MEETUP TOPICS

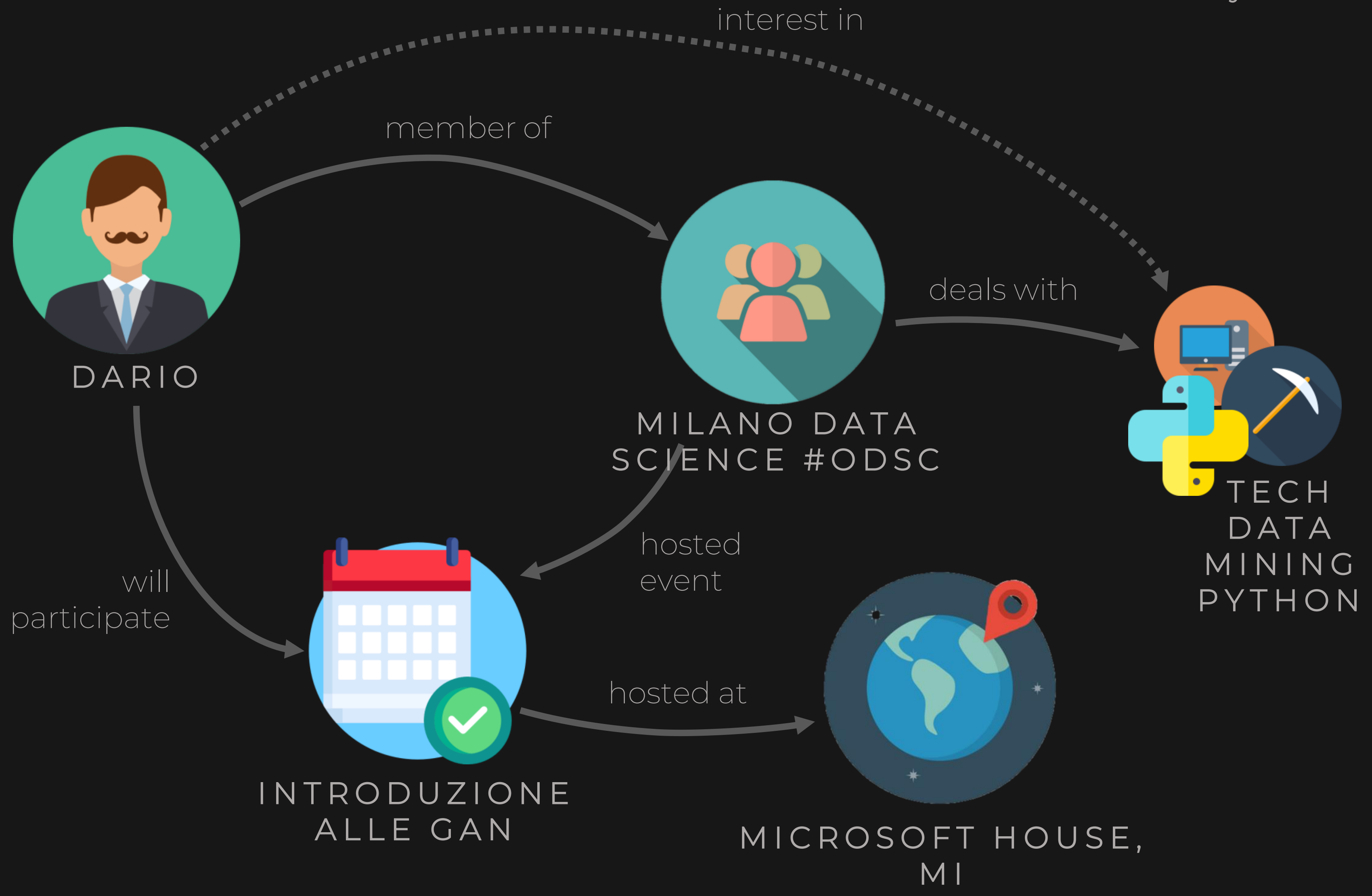
A TEXT MINING AND SEARCH PROJECT

DARIO BERTAZIOLI
FABRIZIO D'INTINOSANTE
MASSIMILIANO PERLETTI

UNIVERSITA' DEGLI STUDI DI MILANO BICOCCA

introduction

available in **186** countries
40 millions users
320k active groups
12k daily events



goal

text classification

topic

Single-Label Multi-Class (SLMC)

pipeline

media
tech
outodoor
business
socializing

from event descriptions predict the event category

PRE-PROCESSING

stemming
lemmatization
tokenization

language detection

LDA clustering

TEXT REPRESENTATION

1 BAG-OF-WORDS
tf-idf
frequency count

2 WORD EMBEDDING
Word2Vec (mean, tf-idf)
Doc2Vec

CLASSIFICATION

Random Forest

Neural Network

DATASET

data

variables

description
event_id
event_name
category

→

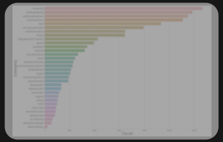
remove

NA values
duplicates

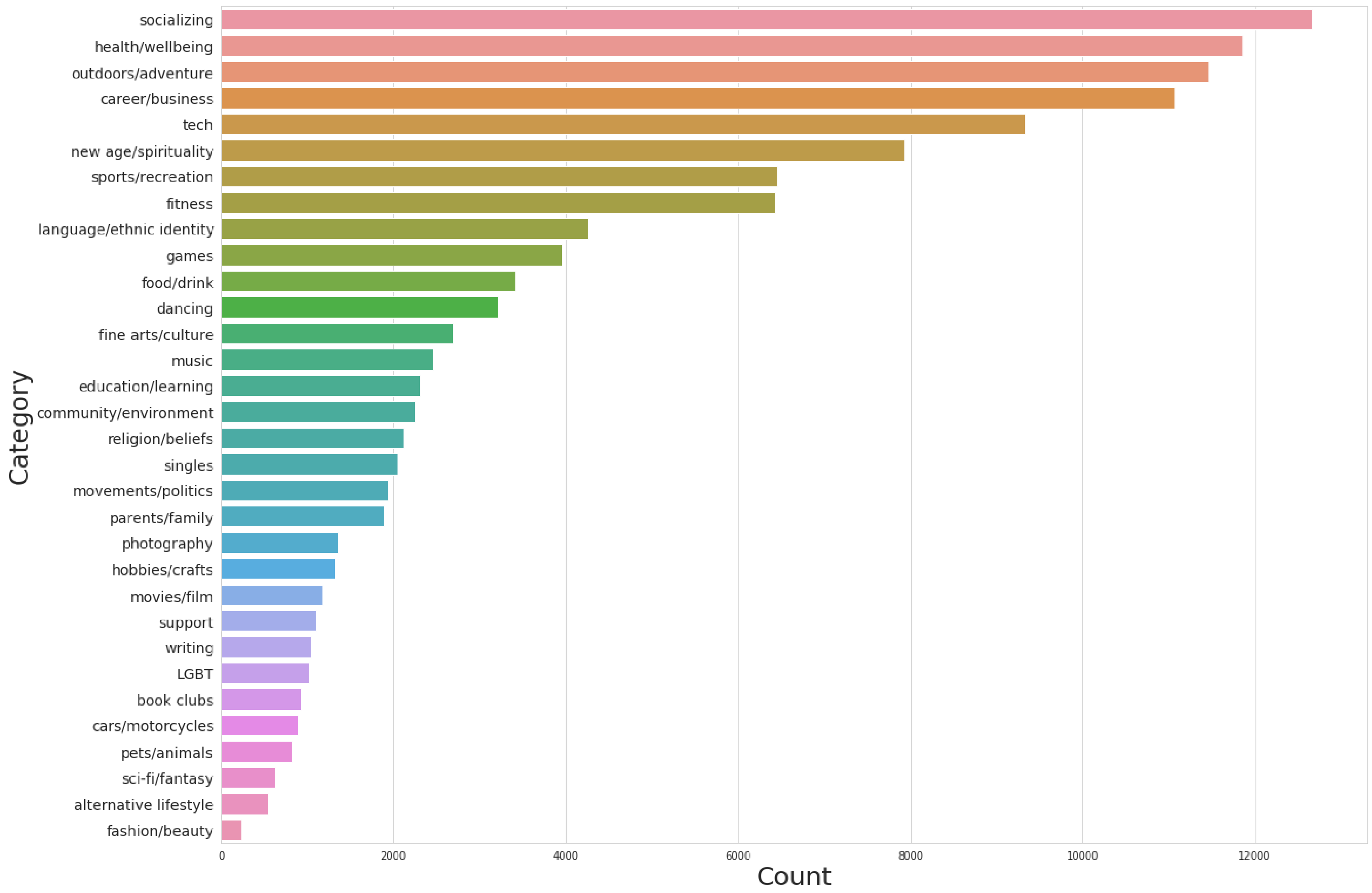
→

134k record

32 class



data



Category
socializing
health/wellbeing
outdoors/adventure
career/business
tech
new age/spirituality
sports/recreation
fitness
language/ethnic identity
games
food/drink
dancing
fine arts/culture
music
education/learning
community/environment
religion/beliefs
singles
movements/politics
parents/family
photography
hobbies/crafts
movies/film
support
writing
LGBT
book clubs
cars/motorcycles
pets/animals
sci-fi/fantasy
alternative lifestyle
fashion/beauty
0
2000
4000
6000
8000
10000
12000
Count

data

union

unbalanced
category

similar
topic

e.g. singles to socializing

Single-Label
Multi-Class
(SLMC)

24 class

PRE-PROCESSING

pre-processing

(garbage in → garbage out)

html & emoji stripping

language detection (minimize api call)

- polyglot

language translation

- Google-translator API
- DeepL API
- python-translate API (Microsoft and other providers)

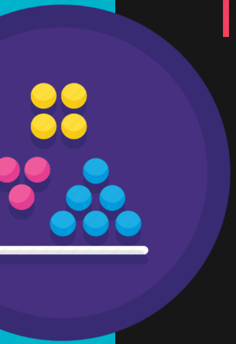
punctuation/special symbols, lowercase

tokenization and stopwords removal

stemming - lemmatization

- SnowballStemmer (multilingual)
- PorterStemmer (English)

latent dirichlet allocation (LDA)



unsupervised method for topic modeling & topic extraction

- generative model (three level bayesian model)

high level idea

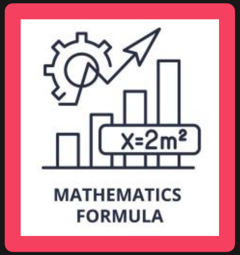
assume your texts comes from a latent-topics generated distribution, try to infer the distribution parameters (thus, the latent topics)

(almost) technically

- "LDA takes de Finetti theorem seriously"
- compute the probability distribution for words in a doc, for a doc in a corpus and for the corpus itself

  exploiting the main property of exchangeability of words and docs
- use Bayesian inference to obtain the posterior distribution of the latent variables

  exploiting variational methods to solve (uncouple) intractable (coupled) equations



interesting note

- some Latent topics are well correspondent with our labels while others have no sense but.. that's a good news!
- clusters of "garbage" helps in defining "badwords" to remove in the cleaning process → ~ 1/2% performance gain by only stripping ~50 badwords (the previously introduced "trick")

latent dirichlet allocation (LDA)

unsupervised m

- generat

high level i

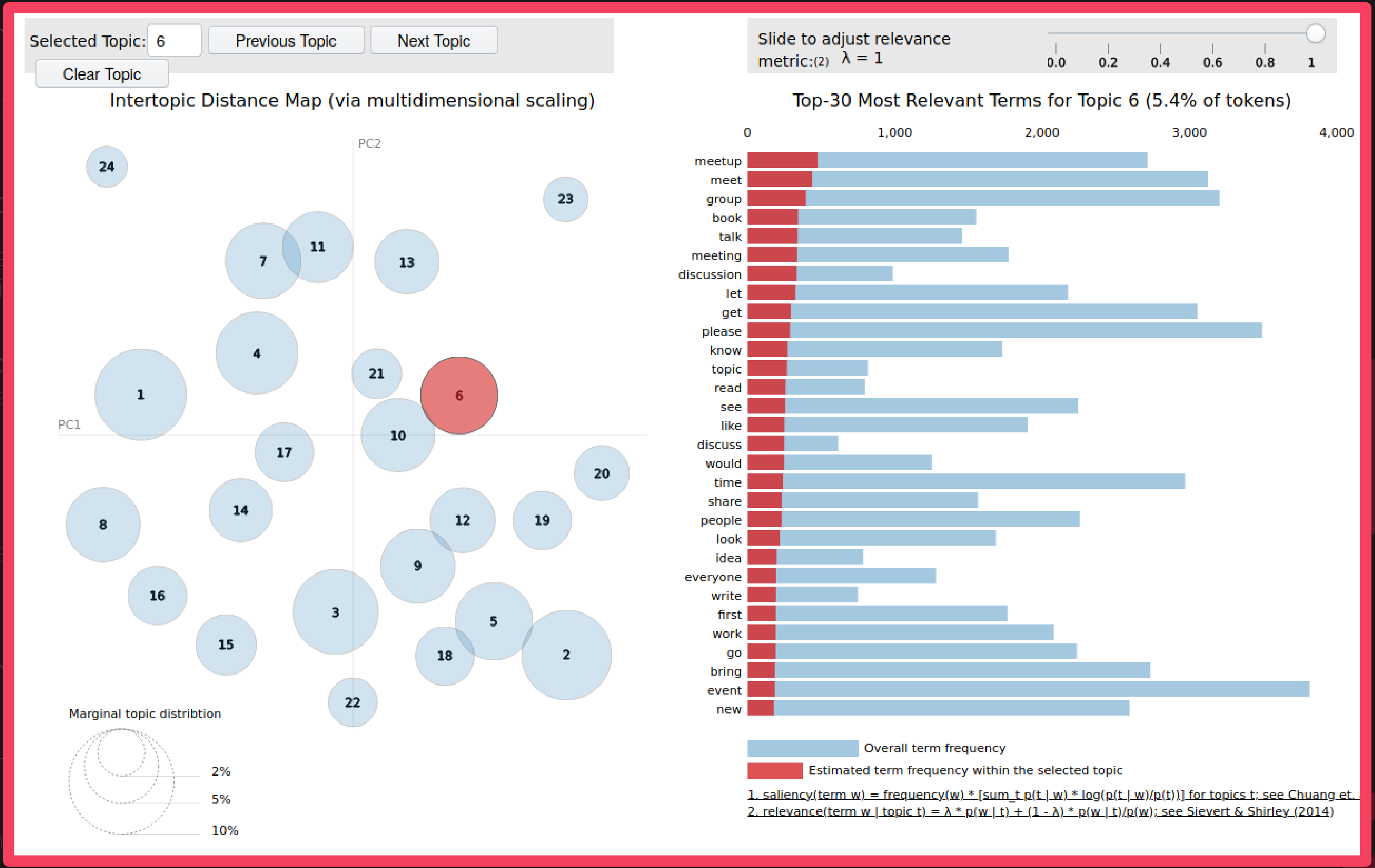
assume your texts
infer the distributi

(almost) technic

- “LDA take
- compute
- use Bayes

interesting r

- some Late
good news
- clusters of
performan



TEXT REPRESENTATION

bag-of-words

1 COUNT

$$\sum$$

TF-IDF 2

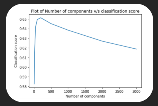
tf-idf vectorization

$$\text{tf-idf(t,d)} = \text{tf(t,d)} \times \text{idf(t)}$$

where

$$\text{idf}(t) = \log \frac{1+n}{1+\text{df}(t)} + 1,$$

sparsity analysis

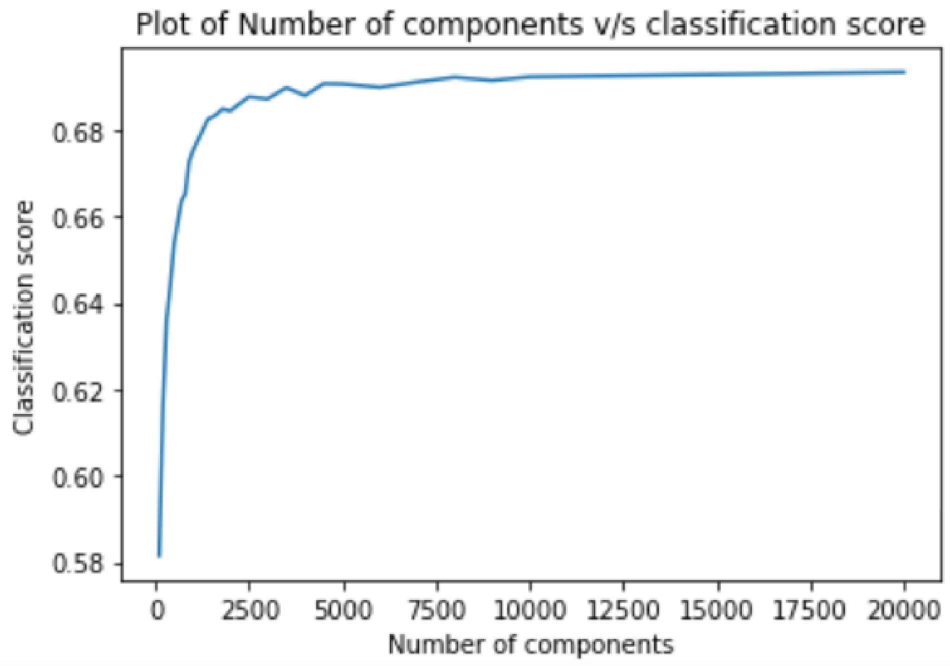


DIMENSIONALITY REDUCTION (SDV)

B

A

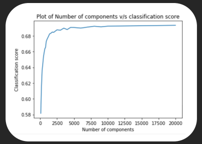
CUT-OFF THRESHOLD



sparsity analysis

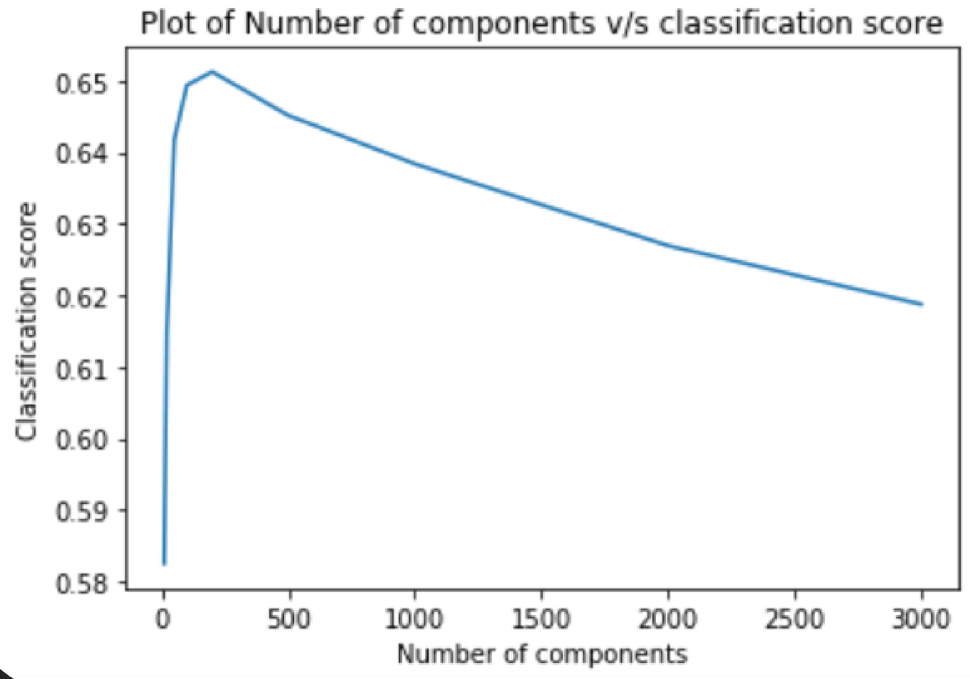
A

CUT-OFF THRESHOLD

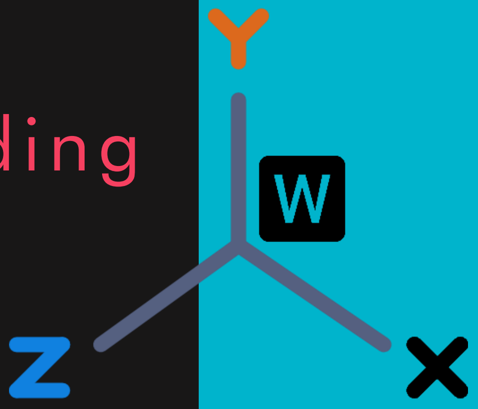


DIMENSIONALITY REDUCTION (SDV)

B



word embedding



1 WORD2VEC

DOC2VEC 2

1

WORD2VEC

TRANSFER LEARNING

word embedding

DOC2VEC

2

word embedding

1 WORD2VEC

TRANSFER LEARNING

CBOW

Window size - 5
Feature dim - 300
Epochs - 10

sintetization strategies

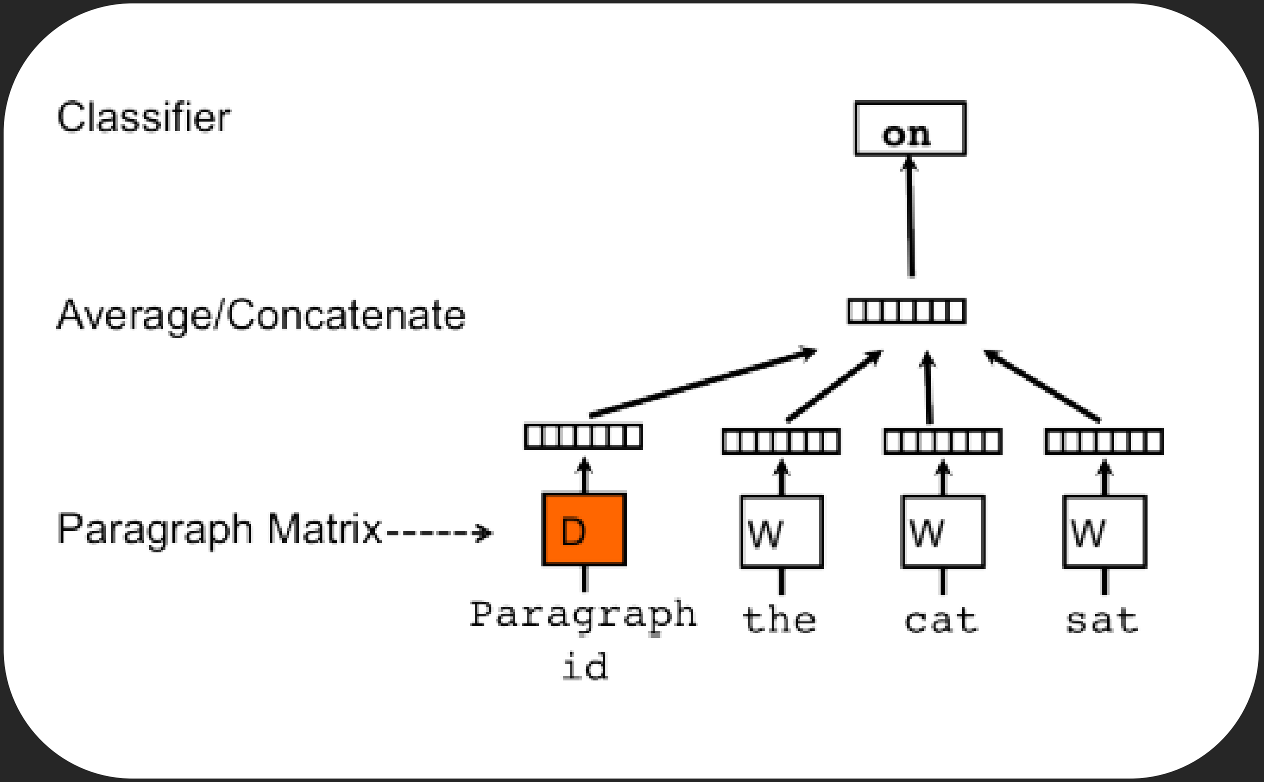
MEAN

TF-IDF

DOC2VEC 2

1 WORD2VEC

word embedding



DOC2VEC 2

1 WORD2VEC

word embedding

PV-DM

PV-DM
conctenate
PV-DBOW

PV-DBOW

PV-DBOW

Feature dim - 300
Epochs - 10

DOC2VEC 2

CLASSIFICATION

classifier

R F

default

parameters

(n_estimator – 100)

N N

hyper-parameters

optimization

classifier

OPTUNA

Random Forest – model surrogate
LCB – activate function

Cross Validation (5-folds)

100 iterations – budget

parameters

neurons (layer dense)
rate (dropout layer)
optimizer and **learning rate**
activation layer (ReLU or LeakyReLU)

N N

objective function

1 – average macro f-measure

RESULTS

first evaluation

1

RF

test each **text representation**

Stemming

Stemming + Badwords Removal

Lemmatization

Lemmatization + Badwords Removal

processing method

first evaluation

1

RF

test each **text representation**

Stemming
Stemming + Badwords Removal
Lemmatization
Lemmatization + Badwords Removal

processing method

Stemming + Badwords Removal

BOW (tf-idf)
W2V

Lemmatization + Badwords Removal

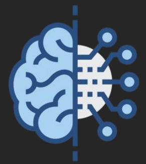
BOW (count)
D2V

evaluation model

2

best processing method
best text representation

RF



NN

evaluation model

2

best processing method
best text representation

RF

NN

Lemmatization + Badwords Removal

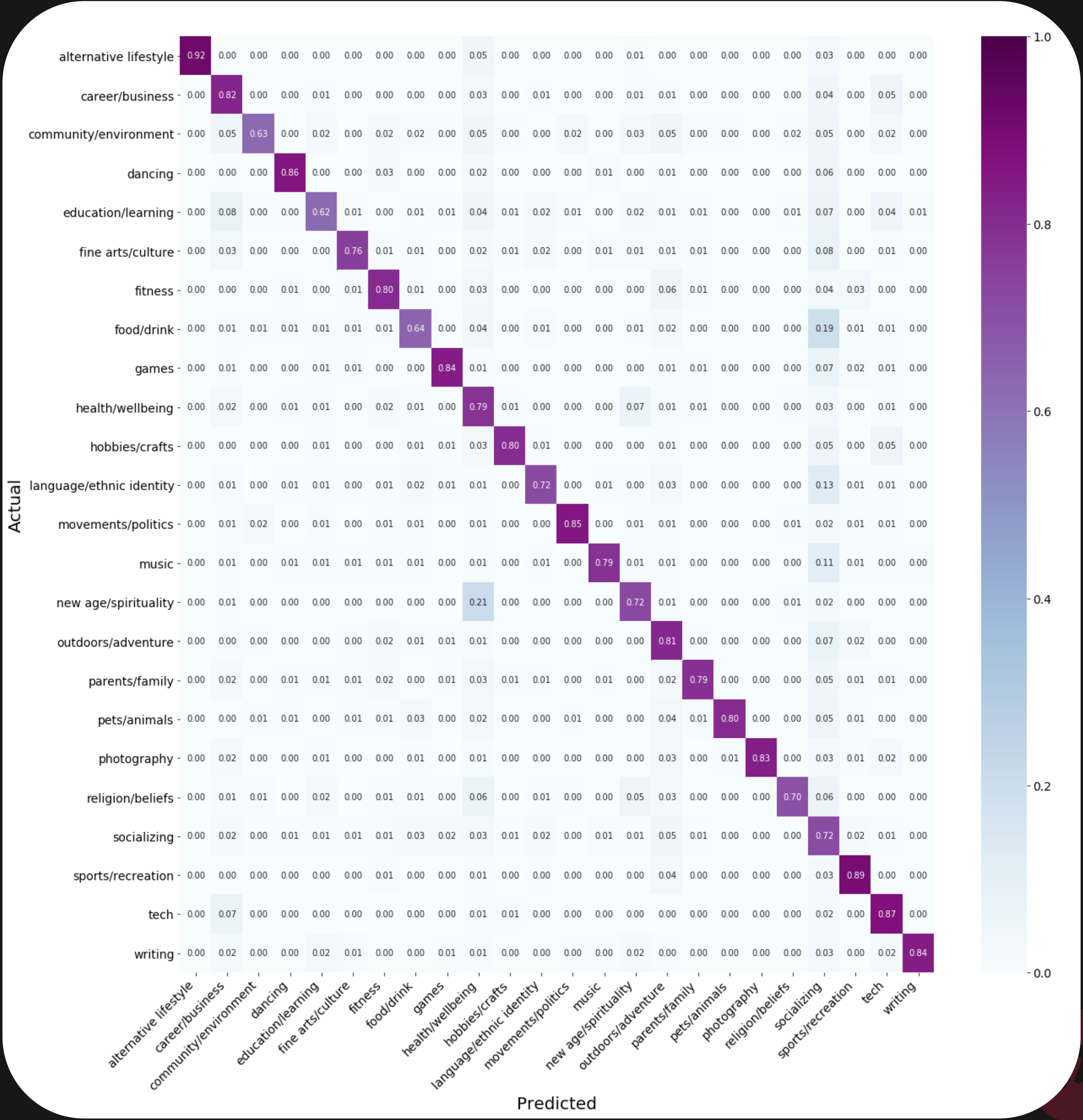
D2V

NN

79,6%

Macro F-measure

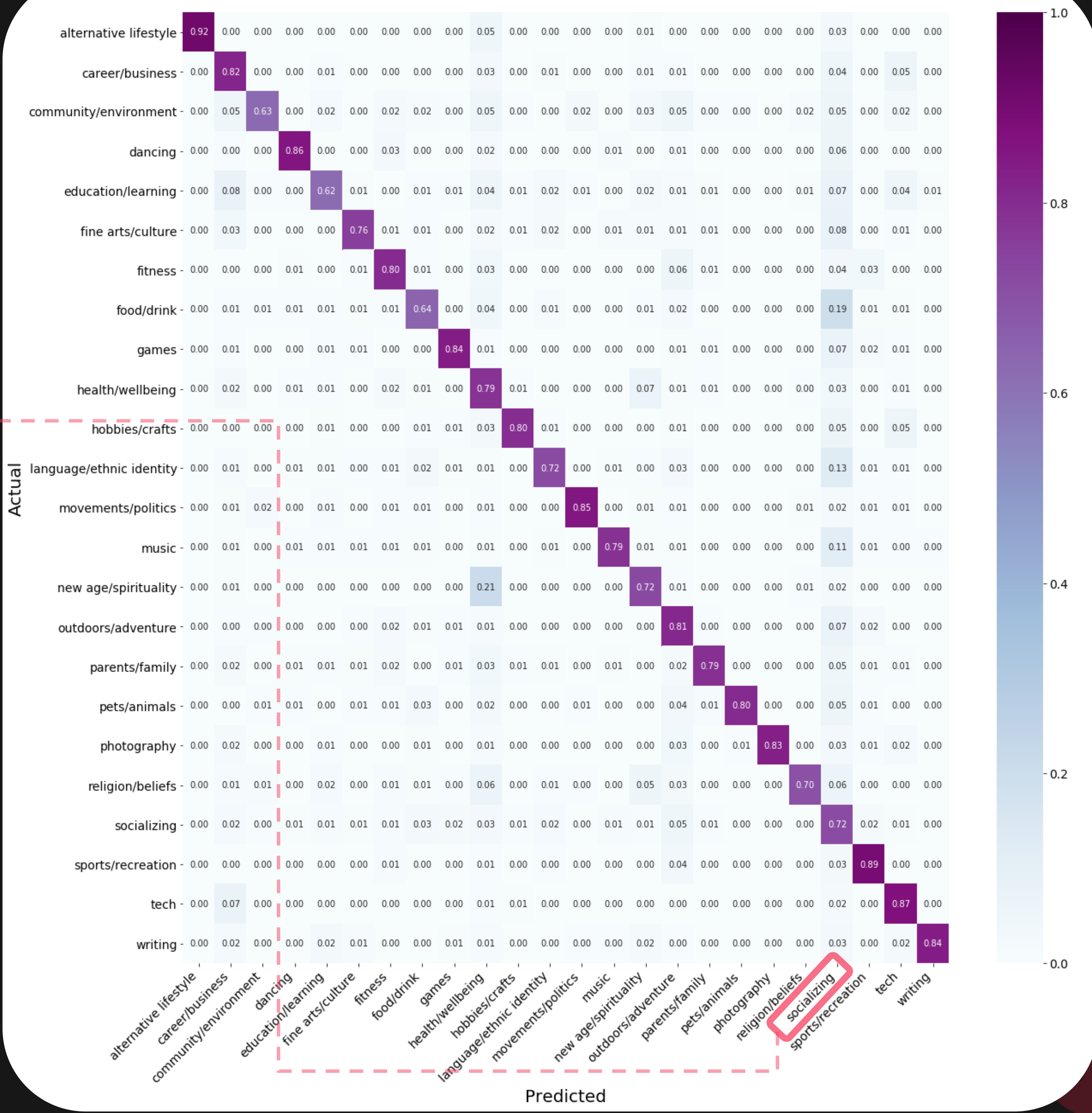
final analysis



final analysis

socializing

most **confusing** category

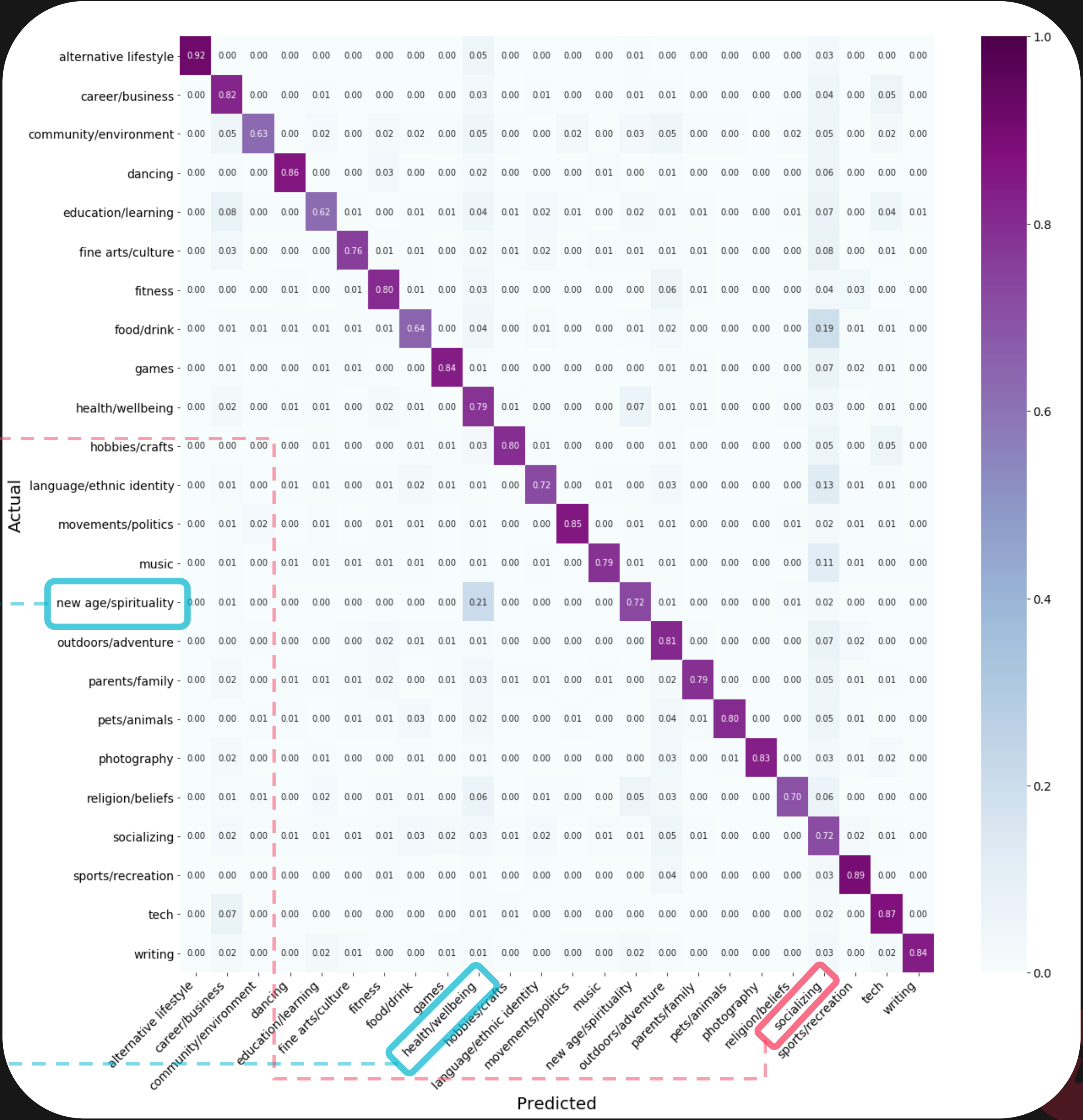


final analysis

socializing

most **confusing** category

new age/spirituality
misclassified
health/wellbeing



CONCLUSIONS

final method

lemmatization and **badwords removal**

doc2vec

NN

78,4%

Top-1 accuracy

94,1%

Top-3 accuracy

final method

lemmatization and **badwords removal**

doc2vec

NN

78,4%
Top-1 accuracy

94,1%
Top-3 accuracy

improvements

more **data**
(under-represented categories)

more **in-depth LDA analysis**
(varying number of produced cluster to select better badwords)

test another algorithm to **improve embeddings**

BERT

THANK YOU

MEETUP TOPICS

(exchangeability) $\longrightarrow$ $p(z_1, \ldots, z_N) = p(z_{\pi(1)}, \ldots, z_{\pi(N)})$

By de Finetti's theorem: $\longrightarrow$ $$p(\mathbf{w}, \mathbf{z}) = \int p(\theta) \left(\prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n) \right) d\theta$$

corpus

Dirac-like topic mixture distribution

distr. parameters

latent topic

n-th word for d-th document

$$p(\mathcal{D} \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d$$

$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)$

joint distribution of a topic mixture θ,
a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$

$p(\mathbf{w} \mid \alpha, \beta)$

marginal distribution of a single document

probability of the entire corpus

Inference: $\longrightarrow$ $$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)}{p(\mathbf{w} \mid \alpha, \beta)}$$

producing the intractable eq (coupled!)
need for variational methods to solve (approx)
-> decoupling

$$p(\mathbf{w} \mid \alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\theta,$$

Processing method	Stemming			Stemming+Badwords Removal		
Feature Extraction Method	Acc	Macro F-Measure	Weighted F-Measure	Acc	Macro F-Measure	Weighted F-Measure
Count	0.692	0.659	0.687	0.692	0.659	0.687
Tf-idf	0.689	0.656	0.685	**0.691**	**0.660**	**0.686**
W2V Tf-idf	0.679	0.639	0.675	**0.683**	**0.648**	**0.680**
W2V Mean	0.668	0.621	0.663	**0.677**	**0.637**	**0.674**
Doc2Vec	0.735	0.727	0.735	0.743	0.737	0.743

Processing method	Lemmatization			Lemmatization+Badwords Removal		
Feature Extraction Method	Acc	Macro F-Measure	Weighted F-Measure	Acc	Macro F-Measure	Weighted F-Measure
Count	0.692	0.661	0.687	**0.693**	**0.662**	**0.688**
Tf-idf	0.688	0.656	0.684	0.688	0.658	0.684
W2V Tf-idf	0.682	0.641	0.678	0.683	0.644	0.679
W2V Mean	0.670	0.623	0.665	0.674	0.631	0.670
Doc2Vec	0.746	0.742	0.746	**0.750**	**0.745**	**0.750**

Model	Processing method	Feature Extraction	Acc. value	Acc. std	Top-3 Acc. value	Top-3 Acc. std	Macro F-Meas. value	Macro F-Meas. std	Weighted F-Meas. value	Weighted F-Meas. std
NN	Lemm.+BR	Count	0.693	0.002	0.907	0.001	0.673	0.004	0.693	0.002
NN	Stemm.+BR	Tf-idf	0.692	0.003	0.908	0.001	0.671	0.003	0.691	0.003
NN	Stemm.+BR	W2V Tf-idf	0.695	0.002	0.901	0.001	0.678	0.002	0.693	0.002
NN	Stemm.+BR	W2V Mean	0.692	0.003	0.901	0.002	0.671	0.005	0.690	0.004
NN	Lemm.+BR	Doc2Vec	**0.784**	0.002	**0.941**	0.001	**0.796**	0.001	**0.784**	0.002
RF	Lemm.+BR	Count	0.693	0.004	0.865	0.002	0.663	0.005	0.689	0.004
RF	Stemm.+BR	Tf-idf	0.693	0.004	0.865	0.002	0.664	0.005	0.689	0.004
RF	Stemm.+BR	W2V Tf-idf	0.685	0.001	0.870	0.001	0.650	0.002	0.682	0.001
RF	Stemm.+BR	W2V Mean	0.678	0.002	0.874	0.001	0.638	0.004	0.674	0.002
RF	Lemm.+BR	Doc2Vec	0.751	0.003	0.911	0.001	0.746	0.003	0.751	0.003