Representation Learning for Text: Word Embeddings

Luciano Barbosa

Centro de Informática UFPE

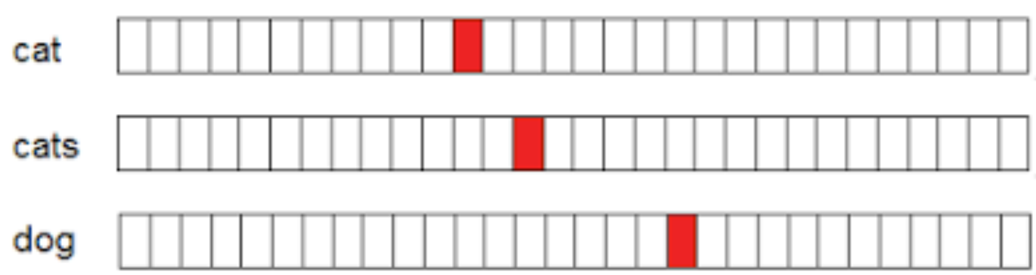
Universidade Federal de Pernambuco

Motivation for Deep Learning

- In theory, feed forward networks with a single hidden layer can compute any function: no need for deep architectures
- However, learning shallow architectures is not always efficient
- Some problems require an exponential number of hidden units

Feature Representation in Text: One Hot Encoding

- Each word mapped to a dimension and represented by a vector
- Example
  - V = {dog, bites, man}
  - D_1: "dog bites man" = {[1,0,0],[0,1,0],[0,0,1]}
  - D_2: "man bites dog" = {[0,0,1],[0,1,0],[1,0,0]}
- Dimensionality: size of vocabulary
- Naïve embedding
- Similar words have different representations



Img-Source: http://veredshwartz.blogspot.com.br/2016/01/representing-words.html

Feature Representation in Text: Bag of Words

- Each document represented by a vector
- Each dimension: a word with its respective weight
- Example
  - V = {the, cat, sat, on, hat, dog, ate, and}
  - D_1: "the cat sat on the hat" = {2, 1, 1, 1, 1, 0, 0, 0}
  - D_2: "the dog ate the cat and the hat" = {3, 1, 0, 0, 1, 1, 1, 1}
- Dimensionality: size of vocabulary
- Pros: Simple and very effective
- Cons:
  - Orderless
  - No notion of semantic similarity

Representation Learning

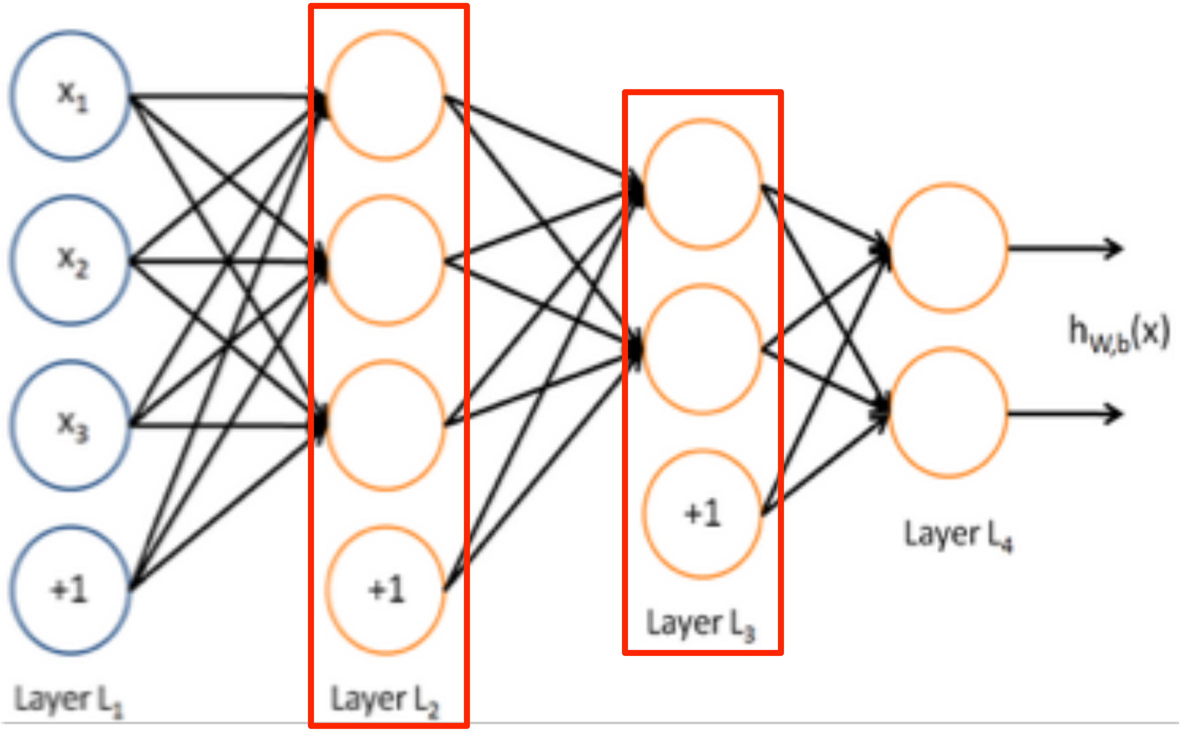
- In ML models, instances are represented by their features
- Motivation:
  - Instance/data representation is essential for effective ML models
  - Less dependent on feature engineering
  - Dimensionality reduction

- Definition:
  - Set of techniques that learn a "better" representation from the raw data

- Distributed representation or embeddings
  - Dense and low dimensional representation
  - Dimensions have no meaning

Example of Representation Learning: MLP

- Different levels of abstraction of the input



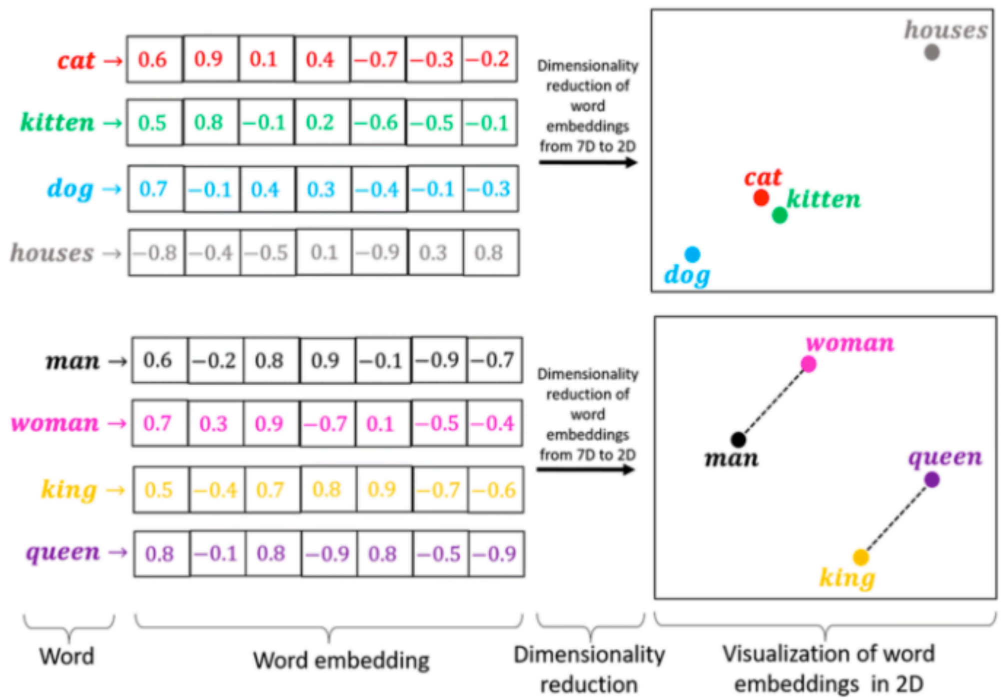
Img-Source: http://ufldl.stanford.edu/wiki/

Embeddings in NLP

- Represent a linguistic unit as a dense vector
- Linguistic unit: character, word, sentence, document
- Map semantic meaning into a geometric space (embedding space)

Word Embeddings

- Embed the context of a word in a low-dimensional vector (e.g., 100, 200)
- Similar words are close in this space



Img-Source: https://medium.com/@hari4om/word-embedding-d816f643140

Word Embeddings

- Built using dimensionality reduction techniques
  - Frequency based models (Latent Semantic Indexing)
  - Prediction based models (Neural networks)

Latent Semantic Indexing

- Build document and word representations
- Based on the co-occurrence of the words in the documents
- Dimensionality reduction: singular value decomposition (SVD)
  - $C = U\Sigma V^T$
  - C: term-document matrix
- Compute reduced C' with fewer dimensions

$C = U\Sigma V^T$

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

=

Term-Document Matrix

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

×

Word Embeddings

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

×

Singular values

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

Doc Embeddings

Reducing to 2 Dimensions

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

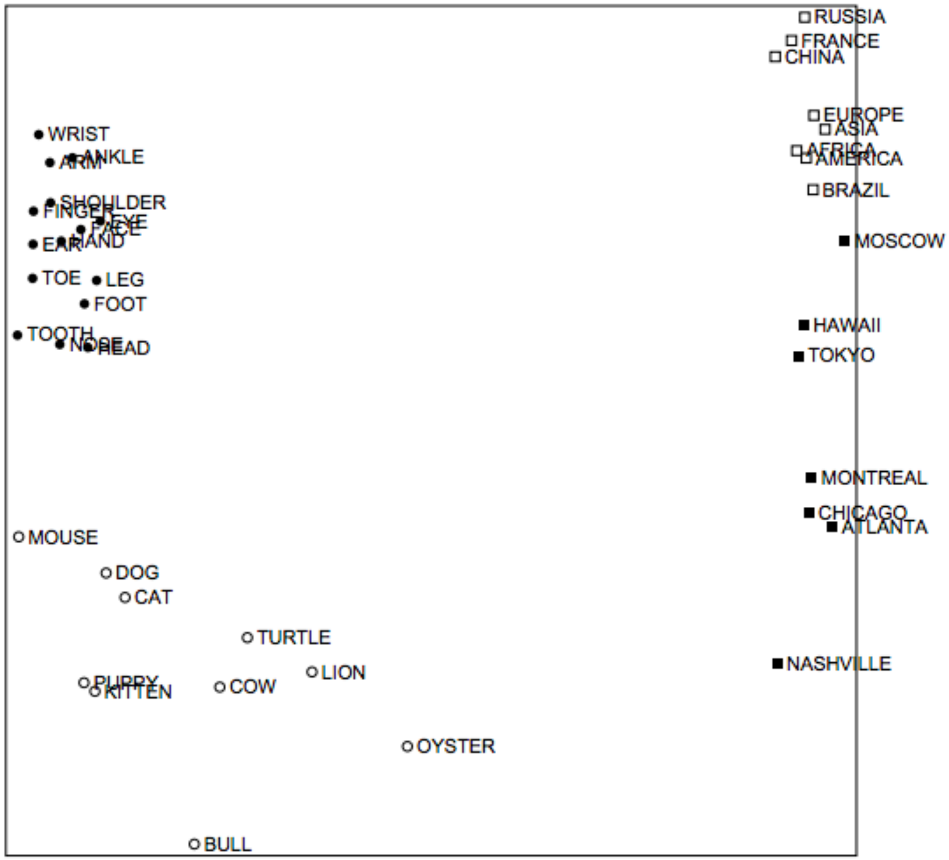
V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Word Embeddings

Setting singular values to 0

Doc Embeddings

Projecting LSI Word Embeddings



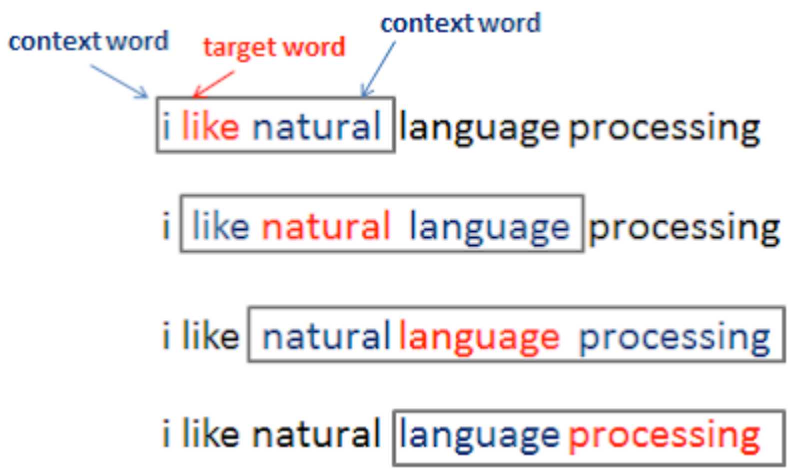
Rohde et al., An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence, 2005

Pros & Cons

- Pros
  - Simple
  - Capture similarity
- Cons
  - Co-occurrence matrix is sparse
  - Quadratic cost (SVD)

Prediction Based Models: CBOW

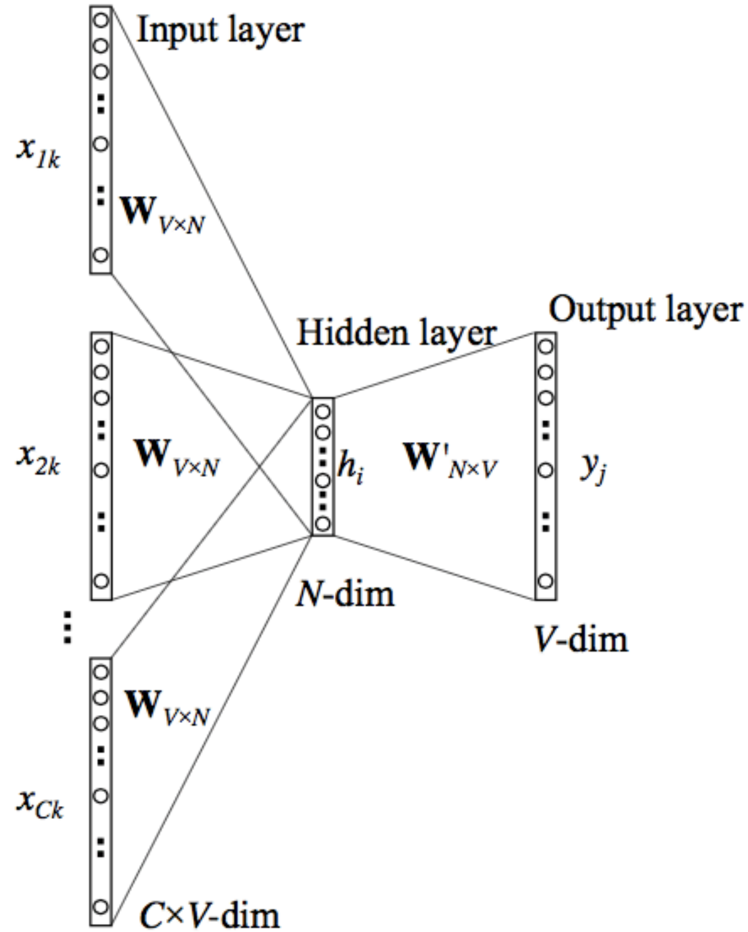
- Predicts word in the context (language modeling)



Img-Source:
https://thinkinfi.com/continuous-bag-of-words-cbow-multi-word-model-how-it-works/

Prediction Based Models: CBOW

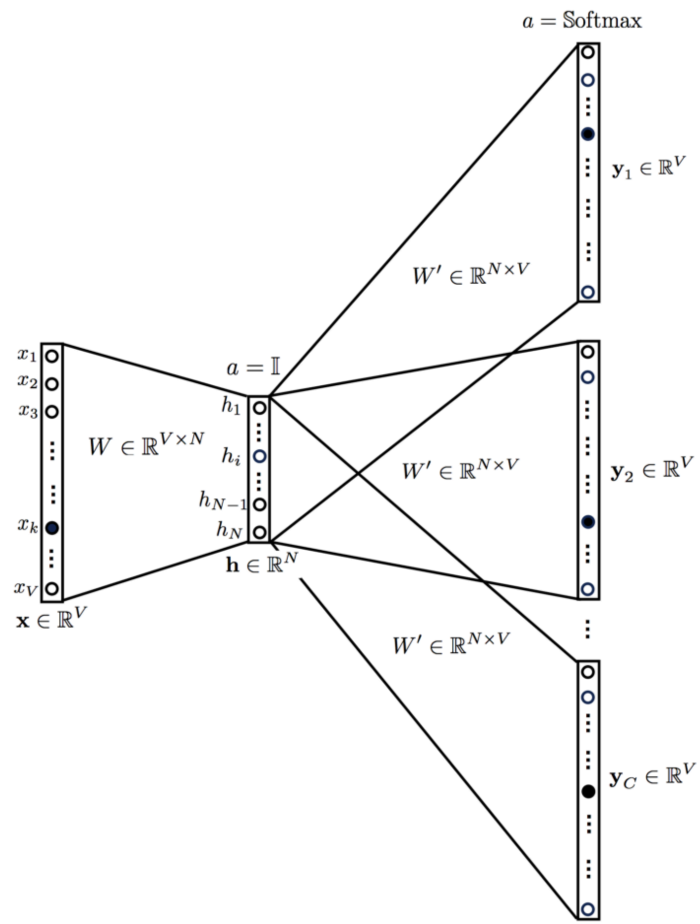
- Input:
  - Word context
  - Representation: one-hot encoding
- Output:
  - Probability distribution wrt words
  - Size: vocabulary
- Hidden layer: embedding



Img-Source: http://arxiv.org/pdf/1301.3781.pdf

Prediction Based Models: Skip-gram

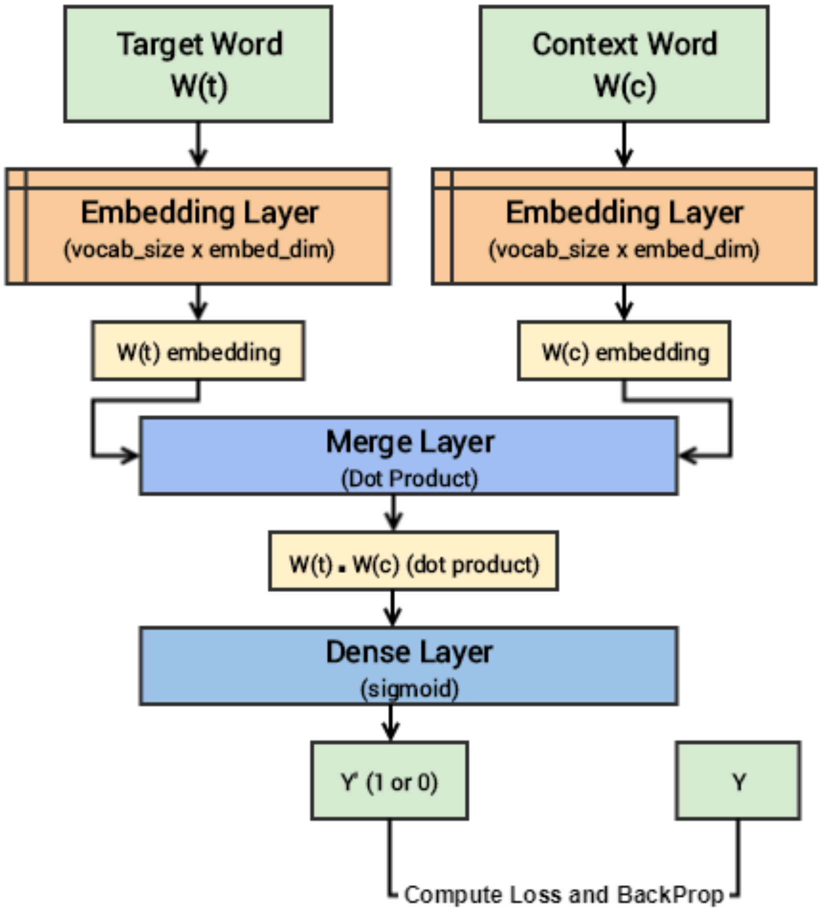
- Predicts the context given a word
- Input:
  - Word
  - Representation: one-hot encoding
- Output:
  - N Probability distributions
  - Size: context size (N) X vocabulary
- Hidden layer: embedding



Img-Source: http://arxiv.org/pdf/1301.3781.pdf

Skip-gram with Negative Sampling

- Previous models are very costly
- Predicts words as neighbors
- Negative examples: words that are not neighbors
- Inputs:
  - Word and its context word
  - Representation: one-hot encoding
- Output:
  - Probability of matching



Img-Source: http://arxiv.org/pdf/1301.3781.pdf

Skip-gram vs Negative Sampling

Skipgram

shalt	not	make	a	machine

input	output
make	shalt
make	not
make	a
make	machine

Negative Sampling

input word	output word	target
make	shalt	1
make	aaron	0
make	taco	0

Img-Source: https://jalammar.github.io/illustrated-word2vec/

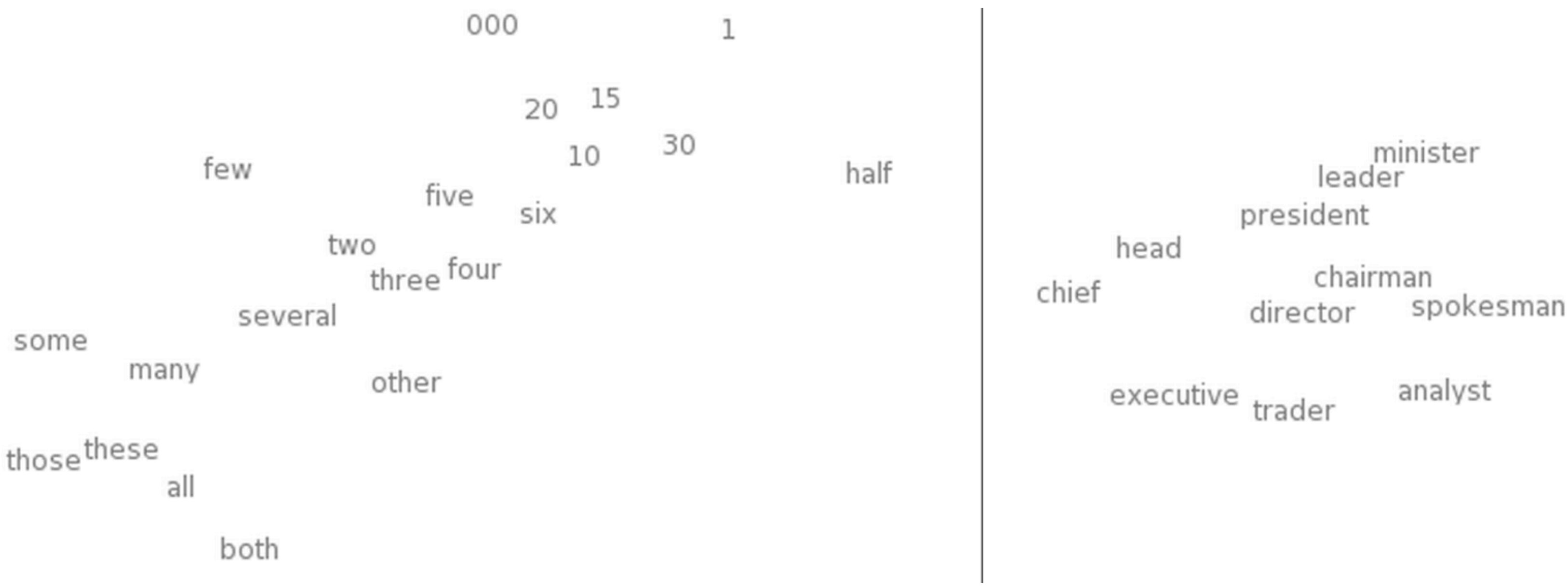
Word Embeddings

- Example: closest words

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

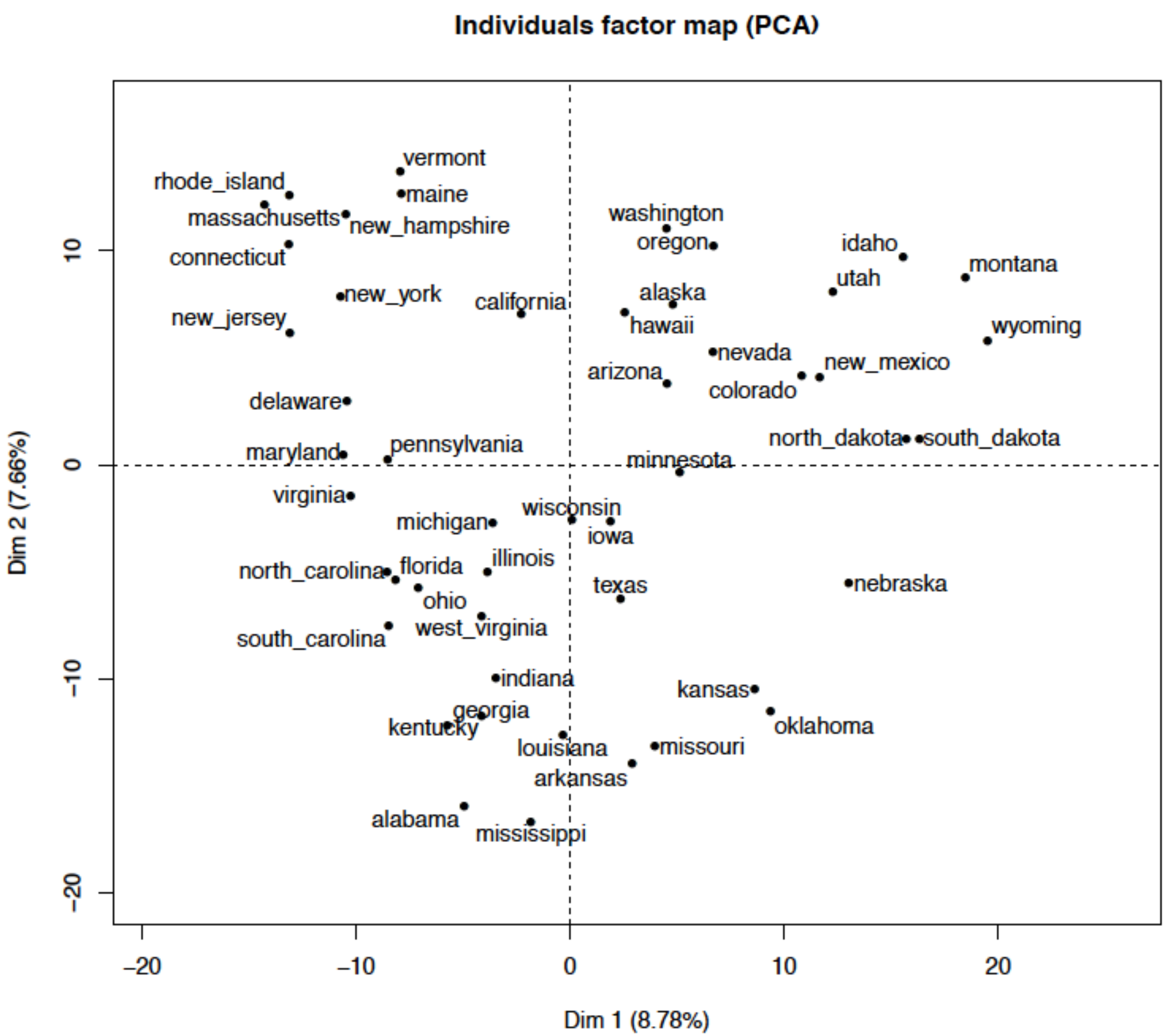
Img-Source: http://arxiv.org/pdf/1301.3781.pdf

Word Embeddings in 2D



Img-Source: http://metaoptimize.s3.amazonaws.com/cw-embeddings-ACL2010/embeddings-mostcommon.EMBEDDING_SIZE=50.png

Word Embeddings in 2D



Existent Tools

- Demo:
  - https://rare-technologies.com/word2vec-tutorial/

- Word2Vec
  - https://code.google.com/p/word2vec/

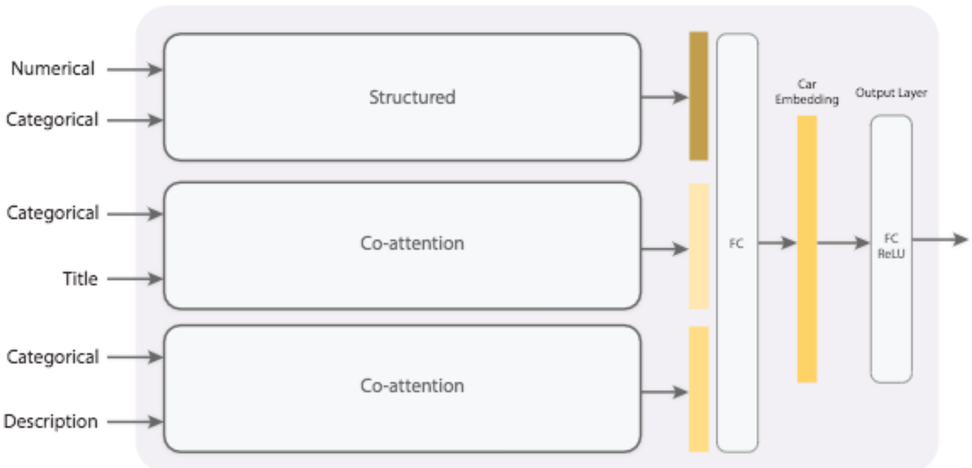
- Doc2Vec
  - Learns dense representations for phrases, sentences and documents
  - https://groups.google.com/forum/#!topic/word2vec-toolkit/Q49FIrNOQRo or Gensim

- GloVe:
  - http://nlp.stanford.edu/projects/glove/

Impact of Word Embeddings

- In most networks for text, word embeddings are the basis
- Having good word embeddings increases significantly your performance
- Which is the best is hard to tell
  – Try all available
  – Use pre-trained vectors available (you can also create yours)
- Corpus selection and tuning the parameters for the task at hand
  – E.g. for sentiment, *good* and *bad* should be far away in vector space

Embeddings for Car Price Prediction

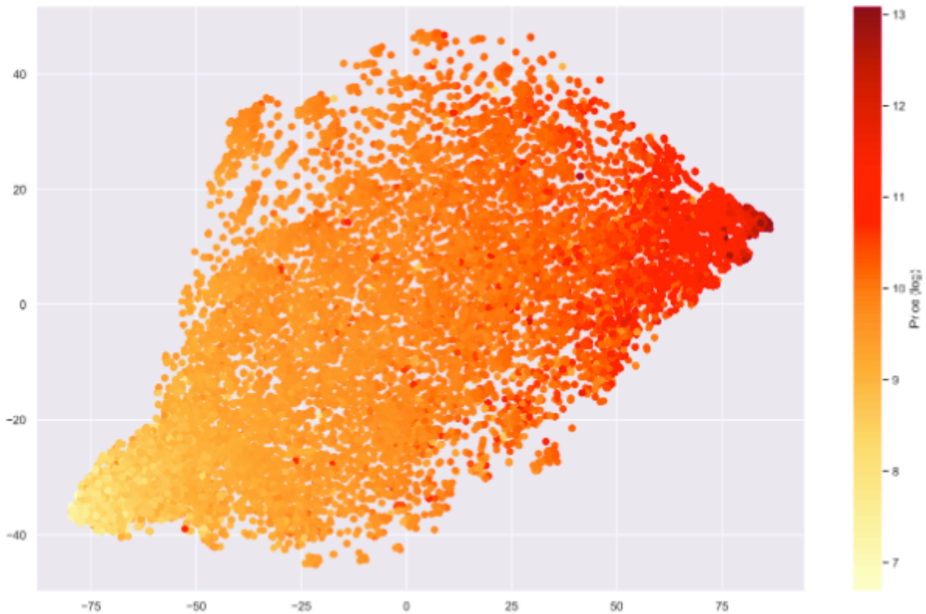


Model

Model	Features	RMSE	MALE
Linear Regression	STR	15,288	0.231
	TXT	39,801	0.217
	STR+TXT	18,555	0.171
	CE	12,926	0.125
SVR	STR	15,410	0.228
	TXT	31,140	0.249
	STR+TXT	17,672	0.185
	CE	13,076	0.127
Random Forest	STR	13,940	0.179
	TXT	16,909	0.193
	STR+TXT	15,440	0.168
	CE	**12,302**	**0.117**
LightGBM	STR	14,929	0.185
	TXT	14,601	0.177
	STR+TXT	13,560	0.130
	CE	12,305	0.120
H2O AutoML	STR	15,351	0.258
	TXT	18,644	0.283
	STR+TXT	20,341	0.299
	CE	12,439	0.118
Regression Layer	CE	13,949	0.126

Results

Embeddings

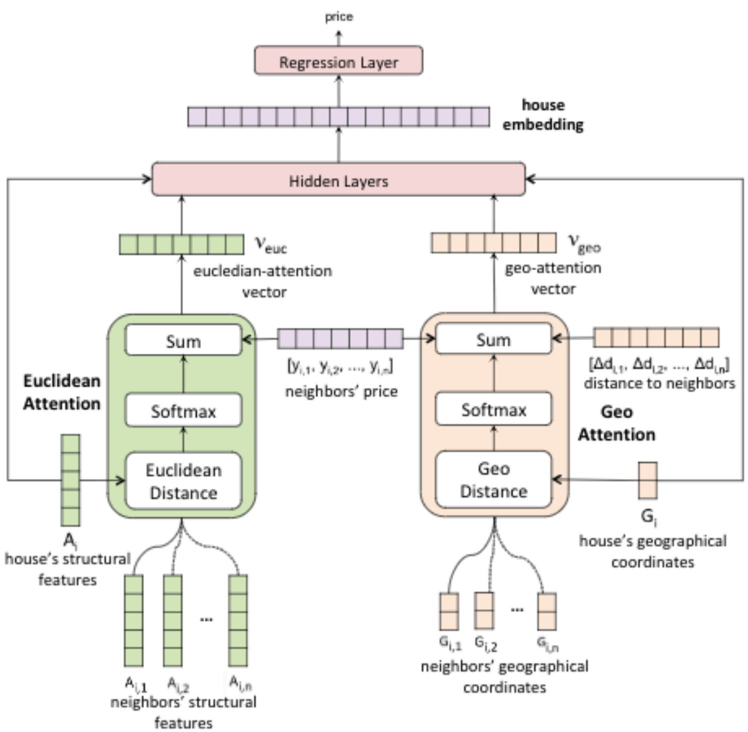


Car Embeddings

Query	Top 5	Average Price (USD)
Lamborghini	-	271,591
	Ferrari	137,567
	Aston	119,979
	Aston Martin	124,978
	Hummer	17,253
	Audi	23,796
Rolls-royce	-	116,298
	Ferrari	137,567
	Bentley	87,142
	Aston	119,979
	Jaguar	23,795
	Studebaker	30,290
Volkswagen	-	12,429
	Ford	19,112
	Volvo	20,431
	Porsche	51,561
	Cadillac	20,480
	Saturn	4,536
Saturn	-	4,536
	Saab	5,484
	Studebaker	30,290
	Honda	14,979
	Land	30,755
	Nissan	14,379

Brand Embeddings

Embeddings for House Price Prediction



Model

		SP			POA		
Mod.	Feature	MALE	RMSE	MAPE	MALE	RMSE	MAPE
LR	HA	0.266	264690	22.54	0.261	153806	22.52
	HA+HC	0.187	201275	14.73	0.241	184296	18.50
	HA+POI	0.257	257596	21.82	0.239	144529	20.77
	HE+POI	0.135	155039	9.93	0.144	94416	**10.01**
	HE	**0.135**	**154964•**	**9.92**	**0.144**	**94201•**	10.08
RF	HA	0.140	158876	10.12	0.160	100752	11.41
	HA+HC	0.159	178738	12.33	0.171	107138	12.82
	HA+POI	0.151	167782	11.52	0.159	100292	11.56
	HE+POI	0.137	**156865**	10.07	**0.146**	**95421**	**9.98**
	HE	**0.137**	157288	**10.06**	0.147	95832	10.22
LG	HA	0.146	161485	11.19	0.256	101434	12.23
	HA+HC	0.148	166866	11.37	0.201	104005	12.51
	HA+POI	0.156	169593	12.48	0.151	97068	10.48
	HE+POI	0.136	156161	**9.96**	0.147	95825	10.38
	HE	**0.136**	**156074**	10.04	**0.147**	**95756**	**10.35**
XB	HA	0.140	159018	10.41	0.154	97256	11.06
	HA+HC	0.156	172180	12.30	0.175	107515	13.57
	HA+POI	0.158	172786	12.47	0.148	**95423**	**10.49**
	HE+POI	0.137	**156685**	10.19	**0.147**	95904	10.65
	HE	**0.137**	157288	**10.07**	0.148	95798	10.58
AS	HA	0.144	161359	10.45	0.169	105156	12.28
	HA+HC	0.165	184744	12.95	0.163	101972	**9.85**
	HA+POI	0.152	167919	11.51	0.161	102113	11.73
	HE+POI	0.135	**155115**	9.92	**0.142•**	94418	9.90
	HE	**0.134•**	166866	**9.79•**	0.143	**94311**	12.22
RL	HE	0.135	155585	9.80	0.143	94492	9.58•

Results

House Embeddings

House/Apartment

Price