Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec

Christopher E. Moody

Saarland University
Seminar: Embeddings for NLP and IR
Lecturer: Cristina España i Bonet

Nora Graichen and Insa Kröger
July 3rd, 2019

Table of Contents

Motivation

What is lda2vec?

word embeddings:	topic models:
→ word2vec, skip-gram method	→ LDA Latent Dirichlet Allocation

Motivation

What is lda2vec?

word embeddings:
→ word2vec,
skip-gram method

topic models:
→ LDA
Latent Dirichlet Allocation

lda2vec:
→ topic model,
builds document representations on
top of word embeddings

What is the general goal of topic models?

produce interpretable document representations,

given a collection of unlabelled documents

→ discover topics or structure

document Farming-X:

40% topic “vegetables”

40% topic “economy”

20% topic “water culture”

Motivation

Why is lda2vec useful?

Motivation

Why is lda2vec useful?

word vectors
→ revealing relationships between words

document vectors
→ revealing topical distributions over documents

Motivation

Why is lda2vec useful?

word vectors
→ revealing relationships between words

document vectors
→ revealing topical distributions over documents

combination

Motivation

Why is lda2vec useful?

word vectors
→ revealing relationships between words

document vectors
→ revealing topical distributions over documents

combination

1. Global document themes with local word patterns

Motivation

Why is lda2vec useful?

word vectors
→ revealing relationships between words

document vectors
→ revealing topical distributions over documents

combination

1. Global document themes with local word patterns
2. Dense word vectors but sparse document vectors

Motivation

Why is lda2vec useful?

word vectors
→ revealing relationships between words

document vectors
→ revealing topical distributions over documents

combination

1. Global document themes with local word patterns
2. Dense word vectors but sparse document vectors
3. Mixture of topic models for interpretability

Table of Contents

Latent Dirichlet Allocation - LDA

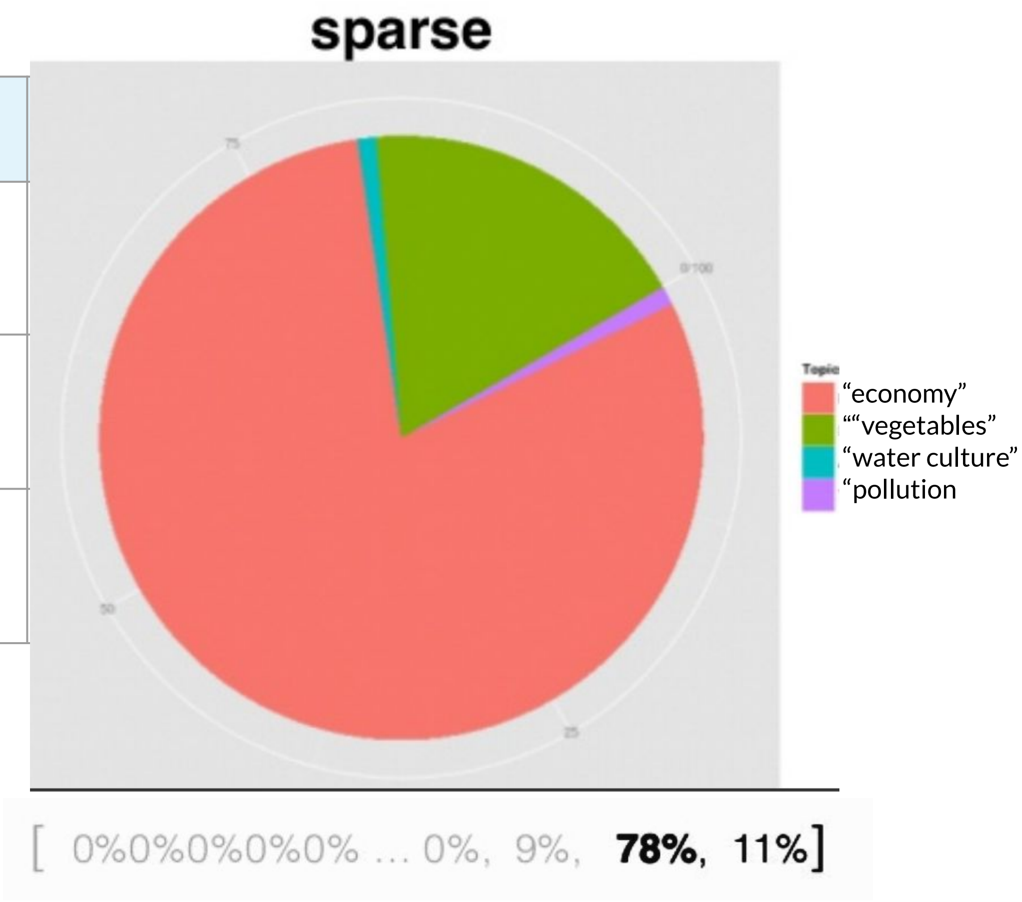
- probabilistic topic model

  → find the structure or topics in unlabelled document collection
- takes advantage of global (document level) information for predicting words
- assumption: word usage is correlated with topic occurrence
- input: number of topics that occur in the collection,

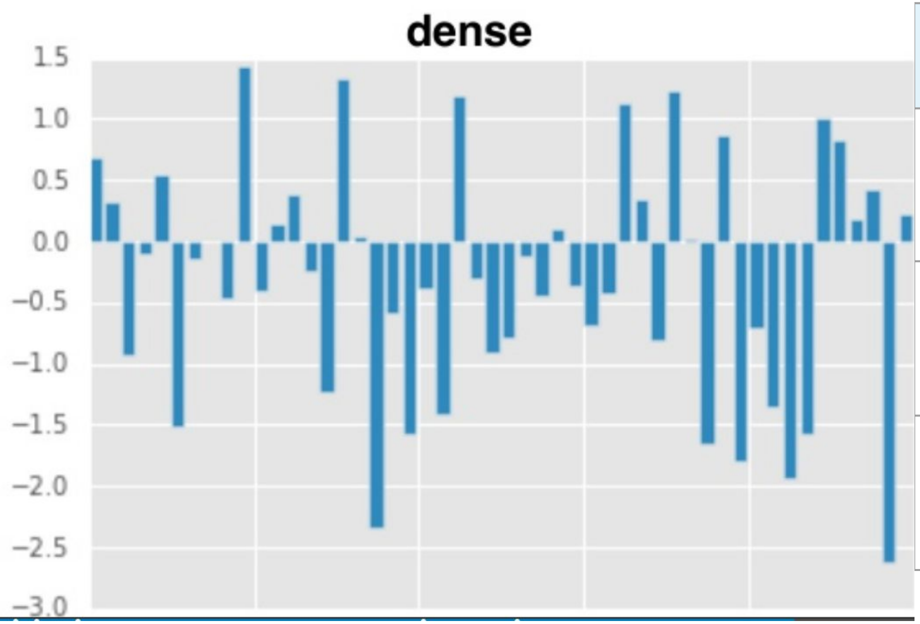
  manually assign a distinct ‘topic’ to the different topic vector

LDA and word2vec

LDA
document representations bag-of-words model
+ global + generally sparse
long distance dependencies



LDA and word2vec

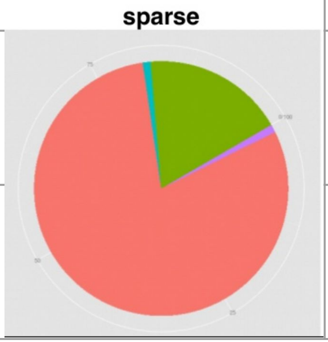


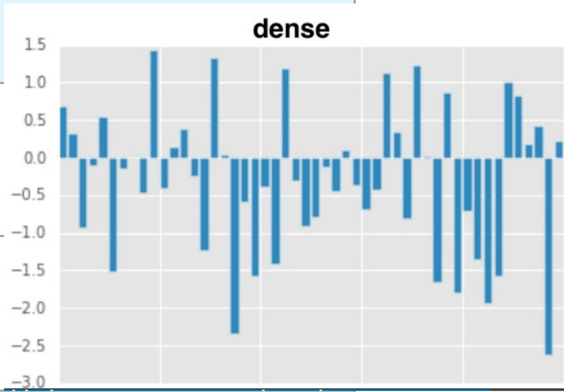
[-0.75, -1.25, -0.55, -0.27, -0.94, 0.44, 0.05, 0.31 … -0.12, +2.2]

word2vec
word representations
+ local - dense
captures rich linguistic relationship king − man + woman = queen

LDA and word2vec

LDA	**word2vec**
document representations bag-of-words model	word representations
+ global + generally sparse	+ local - dense
long distance dependencies	captures rich linguistic relationship king − man + woman = queen



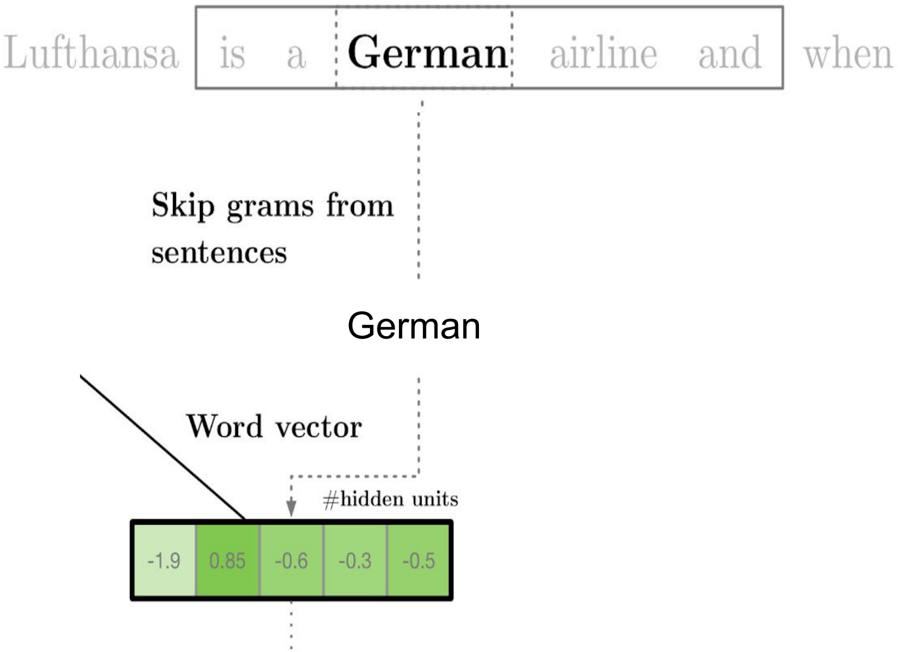


[0%0%0%0%0% … 0%, 9%, **78%**, 11%]

[-0.75, -1.25, -0.55, -0.27, -0.94, 0.44, 0.05, 0.31 … -0.12, +2.2]

Table of Contents

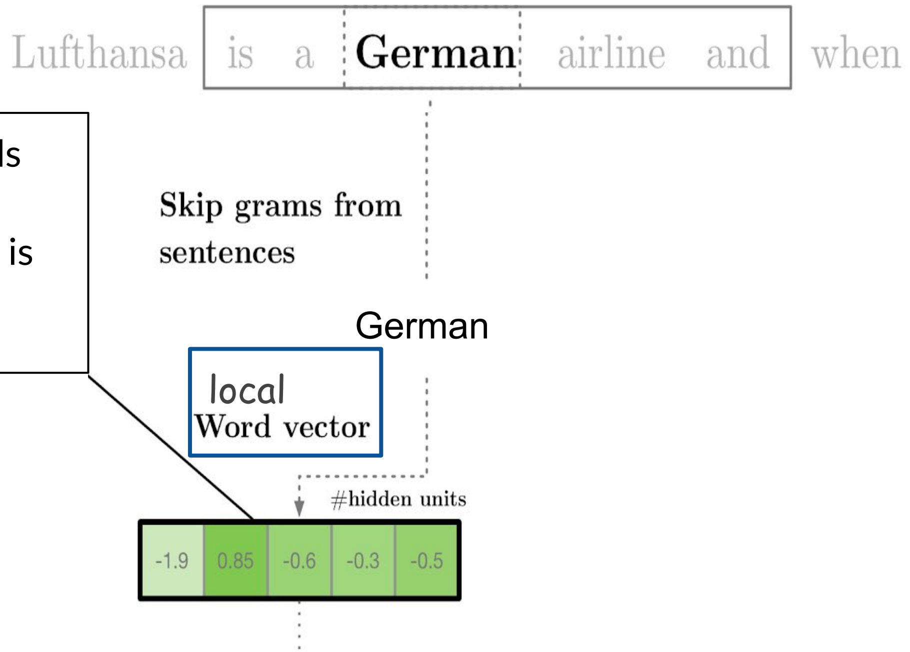
lda2vec - architecture



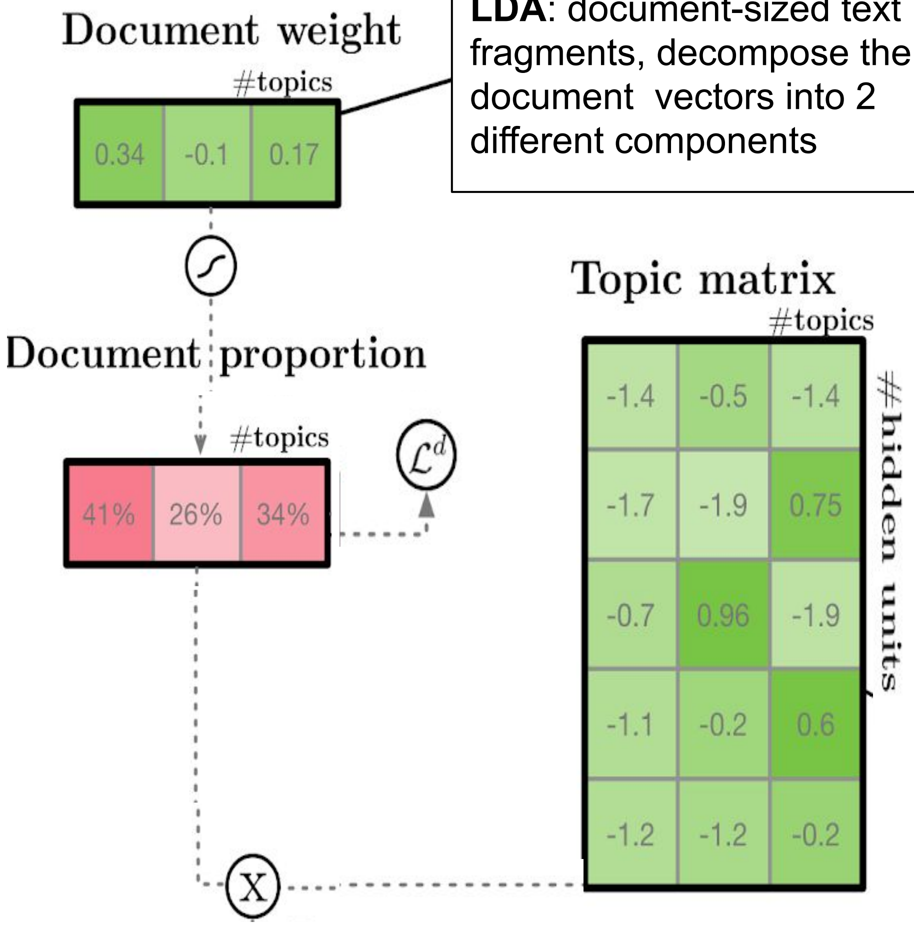
lda2vec - architecture

word vectors predict context words across different documents

⟶ document-specific information is mixed together in the word embeddings



lda2vec - architecture

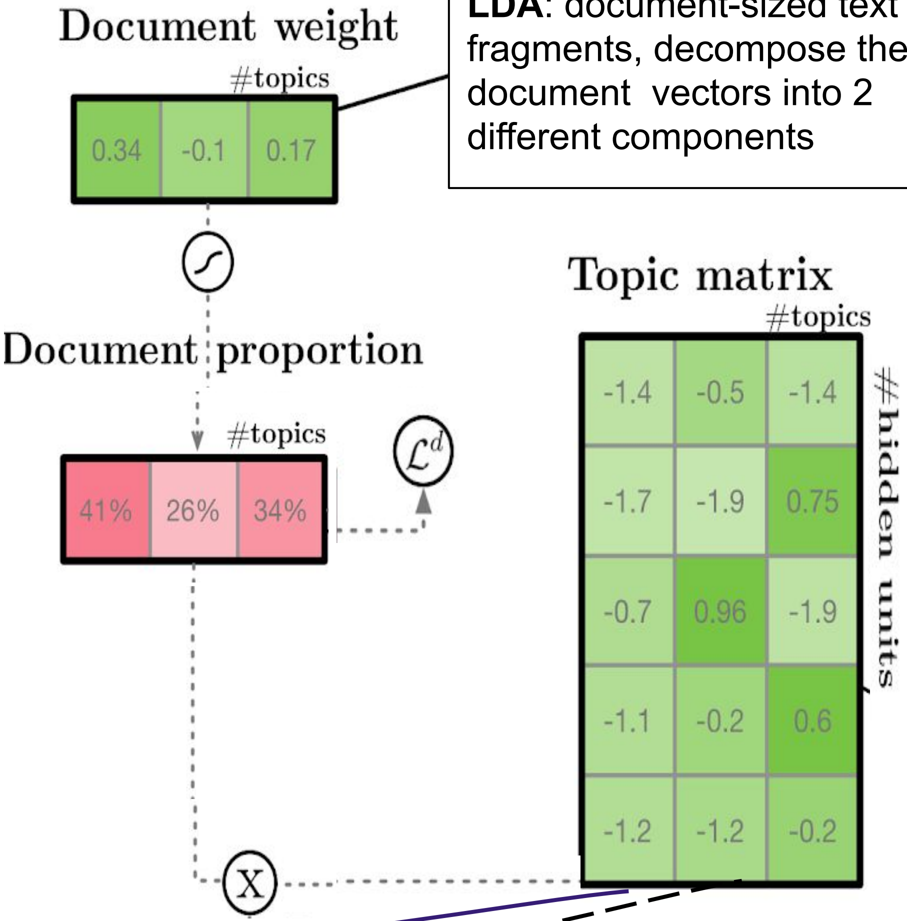


LDA: document-sized text fragments, decompose the document vectors into 2 different components

lda2vec - architecture

topic = most similar words to every topic

"Space"	"Encryption"
astronomical	encryption
Astronomy	wiretap
satellite	encrypt
planetary	escrow
telescope	Clipper

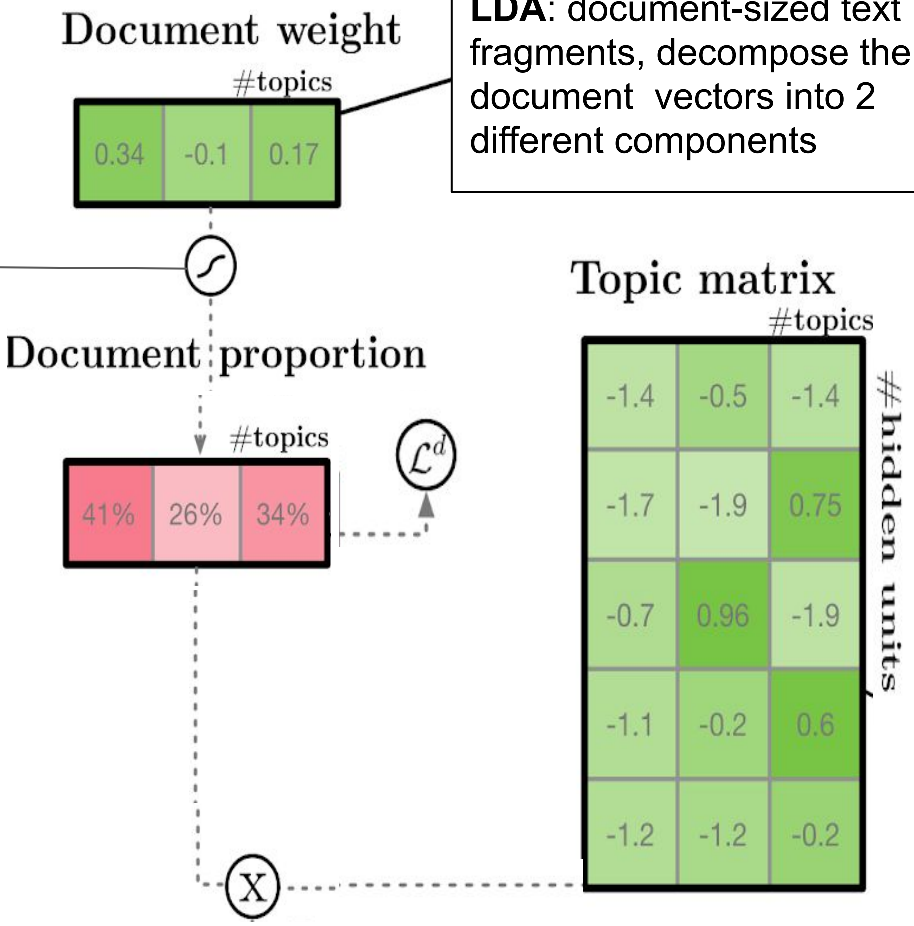


LDA: document-sized text fragments, decompose the document vectors into 2 different components

lda2vec - architecture

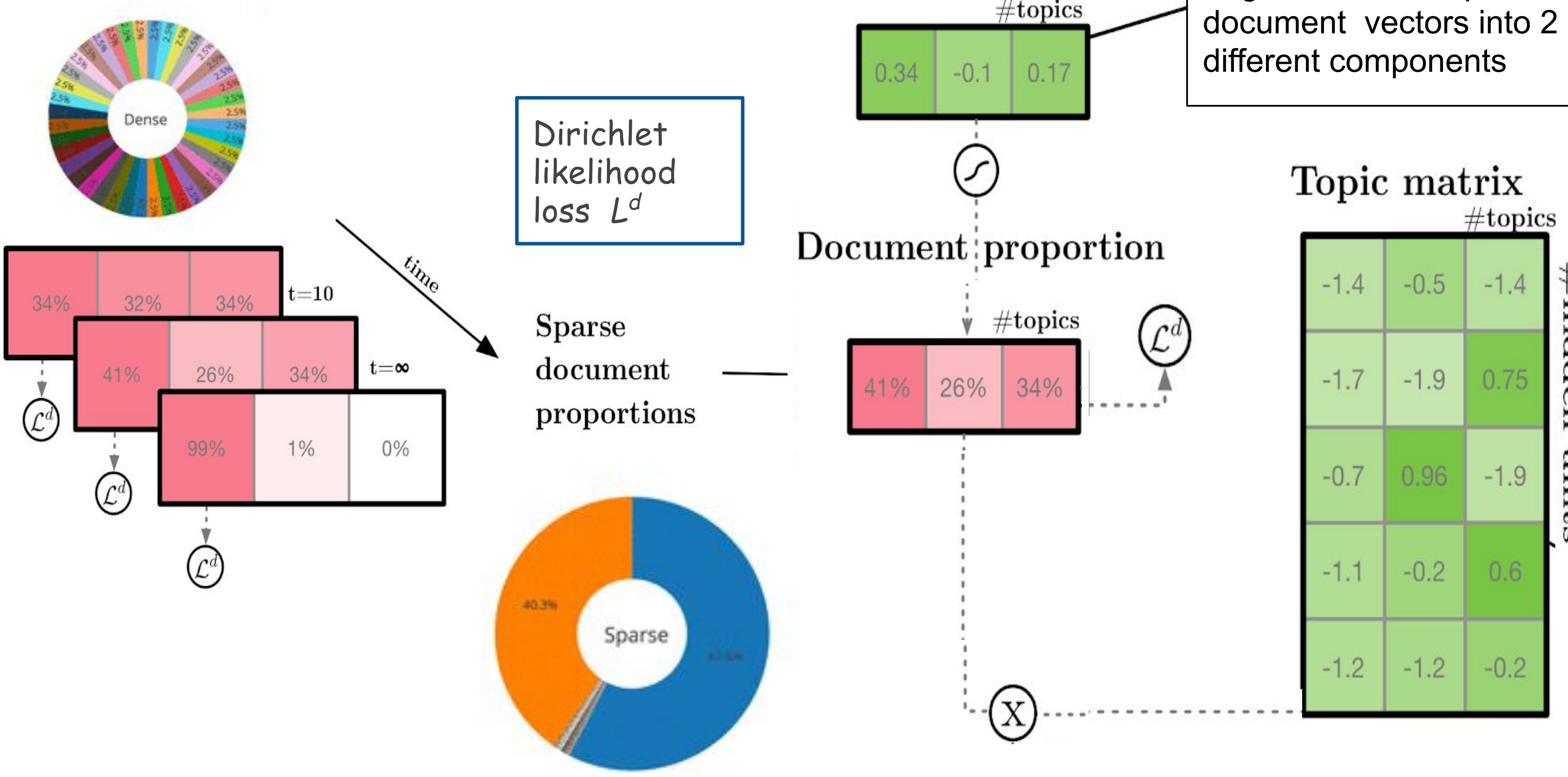
softmax

LDA: document-sized text fragments, decompose the document vectors into 2 different components



lda2vec - architecture

LDA: document-sized text fragments, decompose the document vectors into 2 different components

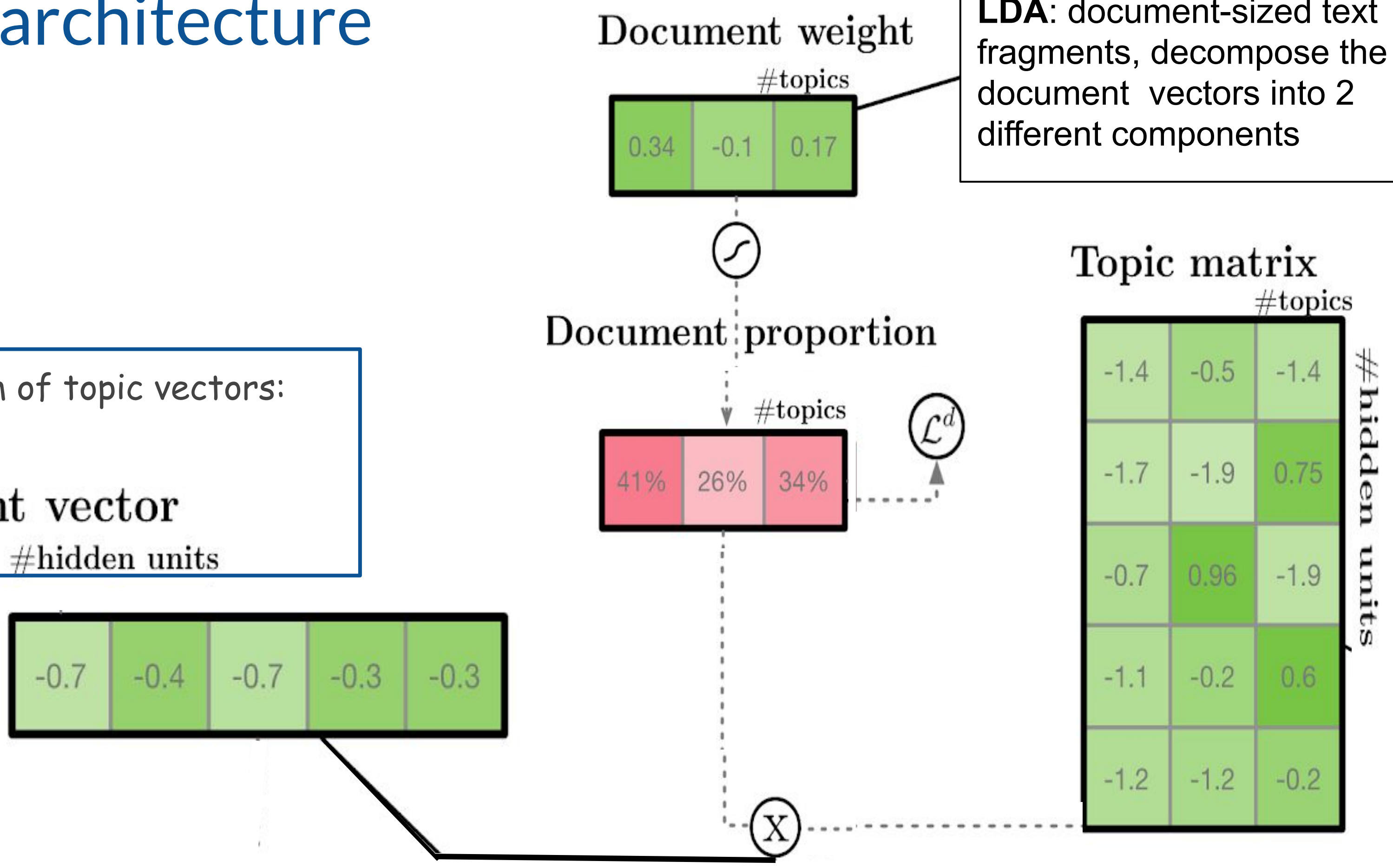


lda2vec - architecture

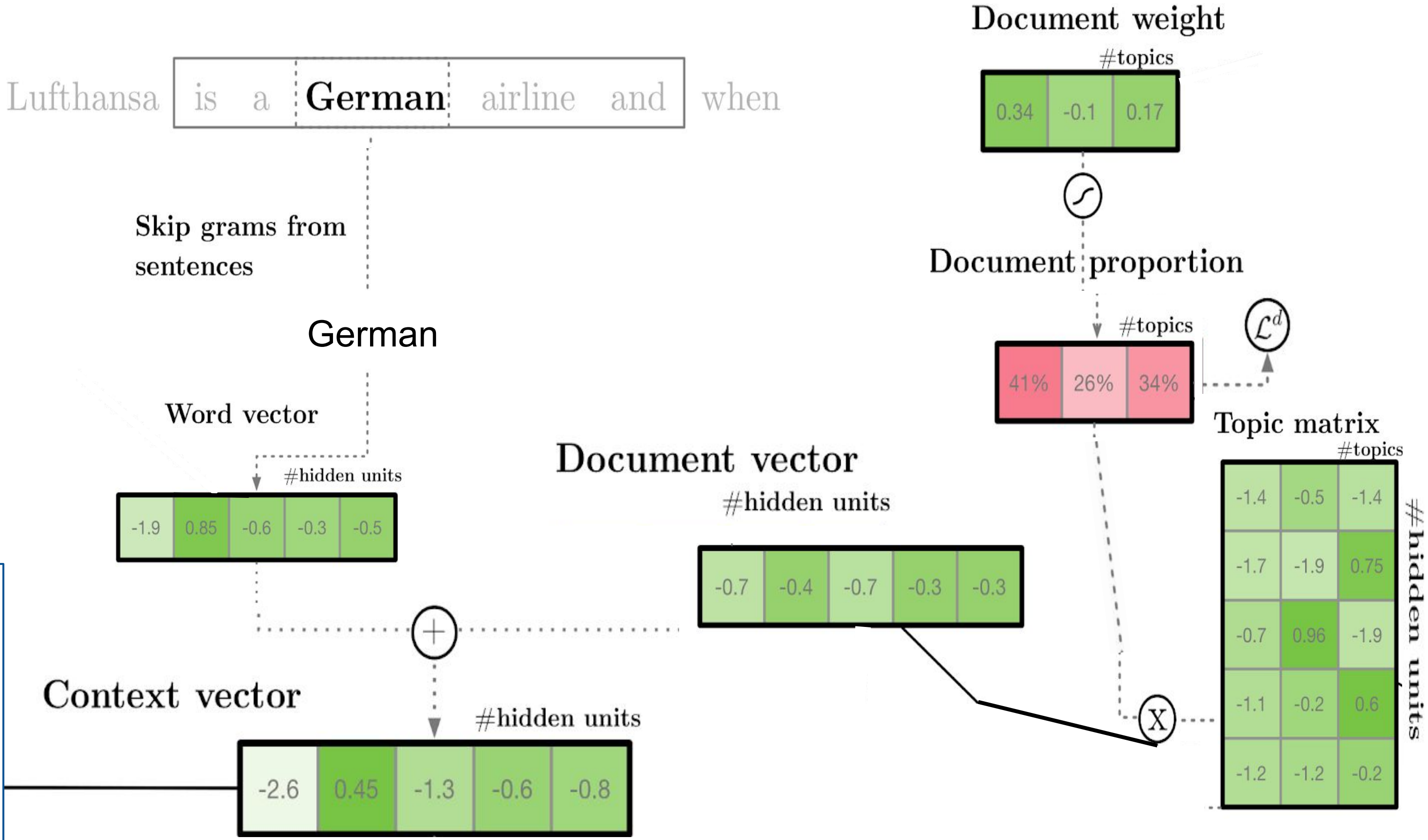
LDA: document-sized text fragments, decompose the document vectors into 2 different components

weighted sum of topic vectors:

global

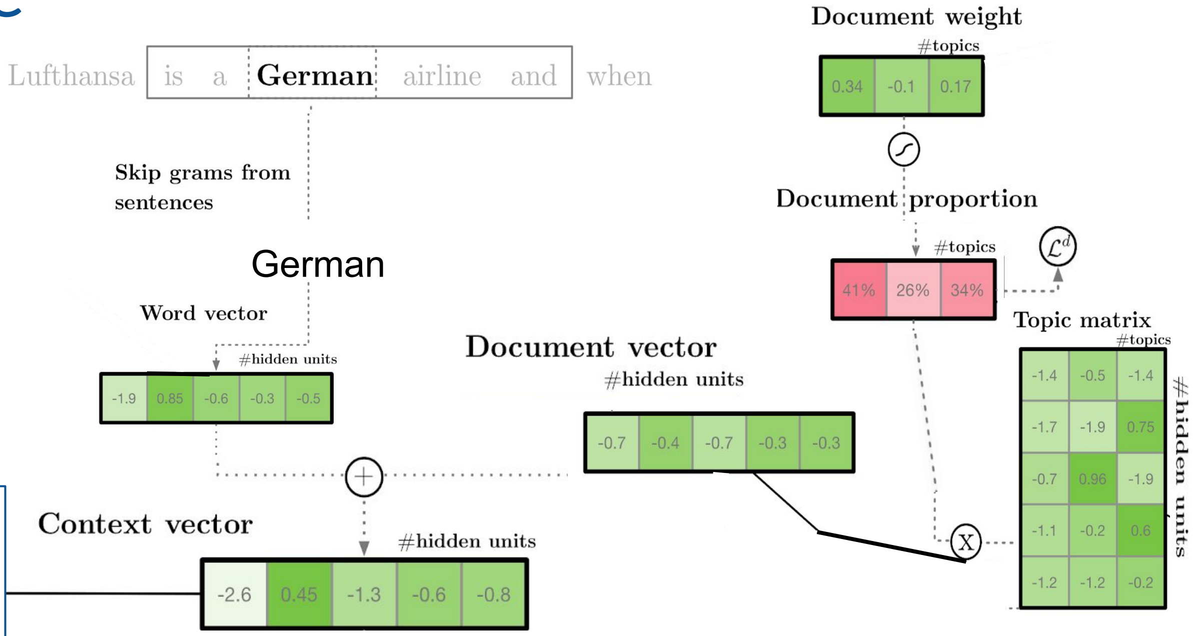


lda2vec - architecture: mix local words with global interpretable document vectors



'specialised' local and global word vectors

lda2vec - architecture

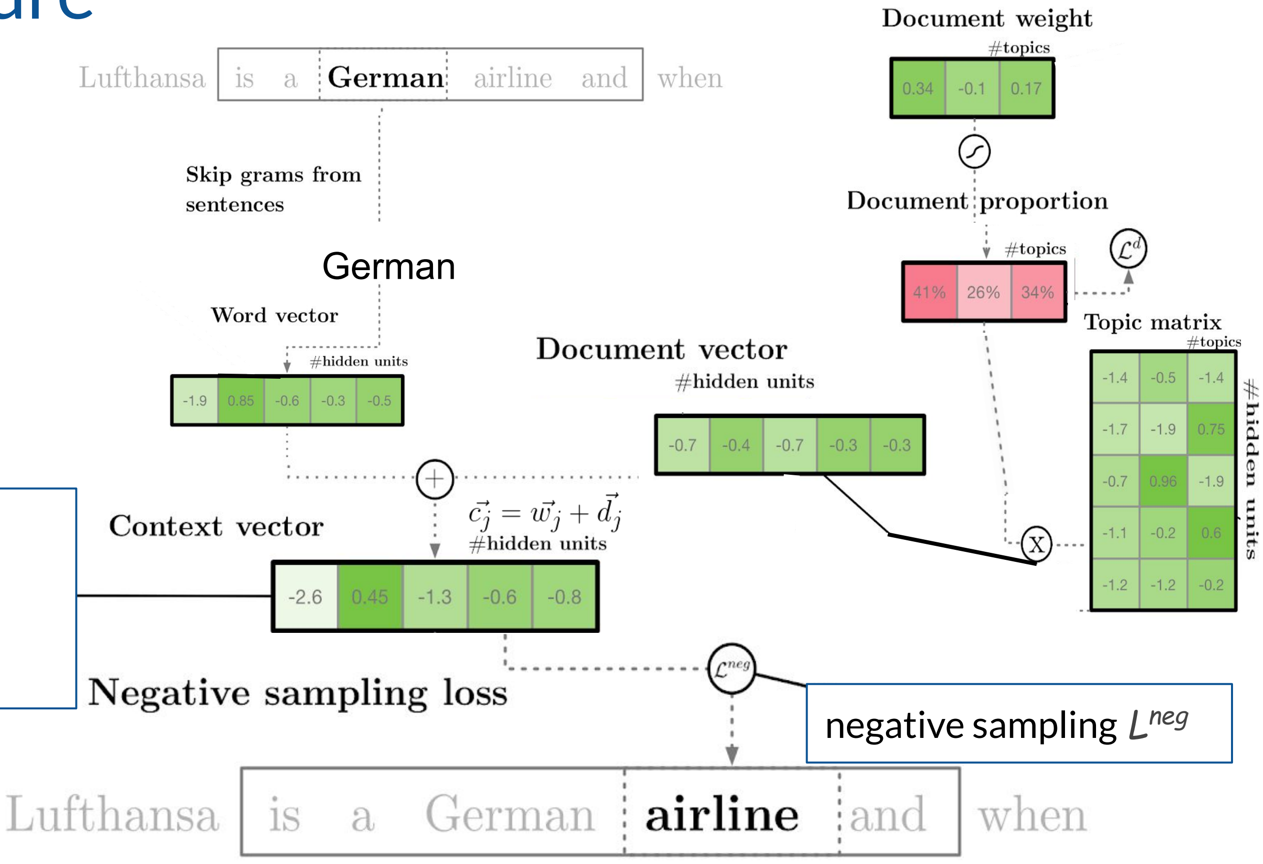


‘specialised’ word vectors:

local inter-word relationships +
document-wide relationships

$$\vec{c_j} = \vec{w_j} + \vec{d_j}$$

lda2vec - architecture



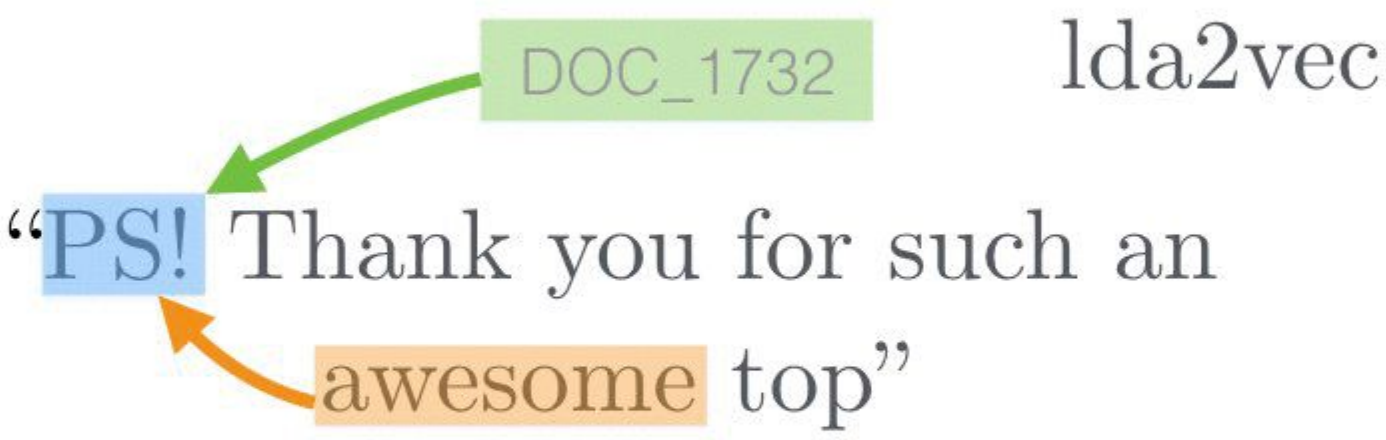
‘specialised’ word vectors:

local inter-word relationships +
document-wide relationships

negative sampling L^{neg}

Table of Contents

lda2vec



- simultaneous global and local prediction
- local features → improve predicting neighbouring words
- global features → capture themes across sentences and documents

lda2vec - loss function

$$L = L^d + \sum_{ij} L_{ij}^{neg}$$

lda2vec - loss function

$$L = \boxed{L^{d}} + \sum_{ij} L_{ij}^{neg}$$

LDA loss

lda2vec - loss function

$$L = \boxed{L^d} + \sum_{ij} L_{ij}^{neg}$$

$$L^d = \lambda \sum_{jk} (\alpha - 1) \; log \; p_{jk}$$

Dirichlet likelihood loss

likelihood of *document j* in *topic k*

lda2vec - loss function

$$L = \boxed{L^d} + \sum_{ij} L_{ij}^{neg}$$

$$L^d = \lambda \sum_{jk} (\alpha - 1)\ log\ \boxed{p_{jk}}$$

document weights

weights are optimized with respect to Dirichlet likelihood

lda2vec - loss function

$$L = \boxed{L^d} + \sum_{ij} L_{ij}^{neg}$$

$$L^d = \lambda \sum_{jk} (\boxed{\alpha} - 1) \; log \; p_{jk}$$

document proportions parameter

$\alpha > 1 \rightarrow$ homogenous

$\alpha < 1 \rightarrow$ sparse

lda2vec - loss function

$$L = \boxed{L^d} + \sum_{ij} L_{ij}^{neg}$$

$$L^d = \lambda \sum_{jk} (\boxed{\alpha} - 1) \ log \ p_{jk}$$

document proportions parameter

$\alpha > 1 \rightarrow$ homogenous

$\alpha < 1 \rightarrow$ sparse

$\alpha = n^{-1} \rightarrow$ sparse memberships over *n* topics

lda2vec - loss function

$$L = \boxed{L^d} + \sum_{ij} L_{ij}^{neg}$$

$$L^d = \boxed{\lambda} \sum_{jk} (\alpha - 1)\ log\ p_{jk}$$

tuning parameter λ
adjusts relevance of given word in topic

lda2vec - loss function

$$L = L^d + \boxed{\sum_{ij} L_{ij}^{neg}}$$

word2vec loss

lda2vec - loss function

$$L = L^d + \boxed{\sum_{ij} L_{ij}^{neg}}$$

$$L_{ij}^{neg} = log\ \sigma(\vec{c_j} \cdot \vec{w_i}) + \sum_{l=0}^{n} log\ \sigma(-\vec{c_j} \cdot \vec{w_l})$$

context vectors are combination of word and document vectors

$$\vec{c_j} = \vec{w_j} + \vec{d_j}$$

lda2vec - loss function

$$\vec{c_j} = \vec{w_j} + \vec{d_j}$$

$$L = L^d + \boxed{\sum_{ij} L_{ij}^{neg}}$$

$$L_{ij}^{neg} = log\ \sigma(\vec{c_j} \cdot \vec{w_i}) + \sum_{l=0}^{n} log\ \sigma(-\vec{c_j} \cdot \vec{w_l})$$

sum of weighted topic vectors

$$\vec{d_j} = \boxed{a_{j0} \cdot \vec{t_0} + a_{j1} \cdot \vec{t_1} + ...}$$

lda2vec - loss function

$$L = L^d + \boxed{\sum_{ij} L_{ij}^{neg}}$$

$$\vec{c_j} = \vec{w_j} + \vec{d_j}$$

$$\vec{d_j} = a_{j0} \cdot \vec{t_0} + a_{j1} \cdot \vec{t_1} + \ldots$$

$$L_{ij}^{neg} = log\ \sigma(\vec{c_j} \cdot \vec{w_i}) + \sum_{l=0}^{n} log\ \sigma\boxed{(-\vec{c_j} \cdot \vec{w_l})}$$

randomly sampled negative words and contexts

Table of Contents

Data - Twenty Newsgroups

- Twenty Newsgroups:
  - around 9’000 unique tokens in 11’300 documents
  - initialized with pretrained vectors
  - n = 20 topics, negative sampling β = 0.75

# of topics	β	Topic Coherences
20	0.75	**0.567**
30	0.75	0.555
40	0.75	0.553
50	0.75	0.547
20	1.00	0.563
30	1.00	0.564
40	1.00	0.552
50	1.00	0.558

Average topic coherences. Topic coherence has been demonstrated to correlate with human evaluations of topic models (Röder et al., 2015). The number of topics chosen is given, as well as the negative sampling exponent parameter β.

Data - Twenty Newsgroups

- Twenty Newsgroups:
  - around 9’000 unique tokens in 11’300 documents
  - initialized with pretrained vectors
  - n = 20 topics, negative sampling β = 0.75

# of topics	β	Topic Coherences
20	0.75	**0.567**
30	0.75	0.555
40	0.75	0.553
50	0.75	0.547
20	1.00	0.563
30	1.00	0.564
40	1.00	0.552
50	1.00	0.558

Average topic coherences. Topic coherence has been demonstrated to correlate with human evaluations of topic models (Röder et al., 2015). The number of topics chosen is given, as well as the negative sampling exponent parameter β.

Topic Label	“Space”	“Encryption”	“X Windows”	“Middle East”
Top tokens	astronomical Astronomy satellite planetary telescope	encryption wiretap encrypt escrow Clipper	mydisplay xlib window cursor pixmap	Armenian Lebanese Muslim Turk sy
Topic Coherence	0.712	0.675	0.472	0.615

Topic coherence discovered by lda2vec in the Twenty Newsgroups dataset.
first row: inferred topic label, below: tokens with highest similarity to the topic
Corpus contains corresponding newsgroups: sci.space, sci.crypt, comp.windows.x and talk.politics.mideast.

Data - Hacker News Comments

- Hacker News Comments corpus:
    - around 110 thousand unique tokens in 66 thousand documents
    - no pretrained vectors but random initialization
    - 256 hidden units
    - $n = 40$ topics, negative sampling power $\beta = 0.75$

Data - Hacker News Comments

“Housing Issues”	“Internet Portals”	“Bitcoin”	“Compensation”	“Gadget Hardware”
more housing	DDG.	btc	current salary	the Surface Pro
basic income	Bing	bitcoins	more equity	HDMI
new housing	Google+	Mt. Gox	vesting	glossy screens
house prices	DDG	MtGox	equity	Mac Pro
short-term rentals	iGoogle	Gox	vesting schedule	Thunderbolt

Topics discovered by lda2vec in the Hacker News comments dataset.
first row: inferred topic label
Tokens formed from noun phrases to capture the unique vocabulary of this specialized corpus

Data - Hacker News Comments

"Housing Issues"	"Internet Portals"	"Bitcoin"	"Compensation"	"Gadget Hardware"
more housing	DDG.	btc	current salary	the Surface Pro
basic income	Bing	bitcoins	more equity	HDMI
new housing	Google+	Mt. Gox	vesting	glossy screens
house prices	DDG	MtGox	equity	Mac Pro
short-term rentals	iGoogle	Gox	vesting schedule	Thunderbolt

Topics discovered by lda2vec in the HNC dataset. first row: inferred topic label. Tokens formed from noun phrases to capture the unique vocabulary of this specialized corpus

Artificial sweeteners	Black holes	Comic Sans	Functional Programming	San Francisco
glucose	particles	typeface	FP	New York
fructose	consciousness	Arial	Haskell	Palo Alto
HFCS	galaxies	Helvetica	OOP	NYC
sugars	quantum mechanics	Times New Roman	functional languages	New York City
sugar	universe	font	monads	SF
Soylent	dark matter	new logo	Lisp	Mountain View
paleo diet	Big Bang	Anonymous Pro	Clojure	Seattle
diet	planets	Baskerville	category theory	Los Angeles
carbohydrates	entanglement	serif font	OO	Boston

Given an example **token** in the top row, the **most similar words** available in the HNC corpus are reported

Data - Hacker News Comments

"Housing Issues"	"Internet Portals"	"Bitcoin"	"Compensation"	"Gadget Hardware"
more housing	DDG.	btc	current salary	the Surface Pro
basic income	Bing	bitcoins	more equity	HDMI
new housing	Google+	Mt. Gox	vesting	glossy screens
house prices	DDG	MtGox	equity	Mac Pro
short-term rentals	iGoogle	Gox	vesting schedule	Thunderbolt

Topics discovered by lda2vec in the HNC dataset. first row: inferred topic label. Tokens formed from noun phrases to capture the unique vocabulary of this specialized corpus

Artificial sweeteners	Black holes	Comic Sans	Functional Programming	San Francisco
glucose	particles	typeface	FP	New York
fructose	consciousness	Arial	Haskell	Palo Alto
HFCS	galaxies	Helvetica	OOP	NYC
sugars	quantum mechanics	Times New Roman	functional languages	New York City
sugar	universe	font	monads	SF
Soylent	dark matter	new logo	Lisp	Mountain View
paleo diet	Big Bang	Anonymous Pro	Clojure	Seattle
diet	planets	Baskerville	category theory	Los Angeles
carbohydrates	entanglement	serif font	OO	Boston

Given an example **token** in the top row, the **most similar words** available in the HNC corpus are reported

Query	Result
California + technology	Silicon Valley
digital + currency	Bitcoin
Javascript - browser + server	Node.js
Mark Zuckerberg - Facebook + Amazon	Jeff Bezos
NLP - text + image	computer vision
Snowden - United States + Sweden	Assange
Surface Pro - Microsoft + Amazon	Kindle

linear relationships in the HNC dataset.
first column: example input query
second column: token most similar to the input

Table of Contents

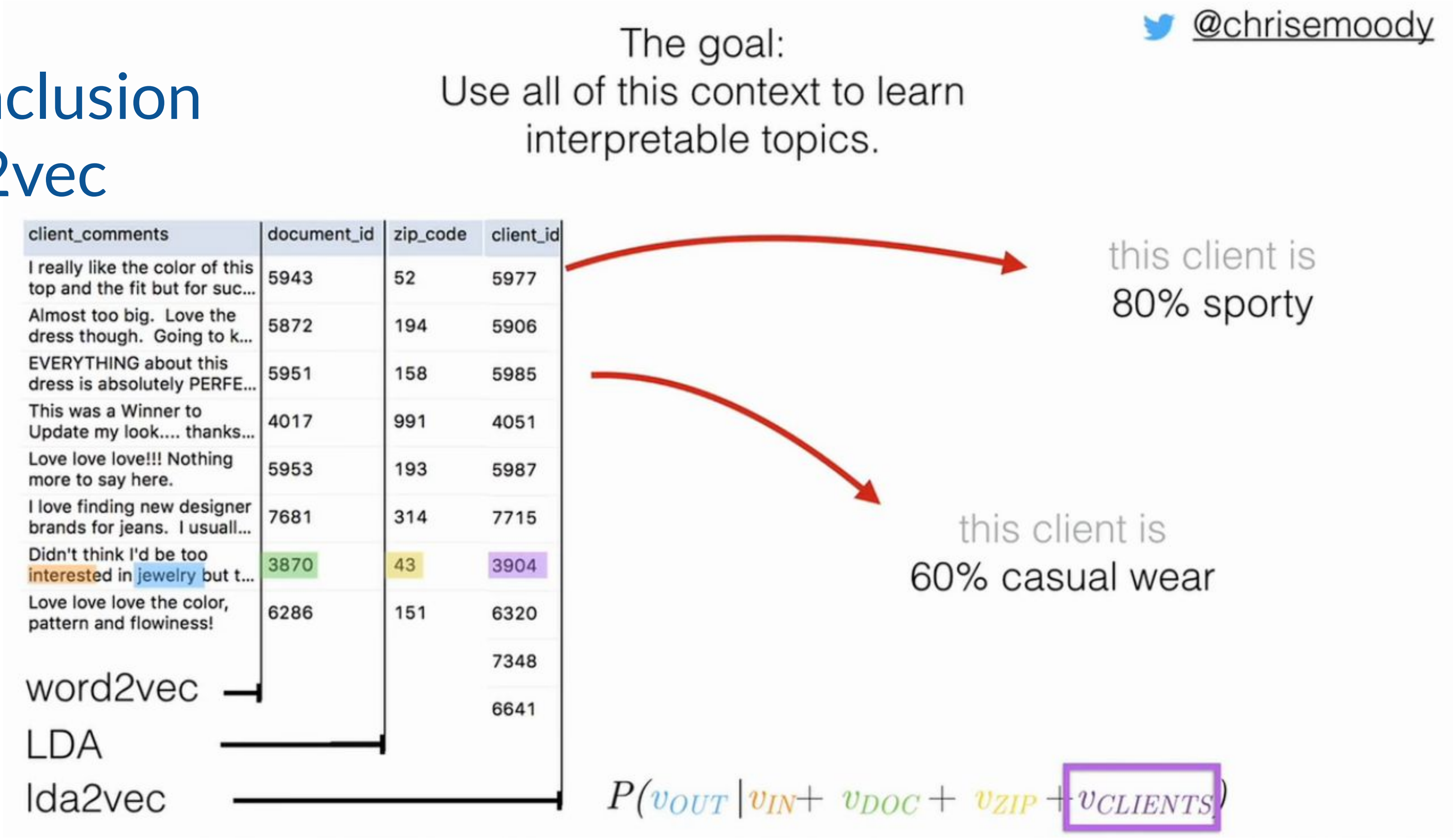
Conclusion lda2vec

- more than just a topic model
- captures semantic meaning of words

  → word, topic, document vectors are trained and embedded in a common representation space
- “human interpretable”
- sparse and interpretable document-to-topic proportions in LDA style
- includes more contexts and features than LDA

  → obtain ‘specialised’ word vectors

Conclusion
lda2vec



client_comments	document_id	zip_code	client_id
I really like the color of this top and the fit but for suc...	5943	52	5977
Almost too big. Love the dress though. Going to k...	5872	194	5906
EVERYTHING about this dress is absolutely PERFE...	5951	158	5985
This was a Winner to Update my look.... thanks...	4017	991	4051
Love love love!!! Nothing more to say here.	5953	193	5987
I love finding new designer brands for jeans. I usuall...	7681	314	7715
Didn't think I'd be too interested in jewelry but t...	3870	43	3904
Love love love the color, pattern and flowiness!	6286	151	6320
			7348
			6641

Weak points

- very experimental
- no real baseline comparison to standalone word2vec and LDA
- heavily computationally expensive
  → GPU’s are needed
- word vector nuances are compressed
  → syntactical / semantical information may get lost
- (some) tables do not contain metrics
  → which scales of similarity?

Weak points: very specific!

@chrisemoody

If you want…

human-interpretable doc topics, use **LDA**.

machine-useable word-level features, use **word2vec**.

topics over user / doc / region / etc. features, use **lda2vec**.
(and you have a GPU)

Strong points

- good documentation for starting
- may reveal trends that only word vectors cannot capture
- easily human interpretable (not just machine readable)
- sparks experimentation and new approaches

Illustration

Jupyter Notebook for illustration purposes

https://nbviewer.jupyter.org/github/cemoody/lda2vec/blob/master/examples/twenty_newsgroups/lda2vec/lda2vec.ipynb#topic=0&lambda=0.06&term=

Questions?

References

- Christopher E. Moody. (2016)
  Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec.
- Introducing our Hybrid lda2vec Algorithm:
  https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec
- Chris Moody introduces lda2vec:
  https://www.youtube.com/watch?v=eHcBeVnAiD4