Word embeddings in 2017

A review of current trends in state-of-the art word vector methods

Théo Matussière
RALI, DIRO
Université de Montréal

Wed 12 Jul 2017

What are word embeddings?

- an *old* idea: G. Hinton first discussed Vector Space Model for words in 1984[1]
- a convenient way to represent words as vectors of $\mathbb{R}^d$

[1] G. E. Hinton. "Distributed representations". In: (1984).

What are they for?

- they embed meaning & sense in a continuous space
- they give access to a *lot* of tools from linear algebra

What are they for?

- they embed meaning & sense in a continuous space
- they give access to a *lot* of tools from linear algebra

Which leads to cool stuff:

> W. L. Hamilton, J. Leskovec, and D. Jurafsky. "Diachronic word embeddings reveal statistical laws of semantic change". In: *arXiv preprint arXiv:1605.09096* (2016), website here

> Ben Schmidt, 2015; gender and language

Cool stuff

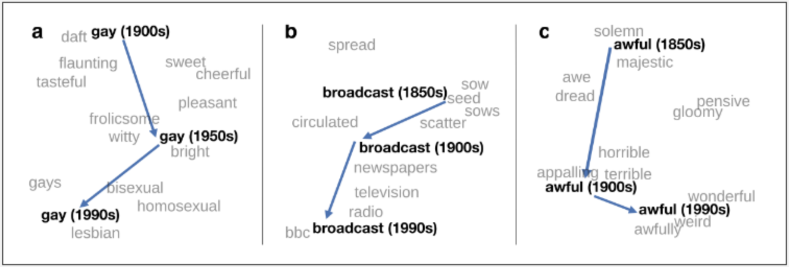


Figure 1: Hamilton, Leskovec, and Jurafsky 2016

Cool stuff

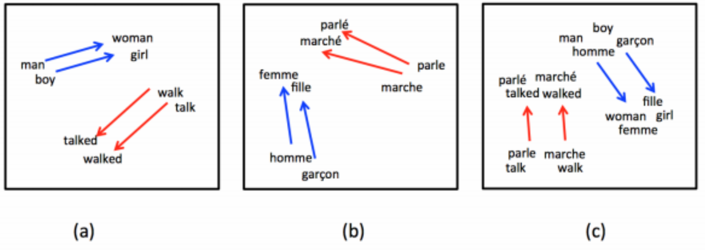


Figure 1. (a & b) Monolingual embeddings have been shown to capture syntactic and semantic features such as noun gender (blue) and verb tense (red). (c) The (idealized) goal of crosslingual embeddings is to capture these relationships across two or more languages.

Figure 1: Gouws, Bengio, and Corrado 2015

Embed meaning?

king - man $+$ woman $=$ queen

Embed meaning?

king - man + woman = queen

i.e. semantical algebra

The big idea

Distributional semantics

The big idea

Distributional semantics

A word is characterized by the company it keeps.

John R. Firth

Summary

Classical Algorithms

- Count-based
- Predictive methods

State of the art

- Improvements
- Single vs. Multi prototypes

The evaluation problem

- Current methods
- Limitations

Classical Algorithms

Basic settings

- pick a corpus
- set k a threshold for rare words
- rank words according to frequency in corpus
- assign to each word its ranking, lexicographically for same-rank words

the	1
of	2
...	...
dog	3324
...	...

Old & new

As per Baroni[2] we'll differentiate

- count-based methods
- predictive methods

Both share a common thing: dimensionality reduction for complex domains.

[2] M. Baroni, G. Dinu, and G. Kruszewski. "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors.". In: *ACL (1)*. 2014, pp. 238–247.

Classical Algorithms

Count-based

Count-based methods: naive

Counting cooccurences in $M \in \mathbb{R}^{|V| \times |V|}$ where:

$$M_{i,j} = \#\{i \text{ within } k \text{ words of } j\}$$

To the embedding space

SVD: Singular Value Decomposition.

To the embedding space

SVD: Singular Value Decomposition.

PCA: Principal Component Analysis.

Count-based methods: subtler

$$M_{i,j} = \mathrm{PMI}(i, j)$$

Count-based methods: subtler

$$M_{i,j} = \mathrm{PMI}(i,j)$$

$$\mathrm{PMI}(i,j) = \log \frac{\mathbb{P}[i,j]}{\mathbb{P}[i]\mathbb{P}[j]}$$

Count-based methods: subtler

$$M_{i,j} = \mathrm{PMI}(i, j)$$

$$\mathrm{PMI}(i, j) = \log \frac{\mathbb{P}[i, j]}{\mathbb{P}[i]\mathbb{P}[j]}$$

$$\hat{\mathrm{PMI}}(i, j) = \log \frac{\#(i, j) * \text{corpus size}}{\#(i)\#(j)}$$

Limitations

SVD are computationally expensive

Prohibitive for 50K+ vocabularies.

Classical Algorithms

Predictive methods

The cool cats

The cool cats

Lot of noise from three papers by Tomas Mikolov in 2013

T. Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119

T. Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013)

T. Mikolov, Q. V. Le, and I. Sutskever. "Exploiting similarities among languages for machine translation". In: *arXiv preprint arXiv:1309.4168* (2013)

Word2Vec

Umbrella term for two different algorithms

Word2Vec

CBOW:	`c0 c1 c2 (w) c3 c4 c5` $\longrightarrow$ `w`
Skip-Gram:	`w` $\longrightarrow$ `c0 c1 c2 (w) c3 c4 c5`

Word2Vec

Deep learning? Not really: with vocabulary matrices $M, C \in \mathbb{R}^{|V|.d}$

$$
\begin{array}{r|ccc|c}
\multicolumn{1}{r}{} & \multicolumn{3}{c}{\leftarrow d \rightarrow} & \multicolumn{1}{c}{} \\
\text{the:} & 0.21 & \dots & \text{-1.01} & \uparrow \\
 & \dots & \dots & \dots & |V| \\
\text{california:} & \text{-0.60} & \dots & 0.09 & \downarrow \\
 & \dots & \dots & \dots &
\end{array}
$$

Word2Vec

Step 1:

$$M \cdot \mathbb{I}[w] = v \in \mathbb{R}^d$$

Word2Vec

Step 1:

$$M \cdot \mathbb{I}[w] = v \in \mathbb{R}^d$$

Step 2:

$$v \cdot C = w \in R^{|V|}$$

$$\sigma(w) = [\sigma(< v, C_i >)]_{1<i<|V|}$$

Word2Vec

Step 1:

$$M \cdot \mathbb{I}[w] = v \in \mathbb{R}^d$$

Step 2:

$$v \cdot C = w \in R^{|V|}$$

$$\sigma(w) = [\sigma(< v, C_i >)]_{1<i<|V|}$$

Step 3: backprop on rows of M, C

Word2Vec

Spread fast because of its:

- speed: runs in half a day where previous algorithms ran in weeks.
  - Hierarchical Softmax
  - Noise Contrastive Estimation
- ease of use: released code
- test set

and despite its incomprehensible paper.

GloVe

Count-based *and* predictive, its objective ponders the dot product by a function of cooccurence.[3]

$$J = \sum_{ij} f(M_{ij})\big(< w_i, w_j > + b + \log M_{ij}\big)$$

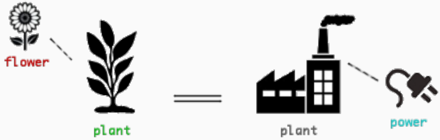
[3] J. Pennington, R. Socher, and C. D. Manning. "Glove: Global Vectors for Word Representation.". In: *EMNLP*. vol. 14. 2014, pp. 1532–1543.

Limitations

polysemy:

Limitations

$$d(x, z) \leq d(x, y) + d(y, z)$$



Count-based and predictive: they are the same![4] (and Baroni was wrong)

(PMI approximation)

[4] O. Levy and Y. Goldberg. "Neural word embedding as implicit matrix factorization". In: *Advances in neural information processing systems*. 2014, pp. 2177–2185.

Another idea

Manual feature engineering. (172K dimensions)

M. Faruqui and C. Dyer. "Non-distributional word vector representations". In: *arXiv preprint arXiv:1506.05230* (2015)

State of the art

Research jungle

Most of the papers introduce incremental innovation to Word2Vec;

- improving the pipeline
- improving the algorithm
- solving the polysemy issue

State of the art

Improvements

Improvements: finer inputs

Part of Speech annotated inputs:

A. Trask, P. Michalak, and J. Liu. "sense2vec-A fast and accurate method for word sense disambiguation in neural word embeddings". In: *arXiv preprint arXiv:1511.06388* (2015)

Improvements: finer inputs

apple	NOUN	1.0	apple	PROPN	1.0
apples	NOUN	.639	microsoft	PROPN	.603
pear	NOUN	.581	iphone	NOUN	.591
peach	NOUN	.579	ipad	NOUN	.586
blueberry	NOUN	.570	samsung	PROPN	.572
almond	NOUN	.541	blackberry	PROPN	.564

Figure 2: From Trask, Michalak, and Liu 2015

Improvements: finer inputs

Syntaxical parsing on Wikipedia fed to standard SGNS:

O. Levy and Y. Goldberg. "Dependency-Based Word Embeddings.". In: *ACL (2)*. Citeseer. 2014, pp. 302–308

Improvements: finer inputs

The dependency-based embeddings are less topical and exhibit more functional similarity than the original skip-gram embeddings.

Levy and Goldberg 2014a

Improvements: finer inputs

Target Word	BoW5	BoW2	Deps
batman	nightwing aquaman catwoman superman manhunter	superman superboy aquaman catwoman batgirl	superman superboy supergirl catwoman aquaman
hogwarts	dumbledore hallows half-blood malfoy snape	evernight sunnydale garderobe blandings collinwood	sunnydale collinwood calarts greendale millfield
turing	nondeterministic non-deterministic computability deterministic finite-state	non-deterministic finite-state nondeterministic buchi primality	pauling hotelling heting lessing hamming
florida	gainesville fla jacksonville tampa lauderdale	fla alabama gainesville tallahassee texas	texas louisiana georgia california carolina
object-oriented	aspect-oriented smalltalk event-driven prolog domain-specific	aspect-oriented event-driven objective-c dataflow 4gl	event-driven domain-specific rule-based data-driven human-centered
dancing	singing dance dances dancers tap-dancing	singing dance dances breakdancing clowning	singing rapping breakdancing miming busking

Figure 2: From Levy and Goldberg 2014a

Improvements: structural changes

All models work with the bag-of-word setting, let's structure it:

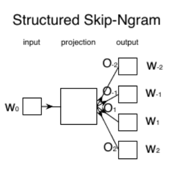


Figure 3: Ling et al. 2015

Improvements: structural changes

New loss based on inequalities to infuse real world knowledge:

> *In particular, these corpus-based methods usually fail to capture the precise meanings for many words. For example, some semantically related but dissimilar words may have similar contexts, such as synonyms and antonyms.* **As a result, these corpus-based methods may lead to some antonymous word vectors being located much closer in the learned embedding space than many synonymous words**.

Q. Liu et al. "Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints.". In: *ACL (1)*. 2015, pp. 1501–1511

Improvements: structural changes

Retrofitting from lexicons: same goal, using wordnet & co.

- fast post processing
- "improves quality"
- nothing on polysemy

M. Faruqui et al. “Retrofitting Word Vectors to Semantic Lexicons”. In: *Proceedings of NAACL*. 2015

Improvements: structural changes

Poincare Embeddings. Very cool idea, from Facebook:

> *Remarkably, [it] allows us therefore to learn embeddings that simultaneously capture the hierarchy of objects (through their norm) as well a their similarity (through their distance).*

(Though this exists in less exciting VSM as well.)

$>$ M. Nickel and D. Kiela. "Poincaré Embeddings for Learning Hierarchical Representations". In: *arXiv preprint arXiv:1705.08039* (2017)

But...

It's all more or less all the same once evaluation is unified...

- model vs parameters
- GloVe tricks
- count-based still useful

O. Levy, Y. Goldberg, and I. Dagan. "Improving distributional similarity with lessons learned from word embeddings". In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 211–225

Light break

The Office US

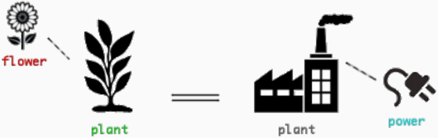
Kevin and context-based disambiguation

State of the art

Single vs. Multi prototypes

Single vs Multi prototypes

What to do with `plant` ?

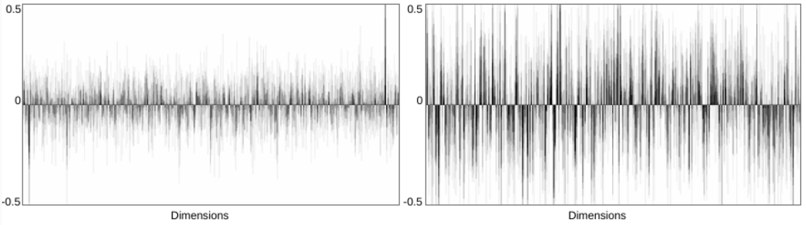


Single vs Multi prototypes

Interpretability and dimensions...

- **option 1**: assume all senses are embedded in the vector and recoverable
- **option 2**: senses might be embedded but untractable, need to assign vectors to each sense

Single vs Multi prototypes



10 random words: *emergency, bluff, buffet, horn, human, like, american, pretend, tongue, green*

10 felines: *cat, lion, tiger, leopard, cougar, cheetah, lynx, bobcat, panther, puma*

Figure 1: Heatmap histogram of 10 random words and 10 co-hyponyms in GloVe

Figure 3: Gladkova and Drozd 2016

Single vs Multi prototypes

Figure 1 compares the overlap of dimensions for 10 random words and 10 co-hyponyms in 300-dimensional GloVe vectors (darker dimensions indicate overlap between more words in the sample). It is clear that there are hundreds of features relevant for felines. We could hypothesize about them ("animal"? "nounhood"? "catness"?), but clearly this embedding has more "feline" features thanwhat we could find in dictionaries or elicit from human subjects. **Some of such features might not even be in our conceptual inventory.** *Perhaps there is a dimension or a group of dimensions created by the co-occurrences with words like jump, stretch, hunt, and purr some "feline behavior" category that we would not find in any linguistic resource.*

Gladkova and Drozd 2016

Sparse coding

Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data efficiently. The aim of sparse coding is to find a set of basis vectors ϕ_i *such that we can represent an input vector* $\mathbf{x}$ *as a linear combination of these basis vectors:*

$$\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$$

Original research from Princeton.

- isotropic property and dimensionality
- formal proof of PMI inducing semantical algebra
- sparse coding

> S. Arora et al. "A latent variable model approach to pmi-based word embeddings". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 385–399

> S. Arora et al. "Linear algebraic structure of word senses, with applications to polysemy". In: *arXiv preprint arXiv:1601.03764* (2016)

Sparse coding

Another attempt, with a twist: non negative sparse vectors for increased interpretability, and "binarization".

> M. Faruqui et al. "Sparse overcomplete word vector representations". In: *Proceedings of ACL*. 2015

Multi prototypes

Naive approach is parametric: set k senses for each word based on clustering method on all context in which the word appears.

$>$ E. H. Huang et al. "Improving word representations via global context and multiple word prototypes". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics. 2012, pp. 873–882

Multi prototypes

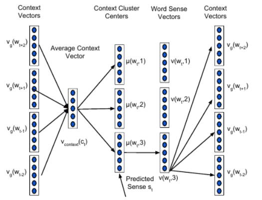


Figure 2: Architecture of Multi-Sense Skip-gram (MSSG) model with window size $R_t = 2$ and $K = 3$. Context c_t of word w_t consists of $w_{t-1}, w_{t-2}, w_{t+1}, w_{t+2}$. The sense is predicted by finding the cluster center of the context that is closest to the average of the context vectors.

> A. Neelakantan et al. "Efficient non-parametric estimation of multiple embeddings per word in vector space". In: *arXiv preprint arXiv:1504.06654* (2015)

Multi prototypes

The beast:

$$p(Y, Z, \boldsymbol{\beta}|X, \alpha, \theta) = \prod_{w=1}^{V} \prod_{k=1}^{\infty} p(\beta_{wk}|\alpha) \prod_{i=1}^{N} \left[p(z_i|x_i, \boldsymbol{\beta})) \prod_{j=1}^{C} p(y_{ij}|z_i, x_i, \theta) \right],$$

$>$ S. Bartunov et al. “Breaking sticks and ambiguities with adaptive skip-gram”. In: *Artificial Intelligence and Statistics*. 2016, pp. 130–138

Multi prototypes

Choosing a granularity parameter:

A **plant** is a living organism that generally does not move and absorbs nutrients from its surroundings. Typically it has been placed deliberately rather than naturally.

Look up *plant* in Wiktionary, the free dictionary.

Plant may also refer to:

In manufacturing and engineering [edit]

- Chemical plant
- Physical plant, often just called "plant", a facility's infrastructure (i.e., "Plant Room")
- Any type of mobile construction machinery
- Another name for a factory (short for "manufacturing plant")
- Processing plant, in process manufacturing

In media and entertainment [edit]

- Plant (snooker), used in British English to refer to a type of combination shot
- *The Plant* (newspaper), student newspaper at Dawson College in Montreal, Quebec, Canada
- PLANT, fictional organization in the anime series *Gundam SEED* and its sequel
- The Plants, a 1950s doo-wop group
- Record Plant recording studios, located at The Plant, in Sausalito, California
- The Plant (film), a 1995 television film
- The Plant

In names [edit]

- Henry B. Plant (1819–1899), American railroad manager
- Richard Plant (writer) (1910–1998), German-born American writer
- Richard Plant (racing driver) (born 1989)
- Robert Plant (born 1948), lead singer of Led Zeppelin

In people [edit]

- Plant (person), anyone assigned to behave as a member of the public during a covert operation (as in a police investigation)
- Plant (professional wrestling), a person hired to pose as a fan who may become involved in the events
- Plant, the creative member of a team in the Belbin Team Inventory
- Plant, a term used for a shill in the U.K.

Multi prototypes

Efficient assignment of word senses on all corpora?

The evaluation problem

Intrinsic value & Downstream tasks

How to rate word vectors?

De facto standards:

- analogy based
- similarity matching

Are there general purpose word embeddings?

The evaluation problem

Current methods

Analogies

Mikolov's test categories:

- capital-common-countries
- capital-world
- currency
- city-in-state
- family

Analogies

Mikolov's test categories:

- gram1-adjective-to-adverb
- gram2-opposite
- gram3-comparative
- gram4-superlative
- gram5-present-participle
- gram6-nationality-adjective
- gram7-past-tense
- gram8-plural
- gram9-plural-verbs

Similarities

Spearman correlation with human rated pairs of (dis)similar words:

- WordSim353
- MEN
- SCWS (contextualized, from Huang et al. 2012)
- ...

Similarities

WordSim353:

Word 1	Word 2	Human (mean)	1	2	3	4	5	6	7	8	9	10	11
love	sex	6.77	9	6	8	8	7	8	8	4	7	2	6
tiger	cat	7.35	9	7	8	7	8	9	8.5	5	6	9	7
tiger	tiger	10.00	10	10	10	10	10	10	10	10	10	10	10
book	paper	7.46	8	8	7	7	8	9	7	6	7	8	9
computer	keyboard	7.62	8	7	9	9	8	8	7	7	6	8	10

The evaluation problem

Limitations

Downstream tasks

Many of downstream applications: Machine Translation, Question Answering, IR, etc...

Current evaluation methods lack rigor, and are not correlated with good scores with end applications.

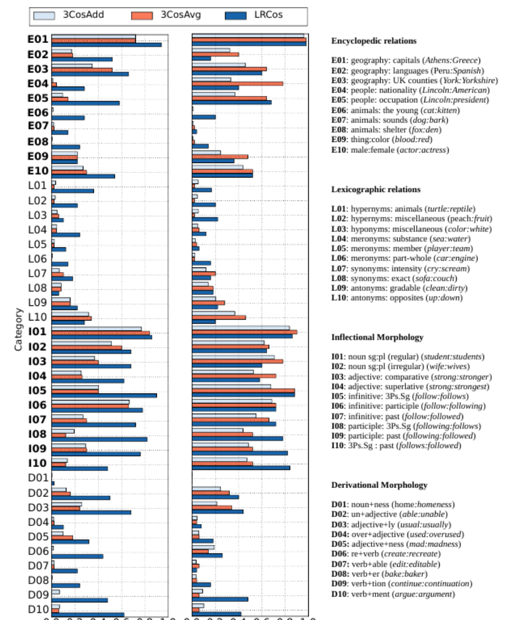
Weaknesses of current test sets

Mikolov analogy tests

Methodology issue, largely uncorrelated with end results.

> Drozd, Gladkova, and Matsuoka 2016

Weaknesses of current test sets



Weaknesses of current test sets

MEN, WordSim353:

Different type of relations rated together: `coffee` and `cup` are related, but not similar in the way `coffee` and `tea` are.

Which of them should have a higher mark?

$>$ SimLex999, Hill, Reichart, and Korhonen 2016, and
$>$ Avraham and Goldberg 2016 (order over ratings)

Word intrusion and evaluating by negative examples?

Overcoming intrinsic & extrinsic evaluation

QVEC method: aligning linguistic features (SemCor & Wordnet) with word vectors to measure the interpretability of dimensions.

> Y. Tsvetkov et al. "Evaluation of word vector representations by subspace alignment". In: (2015)

Conclusion

What to takeaway

- Word vectors are fondamental bricks of NLP
- Research is splitting between single and multi prototyping
- Intrinsic value of word embeddings remains to be defined... or chosen?

Questions

Thank you for your attention!

—

Questions?

theo.matussiere@gmail.com