Word Embeddings for Fun and Profit

http://rare-technologies.com/

Lev Konstantinovskiy

Community Manager at Gensim

@teagermylk

RARE TECHNOLOGIES

Part 1. Word2vec.

Part 2. Document classification.

About

Lev Konstantinovskiy

@teagermylk

https://github.com/tmylk

Moscow ORT School

Moscow Institute of Electronics and Mathematics

Graduate school drop-out in Algebraic Geometry in USA

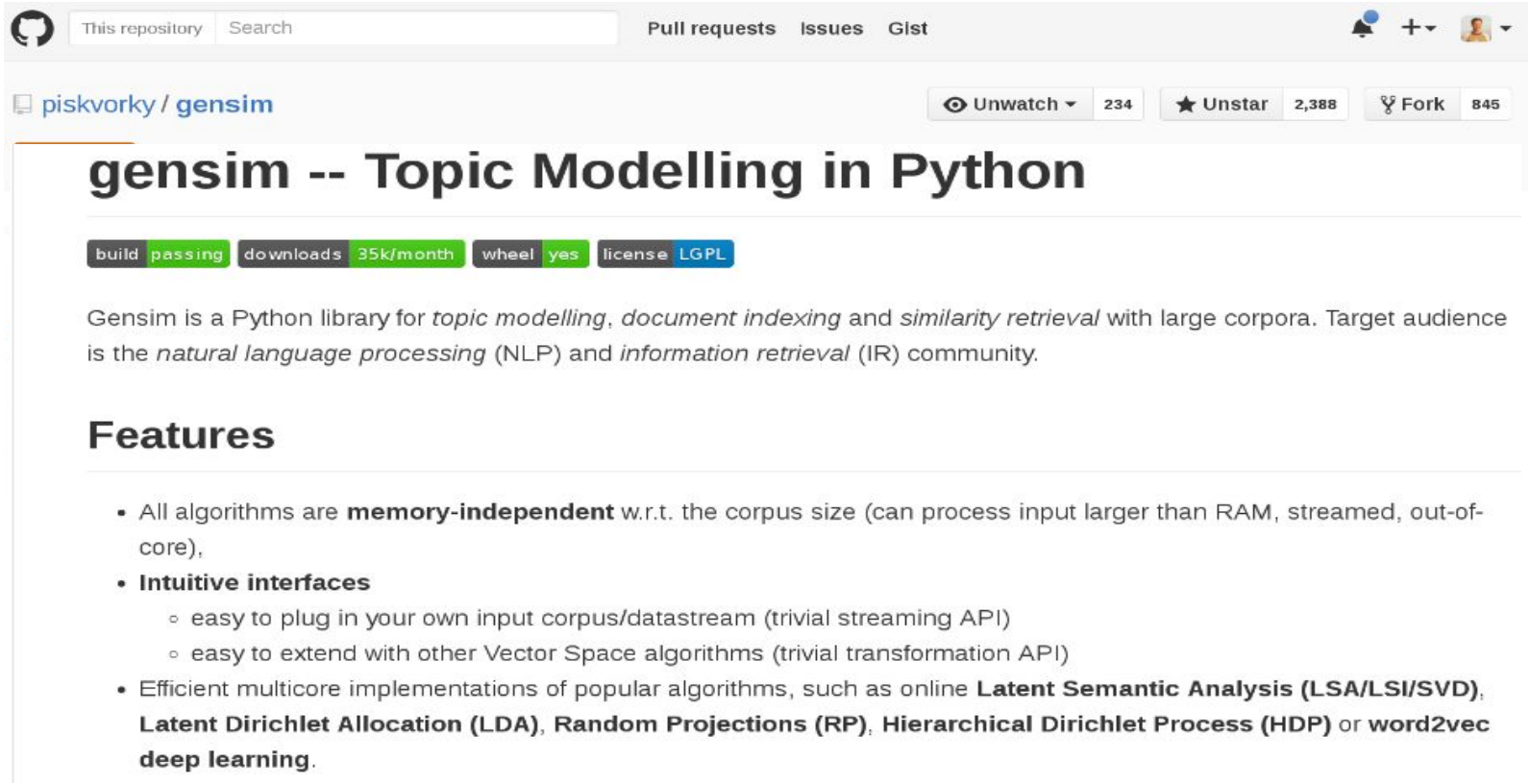
Worked in Trading IT in London

Graduate of Galvanize Data Science Bootcamp in San Francisco

Community manager and consultant at RaRe Technologies .

NLP consulting and open-source development of Natural Language Processing packages in Python.

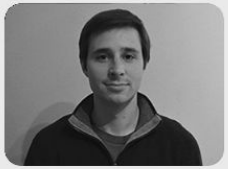
Word2vec and Topic Modelling in Python



RARE TECHNOLOGIES

INCUBATOR GRADUATES

Ólavur Mortensen


Šimon Pavlík

Bhargav Srinivasa Desikan

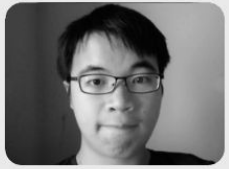
Devashish Deshpande

Hayate Iso

Jayant Jain

Rishab Goel

Mridul Seth


Haoming Jiang

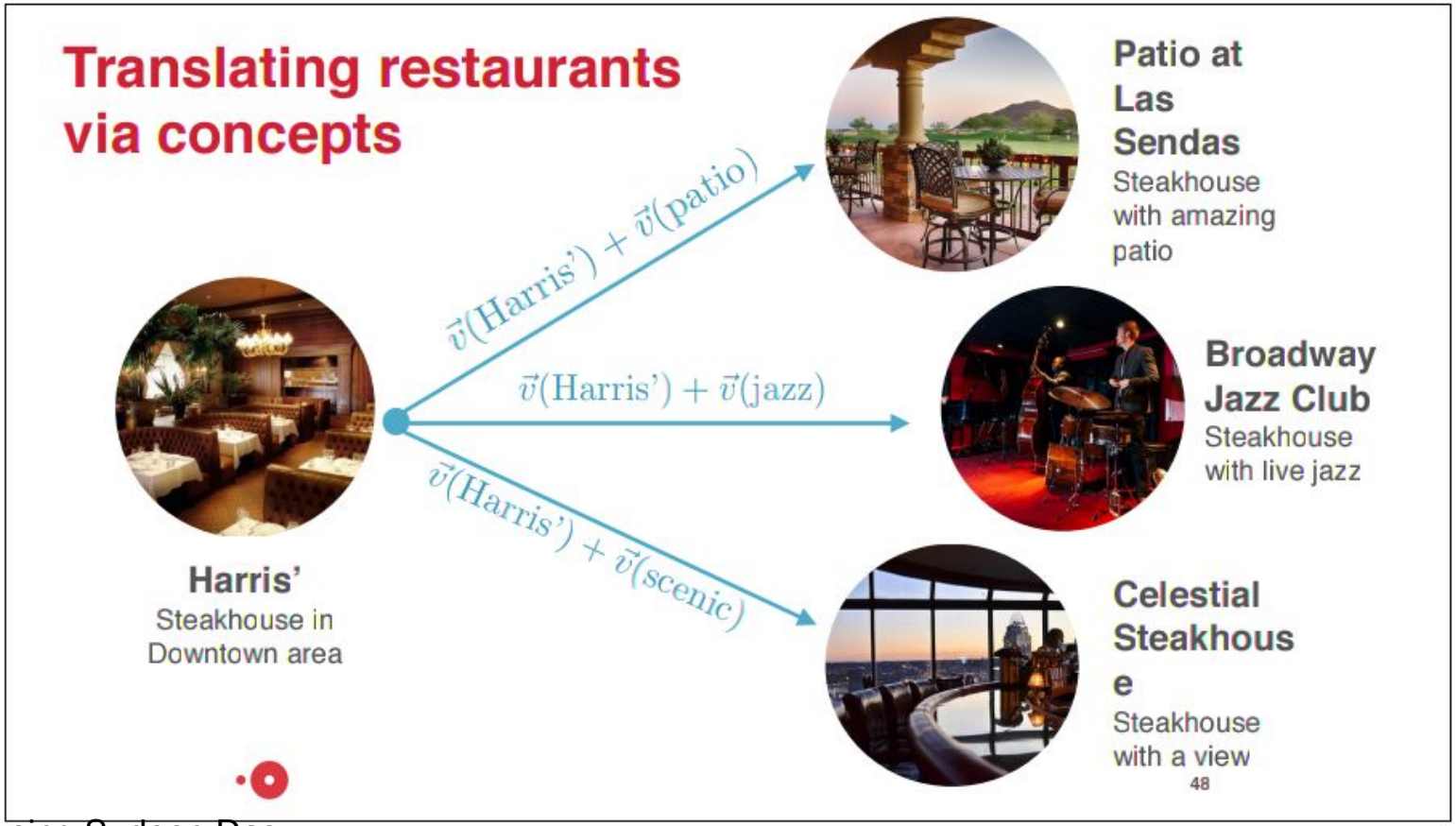
Daniel Roudnitsky

Gensim coding sprint

Date: November
Location: Вышка
Contact me on twitter @teagermylk or email

Topic:
Learn machine learning by improving our tutorials.

Why would I care? Because it is magic!



@datamusing Sudeep Das

http://www.slideshare.net/SparkSummit/using-data-science-to-transform-opentable-into-delgado-das

RARE TECHNOLOGIES

Word embeddings can be used for:

- automated text tagging ***(this talk)***
- recommendation engines
- synonyms and search query expansion
- machine translation
- plain feature engineering

Notebook at: http://small.cat/pen

The business problem to be solved in Part 2

You run a movie studio.

Every day you receive thousands of proposals for movies to make.

Need to send them to the right department for consideration!

One department per genre.

Need to ***classify plots by genre.***

What is a word embedding?

‘Word embedding’ = ‘word vectors’ = ‘distributed representations’

It is a **dense** representation of words in a **low-dimensional vector space**.

One-hot representation:

king = [1 0 0 0.. 0 0 0 0 0]

queen = [0 1 0 0 0 0 0 0 0]

book = [0 0 1 0 0 0 0 0 0]

Distributed representation:

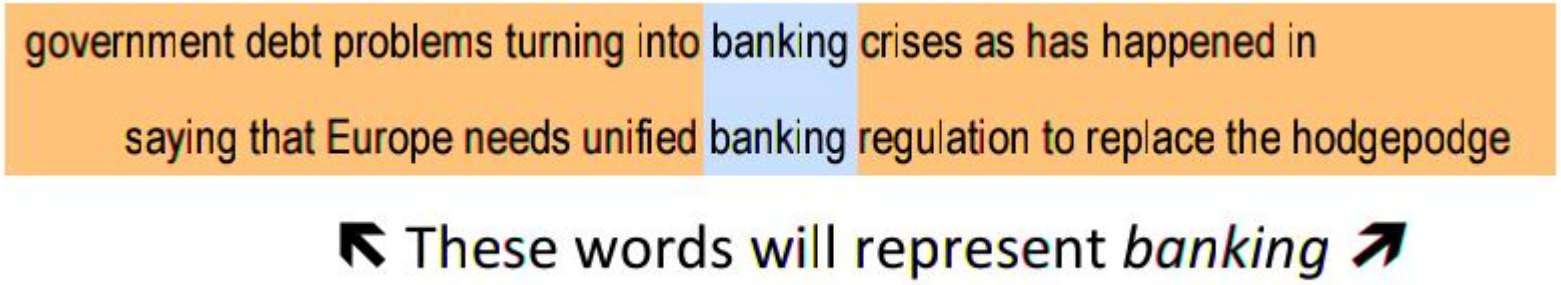
king = [0.9457, 0.5774, 0.2224]

How to come up with an embedding?

word2vec relies on the “Distributional hypothesis”:

“You shall know a word by the company it keeps”

-J. R. Firth 1957



Richard Socher’s NLP course http://cs224d.stanford.edu/lectures/CS224d-Lecture2.pdf

Usual procedure

1. Initialise random vectors
2. Pick an objective function.
3. Do gradient descent.

For the theory, take Richard Sochers's class

Details of Word2Vec

- Predict surrounding words in a window of length m of every word.
- Objective function: Maximize the log probability of any context word given the current center word:
- $$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log p(w_{t+j}|w_t)$$
- Where θ represents all variables we optimize

Richard Socher's NLP course http://cs224d.stanford.edu/lectures/CS224d-Lecture2.pdf

word2vec algorithm

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

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

$$P(\text{the}|\text{over})$$

$$P(\text{fox}|\text{over})$$

$$P(\text{jumped}|\text{over})$$

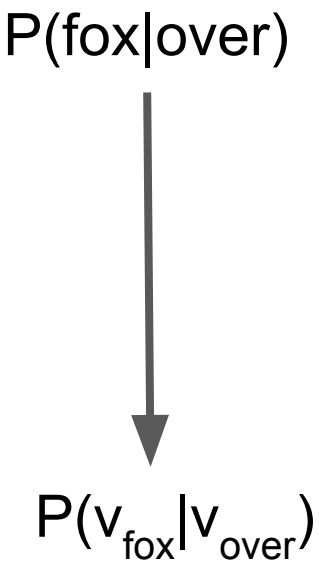
$$P(\text{the}|\text{over})$$

$$P(\text{lazy}|\text{over})$$

$$P(\text{dog}|\text{over})$$

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Probability should depend on the word vectors.



Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

A twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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

v_{IN}

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

$$P(v_{OUT}|v_{IN}) = P(v_{THE}|v_{OVER})$$

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

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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

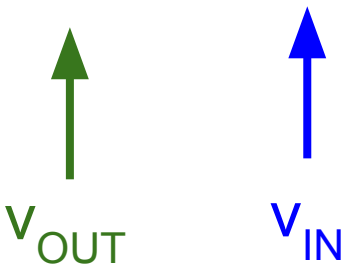
Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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



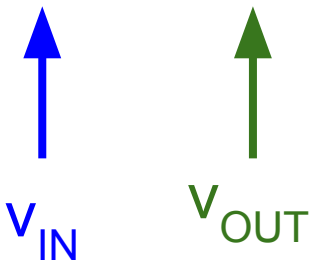
Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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



Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Twist: two vectors for every word

Should depend on whether it's the input or the output.

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

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

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

How to define $P(v_{OUT}|v_{IN})$? First, define similarity.

How similar are two vectors?

Just dot product for unit length vectors

$$v_{OUT} * v_{IN}$$

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Get a probability in [0,1] out of similarity in [-1, 1]

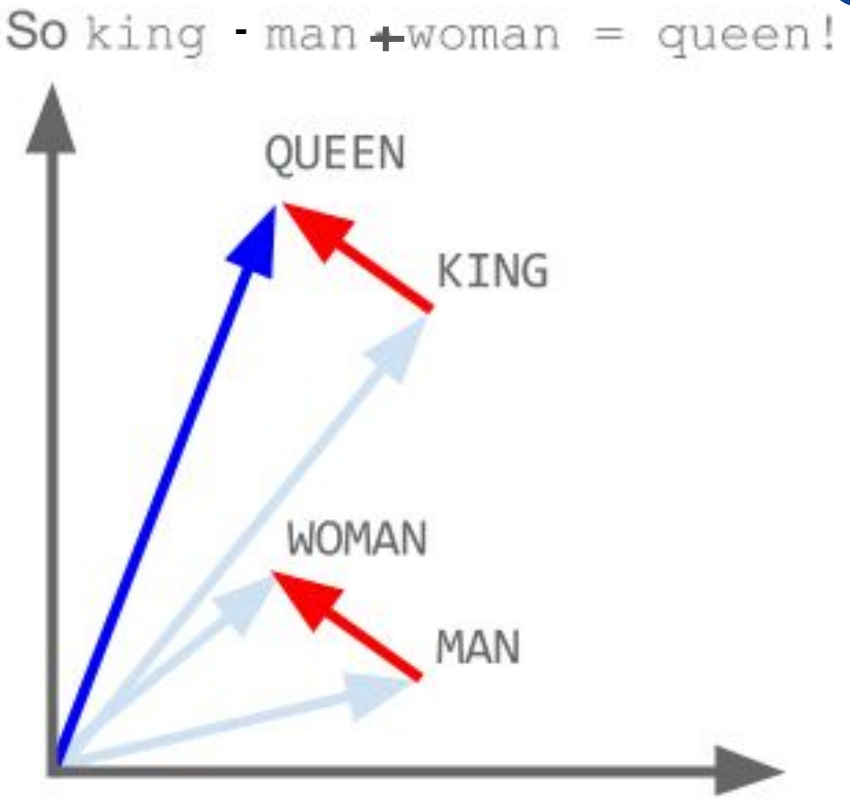
$$softmax = \frac{exp(v_{in} \bullet v_{out})}{\sum_{k \in V} exp(v_{in} \bullet v_k)} = P(v_{out}|v_{in})$$

Normalization term over all out words

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Word2vec is great!

Vector arithmetic



Slide from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Consistent directions

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

Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality 2013

Word2vec is big victory of **unsupervised** learning

Google ran word2vec on 100billion of unlabelled documents.

Then shared their trained model.

Thanks to Google for cutting our training time to zero!. :)

Questions?

Part 2. Document Classification

What is the genre of this plot?

In a future world devastated by disease, a convict is sent back in time to gather information about the man-made virus that wiped out most of the human population on the planet.

RARE TECHNOLOGIES

Of course it is SCI-FI

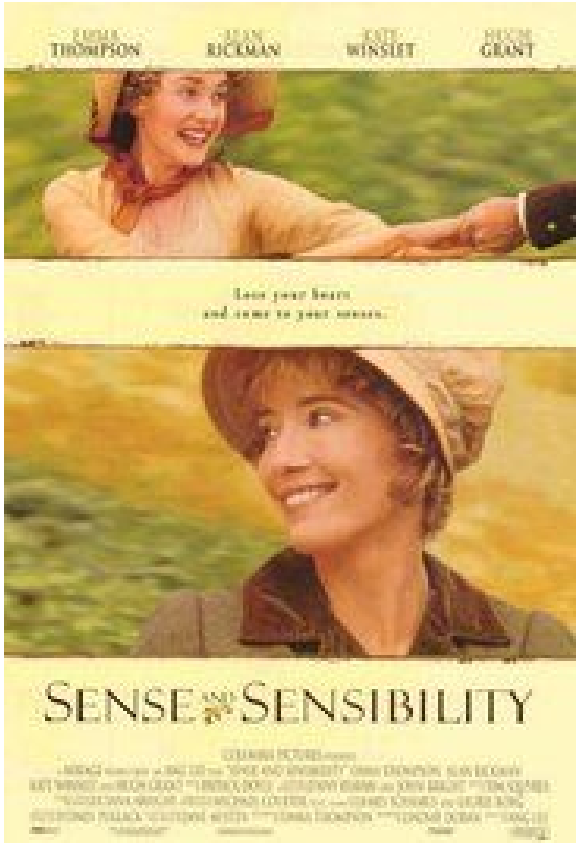
In a future world devastated by disease, a convict is sent back in time to gather information about the man-made virus that wiped out most of the human population on the planet.

What is the genre of this one?

Mrs. Dashwood and her three daughters are left in straitened circumstances. When Elinor forms an attachment for the wealthy Edward Ferrars, his family disapproves and separates them. And though Mrs. Jennings tries to match the worthy (and rich) Colonel Brandon to her, Marianne finds the dashing and fiery John Willoughby more to her taste.

ROMANCE

When Mr. Dashwood dies, he must leave the bulk of his estate to the son by his first marriage, which leaves his second wife and their three daughters (Elinor, Marianne, and Margaret) in straitened circumstances. They are taken in by a kindly cousin, but their lack of fortune affects the marriageability of both practical Elinor and romantic Marianne. When Elinor forms an attachment for the wealthy Edward Ferrars, his family disapproves and separates them. And though Mrs. Jennings tries to match the worthy (and rich) Colonel Brandon to her, Marianne finds the dashing and fiery John Willoughby more to her taste. Both relationships are sorely tried.



The text is very different so should be some signal there

RARE TECHNOLOGIES

sci-fi

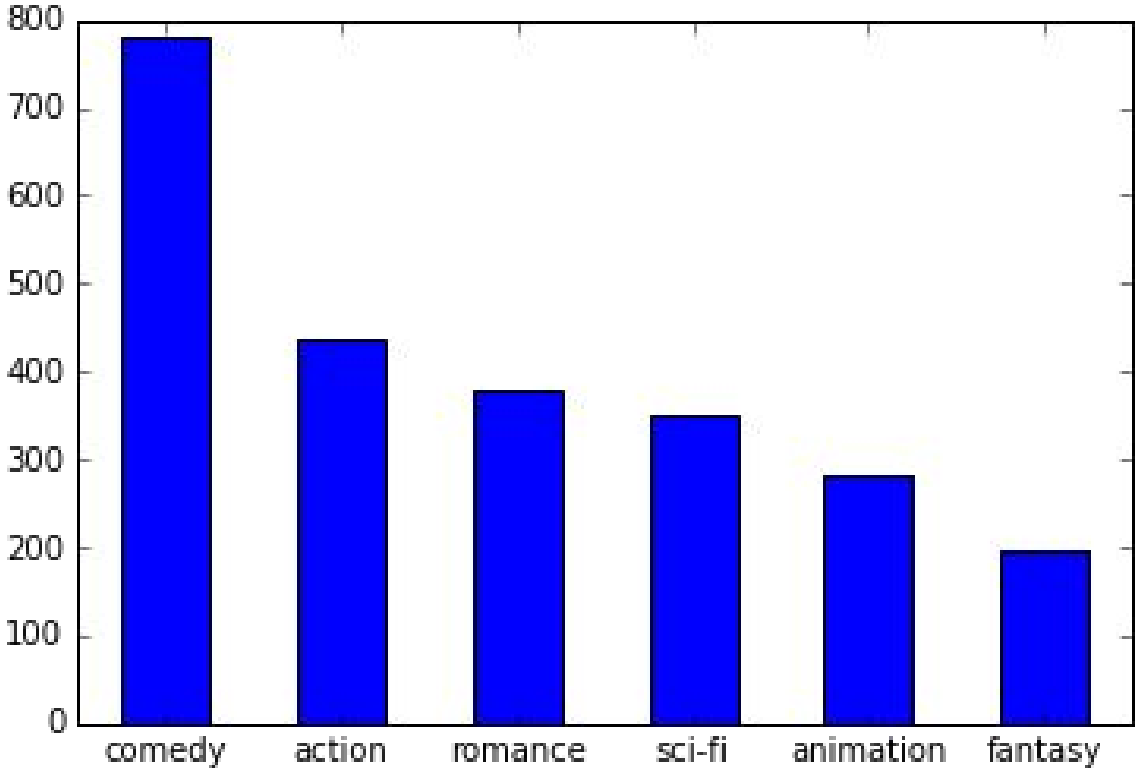
In a future world devastated by disease, a convict is sent back in time to gather information about the man-made virus that wiped out most of the human population on the planet.

romance

Mrs. Dashwood and her three daughters are left in straitened circumstances. When Elinor forms an attachment for the wealthy Edward Ferrars, his family disapproves and separates them. And though Mrs. Jennings tries to match the worthy (and rich) Colonel Brandon to her...

Corpus: 2k movie plots. 170k words.

Unbalanced classes



Disclaimer: First run, "out of the box" performance. No tuning.

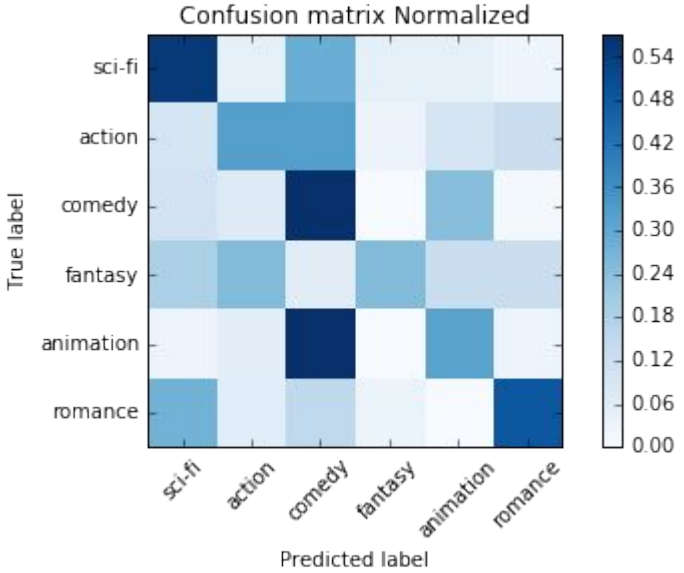
RARE TECHNOLOGIES

Simple baseline: 46%

Embedding	Classifier	Accuracy	Train time,s	Predict time, s
Baseline tf-idf bag of words	Logistic	46	2	1

TF-IDF just counting words in a document, adjusting for doc length, word frequency and word-doc frequency

```
TfidfVectorizer(
    min_df=2,
    tokenizer=nltk.word_tokenize,
    preprocessor=None,stop_words='english')
```



Counting words, char n-grams

are similar: 42, 44.

Let’s look at more advanced techniques

First, let’s load Google’s Word2vec model

Explore word2vec yourself

http://rare-technologies.com/word2vec-tutorial/#app

Word2vec to... A Document Classifier

We need some features to power our favourite classifier (logistic regression/KNN)

We have vectors for words but need vectors for documents.

*How to create a **document** classifier out of a set of **word vectors?***

*For KNN, how similar is one **sequence** of words to another **sequence** of words?*

Averaging word vectors aka ‘Naive document vector’

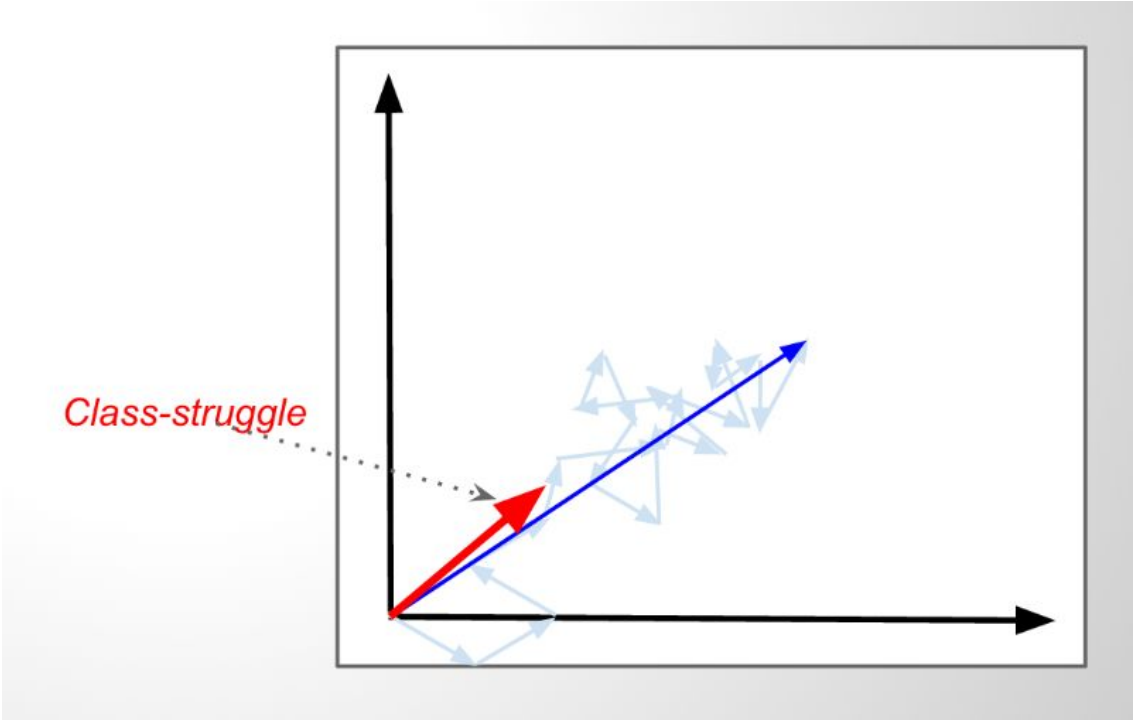
Just add word vectors together!

All words in a book

‘A tale of two cities’

Should add up to

‘class-struggle’



Mike Tamir https://www.linkedin.com/pulse/short-introduction-using-word2vec-text-classification-mike

Disclaimer: First run, "out of the box" performance. No tuning.

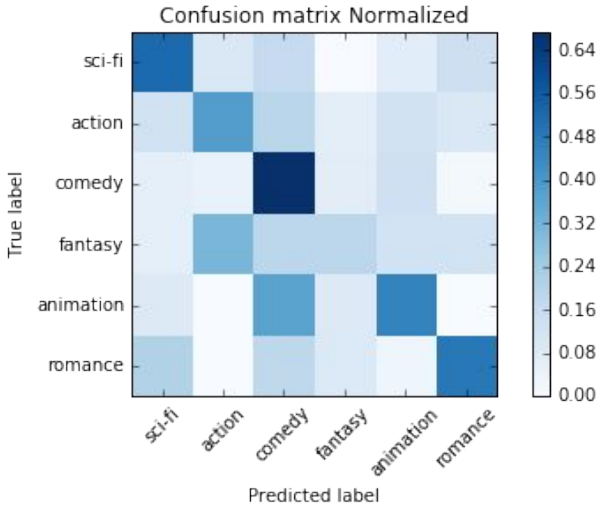
RARE TECHNOLOGIES

Averaging word vectors: 52 % accuracy

Embedding	Classifier	Accuracy	Train time,s	Predict time 250 docs, s
Averaging word vectors from pre-trained Google News	Logistic	52	2 (thanks to Google!)	1

```
# loading 1.5 GB archive into memory
wv = Word2Vec.load_word2vec_format(
                    "GoogleNews-vectors-negative300.bin.gz")
X_train_naive_dv =
                    naived2v_conversion_np(wv,flat_list_train)


logreg = linear_model.LogisticRegression(n_jobs=-1, C=1e5)
logreg = logreg.fit(X_train_naive_dv, train_data['tag'])
```



Why did summarisation not work?

Introducing **Doc2vec**

Tag is ‘a word that is in every context in the doc’

$P(v_{OUT}|v_{IN},COMEDY) = P(v_{FOX}|v_{OVER},v_{COMEDY})$

*“The fox jumped **over** the lazy dog. (COMEDY)”*

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Introducing **Doc2vec**

Tag is ‘a word that is in every context in the doc’

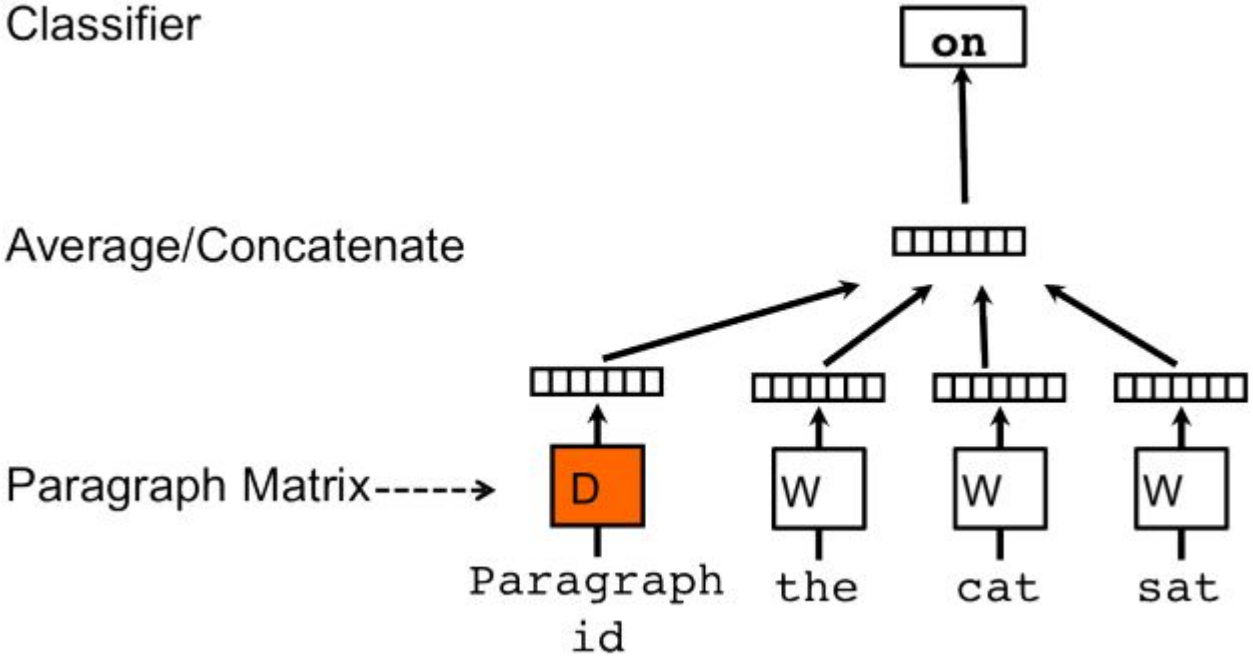
$$P(v_{OUT}|v_{IN},COMEDY) = P(v_{JUMPED}|v_{OVER},v_{COMEDY})$$

*“The fox jumped **over** the lazy dog. (COMEDY)”*

Used with permission from @chrisemoody
http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Doc2vec DM

Tag is ‘a word that is in every context in the doc’



Le, Mikolov. Distributed Representations of Sentences and Documents 2014

Let’s run some code

Disclaimer: First run, "out of the box" performance. No tuning.

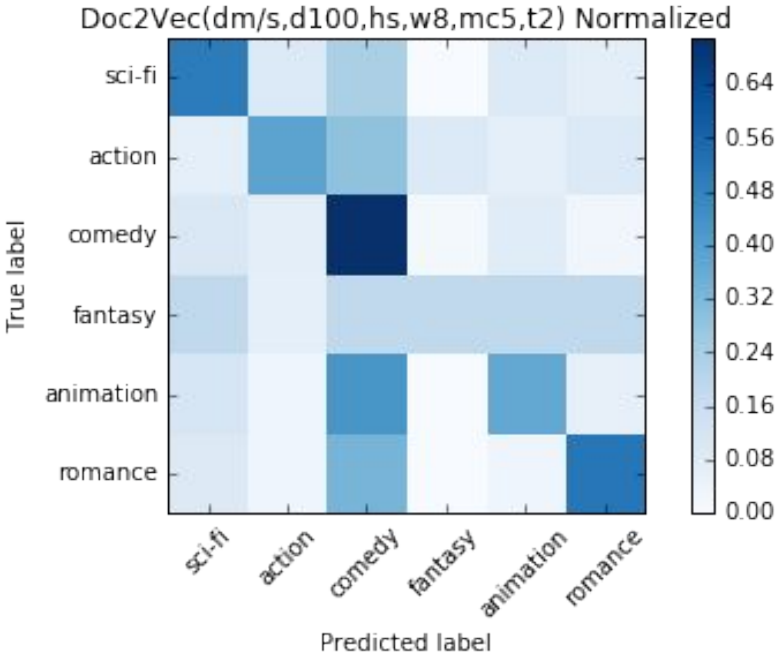
RARE TECHNOLOGIES

Doc2vec: 52 %

Embedding	Classifier	Accuracy	Train time,s	Predict time, s
Doc2vec	Logistic	52	25	1

```
# simple gensim doc2vec api
model = Doc2Vec(
      trainsent, workers=2, size=100, iter=20,
      dm=1)

train_targets, train_regressors = zip(*[
      (doc.tags[0],
      model.infer_vector(doc.words, steps=20))
      for doc in trainsent])
```



Closest words to sci-fi

```
model.most_similar([mdm_alt.docvecs['sci-fi']])
```

```
[('alien', 0.4514704942703247),
 ('express', 0.4008052945137024),
 ('space', 0.4004318714141845 7),
 ('planet', 0.3805035352706909),
 ('ant', 0.3701178431510925 3),
 ('ferocious', 0.3621740341186523 4),
 ('ship', 0.3557941019535064 7),
 ('hole', 0.342262625694274 9)
]
```

Closest words romance

```
model.most_similar([mdm_alt.docvecs['romance']])
[('say', 0.3808212280273437),
 ('skill', 0.3159002363681793),
 ('leads', 0.3063559830188751),
 ('local', 0.3018215596675873),
 ('millionaire', 0.2863730788230896),
 ('located', 0.2845898568630218),
 ('hood', 0.2830425798892975),
 ('heir', 0.2802196145057678),
 ('died', 0.2721515595912933),
 ('indians', 0.2677659392356872)]
```

Closest genres to romance

```
model.docvecs.most_similar('romance')
```

```
[('fantasy', -0.09007323533296585),
 ('sci-fi', -0.0983937606215477),
 ('animation', -0.13281254470348358),
 ('comedy', -0.1537310779094696),
 ('action', -0.16746415197849274)]
```

Bayesian Inversion

RARE TECHNOLOGIES

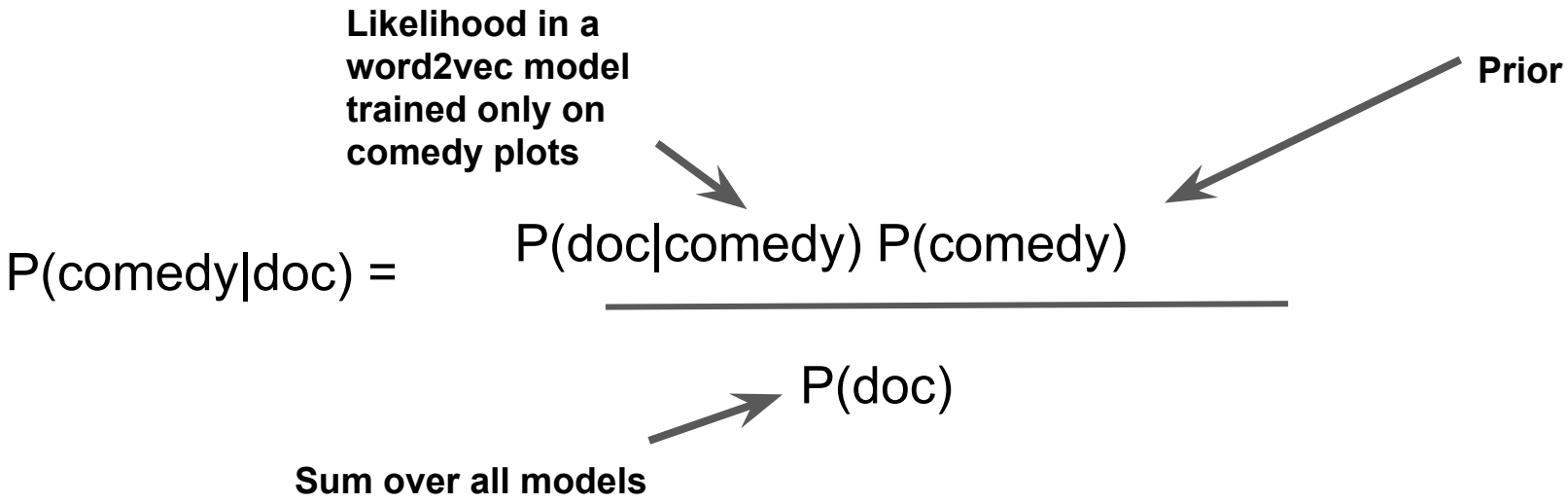
1. Train a word2vec model only on comedy plots.
2. Repeat for each genre. Get 6 models.
3. Take a plot and see which model fits it best

Matt Taddy, “Document Classification by Inversion of Distributed Language Representations”
https://github.com/piskvorky/gensim/blob/develop/docs/notebooks/deepir.ipynb

Bayesian Inversion

Which class does this plot fit in?

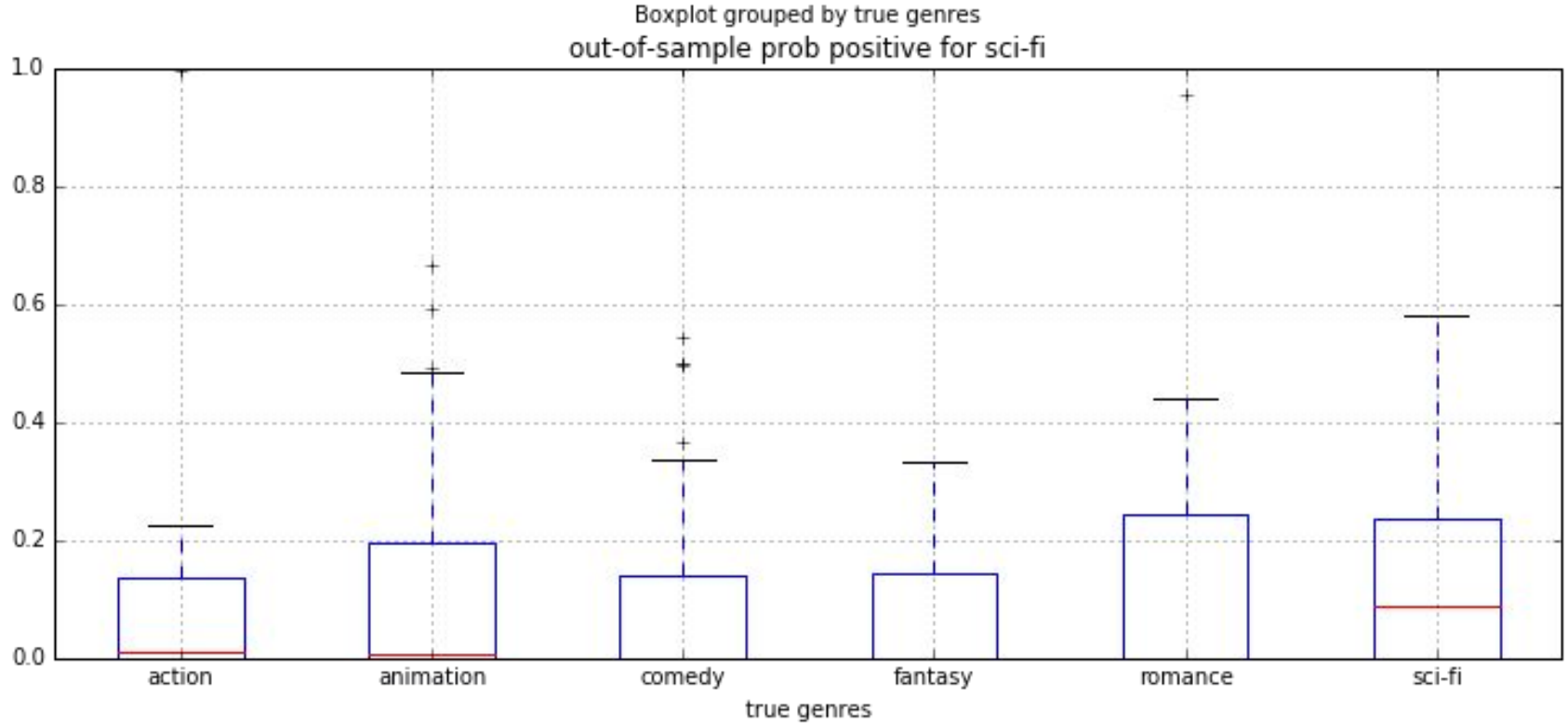
Bayes Rule



Matt Taddy, “Document Classification by Inversion of Distributed Language Representations”
https://github.com/piskvorky/gensim/blob/develop/docs/notebooks/deepir.ipynb

Let’s run some code

Probabilities assigned by sci-fi model to plots of each genre



Disclaimer: First run, "out of the box" performance. No tuning.

RARE TECHNOLOGIES

Bayesian Inversion: only 30%.

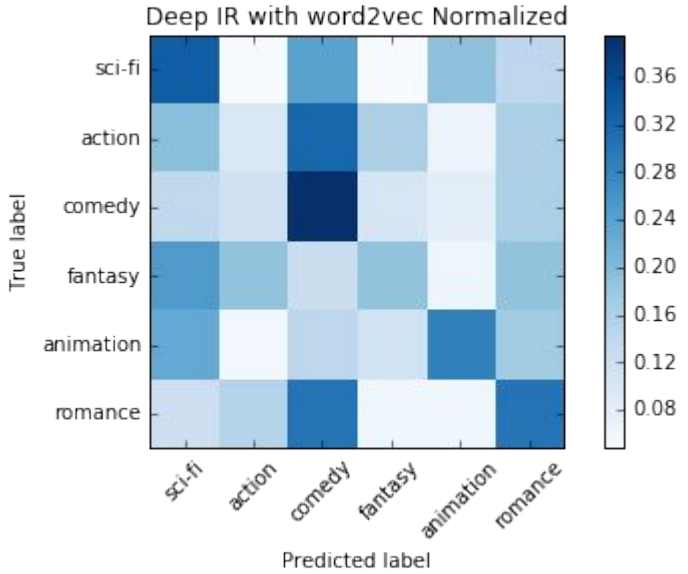
Embedding	Classifier	Accuracy	Train time,s	Predict time, s
Bayesian Inversion	Max likelihood	30	120	1

Usually quite good but corpus too small

Train: six different word2vec models. Use sentences from plots with tag X.

Predict: Average all sentences in a review to get a likelihood.

1000 iterations in word2vec training



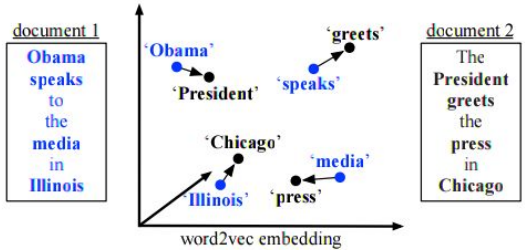
Word Mover's distance

From Word Embeddings To Document Distances

Matt J. Kusner MKUSNER@WUSTL.EDU
Yu Sun YUSUN@WUSTL.EDU
Nicholas I. Kolkin N.KOLKIN@WUSTL.EDU
Kilian Q. Weinberger KILIAN@WUSTL.EDU
Washington University in St. Louis, 1 Brookings Dr., St. Louis, MO 63130

Abstract

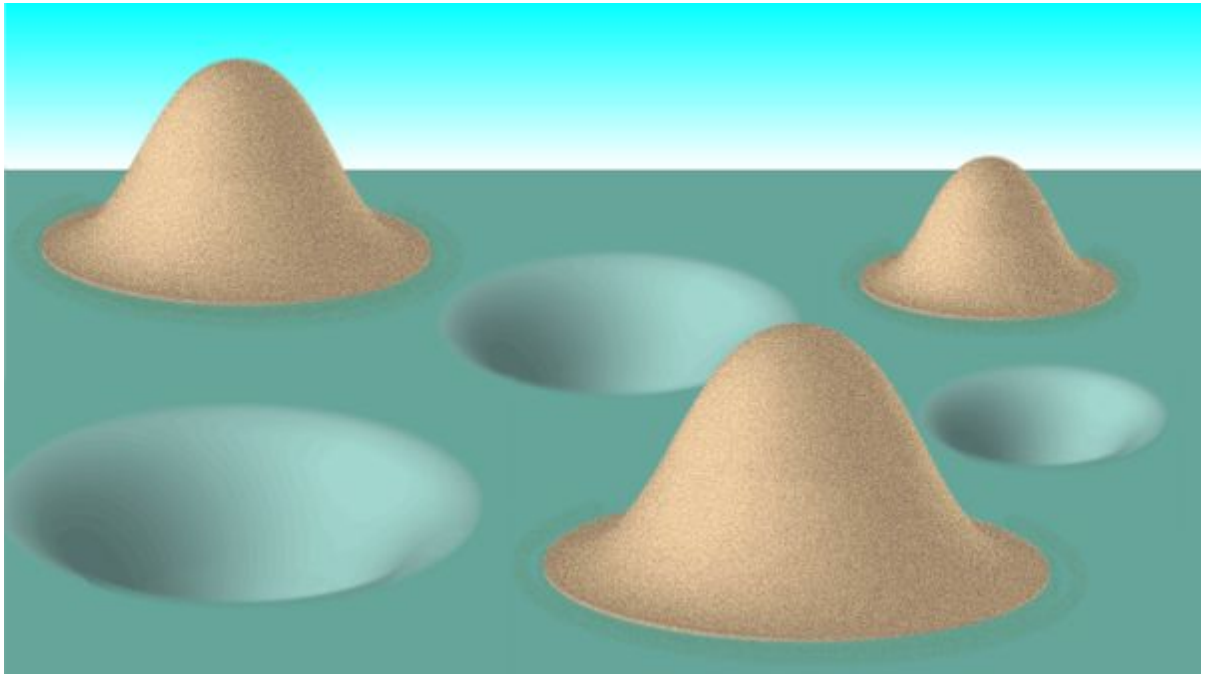
We present the Word Mover's Distance (WMD), a novel distance function between text documents. Our work is based on recent results in word embeddings that learn semantically meaningful representations for words from local co-occurrences in sentences. The WMD distance measures the dissimilarity between two text doc-



http://jmlr.org/proceedings/papers/v37/kusnerb15.pdf
https://github.com/mkusner/wmd

Earth Mover's Distance

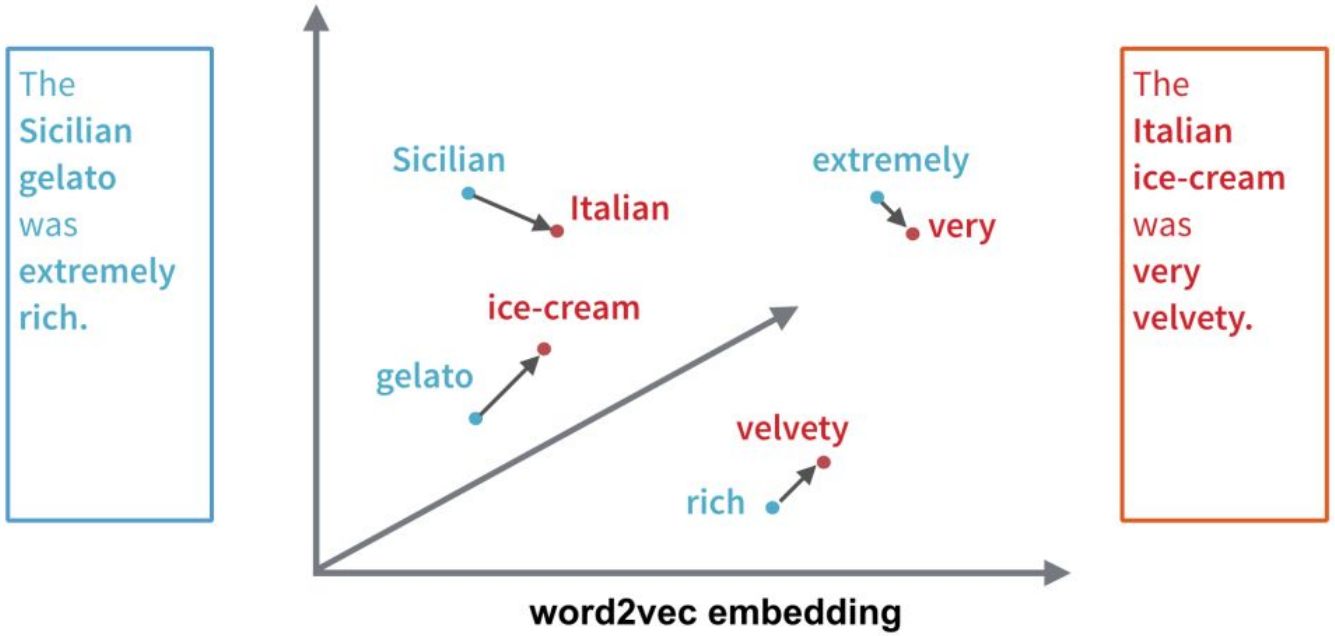
How do you best move piles of sand to fill up holes of the same total volume?



Stated by Monge in 1781. Solved by Kantorovich in

[Image: APS/Alan Stonebraker]

Word Mover's distance



http://tech.opentable.com/2015/08/11/navigating-themes-in-restaurant-reviews-with-word-movers-distance/

RARE TECHNOLOGIES

Distance in gensim

```
from gensim.models import Word2Vec

sentence_gelato = 'The Sicilian gelato was extremely rich'
sentence_icecream = 'The Italian ice-cream was very velvety'
distance = model.wmdistance(sentence_gelato, sentence_icecream) # 1.0175

sentence_orange = 'Oranges are my favorite fruit'
model.wmdistancef(sentence_gelato, sentence_orange) # 1.3363
```

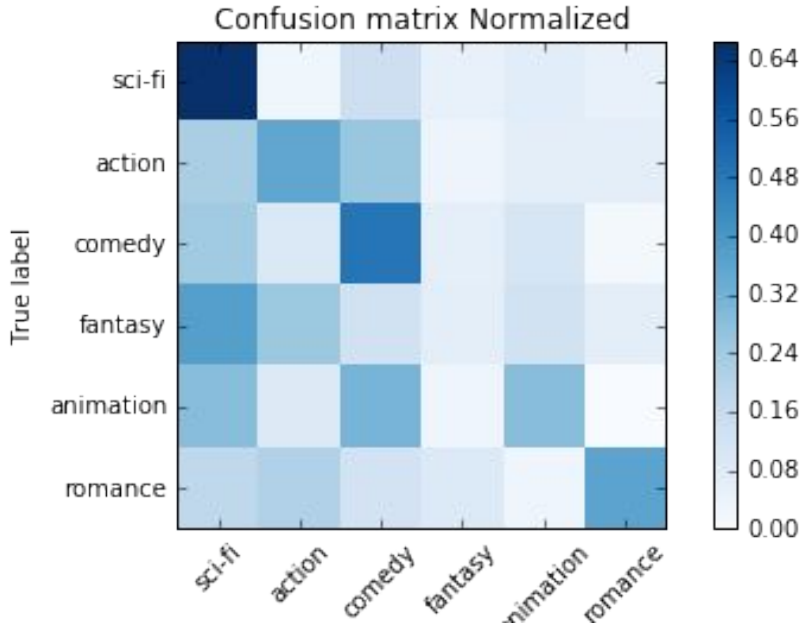
Disclaimer: First run, "out of the box" performance. No tuning.

RARE TECHNOLOGIES

Word Mover's Distance: 42 %

Embedding	Classifier	Accuracy	Train time,s	Predict time, s
Word Movers Distance on Google News	KNN	42	1 (thanks to Google!)	1800

```
WordMoversKNN(
    n_neighbors=1,
    W_embed=wv.syn0norm,
    n_jobs=7)
```



Credit: sklearn api by @vene
http://vene.ro/blog/word-movers-distance-in-python.html

Conclusion

Rough, first run, “out of the box” performance. No tuning.

RARE TECHNOLOGIES

Embedding	Classifier	Accuracy	Train time,s	Predict time on 250 docs, s
Baseline tf-idf bag of words	Logistic	47	2	1
Averaging word vectors from pre-trained Google News	Logistic	**52**	2 (thanks to Google!)	1
Doc2vec	Logistic	**52**	25	1
Bayesian Inversion	Max likelihood	30	120	1
Word Movers Distance on Google News	KNN	42	**1** (thanks to Google!)	1800

No neural network magic out of the box:(

Simple baselines are not much worse than fancy methods.

RARE TECHNOLOGIES

Which model is easiest to tune and debug?

“What caused this error?”

“What is wrong with comedy genre?”

“What can I do to fix this class of errors?”

Rough, first run, “out of the box” performance. No tuning.

RARE TECHNOLOGIES

Embedding	Classifier	Accuracy	Train time,s	Predict time, s	Debug/tune
Baseline tf-idf bag of words	Logistic	47	2	1	**Easy**
Averaging word vectors from pre-trained Google News	Logistic	52	2 (thanks to Google!)	1	Hard
Doc2vec	Logistic	52	25	1	Hard
Bayesian Inversion	Max likelihood	30	120	1	Hard
Word Movers Distance on Google News	KNN	42	1 (thanks to Google!)	1800	Hard

Easiest to understand

Easiest to debug

TF-IDF Bag of words, of course!

RARE TECHNOLOGIES

There is a place for word embeddings

- Small improvement matters (maybe you want your paper accepted)

- Lots of data

- Extra features to existing ensembles

- Need directionality (analogies)

Thanks!

Come and learn about word2vec with us!

Lev Konstantinovskiy

github.com/tmylk

@teagermylk

Events coming up:

- Gensim coding sprint tomorrow Sunday 16 October. Contact me for location.

- Workshop at Yandex Moscow, 26 October