Beyond the Black Box
Word2Vec

Teddy Roland, UCSB

March 31, 2017

Workshop Repository: github.com/teddyroland/BBB-Word2Vec

Why Word2Vec?

1. Word Embeddings
2. Performance on NLP Tasks
3. Really, it’s just Google
4. But also Humanists

1. Word Embeddings

Vector Representations of "Bank"

One-Hot Vector (Sparse)

[… 0 0 1 0 0 …]

Word Embedding (Dense)

[0.025349 0.017121 0.029404 -0.09687 0.12685201 …]

Note: This and all other *word embeddings* in this presentation come from the word2vec model trained on the ECCO-TCP corpus, distributed by Ryan Heuser; http://ryanheuser.org/data/word2vec.ECCO-TCP.txt.zip

2. Performance

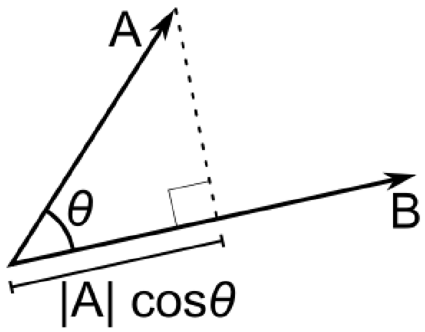
Vector Semantics: Similarity

Most Similar Vectors to “Bank”

Word	*Cosine Similarity*
ground	0.657
turf	0.656
surface	0.648
declivity	0.642
hill	0.637
bridge	0.633
terrace	0.630
channel	0.629

Bank

[0.025349 0.017121 0.029404 -0.09687 …]



Cosine Similarity; image from [Wikipedia]

Vector Semantics: Multiple Valences

Similar to “Bank” but not “River”

Word	*Cosine Similarity*
currency	0.565
payable	0.488
poor’s	0.476
bullion	0.465
exports	0.449
payments	0.447
coining	0.438
redeemable	0.437

Bank

[0.025349 0.017121 0.029404 -0.09687 …]

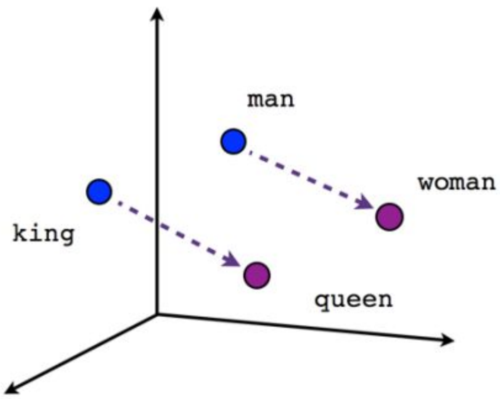
River

[-0.060448 0.134895 -0.082511 -0.147266 …]

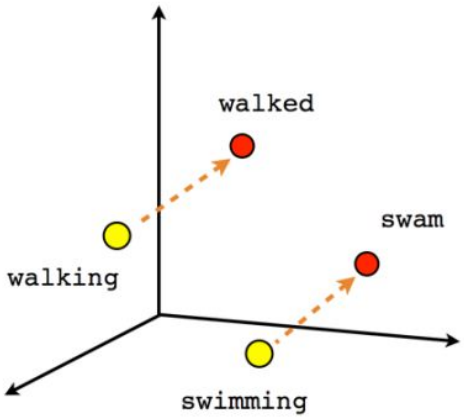
Bank - River

[0.085797 -0.11779 0.111915 0.050396 …]

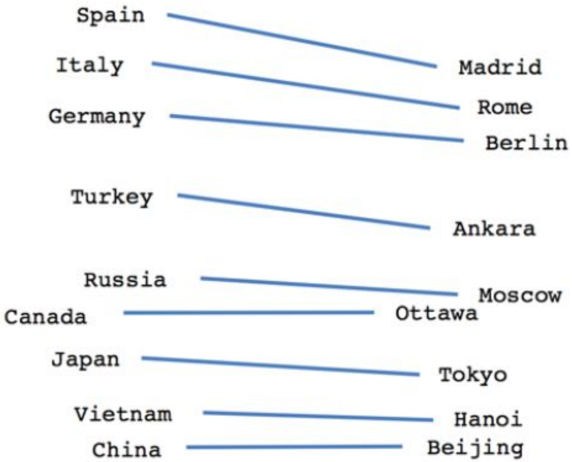
Vector Semantics: Analogy



Male-Female



Verb tense



Country-Capital

Image from Tensorflow tutorial: https://www.tensorflow.org/tutorials/word2vec

3. Google

Accuracy vs. Computation Time

from Mikolov et al (2013) abstract:

We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set.

Existing Architecture (for Image Processing)

Figure 16. Most responsive stimuli on the test set for the cat neuron.

from Le et al (2012) “Building High-level Features Using Large Scale Unsupervised Learning”

Open Sourced!

word2vec, Google Open Source Blog (2013)

“[word2vec] has a very broad range of potential applications [...] We’re open sourcing the code for computing these text representations efficiently (on even a single machine) so the research community can take these models further.”

Neural Network, Google *Research Blog* (2015)

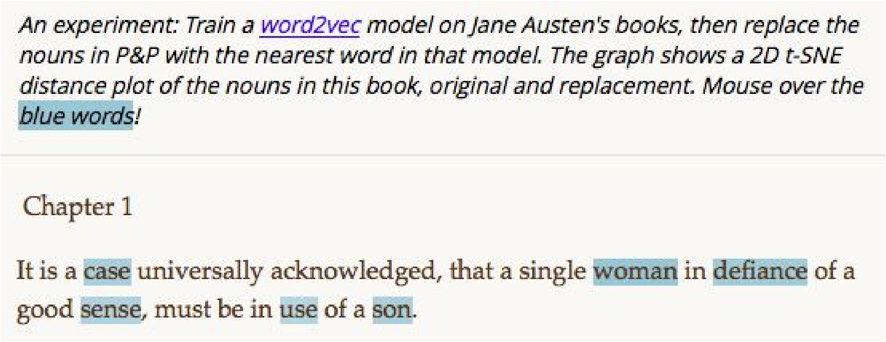
“But the most important thing about TensorFlow is that it’s yours. We’ve open-sourced TensorFlow as a standalone library and associated tools, tutorials, and examples with the Apache 2.0 license so you’re free to use TensorFlow at your institution (no matter where you work).”

4. Humanists

Humanistic Projects

Cherney (2014)

Pride & Prejudice & Word Embedding Distance



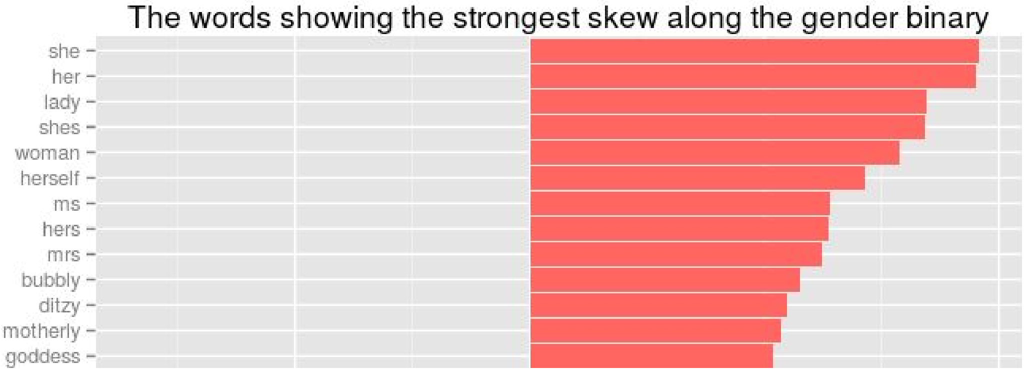
An experiment: Train a word2vec model on Jane Austen's books, then replace the nouns in P&P with the nearest word in that model. The graph shows a 2D t-SNE distance plot of the nouns in this book, original and replacement. Mouse over the blue words!

Chapter 1

It is a case universally acknowledged, that a single woman in defiance of a good sense, must be in use of a son.

Schmidt (2015)

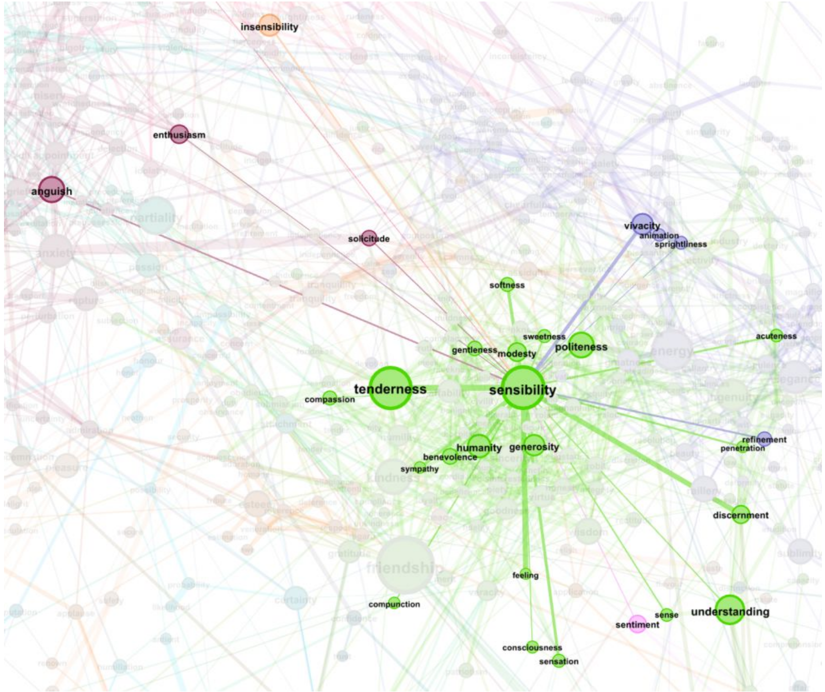
Rejecting the Gender Binary



Humanistic Projects

Heuser (2016)

Word Vectors in the Eighteenth Century



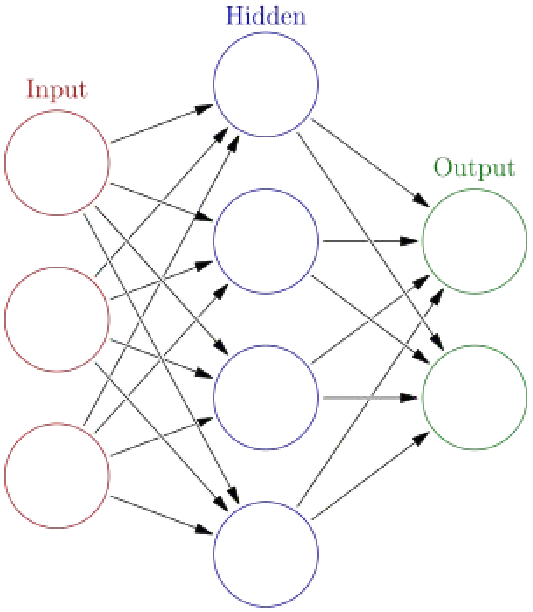
The Algorithm

1. Neural Networks
2. Backpropagation, or the Value of Big Data
3. Function Approximation, or the Black Box
4. Word2Vec as NN

   4a. but not a Black Box!

1. Neural Networks (very briefly)

Neural Network Architecture



Generic Neural Network Diagram;
Image from [Wikipedia]

2. Backpropagation

Simple Algebra?

$$3 * X = 21$$

Not Algebra but Arithmetic

$$3 * X = 21$$

A. Guess a random value for X
B. Observe output from that value (left side of equation)
C. Measure deviation from “correct” answer (right side of equation)
D. Adjust guess to compensate for error

Not Algebra but Arithmetic

$$3 * X = 21$$

A. Guess a random value for X
B. Observe output from that value
C. Measure deviation from “correct” answer
D. Adjust guess to compensate for error

Try **X = 2**

3 * **2** → 6

21 - 6 = 15

… so **add 5** to initial guess

(to increase “output” by 15)

2 + 5 = 7

Not Algebra but Arithmetic

$$3 * X = 21$$

$$3 * X * Y = 21$$

$$3 * \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} * \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ y_{3,1} & y_{3,2} & y_{3,3} \end{bmatrix} = 21$$

3. Function Approximation

Imagine you’re a physicist in 1590...

$$height(t) = height_{initial} - \frac{g}{2} \cdot t^2$$

time (s)	***height (m, approx)***
0	50
1	45
2	30
3	5
~3.3	0

… but with a Neural Network

$0 * X \ * Y \ = 50$

update X, Y

$1 * X' * Y' = 45$

update X’, Y’

$2 * X' * Y' = 30$

update X’, Y’

$3 * X' * Y' = 5$

update X’, Y’

$3.3 * X' * Y' = 0$

update X’, Y’

same dataset but no Law of Nature

4. Word2Vec as NN

Word2Vec Architecture

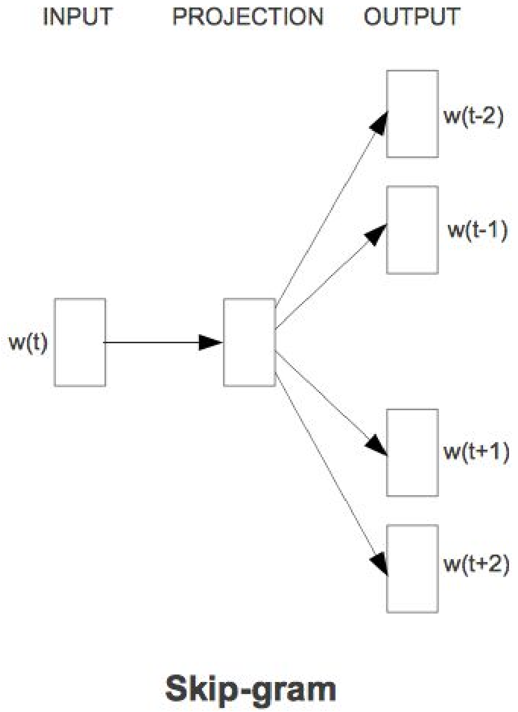


Figure 1. “New model architectures. [...] Skip-gram predicts surrounding words given the current word.” *from* Mikolov et al (2013)

Word2Vec Architecture

The cat <u>sat</u> on the mat.

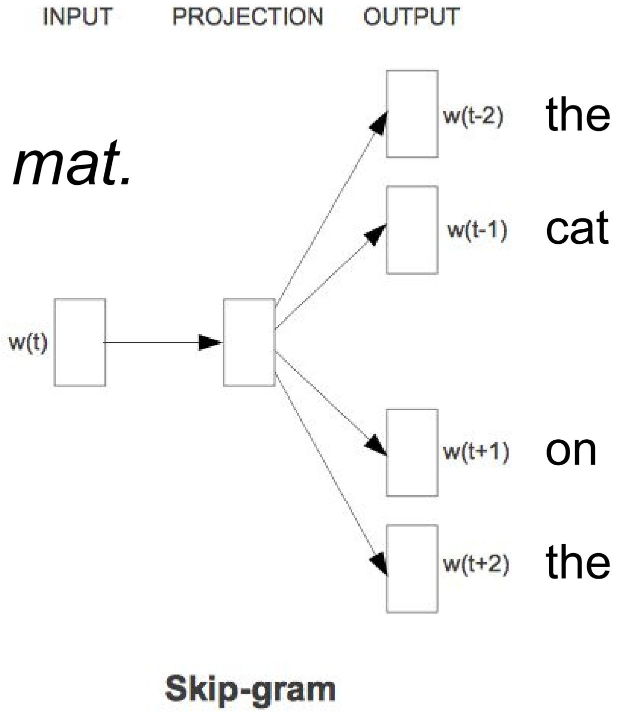


Figure 1. “New model architectures. [...] Skip-gram predicts surrounding words given the current word.” *from* Mikolov et al (2013)

Word2Vec Embeddings

“sat” * $\underline{\mathbf{X}}$ * $\mathbf{Y}$ = [“the”, “cat”, “on”, “the”]

Rows of *X* matrix correspond to each unique word in the corpus.

Word-Context Matrix

	20	2008	2009	2010	2011	able	academic	academy	access	address
american	0	2	0	1	3	1	0	1	0	0
analysis	0	0	0	0	0	1	0	0	0	1
art	0	0	0	0	0	0	0	0	0	0
arts	0	0	0	0	1	0	1	2	0	0
association	0	1	0	2	3	0	0	0	0	0
attention	0	0	0	0	0	0	0	1	0	0
author	0	0	0	0	1	0	0	0	0	0
available	0	0	0	0	0	1	1	0	0	1
based	0	0	0	0	0	0	0	0	0	0
big	0	0	0	1	0	0	1	0	0	0
blog	0	1	0	1	2	0	0	1	1	0

Selection from example word-context matrix for *Debates in the Digital Humanities* (2012).

continued in Jupyter Notebook…

Resources

Computer Science

Le et al (2012), “Building High-level Features Using Large Scale Unsupervised Learning” https://arxiv.org/pdf/1112.6209.pdf

Mikolov et al (2013), “Efficient Estimation of Word Representations in Vector Space” https://arxiv.org/pdf/1301.3781.pdf

Levy, Goldberg (2014), “Neural Word Embedding as Implicit Matrix Factorization” http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf

Open-Source Software Announcements

Google *Research Blog*, Tensorflow https://research.googleblog.com/2015/11/tensorflow-googles-latest-machine_9.html

Google *Open Source Blog*, word2vec https://opensource.googleblog.com/2013/08/learning-meaning-behind-words.html

Resources

Humanities Word2vec Projects

Cherney (2014), “Pride & Prejudice & Word Embedding Distance” http://www.ghostweather.com/files/word2vecpride/

Schmidt (2015), “Rejecting the Gender Binary: A Vector-Space Operation” http://bookworm.benschmidt.org/posts/2015-10-30-rejecting-the-gender-binary.html

Heuser (2016), “Word Vectors in the Eighteenth Century” http://ryanheuser.org/word-vectors/