of input and output word vectors
- Vector values learned by maximizing likelihoods of a text corpus

$p(w_{t-2}|w_t)$ $p(w_{t-1}|w_t)$ $p(w_{t+1}|w_t)$ $p(w_{t+2}|w_t)$

The quick brown fox jumps over the lazy dog

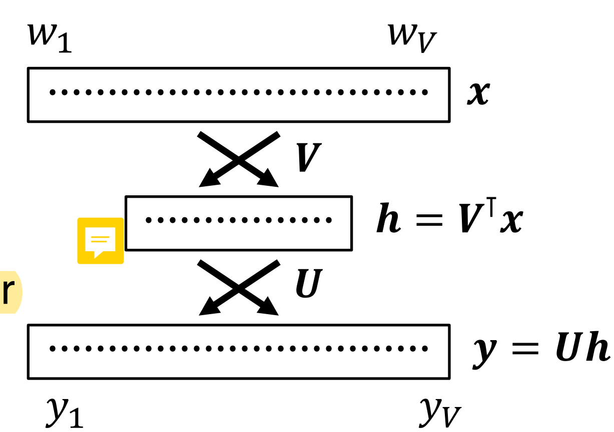
Context window (size 2)

11

WORD2VEC Concepts

- Contextual (input) word, w_c, is represented by vector, $\boldsymbol{v}_{w_c}$
- Predicted (output) word, w_i, is represented by vector, $\boldsymbol{u}_{w_i}$
- Prediction probability $p(w_i|w_c)$ based on dot product $\boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c}$

$$p(w_i|w_c) = softmax\,(\boldsymbol{y})_i = \frac{e^{(\boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c})}}{\sum_{j=1}^{V} e^{(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_c})}}$$

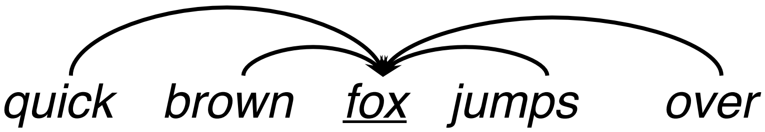


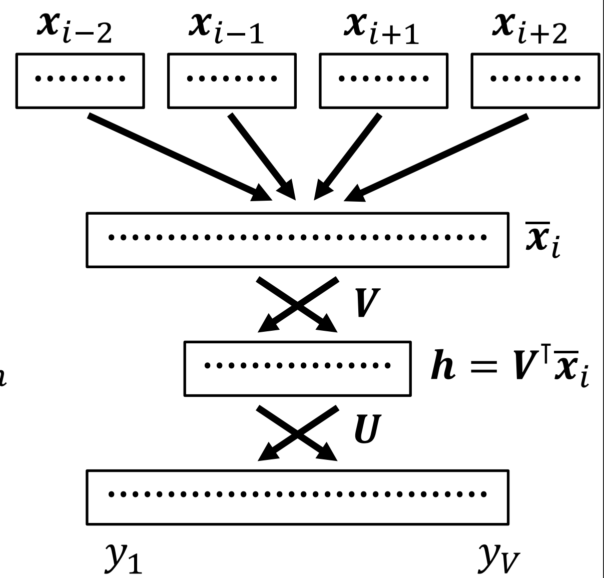
- Words encoded as "one-hot" vector
- No internal non-linearity!

12

CBOW Formulation

- *Continuous Bag-Of-Words* predicts center word from neighbors





- Context is represented by average of neighboring word vectors

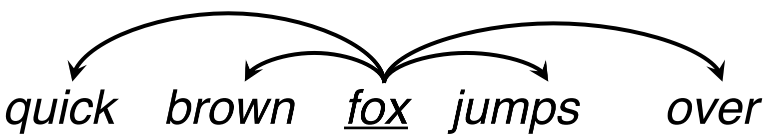
$$\bar{v}_c = \frac{1}{2h} \sum_{\substack{-h \le n \le h \\ n \ne 0}} v_{w_{c+n}} = \frac{V^\top}{2h} \sum_{\substack{-h \le n \le h \\ n \ne 0}} x_{c+n}$$

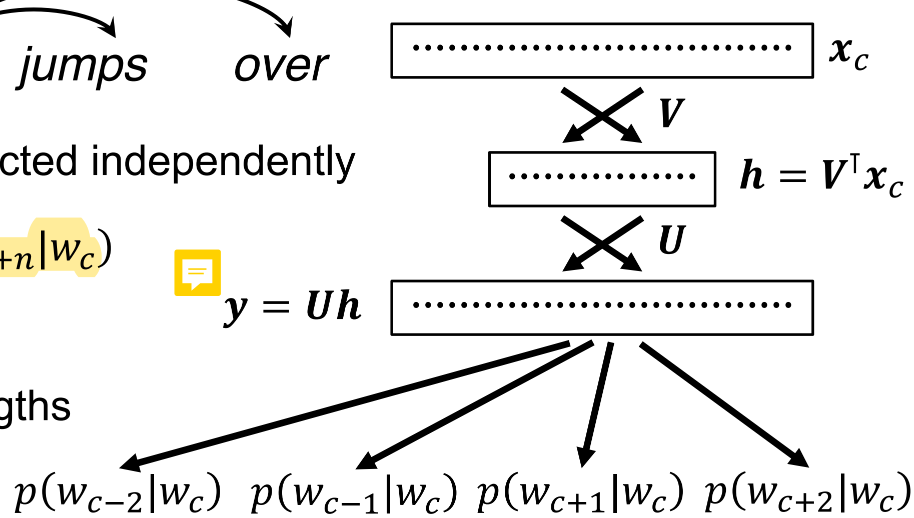
$$= V^\top \bar{x}_c$$

13

Skip-gram Formulation

- *Skip-gram* predicts neighbor words from center word





- Each output is predicted independently

$$\prod_{\substack{-h \le n \le h \\ n \ne 0}} p(w_{c+n}|w_c)$$

- Context window lengths can be sampled

14

WORD2VEC Training

- Training based on a large training corpus $\{w_1, w_2, w_3, \ldots, w_T\}$
- Objective function based on cross-entropy loss

$$L_{CB}(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T} \log p(w_t | w_{t-h}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+h}) = -\frac{1}{T}\sum_{t=1}^{T} \log \frac{e^{(\boldsymbol{u}_{w_t} \cdot \bar{\boldsymbol{v}}_t)}}{\sum_{j=1}^{V} e^{(\boldsymbol{u}_j \cdot \bar{\boldsymbol{v}}_t)}}$$

$$L_{SG}(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T} \sum_{\substack{-h \le n \le h \\ n \neq 0}} \log p(w_{t+n} | w_t) = -\frac{1}{T}\sum_{t=1}^{T} \sum_{\substack{-h \le n \le h \\ n \neq 0}} \log \frac{e^{(\boldsymbol{u}_{w_{t+n}} \cdot \boldsymbol{v}_{w_t})}}{\sum_{j=1}^{V} e^{(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_t})}}$$

- For SGD, compute gradient of loss function, $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$, i.e.,

$$\frac{\partial}{\partial \boldsymbol{v}_k} \mathrm{L}(\boldsymbol{\theta}) \qquad \frac{\partial}{\partial \boldsymbol{u}_l} \mathrm{L}(\boldsymbol{\theta})$$

15

Example Gradient Calculation

$$\frac{\partial}{\partial \boldsymbol{v}_{w_c}} \log p(w_i | w_c) = \frac{\partial}{\partial \boldsymbol{v}_{w_c}} \log \frac{e^{(\boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c})}}{\sum_{j=1}^{V} e^{(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_c})}}$$

$$= \frac{\partial}{\partial \boldsymbol{v}_{w_c}} \boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c} - \frac{\partial}{\partial \boldsymbol{v}_{w_c}} \log \sum_{j=1}^{V} e^{(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_c})}$$

$$= \boldsymbol{u}_{w_i} - \frac{1}{\sum_{k=1}^{V} e^{(\boldsymbol{u}_k \cdot \boldsymbol{v}_{w_c})}} \sum_{j=1}^{V} \frac{\partial}{\partial \boldsymbol{v}_{w_c}} e^{(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_c})}$$

$$= \boldsymbol{u}_{w_i} - \frac{1}{\sum_{k=1}^{V} e^{(\boldsymbol{u}_k \cdot \boldsymbol{v}_{w_c})}} \sum_{j=1}^{V} e^{(\boldsymbol{u}_j \cdot \boldsymbol{v}_{w_c})} \frac{\partial}{\partial \boldsymbol{v}_{w_c}} \boldsymbol{u}_j \cdot \boldsymbol{v}_{w_c}$$

$$= \boldsymbol{u}_{w_i} - \sum_{j=1}^{V} p(w_j | w_c)\, \boldsymbol{u}_{w_j}$$

SGD attempts to move $\boldsymbol{v}_{w_c}$ and $\boldsymbol{u}_{w_i}$ towards each other

16

Computational Issues

- The *softmax* operation is computationally expensive
  - Hierarchical softmax & negative sampling are more efficient
  - Subsampling frequent words is also effective
- Negative Sampling:
  - Replace *softmax* with logistic function $\sigma\left(\boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c}\right)$
  - For each word pair, randomly select a set of negative samples $\mathcal{W}_{NS}$
  - Maximize likelihood that correct output appears & minimize incorrect

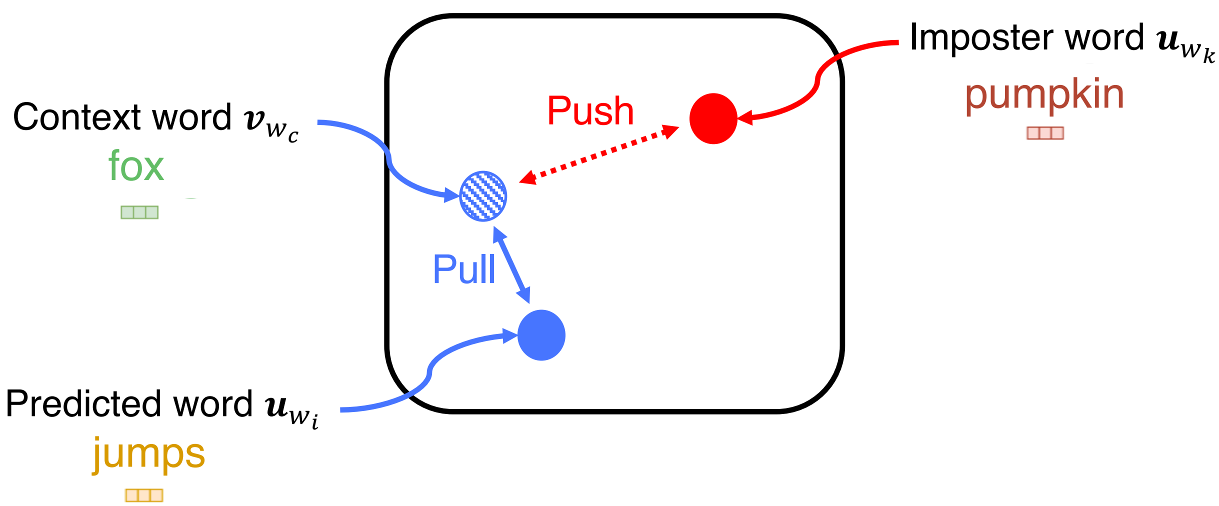
$$\log \sigma\left(\boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c}\right) + \sum_{k \in \mathcal{W}_{NS}} \log\left(1 - \sigma\left(\boldsymbol{u}_{w_k} \cdot \boldsymbol{v}_{w_c}\right)\right)$$

  - An effective sampling distribution is weighted uniform distribution

$$P_{NS}(w) \sim U(w)^{3/4}$$

17

Geometric Interpretation of Negative Sampling



$$\log \sigma\left(\boldsymbol{u}_{w_i} \cdot \boldsymbol{v}_{w_c}\right) + \sum_{k \in \mathcal{W}_{NS}} \log\left(1 - \sigma\left(\boldsymbol{u}_{w_k} \cdot \boldsymbol{v}_{w_c}\right)\right)$$

18

Analogical Reasoning

WORD2VEC embeddings are good at semantic & syntactic analogies

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Mikolov et al, ICLR 2013

19

Analogical Reasoning with Phrases

- WORD2VEC can learn semantic relationships with phrases

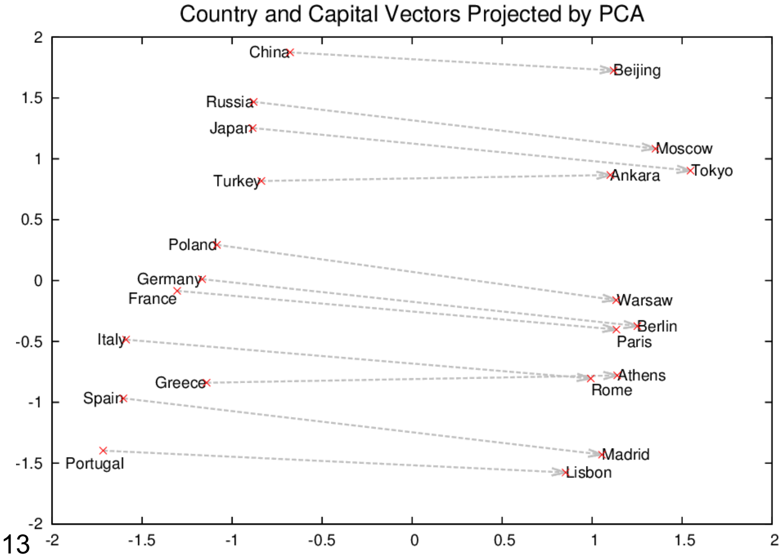
Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Mikolov et al., NeurIPS 2013

20

Analogical Reasoning

Two-dimensional projection shows an ability to learn semantic concepts and linear relations between concepts



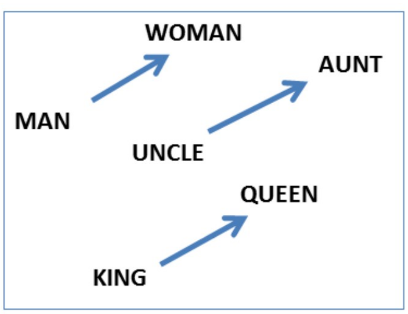
Mikolov et al, NeurIPS 2013

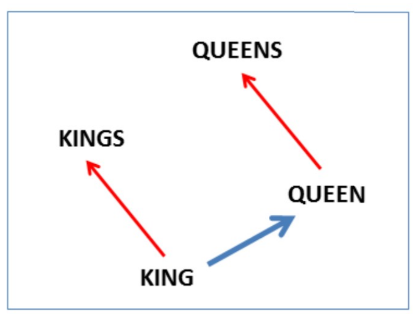
21

Additive Compositionality

WORD2VEC vectors capture semantic relationships via addition

e.g., $\boldsymbol{v}_{king} - \boldsymbol{v}_{man} + \boldsymbol{v}_{woman} \approx \boldsymbol{v}_{queen}$





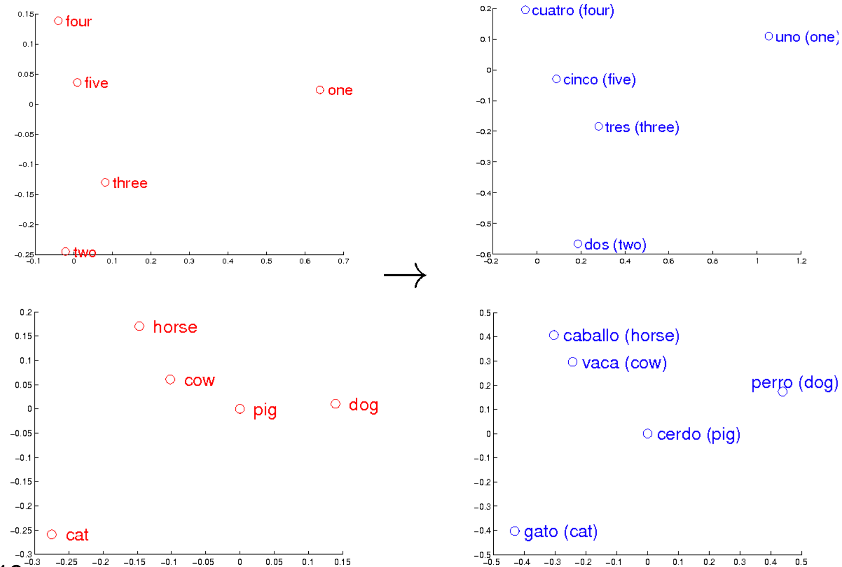
Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Mikolov et al, NeurIPS 2013

22

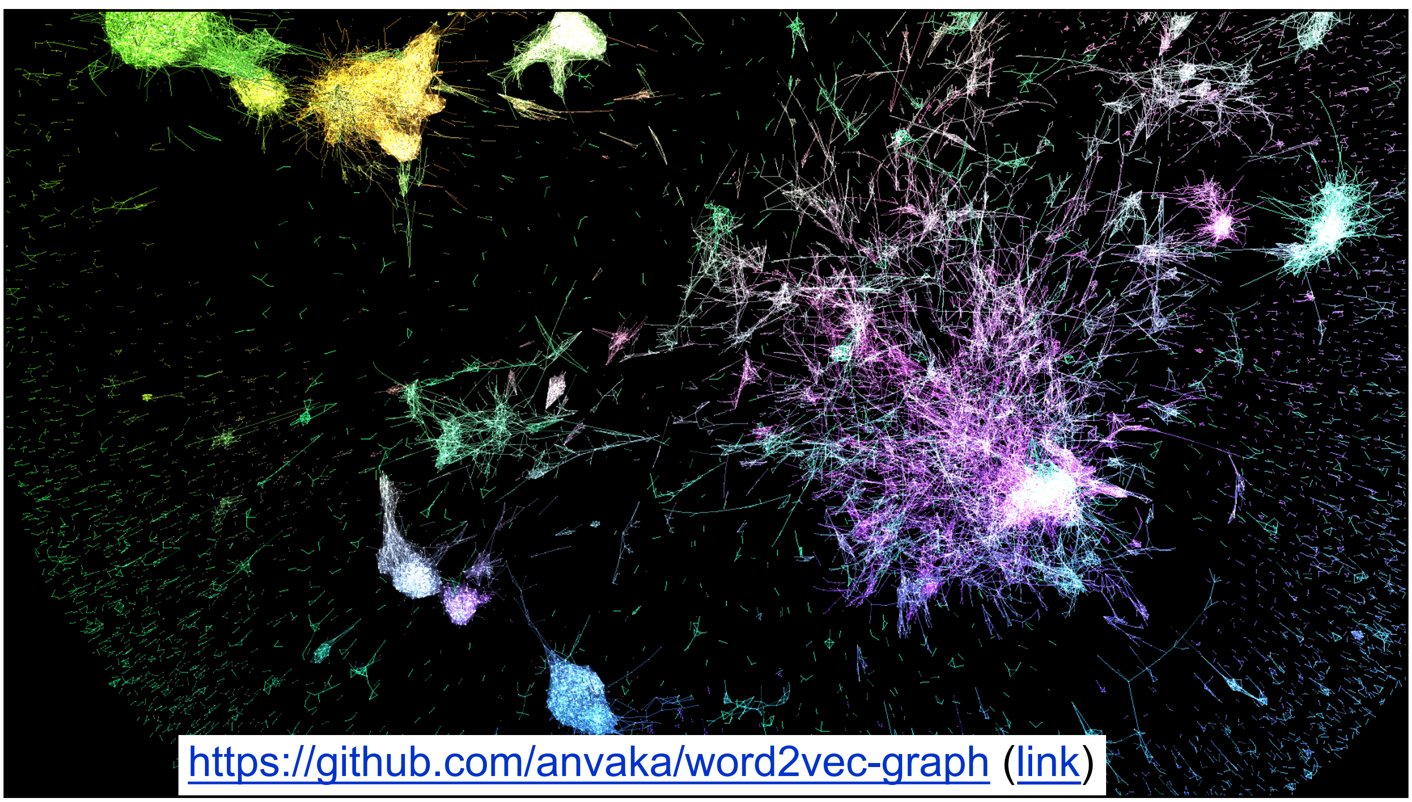
Multilingual Embedding Spaces

Word embedding spaces across languages have geometric similarities



Mikolov et al, arXiv 2013

23



https://github.com/anvaka/word2vec-graph (link)

24

GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

25

Global Vector (GloVe) Embeddings

- WORD2VEC is able to capture syntactic and semantic relationships via local contexts, but ignores global co-occurrence statistics (LSA)
- GloVe is based on idea that ratios of co-occurrence probabilities are informative about meaning relationships between words
  - Define $p(w_j|w_i) = P_{ij} = \frac{X_{ij}}{X_i}$ (X_{ij} counts w_j occurrences in context of w_i)

Prob & ratio	k = solid	k = gas	k = water	k = fashion
$p(k\|ice)$	0.00019	0.000066	0.003	0.000017
$p(k\|steam)$	0.000022	0.00078	0.0022	0.000018
$p(k\|ice)/p(k\|steam)$	8.9	0.085	1.36	0.96

  - Ratios >> 1 or << 1 are informative about meaning relationships

26

GloVe Formulation

- Preserve co-occurrence relation between w_i, w_j, and probe $\widetilde{w}_k$ $\quad F(w_i, w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}}$
  - Linear behavior in vector space: $\quad F\left(\boldsymbol{v}_{w_i} - \boldsymbol{v}_{w_j}, \boldsymbol{u}_{w_k}\right) = \frac{P_{ik}}{P_{jk}}$
  - Scalar distance metric: $\quad F\left(\left(\boldsymbol{v}_{w_i} - \boldsymbol{v}_{w_j}\right) \cdot \boldsymbol{u}_{w_k}\right) = \frac{P_{ik}}{P_{jk}}$
  - Symmetry between w_i and w_j: $\quad \frac{F\left(\boldsymbol{v}_{w_i} \cdot \boldsymbol{u}_{w_k}\right)}{F\left(\boldsymbol{v}_{w_j} \cdot \boldsymbol{u}_{w_k}\right)} = \frac{P_{ik}}{P_{jk}}$
  - Exponential function for F:

$$e^{\boldsymbol{v}_{w_i} \cdot \boldsymbol{u}_{w_k}} = P_{ik} = \frac{X_{ik}}{X_i} \qquad \boxed{\boldsymbol{v}_{w_i} \cdot \boldsymbol{u}_{w_k} + b_i + c_k = \log X_{ik}}$$

27

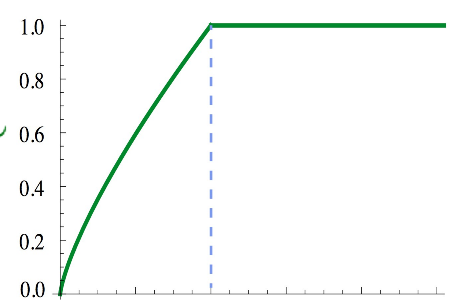
GloVe Formulation (con't)

- The weighted least-squares loss function can be represented as

$$L(\boldsymbol{\theta}) = \sum_{i,j=1}^{V} f(X_{ik}) \left(\boldsymbol{v}_{w_i} \cdot \boldsymbol{u}_{w_j} + b_i + c_j - \log X_{ij}\right)^2$$

  - Note the summations over vocabulary, as opposed to corpus
  - The weighting function is used for zero entries, scales counts < 100

$f \sim$

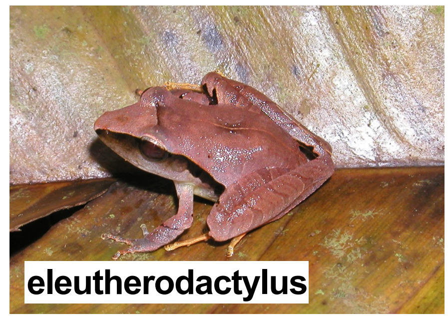


28

GloVe Word Similarities

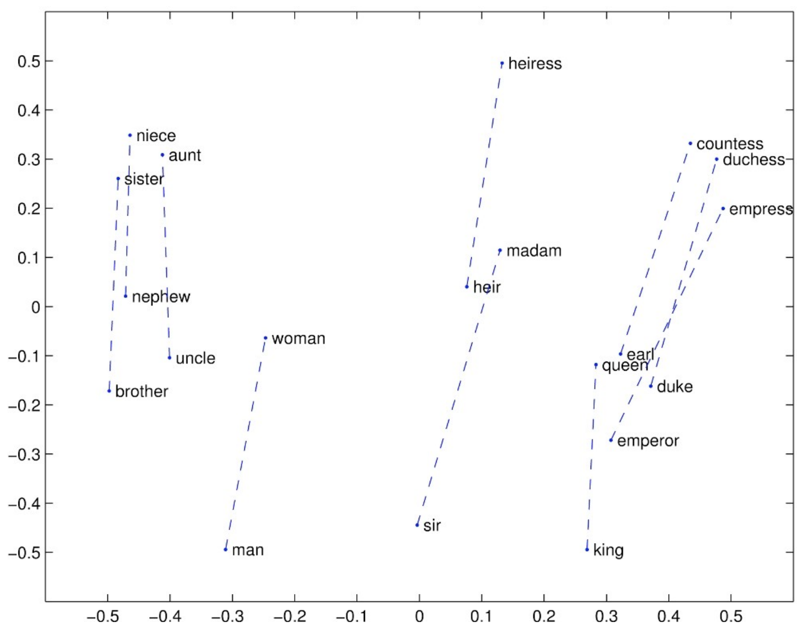
Nearest words to frog:

1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus



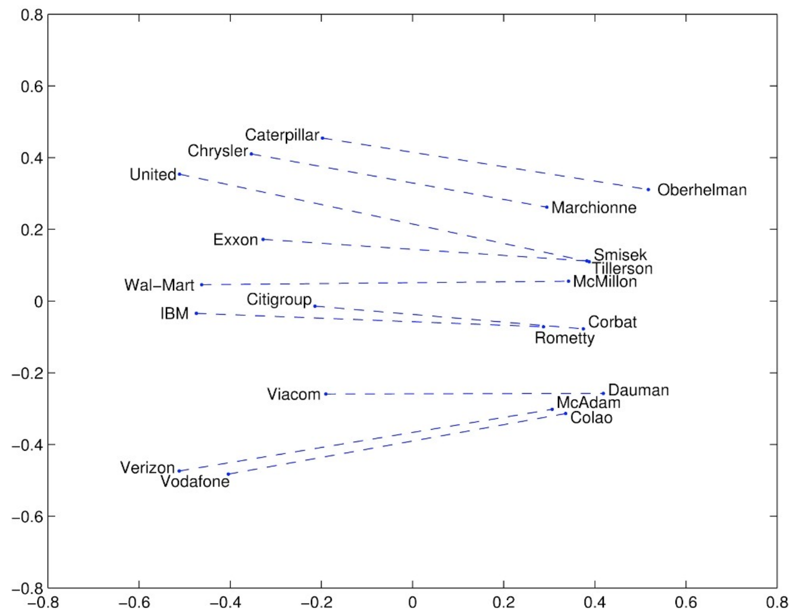
https://nlp.stanford.edu/projects/glove/

GloVe Linear Relationships



https://nlp.stanford.edu/projects/glove/

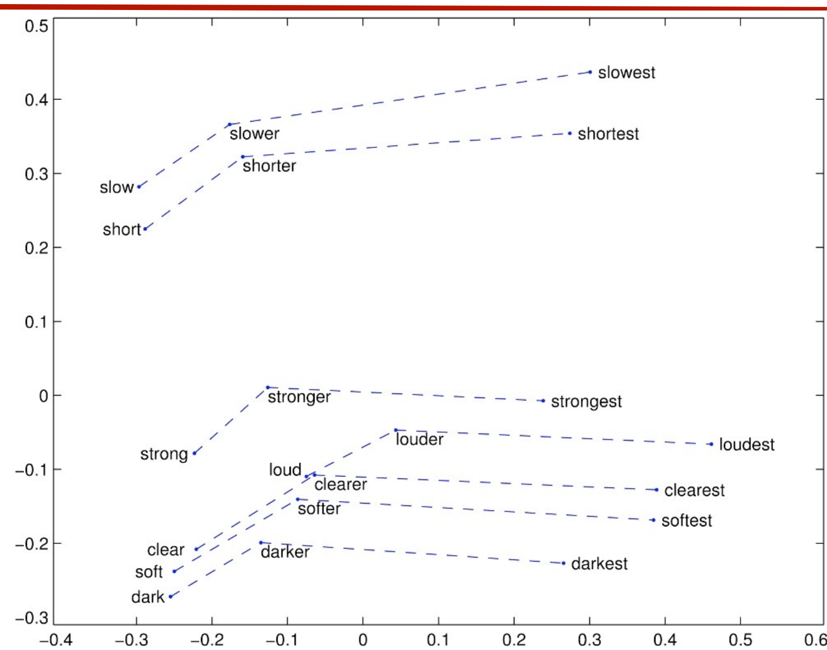
GloVe Company-CEO Relationships



https://nlp.stanford.edu/projects/glove/

31

GloVe Grammatical Relationships



https://nlp.stanford.edu/projects/glove/

32

FastText

Enriching Word Vectors with Subword Information

Piotr Bojanowski* and **Edouard Grave*** and **Armand Joulin** and **Tomas Mikolov**
Facebook AI Research

Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017. Action Editor: Hinrich Schütze.
Submission batch: 9/2016; Revision batch: 12/2016; Published 6/2017.

33

Modeling Subword Information in Vector Representations

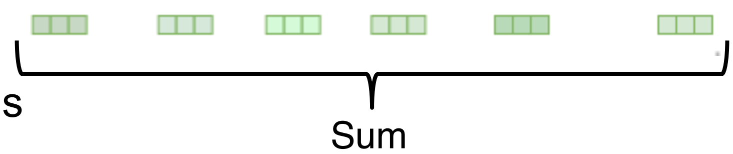
- Word-based embedding representations ignore morphology
  - No mechanism for parameter sharing across words
  - No mechanism to produce vectors for out-of-vocabulary (OOV) words
  - Problematic for morphologically rich languages with large vocabularies
- Since words tend to follow morphological rules, it is possible to improve vector representations using subword level information
- FastText represents words by a bag of character n-grams
  - A vector representation is associated with each character n-gram
  - A word vector is the sum of its character n-gram vectors
  - Training based on an extension of the WORD2VEC skip-gram model

34

FastText Illustration

- Add boundary markers — jumps → <jumps>
- Divide words into character n-grams — <jumps> → <ju jum ump mps ps>

  e.g., Character trigrams
- Context vector based on n-gram and word vectors — <ju jum ump mps ps> <jumps>



- Learn n-gram embeddings via skip-gram training

35

Multilingual Word Analogies

- FastText is better at syntactic, but worse at semantic analogies

Singular/Plural	Base/Comparative	Semantic Analogy
cat → cats	good → better	man → king
dog → ?	rough → ?	woman → ?

	Skip-gram	CBOW	FastText	Skip-gram	CBOW	FastText
Czech	52.8	55.0	**77.8**	25.7	**27.6**	27.5
German	44.5	45.0	**56.4**	66.5	**66.8**	62.3
English	70.1	69.9	**74.9**	**78.5**	78.2	77.8
Italian	51.5	51.8	**62.7**	52.3	**54.7**	52.3

36

WORD2VEC vs GloVe vs FastText

- All are neural methods for learning word embedding vectors
  - WORD2VEC and FastText learn from local contexts
  - GloVe learns from global word co-occurrence statistics
  - All do well with few hundreds of dimensions on many tasks
  - All have publicly available pre-computed vectors
- GloVe is faster to train than WORD2VEC and FastText
  - WORD2VEC and FastText iterate over entire training data
  - GloVe iterates over vocabulary, can be implemented in parallel
- FastText is better able to cope with morphologically rich languages
- No one method does consistently better on all tasks
  - All capture distributional semantics via distributed representations

37

Bias in Word Embeddings

- Machine learning methods that use data to determine model parameters are susceptible to acquiring bias present in the data
  - Word embeddings acquire bias due to the context in which words occur
  - Word embeddings can amplify bias and cause representational harm
  - Attempts to debias word embeddings is an open research problem

Extreme *she*	**Extreme *he***
homemaker	maestro
nurse	skipper
receptionist	protege
librarian	philosopher
socialite	captain
hairdresser	architect
nanny	financier
bookkeeper	warrior
stylist	broadcaster
housekeeper	magician

	Gender stereotype *she-he* analogies	
sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	lovely-brilliant
	Gender appropriate *she-he* analogies	
queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Bolukbasi et al., Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, 2016

38

Concluding Points

- The transition from symbolic representation to distributed vector representations had a major impact on NLP over the last decade
  - WORD2VEC appeared 2013, GloVe in 2014, FastText in 2016
  - Word embeddings were quickly adopted by NLP community
- Word embeddings can be good initializations for NLP models, and fine-tuned with task-specific data
- Embedding vectors have reduced or eliminated the significant feature engineering that went on with earlier (probabilistic) models
- Embedding vector representations have been extended to characters, sentences, documents, graphs etc. for many NLP tasks

39

Final Thought

"…the complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously."

J.R. Firth, Philological Society, 1935

- Starting in 2018, a new generation of contextual embedding representations such as ELMo and BERT have appeared
  - Stay tuned for contextual word embeddings!

40

References

- Extra Readings:
  - Eisenstein, “Natural Language Processing,” 2018 (Chp. 14 Distributional Semantics)
  - Jurafsky & Martin, “Speech and Language Processing,” 2020 (Chp. 6 Vector Semantics)
- On-line resources:
  - https://code.google.com/archive/p/word2vec/
  - https://nlp.stanford.edu/projects/glove/
  - https://fasttext.cc/

41