A word is worth a thousand vectors

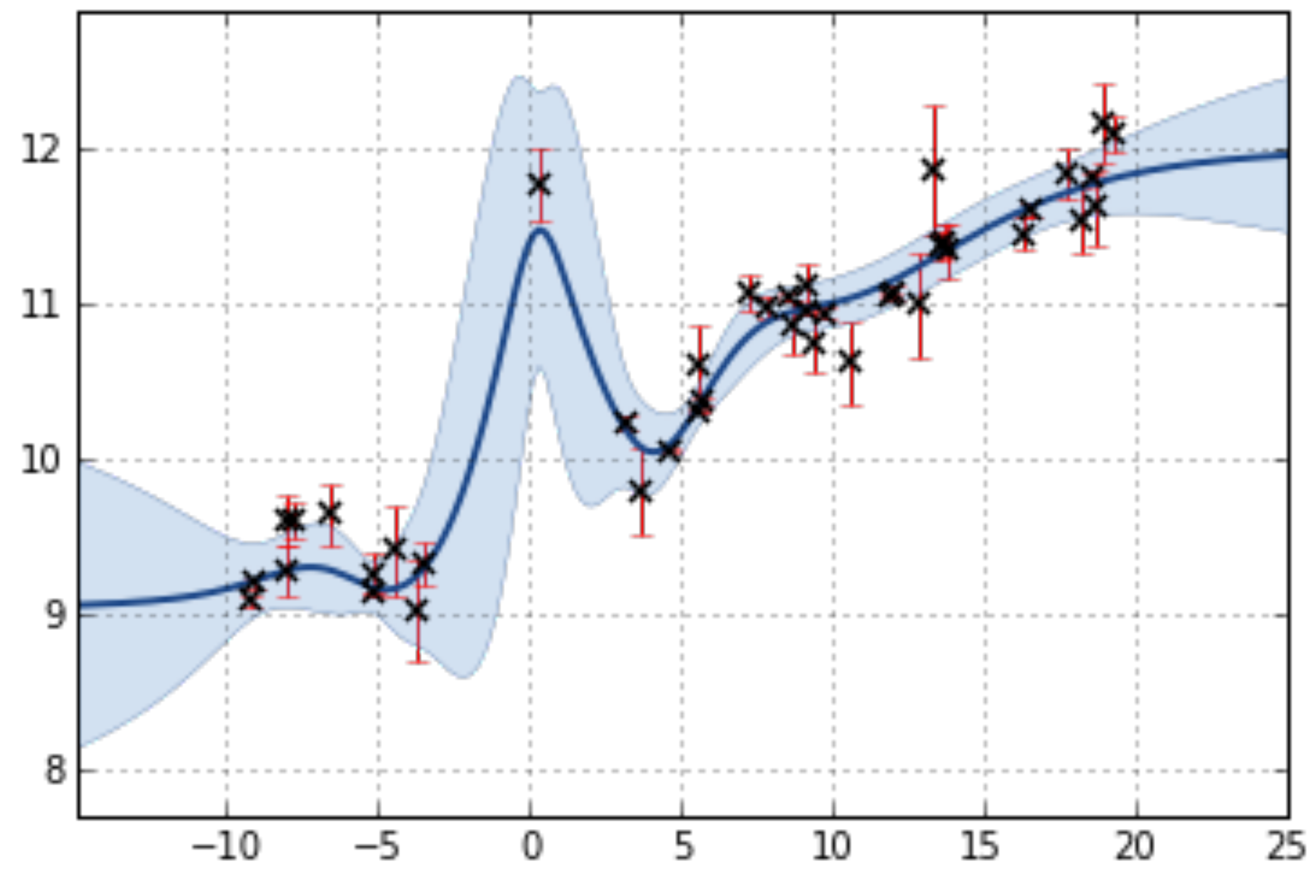
(word2vec, lda, and introducing **lda2vec**)

Christopher Moody
@ Stitch Fix

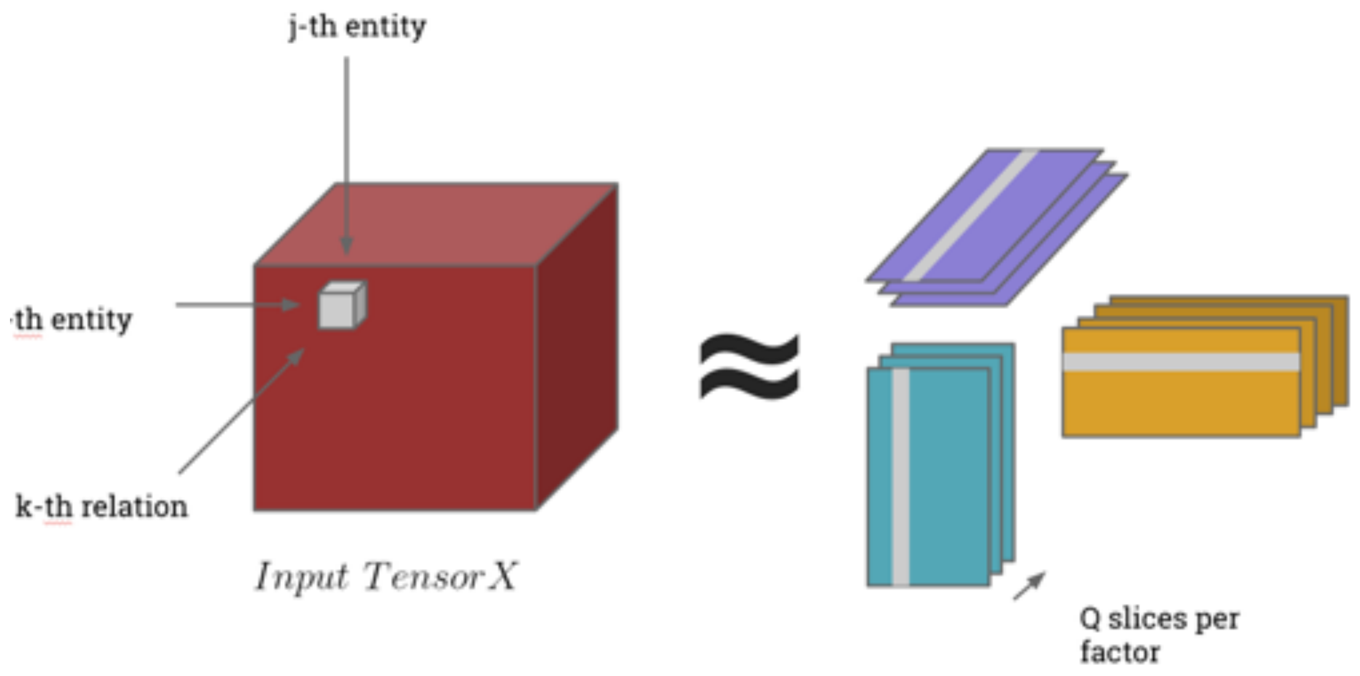
About

- @chrisemoody
- Caltech Physics
- PhD. in astrostats supercomputing
- sklearn t-SNE contributor
- Data Labs at Stitch Fix
- github.com/cemoody

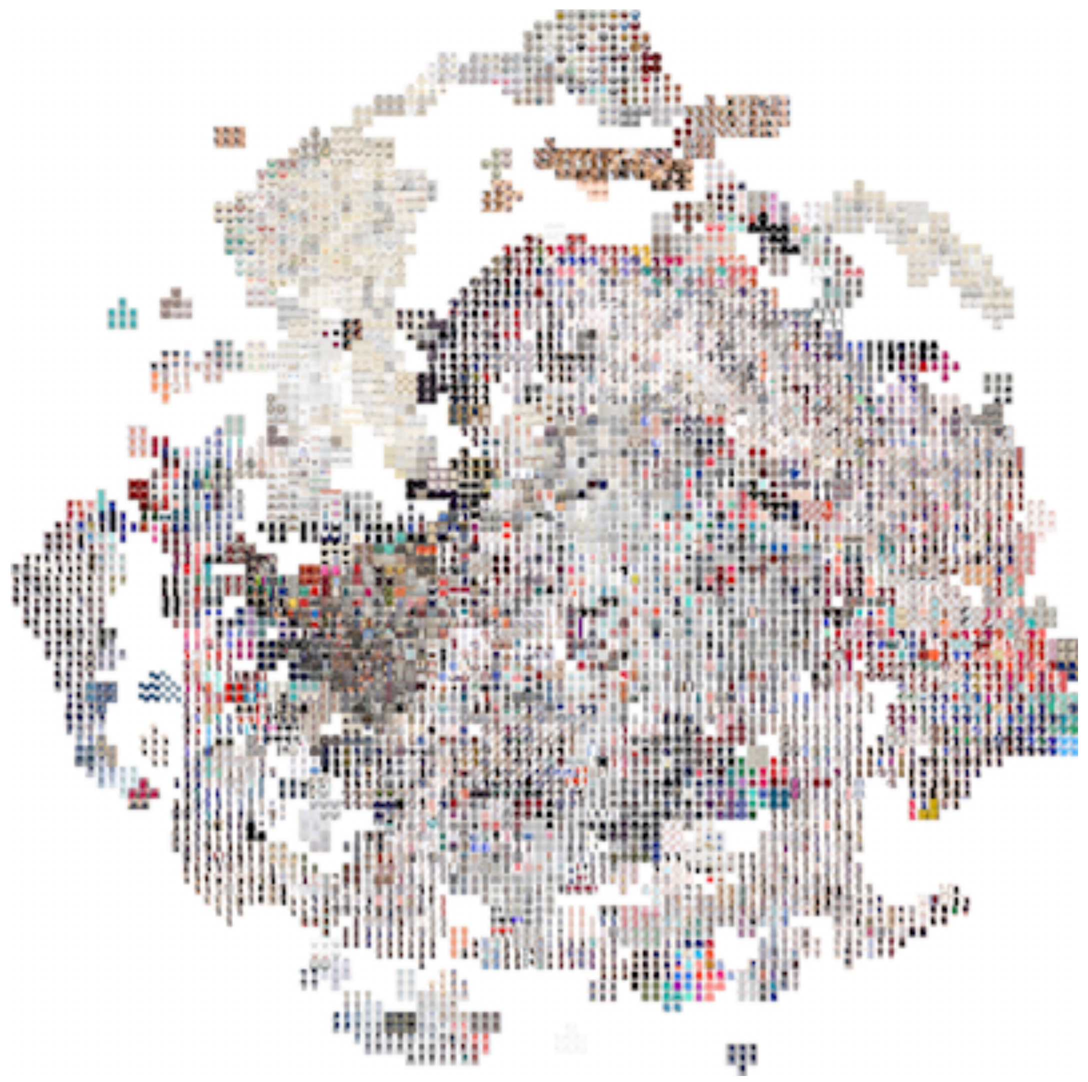
Gaussian Processes



Tensor Decomposition



t-SNE



chainer
deep learning

Chainer

Credit

Large swathes of this talk are from previous presentations by:

- Tomas Mikolov
- David Blei
- Christopher Olah
- Radim Rehurek
- Omer Levy & Yoav Goldberg
- Richard Socher
- Xin Rong
- Tim Hopper

1 **word2vec**

2 **lda**

3 **lda2vec**

word2vec

1. $king$ - $man + woman = queen$
2. Huge splash in NLP world
3. Learns from raw text
4. Pretty simple algorithm
5. Comes pretrained

word2vec

1. Set up an objective function
2. Randomly initialize vectors
3. Do gradient descent

word2vec: learn word vector v_{in}
from it's surrounding context

$$v_{in}$$

“The fox jumped **over** the lazy dog”

Maximize the likelihood of seeing the words given the word **over**.

$$P(the|over)$$
$$P(fox|over)$$
$$P(jumped|over)$$
$$P(the|over)$$
$$P(lazy|over)$$
$$P(dog|over)$$

...instead of maximizing the likelihood of co-occurrence counts.

What should this be?

$$P(fox|over)$$

Should depend on the word vectors.

$$\begin{tikzcd}P(fox|over) \arrow[d] \\ P(v_{fox}|v_{over})\end{tikzcd}$$

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

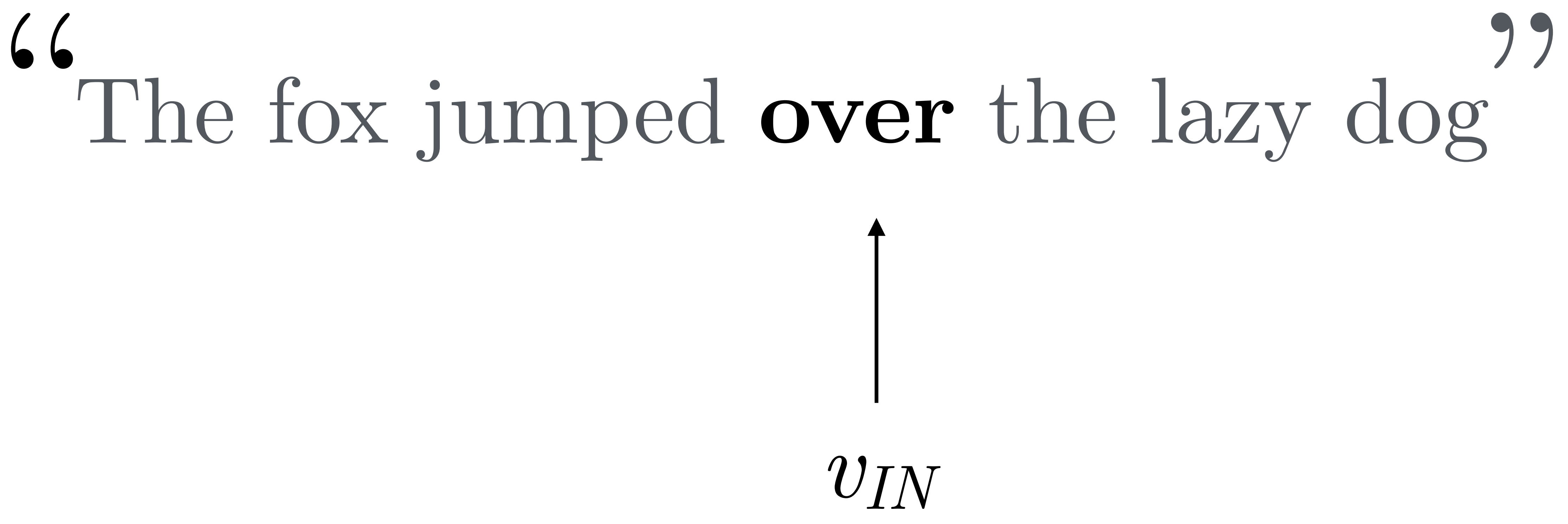
$$P(v_{OUT}|v_{IN})$$

“The fox jumped **over** the lazy dog”

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



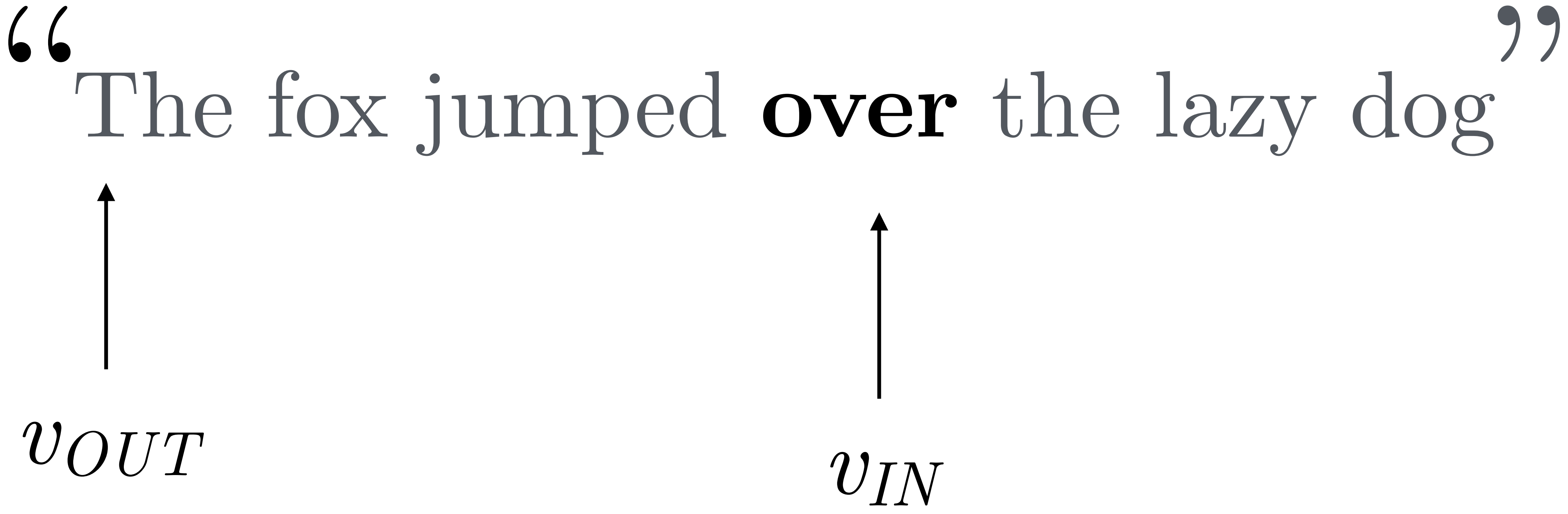
"The fox jumped **over** the lazy dog"

v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



“The fox jumped **over** the lazy dog”

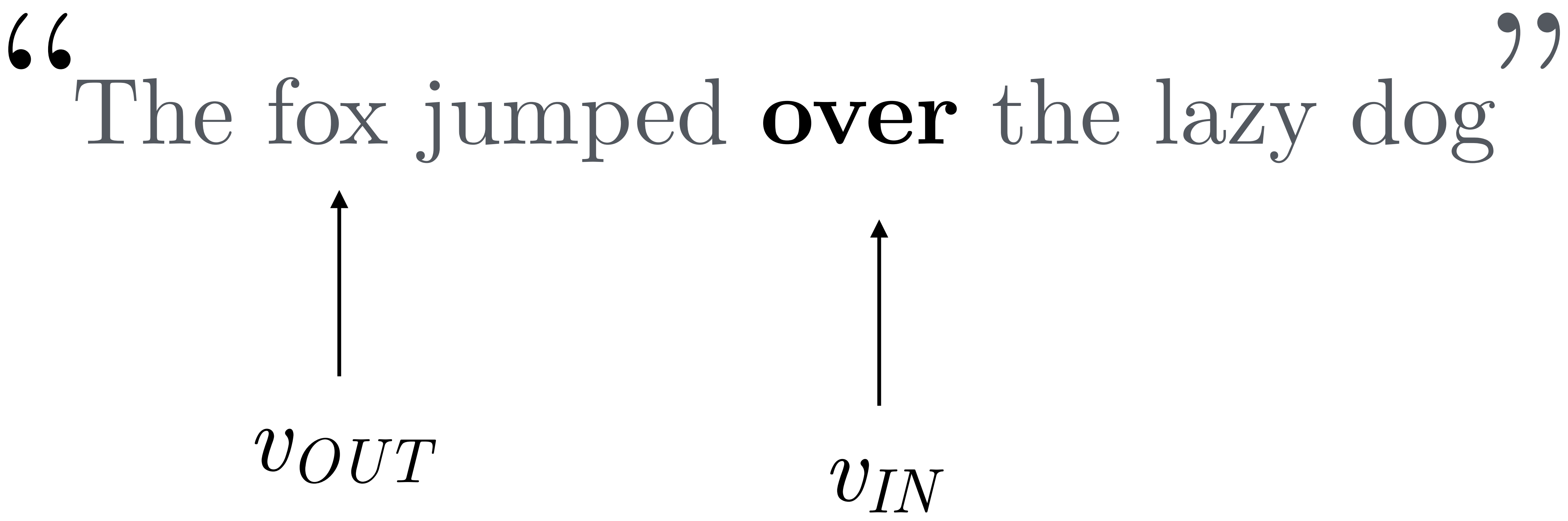
v_{OUT}

v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



"The fox jumped **over** the lazy dog"

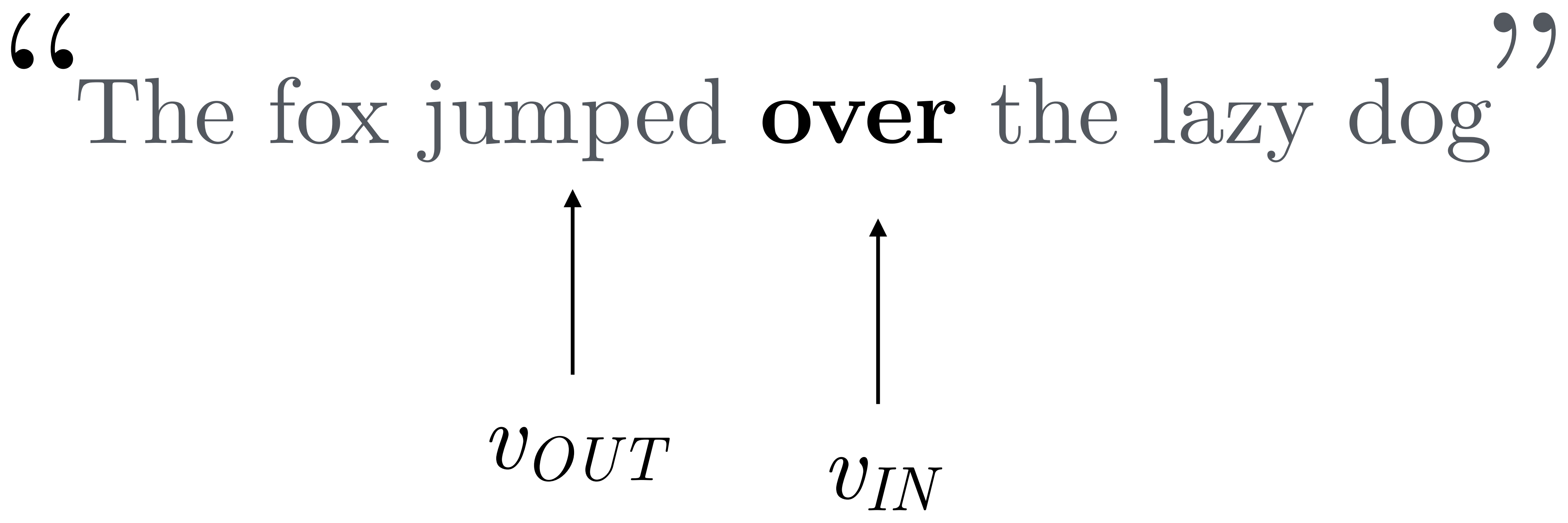
v_{OUT}

v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



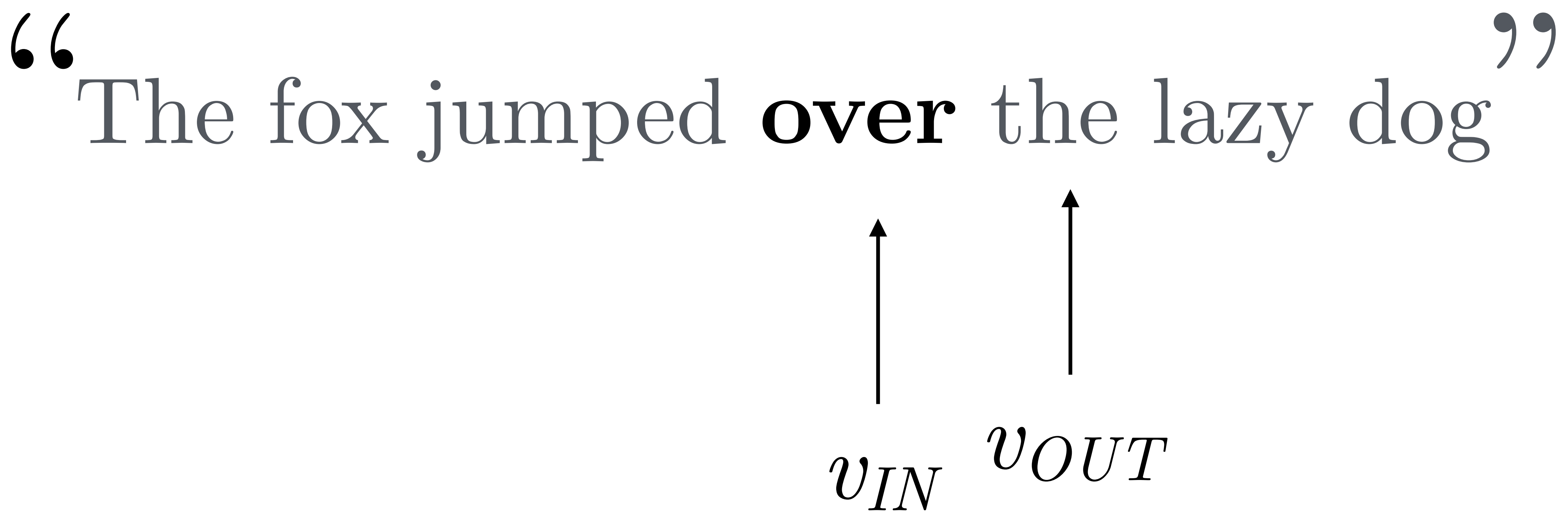
“The fox jumped **over** the lazy dog”

v_{OUT} v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$

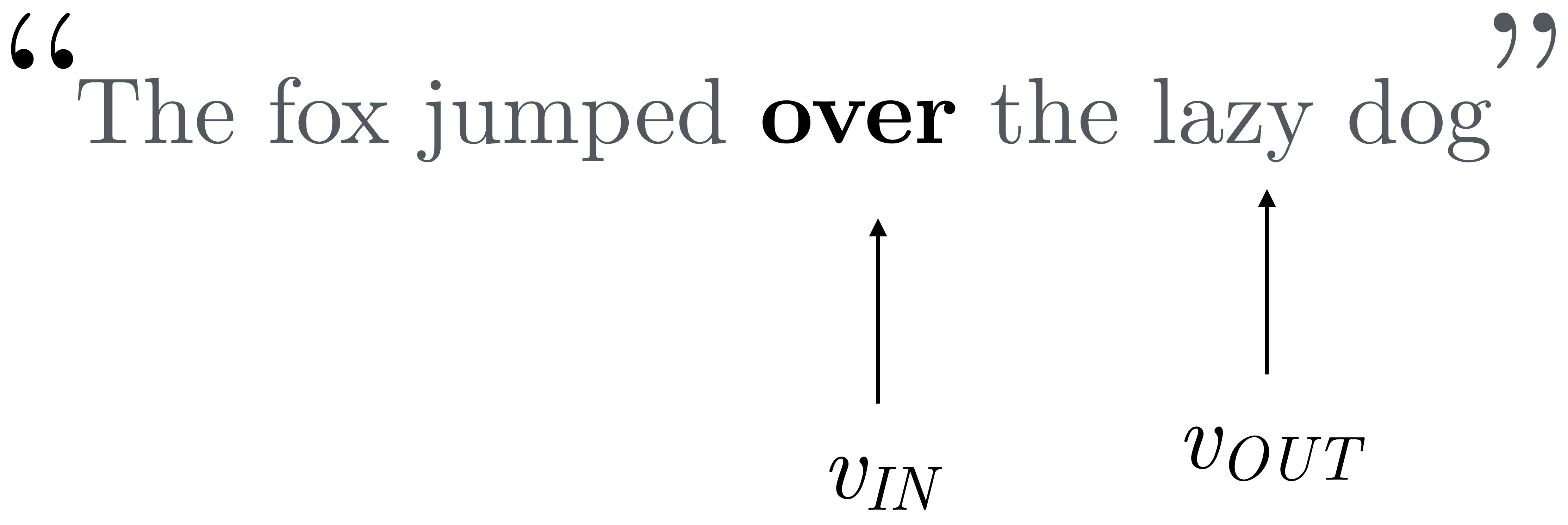


"The fox jumped **over** the lazy dog"

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$

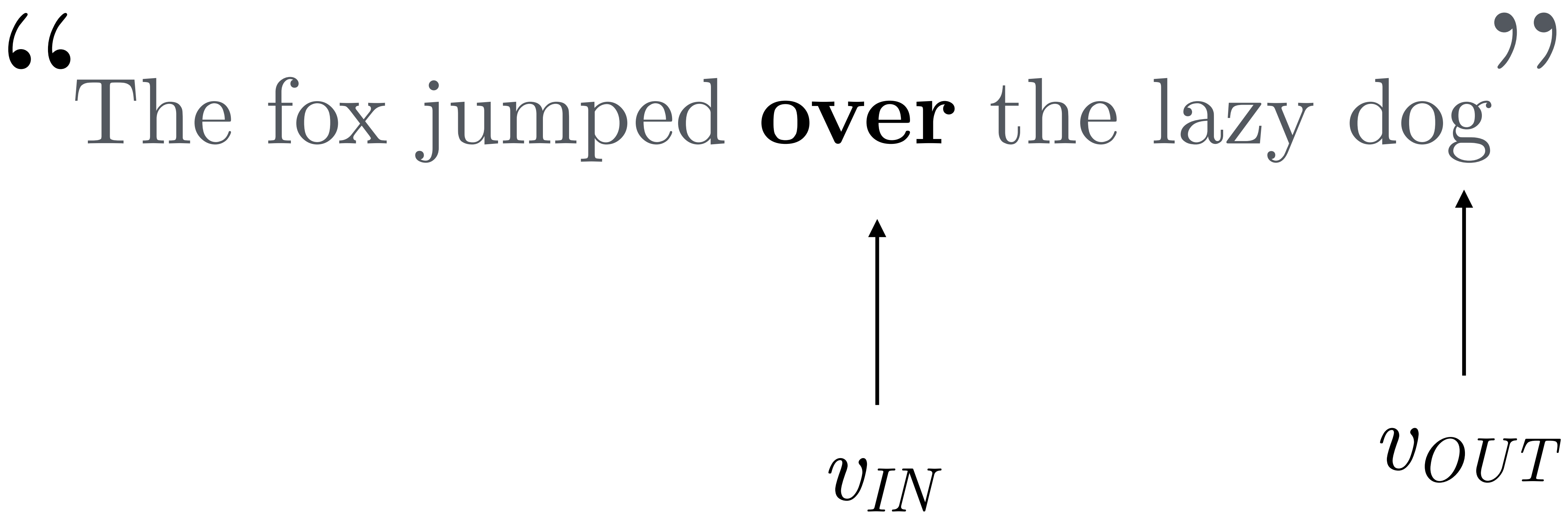


“The fox jumped **over** the lazy dog”

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



“The fox jumped **over** the lazy dog”

v_{IN}

v_{OUT}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$

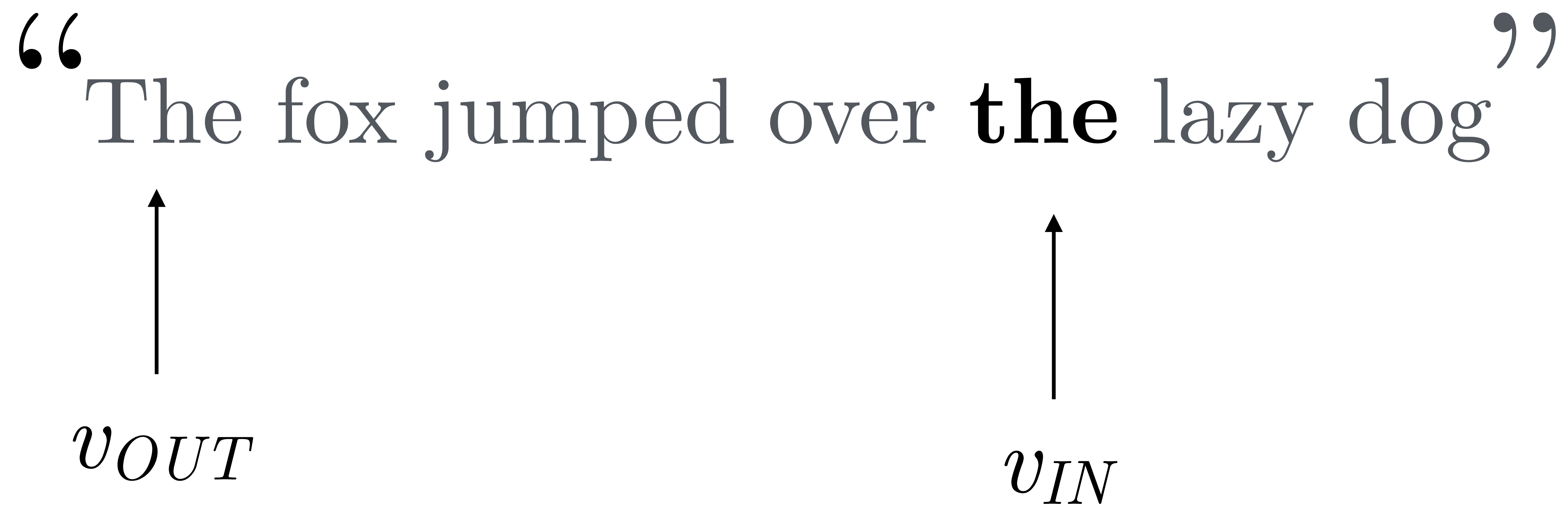
"The fox jumped over **the** lazy dog"

v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



"The fox jumped over **the** lazy dog"

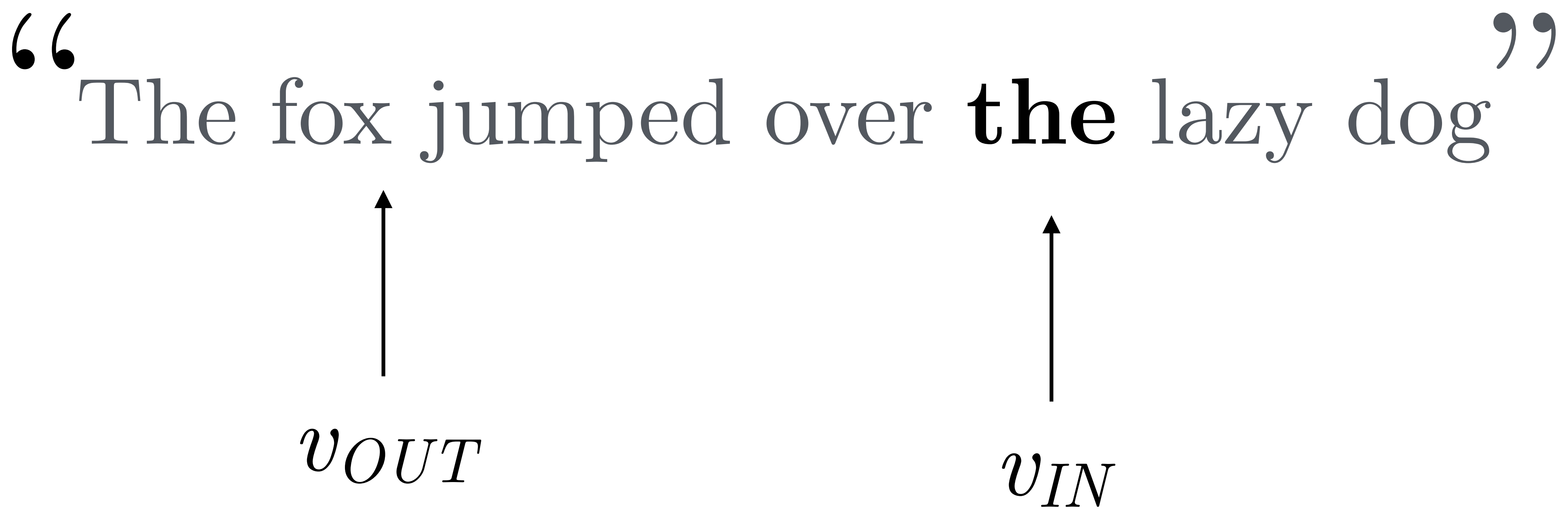
v_{OUT}

v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



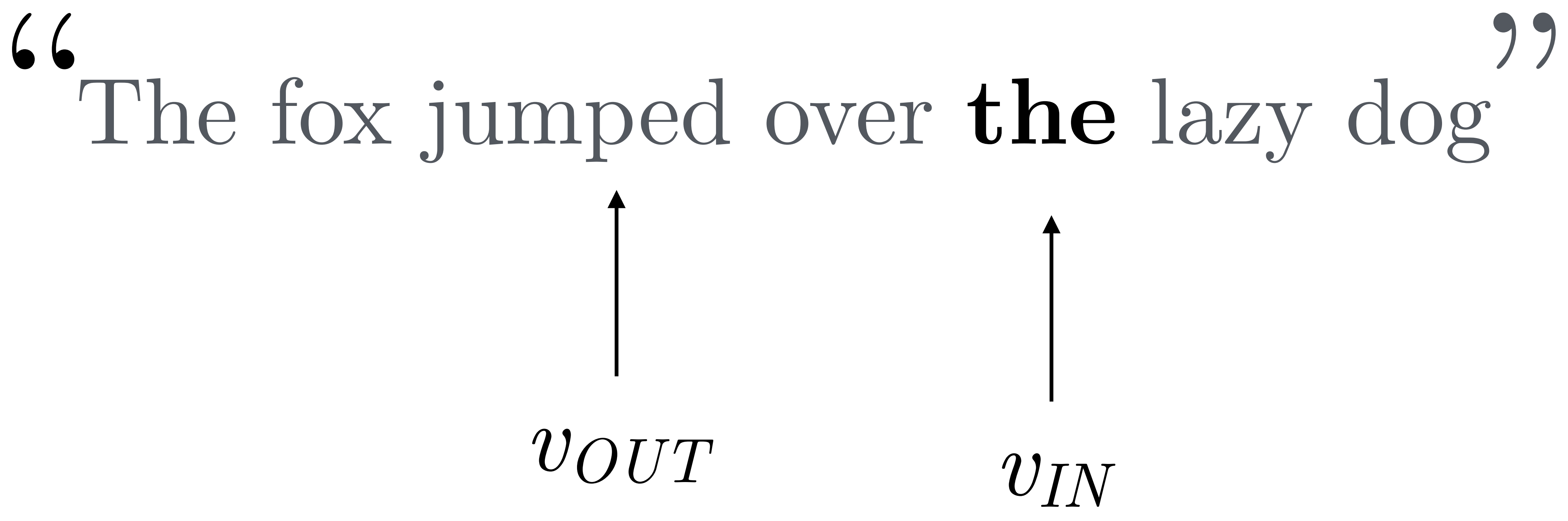
“The fox jumped over **the** lazy dog”

v_{OUT} v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



"The fox jumped over **the** lazy dog"

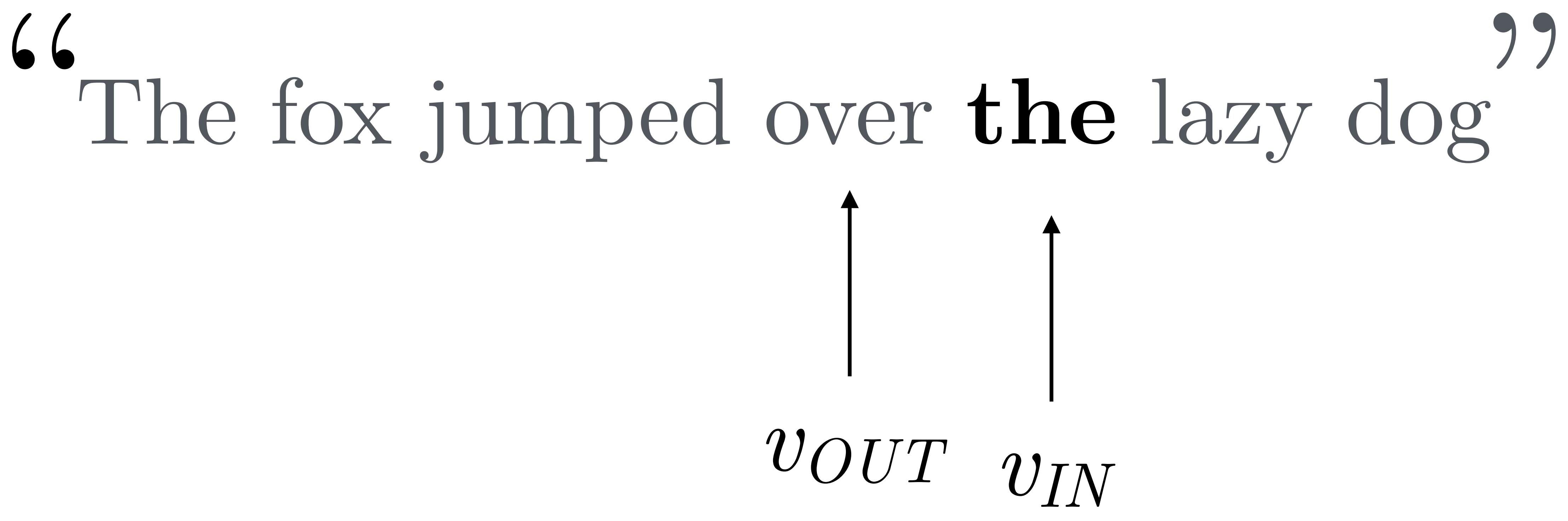
v_{OUT}

v_{IN}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



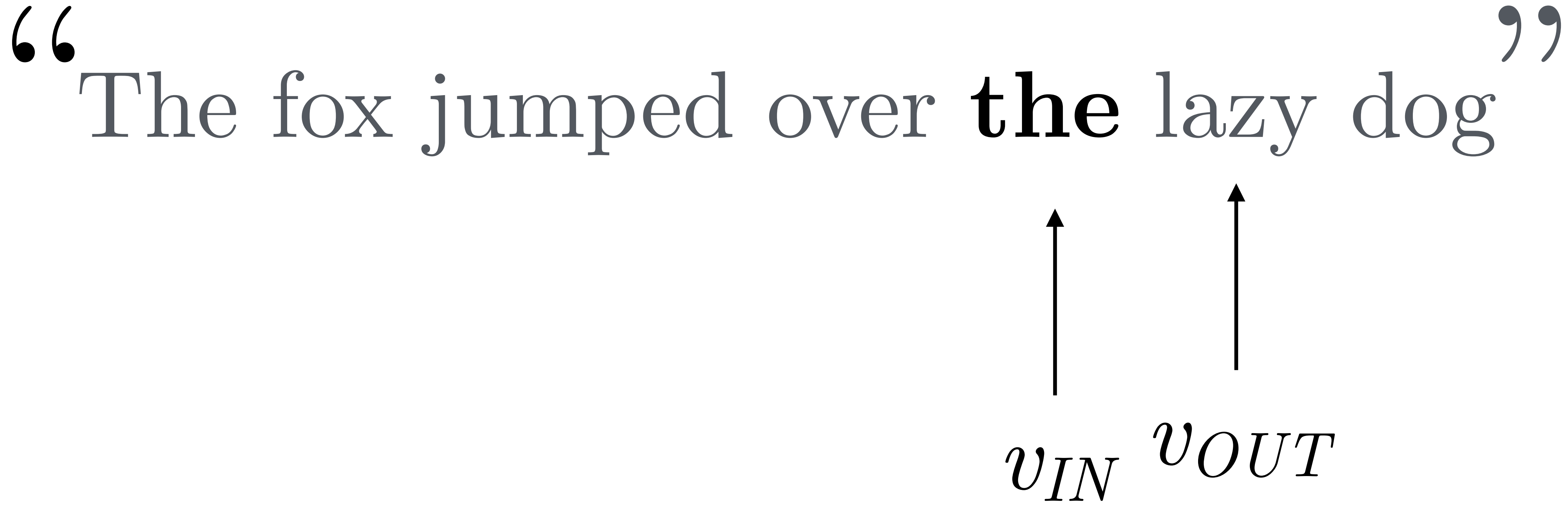
"The fox jumped over **the** lazy dog"

word2vec

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



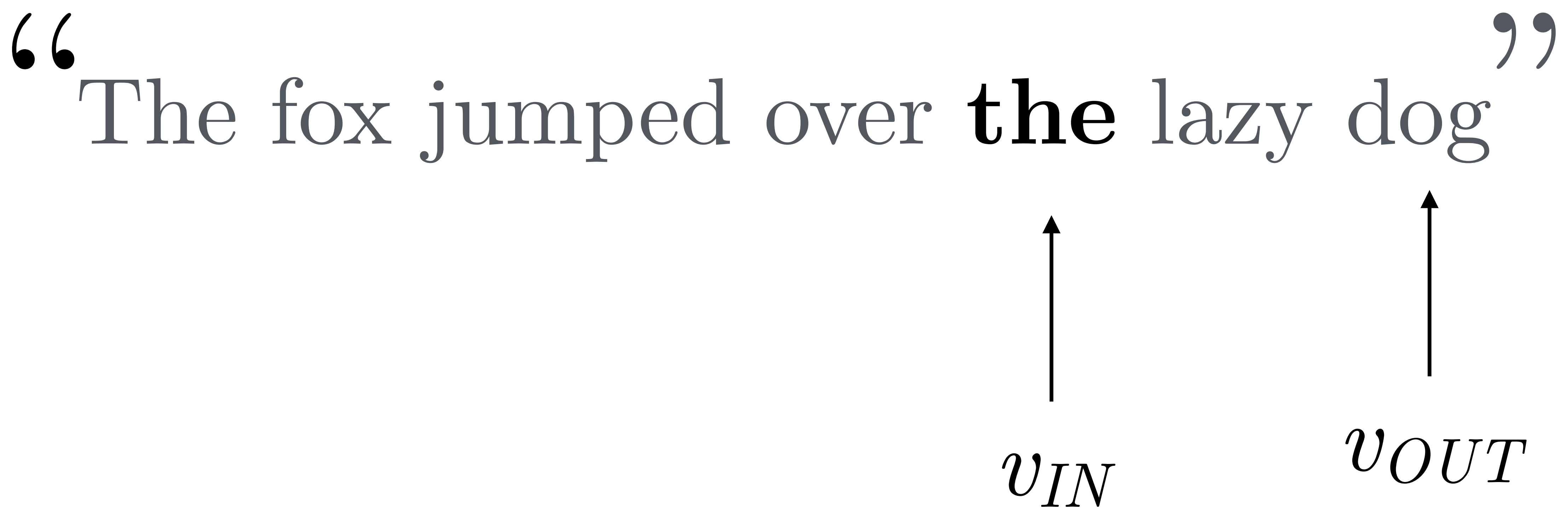
"The fox jumped over **the** lazy dog"

v_{IN} v_{OUT}

Twist: we have *two* vectors for every word.
Should depend on whether it's the input or the output.

Also a *context* window around every input word.

$$P(v_{OUT}|v_{IN})$$



"The fox jumped over **the** lazy dog"

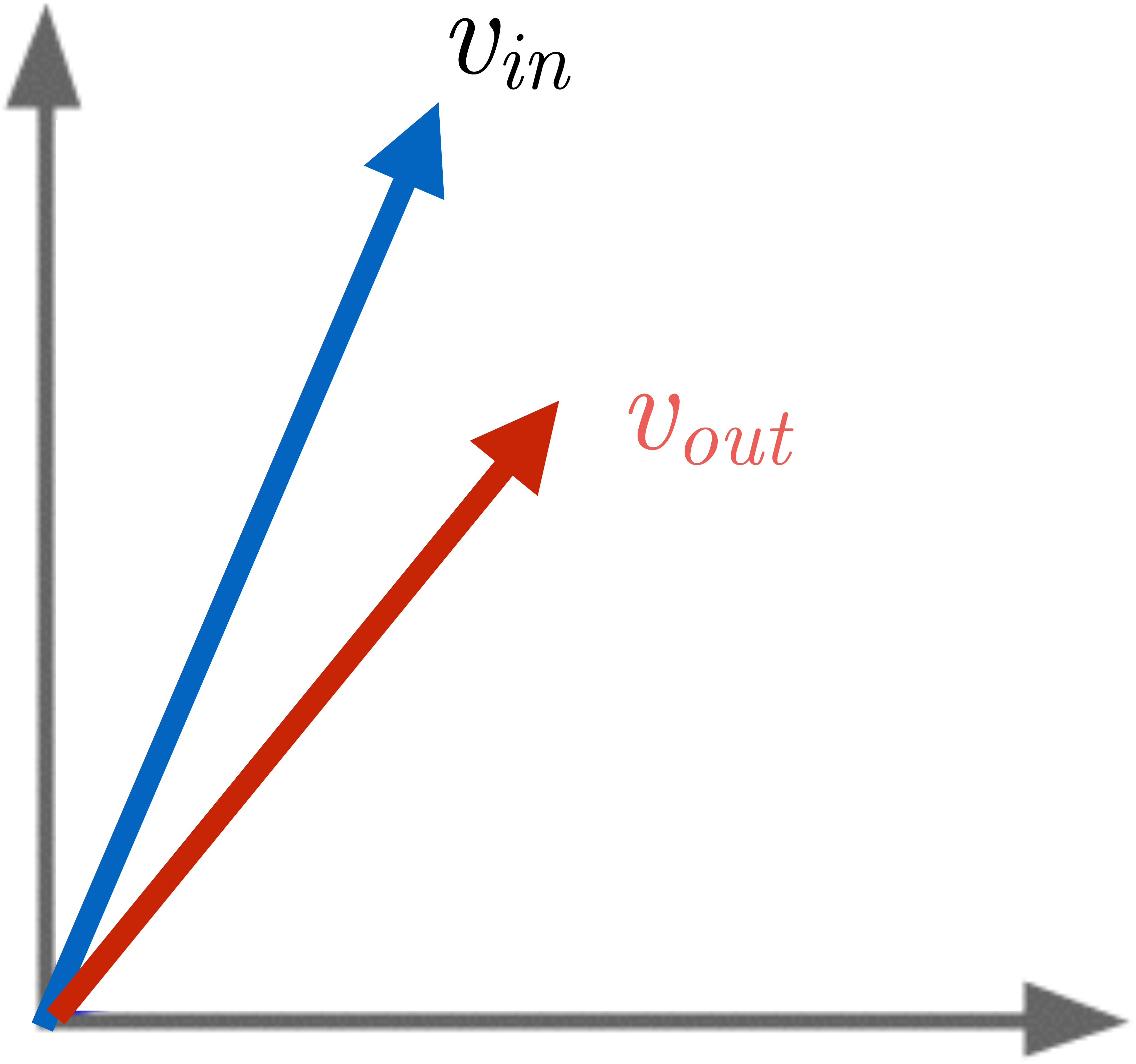
objective

How should we define $P(v_{OUT}|v_{IN})$?

Measure loss between
v_{IN} and v_{OUT}?

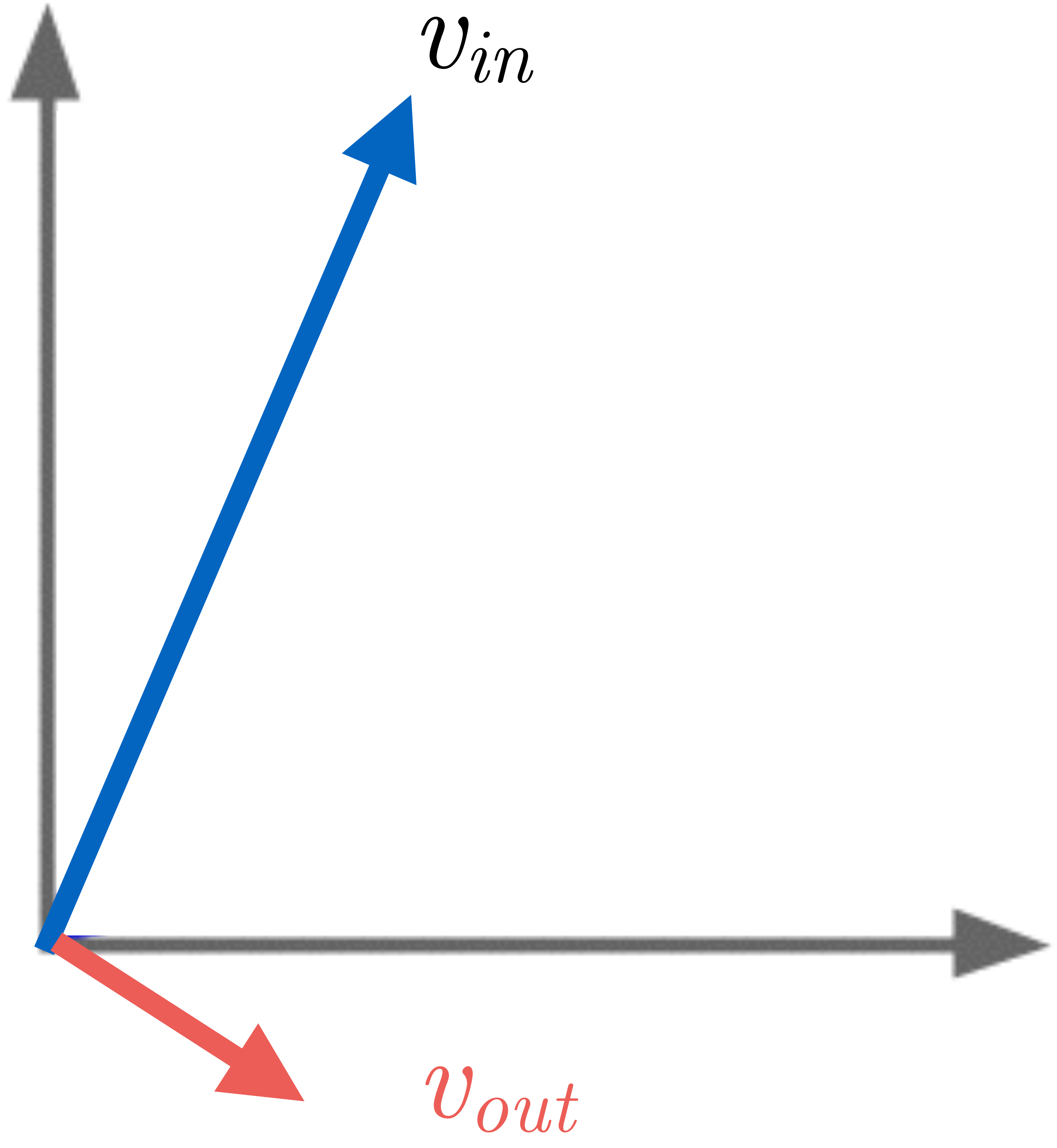
$$v_{in} \bullet v_{out}$$

objective



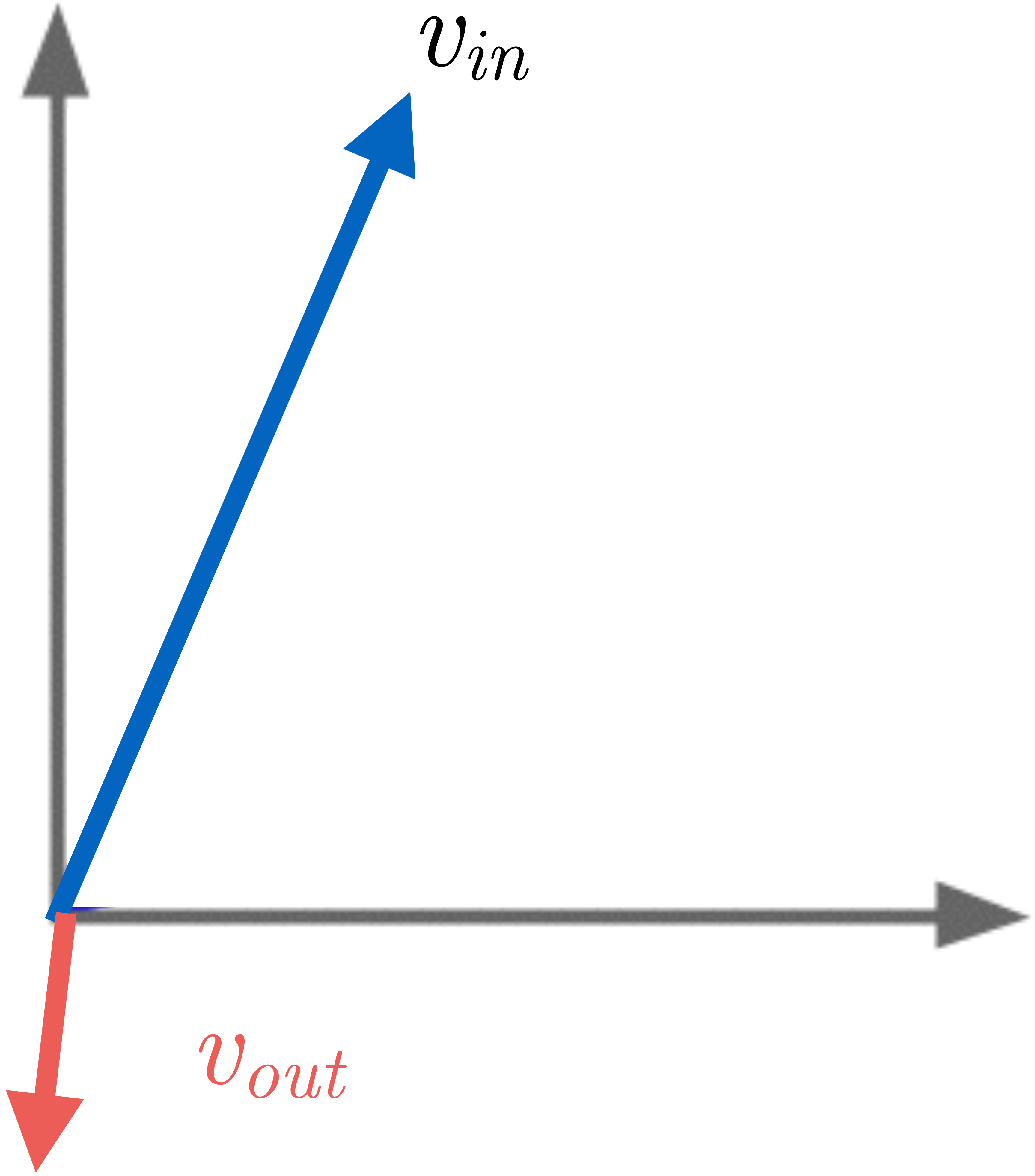
$$v_{in} \cdot v_{out} \sim 1$$

objective



$$v_{in} \cdot v_{out} \sim 0$$

objective



$v_{in} \cdot v_{out} \sim$ -1

objective

$$v_{in} \bullet v_{out} \in [\text{-}1,1]$$

objective

But we'd like to measure a probability.

$$v_{in} \bullet v_{out} \in [-1,1]$$

objective

But we'd like to measure a probability.

$$softmax(v_{in} \bullet v_{out}) \in [0,1]$$

objective

But we'd like to measure a probability.

$$softmax(v_{in} \bullet v_{out})$$

Probability of choosing 1 of N discrete items.
Mapping from vector space to a multinomial over words.

objective

But we'd like to measure a probability.

$$softmax \sim \quad exp(v_{in} \bullet v_{out}) \in [0,1]$$

objective

But we'd like to measure a probability.

$$softmax = \frac{exp(v_{in} \bullet v_{out})}{\underset{k \in V}{\Sigma} exp(v_{in} \bullet v_k)}$$

Normalization term over all words

objective

But we'd like to measure a probability.

$$softmax = \frac{exp(v_{in} \bullet v_{out})}{\sum_{k \in V} exp(v_{in} \bullet v_k)} = P(v_{out}|v_{in})$$

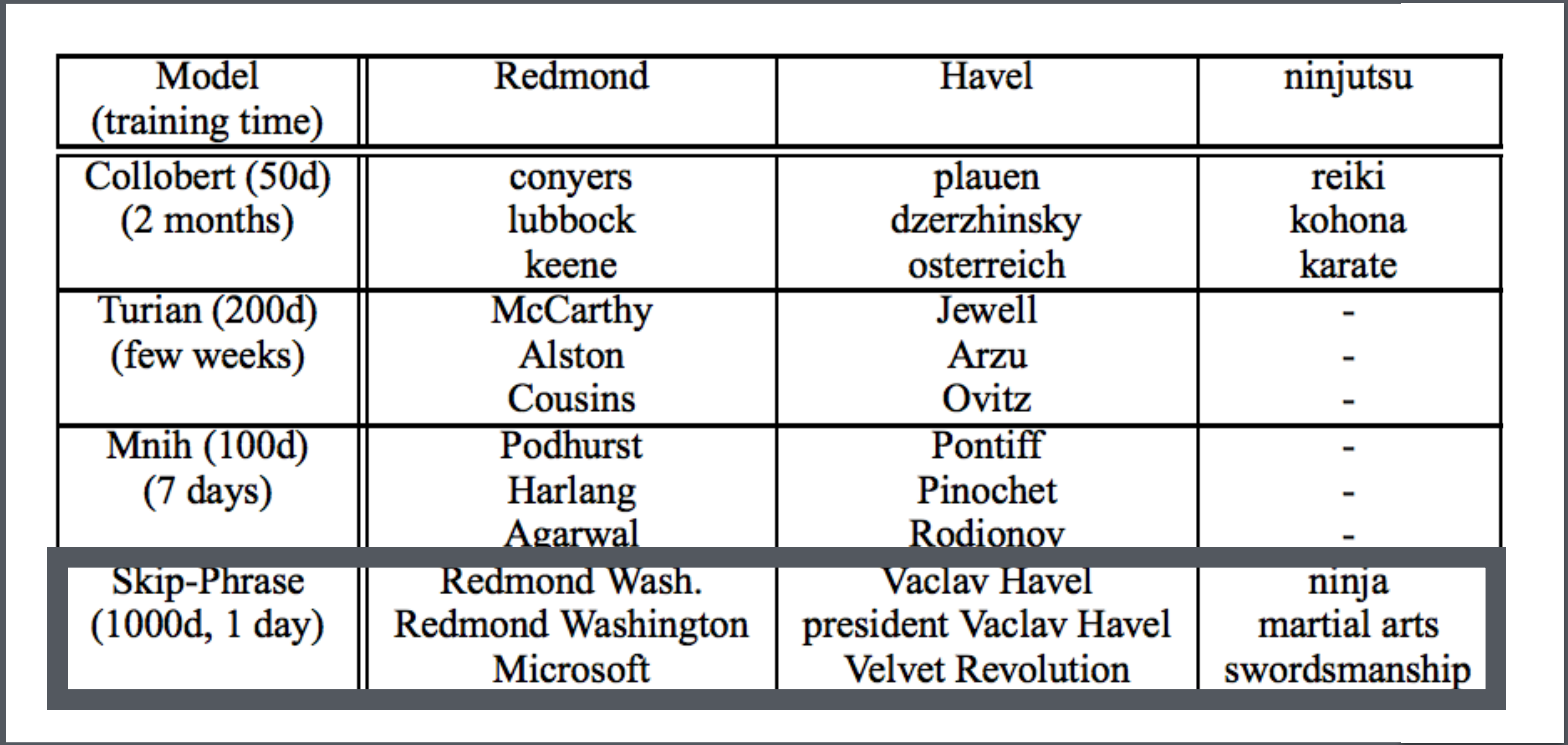
objective

Learn by gradient descent on the softmax prob.

For every example we see update v_{in}

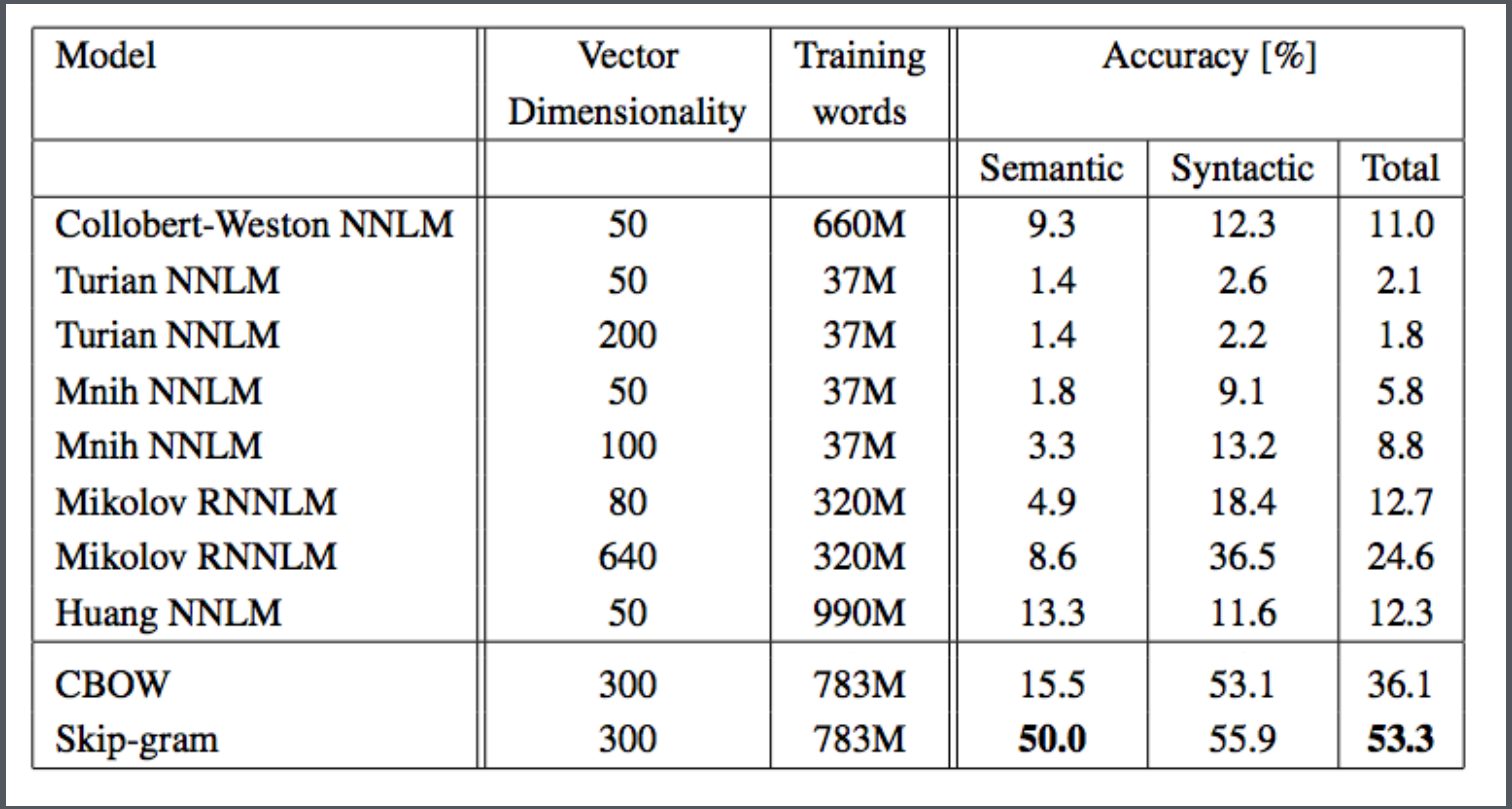
$$v_{in} := v_{in} + \frac{\partial}{\partial v_{in}} P(v_{out}|v_{in})$$

$$v_{out} := v_{out} + \frac{\partial}{\partial v_{out}} P(v_{out}|v_{in})$$



Model (training time)	Redmond	Havel	ninjutsu
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship

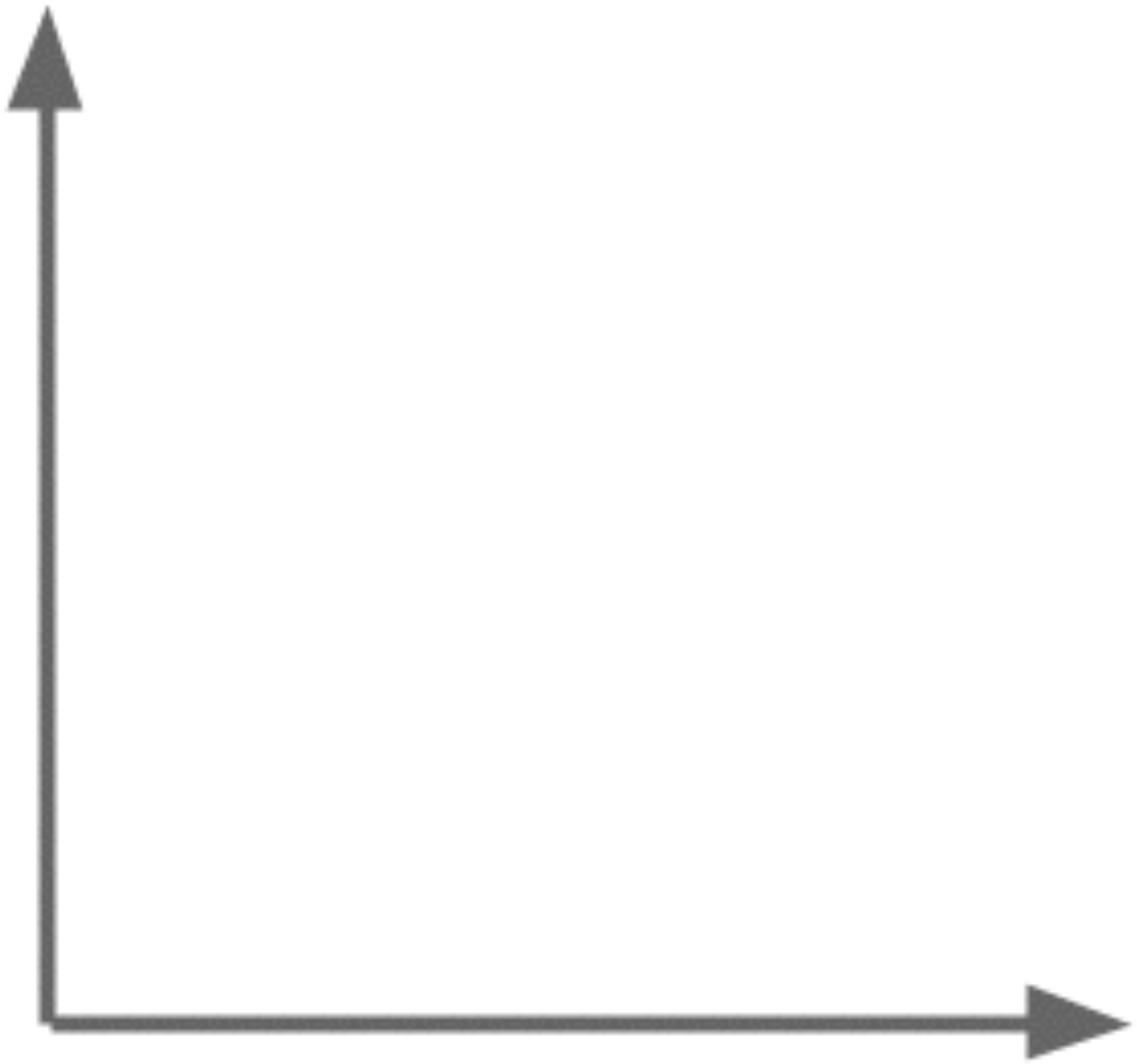
← word2vec



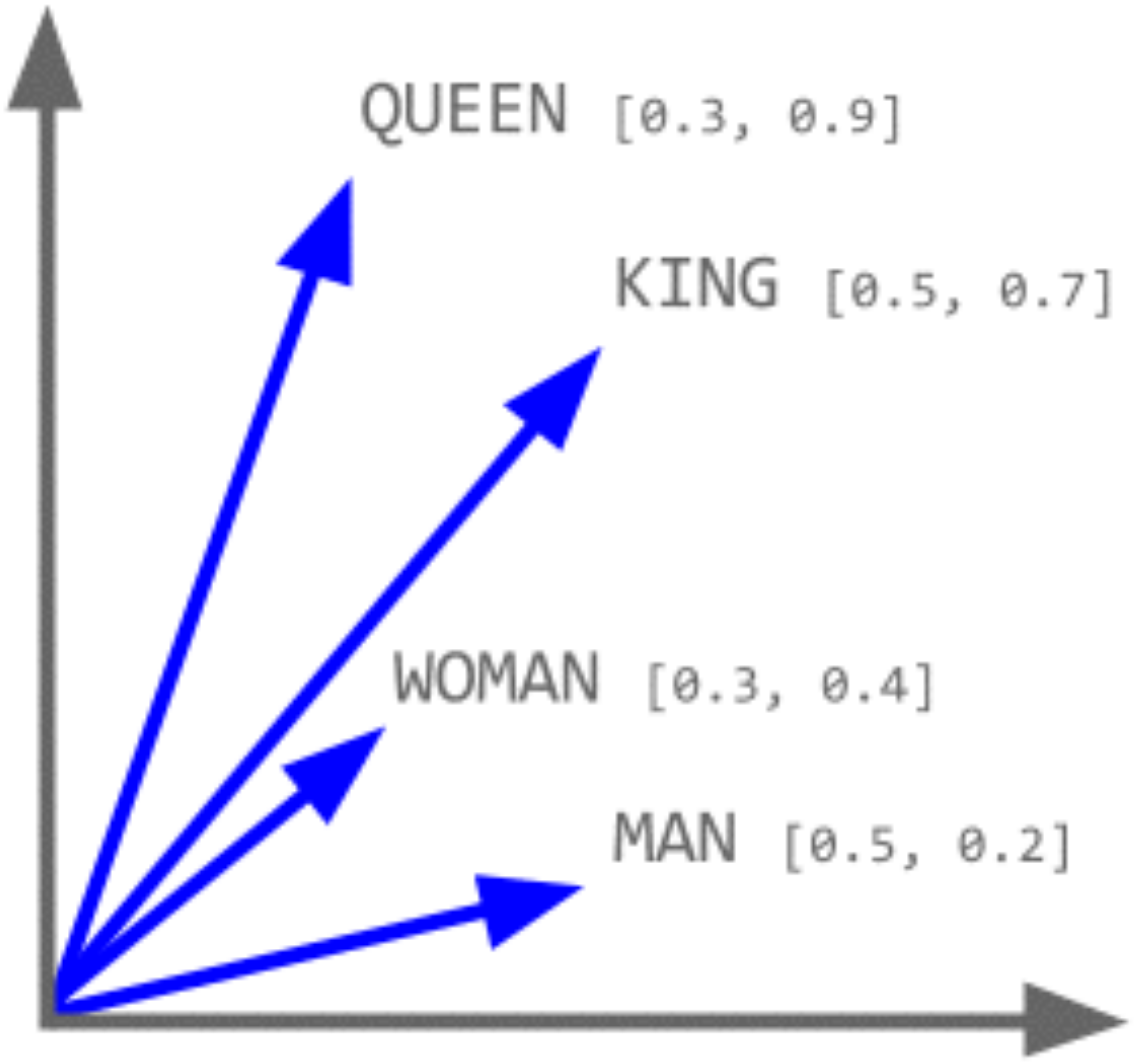
Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

← word2vec

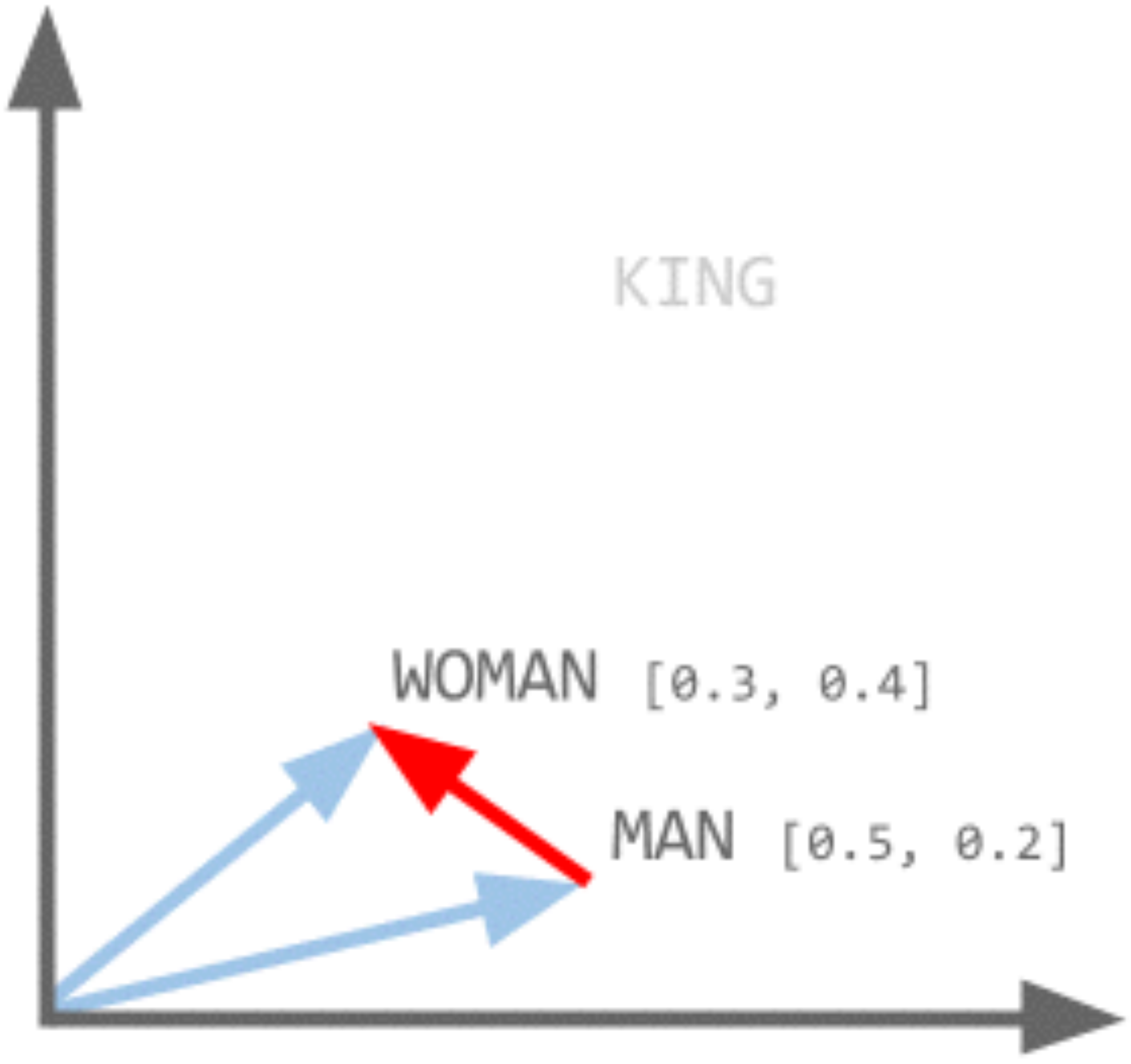
What is `king` + `man` - `woman`?



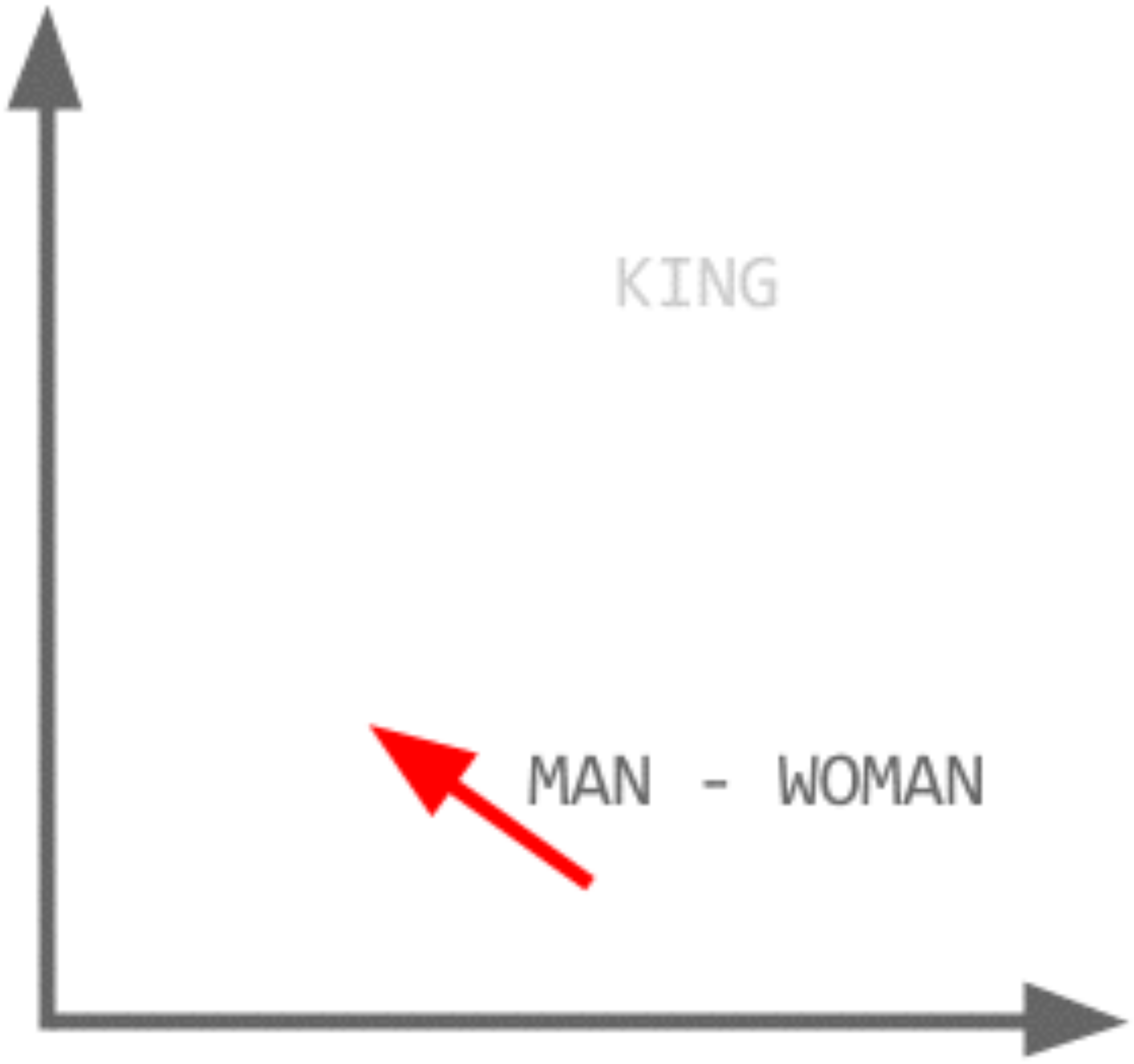
Load up the word vectors



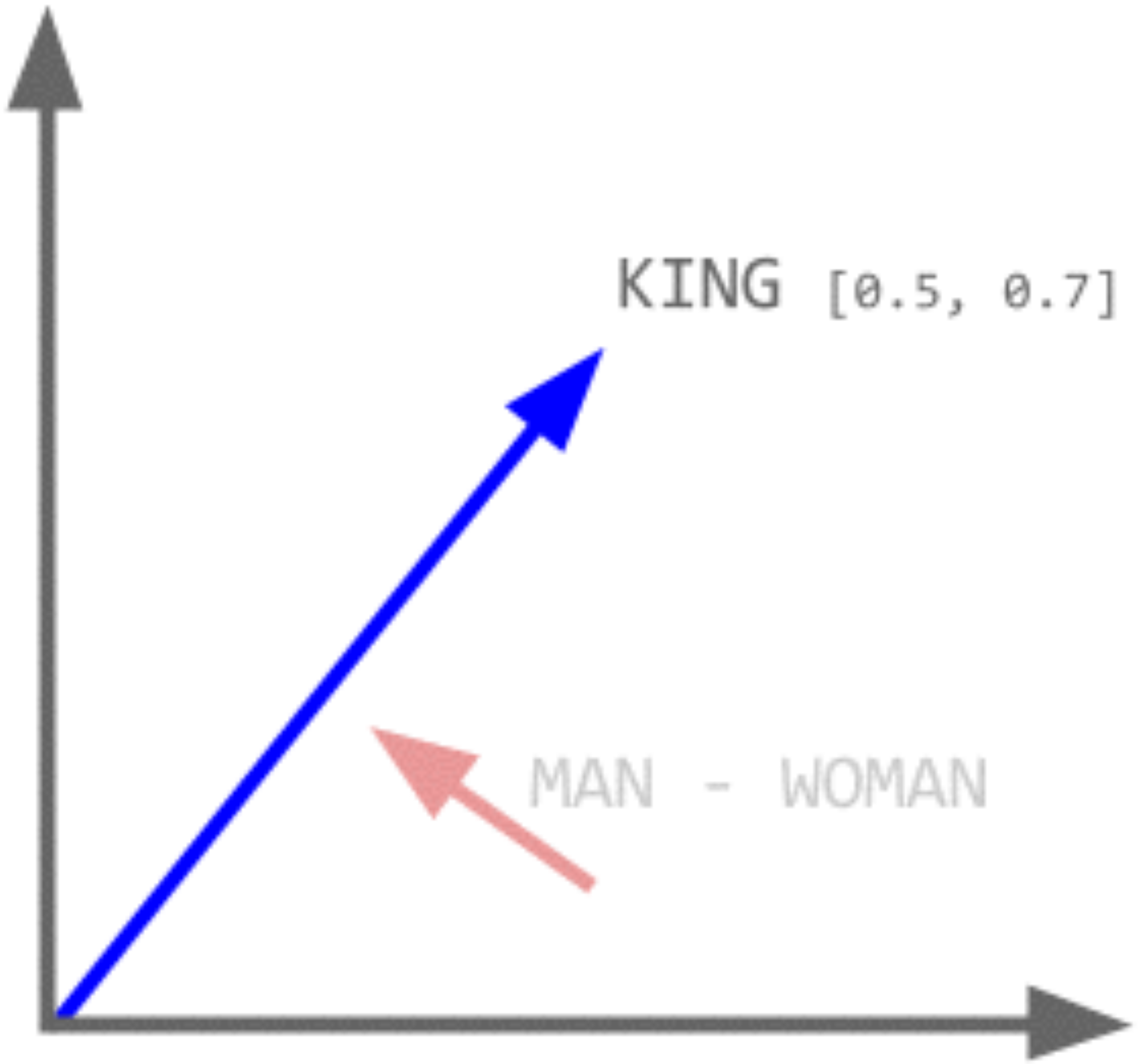
Start with `man - woman`



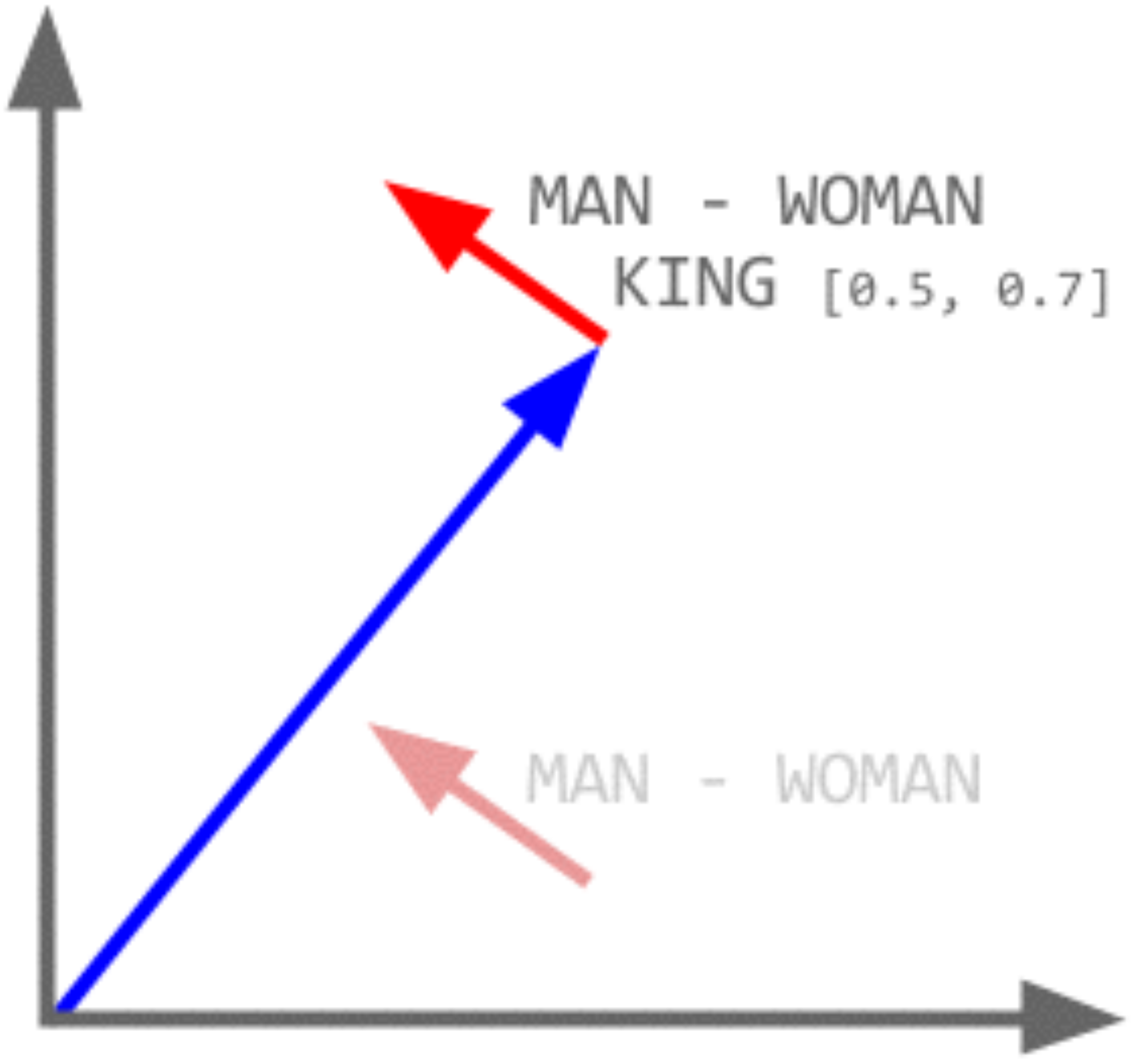
Start with `man - woman`



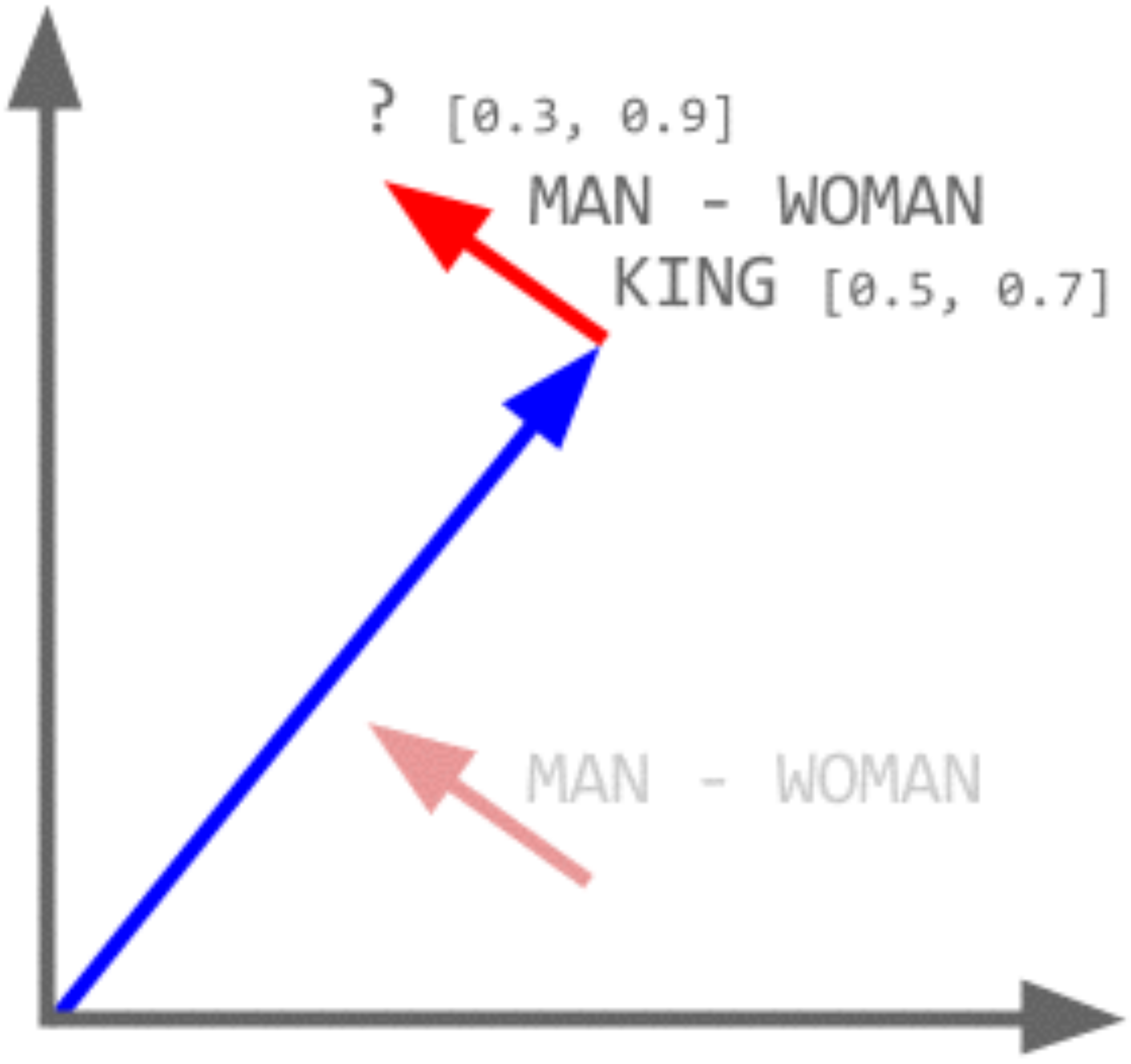
Then take `king`



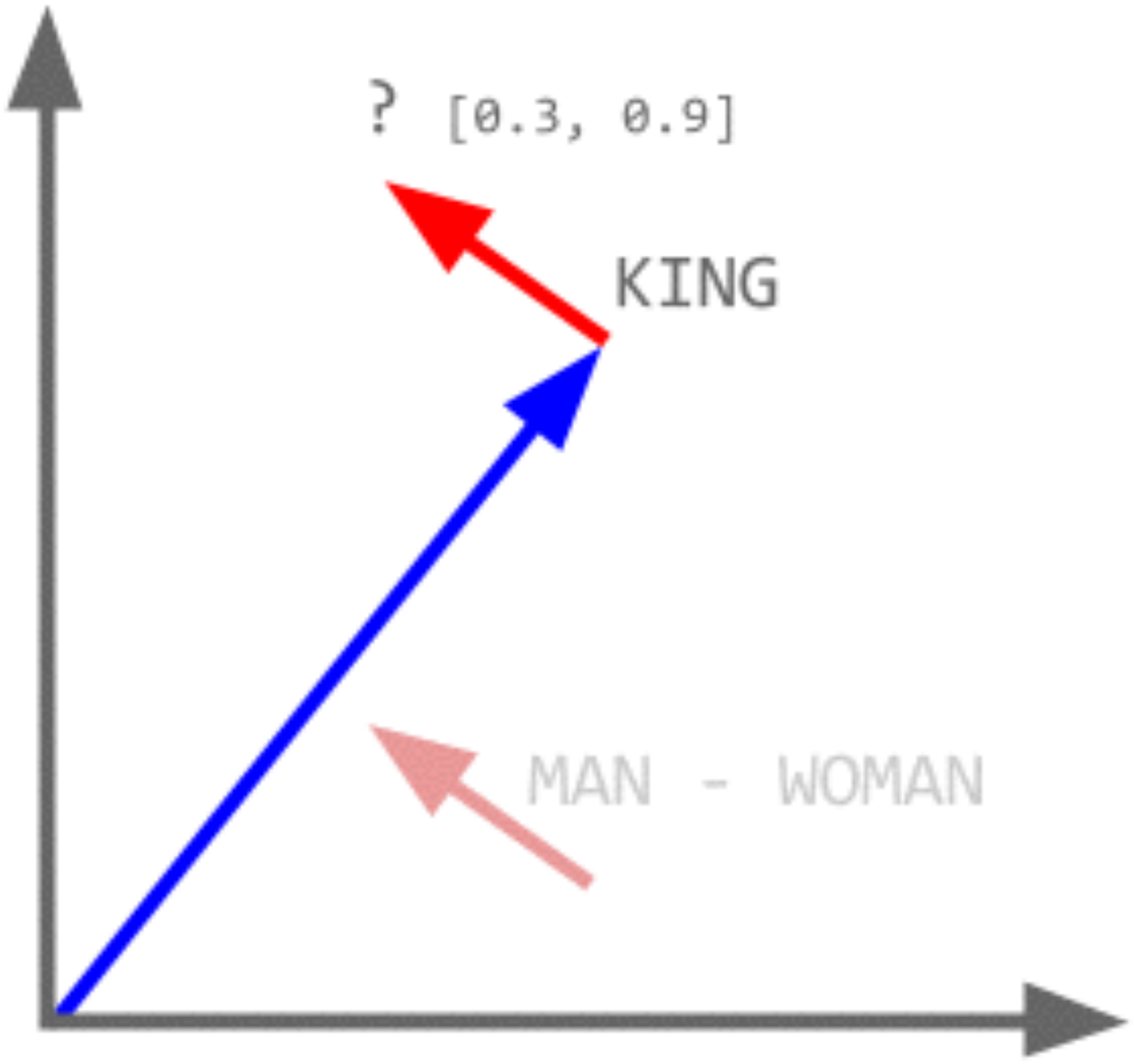
And add `man - woman`



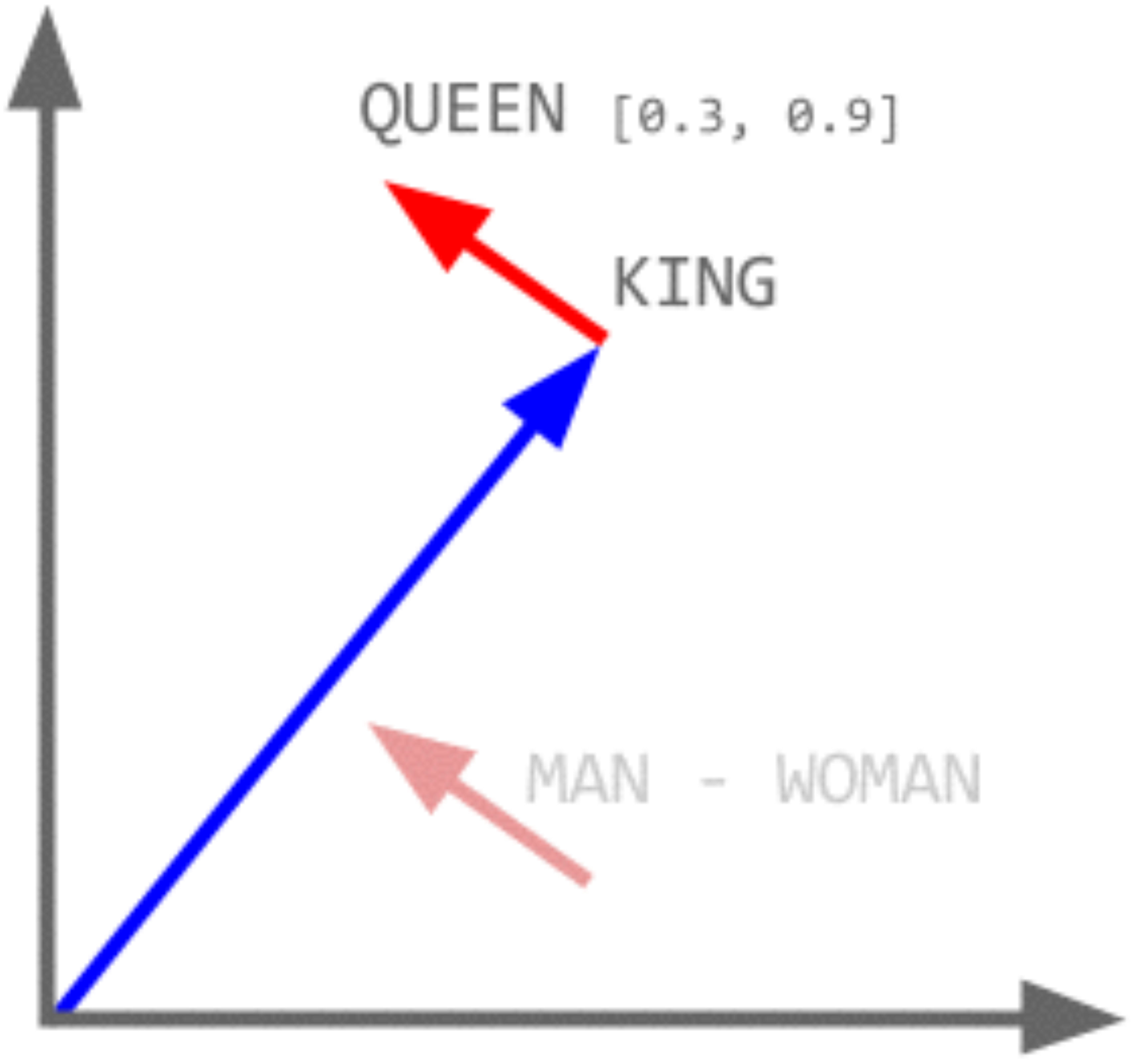
And add `man - woman`



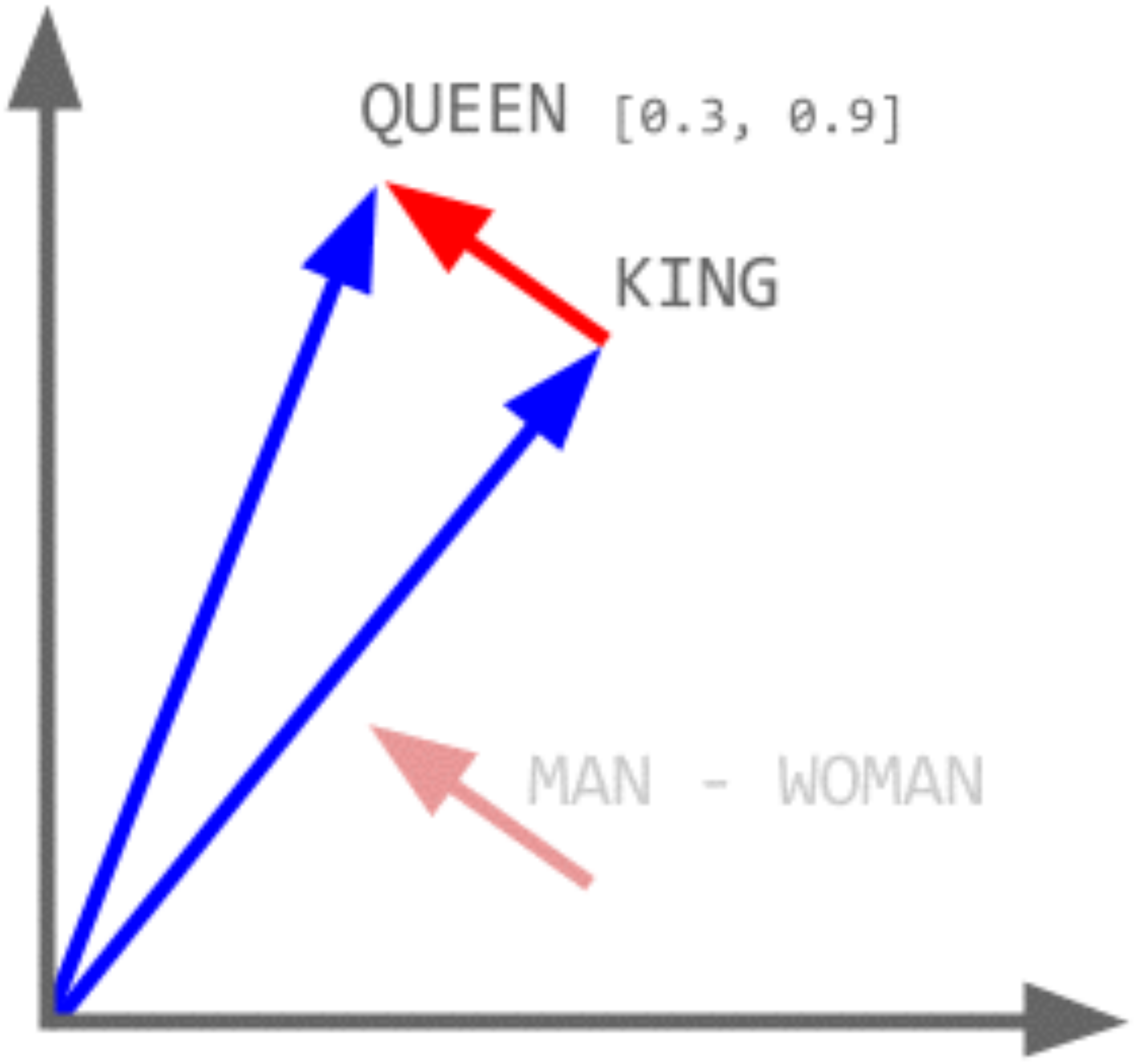
Find nearest word to result



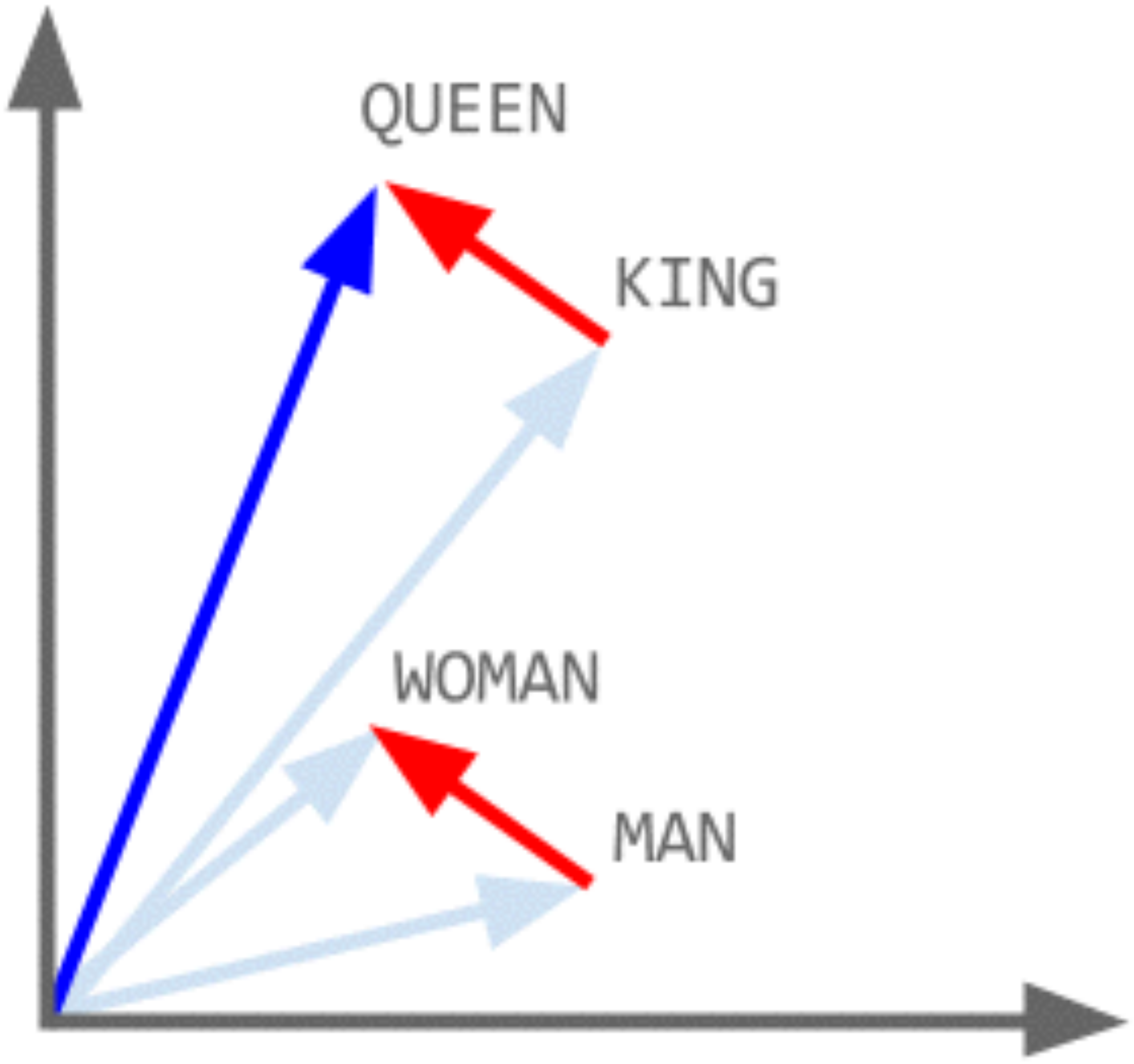
`queen` is closest to resulting vector



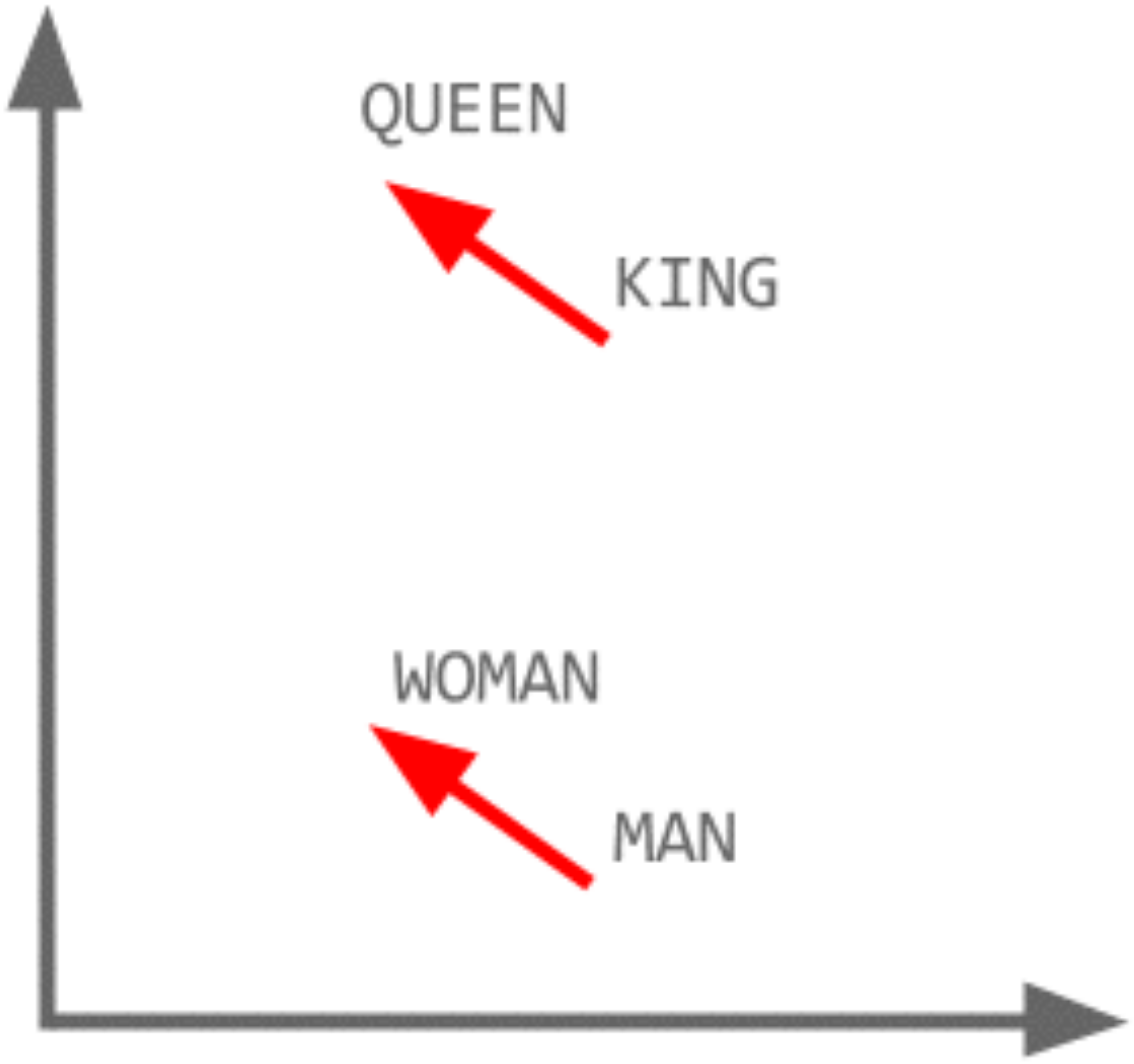
`queen` is closest to resulting vector



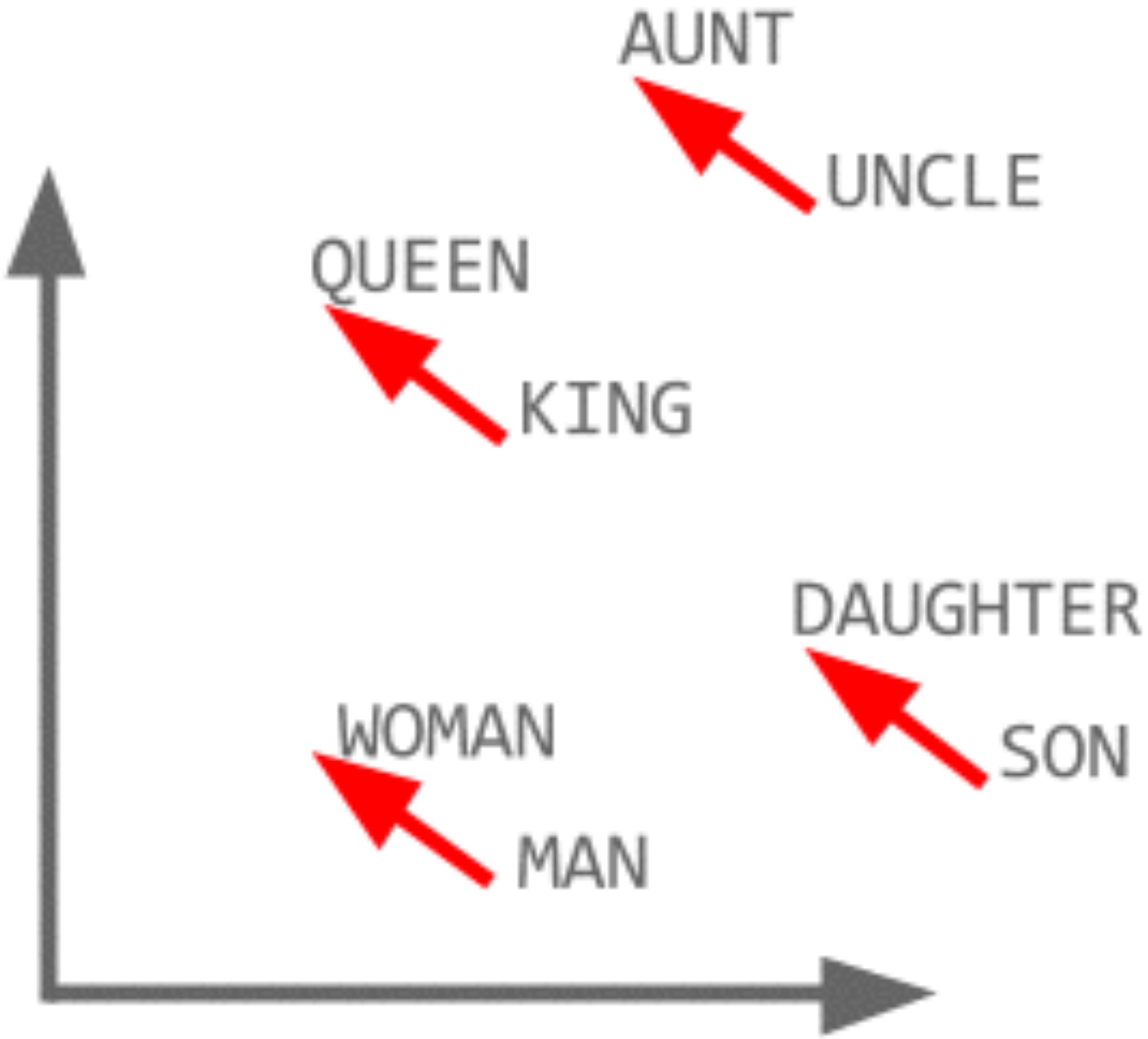
So `king` + `man` - `woman` = `queen!`



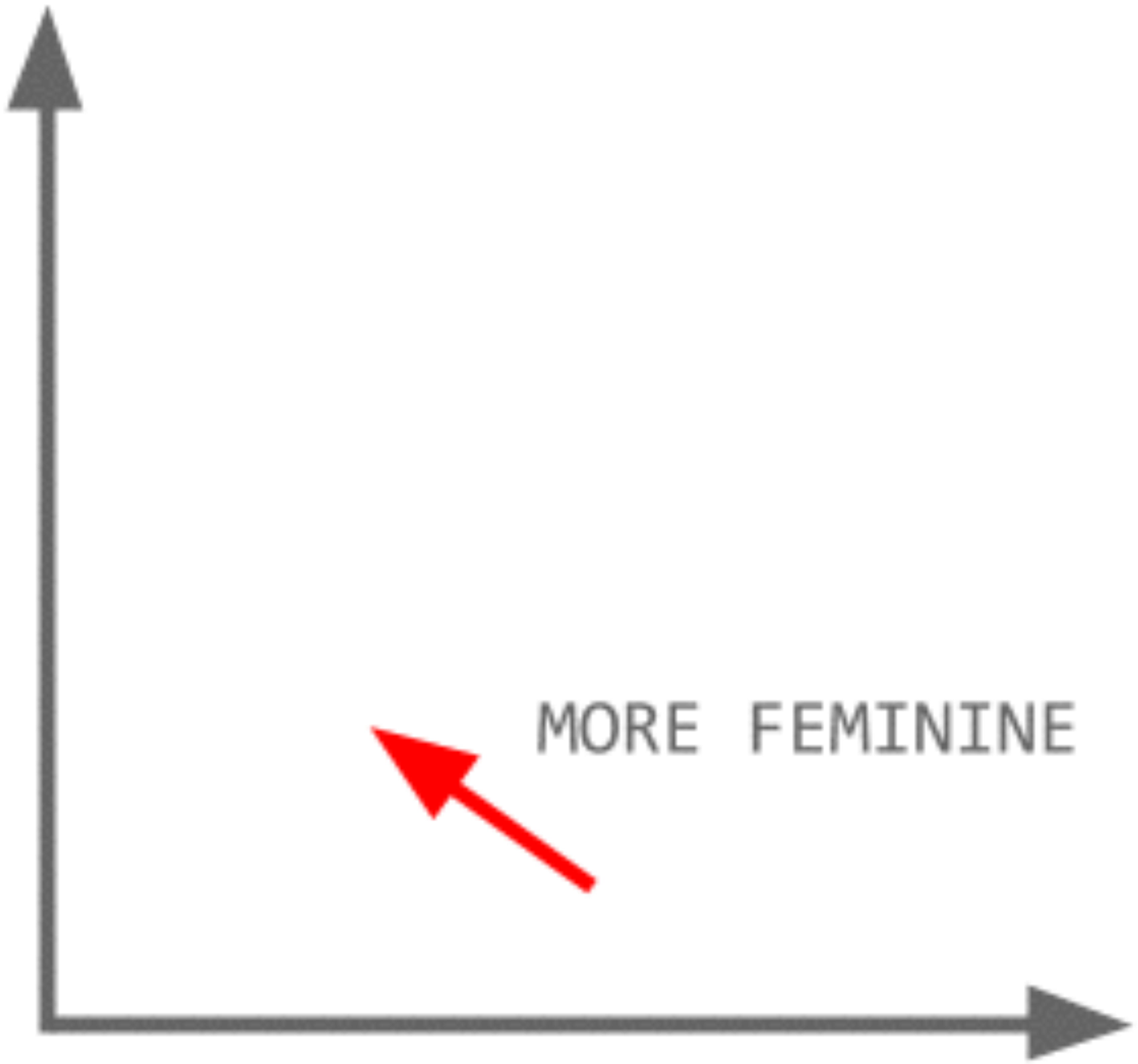
The red direction encodes gender



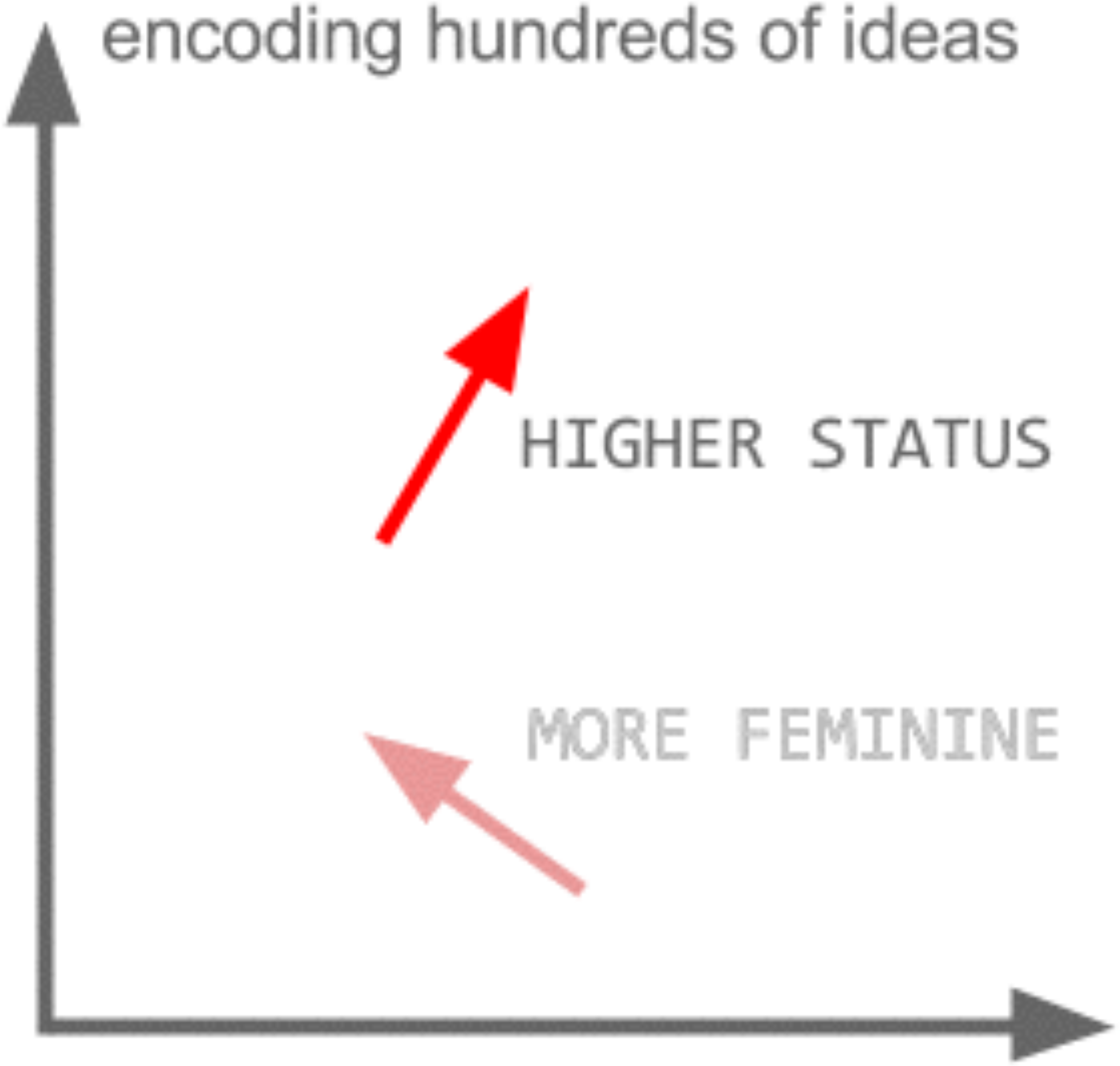
Which is consistent across all words

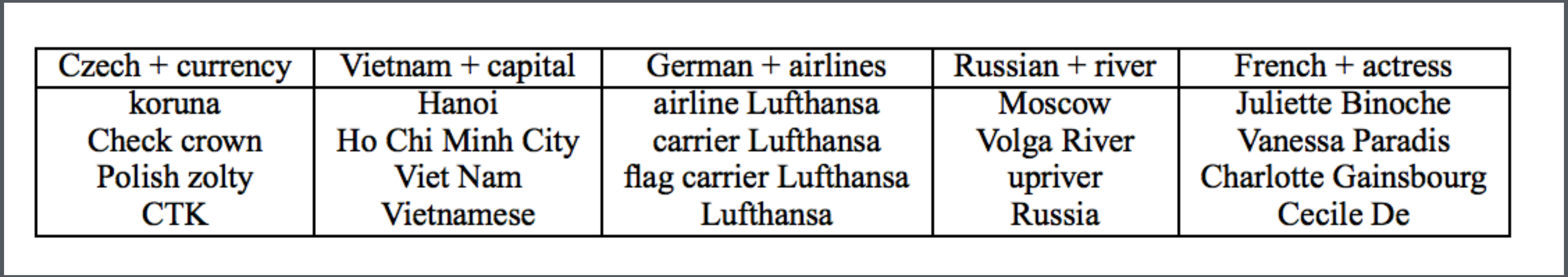


This direction always means gender



We have hundreds of directions encoding hundreds of ideas





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

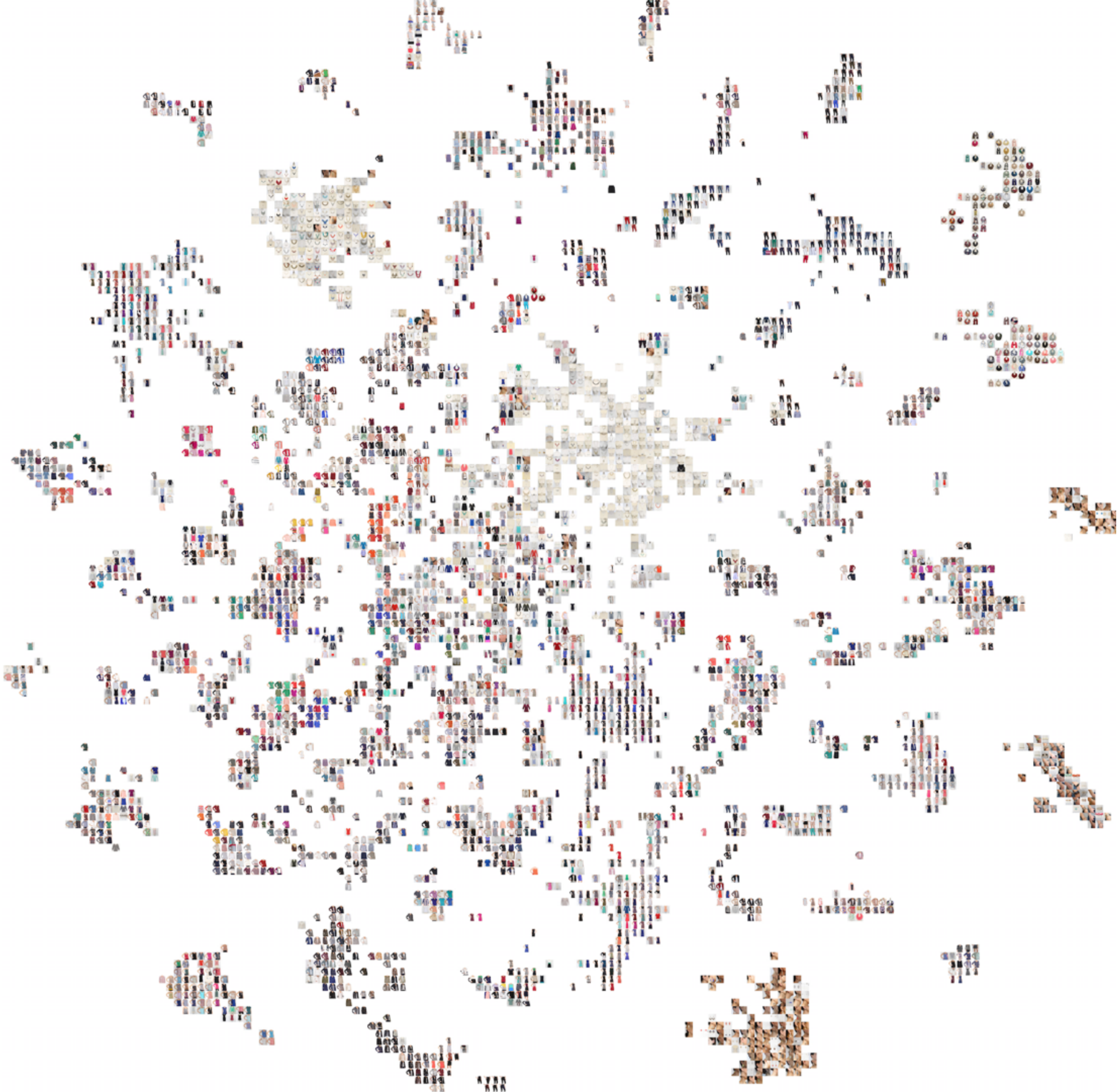
ITEM_3469 + ‘Pregnant’

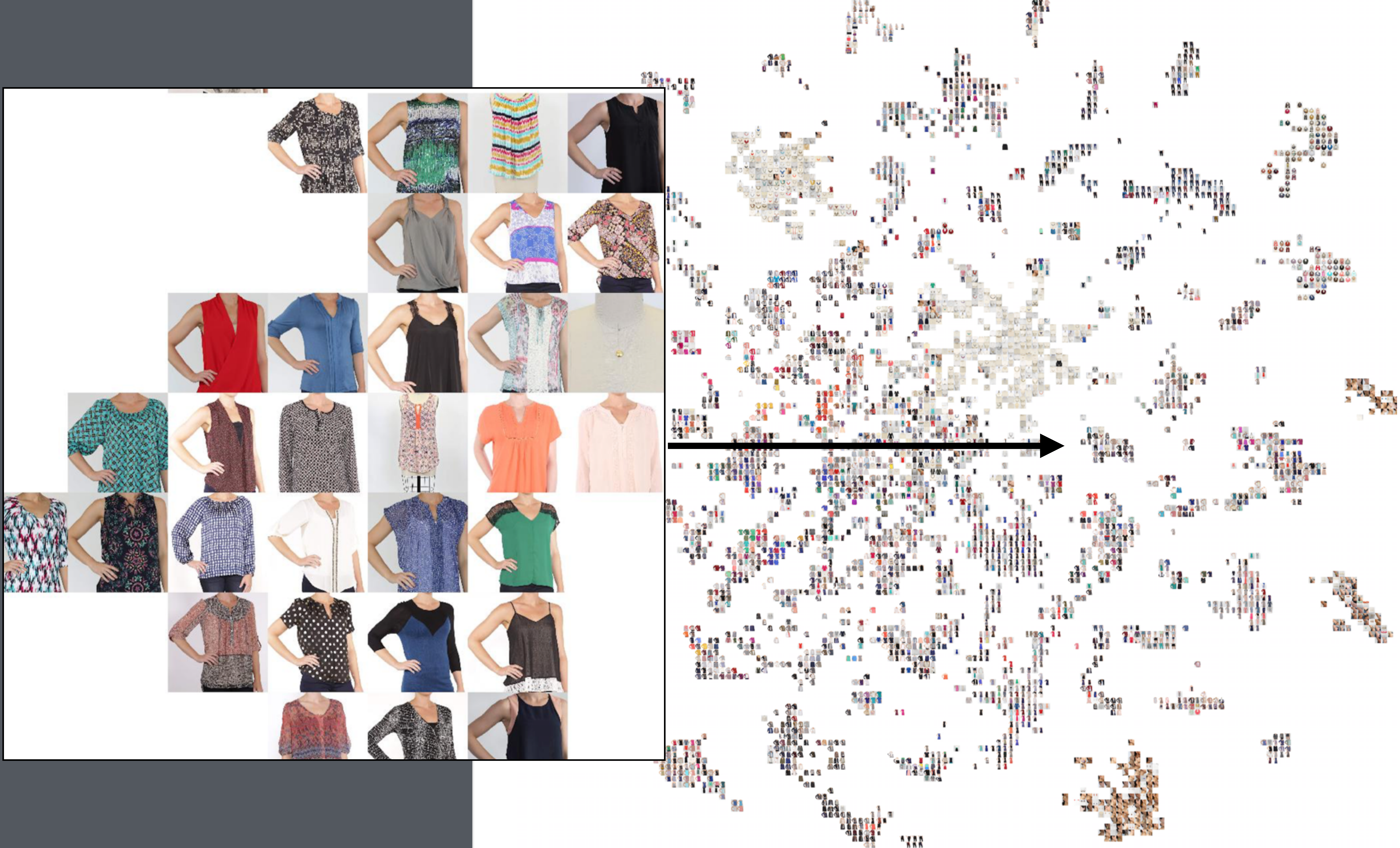
+ ‘Pregnant’

= ITEM_701333
= ITEM_901004
= ITEM_800456

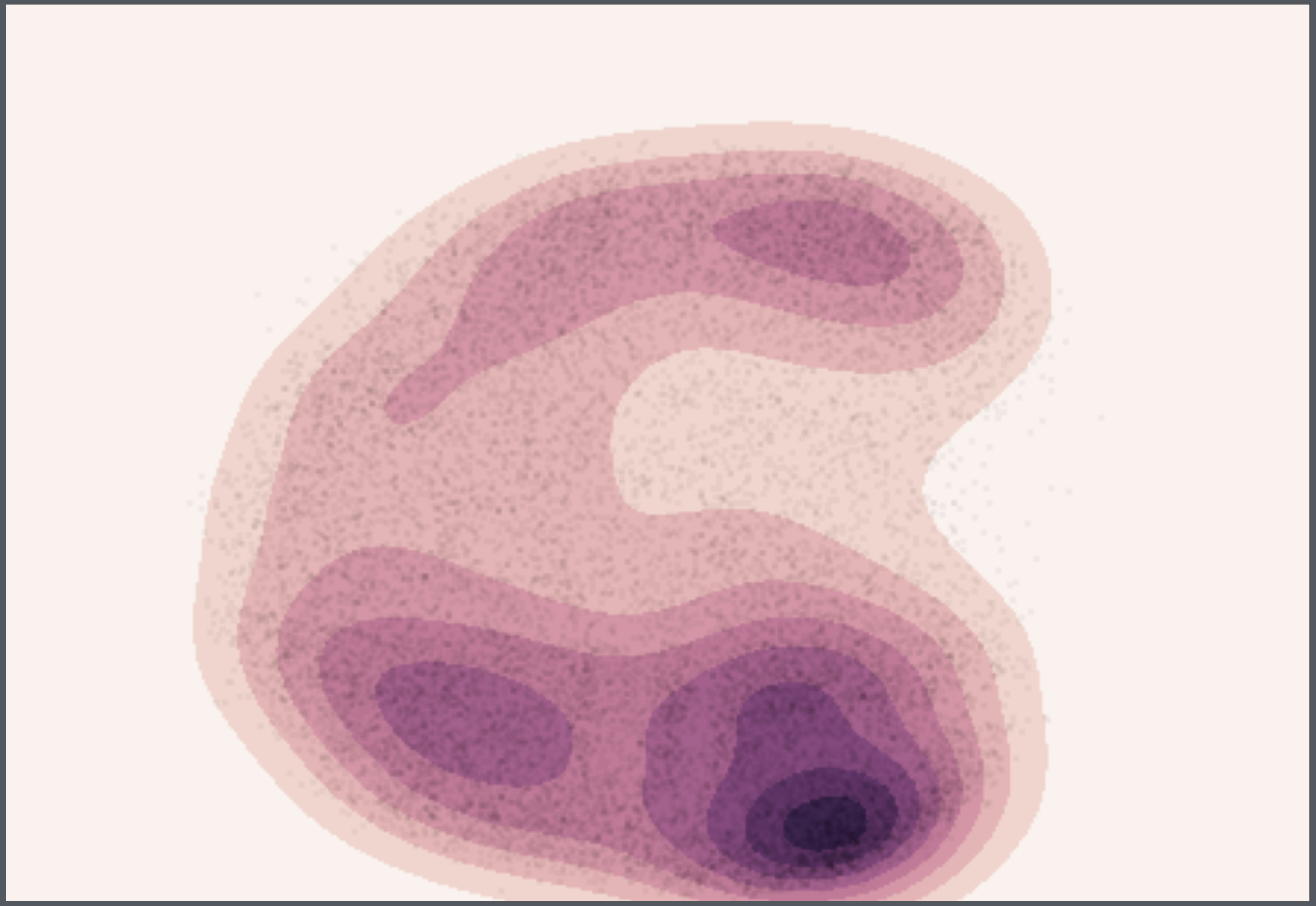
what about **LDA?**

LDA
on Client Item Descriptions



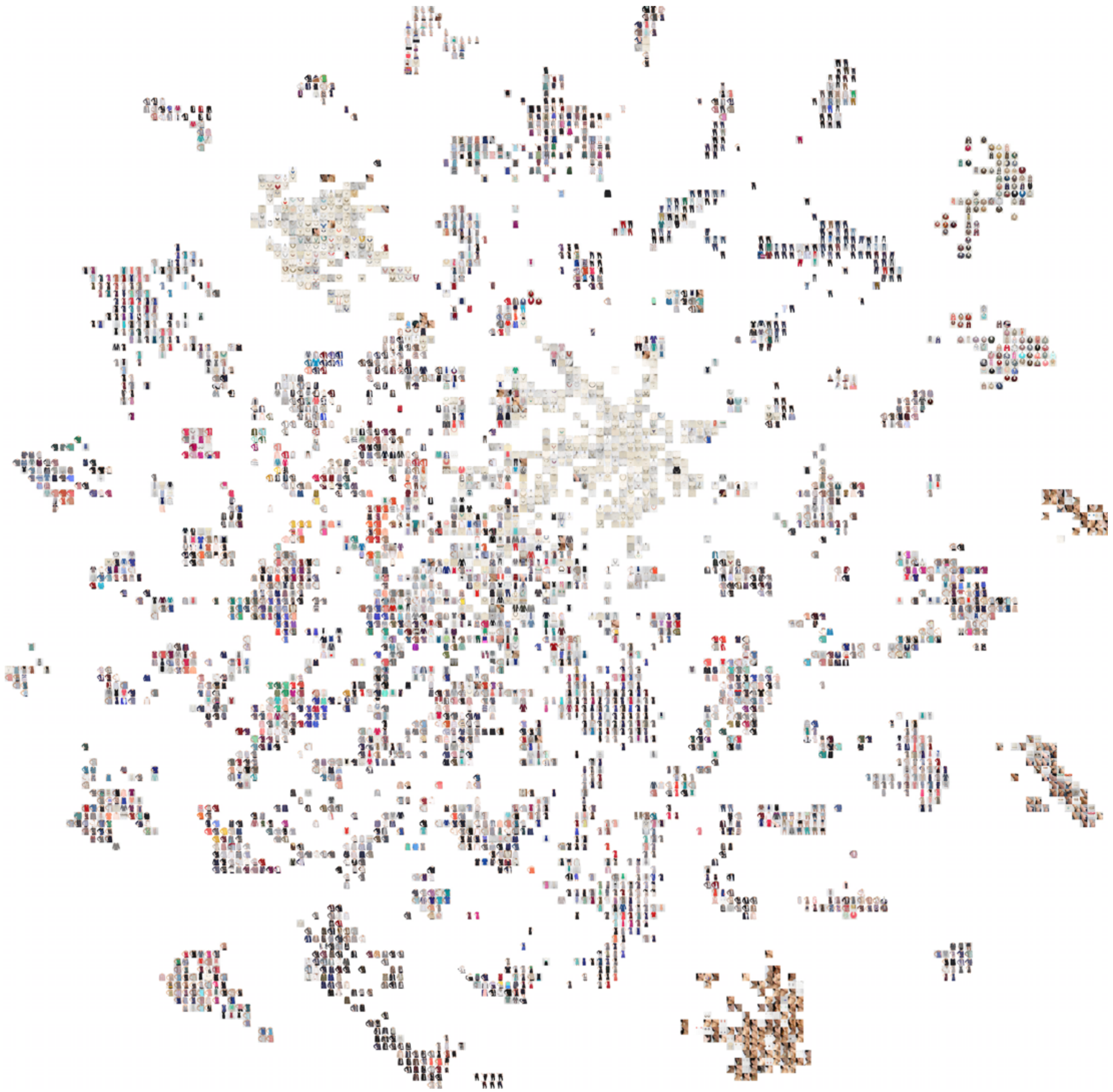


Pairwise gamma correlation from style ratings



Diversity from ratings

Latent style vectors from text



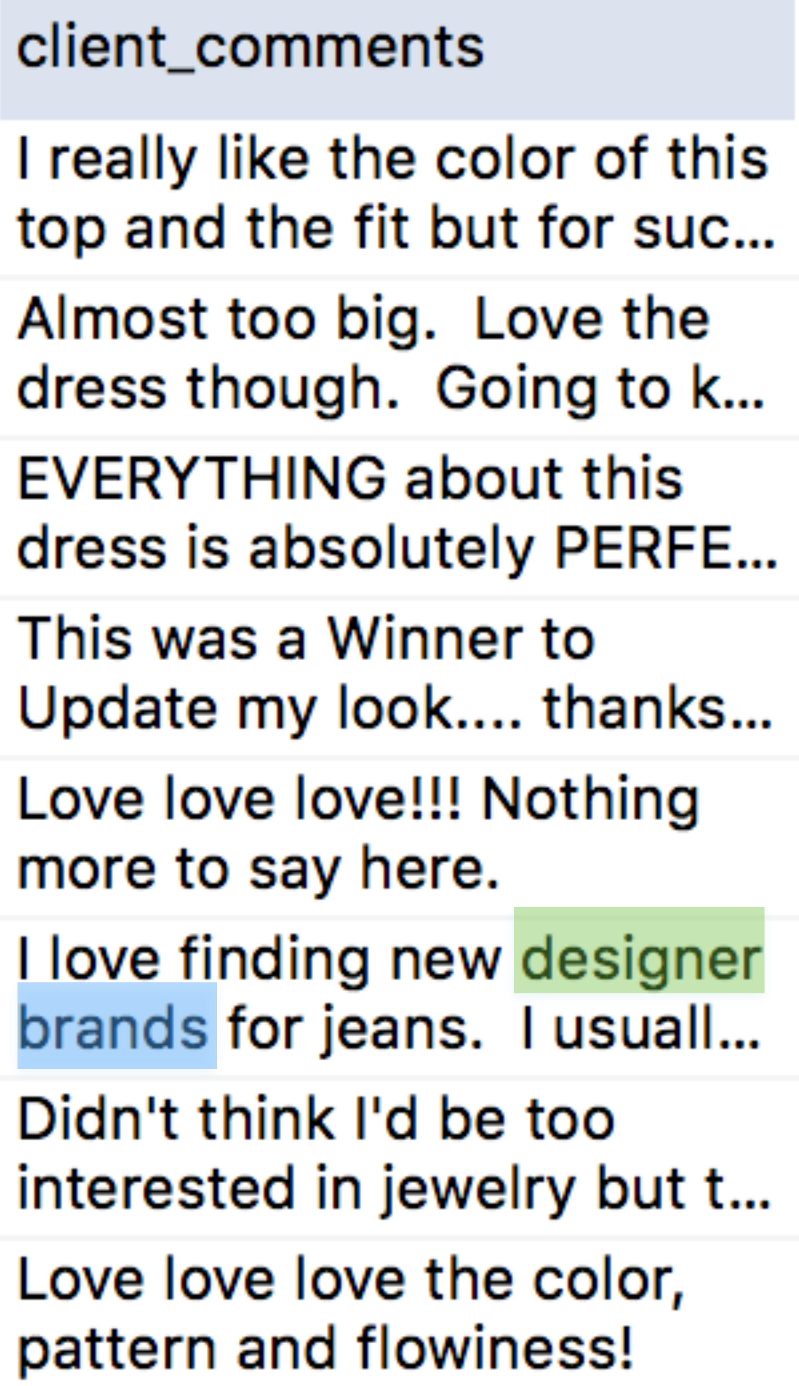
Diversity from text

lda vs **word2vec**

“I love finding new designer brands for jeans”

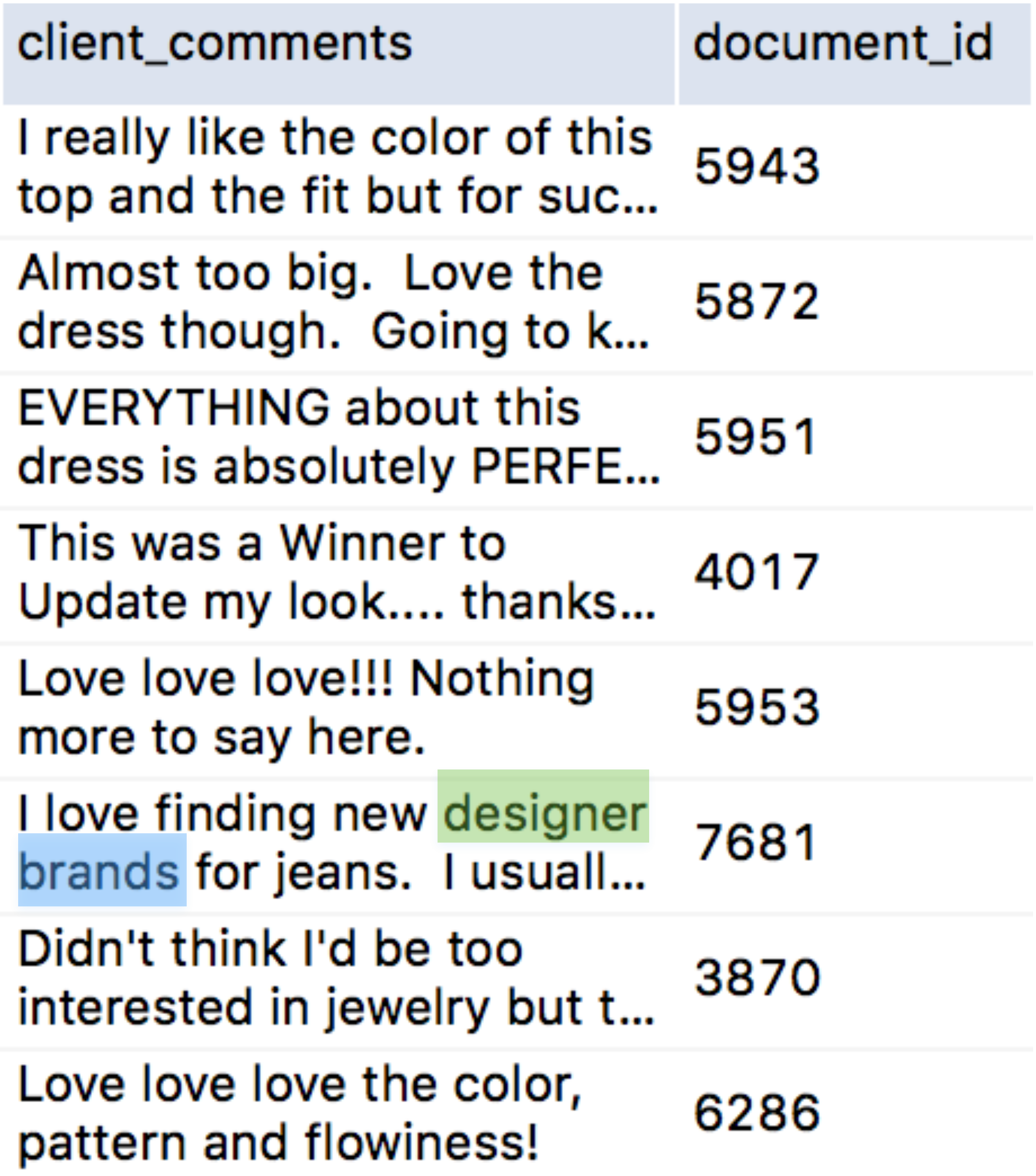
word2vec is *local:*
one word predicts a nearby word

“I love finding new designer brands for jeans”



client_comments
I really like the color of this top and the fit but for suc...
Almost too big. Love the dress though. Going to k...
EVERYTHING about this dress is absolutely PERFE...
This was a Winner to Update my look.... thanks...
Love love love!!! Nothing more to say here.
I love finding new designer brands for jeans. I usuall...
Didn't think I'd be too interested in jewelry but t...
Love love love the color, pattern and flowiness!

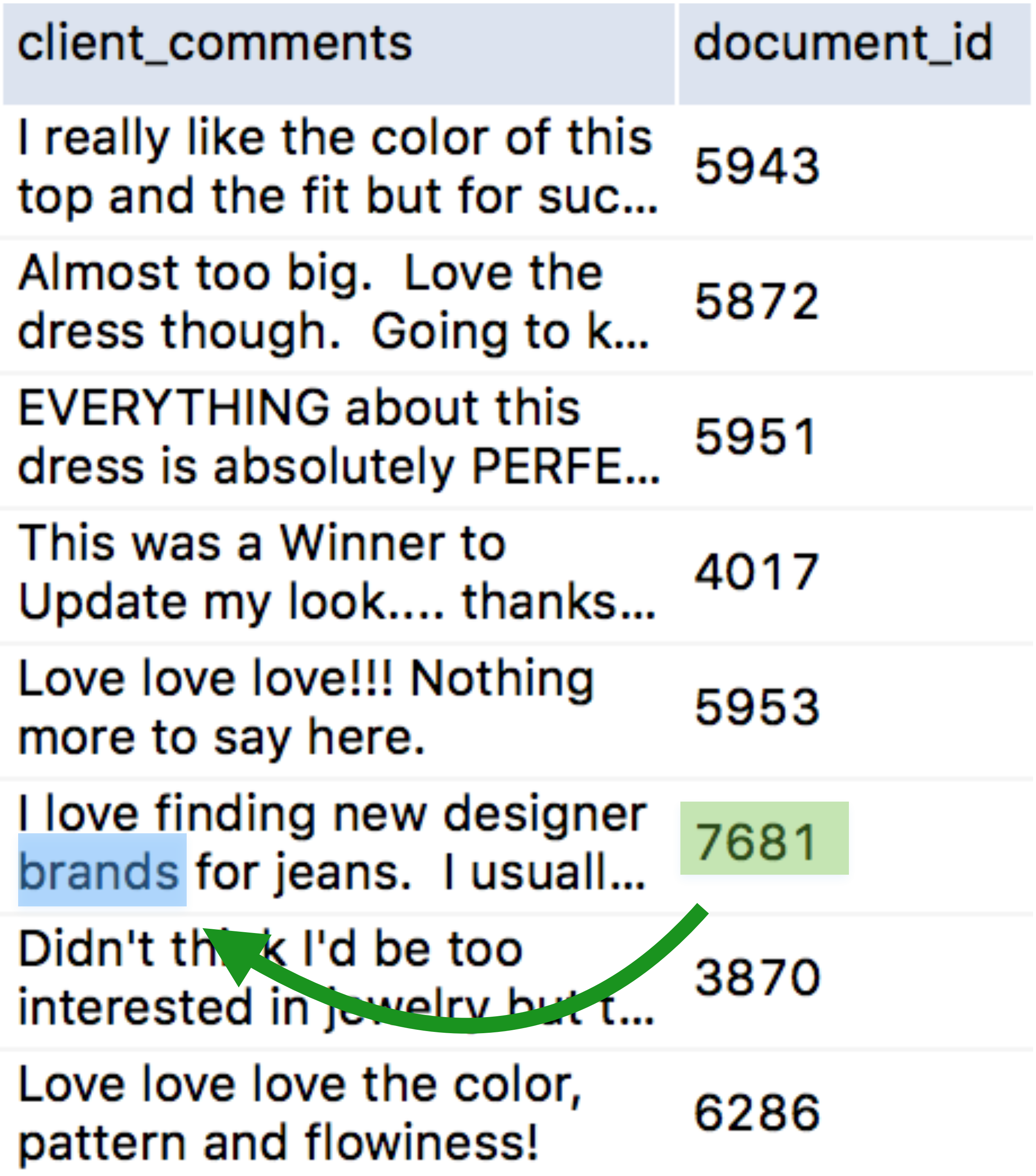
But text is usually organized.



client_comments	document_id
I really like the color of this top and the fit but for suc...	5943
Almost too big. Love the dress though. Going to k...	5872
EVERYTHING about this dress is absolutely PERFE...	5951
This was a Winner to Update my look.... thanks...	4017
Love love love!!! Nothing more to say here.	5953
I love finding new designer brands for jeans. I usuall...	7681
Didn't think I'd be too interested in jewelry but t...	3870
Love love love the color, pattern and flowiness!	6286

"I love finding new designer brands for jeans"

But text is usually organized.



client_comments	document_id
I really like the color of this top and the fit but for suc...	5943
Almost too big. Love the dress though. Going to k...	5872
EVERYTHING about this dress is absolutely PERFE...	5951
This was a Winner to Update my look.... thanks...	4017
Love love love!!! Nothing more to say here.	5953
I love finding new designer brands for jeans. I usuall...	7681
Didn't think I'd be too interested in jewelry but t...	3870
Love love love the color, pattern and flowiness!	6286

"I love finding new designer brands for jeans"

doc 7681

In LDA, documents *globally* predict words.

typical word2vec vector

[-0.75, -1.25, -0.55, -0.12, +2.2]

typical LDA document vector

[0%, 9%, 78%, 11%]

typical word2vec vector

[-0.75, -1.25, -0.55, -0.12, +2.2]

All real values

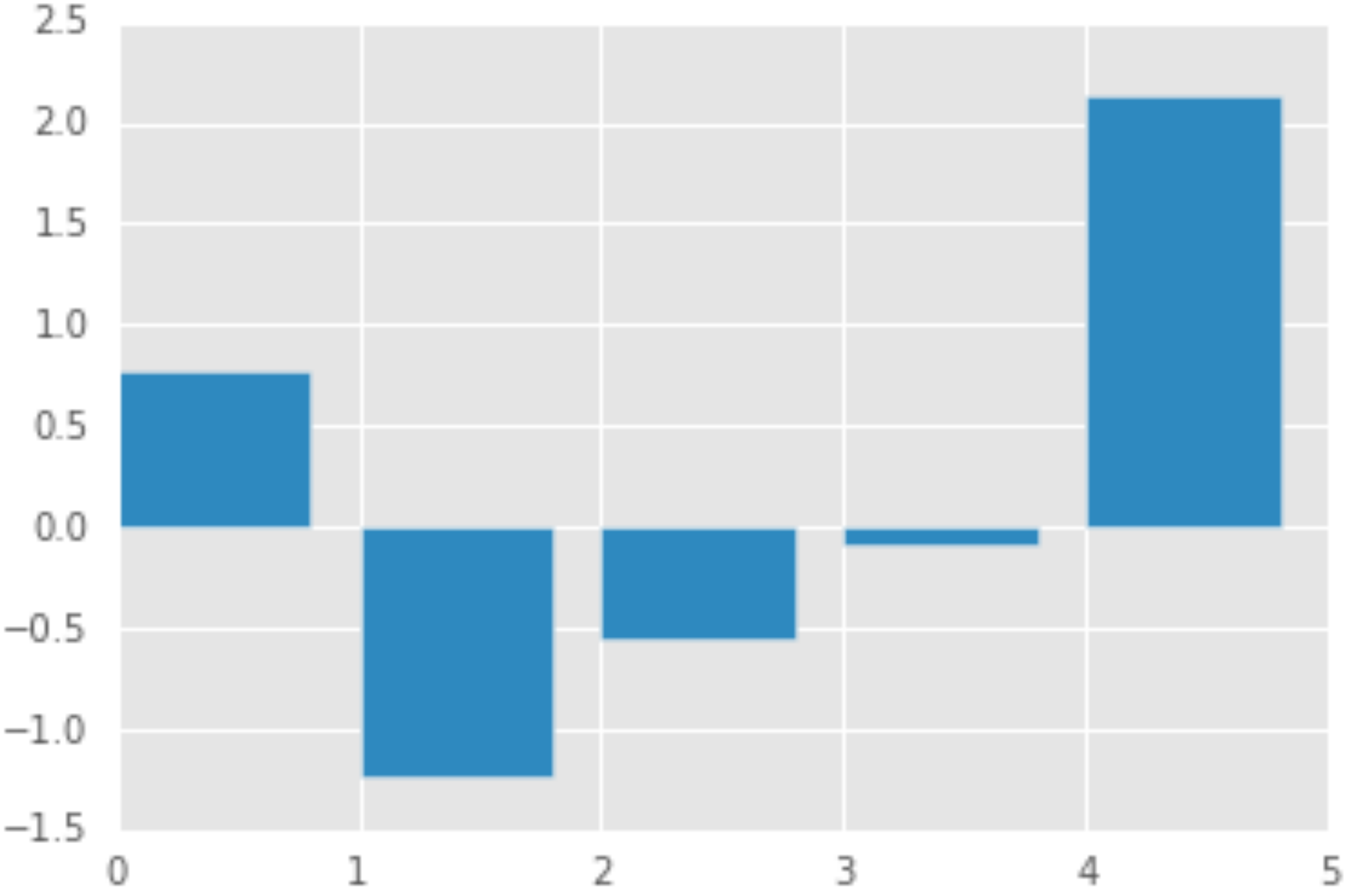
typical LDA document vector

[0%, 9%, **78%**, 11%]

All sum to 100%

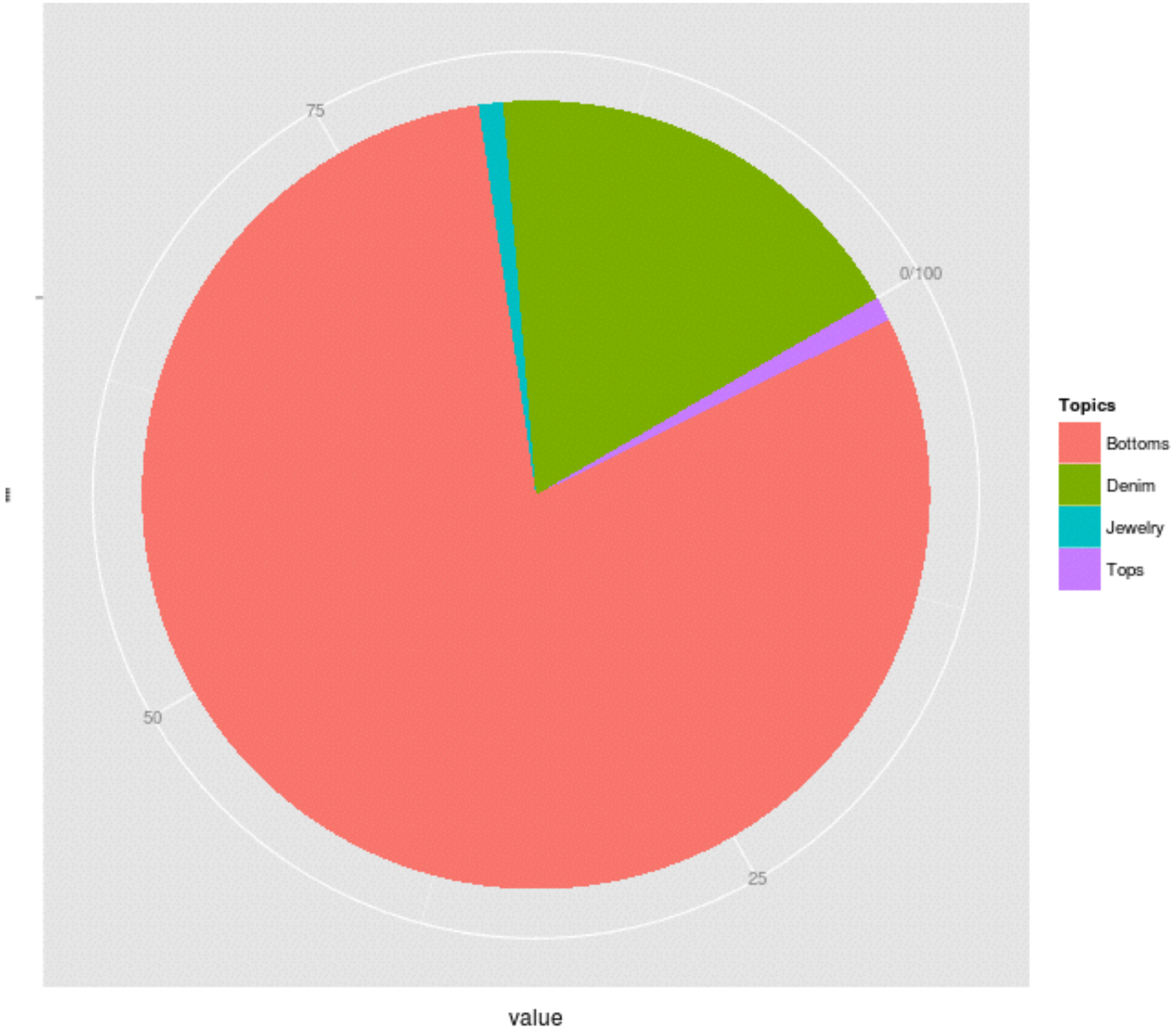
5D word2vec vector

[-0.75, -1.25, -0.55, -0.12, +2.2]



5D LDA document vector

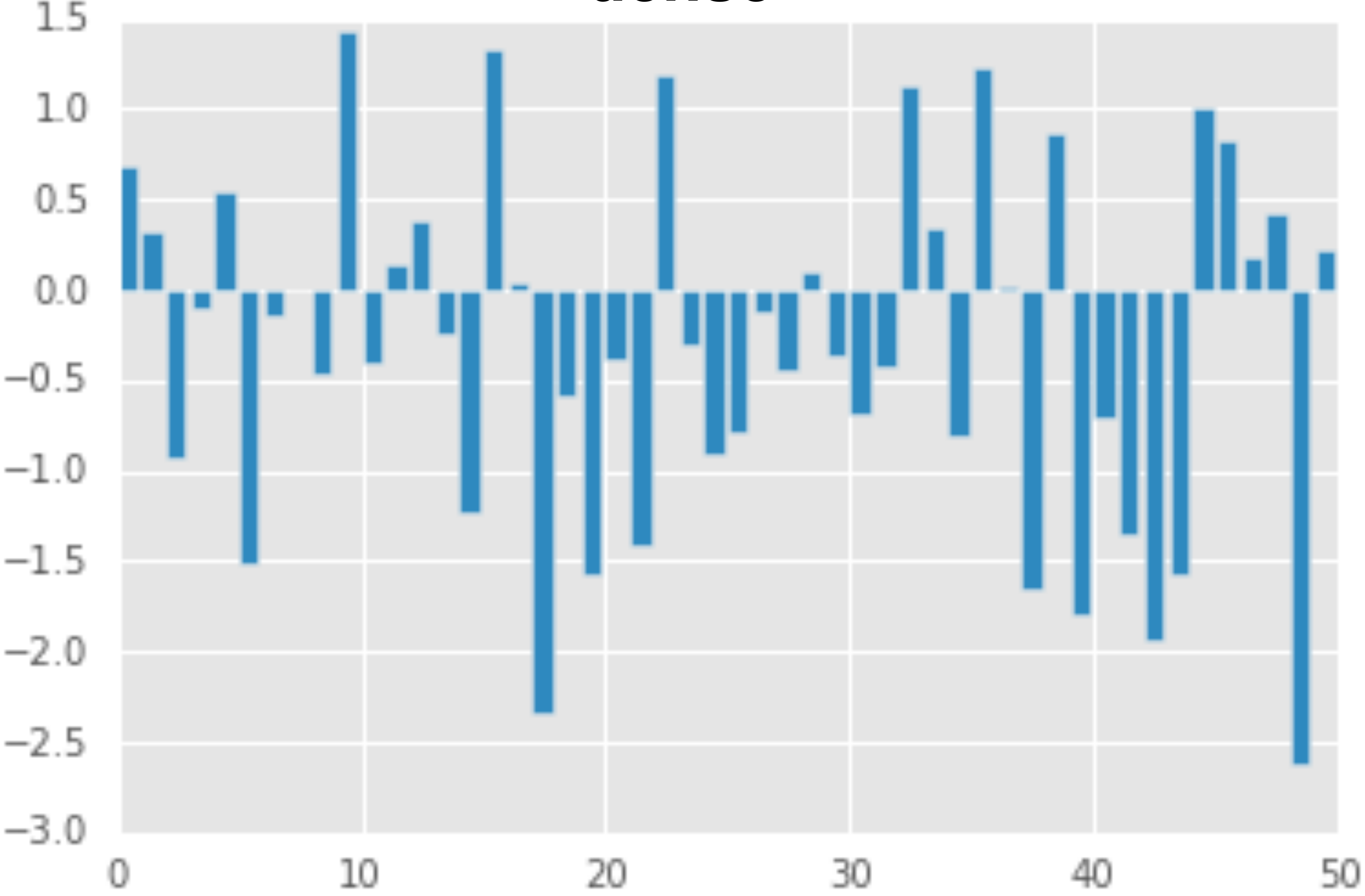
[0%, 9%, **78%**, 11%]



100D word2vec vector

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

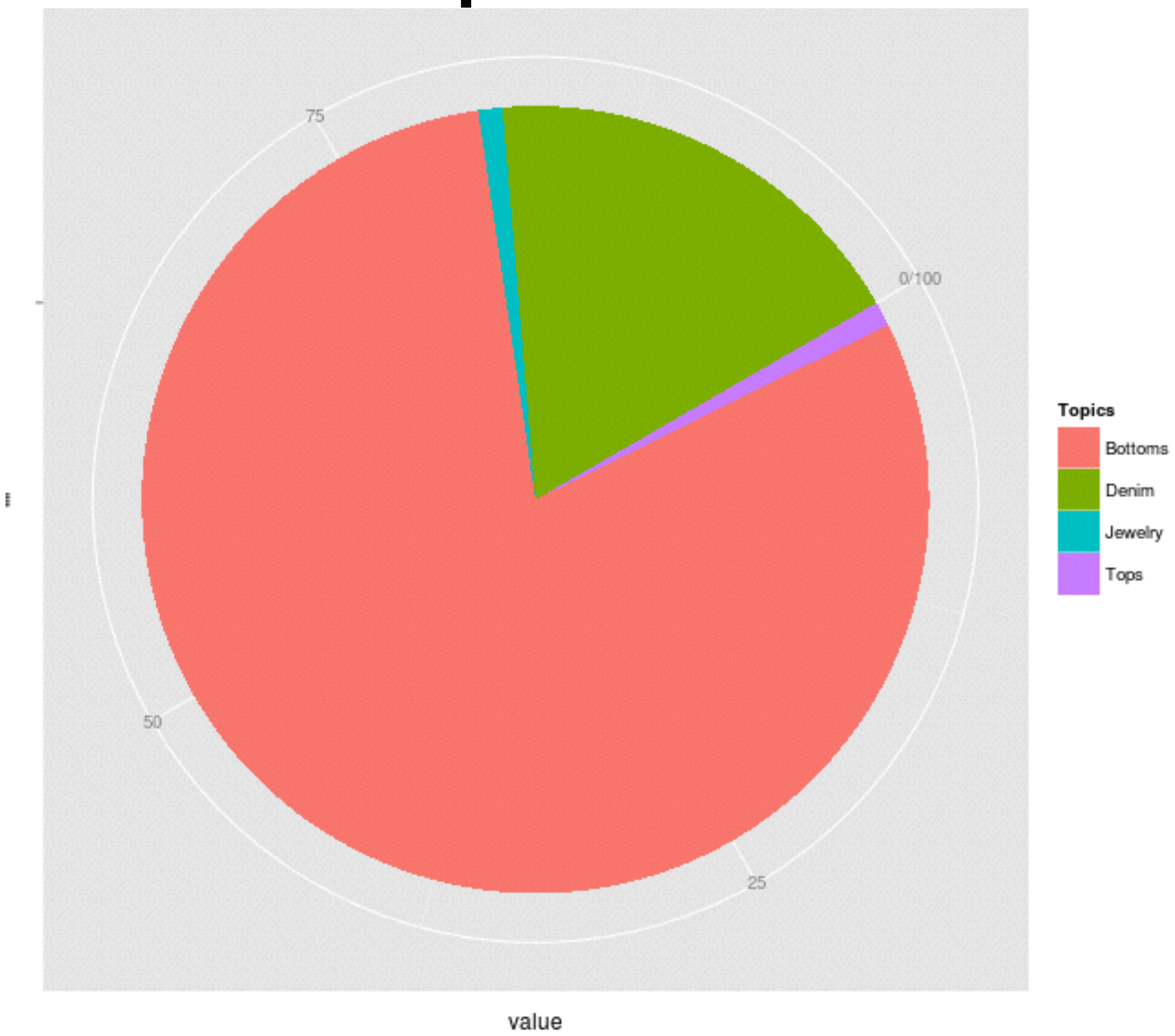
dense



100D LDA document vector

[0%0%0%0%0% … 0%, 9%,   **78%**, 11%]

sparse



100D word2vec vector

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

Similar in 100D ways
(very **flexible**)

100D LDA document vector

[0%0%0%0%0% … 0%, 9%, **78%**, 11%]

Similar in fewer ways
(more **interpretable**)

+mixture
+sparse

can we do both? **lda2vec**

The goal:
Use all of this context to learn interpretable topics.

client_comments

I love finding new designer brands for jeans. I usuall...

Didn't think I'd be too interested in jewelry but t...

word2vec $P(v_{OUT} | v_{IN})$

The goal:
Use all of this context to learn interpretable topics.

client_comments	document_id
[redacted]	5943
[redacted]	5872
[redacted]	5951
[redacted]	4017
[redacted]	5953
I love finding new designer brands for jeans. I usuall...	7681
Didn't think I'd be too interested in jewelry but t...	3870
[redacted]	6286

word2vec

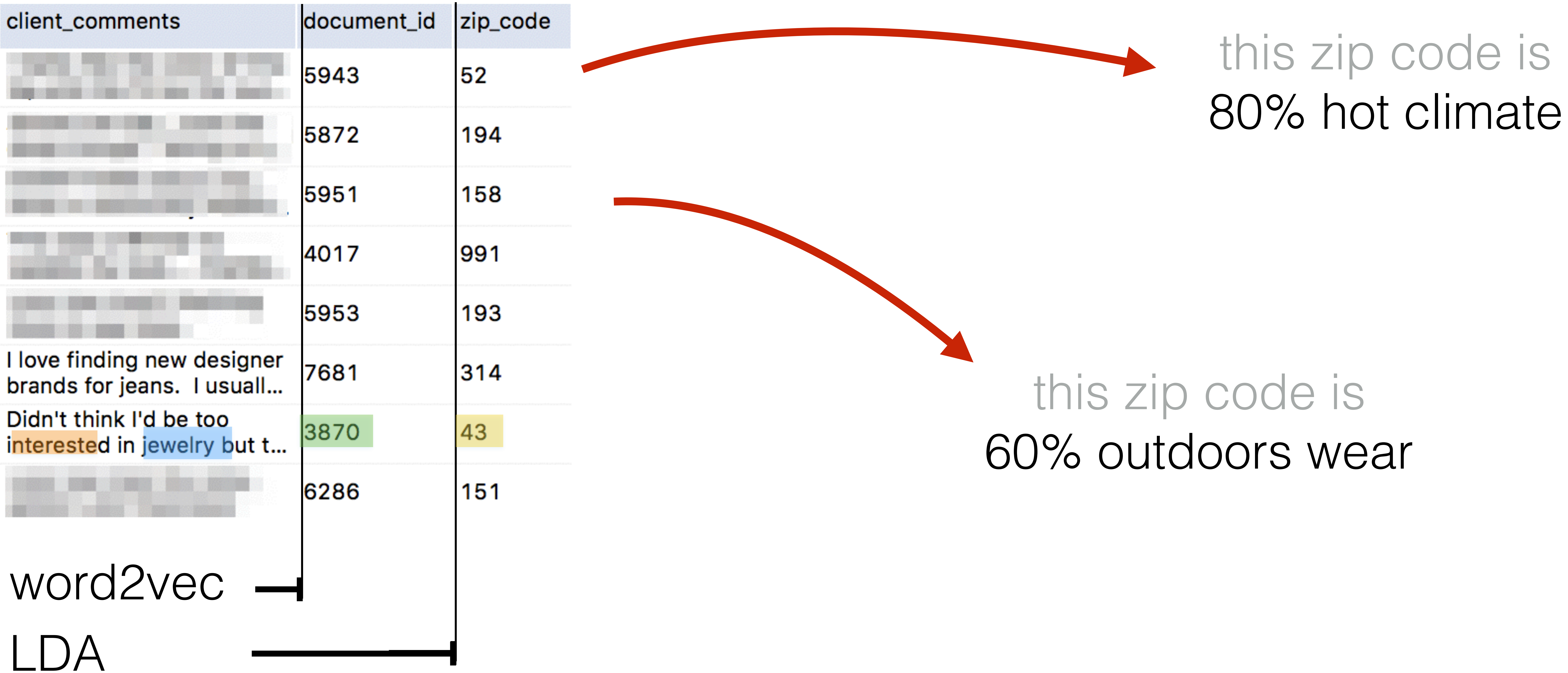
LDA

$$P(v_{OUT} | v_{DOC})$$

this document is
80% high fashion

this document is
60% style

The goal:
Use all of this context to learn interpretable topics.



client_comments	document_id	zip_code
[illegible]	5943	52
[illegible]	5872	194
[illegible]	5951	158
[illegible]	4017	991
[illegible]	5953	193
I love finding new designer brands for jeans. I usuall...	7681	314
Didn't think I'd be too interested in jewelry but t...	3870	43
[illegible]	6286	151

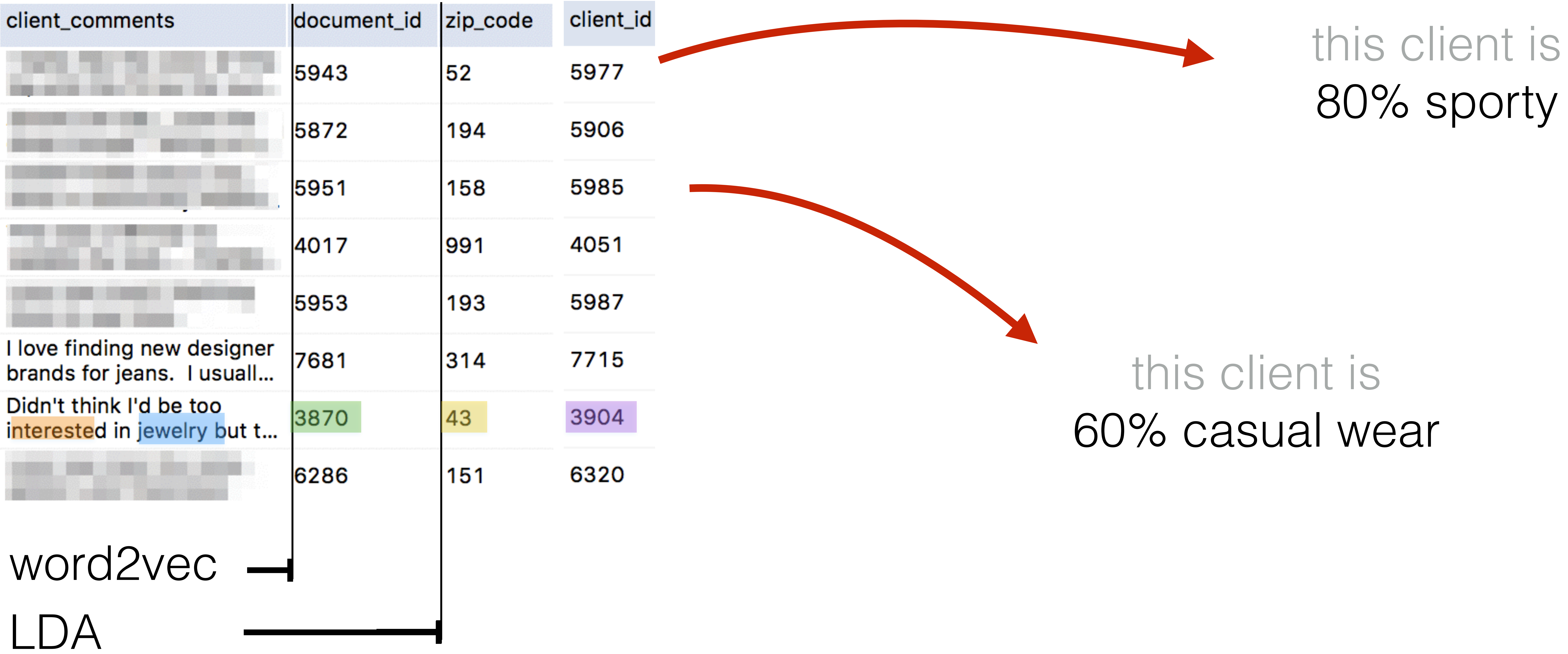
word2vec

LDA

this zip code is
80% hot climate

this zip code is
60% outdoors wear

The goal:
Use all of this context to learn interpretable topics.



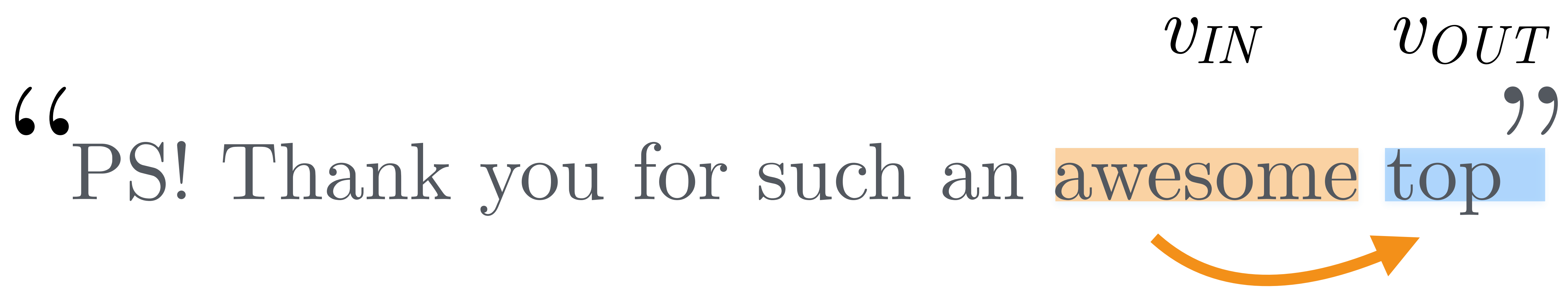
client_comments	document_id	zip_code	client_id
[redacted]	5943	52	5977
[redacted]	5872	194	5906
[redacted]	5951	158	5985
[redacted]	4017	991	4051
[redacted]	5953	193	5987
I love finding new designer brands for jeans. I usuall...	7681	314	7715
Didn't think I'd be too interested in jewelry but t...	3870	43	3904
[redacted]	6286	151	6320

this client is
80% sporty

this client is
60% casual wear

word2vec

LDA

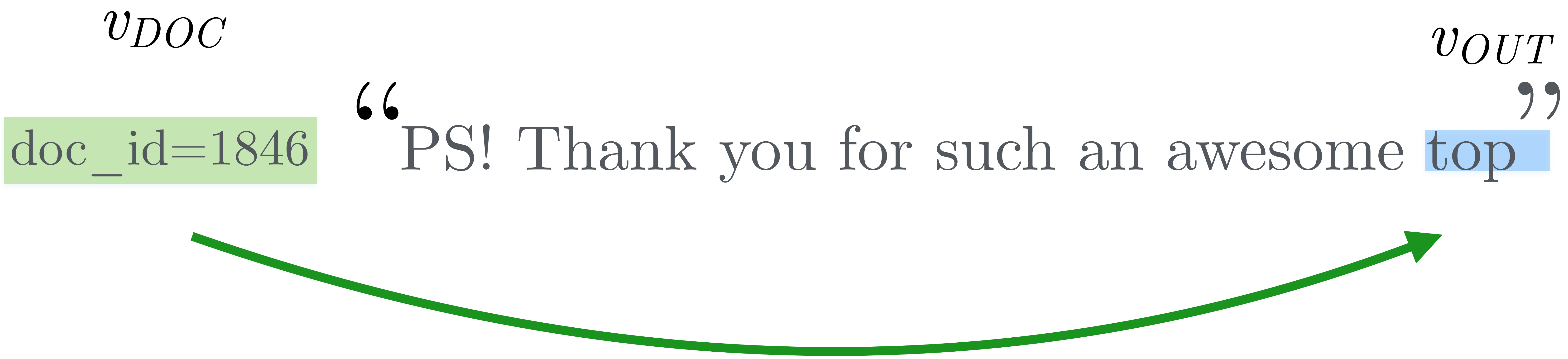


v_{IN} v_{OUT}

“PS! Thank you for such an awesome top”

word2vec predicts *locally:*
one word predicts a nearby word

$$P(v_{OUT}|v_{IN})$$



v_{DOC}

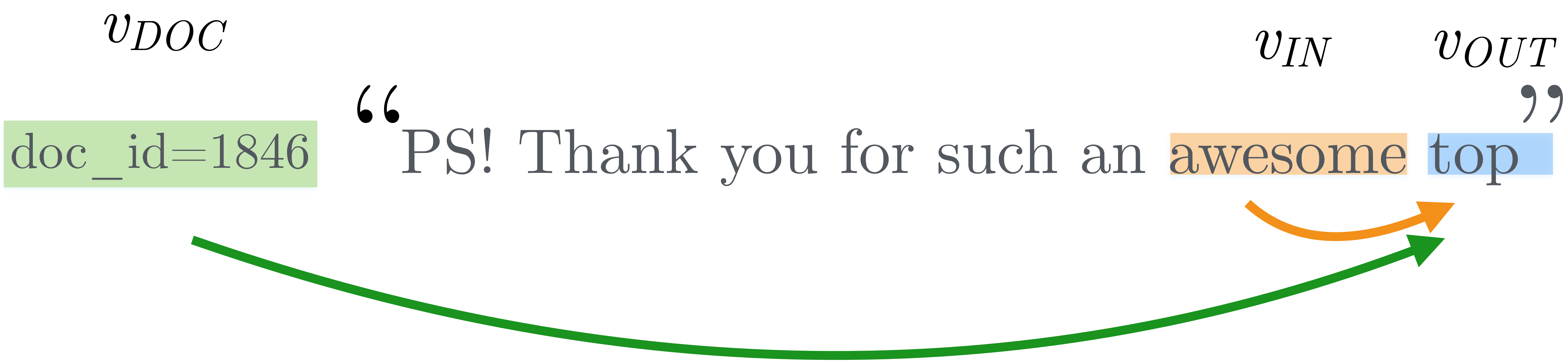
doc_id=1846

v_{OUT}

“PS! Thank you for such an awesome top”

LDA predicts a word from a *global* context

$$P(v_{OUT}|v_{DOC})$$



v_{DOC} v_{IN} v_{OUT}

doc_id=1846 “PS! Thank you for such an awesome top”

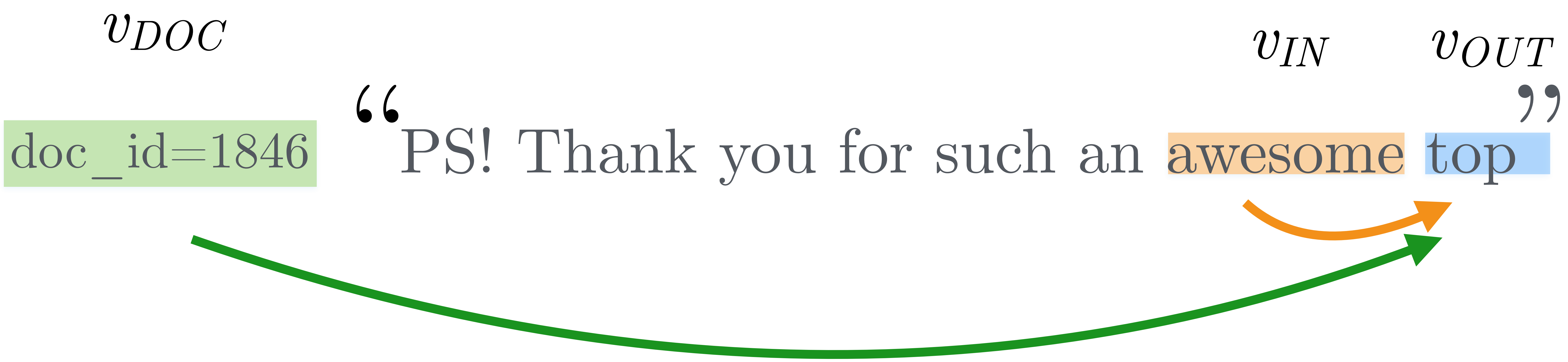
can we predict a word both *locally* and *globally* ?

v_{DOC} v_{IN} v_{OUT}

doc_id=1846 "PS! Thank you for such an awesome top"

can we predict a word both *locally* and *globally* ?

$$P(v_{OUT} | v_{IN} + v_{DOC})$$



v_{DOC} v_{IN} v_{OUT}

doc_id=1846 “PS! Thank you for such an awesome top”

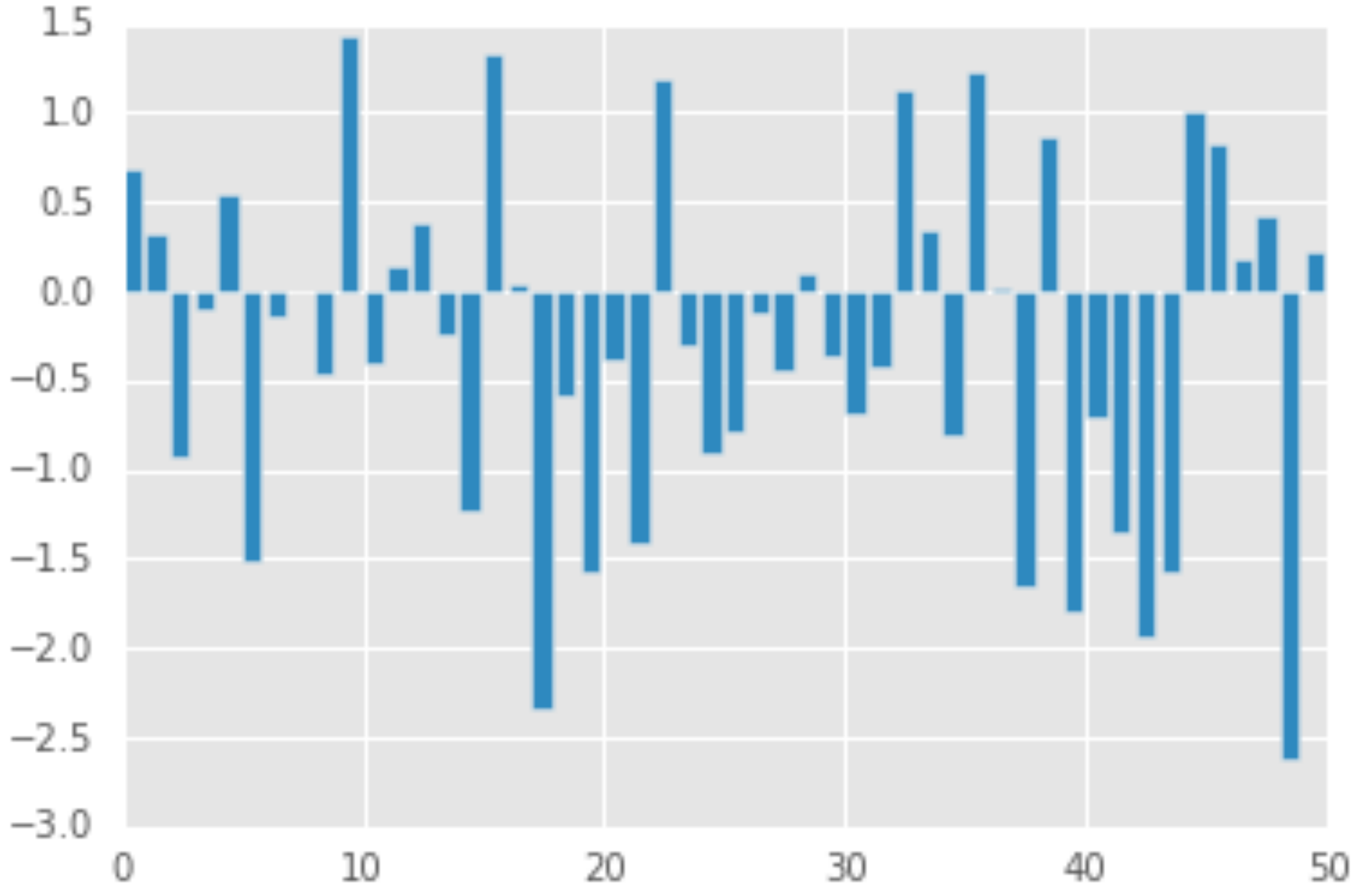
can we predict a word both *locally* and *globally* ?

$$P(v_{OUT}|v_{IN}+\boxed{v_{DOC})}$$

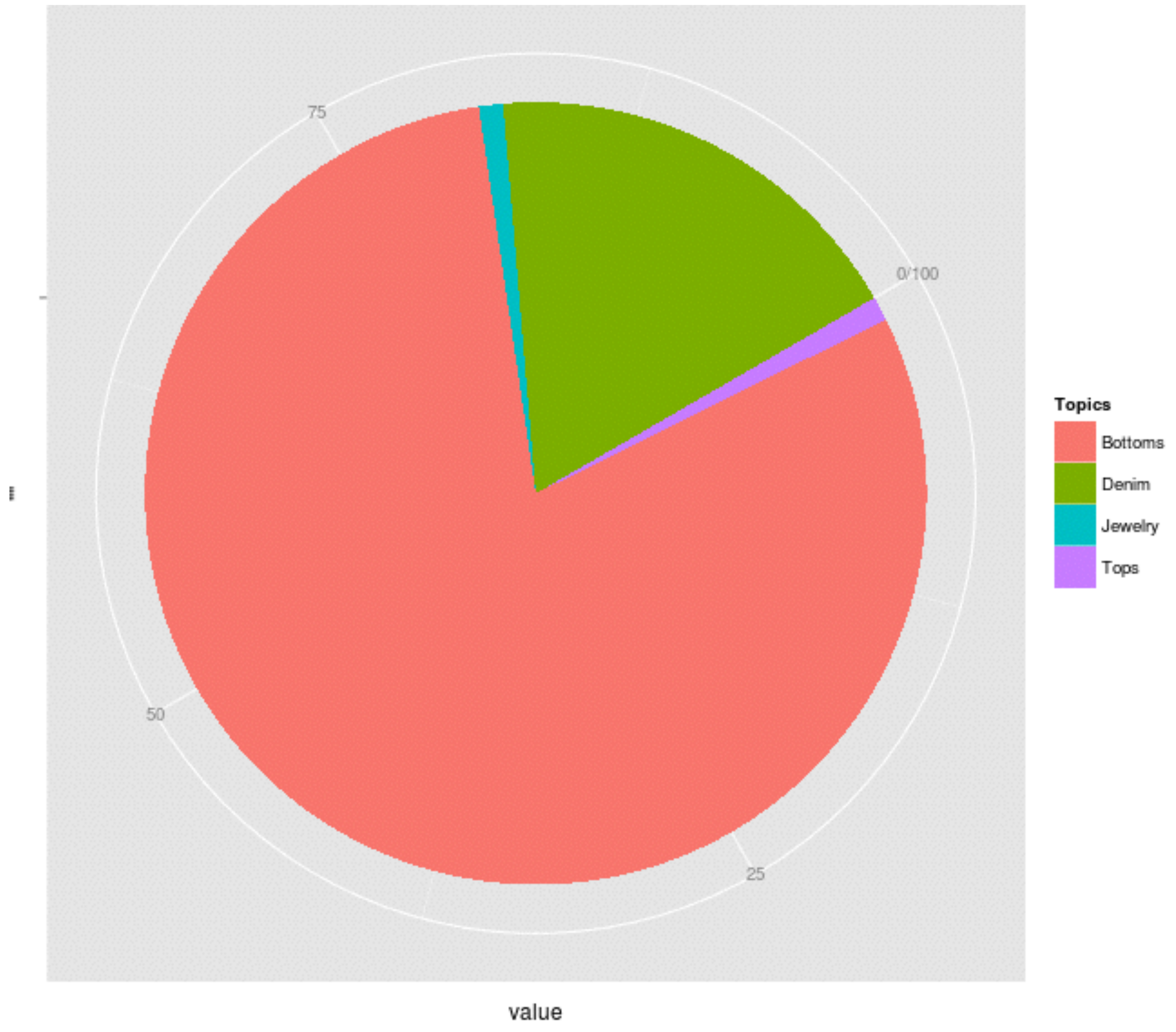
*very similar to the Paragraph Vectors / doc2vec

This works! 😀 But v_{DOC} isn’t as interpretable as the LDA topic vectors. 😔

This works! 😀 But v_{DOC} isn’t as interpretable as the LDA topic vectors. 😔



This works! 😀 But v_{DOC} isn't as interpretable as the LDA topic vectors. 😔



This works! 😀 But v_{DOC} isn’t as interpretable as the LDA topic vectors. 😔

We’re missing *mixtures* & *sparsity.*

This works! 😀 But v_{DOC} isn’t as interpretable as the LDA topic vectors. 😔

Let’s make v_{DOC} into a mixture…

Let's make $\boldsymbol{v_{DOC}}$ into a mixture…

$$v_{DOC} = a\ v_{topic1} + b\ v_{topic2} + \ldots$$

(up to k topics)

Let’s make v_{DOC} into a mixture…

$$v_{DOC} = a \; v_{topic1} + b \; v_{topic2} + \ldots$$

Trinitarian
baptismal
Pentecostals
Bede
schismatics
excommunication

Let’s make $\boldsymbol{v_{DOC}}$ into a mixture…

topic 1 = “religion”

Trinitarian
baptismal
Pentecostals
Bede
schismatics
excommunication

$$v_{DOC} = a\ v_{topic1} + b\ v_{topic2} + \ldots$$

Let's make v_{DOC} into a mixture…

topic 1 = "religion"

Trinitarian
baptismal
Pentecostals
Bede
schismatics
excommunication

$$v_{DOC} = a\ v_{topic1} + b\ v_{topic2} + \ldots$$

Milosevic
absentee
Indonesia
Lebanese
Isrealis
Karadzic

Let's make v_{DOC} into a mixture…

$$v_{DOC} = a \; v_{topic1} + b \; v_{topic2} + \ldots$$

topic 1 = "religion"

Trinitarian
baptismal
Pentecostals
bede
schismatics
excommunication

topic 2 = "politics"

Milosevic
absentee
Indonesia
Lebanese
Isrealis
Karadzic

Let's make v_{DOC} into a mixture…

topic 1 = "religion"

Trinitarian
baptismal
Pentecostals
bede
schismatics
excommunication

$$v_{DOC} = \mathit{10\%}\ religion + \mathit{89\%}\ politics + \ldots$$

topic 2 = "politics"

Milosevic
absentee
Indonesia
Lebanese
Isrealis
Karadzic

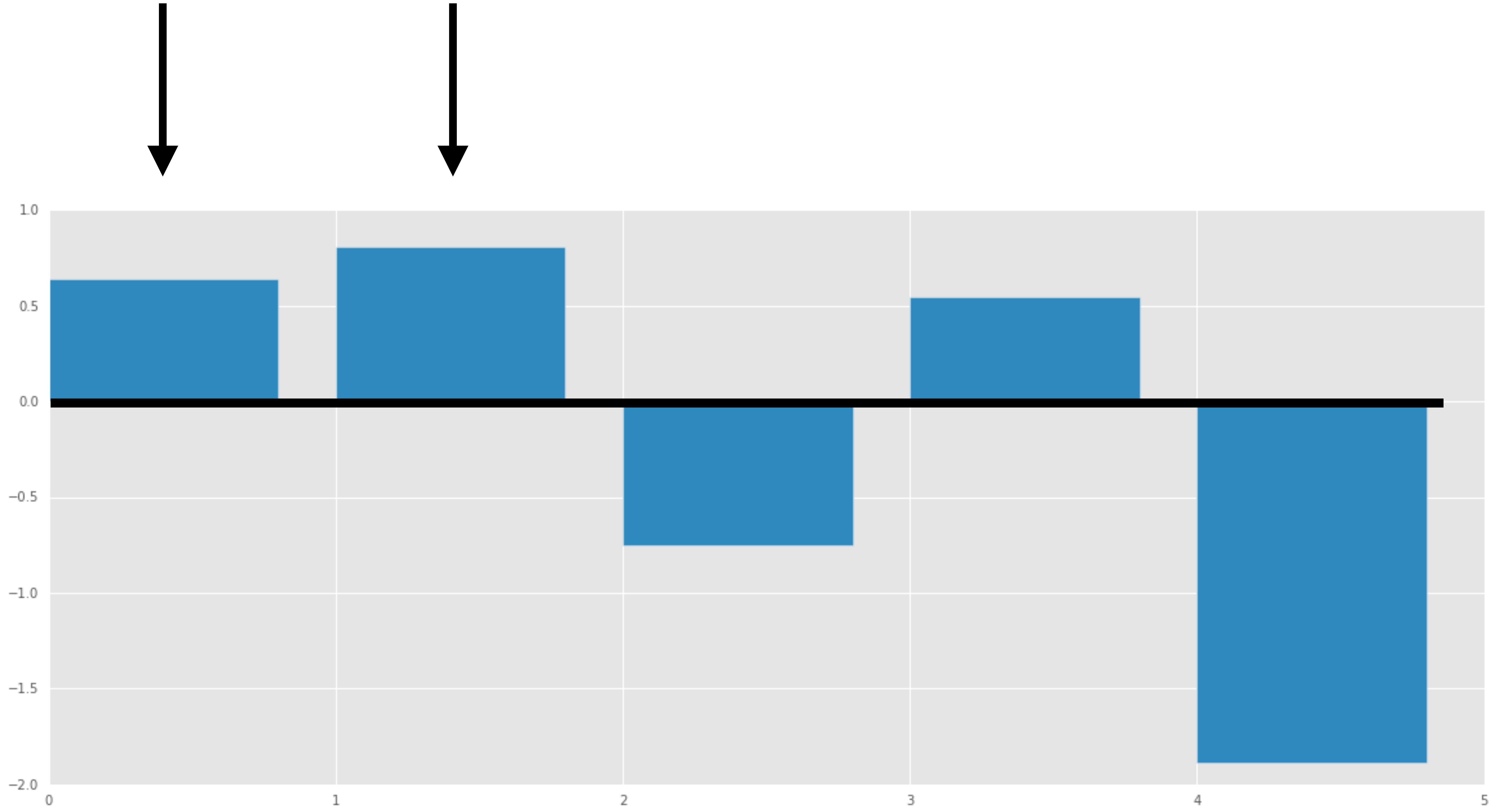
Let's make v_{DOC} *sparse*

$$v_{DOC} = a \; v_{religion} + b \; v_{politics} + \ldots$$

[-0.75, -1.25, ...]

Let's make v_{DOC} *sparse*

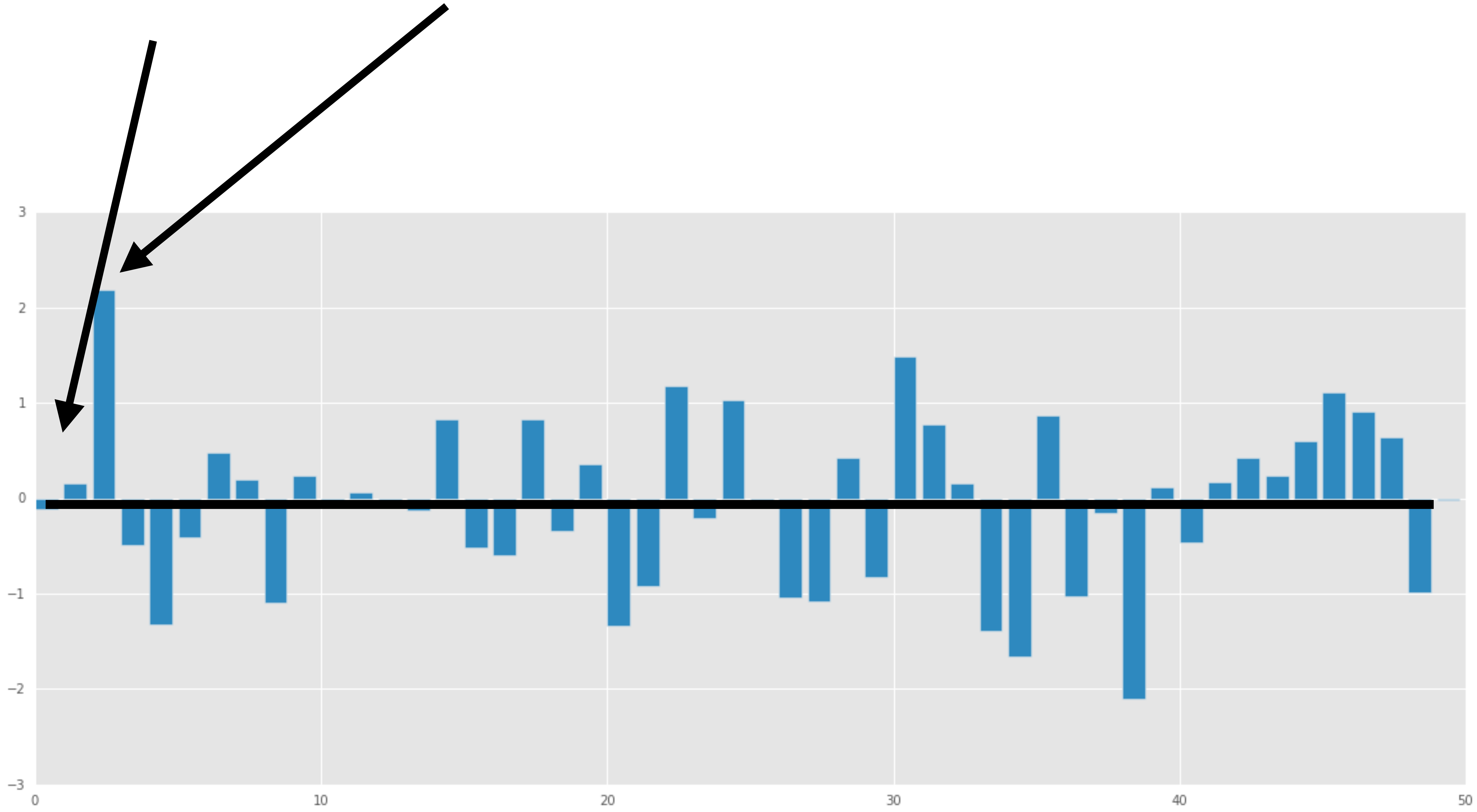
$$v_{DOC} = a \ v_{religion} + b \ v_{politics} + \ldots$$



lda2vec

Let's make v_{DOC} *sparse*

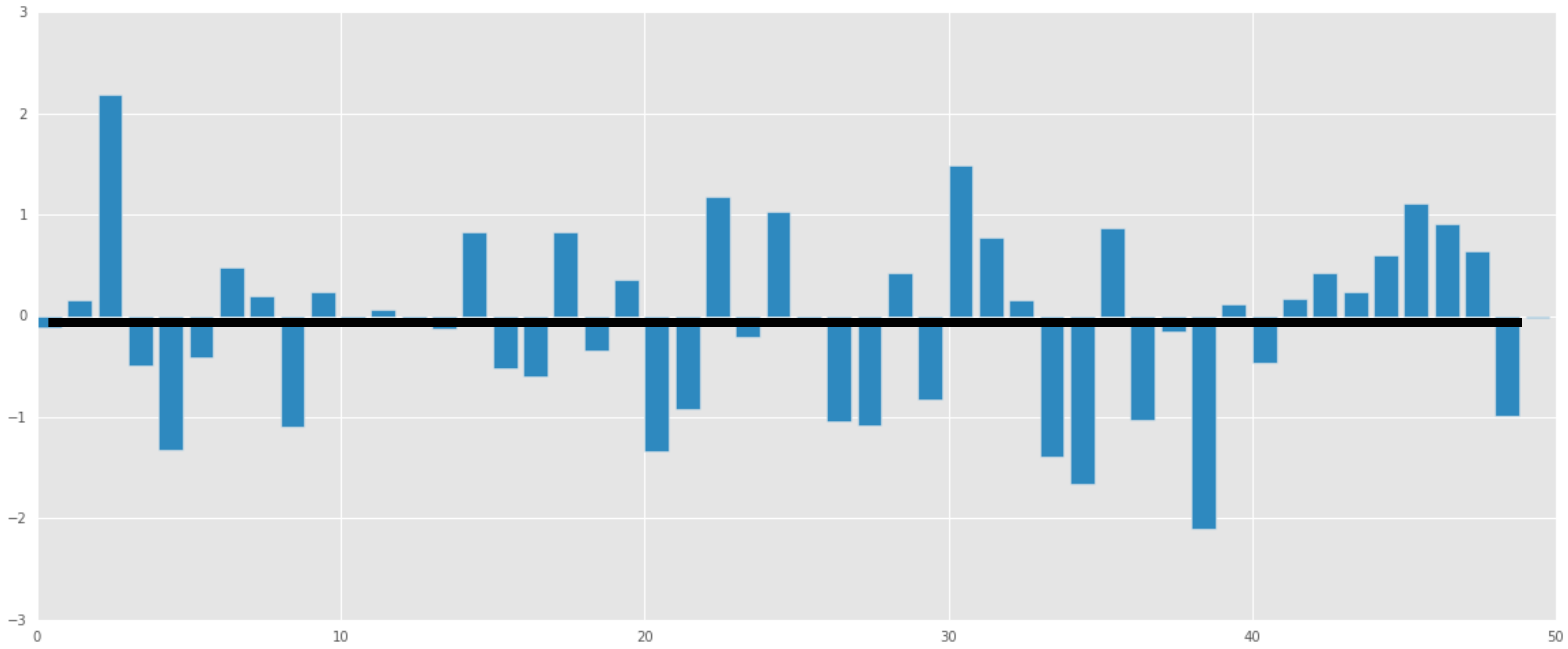
$$v_{DOC} = a\ v_{religion} + b\ v_{politics} + \ldots$$



Let's make v_{DOC} *sparse*

$$v_{DOC} = a \; v_{religion} + b \; v_{politics} + \ldots$$

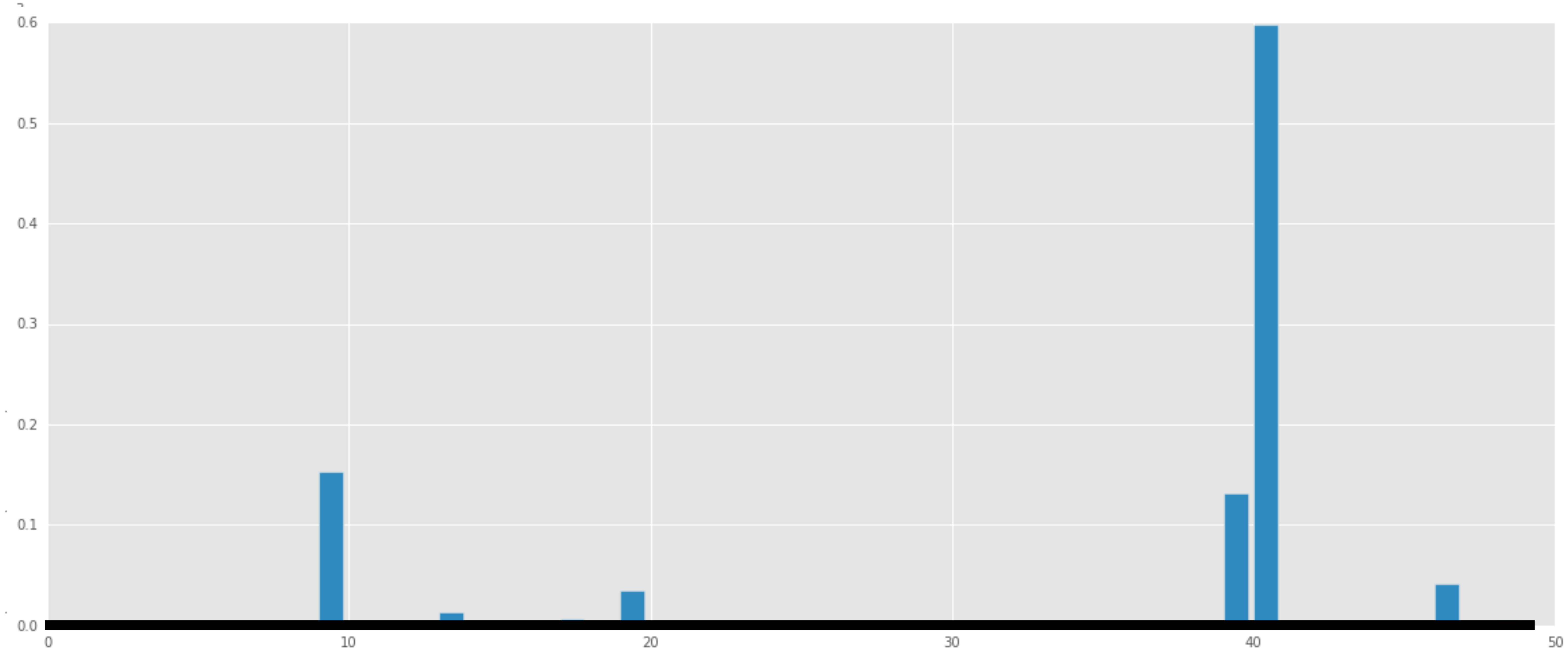
$$\{a, \; b, \; c\ldots\} \sim dirichlet(alpha)$$



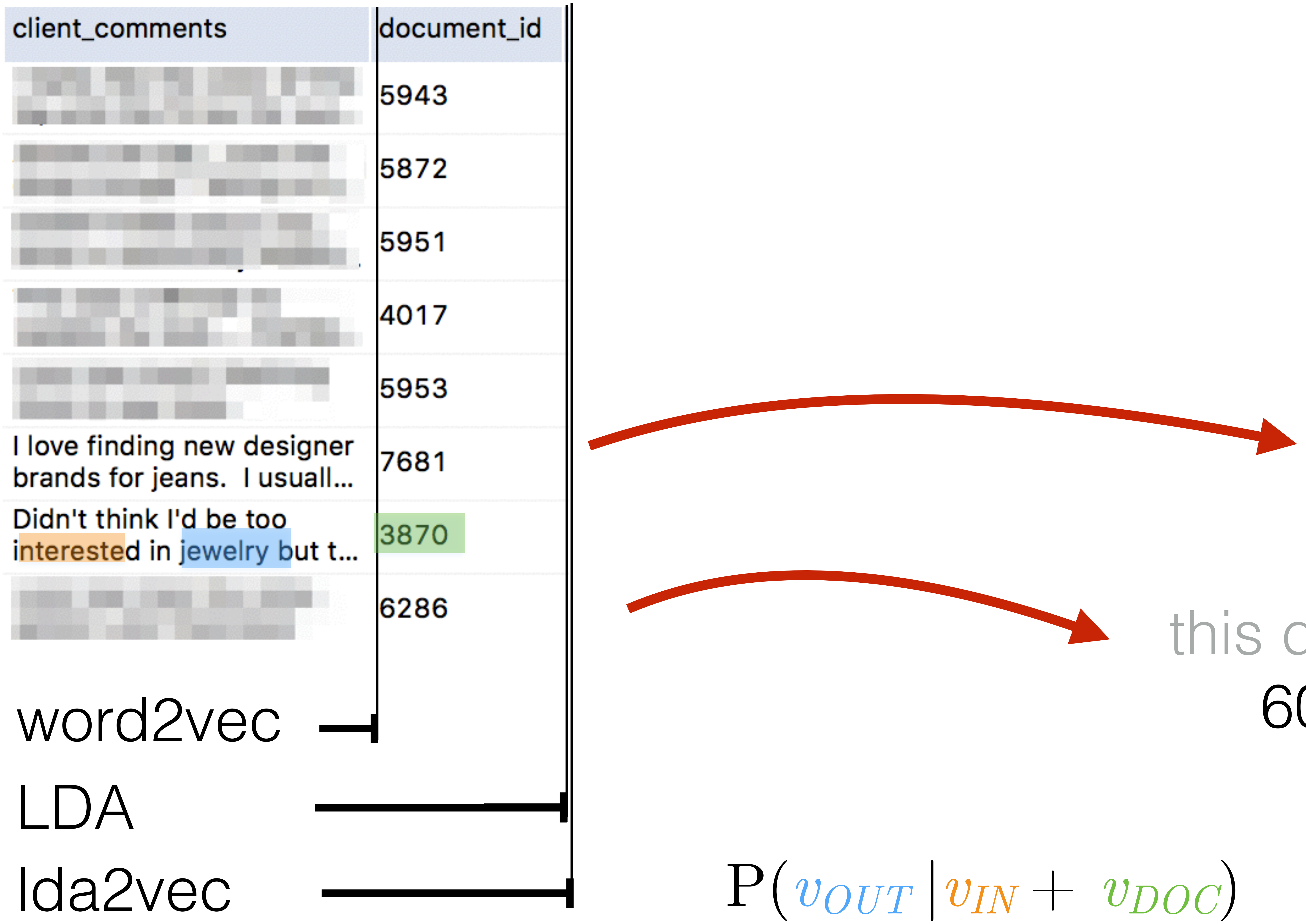
Let's make v_{DOC} *sparse*

$$v_{DOC} = a\ v_{religion} + b\ v_{politics} + \ldots$$

$$\{a,\ b,\ c\ldots\} \sim dirichlet(alpha)$$



The goal:
Use all of this context to learn interpretable topics.



client_comments	document_id
[redacted]	5943
[redacted]	5872
[redacted]	5951
[redacted]	4017
[redacted]	5953
I love finding new designer brands for jeans. I usuall...	7681
Didn't think I'd be too interested in jewelry but t...	3870
[redacted]	6286

word2vec

LDA

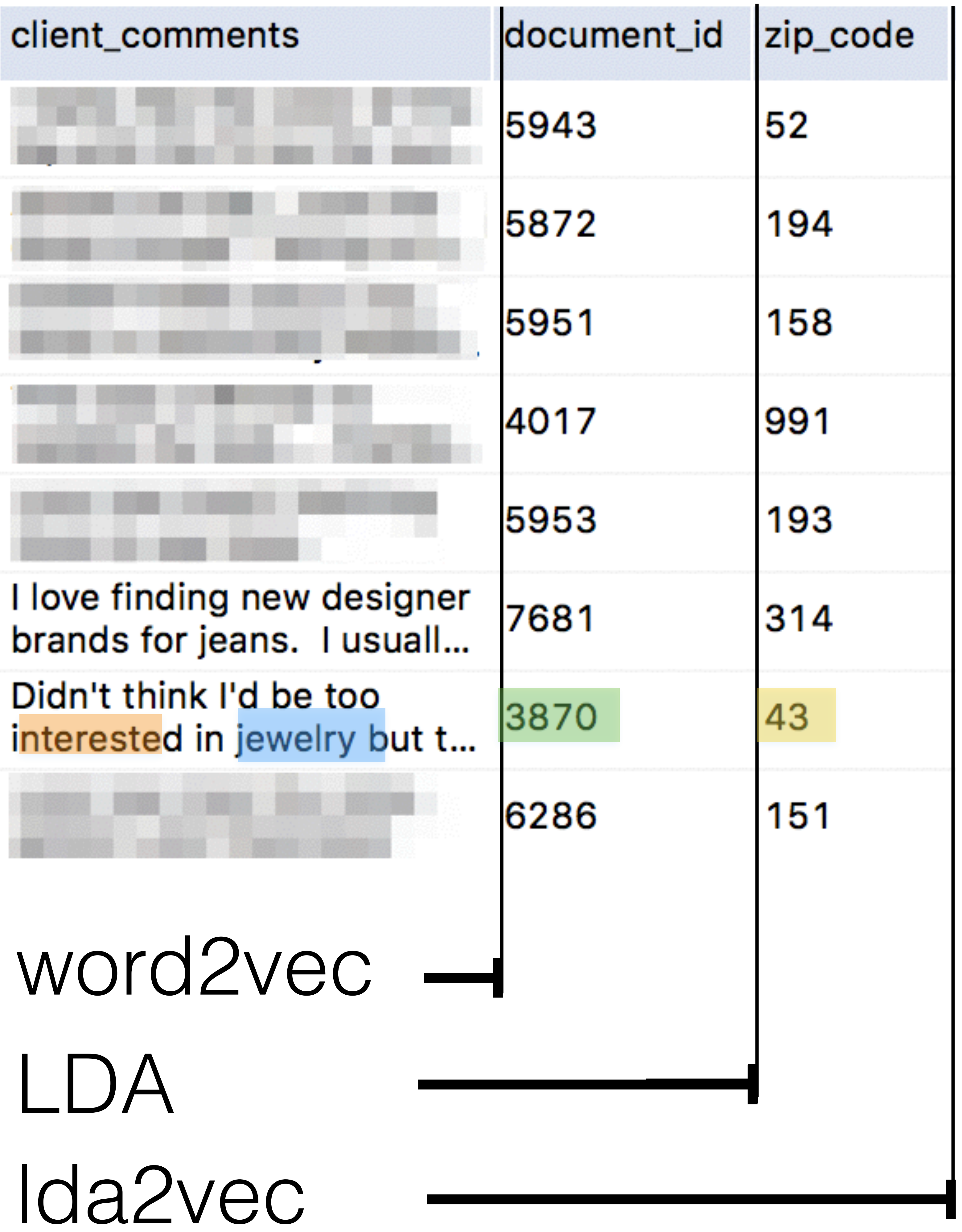
lda2vec

this document is
80% high fashion

this document is
60% style

$$P(v_{OUT} \mid v_{IN} + v_{DOC})$$

The goal:
Use all of this context to learn interpretable topics.



client_comments	document_id	zip_code
[redacted]	5943	52
[redacted]	5872	194
[redacted]	5951	158
[redacted]	4017	991
[redacted]	5953	193
I love finding new designer brands for jeans. I usuall...	7681	314
Didn't think I'd be too interested in jewelry but t...	3870	43
[redacted]	6286	151

word2vec

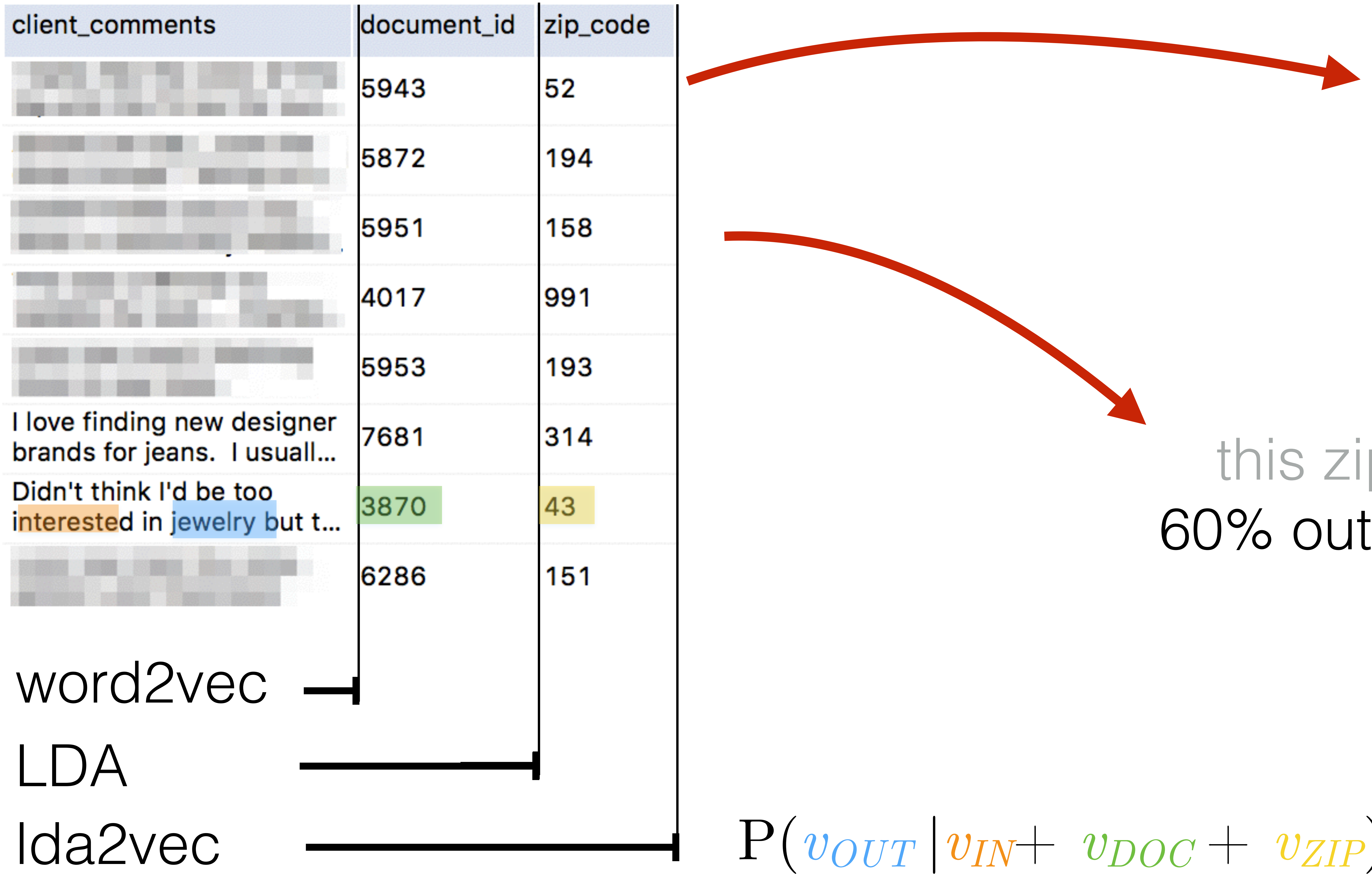
LDA

lda2vec

$$P(v_{OUT} \mid v_{IN} + v_{DOC} \boxed{+ v_{ZIP}})$$

@chrisemoody

The goal: Use all of this context to learn interpretable topics.



client_comments	document_id	zip_code
[redacted]	5943	52
[redacted]	5872	194
[redacted]	5951	158
[redacted]	4017	991
[redacted]	5953	193
I love finding new designer brands for jeans. I usuall...	7681	314
Didn't think I'd be too interested in jewelry but t...	3870	43
[redacted]	6286	151

this zip code is
80% hot climate

this zip code is
60% outdoors wear

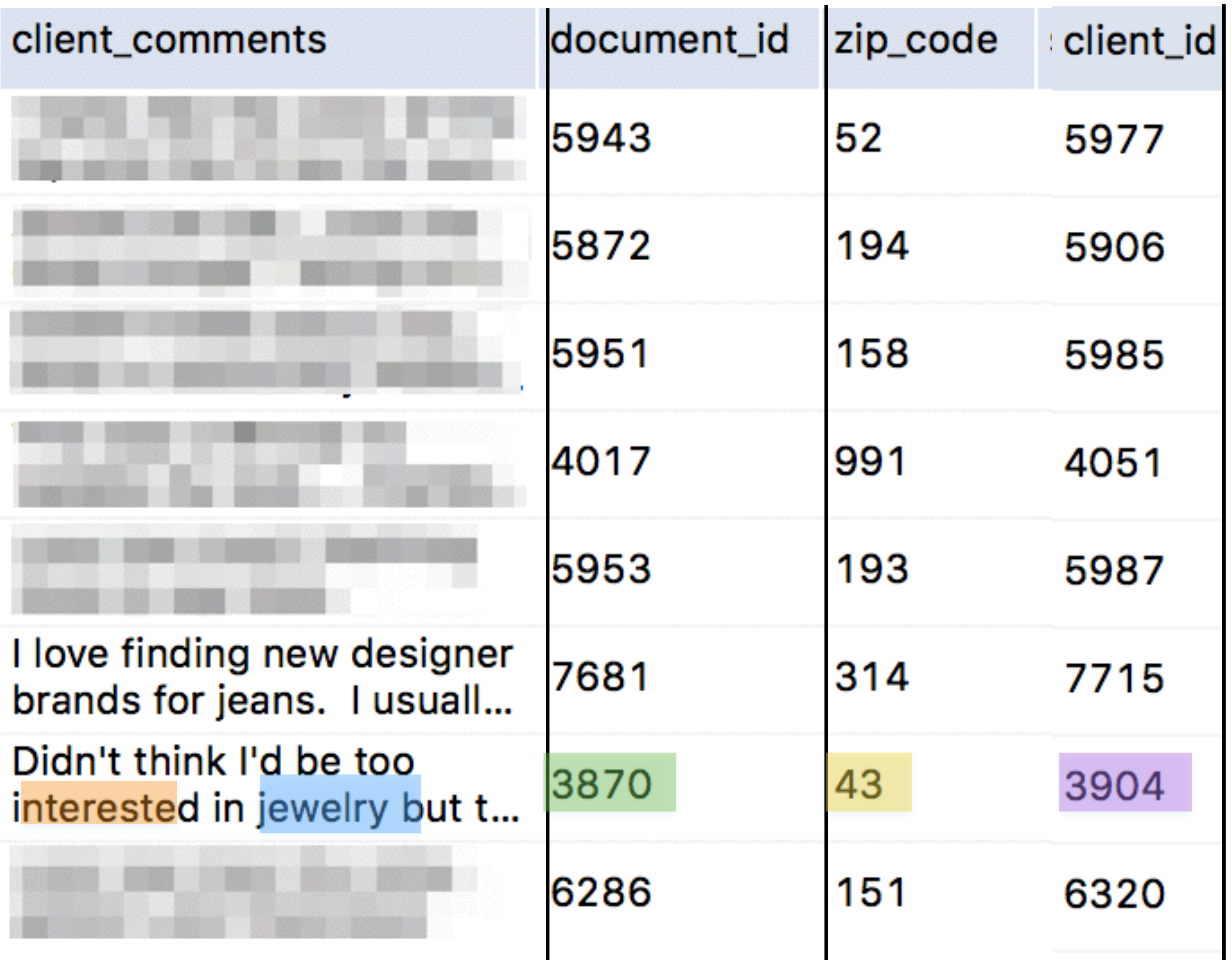
word2vec

LDA

lda2vec

$$P(v_{OUT} | v_{IN} + v_{DOC} + v_{ZIP})$$

The goal:
Use all of this context to learn interpretable topics.



client_comments	document_id	zip_code	client_id
[redacted]	5943	52	5977
[redacted]	5872	194	5906
[redacted]	5951	158	5985
[redacted]	4017	991	4051
[redacted]	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
[redacted]	6286	151	6320

this client is
80% sporty

this client is
60% casual wear

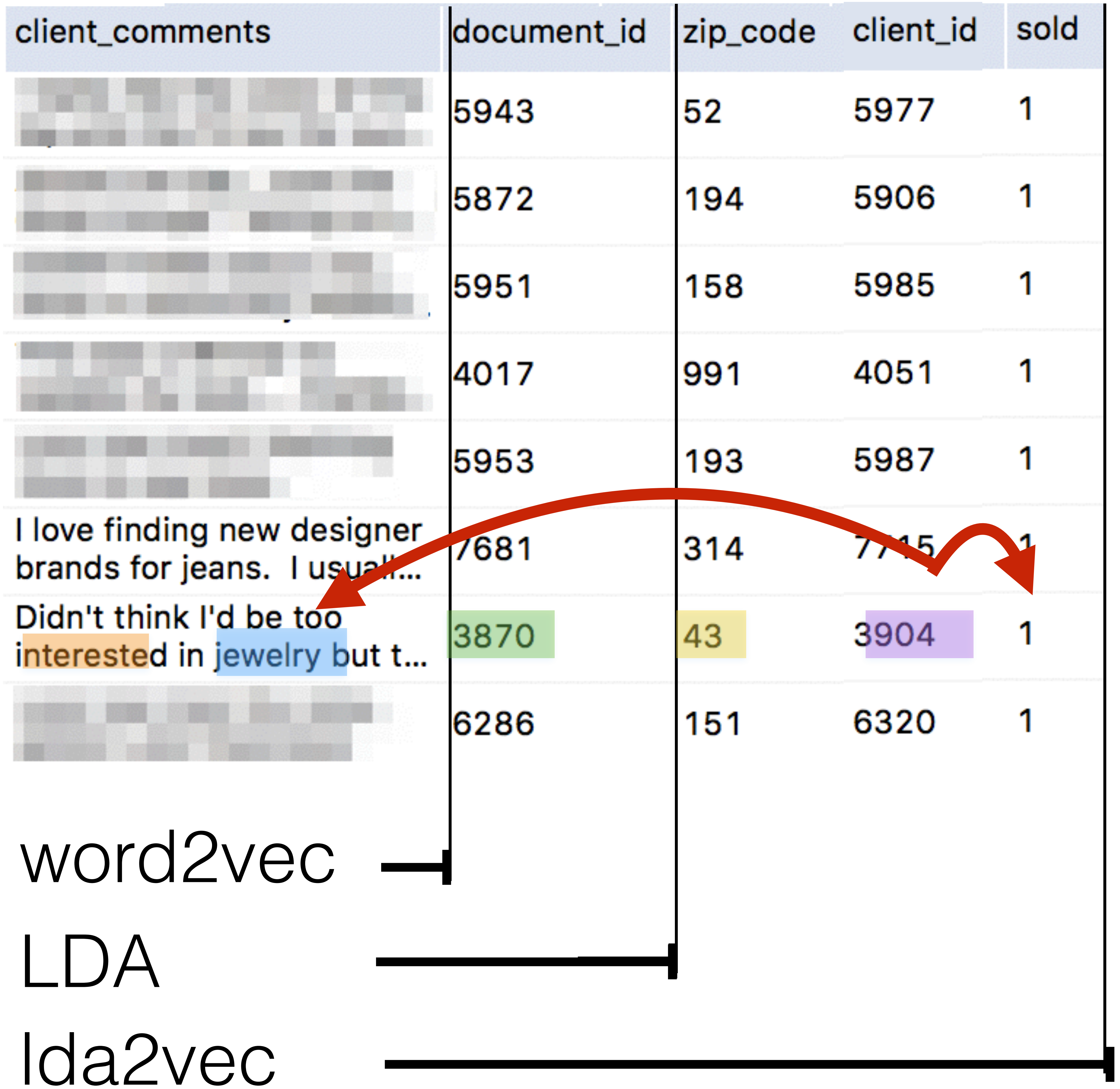
word2vec

LDA

lda2vec

$$P(v_{OUT} \mid v_{IN} + v_{DOC} + v_{ZIP} + \boxed{v_{CLIENTS}})$$

The goal:
Use all of this context to learn interpretable topics.



client_comments	document_id	zip_code	client_id	sold
	5943	52	5977	1
	5872	194	5906	1
	5951	158	5985	1
	4017	991	4051	1
	5953	193	5987	1
I love finding new designer brands for jeans. I usuall...	7681	314	7715	1
Didn't think I'd be too interested in jewelry but t...	3870	43	3904	1
	6286	151	6320	1

word2vec

LDA

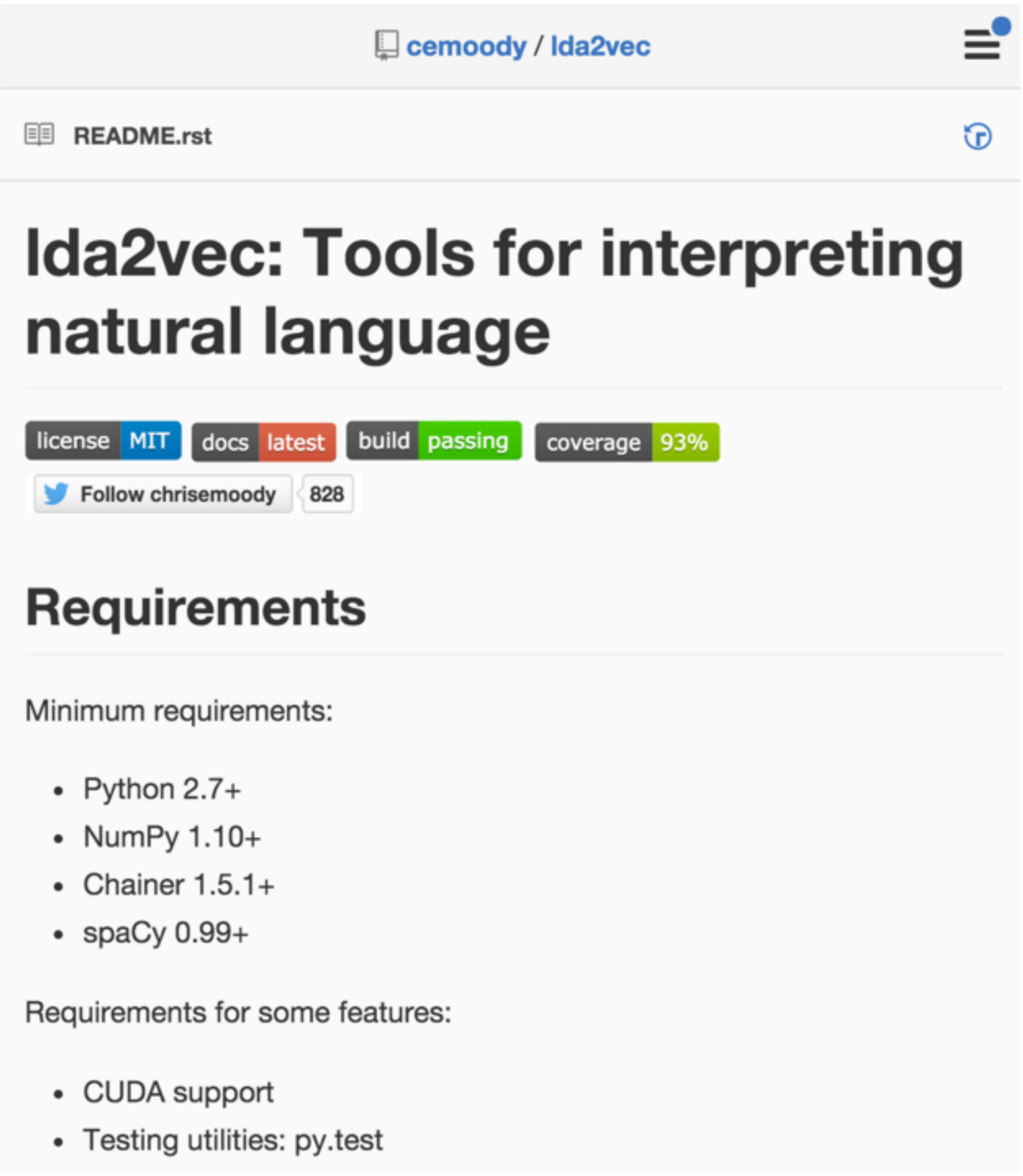
lda2vec

Can also make the topics *supervised* so that they predict an outcome.

$$P(v_{OUT} \mid v_{IN} + v_{DOC} + v_{ZIP} + v_{CLIENTS})$$

$$P(sold \mid v_{CLIENTS})$$

@chrisemoody



cemoody / lda2vec

README.rst

lda2vec: Tools for interpreting natural language

license MIT | docs latest | build passing | coverage 93%

Follow chrisemoody 828

Requirements

Minimum requirements:

- Python 2.7+
- NumPy 1.10+
- Chainer 1.5.1+
- spaCy 0.99+

Requirements for some features:

- CUDA support
- Testing utilities: py.test

```
In [31]: pyLDAvis.display(prepared_data)
Out[31]:
```

Selected Topic: 15

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

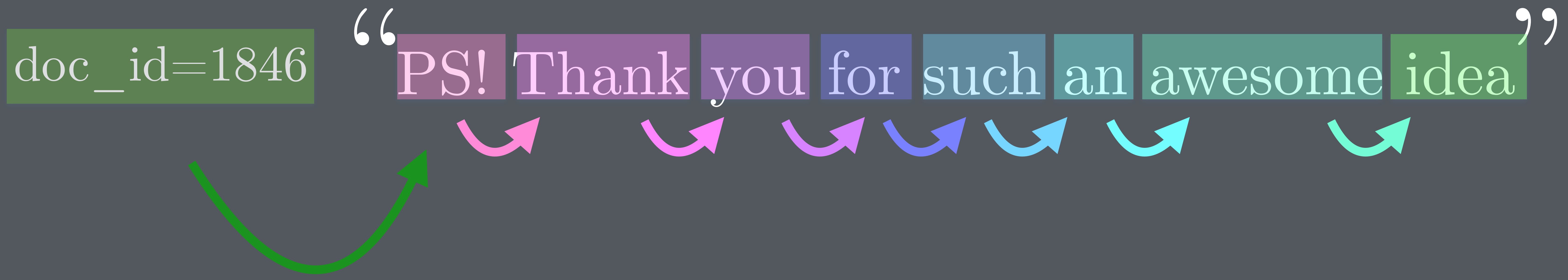
uses pyldavis

API Ref docs (no narrative docs)
GPU
Decent test coverage

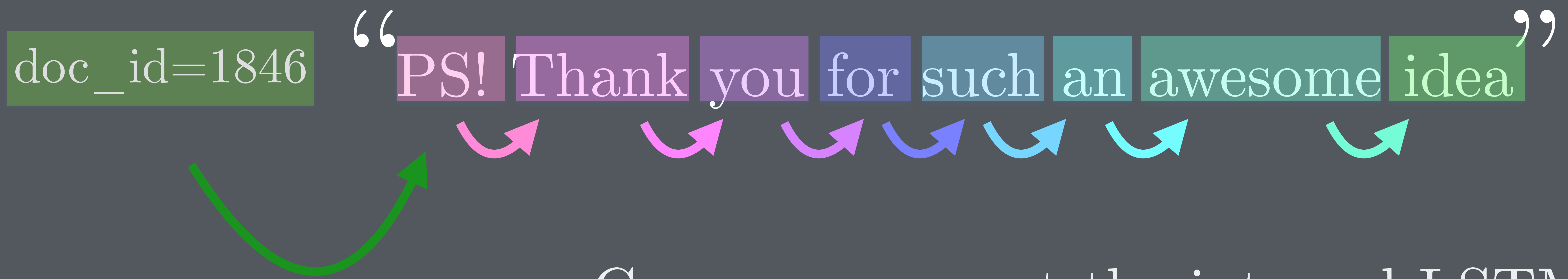
github.com/cemoody/lda2vec

Can we model topics to sentences?

lda2lstm



doc_id=1846 “PS! Thank you for such an awesome idea”

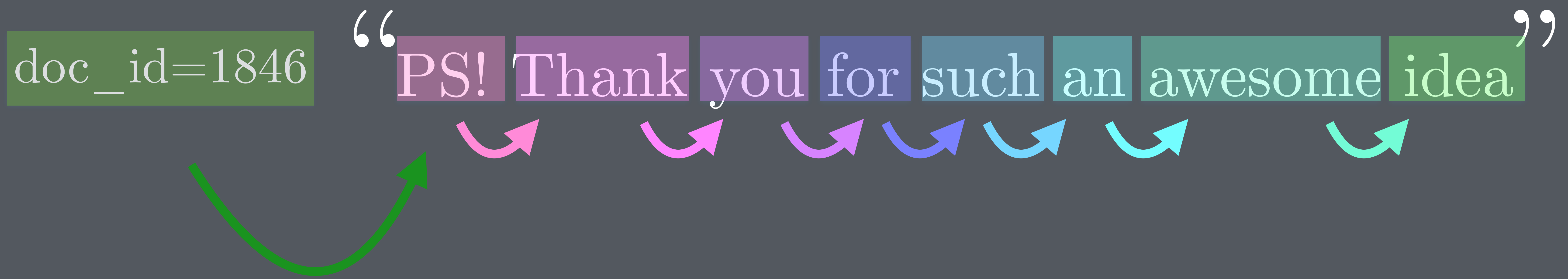


doc_id=1846 “PS! Thank you for such an awesome idea”

Can we represent the internal LSTM
states as a dirichlet mixture?

Can we model topics to sentences?

lda2lstm



doc_id=1846 "PS! Thank you for such an awesome idea"

Can we model topics to images?

lda2ae

TJ Torres

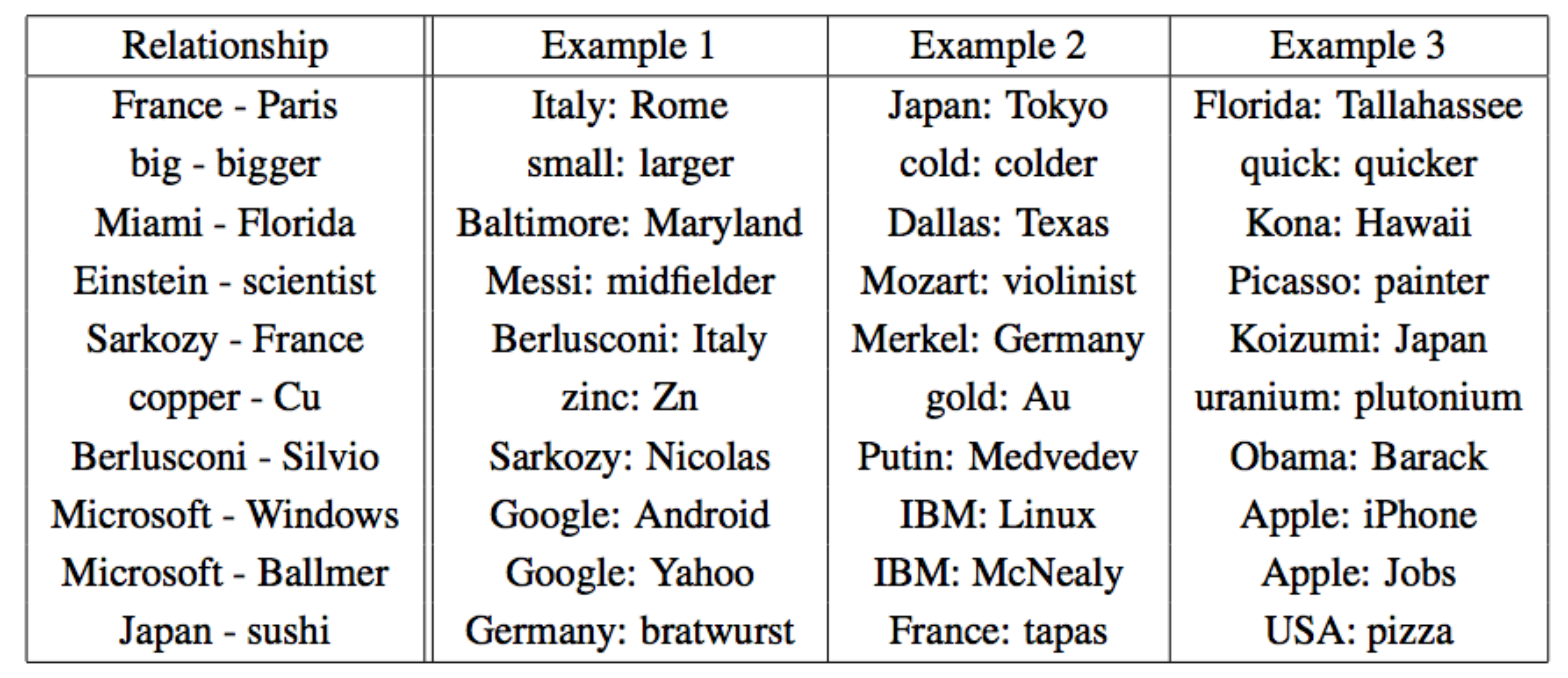
?

@chrisemoody

Multithreaded
Stitch Fix

Bonus slides

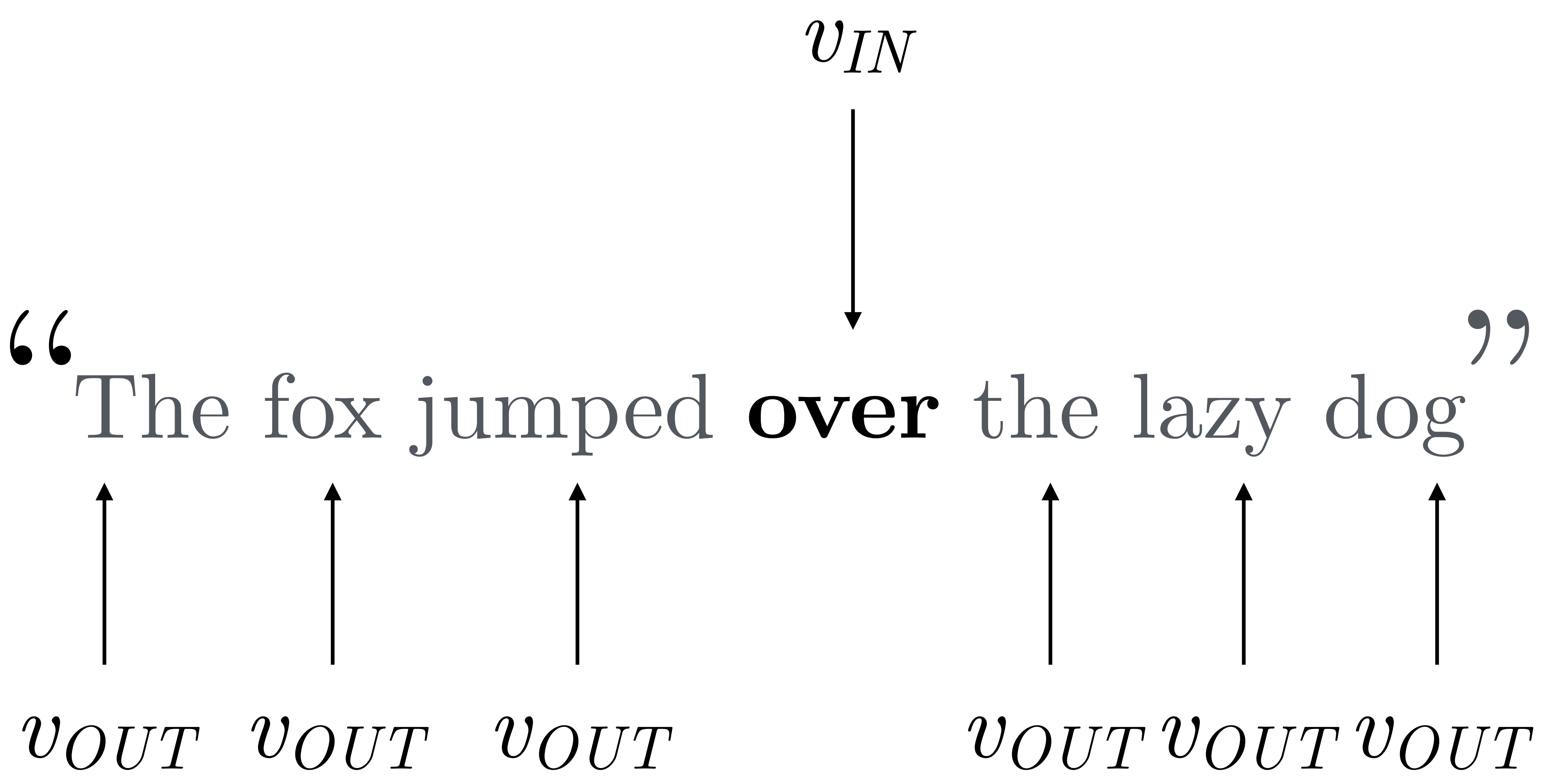
- Paragraph Vectors
  - (Just extend the context window)
- Content dependency
  - (Change the window grammatically)
- Social word2vec (deepwalk)
  - (Sentence is a walk on the graph)
- Spotify
  - (Sentence is a playlist of song_ids)
- Stitch Fix
  - (Sentence is a shipment of five items)



Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

SkipGram

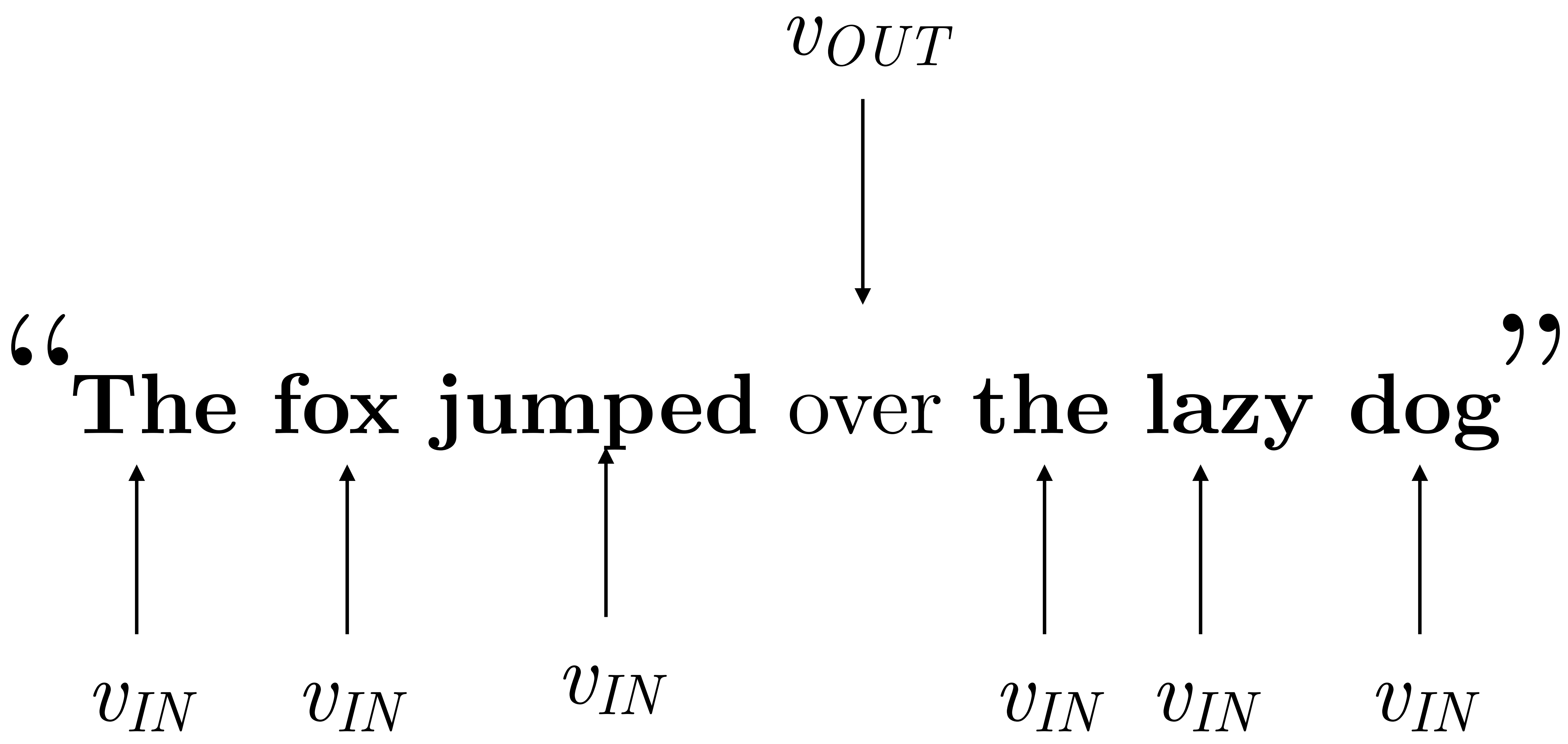
Guess the context
given the word



Better at syntax.
(this is the one we went over)

CBOW

Guess the word
given the context



~20x faster.
(this is the alternative.)


holidays
wedding
month
destinations
vacation
season
bachelorette
shower
party
rehearsal
luau
birthday
wedding
events
weddings
dinners
cruise
vacationed
honeymooning
vegas
holidaying
destination
vacationers
honeymoons
getaways
vacations
trips
getaway
resort
resorts
waikiki
february
siesta
mustique
oceanfront
honeymoon
cancun
cozumel
disneyworld
nov.
oct.
dec.
sept.
may.
feb.
june.
april.
march.
th.
st.
march..
brunch
lunch
holidays
christmas
thanksgiving
valentines
easter
dinner
holiday
concert
reunion
outing
relaxing
outings
attending
evenings
night
evening
friends
headed
attend
engagement
weekends
fancy
town
weekend
days
day
kids
summertime
planned
traveling
visiting
visit
springtime
journey
travels
trip
beach
home
heading
nyc
later
staying
arrived
travel
island
hawaii
mexico
business
summery
splurge
planning
moved
sept
arrives
late
arrive
march
arriving
florida
california
florida.
approaching
november
october
august
married
rome.
thailand.
disney.
disneyland.
stuff
options
perhaps
break
starting
working
december
family
europe.
hawaii.
italy.
vegas.
jamaica.
trend
clothes
especially
april
september
february
july
january
resort.
beach.
spain.
greece.
bahamas.
maui.
aruba.
wonderful
surprise
now
after
before
babymoon
mexico.
rica.
caribbean.
beginning
june
end
honeymoon.
australia.
i
in
week
coming
birthday.
cruise.
vacay.
vaca.
going
to
month
here
months
upcoming
trip.
vacation..
destination.
cancun.
belize.
vacations.
focus
focused
early
weeks
trip..
summer.
season
live
getaway.
spent
spend
summer
winter
fall
spring
weather
warm
hot
warmer
congr

LDA
Results

Great Stylist

Perfect

I loved every choice in this fix!! Great job!

History

LDA
Results

Body Fit

My measurements are 36-28-32. If that helps.
I like wearing some clothing that is fitted.
Very hard for me to find pants that fit right.

History

LDA
Results

Sizing

Excited for next

Really enjoyed the experience and the pieces, sizing for tops was too big. Looking forward to my next box!

History

LDA Results

Almost Bought

Perfect

It was a great fix. Loved the two items I kept and the three I sent back were close!

History

What I didn’t mention

A lot of text (only if you have a specialized vocabulary)

Cleaning the text

Memory & performance

Traditional databases aren’t well-suited

False positives

and now for something **completely crazy**

All of the following ideas will change what ‘words’ and ‘context’ represent.

What about summarizing documents?

On the day he took office, President Obama reached out to America's enemies, offering in his first inaugural address to **extend** a hand if you are willing to unclench your fist. More than six years later, he has arrived at a moment of truth in testing that

On the day he took office, President Obama reached out to America's enemies, offering in his first inaugural address to **extend** a hand if you are willing to unclench your fist. More than six years later, he has arrived at a moment of truth in testing that

The framework nuclear agreement he reached with Iran on Thursday did not provide the definitive answer to whether Mr. Obama's audacious gamble will pay off. The fist Iran has shaken at the so-called Great Satan since 1979 has not completely relaxed.

Normal skipgram extends C words before, and C words after.

paragraph vector

IN

doc_1347

OUT OUT

On the day he took office, President Obama reached out to America's enemies, offering in his first inaugural address to extend a hand if you are willing to unclench your fist. More than six years later, he has arrived at a moment of truth in testing that

The framework nuclear agreement he reached with Iran on Thursday did not provide the definitive answer to whether Mr. Obama's audacious gamble will pay off. The fist Iran has shaken at the so-called Great Satan since 1979 has not completely relaxed.

OUT OUT

A document vector simply extends the context to the whole document.

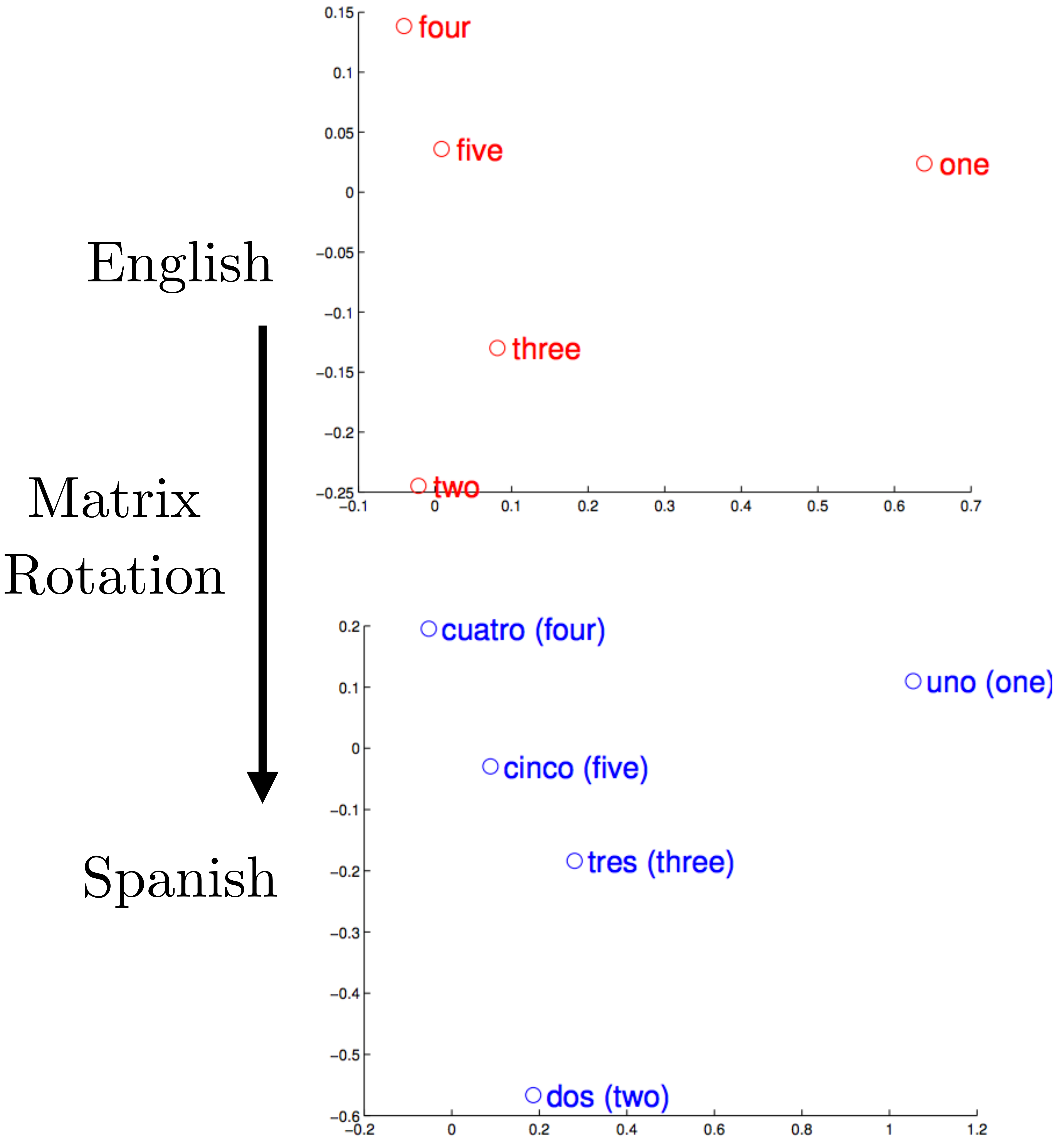
```
from gensim.models import Doc2Vec
fn = “item_document_vectors”
model = Doc2Vec.load(fn)
model.most_similar('pregnant')
matches = list(filter(lambda x: 'SENT_' in x[0], matches))

# ['...I am currently 23 weeks pregnant...',
#  '...I'm now 10 weeks pregnant...',
#  '...not showing too much yet...',
#  '...15 weeks now. Baby bump...',
#  '...6 weeks post partum!...',
#  '...12 weeks postpartum and am nursing...',
#  '...I have my baby shower that...',
#  '...am still breastfeeding...',
#  '...I would love an outfit for a baby shower...']
```

sentence search

translation

(using just a rotation matrix)



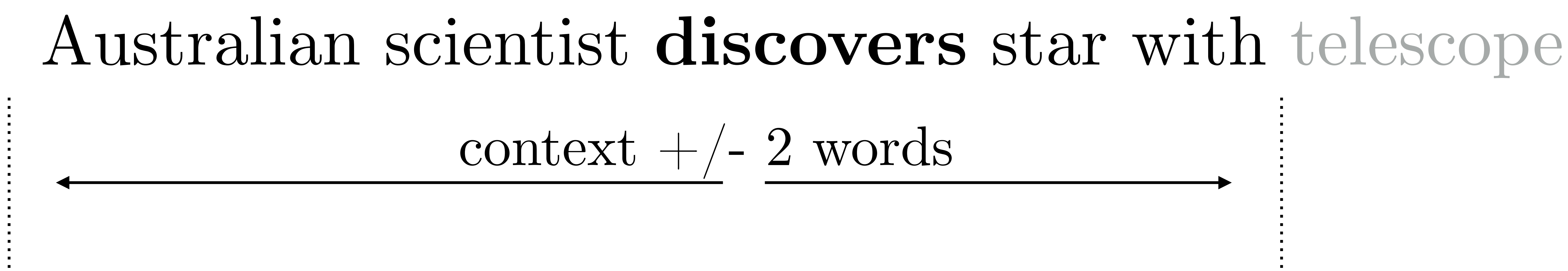
English

Matrix Rotation

Spanish

Mikolov 2013

context dependent



Australian scientist **discovers** star with telescope

context +/- 2 words

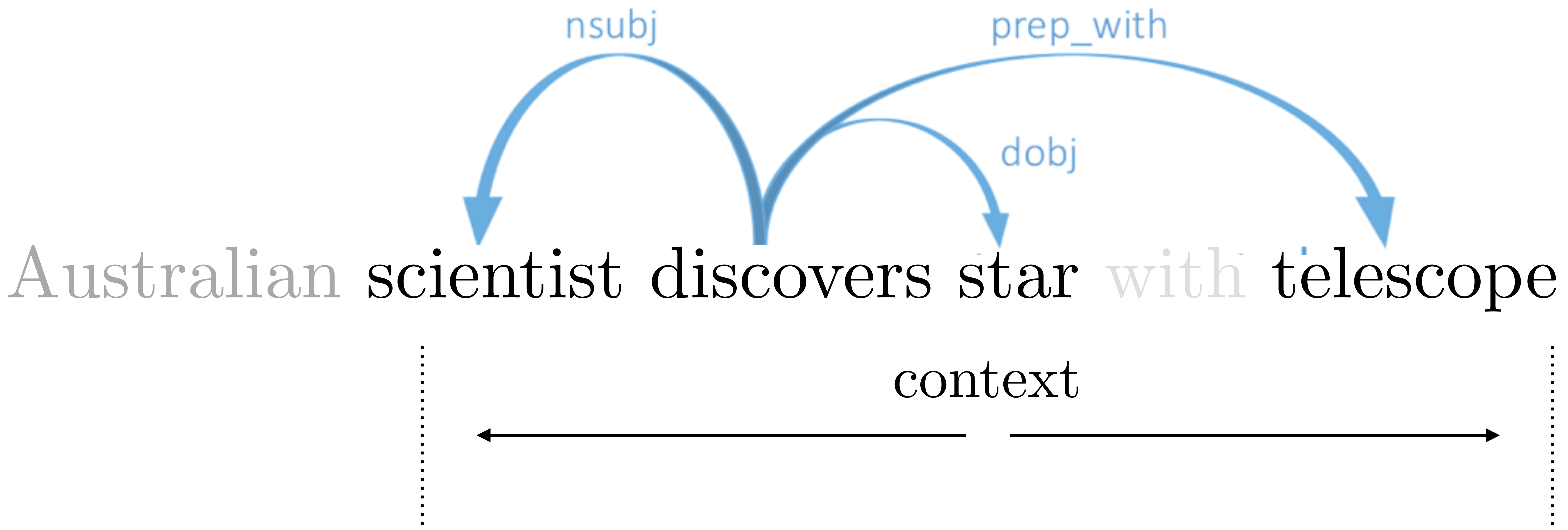
Levy & Goldberg 2014

context dependent

Australian scientist discovers star with telescope

Levy & Goldberg 2014

context dependent

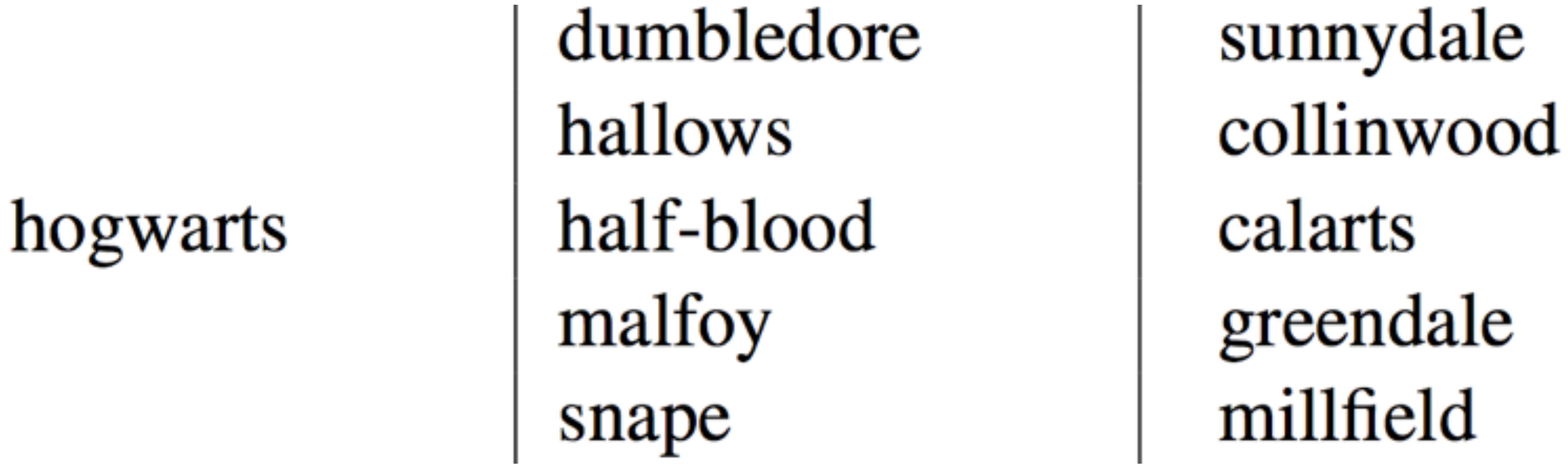


Australian scientist discovers star with telescope

context

Levy & Goldberg 2014

context dependent



	BoW	DEPS
hogwarts	dumbledore hallows half-blood malfoy snape	sunnydale collinwood calarts greendale millfield

topically-similar vs ‘functionally’ similar

Levy & Goldberg 2014

context dependent

Also show that SGNS is simply factorizing:

$$w * c = PMI(\mathrm{w},\ \mathrm{c})\ \text{-}\ log\ \mathrm{k}$$

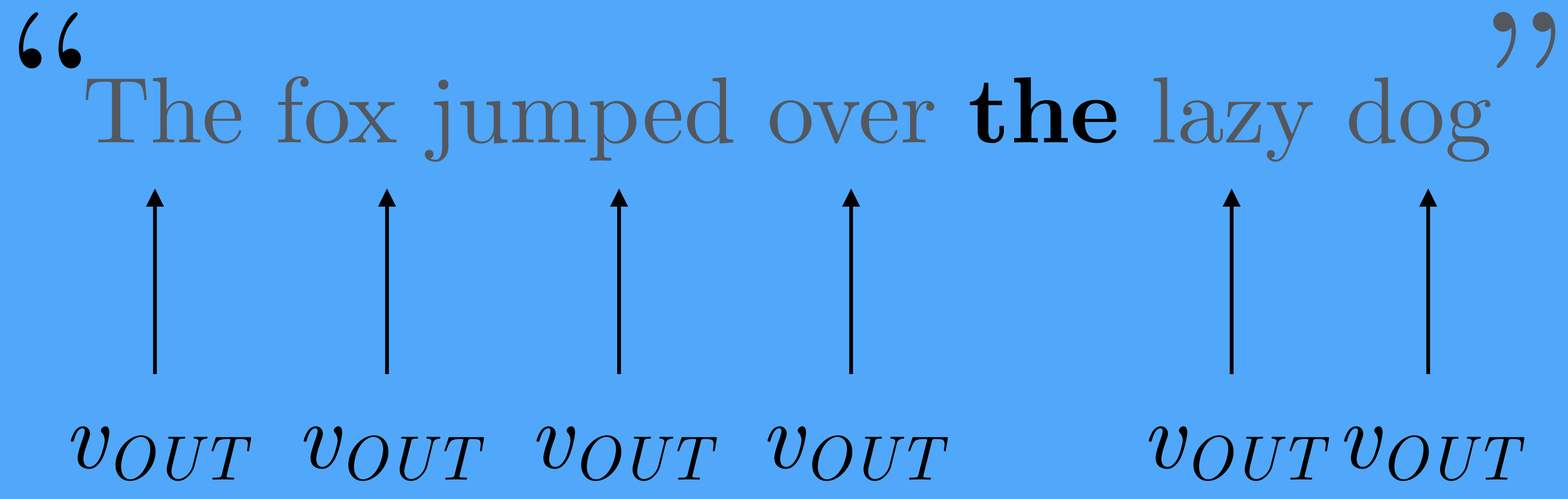
This is **completely** amazing!

Intuition: positive associations (canada, snow) stronger in humans than negative associations (what is the opposite of Canada?)

Levy & Goldberg 2014

word2vec

learn word vectors from sentences



“The fox jumped over **the** lazy dog”

v_{OUT} v_{OUT} v_{OUT} v_{OUT} v_{OUT} v_{OUT}

deepwalk

‘words’ are graph vertices
‘sentences’ are random walks on the graph

$$v_{46} \rightarrow v_{45} \rightarrow v_{71} \rightarrow v_{24} \rightarrow v_5$$

Perozzi et al 2014

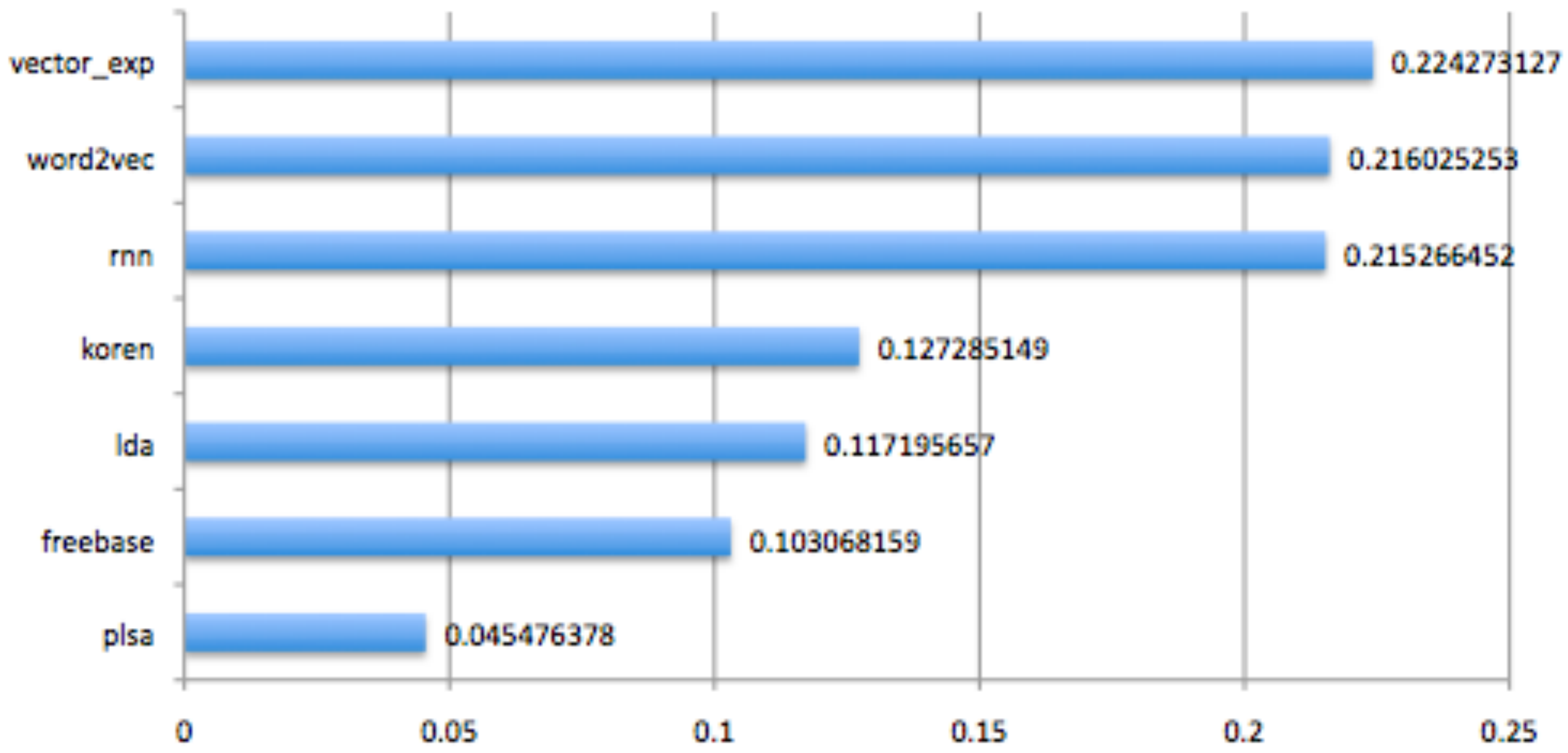
Playlists at Spotify

‘words’ are songs
‘sentences’ are playlists

sequence learning

Playlists at Spotify

Great performance on ‘related artists’



Erik Bernhardsson

Fixes at Stitch Fix

Let’s try:
‘words’ are styles
‘sentences’ are fixes

sequence learning

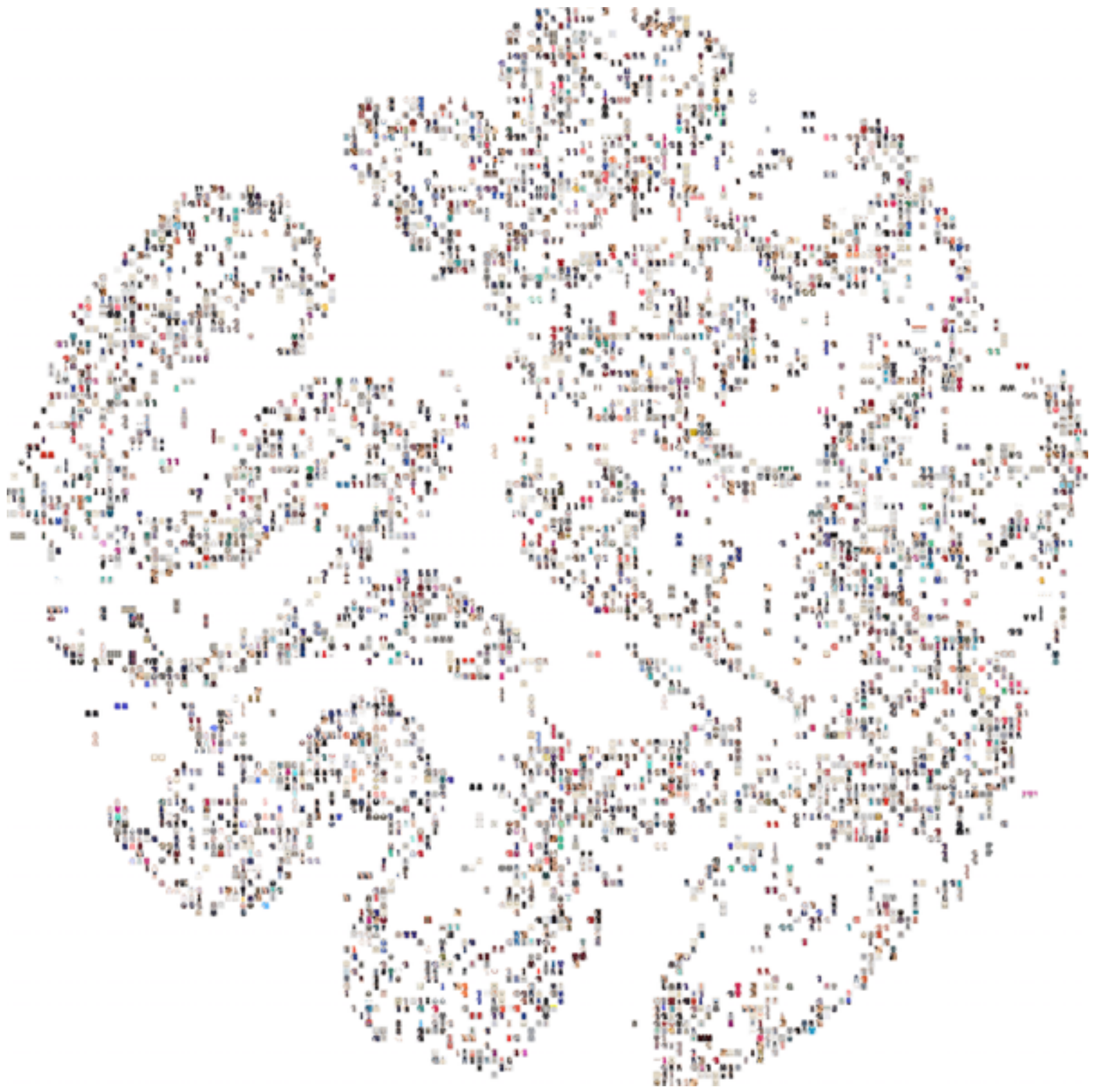
Fixes at Stitch Fix

Learn similarity between styles because they co-occur

Learn ‘coherent’ styles

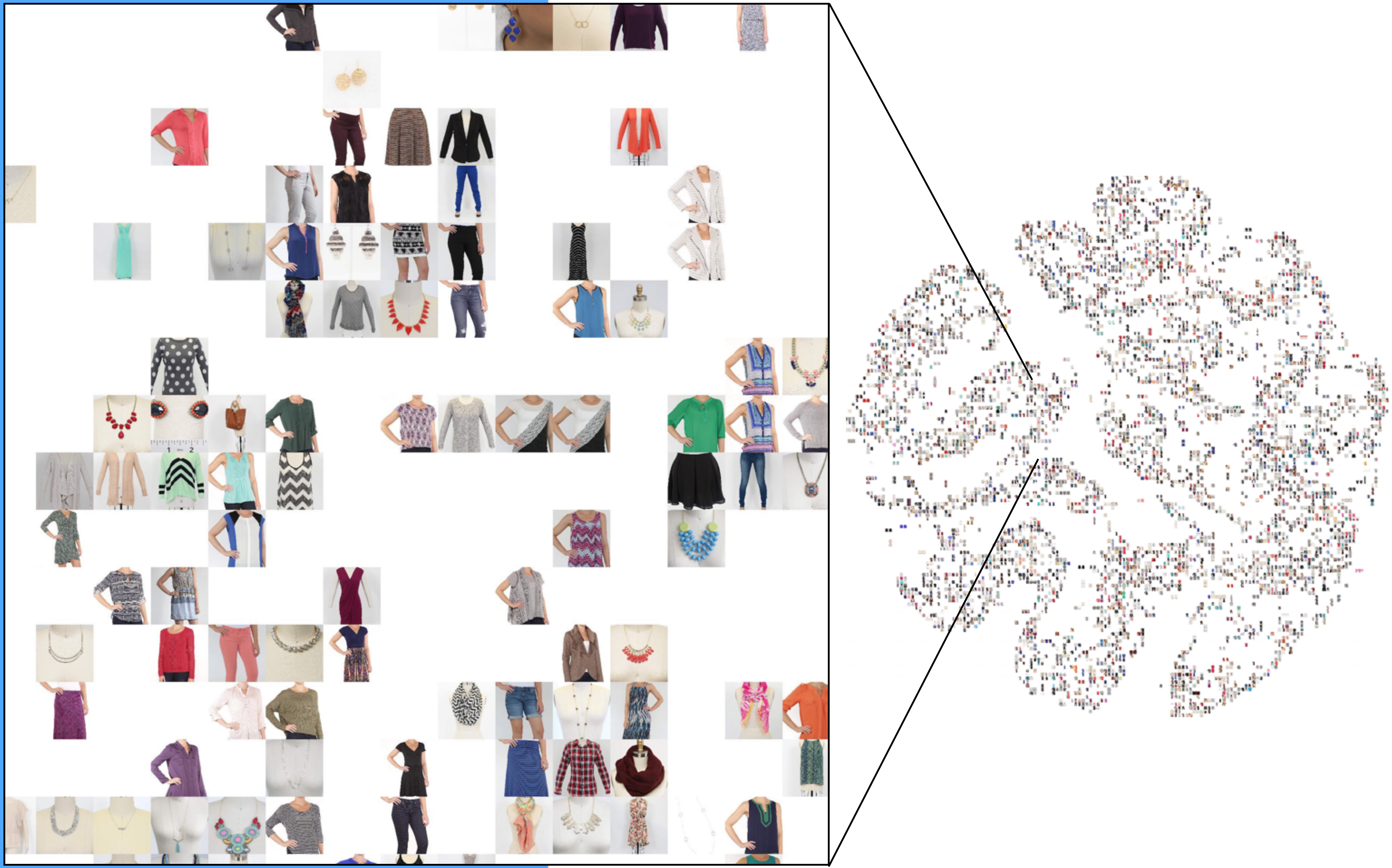
sequence learning

Fixes at Stitch Fix?

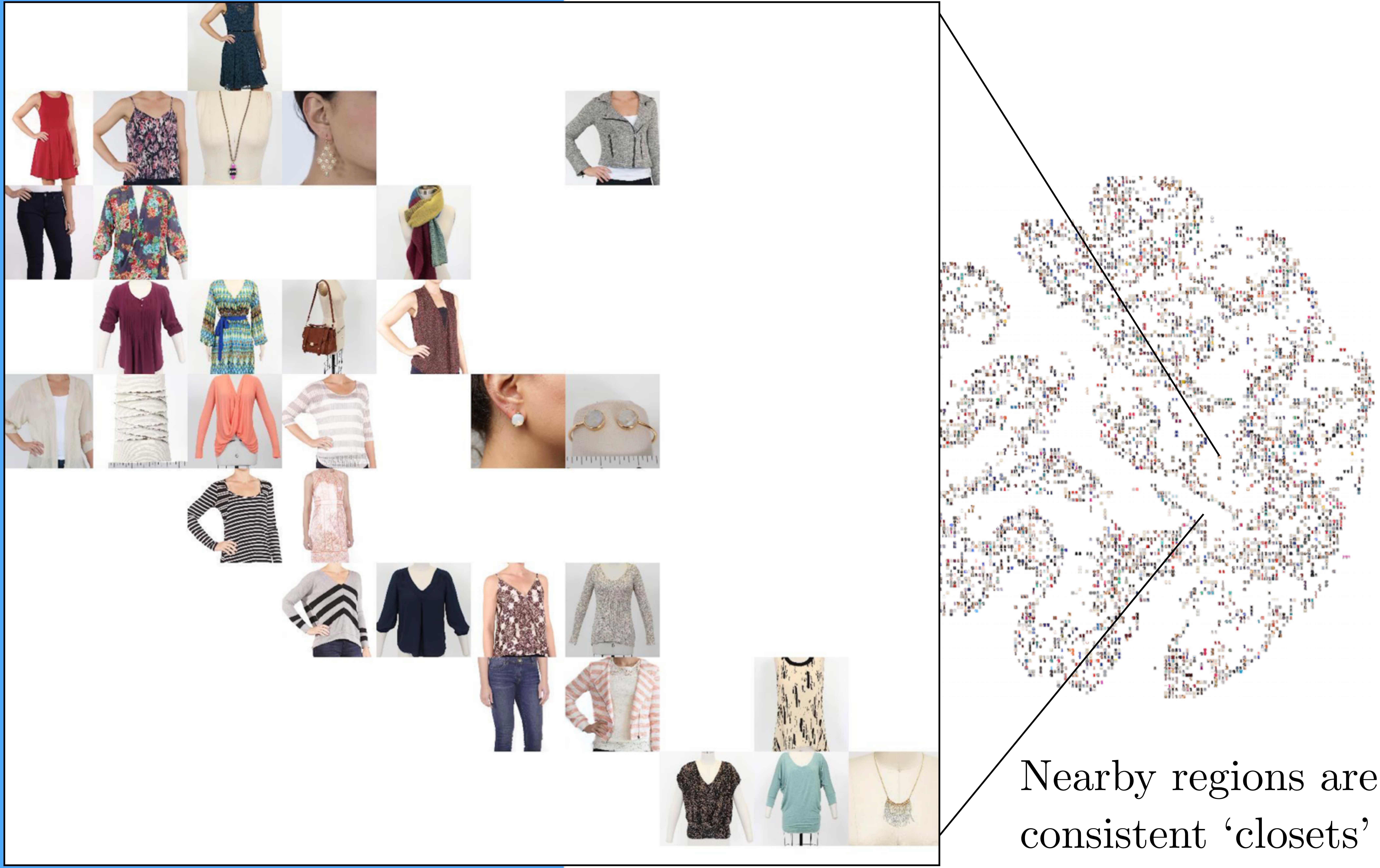


Got lots of structure!

sequence learning



sequence learning



Nearby regions are consistent ‘closets’

sequence learning

A specific lda2vec model

Our text blob is a comment that comes from a region_id and a style_id

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

$$context = \vec{c_{ij}} = \vec{region}_i + style_j$$

$$\vec{region}_i = \Sigma_{k=0}^{n\_topics} u_{ik} \cdot \vec{m}_k$$

$$\vec{style}_j = \Sigma_{l=0}^{n\_topics} u_{jl} \cdot \vec{n}_l$$

$$\vec{u} \sim dirichlet(\alpha_1)$$

$$\vec{v} \sim dirichlet(\alpha_2)$$

$$take\_rate\_in\_region \sim 5.0 * \sigma(W \cdot \vec{u})$$

The full likelihood model

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

$$context = \vec{c_{ij}} = \vec{region}_i + \vec{style}_j$$

$$\vec{region}_i = \Sigma_{k=0}^{n\_topics} u_{ik} \cdot \vec{m}_k$$

$$\vec{style}_j = \Sigma_{l=0}^{n\_topics} u_{jl} \cdot \vec{n}_l$$

$$\vec{u} \sim dirichlet(\alpha_1)$$

$$\vec{v} \sim dirichlet(\alpha_2)$$

$$take\_rate\_in\_region \sim 5.0 * \sigma(W \cdot \vec{u})$$

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

First part of the loss function is given context predict word.

Don’t predict a negative word. These are words that are in our vocabulary somewhere, but not in our example.

We get negative samples **not** uniformly, but proportional to the word frequency^¾ (yes, the ¾ power is weird and ad hoc but totally works awesomely for word2vec)

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

$$context = \vec{c_{ij}} = \vec{region_i} + style_j$$

Context is made up from more than one part -- many ‘contexts’ available.

In this case, instead of one document, we can have many regions, or styles.

In LDA, this context is a single term: the latent document vector that ‘generates’ words.

In word2vec, this context is the ‘pivot’ word. Word2vec picks a random ‘context’ word in the corpus, centers a window around it, and tries to predict other words within that context.

In both word2vec and LDA context is one term, either a document or a word. For lda2vec, we can more than one term, we can have as many contexts as we like!

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

$$context = \vec{c_{ij}} = \vec{region_i} + \vec{style_j}$$

$$\vec{region_i} = \Sigma_{k=0}^{n\_topics} u_{ik} \cdot \vec{m_k}$$

$$\vec{style_j} = \Sigma_{l=0}^{n\_topics} u_{jl} \cdot \vec{n_l}$$

Each context (e.g., region or style) is decomposed into topics vectors and weights on those common topics vectors. One context has one shared set of topic vectors (think of these as cluster centroids) and every ‘document’ in that context (think of 1 of 50 states, 1 of 20k styles) has a weight/membership onto each of those topic vectors (think topics like northeast, midwest for region or tops, bottoms, boho, romantic for style topics)

This forces the context vectors onto **a limited set of basis vectors**. Interpret this set, and you can generalize what each region vector and style vector means. For example, one topics vector might be close to the word vector for ‘hand_bag’, ‘purse’, ‘bag’ indicating that that topic is a handbags topic. And then anything with big weight in that topic might be a handbag.

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

$$context = \vec{c_{ij}} = \vec{region_i} + \vec{style_j}$$

$$\vec{region_i} = \Sigma_{k=0}^{n\_topics} u_{ik} \cdot \vec{m_k}$$

$$\vec{style_j} = \Sigma_{l=0}^{n\_topics} u_{jl} \cdot \vec{n_l}$$

$$\vec{u} \sim dirichlet(\alpha_1)$$

$$\vec{v} \sim dirichlet(\alpha_2)$$

But the weights can still end up being very dense -- which meant everyone of my documents was a mixture of almost every component. This made it difficult to interpret what the document was, because it had membership in many groups.

So next we enforce a simplex with dirichlet & enforce sparsity with the concentration on the weights. The dirichlet is also nice but not critical, we could've had a non-negative decomposition or just stuck with all reals. But since Dirichlet components sum to 100%, it is easier to explain to analysts that a document is "10% of some_topic + 90% some_other_topic" rather than saying "-2.3 * some_topic and +0.5 of some_other_topic".

[redacted]

$$L = \sigma(c * w) + \sigma(-c * w_{neg})$$

$$context = \vec{c}_{ij} = \vec{region}_i + style_j$$

$$\vec{region}_i = \Sigma_{k=0}^{n\_topics} u_{ik} \cdot \vec{m}_k$$

$$\vec{style}_j = \Sigma_{l=0}^{n\_topics} u_{jl} \cdot \vec{n}_l$$

$$\vec{u} \sim dirichlet(\alpha_1)$$

$$\vec{v} \sim dirichlet(\alpha_2)$$

$$take\_rate\_in\_region \sim 5.0 * \sigma(W \cdot \vec{u})$$

Finally, we can make this ‘supervised’ by saying that the topic weights correlate through (matrix W) with some target outcome.

Let's make v_{DOC} into a mixture…

topic 1 = "religion"

Trinitarian
baptismal
Pentecostals
bede
schismatics
excommunication

$$v_{DOC} = \mathit{10\%}\ \mathit{religion} + \mathit{89\%}\ \mathit{politics} + \ldots$$

topic 2 = "politics"

Milosevic
absentee
Indonesia
Lebanese
Isrealis
Karadzic