UNIVERSITÀ DEGLI STUDI DI PADOVA

Network Science

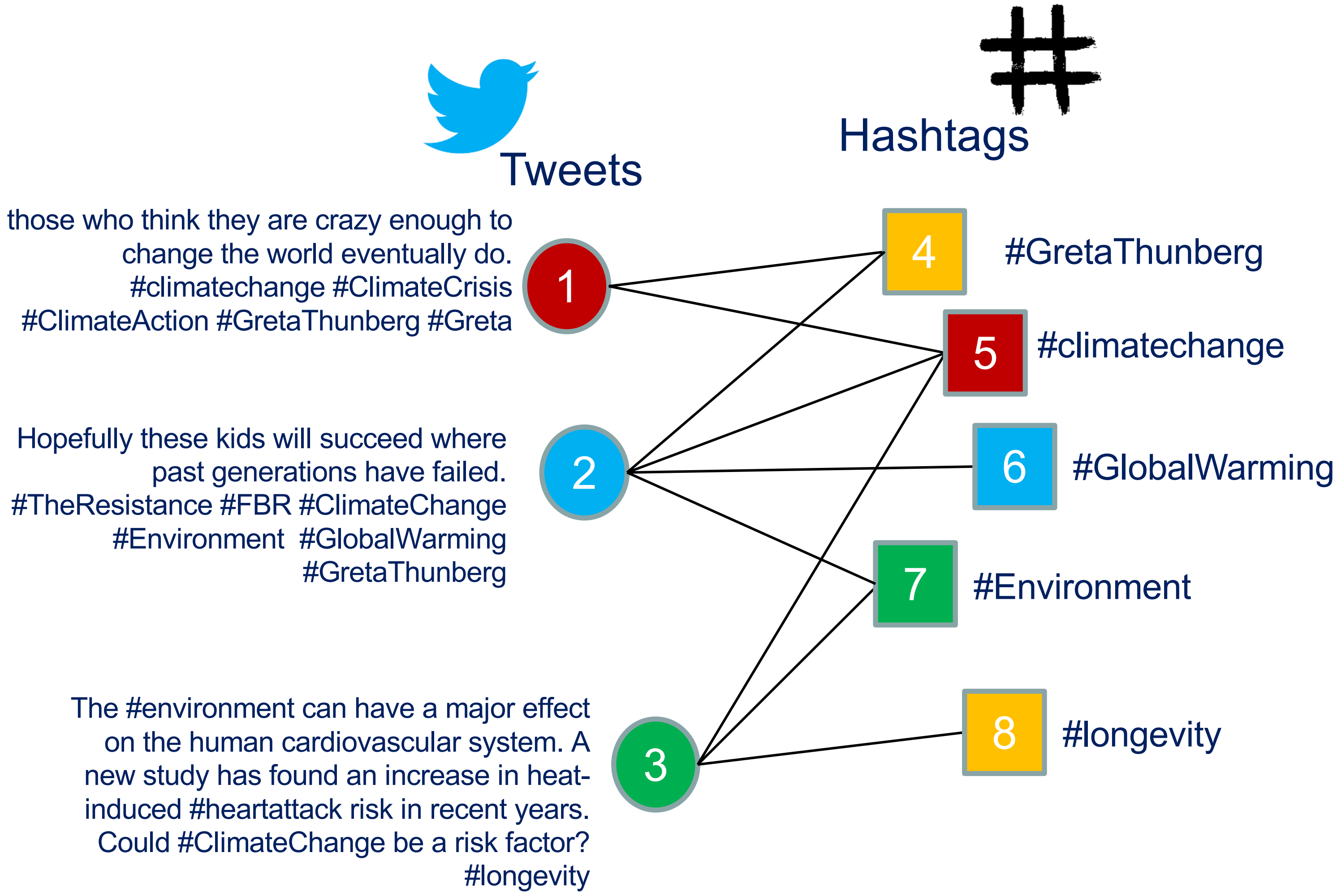
A.Y. 23/24

ICT for Internet & multimedia, Data science, Physics of data

Semantic networks

network science tools for their study

Conceptual picture
of a semantic network on Twitter



An overview

what we will be investigating about semantic networks

- Data collection + polishing
- Building the semantic network (bipartite/projections)
- Topic (i.e., community) detection
  - Modularity & InfoMap
  - Non-negative matrix factorization (NMF)
  - Latent Dirichlet allocation (LDA)
  - Variational auto-encoders (VAE)
  - Embeddings and BERTopic

Data collection

how to get data from the Internet using APIs

Twitter

no longer available unless you pay 5k$ per month ☹

https://developer.twitter.com/en/portal/dashboard

Twitter's plan to cut off free data access evokes 'fair amount of panic' among scientists

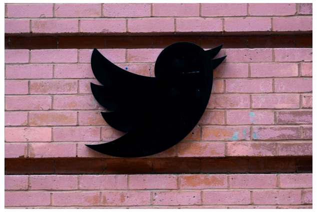
Social media platform's intent to increase revenue could end or limit many research projects

8 FEB 2023 • 4:35 PM ET • BY KAI KUPFERSCHMIDT

Twitter's plan to charge researchers for data access puts it in EU crosshairs

Elon Musk's social media giant plans to charge academics to access its data – in potential violation of Europe's content rules

BY MARK SCOTT
MARCH 22, 2023



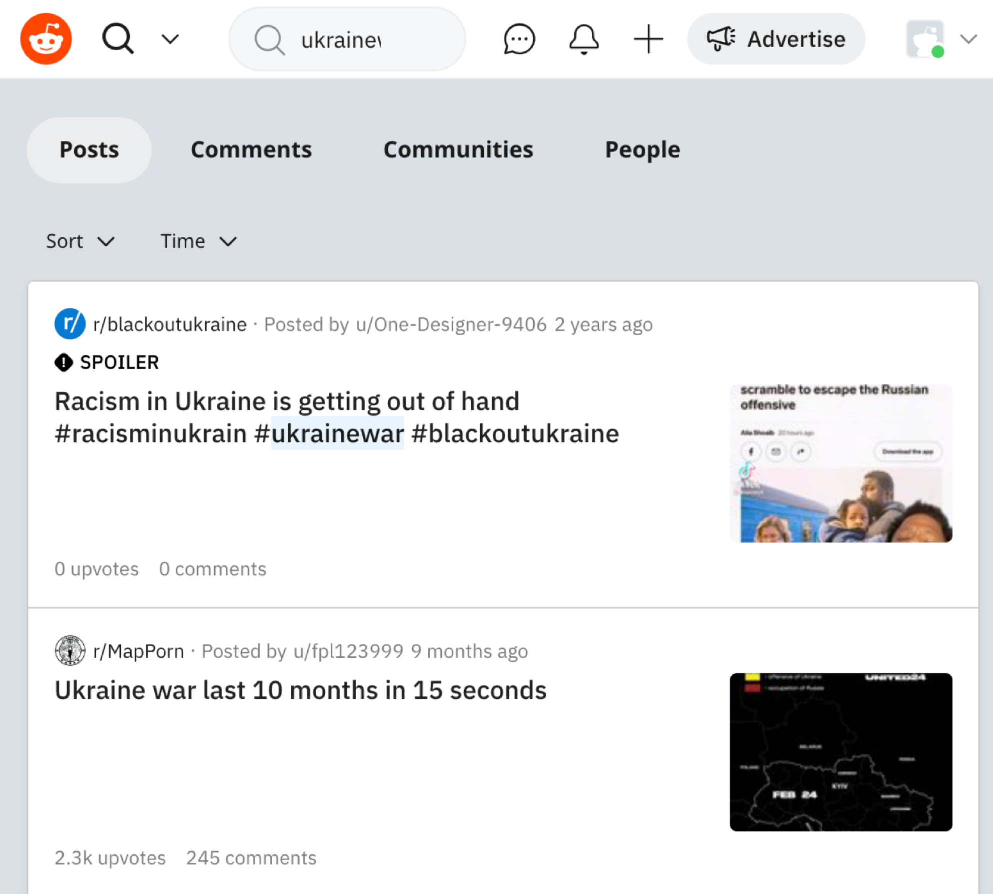
Academic researchers blast Twitter's data paywall as 'outrageously expensive'

By Brian Fung, CNN
Published 11:40 AM EDT, Wed April 5, 2023

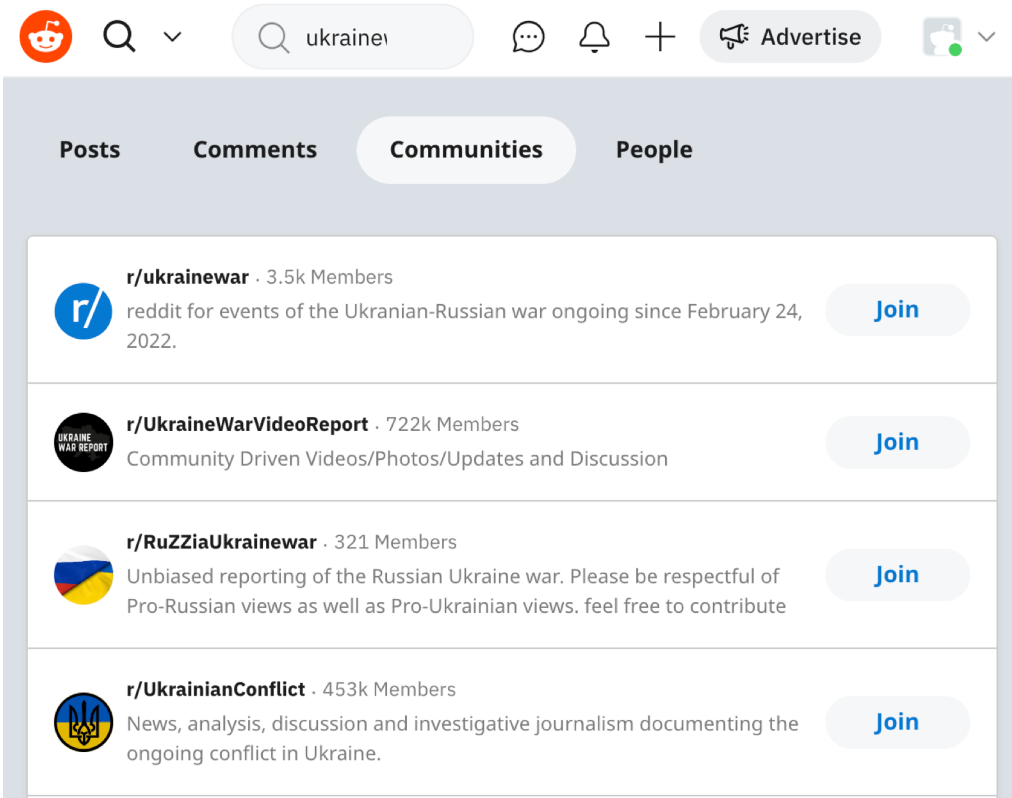
Reddit

https://www.reddit.com

Reddit



Subreddit



UNIVERSITÀ DEGLI STUDI DI PADOVA

Reddit apps

https://www.reddit.com/prefs/apps

GET NEW REDDIT | MY SUBREDDITS | HOME - POPULAR - ALL - RANDOM - USERS | | ASKREDDIT - GAMING - PICS - TODAYILEARNED - FUNNY - WORLDNEWS - NEWS - MOVIES - M

reddit PREFERENCES options apps RSS feeds friends blocked password/email delete Upbeat-Lychee-6630 (1) |

developed applications

ns2023

personal use script

Qbdk-FkA9jSQB9T7drY8UQ

client_id

download reddit context for the network science course, at the unioversity of Padova, of which I am the instructor

change icon

secret jtGPdqiaTj6hCWcvRPeS_nMNEnVkxw

client_secret

developers Upbeat-Lychee-6630 (that's you!) remove

username

add developer:

name ns2023

description download reddit context for the network science course, at the unioversity of Padova,

about url

redirect uri https://localhost:8080

update app

delete app

create another app...

register asap… will be using this in the 1st lab

Python Reddit API Wrapper – PRAW

https://praw.readthedocs.io/en/stable/ (v7.7.1)

```
!pip install praw
```

your own description of your app, including version and username

from Reddit apps

```
import pandas as pd
import praw
reddit = praw.Reddit(client_id='Qbdk-FkA9jSQB9T7drY8UQ',
                     client_secret="jtGPdqiaTj6hCWcvRPeS_nMNEnVkxw",
                     user_agent='reddit scraper 1.0 by u/Upbeat-Lychee-6630',
                     check_for_async=False)
print(reddit.read_only)
```

```
True
```

"relevance", "hot", "top", "new", or "comments"

max 250 per call

```
df = pd.DataFrame([vars(post) for post in reddit.subreddit("all")
                   .search("#ukrainewar", sort='top', limit=10)])
```

can also add time_filter = "all", "day", "hour", "month", "week", or "year"

```
df.to_excel('drive/MyDrive/Colab Notebooks/samples.xlsx',
            index=True)
```

Reddit data
an example

there is a list of 116 entries per post, on which you can choose!!!

from this you extract the date

	title	created	score	upvote_ratio	ups	num_comments	selftext
0	Damn...we blinked and missed the T-34 stage of...	1.666899e+09	10394	0.99	10394	738	
1	Finnish🇫🇮 volunteer sends greetings home from ...	1.680237e+09	2095	1.00	2095	57	
2	Guess having 5 trucks fall into your office ca...	1.663341e+09	1974	1.00	1974	88	
3	[META] Important - Russia-Ukraine Crisis/War: ...	1.645712e+09	1284	0.89	1284	1	Hi, /u/Anonim97 here.\n\nWe - as a mods of 40k...
4	V*tniks coping hard Over the counter offensive...	1.662956e+09	1081	1.00	1081	86	
...	...	...	...	...	...	...	...
0	Russia Ukraine War.	1.695349e+09	1	1.00	1	0	

TikTok

developers.tiktok.com

TikTok for developers

Products Docs Support Blog

- Content Posting API
- Display API
- Research API
  - About Research API
  - Getting Started
  - Frequently Asked Questions
  - Codebook
  - API Reference
    - Query Videos
    - Query User Info
    - Query Video Comments

Query Videos

Request

HTTP URL	https://open.tiktokapis.com/v2/research/video/query/
HTTP Method	POST
Scopes	research.data.basic

Headers

Key	Type	Description

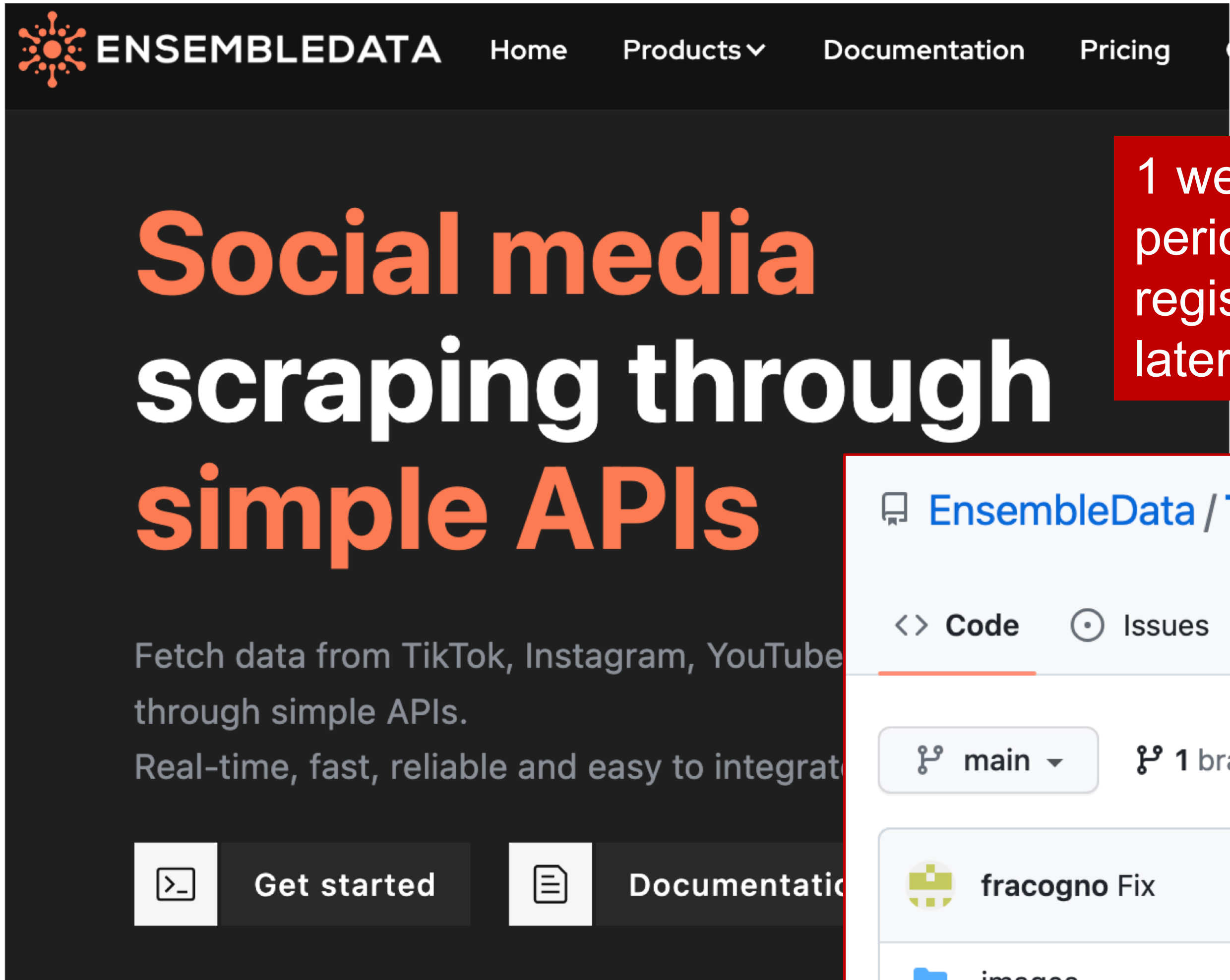
TikTok for developers

TikTok Research API application received

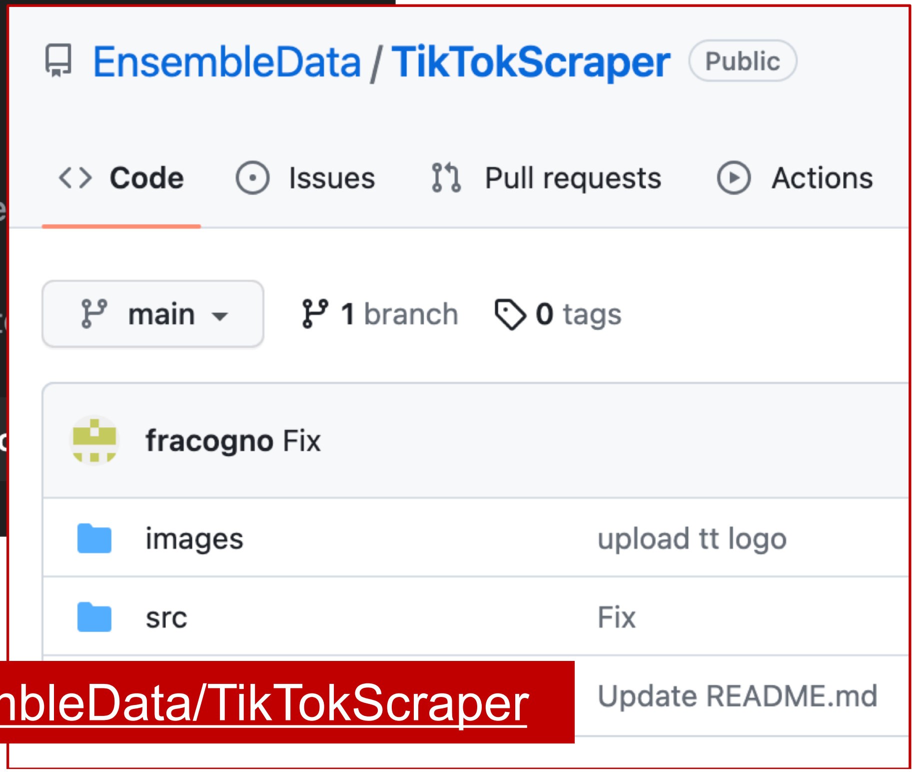
Thank you for applying to gain access to TikTok's research APIs. We have received your application and will proceed with the review process. You will be notified of the result via email within 3-4 weeks. If you have any questions, please contact our Support team.

TikTok

https://www.ensembledata.com/



1 week trial period… register later on



https://github.com/EnsembleData/TikTokScraper

Data preprocessing

how to polish raw data from the Internet

NLP cleaning process
of text sources

1. Superficial cleaning

Removing website links
Removing accented characters
Removing text inside square brackets
Removing moderator messages
Removing double spaces
Removing non-text special words and characters
Removing extra-used new lines
Limiting all the repetitions to two characters and removing the extra characters
Removing punctuation except main sentence punctuation
Removing sentences that represent the rules of the community

Fixing **contractions**
Removing **emoji**
Removing **hashtags** and **mentions**
Removing **numbers**
Lowercasing
Correct **spellings**

2. Subsentence

Tokenise subsentences

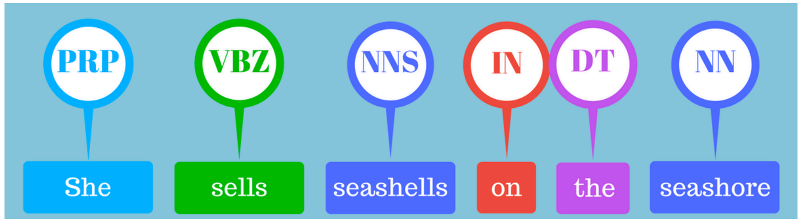
useful for long text samples (e.g., Reddit)

the bare minimum to polish the text, useful as an input to sentiment analysis

3. Deep cleaning

Stop word removal
Word tokenization
POS tagging
Lemmatization

truly polished text, useful for building a semantic network



SpaCy part-of-speech (POS) tags

https://spacy.io/

POS	description	example
ADJ	**adjective**	big, old, green, incomprehensible, first
ADP	adposition	in, to, during
ADV	**adverb**	very, tomorrow, down, where, there
AUX	auxiliary	is, has (done), will (do), should (do)
CONJ	conjunction	and, or, but
CCONJ	coordinating conjunction	and, or, but
DET	determiner	a, an, the
INTJ	interjection	psst, ouch, bravo, hello
NOUN	**noun**	girl, cat, tree, air, beauty
NUM	numeral	1, 2017, one, seventy-seven, IV, MMXIV

POS	description	example
PART	particle	’s, not,
PRON	**pronoun**	I, you, he, she, myself, themselves, somebody
PROPN	**proper noun**	Mary, John, London, NATO, HBO
PUNCT	punctuation	., (,), ?
SCONJ	subordinating conjunction	if, while, that
SYM	symbol	$, %, §, ©, +, −, ×, ÷, =, :), 😝
VERB	**verb**	run, runs, running, eat, ate, eating
X	other	sfpksdpsxmsa
SPACE	space	

Example

ukrainewar from Reddit

true date

superficial cleaning

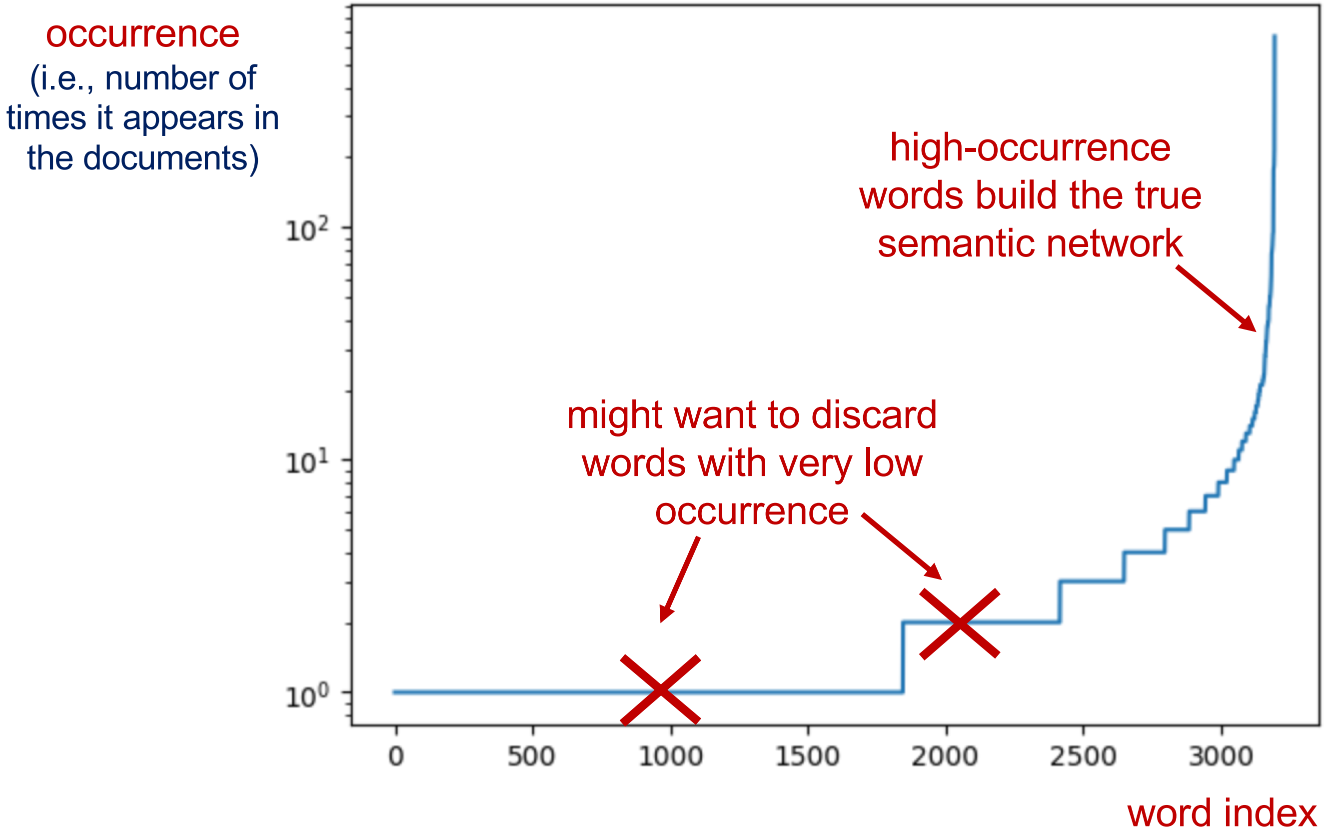
deep cleaning

	title	created	score	upvote_ratio	selftext	title_sup_clean	title_deep_clean	title_deep_clean_pos
	Damn...we blinked and missed the T-34 stage of...	2022-10-27	10390	0.99	NaN	damn we blinked and missed the t stage of the ...	damn blink miss t stage war	[damn ADV, blink VERB, miss VERB, t PROPN, sta...
	Finnish🇫🇮 volunteer sends greetings home from ...	2023-03-31	2095	1.00	NaN	finnish volunteer sends greetings home from so...	finnish volunteer send greeting home	[finnish ADJ, volunteer NOUN, send VERB, greet...
	Guess having 5 trucks fall into your office ca...	2022-09-16	1980	1.00	NaN	guess having trucks fall into your office can ...	guess have truck fall office significant emoti...	[guess VERB, have VERB, truck NOUN, fall VERB,...
	[META] Important - Russia-Ukraine Crisis/War: ...	2022-02-24	1280	0.89	Hi, /u/Anonim97 here.\n\nWe - as a mods of 40k...	important russia ukraine crisis war info and ...	important russia ukraine crisis war info way help	[important ADJ, russia PROPN, ukraine PROPN, c...
	V*tniks coping hard Over the counter offensive...	2022-09-12	1076	1.00	NaN	v tniks coping hard over the counter offensive...	tnik cope hard counter offensive traitor pfp lmao	[tnik NOUN, cope VERB, hard ADJ, counter NOUN,...
	...	...	...	...	...	...	...	...

only ADJ, ADV, NOUN, PRON, PROPN, VERB kept

Example

always check words occurrencies



Building the semantic network

bipartite and projected counterparts

UNIVERSITÀ DEGLI STUDI DI PADOVA

Probability matrices
linking words to documents

number of occurrences of words in documents

$$N_{wd} = \begin{array}{c} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{l} \text{\#globalwarming} \\ \text{\#climatechange} \\ \text{\#climateaction} \\ \text{\#gretathunberg} \\ \text{\#environment} \end{array}$$

probability of words given a documents

$$P_{w|d} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{5} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{5} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{1}{5} & 0 \\ 0 & 0 & \frac{1}{5} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{1}{5} & \frac{1}{4} \end{bmatrix}$$

we capture the statistical properties by normalizing by columns

we identify a document probability

$$p_d = \begin{cases} \dfrac{1}{D} & \text{equally likely} \\ \dfrac{n_d}{\sum_d n_d} & \text{custom} \end{cases}$$

Probability matrices

projecting to words or documents

bipartite network

joint probability of words and documents

$\boldsymbol{P}_{wd} = \boldsymbol{P}_{w|d}\,\mathrm{diag}(\boldsymbol{p}_d)$

0	$^1/_8$	$^1/_{20}$	$^1/_{16}$
$^1/_{12}$	$^1/_8$	$^1/_{20}$	$^1/_{16}$
$^1/_{12}$	0	$^1/_{20}$	0
0	0	$^1/_{20}$	$^1/_{16}$
$^1/_{12}$	0	$^1/_{20}$	$^1/_{16}$

marginal probabilities

$\boldsymbol{p}_w = \boldsymbol{P}_{wd}\,\boldsymbol{1}$ $\quad$ $\boldsymbol{p}_d = \boldsymbol{P}_{wd}^{\mathrm{T}}\,\boldsymbol{1}$

$$p_{w_1,w_2} = \sum_d p_{w_1|d,\cancel{w_2}}\, p_{w_2,d}$$

$$\boxed{\boldsymbol{P}_{ww} = \boldsymbol{P}_{wd}\,\mathrm{diag}(\boldsymbol{p}_d)^{-1}\,\boldsymbol{P}_{wd}^{\mathrm{T}}}$$

$\boldsymbol{p}_w = \boldsymbol{P}_{ww}\,\boldsymbol{1}$

projection on words

projection on documents

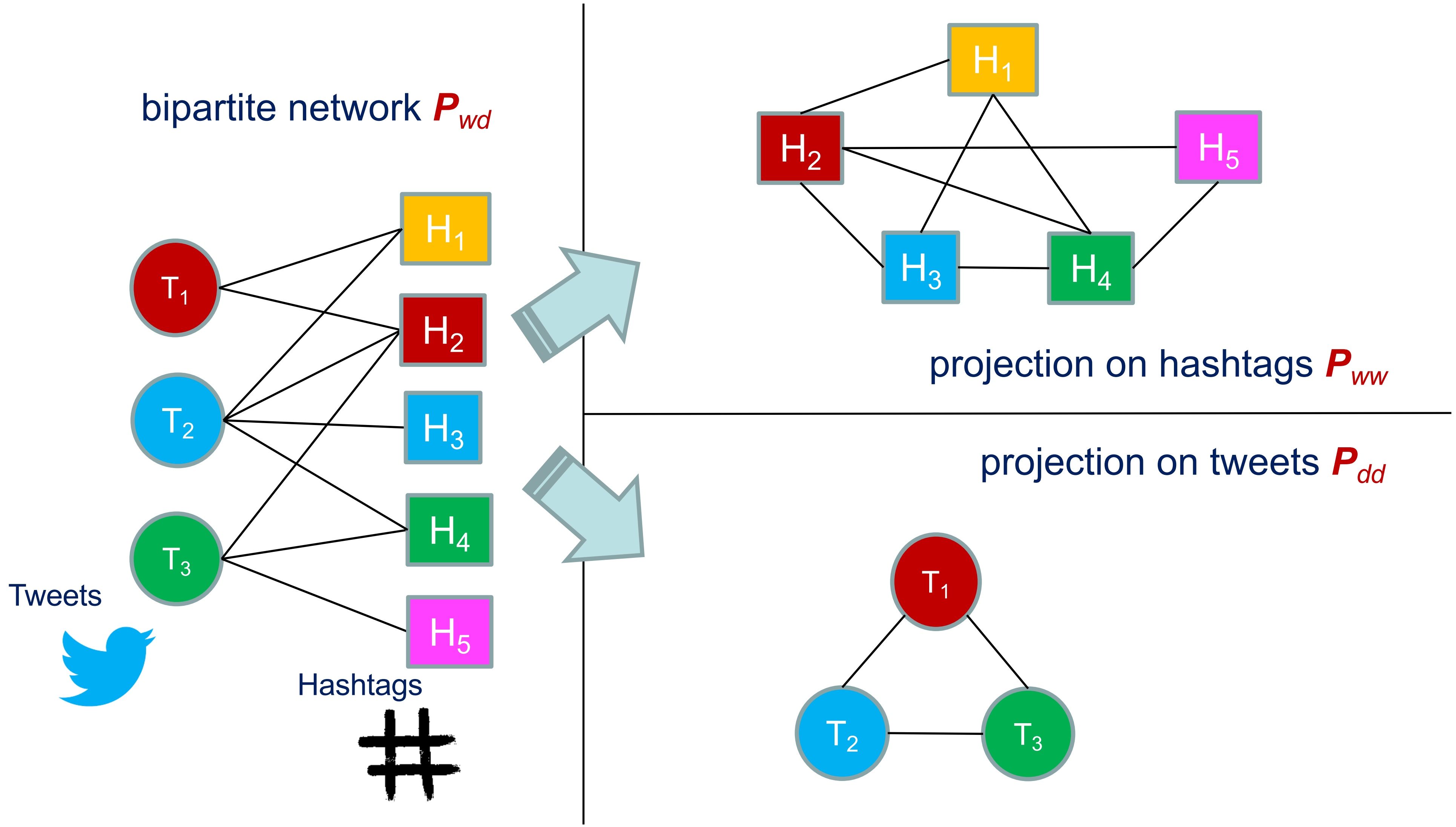
$$p_{d_1,d_2} = \sum_w p_{d_1|w,\cancel{d_2}}\, p_{d_2,w}$$

$$\boxed{\boldsymbol{P}_{dd} = \boldsymbol{P}_{wd}^{\mathrm{T}}\,\mathrm{diag}(\boldsymbol{p}_w)^{-1}\,\boldsymbol{P}_{wd}}$$

$\boldsymbol{p}_d = \boldsymbol{P}_{dd}\,\boldsymbol{1}$

Bipartite and projected networks

a comparison



The role of TF-IDF

term frequency – inverse document frequency

term frequency = frequency (probability) of the word in the document

inverse document frequency = (log) fraction of documents that contain the word

$$\text{TF-IDF}_{w|d} = p_{w|d} \cdot -\log\left(\frac{\sum_d (nwd > 0)}{D}\right)$$

- An heuristic
- Punishes words that appear in many documents
- Enhances words that are document specific

$$\boldsymbol{T}_{wd} = \textbf{TF-IDF}_{w|d}\ \mathrm{diag}(\textbf{tf-idf}_d)^{-1} \cdot \mathrm{diag}(\boldsymbol{p}_d)$$

→ $\boldsymbol{T}_{ww}$, $\boldsymbol{T}_{dd}$

to be used in place of $\boldsymbol{P}_{wd}$

normalization vector $\textbf{tf-idf}_d = \textbf{TF-IDF}_{w|d}{}^T\, \boldsymbol{1}$ (to guarantee that columns sum up to 1)

Topic detection

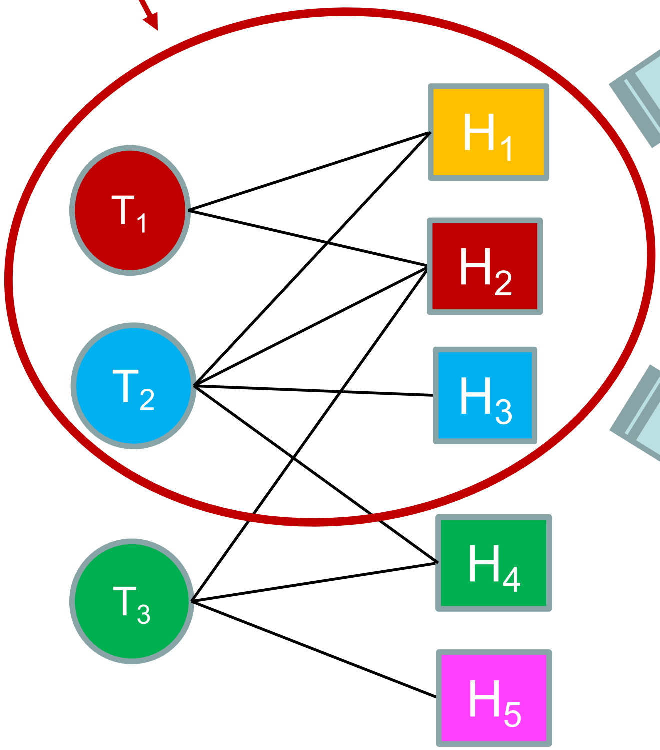
i.e., community detection in semantic networks

Topic detection

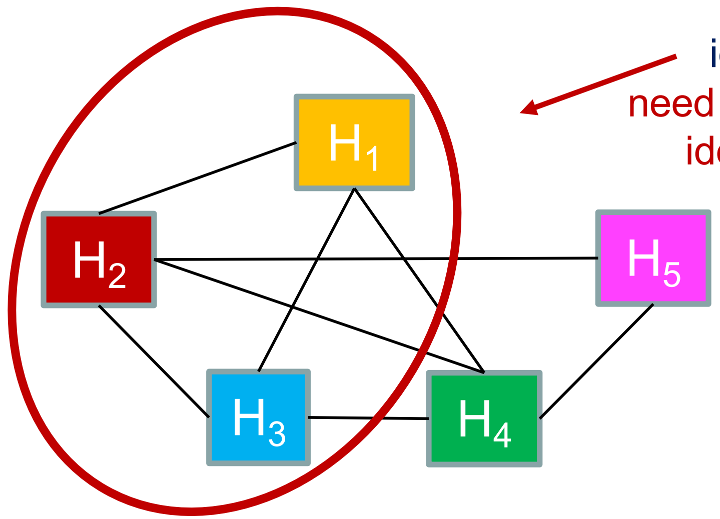
in bipartite and projection networks

bipartite network P_{wd} or T_{wd}

a community identifies both documents and words

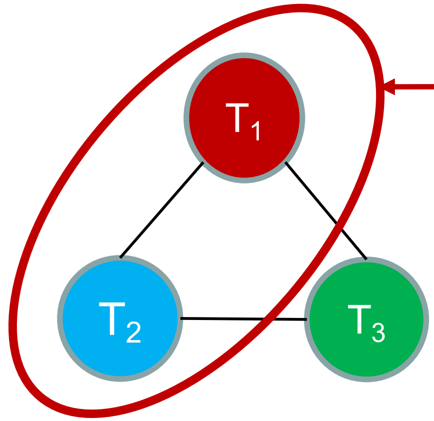


a community identifies words
need to use TopicRank to identify documents



projection on words P_{ww} or T_{ww}

projection on documents P_{dd} or T_{dd}

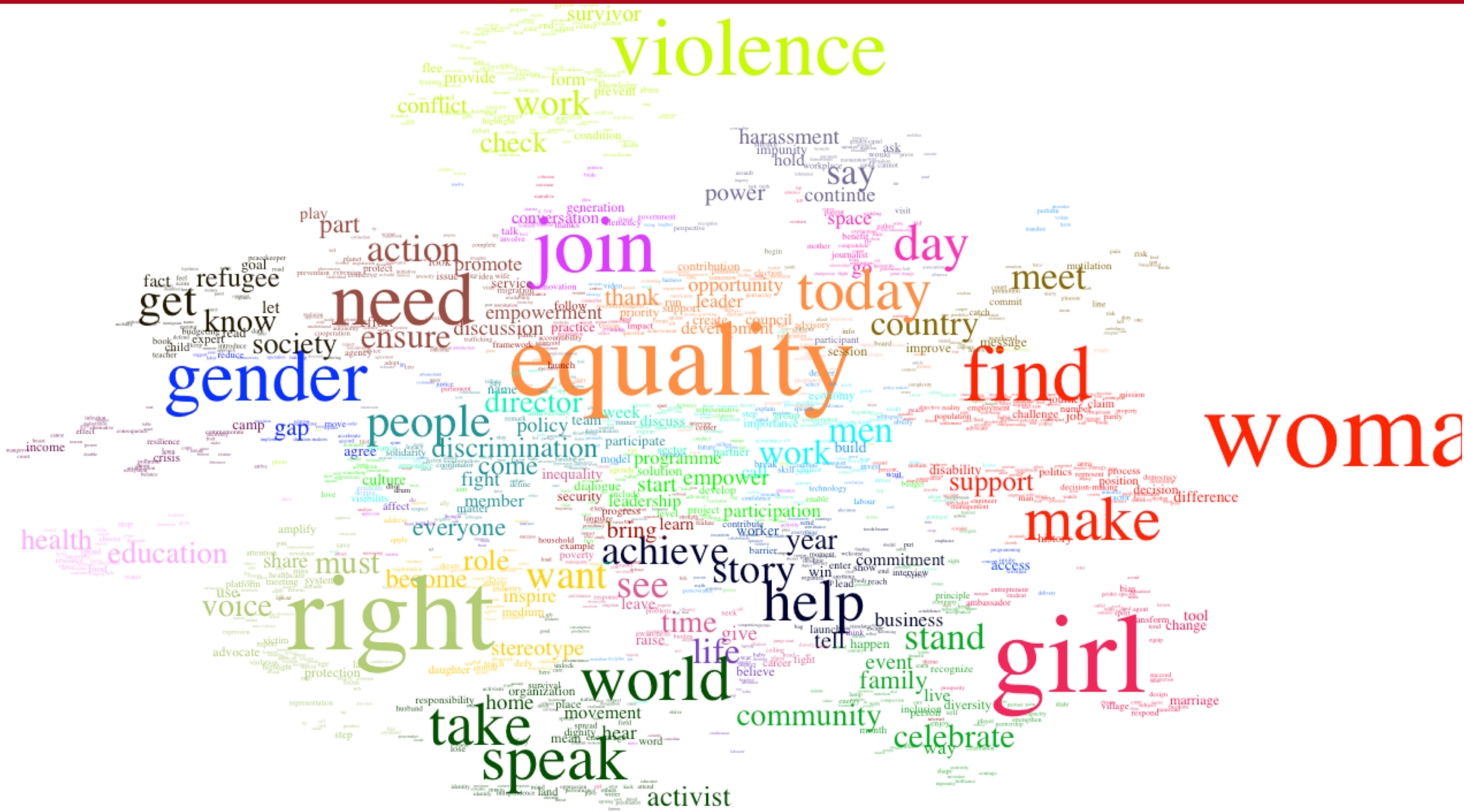


a community identifies documents
need to use TopicRank to identify words

UNIVERSITÀ DEGLI STUDI DI PADOVA

Projection on words

an example

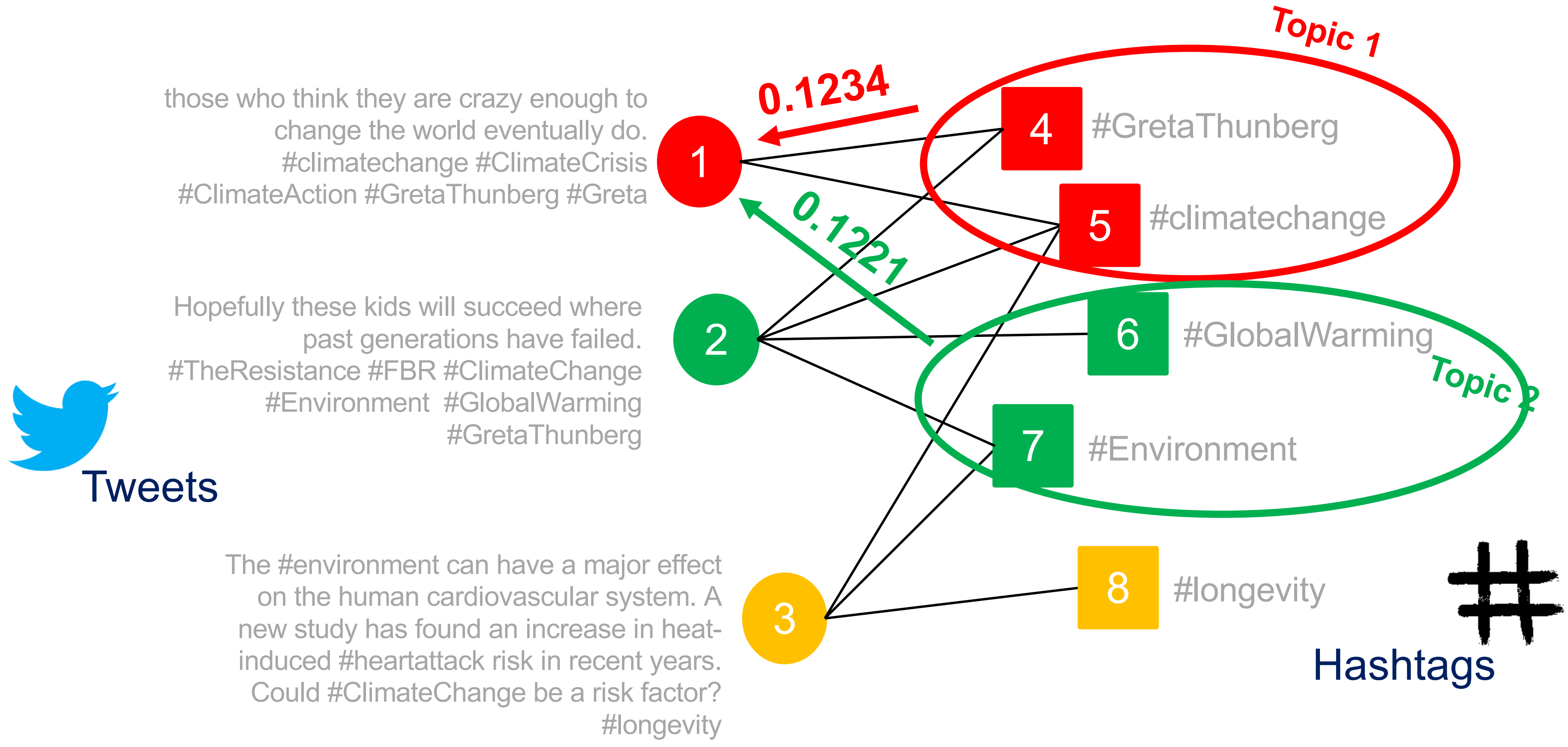


#metoo tweets

Projection on words

assigning documents to topics via TopicSpecific PageRank

Tweet 1 is assigned to **Topic 1** !!!



Normalized mutual information

a wrap-up in topic detection

statistical dependencies about words and topics

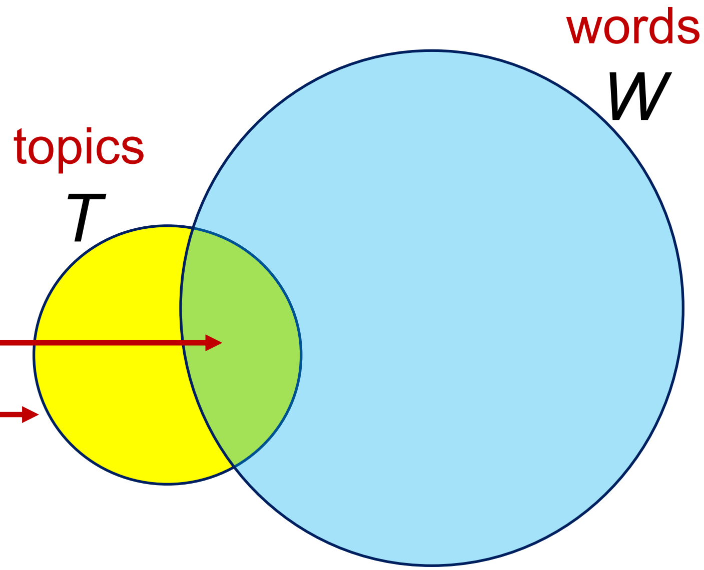
$$\boldsymbol{P}_{\mathrm{wt}} = \boldsymbol{P}_{wd}\, \boldsymbol{C}^{\mathrm{T}}$$

probability of a topic

$$\boldsymbol{p}_{\mathrm{t}} = \boldsymbol{P}_{wt}^{\ \mathrm{T}}\, \boldsymbol{1}$$

fraction of knowledge related to the topic that is explained by words (equal to 1 if topics use different words)

$$\mathrm{NMI} = \frac{I(W;T)}{H(T)}$$



Modularity and normalized cut

a wrap-up in topic detection

C topic assignment to be assessed for quality

$$\boldsymbol{P}_{tt} = \boldsymbol{C}\,\boldsymbol{P}_{dd}\,\boldsymbol{C}^T$$

can be interpreted as a probability matrix linking topics, its entries are the sum of the links of **A** from topic i to topic j

P_{11}	P_{12}	P_{13}
P_{21}	P_{22}	P_{23}
P_{31}	P_{32}	P_{33}

$$\boldsymbol{p}_t = \boldsymbol{P}_{tt}\,\boldsymbol{1}$$

can be interpreted as the probability vector of topics

modularity

$$\boxed{Q = \sum_t \left(P_{tt} - p_t^2\right) < 1}$$

to be maximized

normalized cut

normalized version

$$\boxed{\mathrm{Ncut} = 1 - \frac{\sum_t P_{tt}/pt}{\sum_t 1} > 0}$$

to be minimized

PageRank vector (ranking of documents)

$\boldsymbol{r} = (1\text{-}c)\, \boldsymbol{P}_{d|d}\, \boldsymbol{r} + c\, \boldsymbol{1}/N$

$\boldsymbol{P}_{d|d} = \boldsymbol{P}_{dd}\,\text{diag}^{-1}(\boldsymbol{p}_d)$

Here $\boldsymbol{c}_i$ is the ith row of $\boldsymbol{C}$

$$q_i = \left(1 - (1-c)\frac{\boldsymbol{c}_i \boldsymbol{1}}{N}\right) \boldsymbol{z}_i \boldsymbol{1} - c\, \boldsymbol{c}_i \boldsymbol{P}_{d|d} \boldsymbol{z}_i^T$$

$$\boldsymbol{z}_i = \boldsymbol{c}_i\, \text{diag}(\boldsymbol{r})$$

$$\boxed{\text{InfoMap} = f(\boldsymbol{q}) + \sum_i f([q_i, \boldsymbol{z}_i])}$$

entropy function

$$f(\boldsymbol{x}) = -\sum_i x_i \log\left(\frac{x_i}{\sum_j x_j}\right)$$

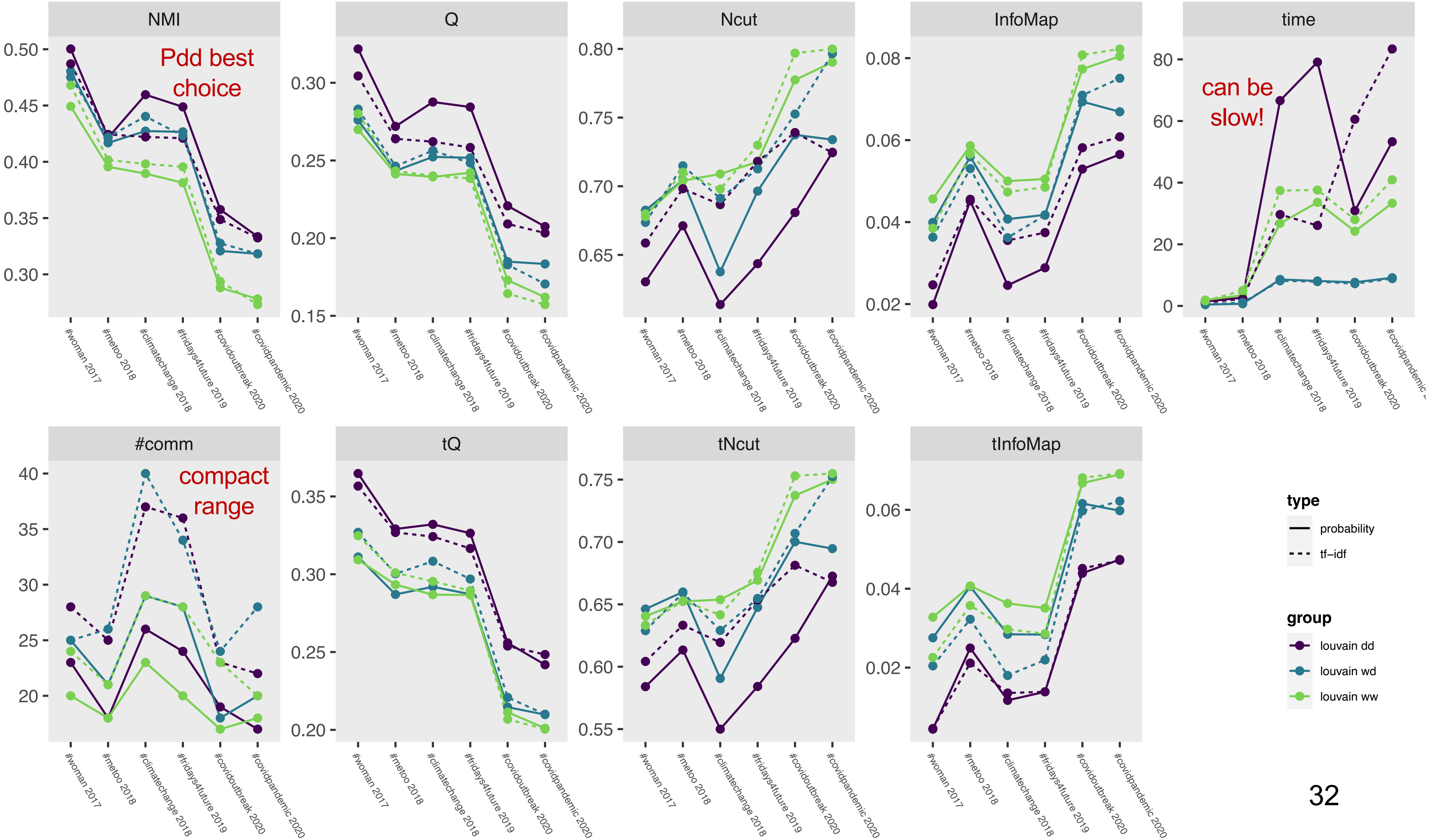
normalized version

$$\boxed{\frac{\text{InfoMap}}{f(\boldsymbol{r})} - 1}$$

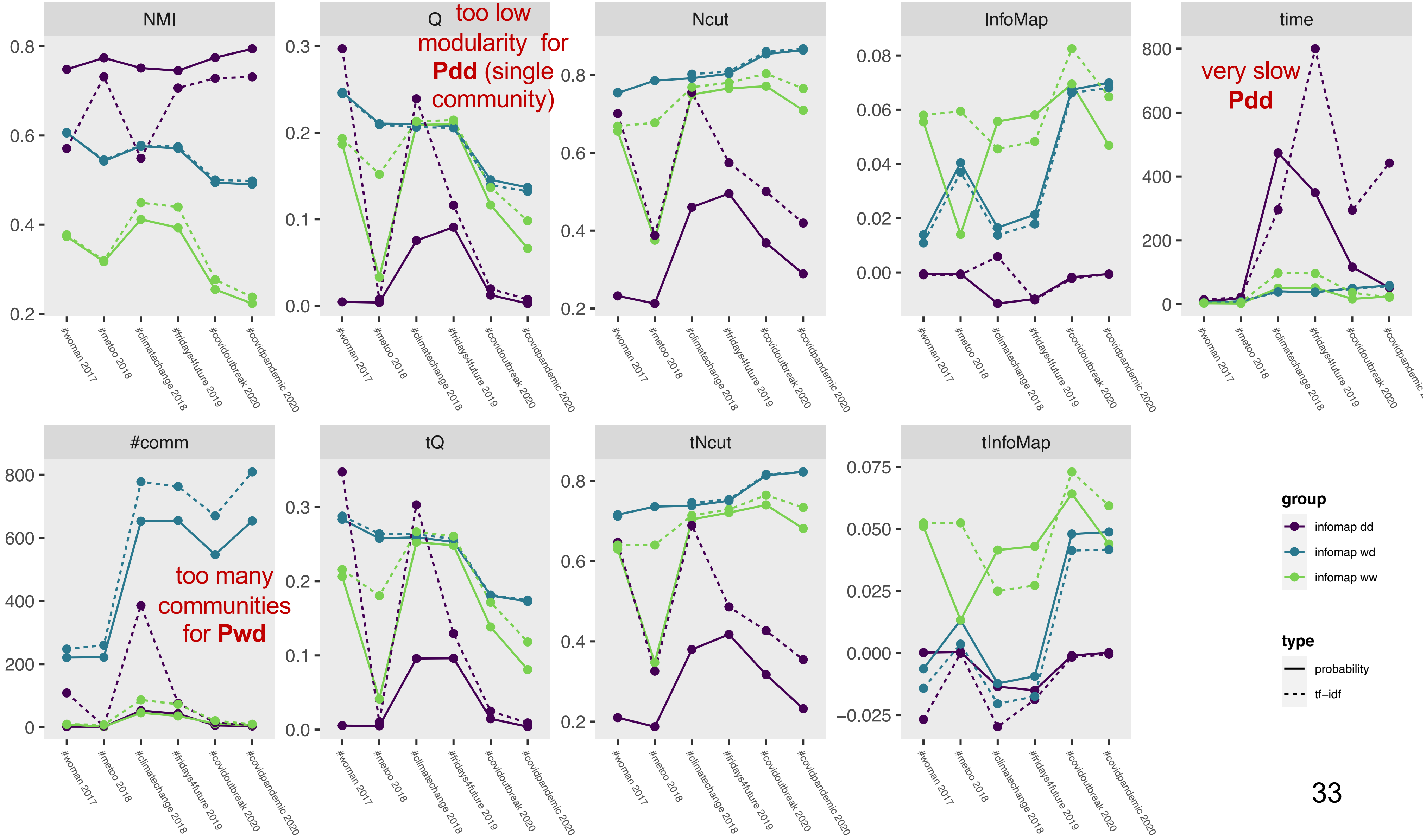
to be minimized

Università degli Studi di Padova

A comparison
of the different approaches - Louvain



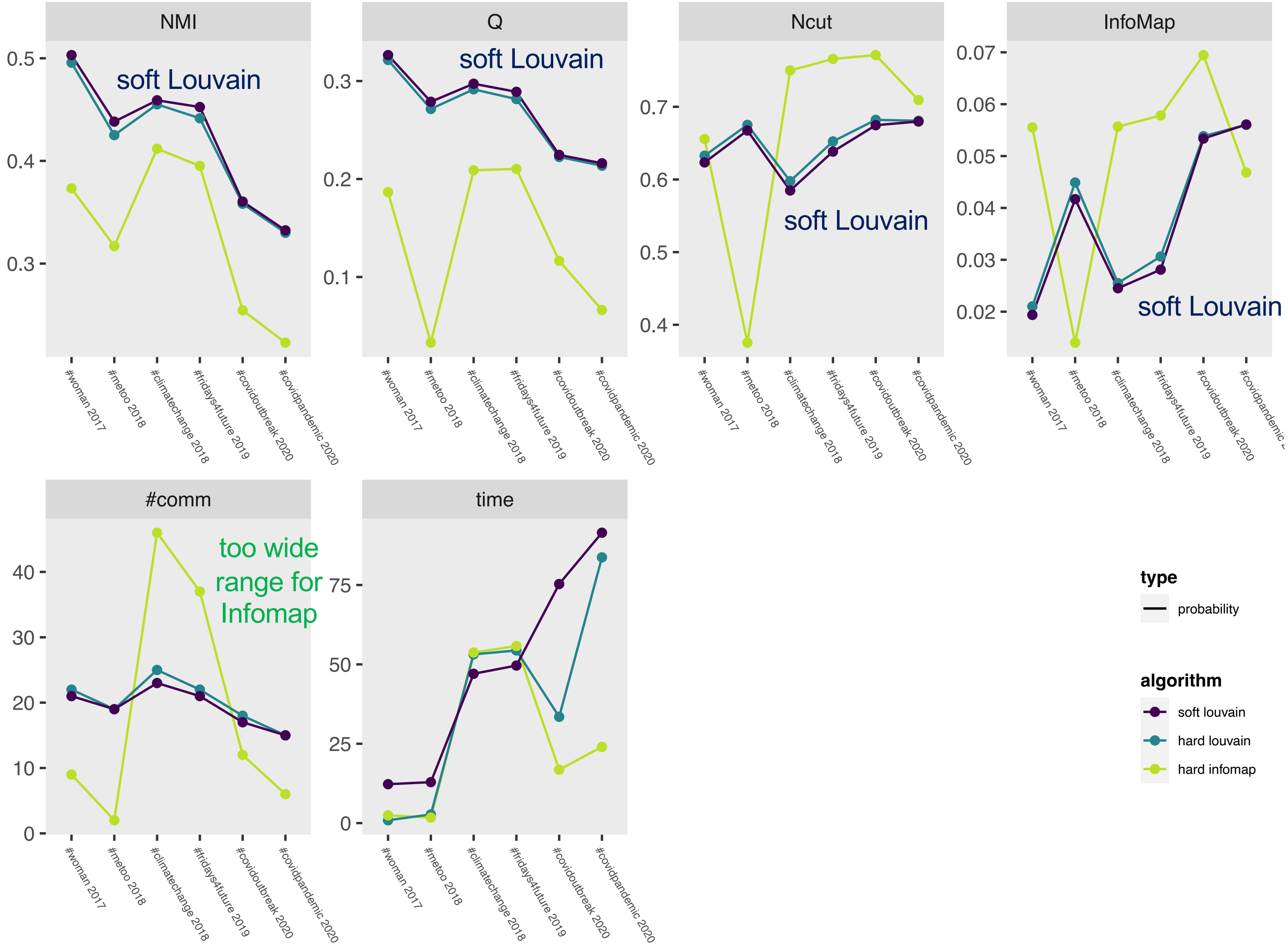
A comparison
of the different approaches - Infomap



UNIVERSITÀ DEGLI STUDI DI PADOVA

A comparison

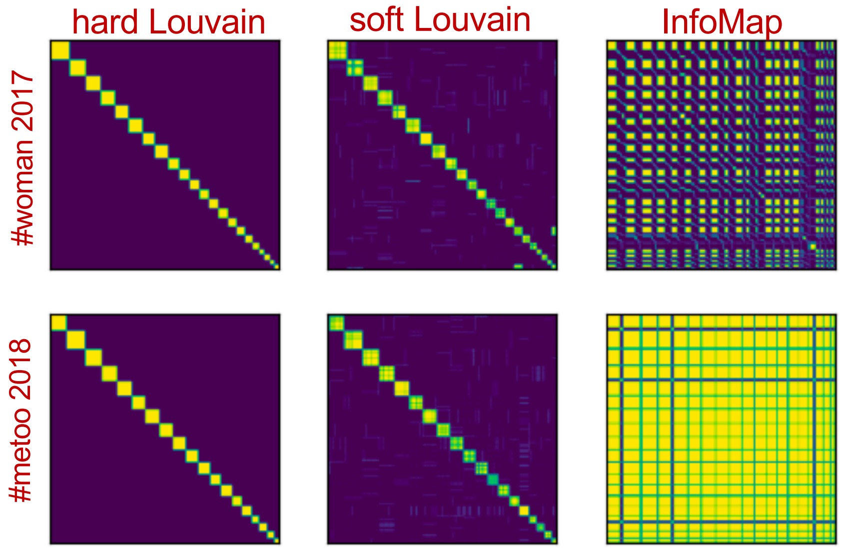
hard/soft Louvain Pdd versus InfoMap Pww

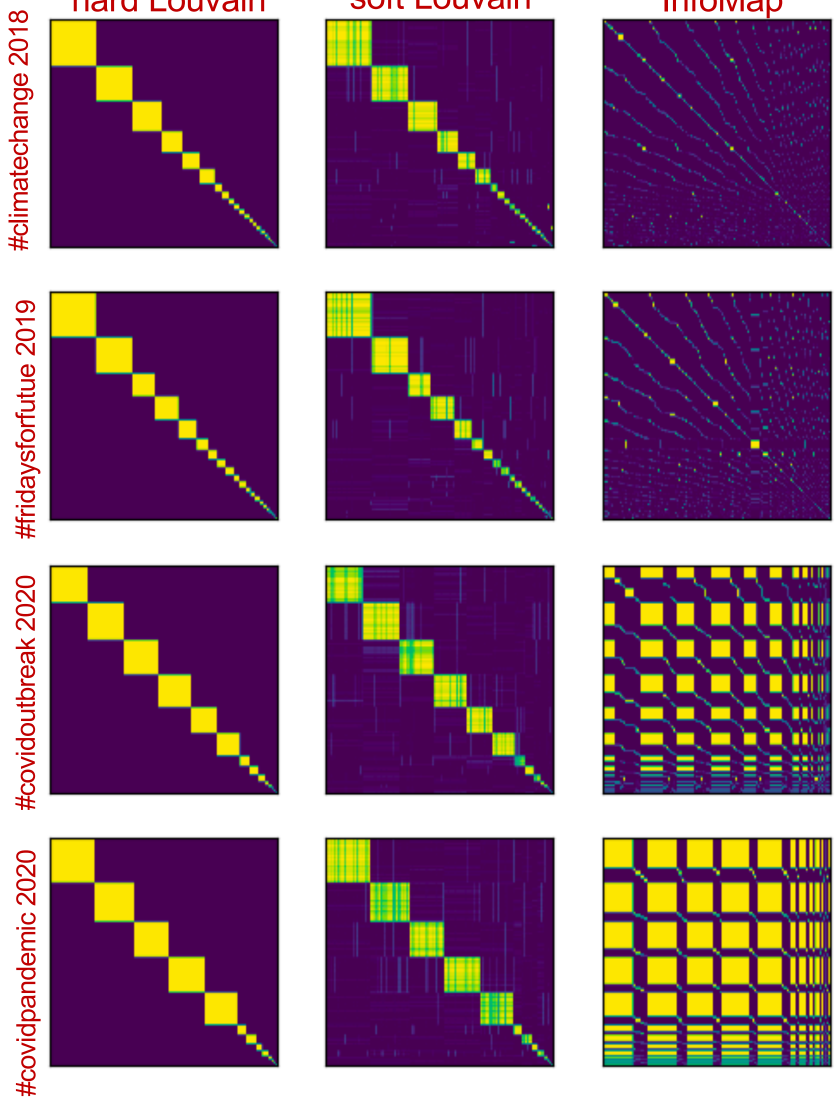


Università degli Studi di Padova

A comparison

a $C^T C$ pattern overview





InfoMap has either very big communities or not related to Louvain

A comparison

takeaways

- Louvain Pdd – provides the best results

  produces balanced clusters

- Louvain soft – slightly strengthens the result

- Bipartite networks – run much faster

  but performance deteriorates

- InfoMap – not robust ☹

  would be nice to see BigCLAM and SBMs

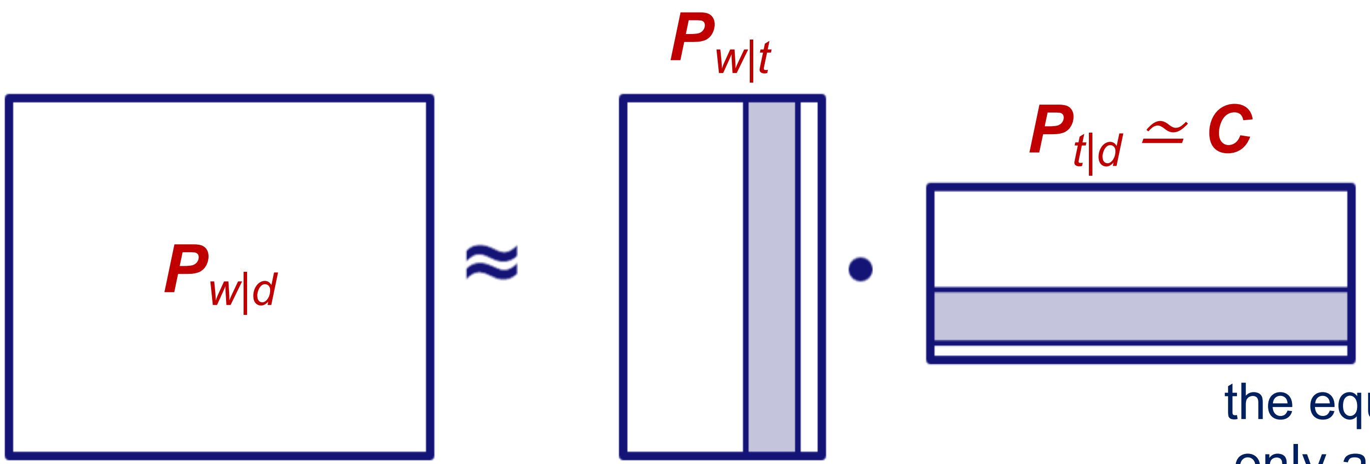
  … your task! ☺

Non-negative Matrix Factorization

and its application to topic detection

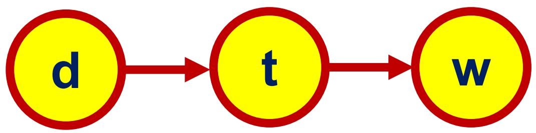
NMF = nonnegative matrix factorization

rationale



the equivalence is only approximate

underlying model



each document is associated to a topic distribution (one of the colums of C)

each topic is associated to a word distribution (one of the columns of $P_{w|t}$)

NMF optimization

Frobenius norm and generalized Kullbak-Leibler divergence

A = ***P***$_{w|d}$ is column stochastic

$$\mathrm{argmin}_{\boldsymbol{W}\geq\boldsymbol{0},\boldsymbol{H}\geq\boldsymbol{0}} \sum_{ij} |A_{ij} - [\boldsymbol{WH}]_{ij}|^2$$

minimizing the Frobenius norm does not ensure a column stochastic product ***W H***

$$\mathrm{argmin}_{\boldsymbol{W}\geq\boldsymbol{0},\boldsymbol{H}\geq\boldsymbol{0}} \sum_{ij} A_{ij} \log\left(\frac{A_{ij}}{[\boldsymbol{WH}]_{ij}}\right) - A_{ij} + [\boldsymbol{WH}]_{ij}$$

$$f(y) = x \log\left(\frac{x}{y}\right) - x + y$$

$$f'(y) = -\frac{x}{y} + 1 = 0 \rightarrow \mathrm{y} = x$$

minimizing the generalized Kullback-Leibler divergence ensures a column stochastic product ***W H***

Ho & Van Dooren. "Non-negative matrix factorization with fixed row and column sums." (2008)

NMF in Python

sklearn package

```
from sklearn.decomposition import NMF
Pwgd = Pwd/Pwd.sum(axis=0).flatten()
```

wisely initialize for best performance

run on different number of topics, then choose the best fit, e.g., according to modularity

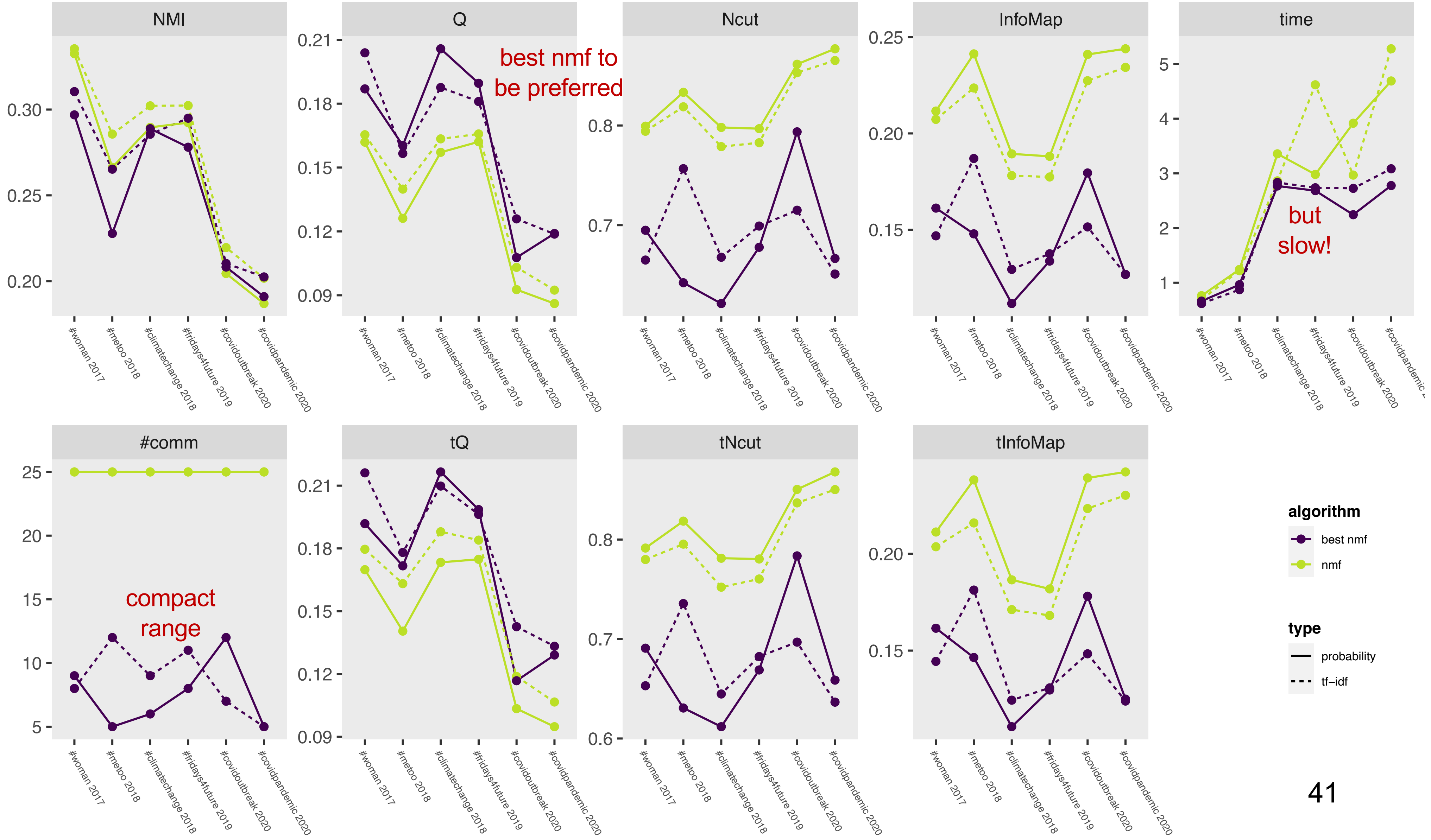
choose generalized Kullback-Leibler divergence, and the related solver

```
# fit nmf model X = W*H
model = NMF(n_components=i, init='nndsvd',
            solver='mu', beta_loss='kullback-leibler')
W = model.fit_transform(Pwgd)
H = sps.csr_matrix(model.components_)
# column normalized versions
H = sps.diags(W.sum(axis=0).flatten())*H # Ptgd
W = W/W.sum(axis=0).flatten() # Pwgt
# community assignment C
C = sps.csr_matrix(np.transpose(H/H.sum(axis=0).flatten()))
```

need to make W column stochastic, to have H column stochastic too

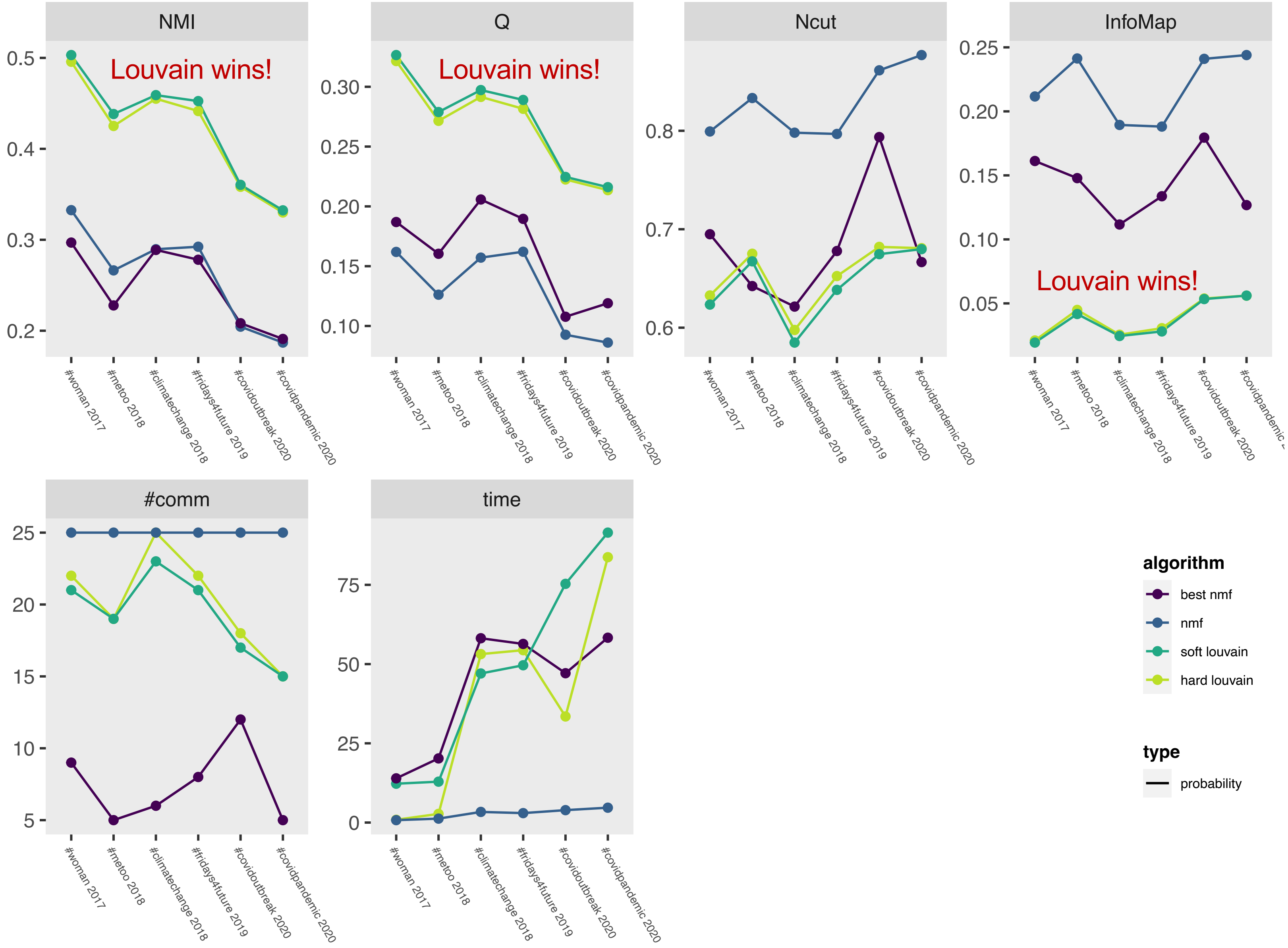
force column stochasticity in H (not needed though)

A comparison
of the different approaches



UNIVERSITÀ DEGLI STUDI DI PADOVA

A comparison
with Louvain



Takeaways
on NMF

- Naturally provides a soft topic assignment
- NMF – not strikingly good

  probably due to the fact that we want to express
  a sparse matrix through an eigenvector-like product
  with few eigenvectors (the fit is far from ideal)
- Comparison – with Louvain

  much weaker
- Complexity – generally slow

  need to test it for different numbers of topics ☹
  fast for fixed topic number

Latent Dirichlet allocation

LDA = a stochastic model for topic detection

LDA model

Blei,, Ng, Jordan. "Latent dirichlet allocation." (2003)
https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=https://githubhelp.com

document

topics vector
length N_d

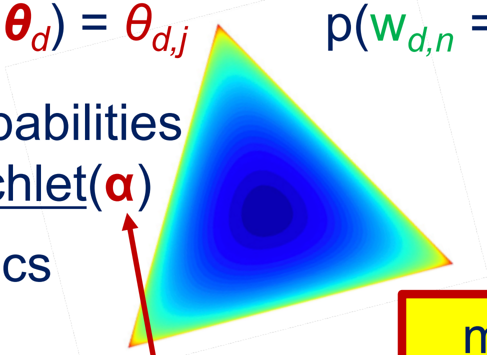
words vector
length N_d

$d \longrightarrow \mathbf{t}_d \longrightarrow \mathbf{w}_d$

topics distribution
$p(t_{d,n} = j \mid \boldsymbol{\theta}_d) = \theta_{d,j}$

words distribution
$p(w_{d,n} = i \mid \boldsymbol{\beta}, t_{d,n} = j) = \boldsymbol{\beta}_{i,j}$

topics probabilities
$\boldsymbol{\theta}_d \sim$ Dirichlet($\boldsymbol{\alpha}$)



K topics

$\boldsymbol{\theta}_d$ is the topic assignment $\boldsymbol{C} = \boldsymbol{P}_{t|d}$

the number of topics K is fixed

vector $\boldsymbol{\alpha}$, to be estimated, is the topic probability $\boldsymbol{p}_t = \boldsymbol{\alpha}/|\boldsymbol{\alpha}|_1$

matrix $\boldsymbol{\beta}$, to be estimated, is the probability of words given a topic, $\boldsymbol{P}_{w|t} = \boldsymbol{\beta}$

LDA optimization

can be solved using variational inference = suboptimum approach

topics assignment probability (Dirichlet)

$$p(\boldsymbol{\theta}_d \mid \boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\sum_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} [\theta_{d,k}]^{\alpha_k - 1}$$

words probability

$$p(\boldsymbol{w}_d \mid \boldsymbol{\beta}, \boldsymbol{\theta}_d) = \prod_{n=1}^{N_d} [\boldsymbol{\beta}\, \boldsymbol{\theta}_d]_{w_{d,n}}$$

this dependence between $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ is the trickiest part

overall probability

$$p(\text{corpus} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_d \int p(\boldsymbol{w}_d \mid \boldsymbol{\beta}, \boldsymbol{\theta}_d)\, p(\boldsymbol{\theta}_d \mid \boldsymbol{\alpha})\, d\boldsymbol{\theta}_d$$

target optimization

$$\boxed{\operatorname{argmax}_{\boldsymbol{\alpha},\boldsymbol{\beta}}\, p(\text{corpus} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})} \Rightarrow$$

$$\boxed{C = P_{t|d} = \theta}$$

$$\boxed{P_{wt} = \beta \operatorname{diag}(\alpha / |\alpha|_1)}$$

this is what we get

LDA in Python

sklearn package

```
from sklearn.decomposition import LatentDirichletAllocation
```

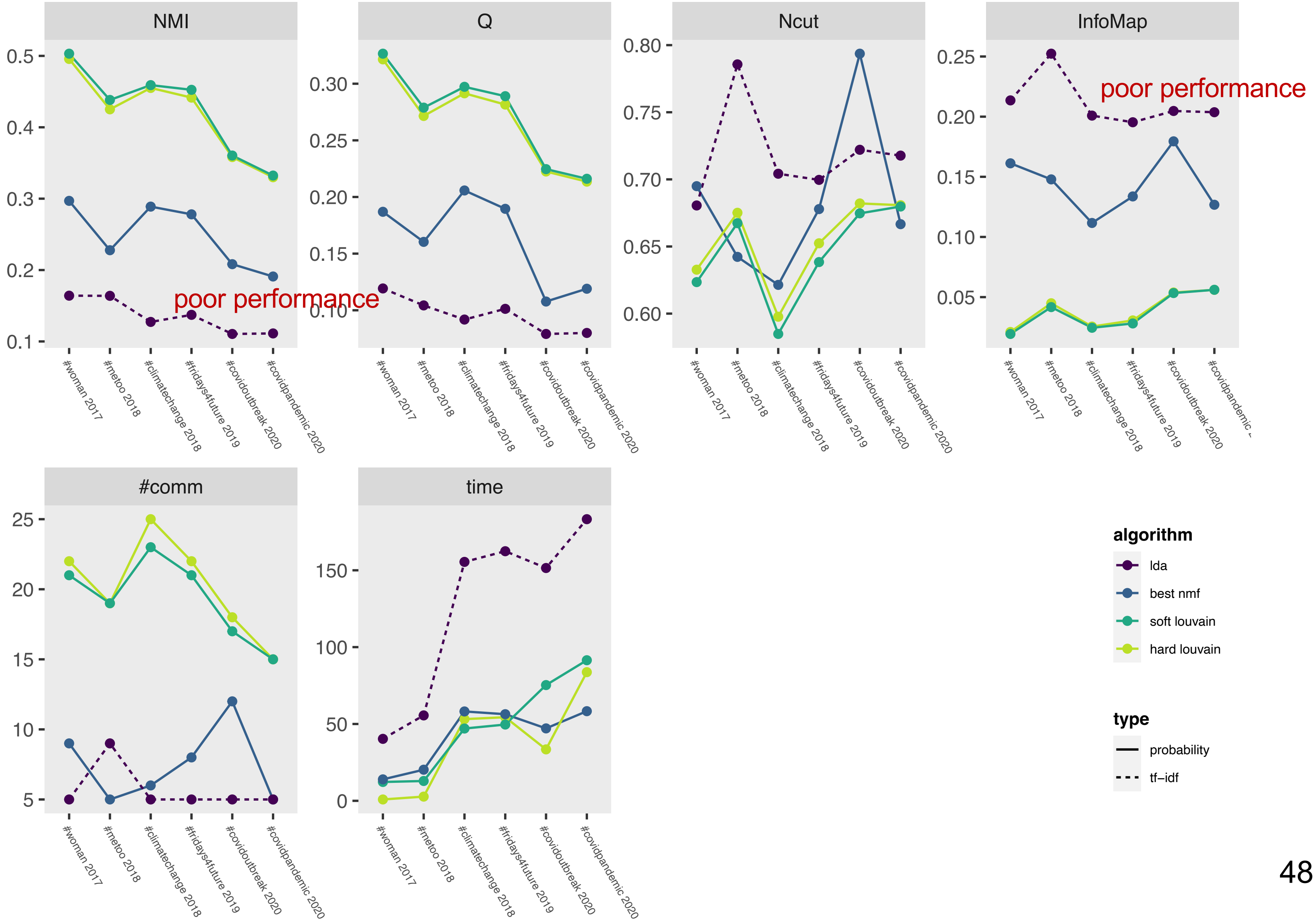
```
# fit lda model
lda = LatentDirichletAllocation(n_components=i,
                                learning_method="batch")
lda.fit(Mwd.T)
# community assignment C = Ptgd'
C = sps.csr_matrix(lda.transform(Mwd.T))
```

initialise and fit model

LDA

extract topic assignment

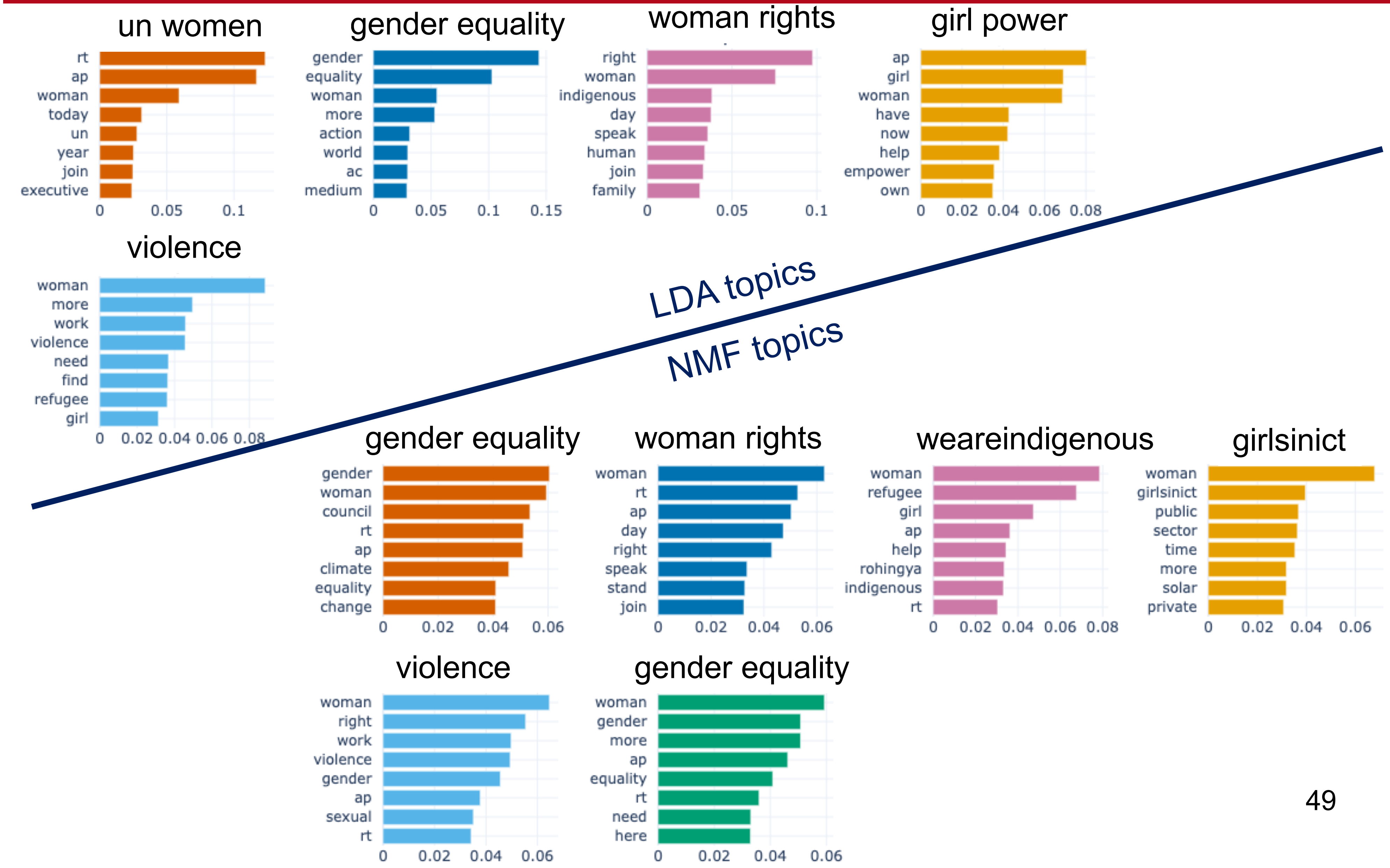
A comparison
with NMF and Louvain



Università degli Studi di Padova

A comparison

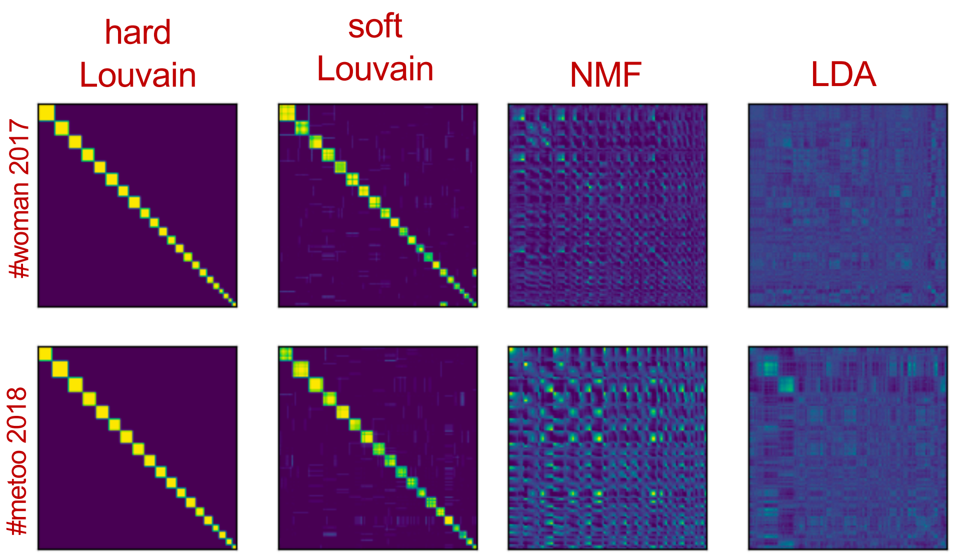
NMF versus LDA topics

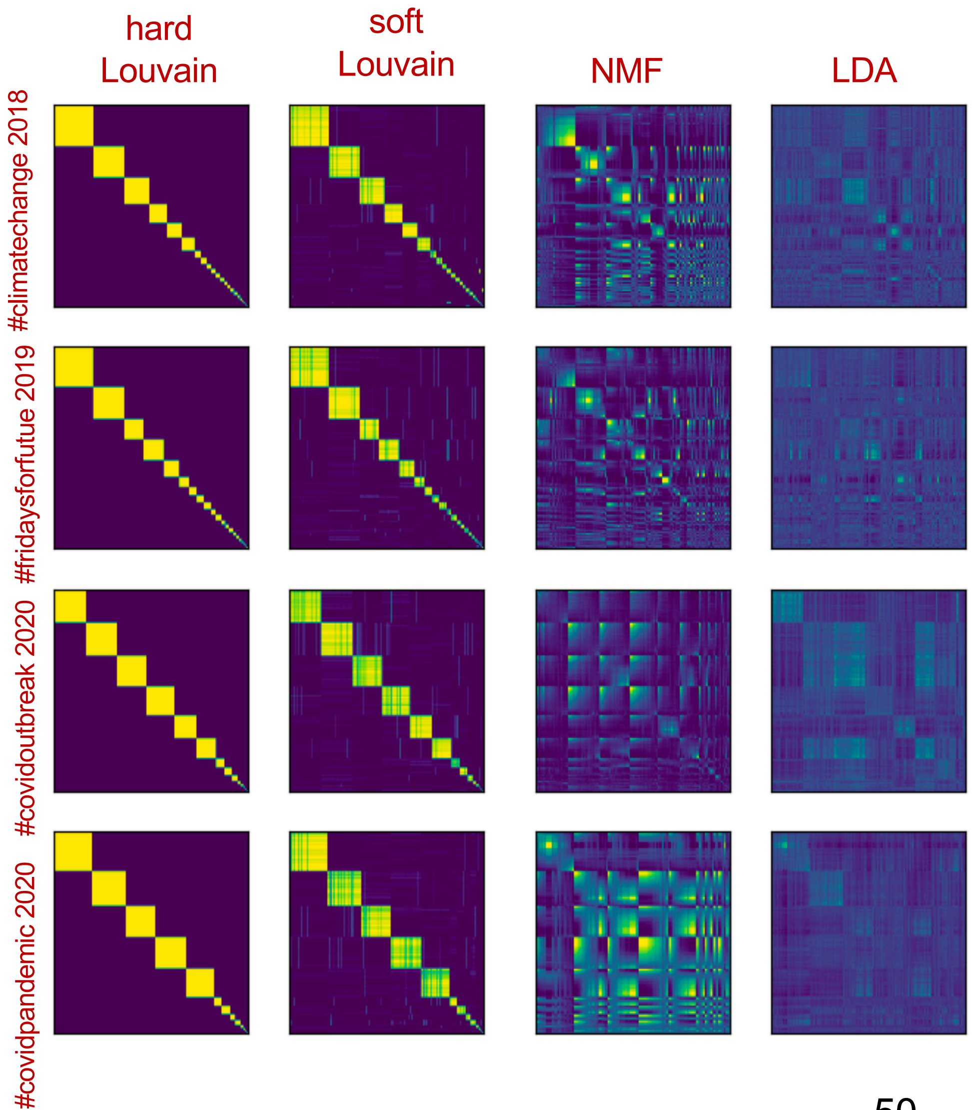


Università degli Studi di Padova

A comparison

a C^TC pattern overview





NMF and LDA identify few communities hardly related to Louvain

Takeaways
on LDA

- Naturally provides a soft topic assignment
- LDA – not strikingly good
  - same eigenvector-like product as NMF
  - worse than NMF … known issue ☹
  - probably due to the Dirichlet assumption (questionable)
  - and the variational inference (suboptimum approach)
- Comparison – with Louvain
  - much weaker
- Complexity – generally slow
  - need to test it for different numbers of topics ☹
  - fast for fixed topic number

Variational Auto Encoders

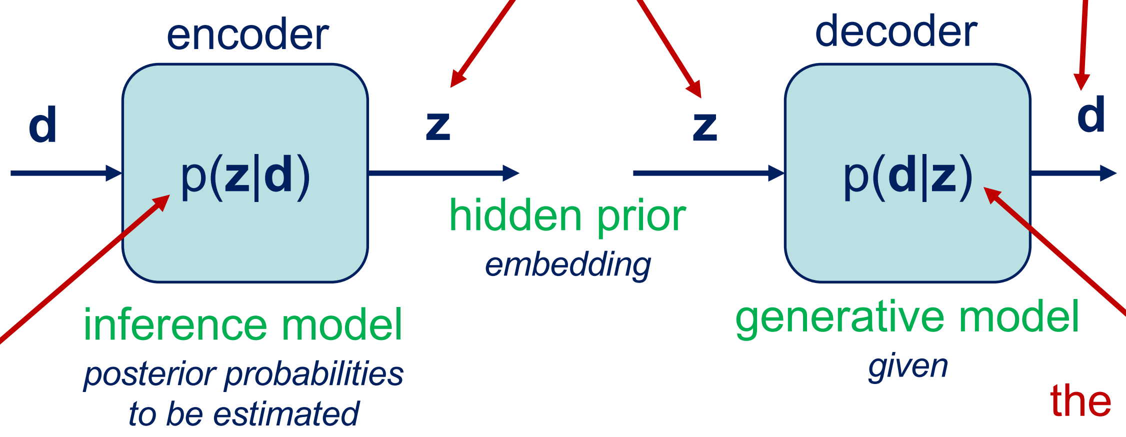
an application to topic analysis

Variational Auto-Encoders

Kingma, Welling, "Auto-encoding variational Bayes," (2013)

the hidden prior in our case is the topic (distribution)

the output in our case is the document (collection of words)



the (stochastic) model explains how a document is generated from a topic (distribution)

but we are interested in the inverse link that, given a document tells what topic it is associated with

$$p(\boldsymbol{z}|\boldsymbol{d}) = \frac{p(\boldsymbol{d}|\boldsymbol{z})\,p(\boldsymbol{z})}{p(\boldsymbol{d})} \cong q(\boldsymbol{z}|\boldsymbol{d})$$

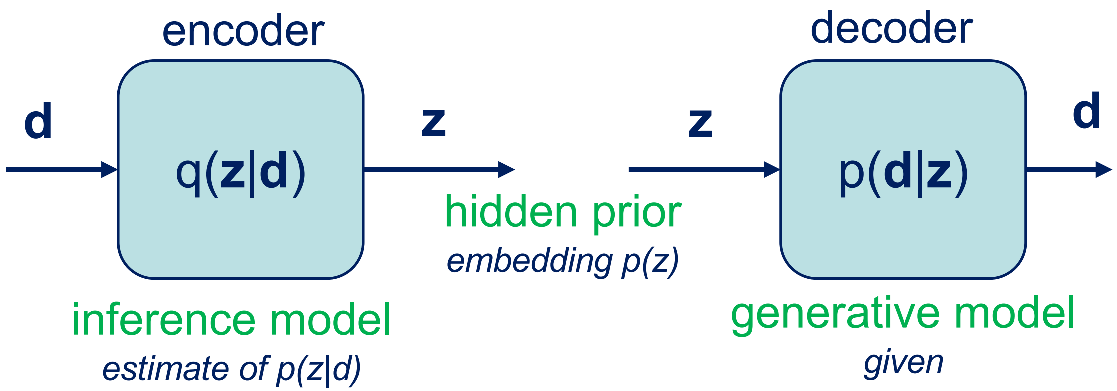
impossible to know in the closed form

needs an a-priori model for the embedding

is approximated by a simple alternative model

VAE optimization rationale

ELBO = evidence lower bound



ELBO

$$\mathcal{L}_{\theta,\phi}(\boldsymbol{d}) \;\leq\; \log p_\theta(\boldsymbol{d})$$

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(\boldsymbol{d}) &= \log p_\theta(\boldsymbol{d}) - D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}|\boldsymbol{d}) \middle\| p_\theta(\boldsymbol{z}|\boldsymbol{d})\right) \\ &= \int dz\; q_\phi(\boldsymbol{z}|\boldsymbol{d}) \log\left(\frac{p_\theta(\boldsymbol{z},\boldsymbol{d})}{q_\phi(\boldsymbol{z}|\boldsymbol{d})}\right) \\ &= \underbrace{\int dz\; q_\phi(\boldsymbol{z}|\boldsymbol{d}) \log\left(p_\theta(\boldsymbol{d}|\boldsymbol{z})\right)}_{\mathcal{L}_1} + \underbrace{\int dz\; q_\phi(\boldsymbol{z}|\boldsymbol{d}) \log\left(\frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{d})}\right)}_{\mathcal{L}_2}\end{aligned}$$

to be maximized wrt parameters θ and ϕ

provides fitting on p(**z**), p(**d**|**z**), and q(**z**|**d**)

a-priori model (given)

inference model (approximate)

generative model (given)

L2 ELBO function

usually has a compact expression

$$\underbrace{\int dz\, q_\phi(\boldsymbol{z}|\boldsymbol{d}) \log\left(\frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{d})}\right)}_{\mathcal{L}_2}$$

both should have a simple parametrization on θ and ϕ

e.g., the Gaussian case

$$p_\theta(\boldsymbol{z}) = \frac{1}{\sqrt{\det\left(2\pi\, \mathrm{diag}\left(\boldsymbol{\sigma}_\theta^2\right)\right)}} \exp\left(-½(\boldsymbol{z}-\boldsymbol{\mu}_\theta)^T \mathrm{diag}^{-1}\left(\boldsymbol{\sigma}_\theta^2\right)(\boldsymbol{z}-\boldsymbol{\mu}_\theta)\right)$$

$$q_\phi(\boldsymbol{z}|\boldsymbol{d}) = \frac{1}{\sqrt{\det\left(2\pi\, \mathrm{diag}\left(\boldsymbol{\sigma}_\phi^2(\boldsymbol{d})\right)\right)}} \exp\left(-½\left(\boldsymbol{z}-\boldsymbol{\mu}_\phi(\boldsymbol{d})\right)^T \mathrm{diag}^{-1}\left(\boldsymbol{\sigma}_\phi^2(\boldsymbol{d})\right)\left(\boldsymbol{z}-\boldsymbol{\mu}_\phi(\boldsymbol{d})\right)\right)$$

$$\mathcal{L}_2(\theta,\phi) = ½ \sum_i 1 + \log\left(\frac{\sigma_{\phi,i}^2(\boldsymbol{d})}{\sigma_{\theta,i}^2}\right) - \frac{\sigma_{\phi,i}^2(\boldsymbol{d})}{\sigma_{\theta,i}^2} - \frac{\left(\mu_{\phi,i}(\boldsymbol{d}) - \mu_{\theta,i}\right)^2}{\sigma_{\theta,i}^2}$$

L1 ELBO function

approximated through Monte Carlo estimation

$$\underbrace{\int dz\, q_\phi(\boldsymbol{z}|\boldsymbol{d}) \log\left(p_\theta(\boldsymbol{d}|\boldsymbol{z})\right)}_{\mathcal{L}_1}$$

mostly too complex to be written in the closed form

solution: Monte Carlo approximation

$$\mathcal{L}_{\mathbf{1}}(\theta,\phi) = \frac{1}{L}\sum_{\ell=1}^{L} \log\left(p_\theta(\boldsymbol{d}|\boldsymbol{z}_\ell)\right)$$

samples generated according to the correct distribution

$$\boldsymbol{z}_\ell \sim q_\phi(\boldsymbol{z}|\boldsymbol{d})$$

e.g., the Gaussian case

$$\boldsymbol{z}_\ell = \boldsymbol{\mu}_\phi(\boldsymbol{d}) + \boldsymbol{\sigma}_\phi(\boldsymbol{d})\, \boldsymbol{n}_\ell$$

need to generate these once, then use them throughout the process

$$\boldsymbol{n}_\ell \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Take 1 - NVDM

Miao, Yu, Blunsom, "Neural variational inference for text processing," (2016)
http://proceedings.mlr.press/v48/miao16.pdf

here **W** builds a topic-to-word map that identifies the word distribution inside each document given the topic distribution

somehow related to a topic, but we do not know how

word probabilities

$\mathbf{z}$ → softmax($\mathbf{Wz+b}$) → $\mathbf{p}_w$ → $\mathbf{d}$

hidden prior

p($\mathbf{z}$) ***Gaussian*** *distributed*

$\boldsymbol{z} = \boldsymbol{\mu}_1 + \boldsymbol{\sigma}_1\, \boldsymbol{n}$

with ***n*** *normalized Gaussian*

document

$p_\theta(\mathbf{d}|\mathbf{z})$ ***multinomial***

distributed according to $\boldsymbol{p}_w(\boldsymbol{z})$

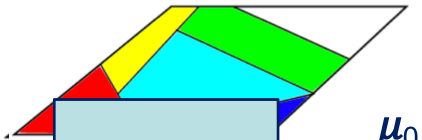
probability function for the multinomial

$\log(p_\theta(\mathbf{d}|\mathbf{z}))$

$= \mathbf{d}^T \log(\mathbf{p}_w(\mathbf{z})) + \text{const}$

d *one-hot-representation = number of occurrences of words in the document*

different document regions are associated with different output values



$\mathbf{d}$ → **FFN** → $\boldsymbol{\mu}_0$, $\log \boldsymbol{\sigma}_0^2$ → $\mathbf{z}$

document

one-hot-representation of appearing words

e.g., 2x 2-layer, ReLu activation, ***linear*** *output*

hidden prior

$q_\phi(\mathbf{z}|\mathbf{d})$ ***Gaussian*** *distributed*

$\boldsymbol{z} = \boldsymbol{\mu}_0 + \boldsymbol{\sigma}_0\, \boldsymbol{n}$

with ***n*** *normalized Gaussian*

NVDM ELBO

optimization target, to be maximized

one-hot-representation of a document = number of occurrences of words in the document

normalized Gaussian samples

decoder model

encoder model

$$\mathcal{L}(\theta,\phi) = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{m} \boldsymbol{d}_m^T \log\left(\underbrace{\mathrm{softmax}\big(\boldsymbol{b}+\boldsymbol{W}}_{\text{decoder map}}\big(\boldsymbol{\mu}_0(\boldsymbol{d}_m)+\boldsymbol{\sigma}_0(\boldsymbol{d}_m)\,\boldsymbol{n}_{m,\ell}\big)\big)\right)$$

$$+\,½\sum_{m}\sum_{i} 1+\log\left(\frac{\sigma_{0,i}^2(\boldsymbol{d}_m)}{\sigma_{1,i}^2}\right)-\frac{\sigma_{0,i}^2(\boldsymbol{d}_m)}{\sigma_{1,i}^2}-\frac{\big(\mu_{0,i}(\boldsymbol{d}_m)-\mu_{1,i}\big)^2}{\sigma_{1,i}^2}$$

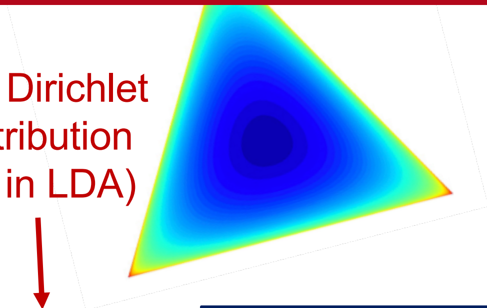
a-priori model

Not very clear where the topic is, though!

UNIVERSITÀ DEGLI STUDI DI PADOVA

Take 2 - DirVAE

Joo, Li, Park, Moon, "Dirichlet variational autoencoder," (2020)
https://www.sciencedirect.com/science/article/pii/S003132032030317



a multigamma distribution is introduced

$\mathbf{z}$/sum($\mathbf{z}$)

true Dirichlet distribution (as in LDA)

z → softmax(log()) → **t** → softmax(**Wt**+**b**) → ($\mathbf{p}_w$) → **d**

hidden prior
p(**z**) ***multigamma*** *distributed with parameters* $\boldsymbol{\alpha}_1$ and β=1

topic
Dirichlet *distributed*

document
p_θ(**d**|**z**) ***multinomial*** *distributed*

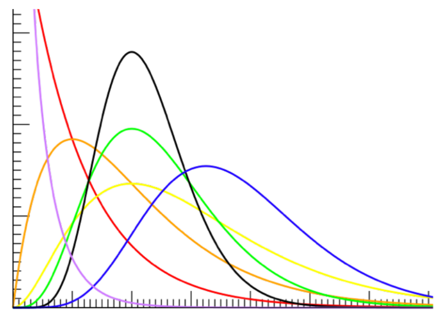
shape vector

d → **FFN** → ($\boldsymbol{\alpha}_0$>0) → **z**

document

e.g., 2x 2-layer, ReLu activation, ***softplus*** *ln(1+e^x) output*

hidden prior
q_ϕ(**z**|**d**) ***multigamma*** *distributed with parameters* $\boldsymbol{\alpha}_0$ and β=1



DirVAE ELBO

optimization target, to be maximized

$$\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\alpha}) = (\boldsymbol{u}\, \boldsymbol{\alpha}\, \Gamma(\boldsymbol{\alpha}_1))^{1/\boldsymbol{\alpha}}$$

approx. uniform to multigamma map

normalized uniform samples

one-hot-representation of a document = number of occurrences of words in the document

decoder model

encoder model

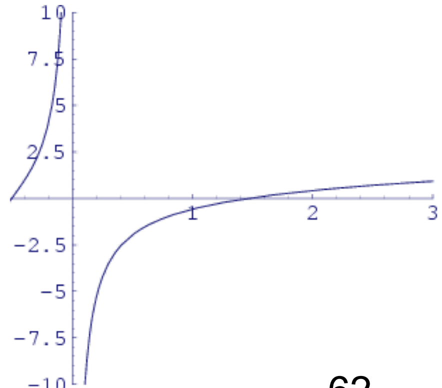
$$\mathcal{L}\,(\theta, \phi) = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{m} \boldsymbol{d}_m^T \log\Bigg(\underbrace{\text{softmax}\Big(\boldsymbol{b} + \boldsymbol{W}\,\text{softmaxlog}}_{\text{decoder map}}\Big(\boldsymbol{f}\big(\boldsymbol{u}_{m,\ell}, \boldsymbol{\alpha}_0(\boldsymbol{d}_m)\big)\Big)\Big)\Bigg)$$

$$+ \sum_{m}\sum_{i} \log\left(\frac{\Gamma(\alpha_{0,i}(\boldsymbol{d}_m))}{\Gamma(\alpha_{1,i})}\right) - \left(\alpha_{0,i}(\boldsymbol{d}_m) - \alpha_{1,i}\right)\psi(\alpha_{0,i}(\boldsymbol{d}_m))$$

a-priori model

digamma function

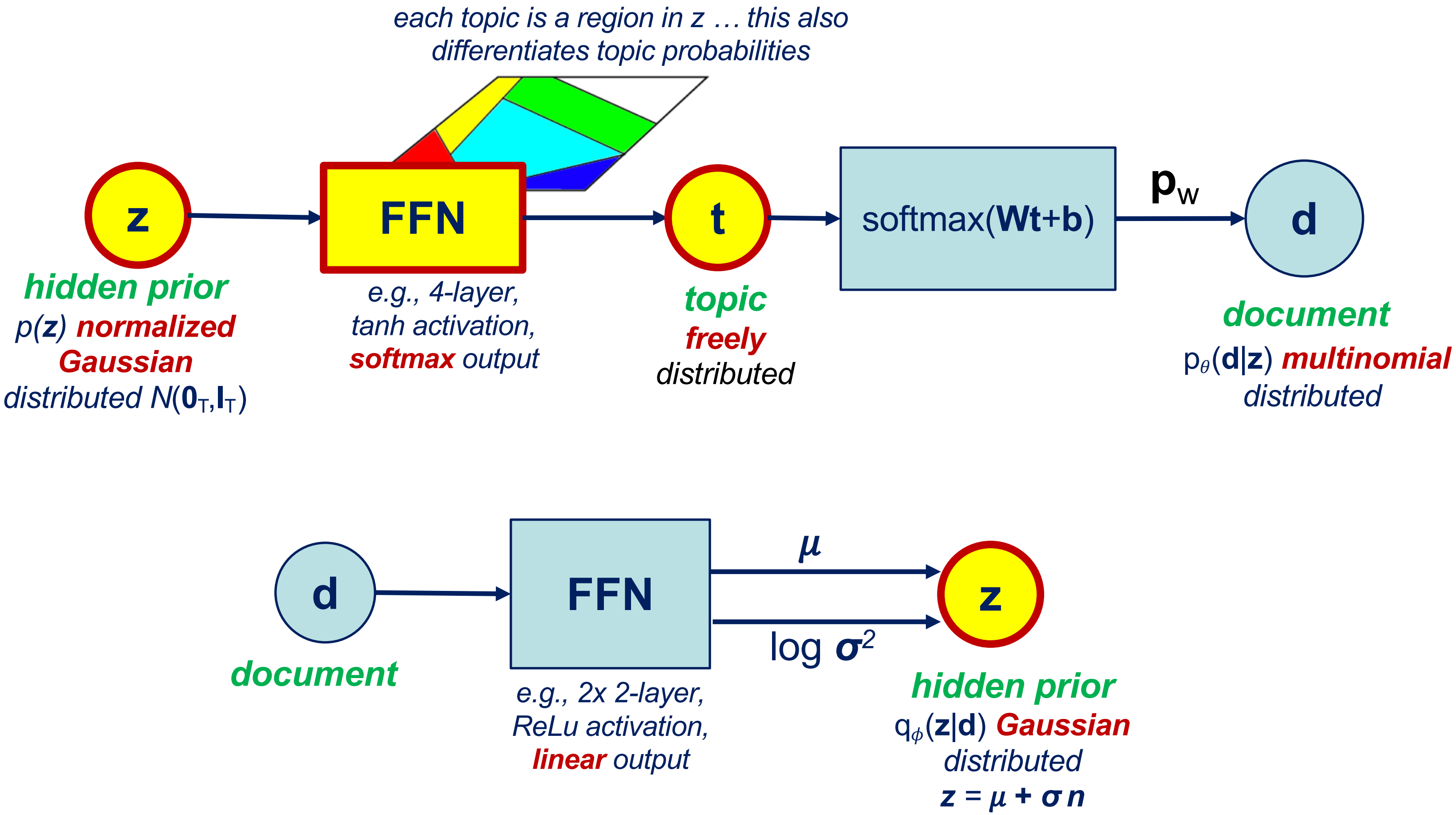
$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$$



Now we know where the topic is!

Take 3 – NFTM

Gui, Lin, et al. "Multi task mutual learning for joint sentiment classification and topic detection," (2020)
https://ieeexplore.ieee.org/document/9112648



NFTM ELBO

optimization target, to be maximized

one-hot-representation of a document = number of occurrences of words in the document

normalized Gaussian samples

decoder model

encoder model

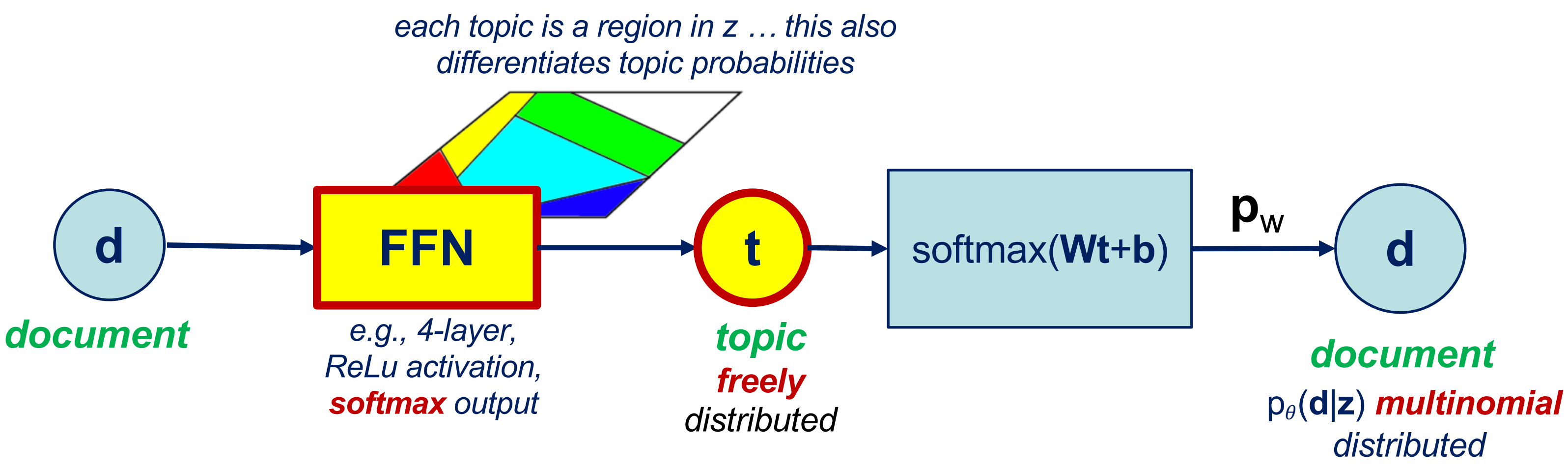
$$\mathcal{L}(\theta,\phi) = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{m} \boldsymbol{d}_m^T \log\Big(\underbrace{\mathrm{softmax}\big(\boldsymbol{b} + \boldsymbol{W}\,\mathrm{FFN}_1}_{\text{decoder map}}(\boldsymbol{\mu}_0(\boldsymbol{d}_m) + \boldsymbol{\sigma}_0(\boldsymbol{d}_m)\,\boldsymbol{n}_{m,\ell})\big)\Big)$$

$$+ \tfrac{1}{2}\sum_{m}\sum_{i} 1 + \log\big(\sigma_{0,i}^2(\boldsymbol{d}_m)\big) - \sigma_{0,i}^2(\boldsymbol{d}_m) - \big(\mu_{0,i}(\boldsymbol{d}_m)\big)^2$$

Our estimate of the topic distribution for the *m*th document!

$$\boxed{\boldsymbol{c}_m = \frac{1}{L}\sum_{\ell=1}^{L} \mathrm{FFN}_1(\boldsymbol{\mu}_0(\boldsymbol{d}_m) + \boldsymbol{\sigma}_0(\boldsymbol{d}_m)\,\boldsymbol{n}_{m,\ell})}$$

NFTM relaxation
mimics NMF

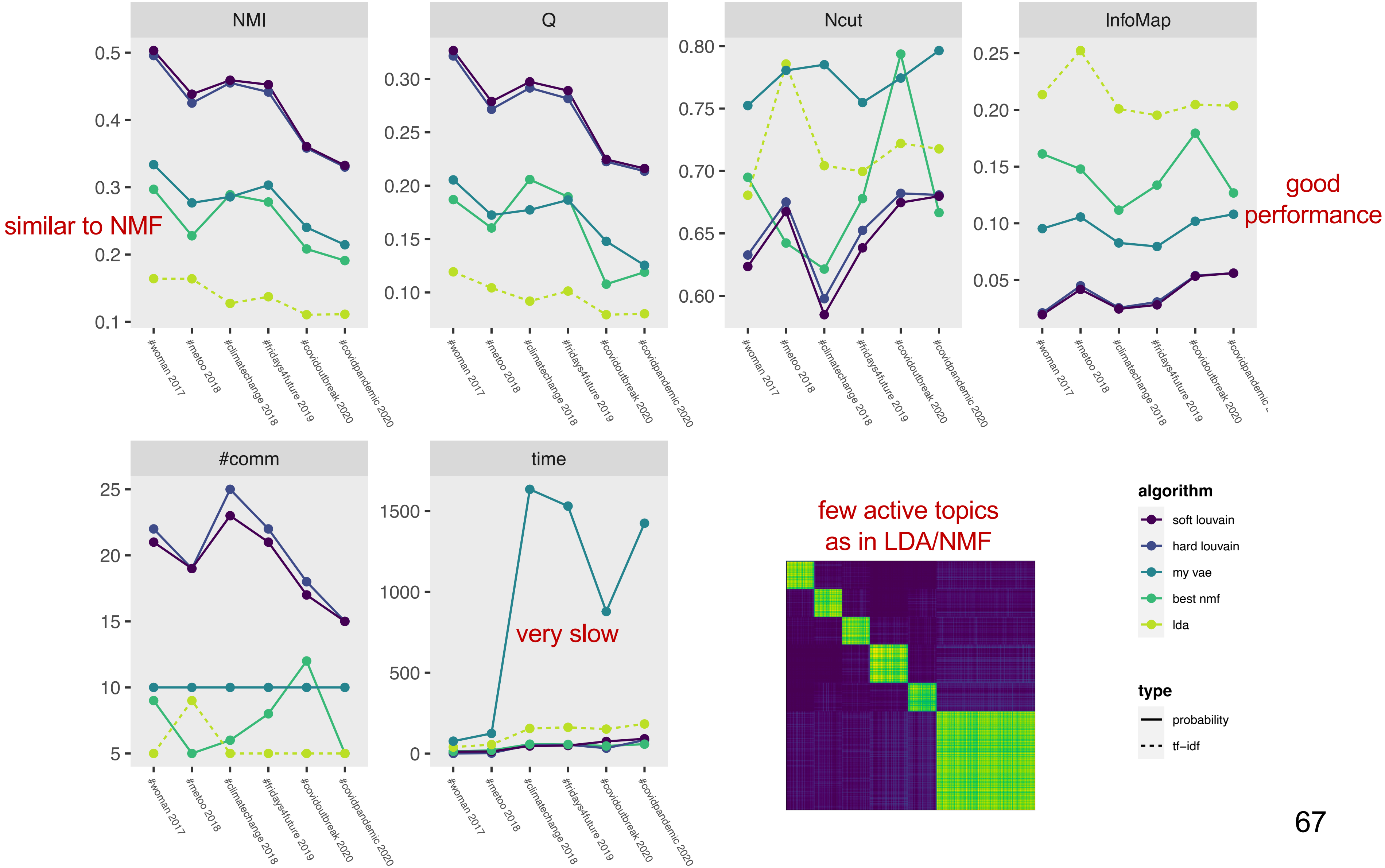


one-hot-representation of a document = number of occurrences of words in the document

decoder model **encoder model**

$$\mathcal{L}\,(\theta, \phi) = \sum_{m} \boldsymbol{d}_m^T \log\big(\mathrm{softmax}(\boldsymbol{b} + \boldsymbol{W}\,\mathrm{FFN}(\boldsymbol{d}_m))\big)$$

A comparison
with NMF, LDA, and Louvain



Takeaways

on VAE applied to topic detection

- Naturally provides a soft topic assignment

- VAE – interesting approach
  - more flexible model than NMF or LDA
  - gives improvements

- Comparison – with Louvain
  - still far away
  - would be nice to see other Deep Learning approaches

… your task! ☺

Transformer Architecture

with application to BERT, RoBERTa, OpenAI GPT

Attention (machine learning)

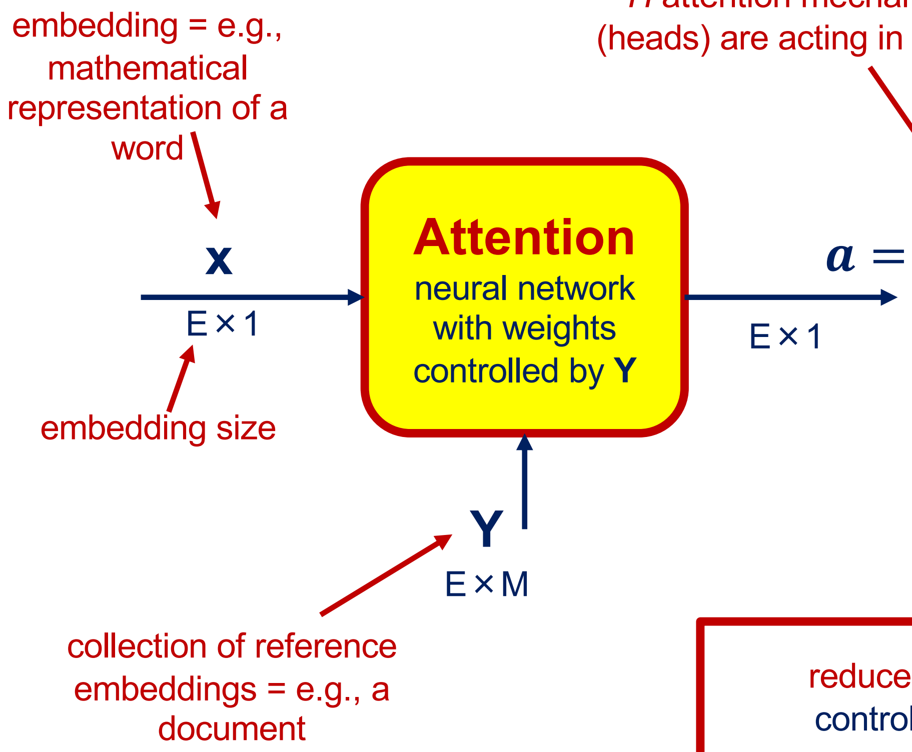
Article Talk

From Wikipedia, the free encyclopedia

In artificial neural networks, **attention** is a technique that is meant to mimic cognitive attention. This effect enhances some parts of the input data while diminishing other parts — the motivation being that the network should devote more focus to the important parts of the data, even though they may be small portion of an image or sentence. Learning which part of the data is more important than another depends on the context, and this is trained by gradient descent.

The Attention Module

Vaswani, Ashish, et al. "Attention is all you need" (2017)



H attention mechanisms (heads) are acting in parallel

$$\boldsymbol{a} = \sum_{j=1}^{H} \underbrace{\boldsymbol{W}_{2j}\,\boldsymbol{Y}}_{\text{expansion}}\ \mathrm{softmax}(\underbrace{\boldsymbol{Y}^T \boldsymbol{W}_{1j}\,\boldsymbol{x}}_{\text{inner products}})$$

similar in spirit to projecting **x** onto the space spanned by **Y**, but with **softmax** normalization

reduced-rank mappings, K<E, control the overall complexity

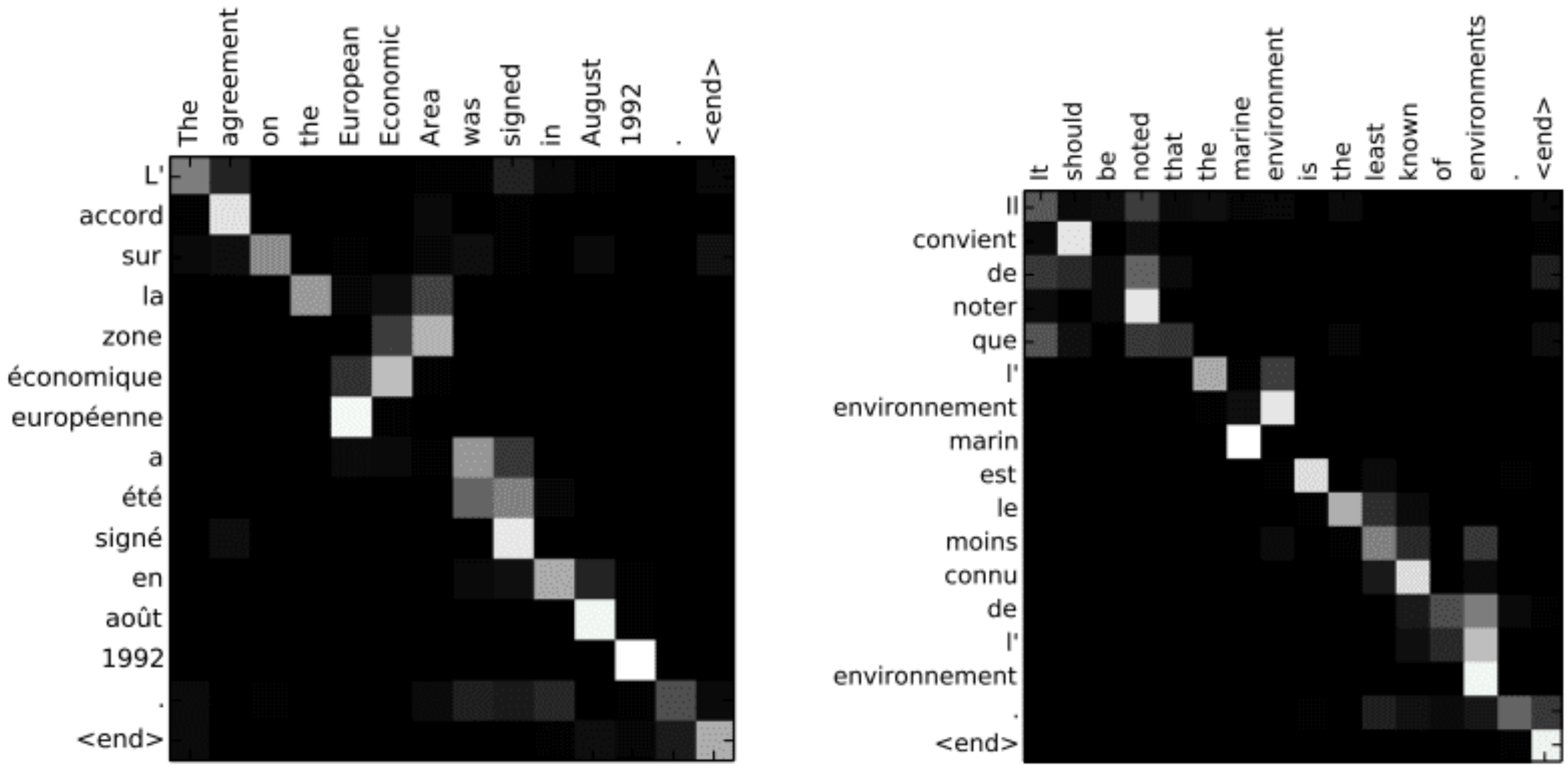
$\boldsymbol{W}_{ij} = \boldsymbol{V}_{ij}\ \boldsymbol{U}_{ij}$

E × E, E × K, K × E

overall complexity 4EKH

Visualizing Attention

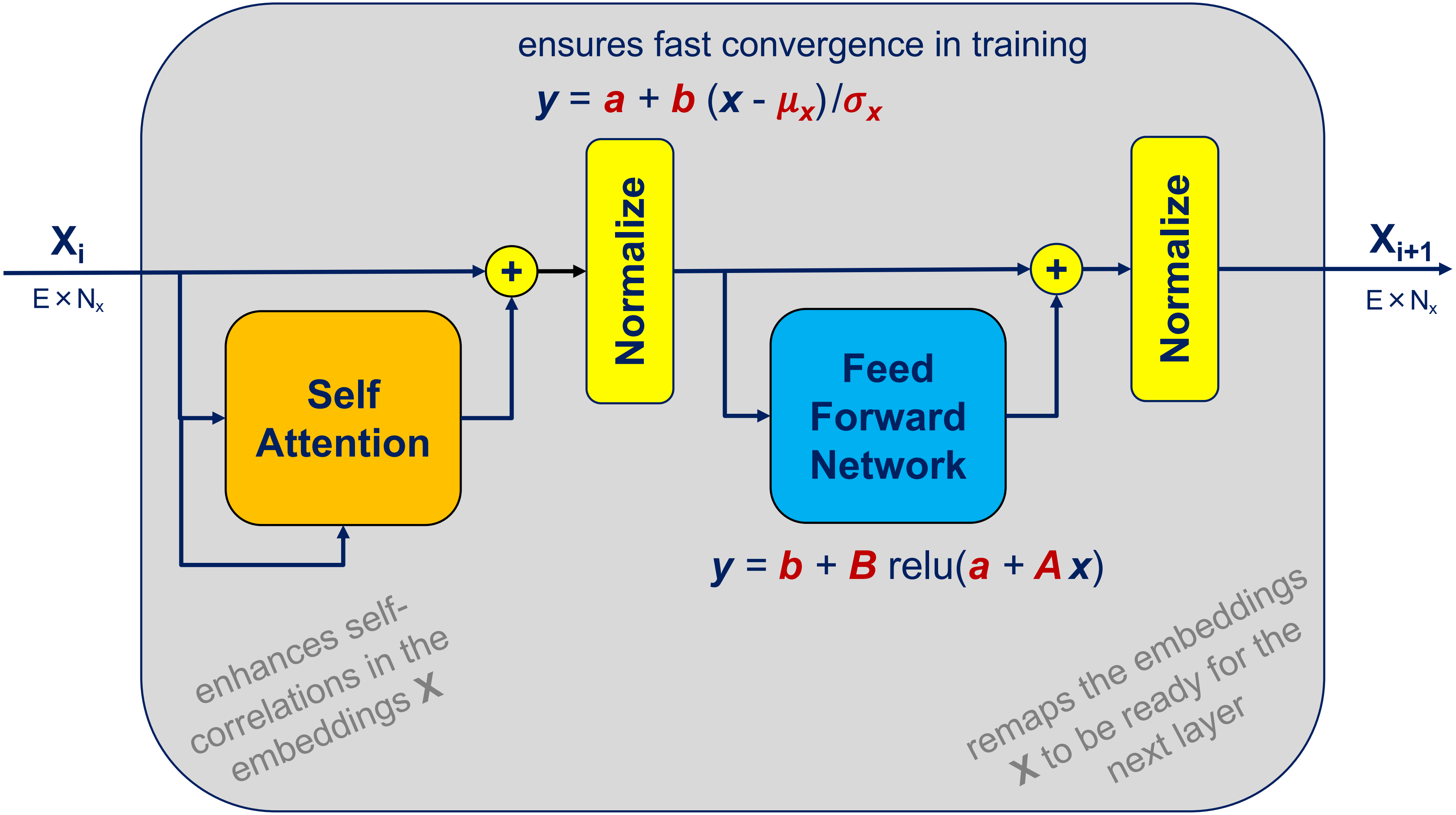
in a translation experiment (X English, Y French)



softmax($\boldsymbol{Y}^T \boldsymbol{W}_{1j} \boldsymbol{X}$)

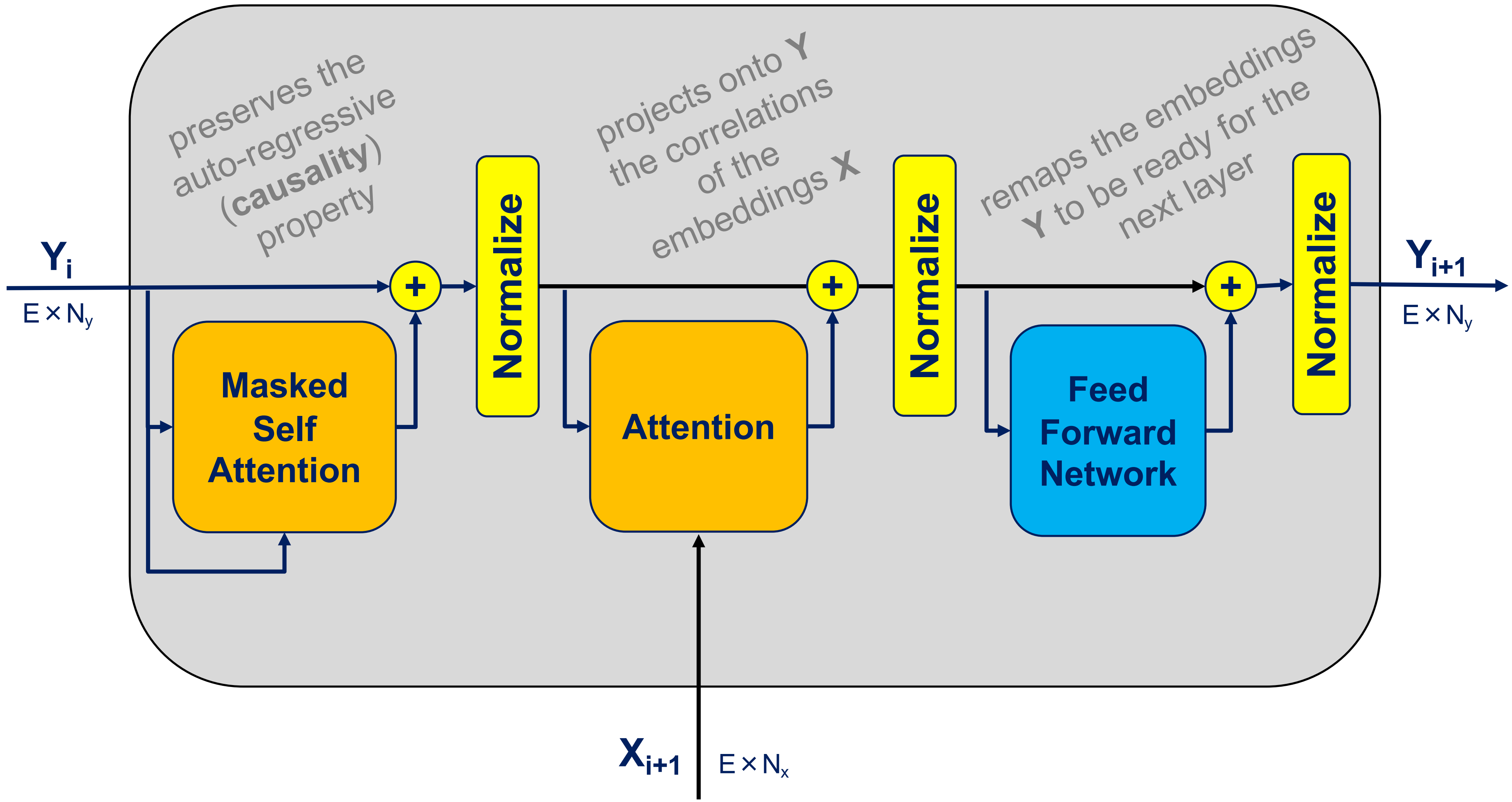
Encoder

a serie of multi-head self-attention modules



Decoder

a serie of attention modules preserving causality

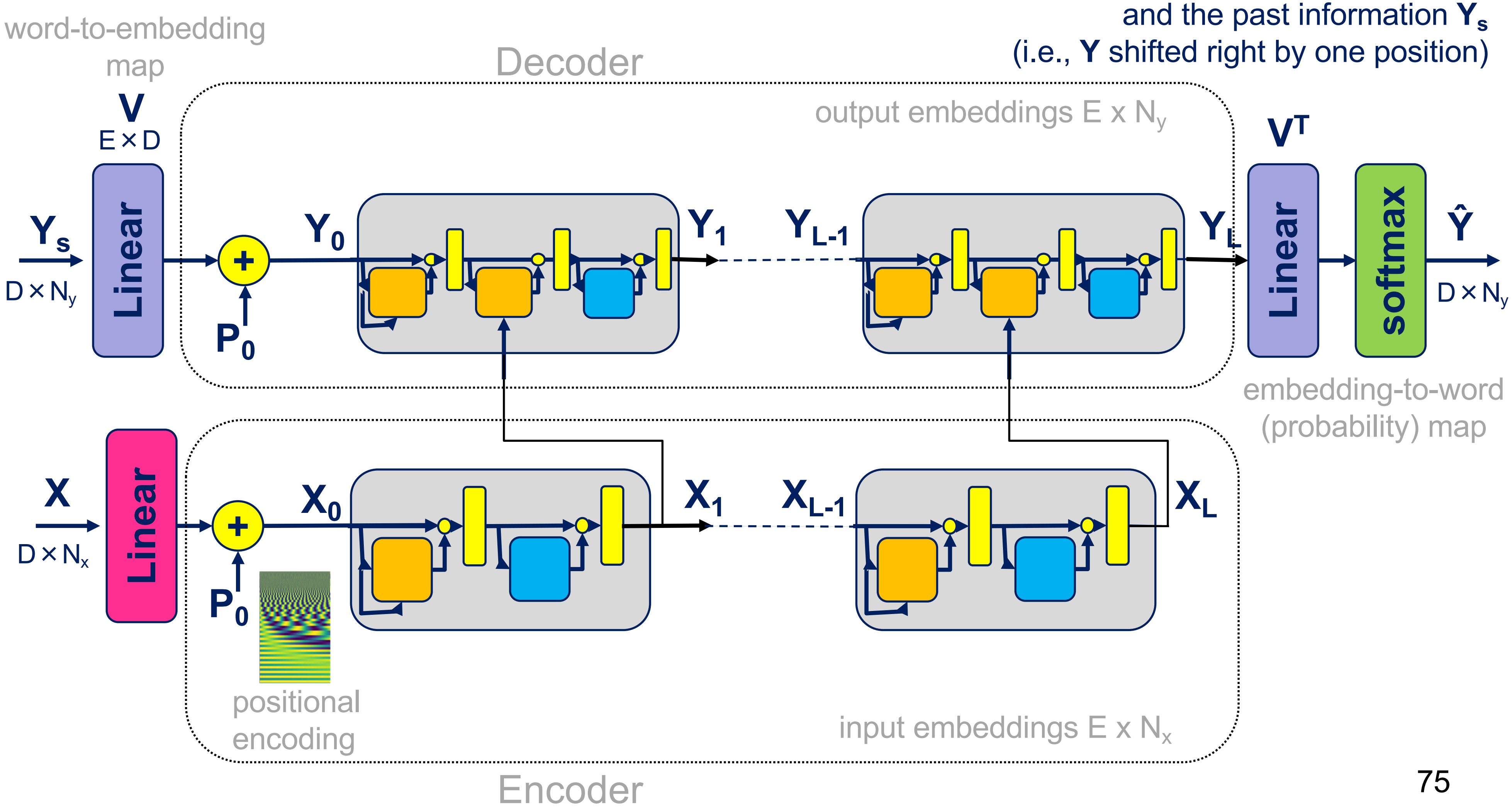


Transformer Architecture

Vaswani, Ashish, et al. "Attention is all you need" (2017)

Google's patent https://patents.google.com/patent/US10452978B2/en

estimate $\hat{\mathbf{Y}}$,
given the alternative representation $\mathbf{X}$,
and the past information $\mathbf{Y}_s$
(i.e., $\mathbf{Y}$ shifted right by one position)



The Annotated Transformer

http://nlp.seas.harvard.edu/2018/04/03/attention.html

tensor2tensor library https://github.com/tensorflow/tensor2tensor

harvardnlp

Members PI Code Publications

The Annotated Transformer

Apr 3, 2018

There is now a new version of this blog post updated for modern PyTorch.

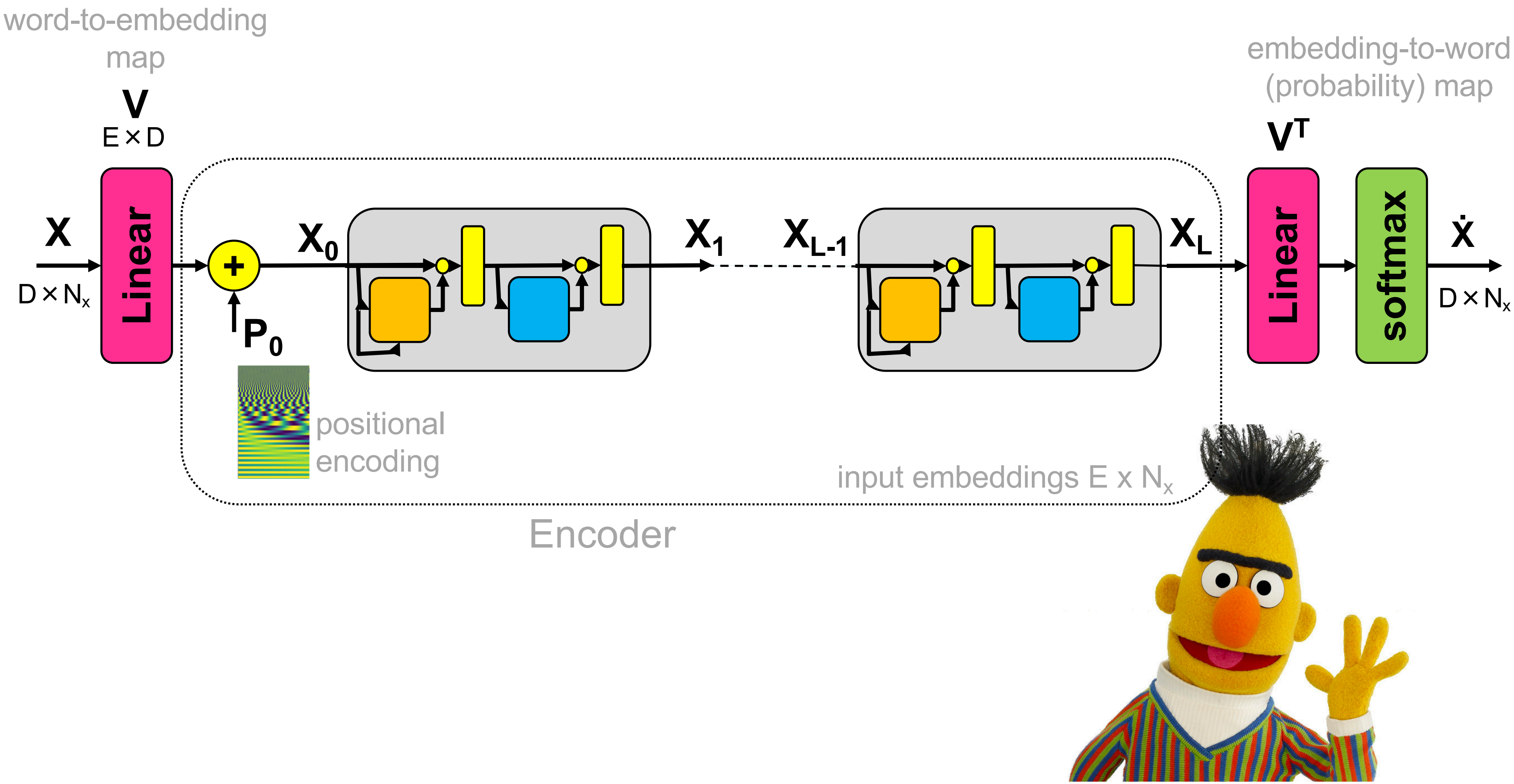
```
from IPython.display import Image
Image(filename='images/aiayn.png')
```

Attention Is All You Need

BERT

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding" (2018)

https://github.com/google-research/bert



Created by researchers at Google AI Language

BERT parameters

	Embeddings size E	Self-attention heads H	Head dimension K = E/H	FFN inner size I = 4E	Parameters per layer $12E^2+9E$	Layers L	Dictionary size D	Total parameters
BERT base	768	12	64	3072	7.1M	12	30.5K	110M
BERT large	1024	16	64	4096	12.6M	24	30.5K	340M

max tokens $N_x = 512$

Created by researchers at Google AI Language

BERT pre-training procedure

BooksCorpus (800M words) + English Wikipedia (2,500M words)

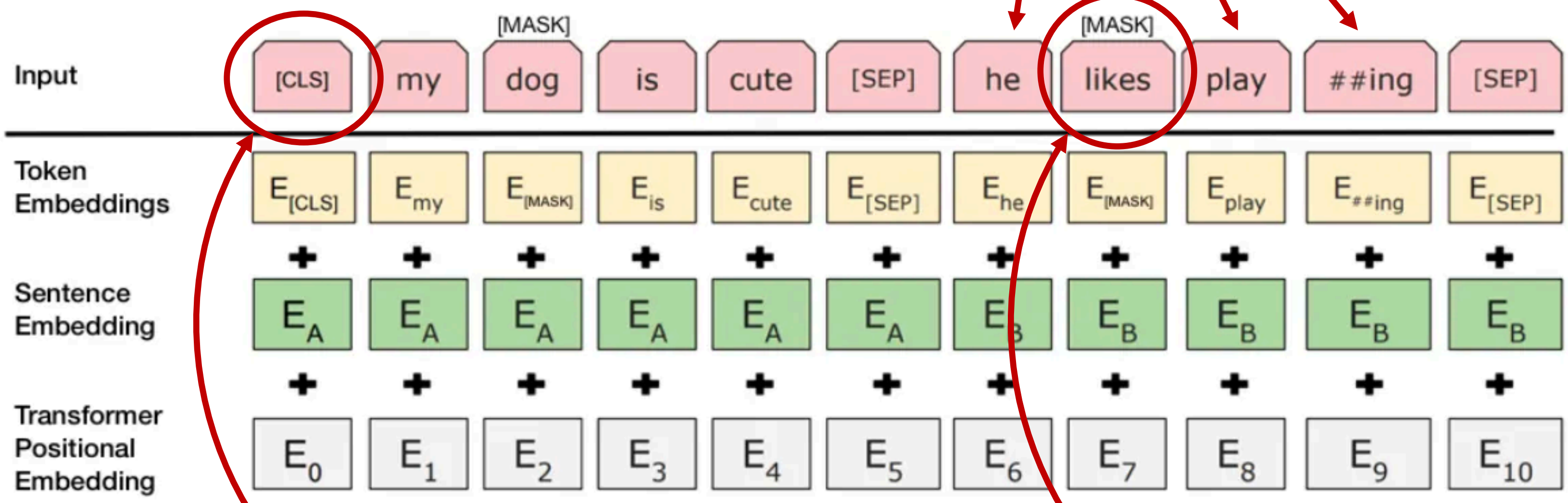
Masked Language Model

15% masked tokens replaced with:

- [MASK] token (80% of the times)
- Original token (10%)
- Random token (10%)

Next Sequence Prediction

- Next sequence (50% of the times)
- Random sequence (50%)



Output [CLS] fed into an additional output layer for softmax classification (of correct/wrong next sequence)

Output masked tokens fed into the output layer V^T and evaluated for probability of correct estimate

RoBERTa

Liu, Yinhan, et al. "Roberta: A robustly optimized BERT pretraining approach" (2019)

Larger training corpora (10x larger)
training on BookCorpus + Wikipedia and also CC-News, OpenWebText, Stories

Dynamic masking
training data was duplicated 10 times so that each sequence is masked in 10 different ways over the 40 epochs of training

Full-sentences without NSP loss
full sentences sampled contiguously from one or more documents, such that the total length is at most 512 tokens

Large mini-batches

A larger byte-level BPE (byte pair encoding) of 50K subword units
a hybrid between character- and word-level representations that allows handling the large vocabularies common in natural language corpora

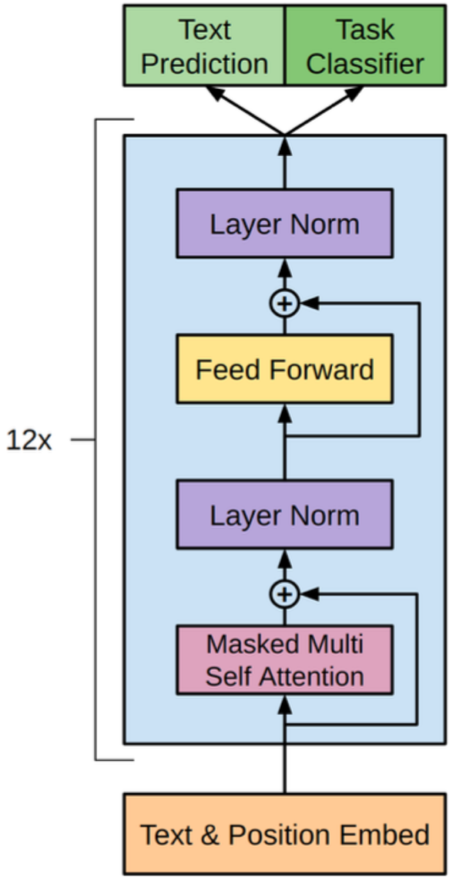
Optimized by University of Washington & Facebook AI

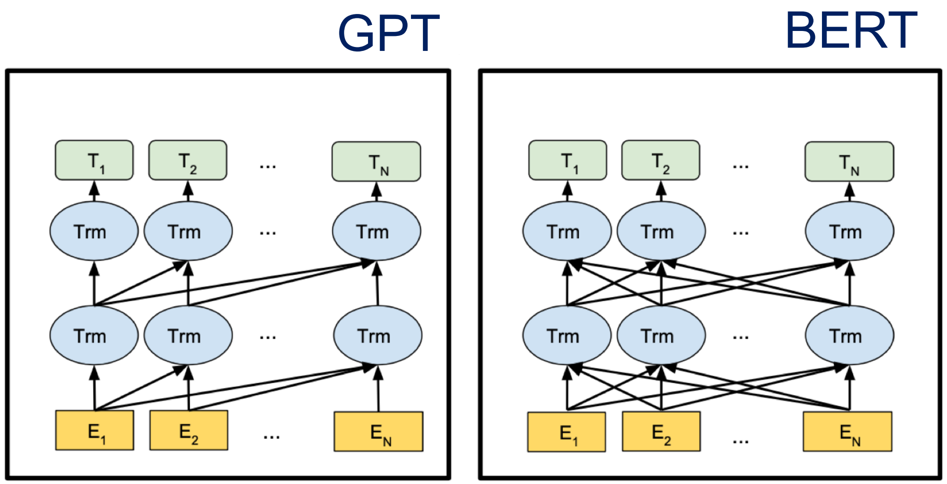
Generative Pre-Training (GPT)

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018)

(unsupervised) pre-training on Language Modelling (no mask)

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$





same parameters of BERT-base, but with Masked Attention
trained on BookCorpus only

Created by researchers at OpenAI

GPT-2

Radford, Alec, et al. "Language models are unsupervised multitask learners" (2019)

McCann et al. (2018)

language provides a flexible way to specify tasks, inputs, and outputs all as a sequence of symbols… it is therfore possible to **train a single model** with **sufficient capacity** to infer and perform many **different tasks**

model gets complex!

data gets larger!

Parameters	Layers	d_{model}	
117M	12	768	GPT, BERT-base
345M	24	1024	BERT-large
762M	36	1280	
1542M	48	1600	GPT-2

WebText

scraping all outbound links (45M links) from Reddit, a social media platform, which received at least 3 karma – exclude WikiPedia

GPT-3

Brown, Tom, et al. "Language models are few-shot learners" (2020)

increasingly larger data and model!

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Layer normalization at the input (plus one at the output)

Sparse attention patterns

alternating dense and locally banded sparse attention patterns in the layers

Byte-level BPE (byte pair encoding) of 50K subword units

also prevent BPE from merging across character categories (to avoid dog, dog!, dog?)

Modified initialization

Created by researchers at OpenAI

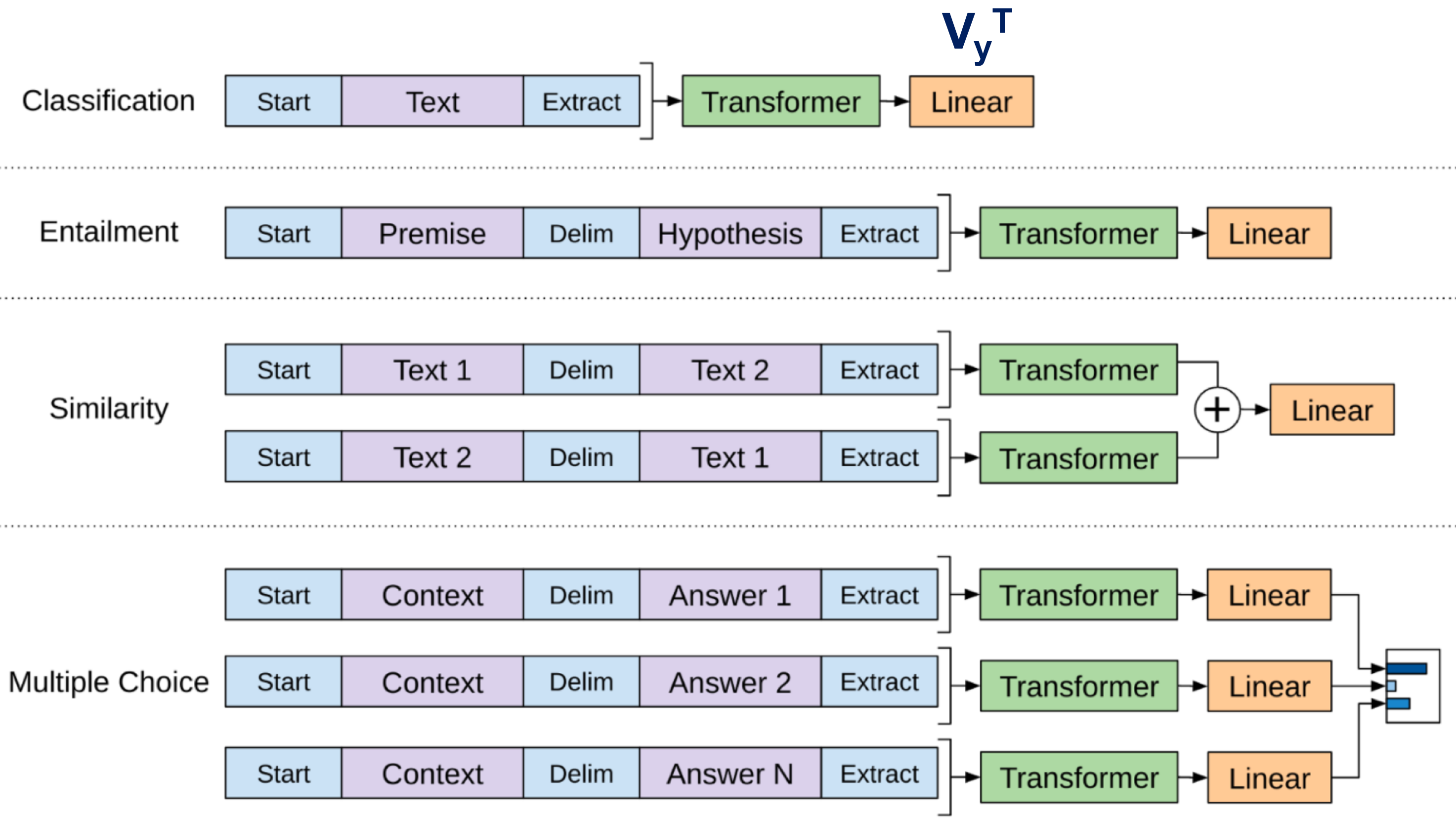
Supervised fine-tuning

training on specific tasks

$$L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

log softmax($\mathbf{V_y}^T\mathbf{X_L}$)

Language Modelling loss



NLP tasks

some fine-tuning possibilities in NLP

Task	Description	Possible approach
Masked language prediction	predict masked words in a text	This is what BERT model is pre-trained for
Text classification or Sentiment analysis	assign a label to a given sequence of text	Apply linear transform+softmax on K classes, and train the model for the specific classification task
Text translation	translate a text	Need to pre-train a full Transfomer Architecture for this task
Summarization	generate a summary of a document	GPT example: context given by a document; then generate 100 tokens by top-2 random sampling (Fan et al., 2018), i.e., take at each step the most likely next word at random among the top-2 candidates; finally select first 3 sentences as abstract
Question answering	answer a question	GPT example: the context of the language model is seeded with example question answer pairs which helps the model infer the short answer style of the dataset
Document question answering	answer a question on a given text	GPT example: context seeded by a text; then as for question answering
Conversational	ChatBot	InstructGPT/ChatGPT: Fine-tuned models using reinforcement learning from human feedback

Hugging Face

https://huggingface.co/docs/transformers/v4.29.1/en/index

State-of-the-art Machine Learning
for PyTorch, TensorFlow, and JAX

PyTorch TensorFlow

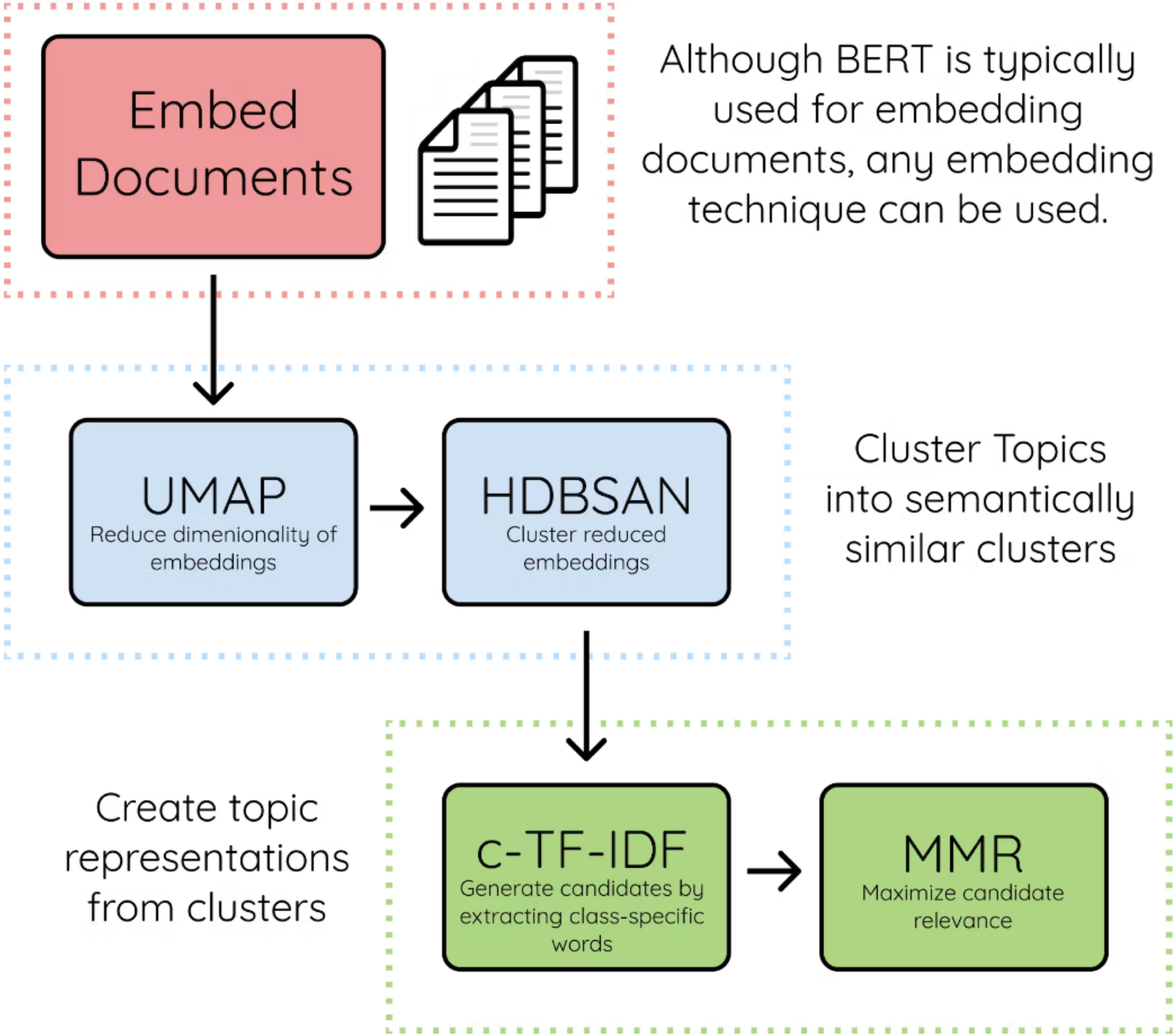
ALBERT, BART, **BERT**, BigBird, BigBird-Pegasus, BioGpt, BLOOM, CamemBERT, CANINE, ConvBERT, CTRL, Data2VecText, DeBERTa, DeBERTa-v2, DistilBERT, ELECTRA, ERNIE, ErnieM, ESM, FlauBERT, FNet, Funnel Transformer, GPT-Sw3, **OpenAI GPT-2**, GPTBigCode, GPT Neo, GPT NeoX, GPT-J, I-BERT, LayoutLM, LayoutLMv2, LayoutLMv3, LED, LiLT, LLaMA, Longformer, LUKE, MarkupLM, mBART, MEGA, Megatron-BERT, MobileBERT, MPNet, MVP, Nezha, Nyströmformer, OpenLlama, **OpenAI GPT**, OPT, Perceiver, PLBart, QDQBert, Reformer, RemBERT, **RoBERTa**, RoBERTa-PreLayerNorm, RoCBert, RoFormer, SqueezeBERT, TAPAS, Transformer-XL, XLM, XLM-RoBERTa, XLM-RoBERTa-XL, XLNet, X-MOD, YOSO

BERT Topic

exploiting embeddings for topic detection

BERTopic

Grootendorst, «BERTopic: Neural topic modeling with a class-based TF-IDF procedure» (2022) https://arxiv.org/abs/2203.05794



BERTopic in Python

bertopic package https://maartengr.github.io/BERTopic/

```
!pip install bertopic
from bertopic import BERTopic
from sentence_transformers import SentenceTransformer
```

initialise model

```
sentence_model = SentenceTransformer("all-MiniLM-L6-v2")
bert_model = BERTopic(embedding_model=sentence_model,
                      min_topic_size=20,nr_topics='auto')
```

fit model

```
docs = list(df2["text_sup_clean"])
topics, probabilities = bert_model.fit_transform(docs)
```

```
topics = bert_model.reduce_outliers(docs, topics)
```

reduce outliers

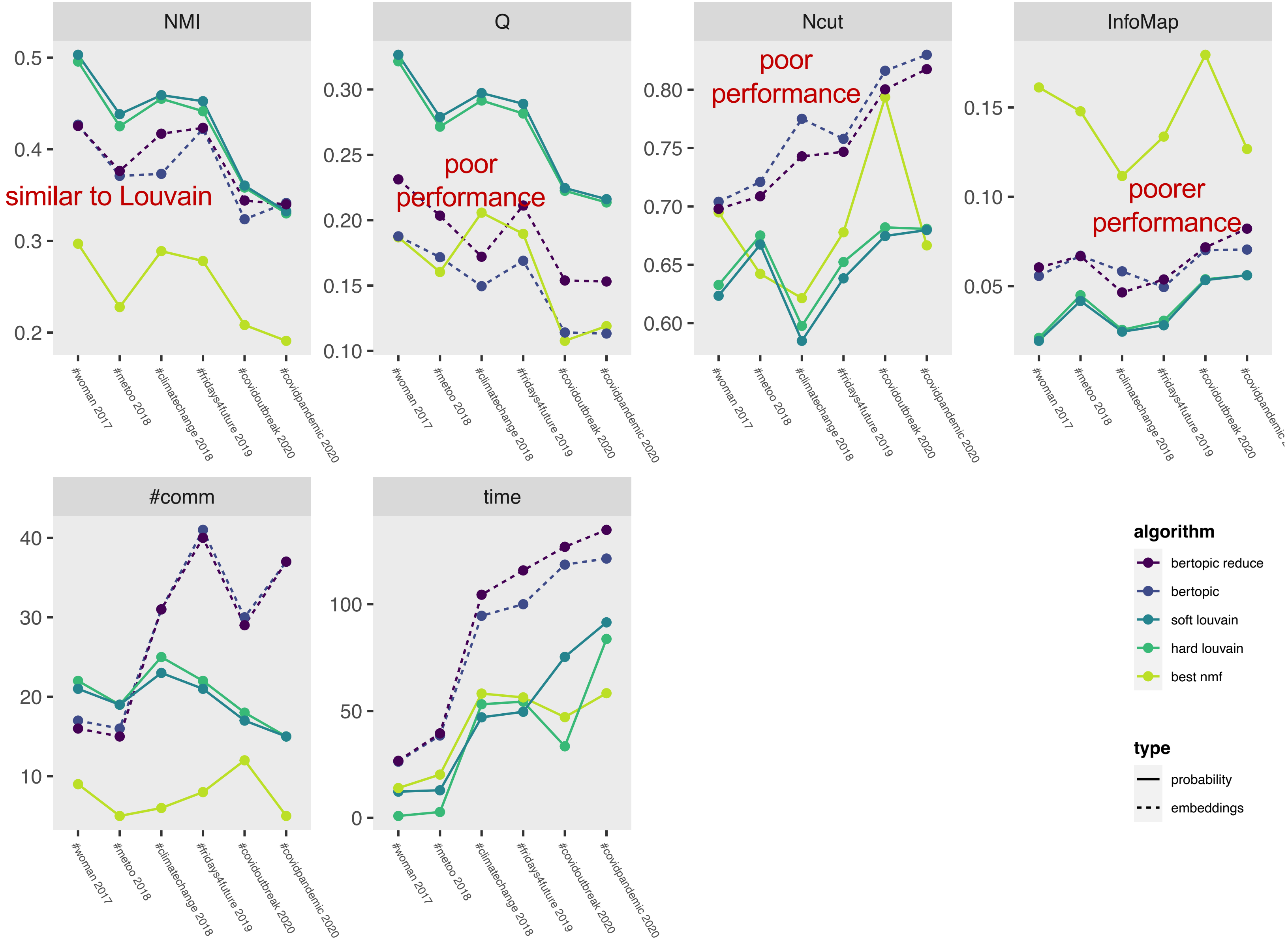
```
# extract community assignments
C = sps.csr_matrix((len(topics),max(topics)+2))
for i in range(C.shape[1]):
  C[np.array(topics)==(i-1),i] = 1

# remove zero assignments
C = C[:,np.unique(scipy.sparse.find(C)[1])]
```

extract C from topic assignment

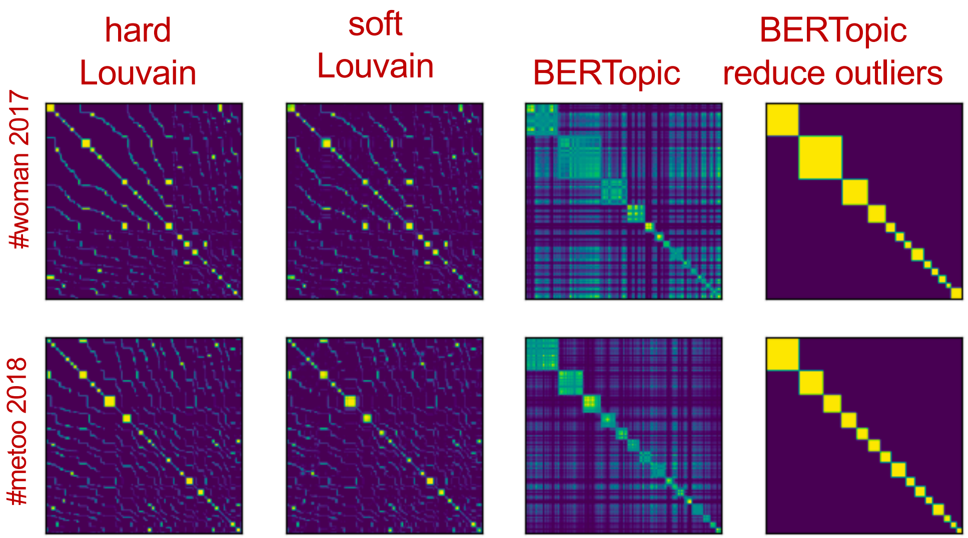
Università degli Studi di Padova

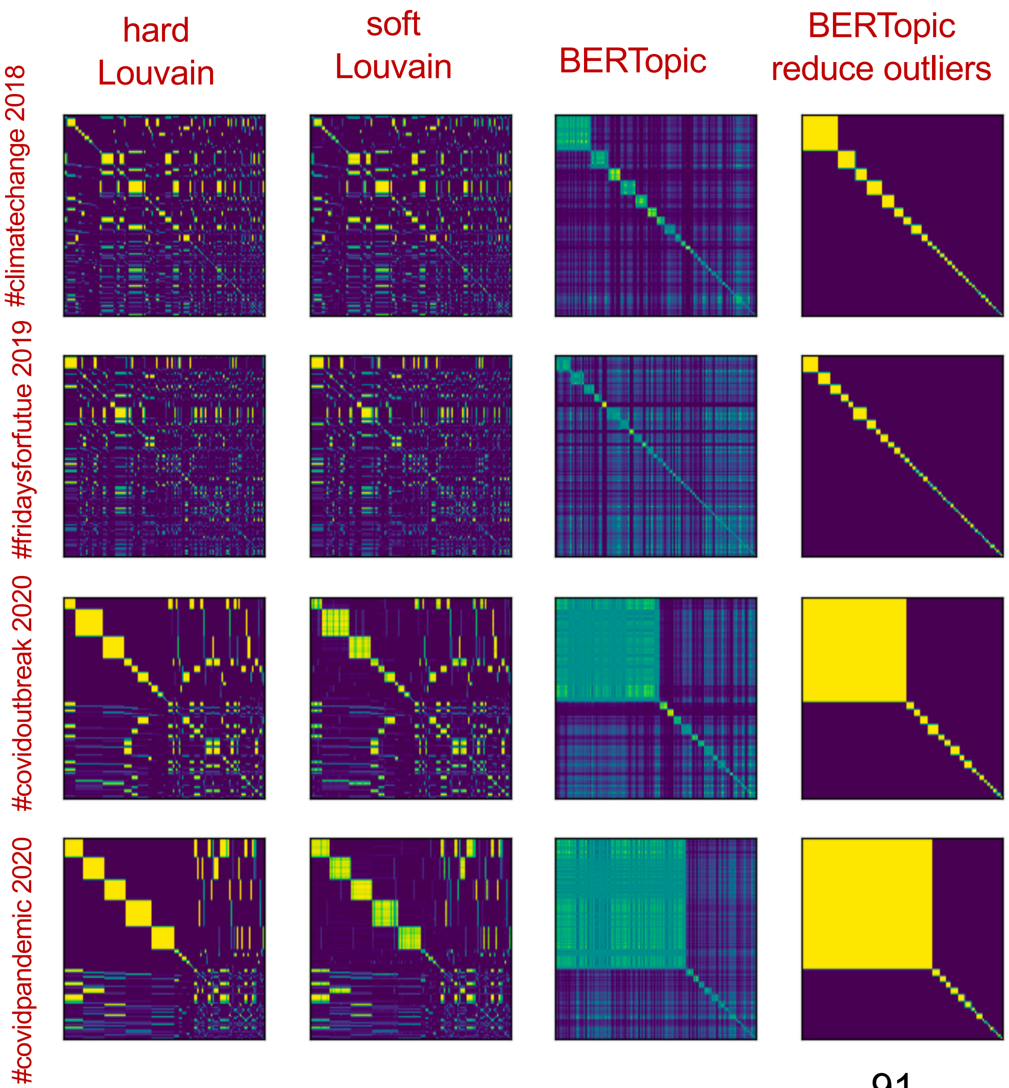
A comparison
with NMF and Louvain



A comparison

a $\boldsymbol{C^T C}$ pattern overview





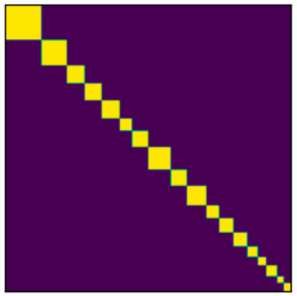
BERTopic and Louvain mostly identify different communities, more balanced in Louvain

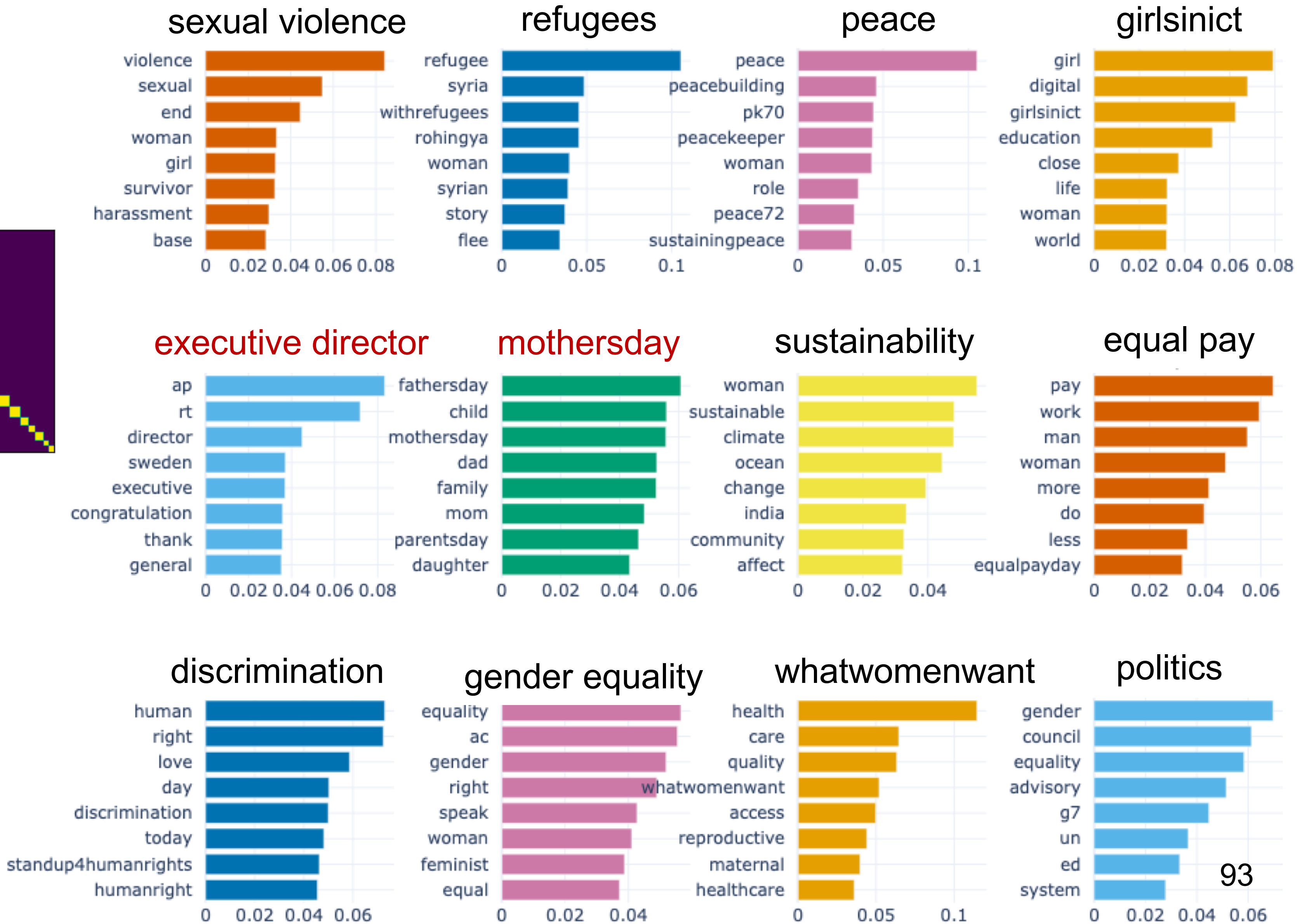
bert_model.visualize_documents(docs)

#metoo2018

bert_model.visualize_barchart()

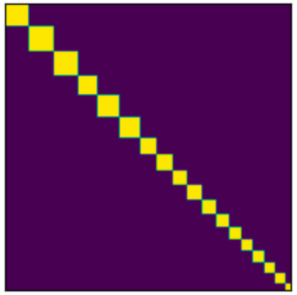
#metoo2018

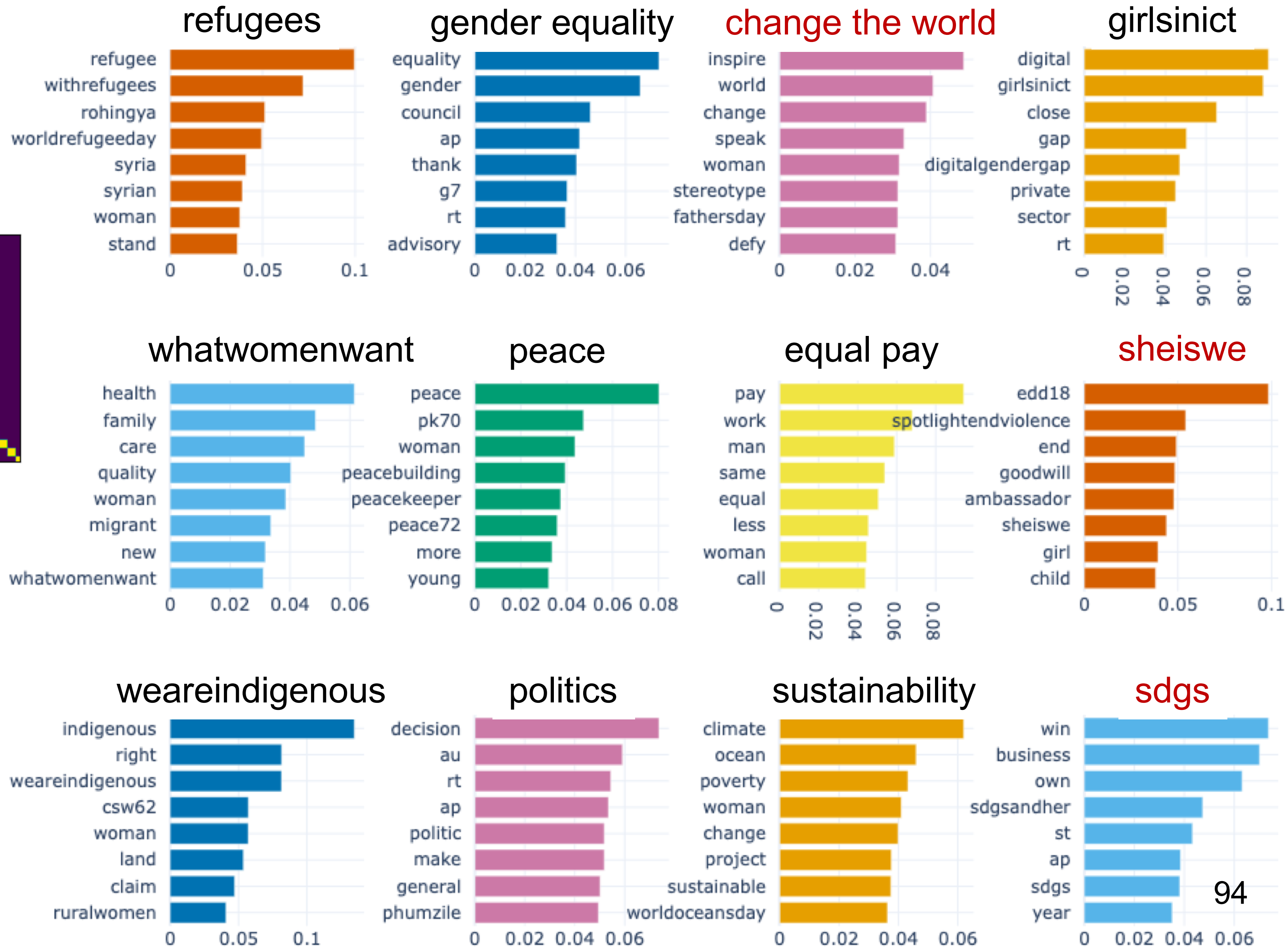




Same barchart with Louvain

#metoo2018





How to use BERTopic barchart

with your own topic assignment *C* in python

```python
def bertopic_overwrite(bert_model_in,docs,C):
  bert_model = copy.deepcopy(bert_model_in)

  # build the documents dataframe: 'Document' + "Topic"
  documents = pd.DataFrame(docs,columns=['Document'])
  tmp = np.array([C[i].argmax() for i in range(C.shape[0])])
  documents["Topic"] = tmp

  # update topic assignment
  bert_model.topics_ = tmp.tolist()

  # build cf-idf values
  documents_per_topic = documents.groupby(['Topic'],
                  as_index=False).agg({'Document': ' '.join})
  c_tf_idf_, words = bert_model._c_tf_idf(documents_per_topic)
  bert_model.c_tf_idf_ = c_tf_idf_

  # extract words representations
  topic_representations_ = bert_model._extract_words_per_topic(words, documents)
  bert_model.topic_representations_ = topic_representations_
  bert_model.topic_labels_ = {key: f"{key}_" + "_".join([word[0] for word in values[:4]])
                              for key, values in
                              bert_model.topic_representations_.items()}

  # exit
  return bert_model
```

Takeaways

on BERTopic applied to topic detection

- Naturally provides a hard topic assignment
- Useful tool
- More readable output with deep cleaned text
  but same performance
- Comparison – with Louvain
  weaker in general, especially in modularity
  equivalent NMI = relevant topics
  lower modularity = the documents that identify the topics are less distinguishable
  higher complexity involved
  less balanced topics, but generally meaningful
  topics correlated with Louvain

Sentiment analysis

adding useful insights to your data

Socio-psychological markers
beyond simple sentiment

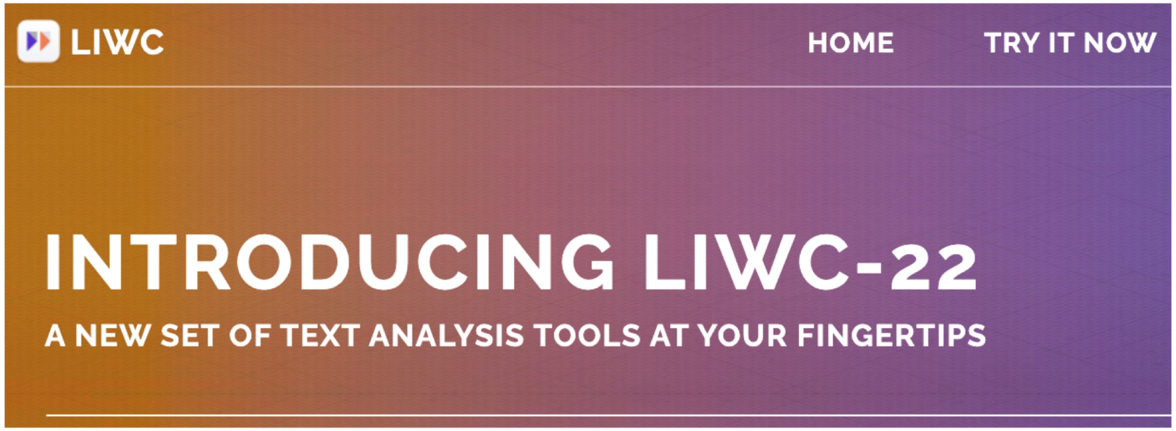
- Sentiment – e.g., positive, negative, neutral
  enduring cognitive content that defines the affective state
- Emotion – e.g., anger, disgust, fear, joy, sadness
  intense affective state of short duration with a precise cause
- Ingroup bias – e.g., use of pronouns I, we, us
  tendency to favor one's own group over other groups
- Outgroup bias – e.g., use of pronoun they
  tendency to dislike members of groups we don't identify with
- Agency – e.g., use of action verbs do, take, make
  perception that an individual is able to contribute to/a group can collectively reach a social change

UNIVERSITÀ DEGLI STUDI DI PADOVA

LIWC linguistic inquiry and word count

Tausczik, Pennebaker. "The psychological meaning of words: LIWC and computerized text analysis methods." (2010)

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=79d2494cc10a9633c42115df84bb74ed447080f6



https://www.liwc.app/

- word count (or dictionary) methodology
- over 60 dictionaries coded and validated for their accuracy in reflecting psychological content
- simplicity of implementation and usage
- state-of-the-art in psychology
- one licence available in the instructor's PC ☺

UNIVERSITÀ DEGLI STUDI DI PADOVA

LIWC categories

ingroup and outgroup

Category	Examples	Words in Category	Psychological Correlates
Linguistic processes			
Word count			Talkativeness, verbal fluency
Words/sentence			Verbal fluency, cognitive complexity
Dictionary words	(Percentage of all words captured by the program)		Informal, nontechnical language
Words >6 letters	(Percentage of all words longer than 6 letters)		Education, social class
Total function words		464	
Total pronouns	I, them, itself	116	Informal, personal
Personal pronouns	I, them, her	70	Personal, social
First-person singular	I, me, mine	12	Honest, depressed, low status, personal, emotional, informal
First-person plural	We, us, our	12	Detached, high status, socially connected to group (sometimes)
Second person	You, your, thou	20	Social, elevated status
Third-person singular	She, her, him	17	Social interests, social support
Third-person plural	They, their, they'd	10	Social interests, out-group awareness (sometimes)

ingroup

outgroup

LIWC categories

goal orientation, aggression, social concern, emotionality

Category	Examples	Words in Category	Psychological Correlates
Indefinite pronouns	It, it's, those	46	
Articles	A, an, the	3	Use of concrete nouns, interest in objects and things
Common verbs	Walk, went, see	383	
Auxiliary verbs	Am, will, have	144	Informal, passive voice
Past tense	Went, ran, had	145	Focus on the past
Present tense	Is, does, hear	169	Living in the here and now
Future tense	Will, gonna	48	Future and goal oriented
Adverbs	Very, really, quickly	69	
Prepositions	To, with, above	60	Education, concern with precision
Conjunctions	And, but, whereas	28	
Negations	No, not, never	57	Inhibition
Quantifiers	Few, many, much	89	
Numbers	Second, thousand	34	
Swear words	Damn, piss, fuck	53	Informal, aggression,
Psychological processes			
Social processes	Mate, talk, they, child	455	Social concerns, social support
Family	Daughter, husband	64	
Friends	Buddy, friend, neighbor	37	
Humans	Adult, baby, boy	61	
Affective processes	Happy, cried, abandon	915	Emotionality

focus on past, present or future

Università degli Studi di Padova

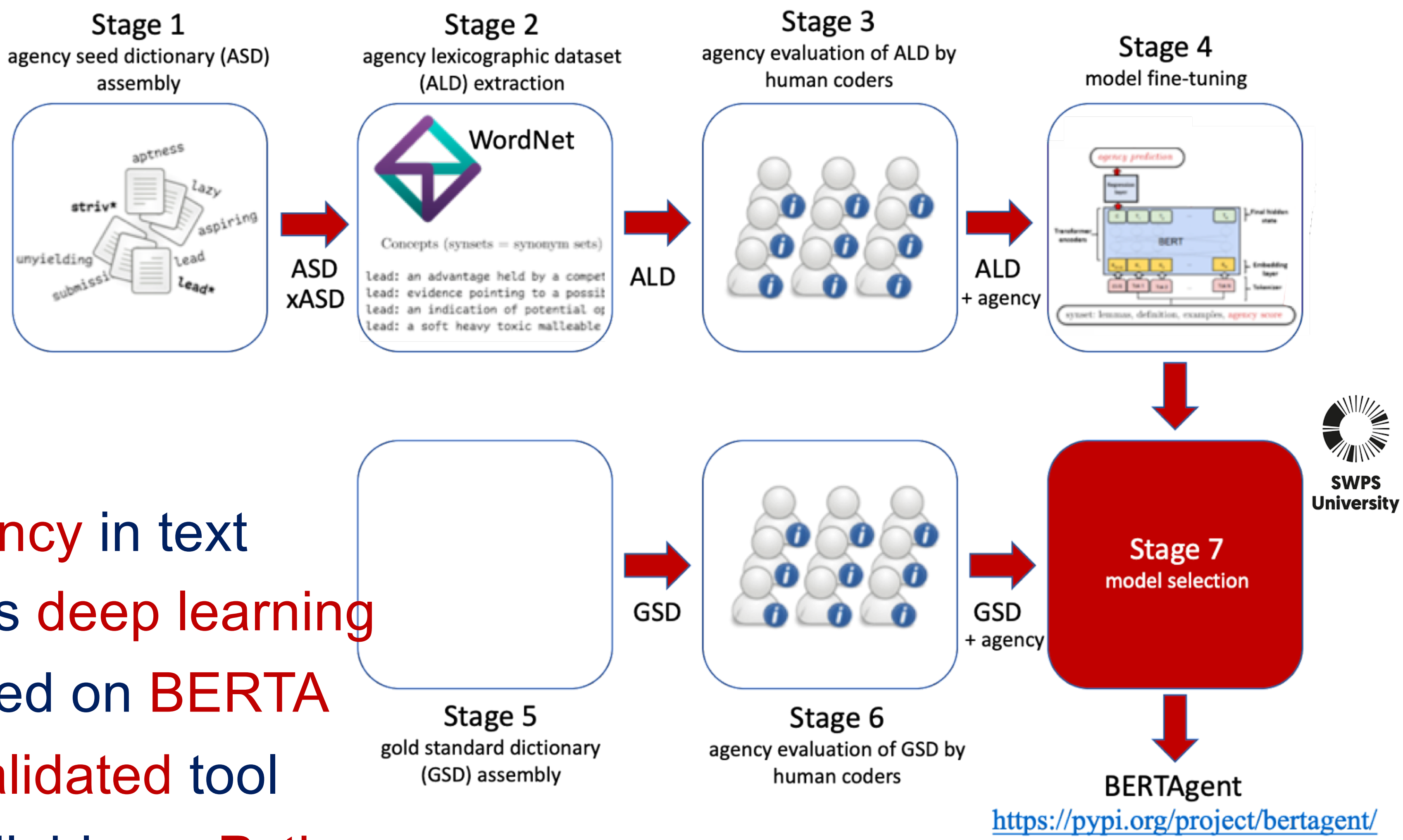
LIWC categories
a full list

WC	Analytic	Clout	Authentic	Tone	WPS	Sixltr	Dic	function	pronoun
ppron	**i**	**we**	**you**	**shehe**	**they**	ipron	article	prep	auxverb
adverb	conj	negate	verb	adj	compare	interrog	number	quant	affect
posemo	**negemo**	**anx**	**anger**	**sad**	social	family	friend	female	male
insight	cause	discrep	tentat	certain	differ	percept	see	hear	feel
bio	**body**	**health**	**sexual**	ingest	drives	**affiliation**	**achieve**	**power**	**reward**
risk	**focus past**	**focus present**	**focus future**	relativ	motion	space	time	work	leisure
home	money	**relig**	death	**informal**	**swear**	netspeak	assent	nonflu	filler
AllPunc	Period	Comma	Colon	SemiC	QMark	Exclam	Dash	Quote	Apostro
Parenth	cogproc								

Choose the ones of interest to your project!

BERTAgent

Nikadon et al., «BERTAgent: A novel tool to quantify agency in textual data,» (2023) https://psyarxiv.com/qw6u3



- ❑ agency in text
- ❑ uses deep learning
- ❑ based on BERTA
- ❑ a validated tool
- ❑ available on Python

UNIVERSITÀ DEGLI STUDI DI PADOVA

Validation of BERTAgent

deep learning wins versus DWC = dictionary word count

Variable	*M*	*SD*	1	2	3	4	5	6	7	8
1. HumEval	0.12	1.54								
2. PietA	0.05	0.05	.17** [.06, .28]		-1.25	0.28	0.05	5.35**	-1.78	-10.95**
3. PietB	0.02	0.03	.25** [.14, .35]	.40** [.30, .49]		1.27	1.16	6.58**	-0.70	-10.00**
4. PietC	0.05	0.05	.17** [.06, .28]	.99** [.99, 1.00]	.40** [.30, .49]		0.03	5.34**	-1.80	-10.93**
5. NicoPos	0.03	0.04	.17** [.05, .27]	.18** [.07, .29]	.23** [.12, .34]	.17** [.06, .28]		5.49**	-3.81**	-11.08**
6. NicoNeg	0.01	0.03	-.28** [-.38, -.17]	-.10 [-.21, .01]	-.01 [-.12, .11]	-.10 [-.21, .02]	-.03 [-.14, .09]		-5.73**	-13.40**
7. NicoCom	0.02	0.05	.30** [.19, .40]	.20** [.09, .31]	.19** [.08, .30]	.19** [.08, .30]	.82** [.78, .85]	-.60** [-.67, -.52]		-10.38**
8. BATot	0.09	0.35	.78** [.73, .82]	.21** [.10, .31]	.24** [.13, .34]	.20** [.09, .31]	.22** [.11, .33]	-.42** [-.51, -.33]	.42** [.33, .51]	

Human evaluation

BERTAgent

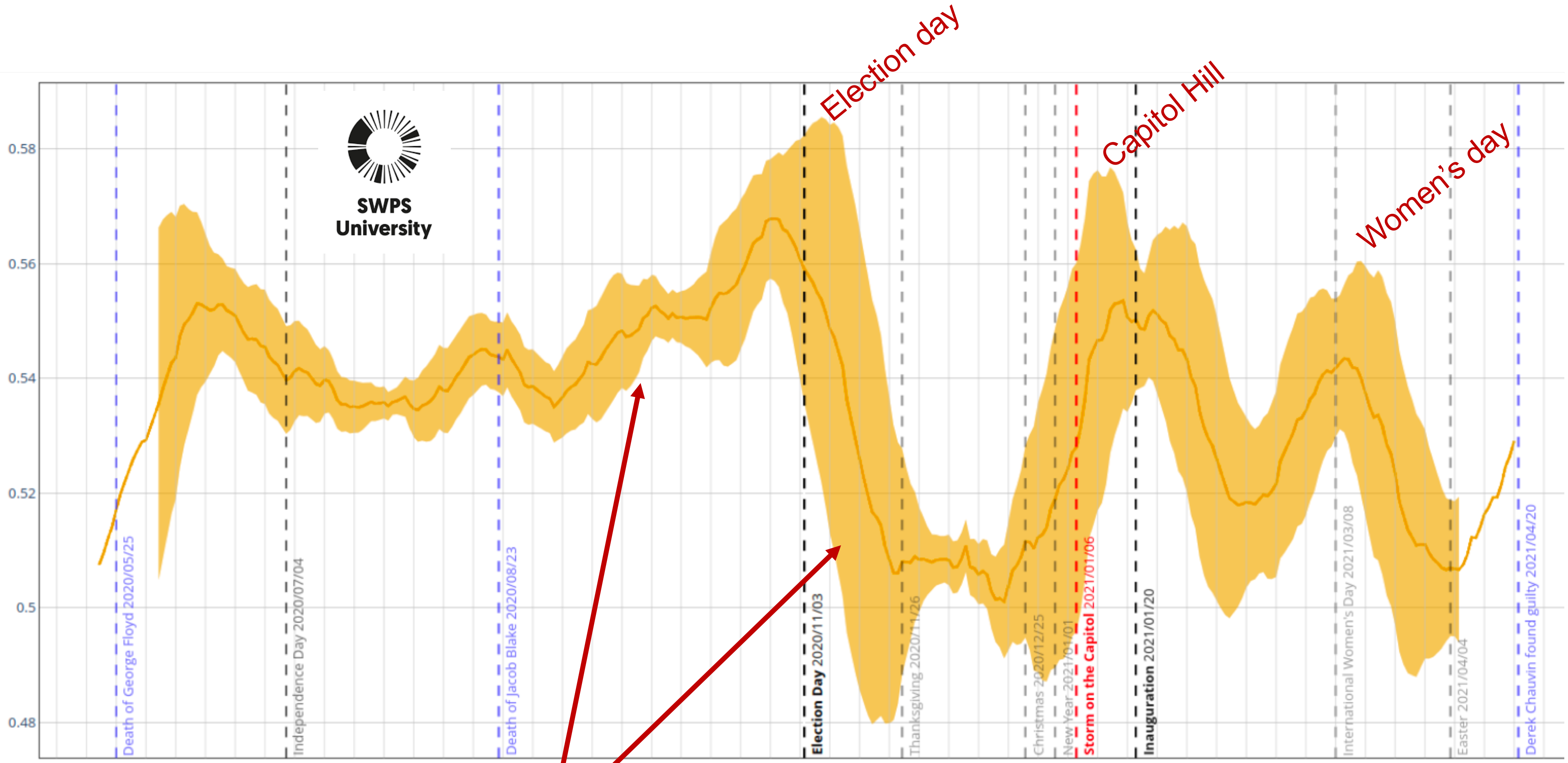
best correlation with Human evaluation

Z-statistics: correlation is statistically more relevant that DWC

Agency in US elections

Twitter, 2020-2021

by Jan Nikadon @ swps



agency raises before
elections than drops

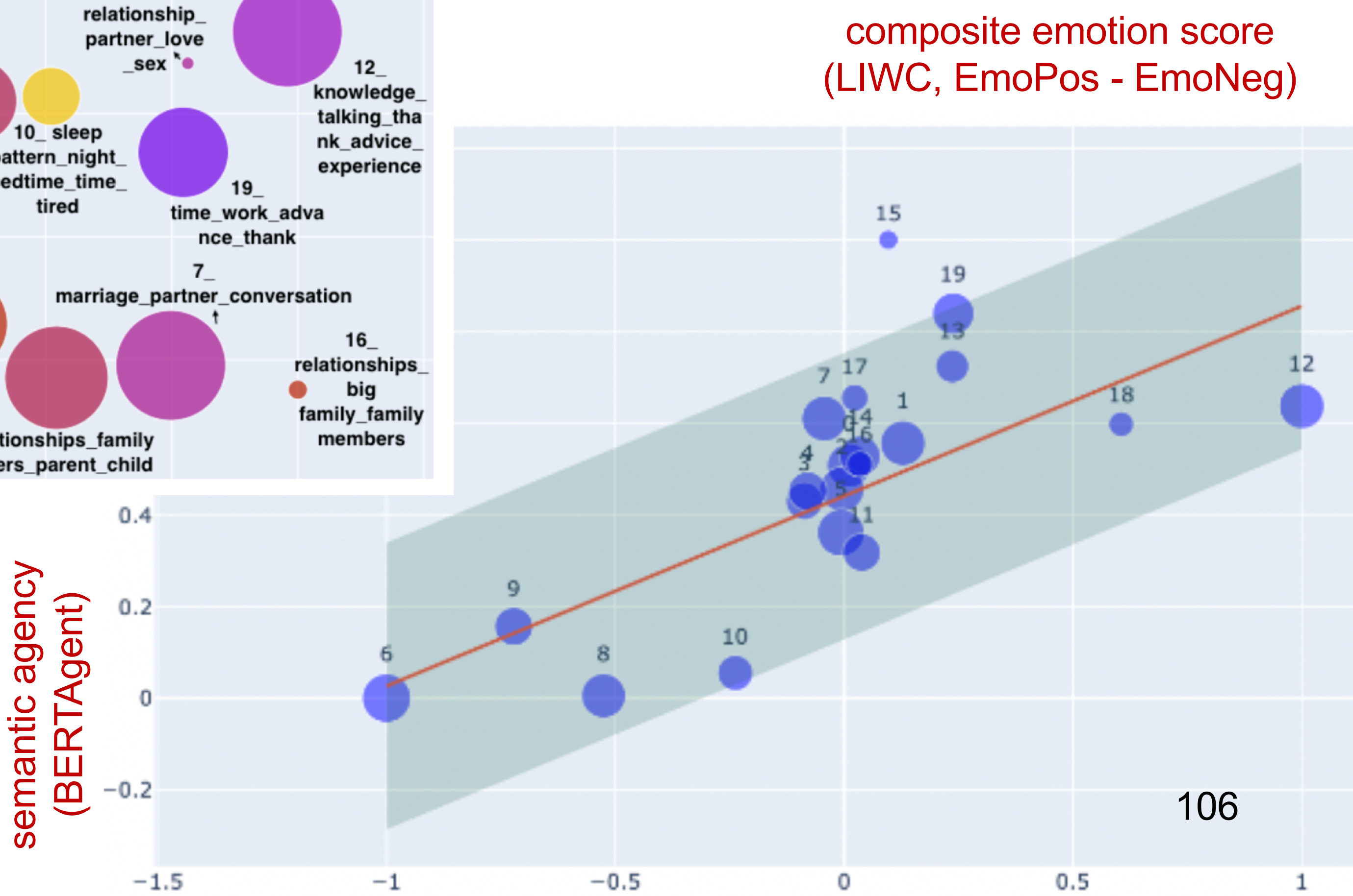
UNIVERSITÀ DEGLI STUDI DI PADOVA

Agency in postpartum depression

Reddit Posts 2021

by Selen Arslan @ unipd/swps

reddit



SWPS University

BERTAgent in Python

Free software: GNU General Public License v3

PyPi: https://pypi.org/project/bertagent/

```
[1] !pip install bertagent
```

install, import, set instance

```
from bertagent import BERTAgent
ba0 = BERTAgent()
```

input sentences must be superficially cleaned

```
# provide example sentences
sents = ["hardly wo
         "hard work
         "striving
         "strugglin
         "strugglin
         "unable to
         "this car
         "this car
         "this poli
         ]
```

SWPS University

BERTAgent output

```
'hardly working individual' : -0.57
'hard working individual' : 0.44
'striving to achieve my goals' : 0.73
'struggling to achieve my goals' : -0.67
'struggling to survive' : -0.52
'unable to survive' : -0.57
'this car runs on gasoline with lead' : -0.03
'this car runs on gasoline and it will lead us' : 0.09
'this politician runs for office and he will lead us' : 0.58
```

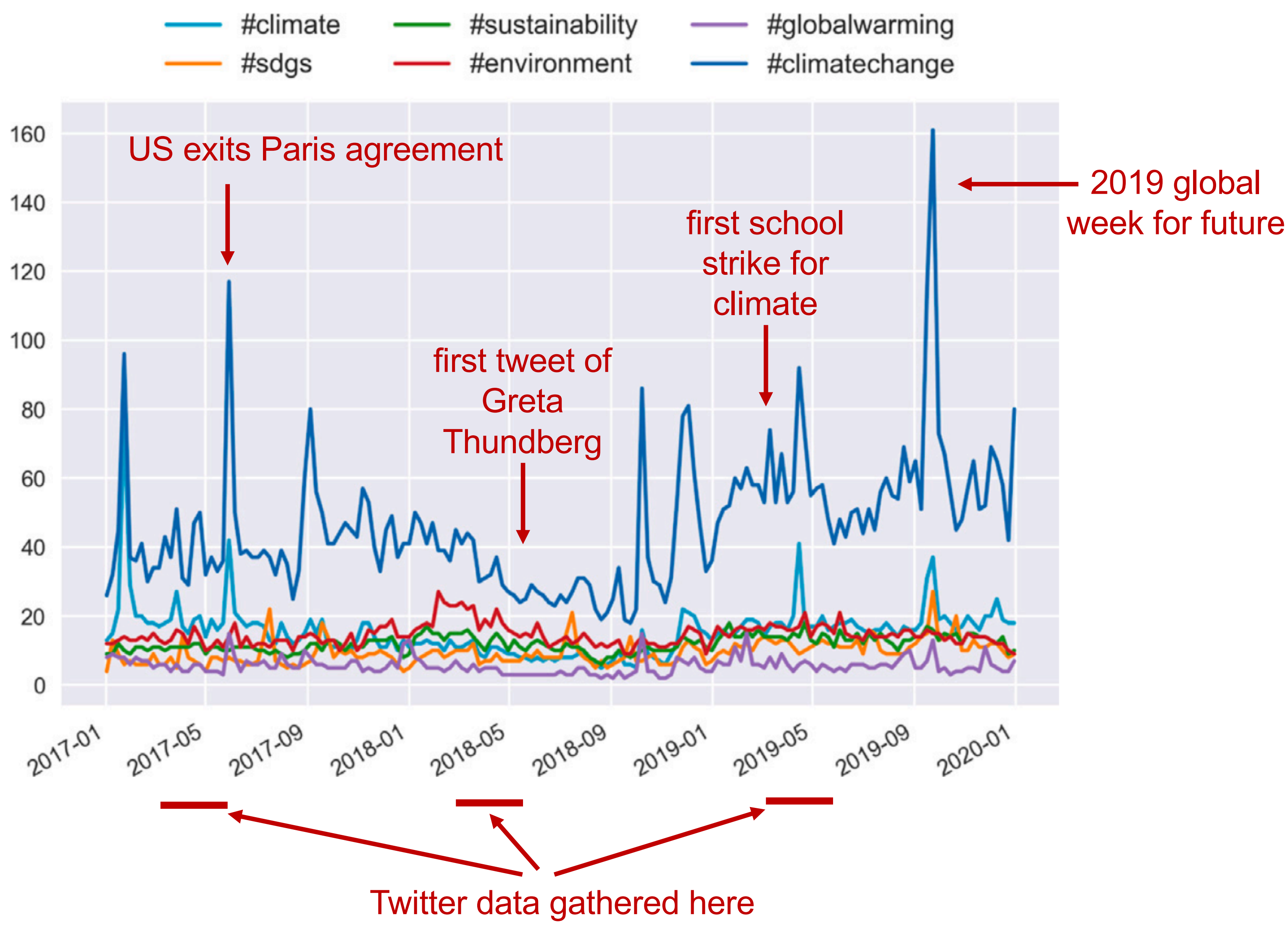
```
# assign agency
vals = ba0.predict(sents)
# print results
for item in zip(sents, vals):
    print(f"  {item[0]!r} : {item[1]:.2f}")
```

run BERTAgent

Using sentiment analysis

an overview on how it can be useful in your projects

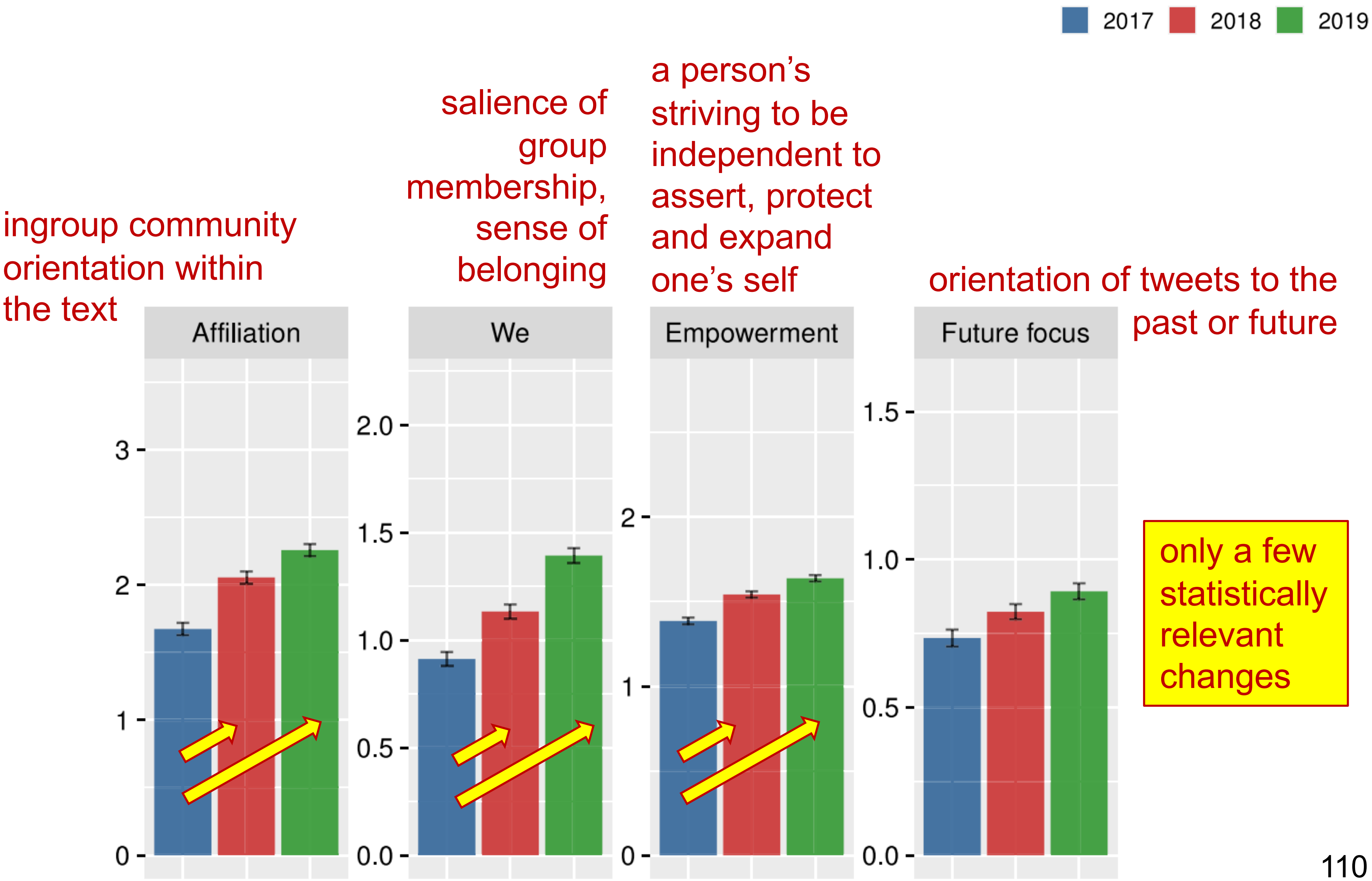
#climateaction
on Twitter in 2017, 2018, 2019



UNIVERSITÀ DEGLI STUDI DI PADOVA

Socio-psychological linguistic markers

a view on the entire tweets corpus



only a few statistically relevant changes

Statistically relevant increase
by Student/Welch t-test

Student's *t*-test

Article Talk

From Wikipedia, the free encyclopedia

A ***t*-test** is a type of statistical analysis used to compare the averages of two groups and determine whether the differences between them are more likely to arise from random chance. It is any statistical hypothesis test in which the test statistic follows a Student's *t*-distribution under the null hypothesis. It is most commonly applied when the test statistic would follow a normal distribution if the value of a scaling term in the test statistic were known (typically, the scaling term is unknown and is therefore a nuisance parameter). When the scaling term is estimated based on the data, the test statistic—under certain conditions—follows a Student's *t* distribution. The *t*-test's most common application is to test whether the means of two populations are different.

Student/Welch t-test

hypothesis and t value

Assumption:

the average values of the two populations being compared should follow a normal distribution (e.g., with many samples)

Hypothesis (to be tested):

H0 - the two sets have equal average value, $m_x = m_y$

H1 - the two sets have different average value, $m_x \neq m_y$

sample averages

Student's t distribution with ν degrees of freedom (an approximation under H0)

Test statistic: $t = \dfrac{\bar{x} - \bar{y}}{\sqrt{\dfrac{s_x^2}{N_x} + \dfrac{s_y^2}{N_y}}} \sim t_\nu$

$$\hat{\nu} = \frac{\left(\dfrac{s_x^2}{N_x} + \dfrac{s_y^2}{N_y}\right)^2}{\dfrac{\left(s_x^2/N_x\right)^2}{N_x - 1} + \dfrac{\left(s_y^2/N_y\right)^2}{N_y - 1}}$$

unbiased estimator of variance

samples sizes

estimate of ν (degrees of freedom) used for calculations

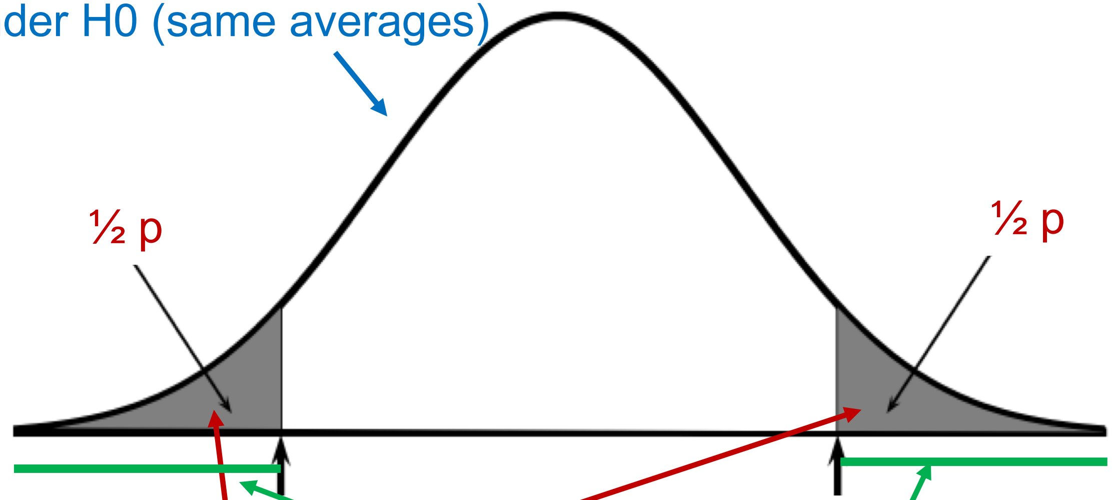
Student/Welch t-test

p value = probability of error under H0

we declare different average values H1

p-value	evidence
$< .01$	very strong evidence against H_0
$.01 - .05$	strong evidence against H_0
$.05 - .10$	weak evidence against H_0
$> .1$	little or no evidence against H_0

Student-t PDF with ν degrees of freedom under H0 (same averages)



probability p that, by confirming H1, we are making errors on H0

(conservative) region choice to declare H1 from the t value

the *p* value informs on whether a difference exists (statistically)

it is a false positive rate

Cohen's d-value

evaluating effect sizes

$$\text{Cohen's } d = \frac{\bar{x}-\bar{y}}{\sqrt{a\, s_x^2+(1-a)\, s_y^2}} \qquad a = \frac{N_x-1}{N_x+N_y-2}$$

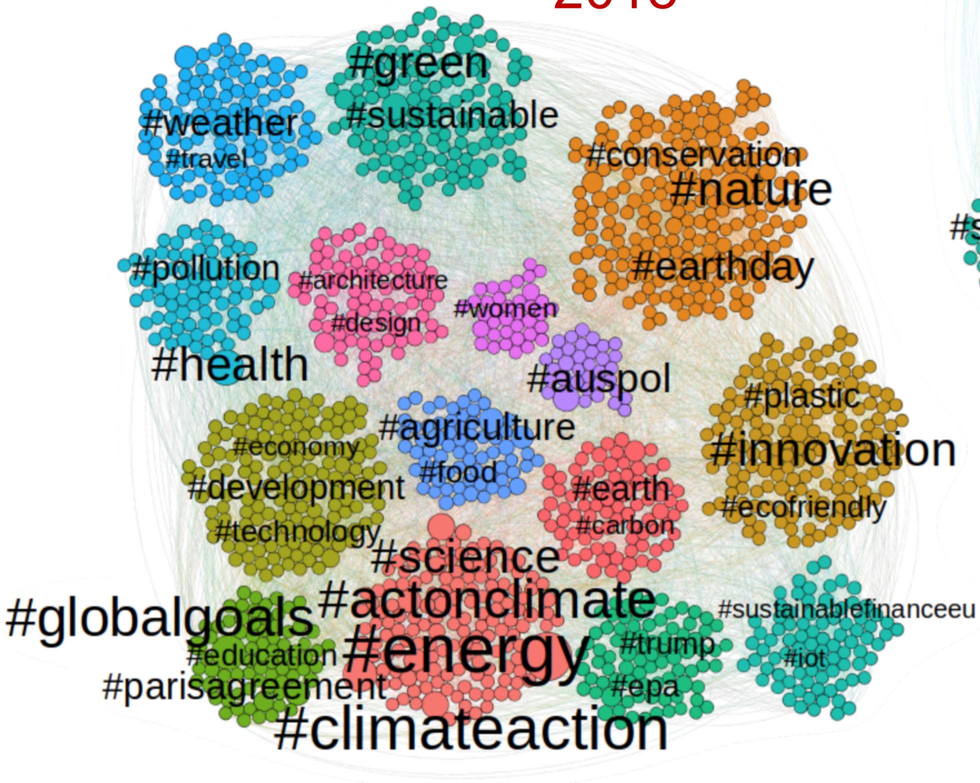
Relative size	Effect size
	0.0
Small	0.2
Medium	0.5
Large	0.8
	1.4

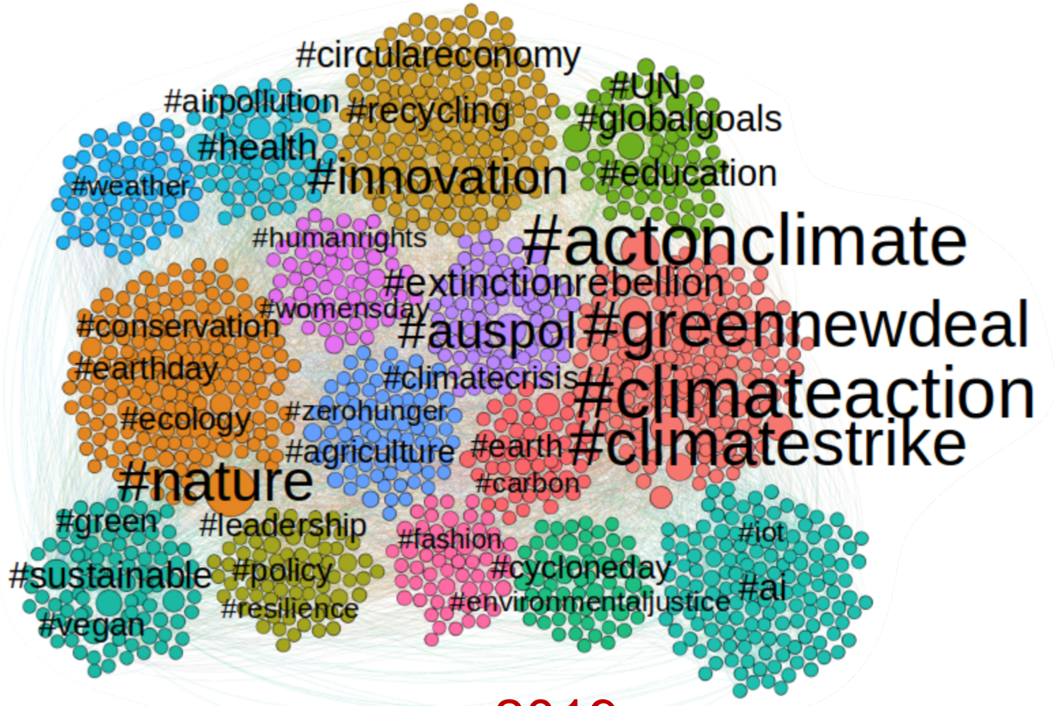
we confirm H1

the *d* value informs on the size of the effect

Topics in #climateaction
on Twitter in 2017, 2018, 2019

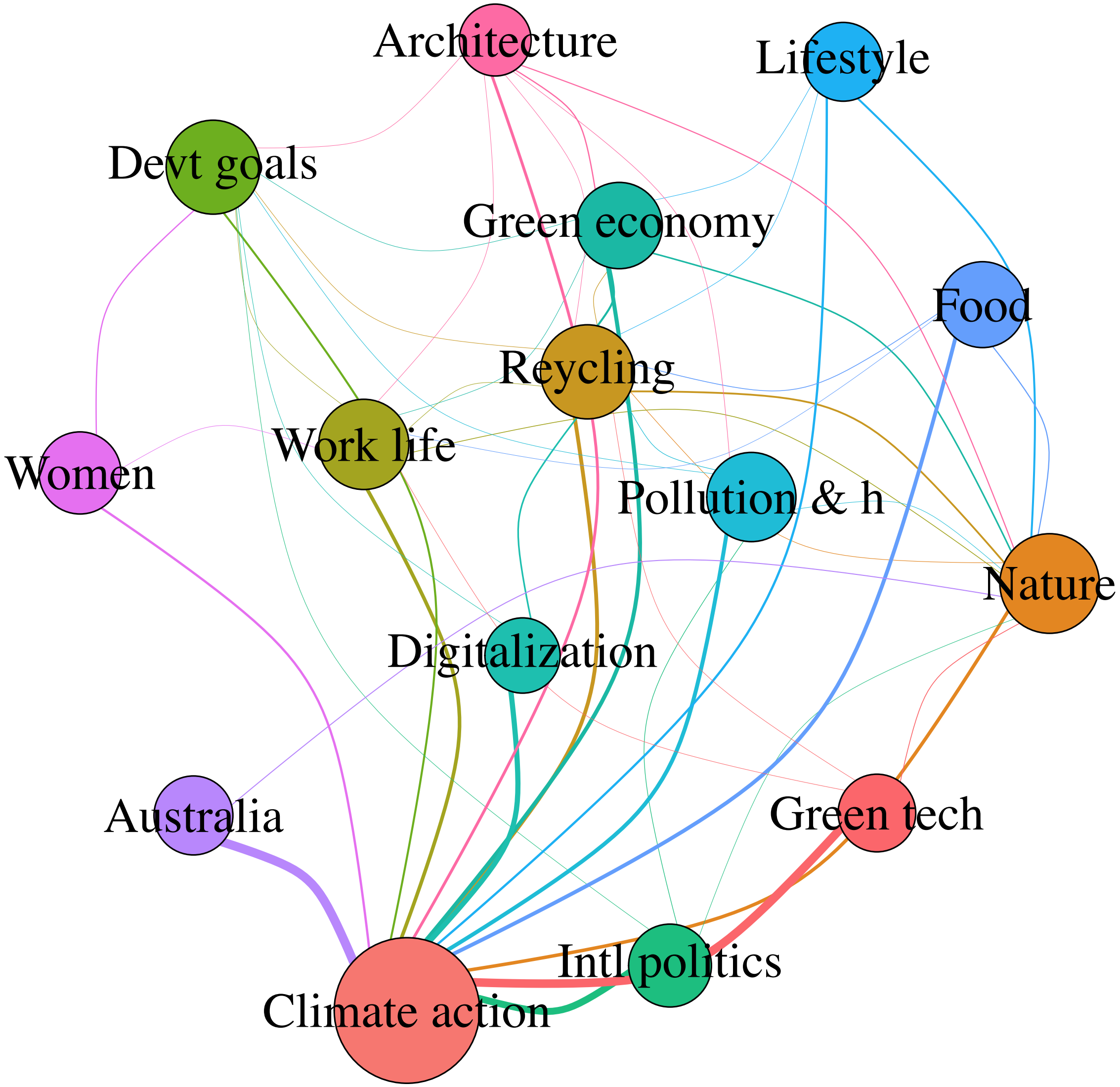
2018





2019

Topics interdependencies

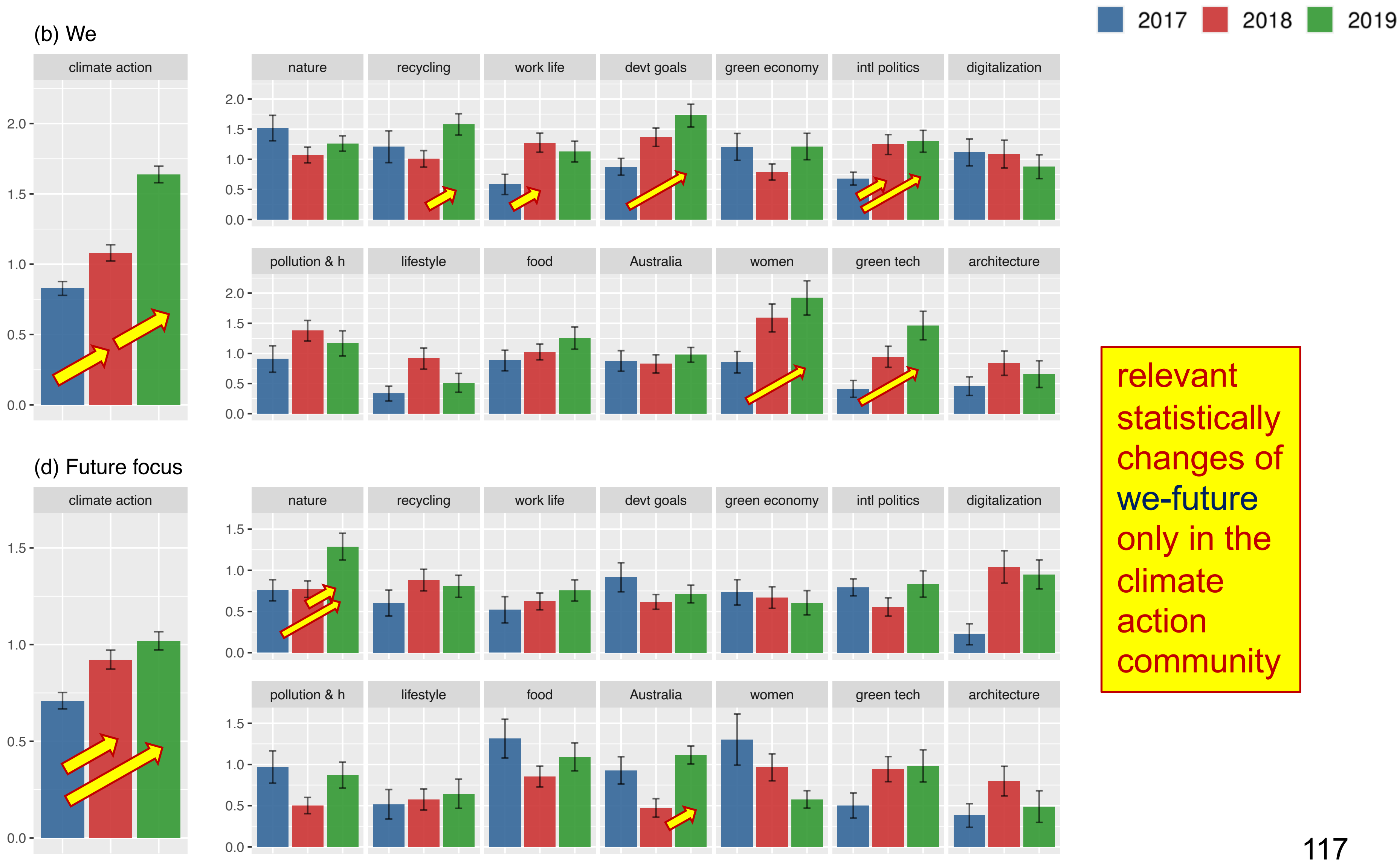


projecting the adjacency matrix on topics

P_{11}	P_{12}	P_{13}
P_{21}	P_{22}	P_{23}
P_{31}	P_{32}	P_{33}

Socio-psychological linguistic markers

a view inside topics



relevant statistically changes of we-future only in the climate action community

Wrap-up

on topic detection

Takeaways

for your projects

- What available tools should be used
  - Louvain & BERTopic
  - compare their performance through NMI, modularity, etc.
  - LIWC & BERTAgent
  - to enrich your analysis under a socio-psycological lens
- What available tools should NOT be used
  - InfoMap, NMF & LDA
  - they show poor performance
- What would be nice to see implemented
  - soft Louvain made fast
  - performance of BigCLAM and SMBs
  - NFTM VAE and its performance