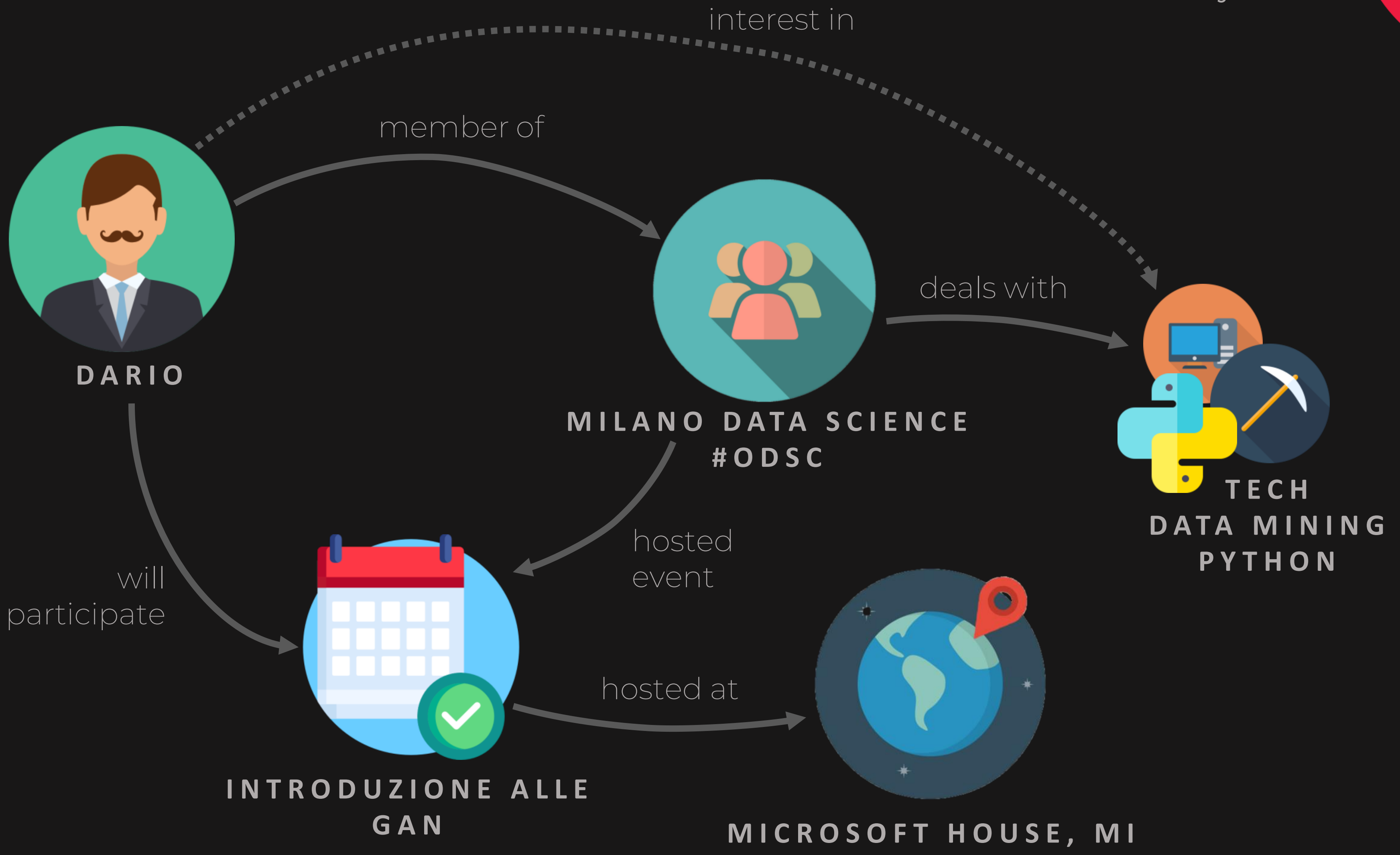
MEETOLOGY

A DATA SEMANTIC PROJECT

DARIO BERTAZIOLI
FABRIZIO D'INTINOSANTE
MASSIMILIANO PERLETTI
UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

INTRODUCTION

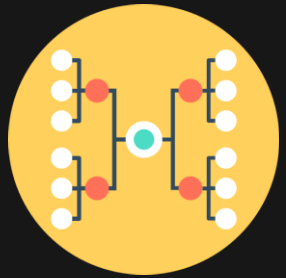
available in **186** countries
40 millions users
320k active groups
12k daily events



OVERVIEW

ontology definition
(using JSON-LD format)
S. J. Chalk, (2016)

messages enrichment
(weather and demographic data)



topic categorization

A. K. Uysal and S. Gunal, (2014)

tf-idf

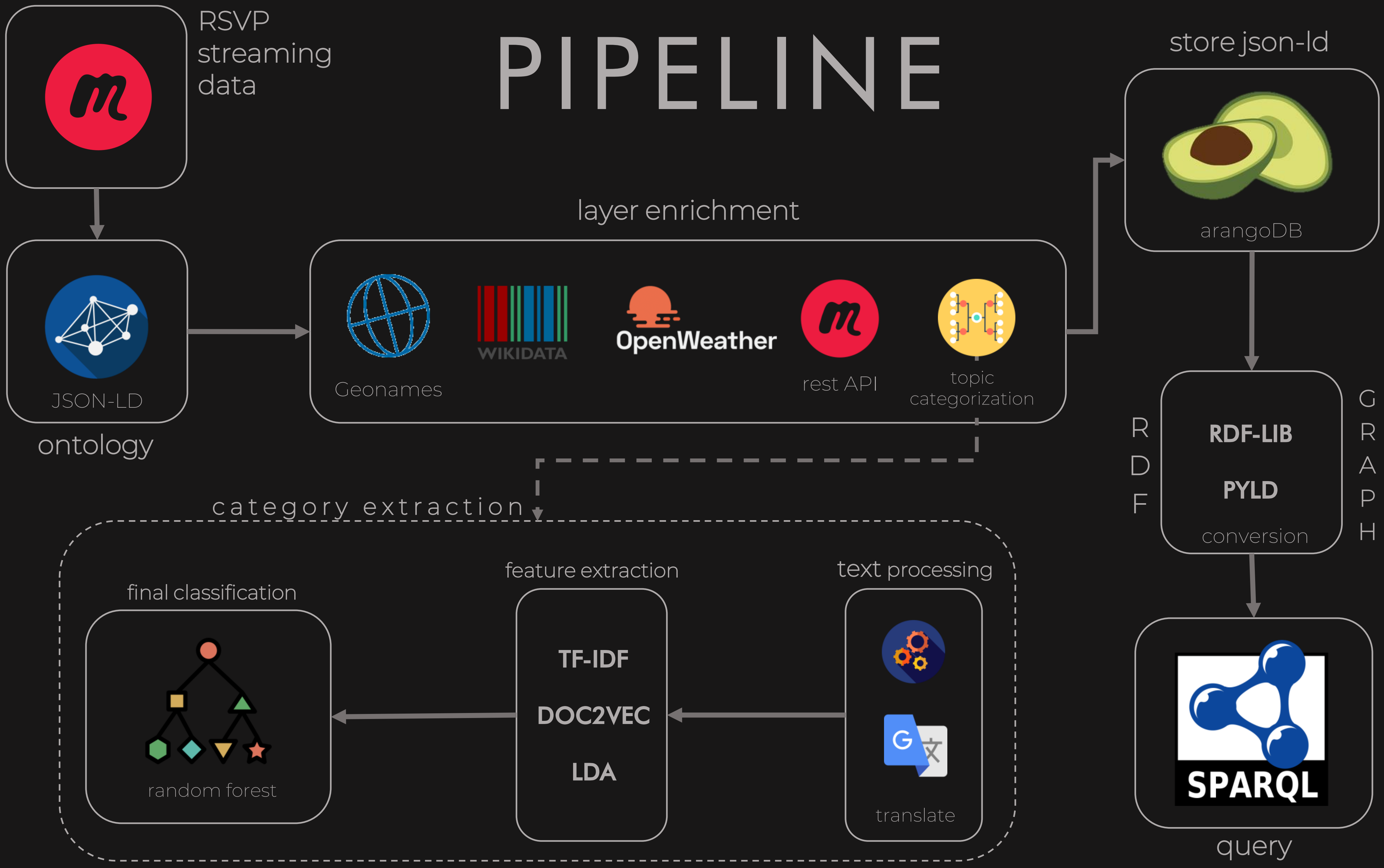
doc2vec

Q. Le and T. Mikolov, (2014)

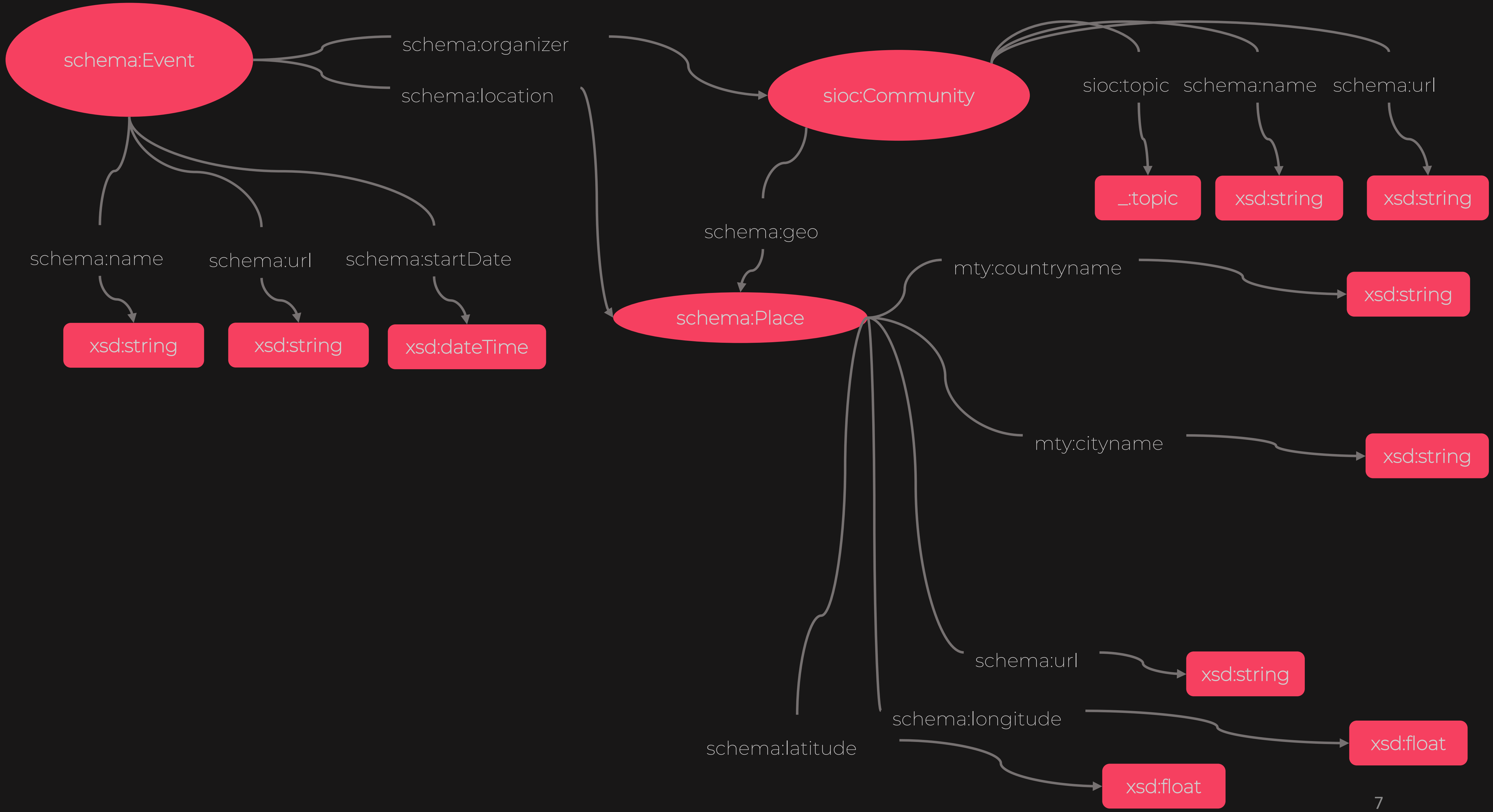
LDA

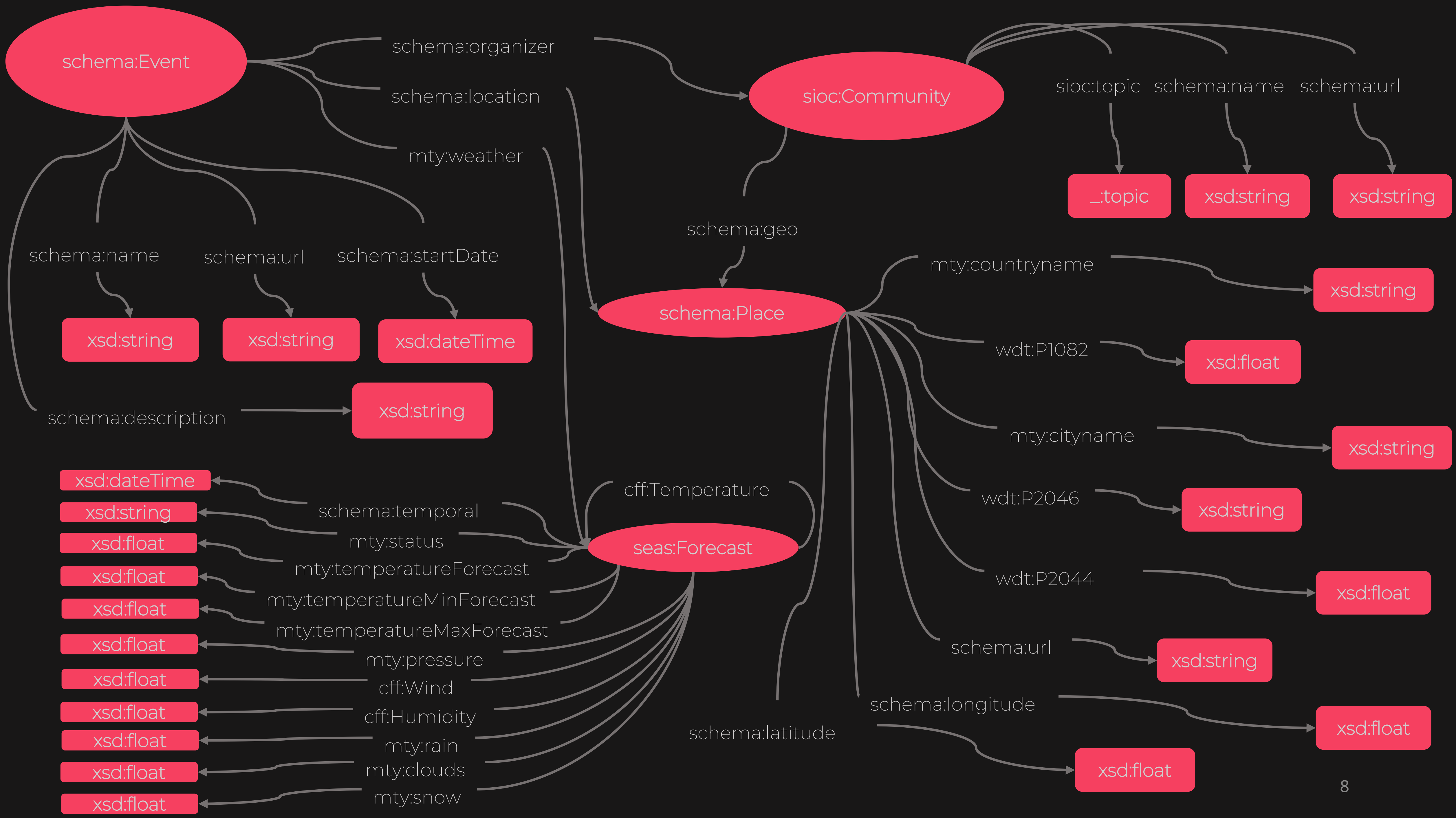
D. M. Blei, A. Y. Ng, M. I. Jordan, (2003)

PIPELINE


RSVP
streaming
data
JSON-LD
ontology
layer enrichment
Geonames
WIKIDATA
OpenWeather
rest API
topic
categorization
store json-ld
arangoDB
RDF-LIB
PYLD
conversion
RDF
GRAPH
SPARQL
query
category extraction
text processing
translate
feature extraction
TF-IDF
DOC2VEC
LDA
final classification
random forest

ONTOLOGY DEFINITION


schema:Event
schema:organizer
schema:location
sioc:Community
sioc:topic
schema:name
schema:url
_:topic
xsd:string
xsd:string
schema:geo
schema:name
schema:url
schema:startDate
xsd:string
xsd:string
xsd:dateTime
schema:Place
mty:countryname
xsd:string
mty:cityname
xsd:string
schema:url
xsd:string
schema:longitude
xsd:float
schema:latitude
xsd:float


schema:Event
schema:organizer
schema:location
mty:weather
sioc:Community
sioc:topic
schema:name
schema:url
_:topic
xsd:string
xsd:string
schema:name
schema:url
schema:startDate
xsd:string
xsd:string
xsd:dateTime
schema:description
xsd:string
schema:geo
schema:Place
mty:countryname
xsd:string
wdt:P1082
xsd:float
mty:cityname
xsd:string
wdt:P2046
xsd:string
wdt:P2044
xsd:float
schema:url
xsd:string
schema:longitude
xsd:float
schema:latitude
xsd:float
cff:Temperature
seas:Forecast
schema:temporal
xsd:dateTime
mty:status
xsd:string
mty:temperatureForecast
xsd:float
mty:temperatureMinForecast
xsd:float
mty:temperatureMaxForecast
xsd:float
mty:pressure
xsd:float
cff:Wind
xsd:float
cff:Humidity
xsd:float
mty:rain
xsd:float
mty:clouds
xsd:float
mty:snow
xsd:float

{
"venue": {
"venue_name": "Phil Foster Park",
"lon": -80.041481,
"lat": 26.784731,
"venue_id": 13027532
},
"visibility": "public",
"response": "yes",
"guests": 0,
"member":{....}

add context

▼ @context {43}
@version : 1.1
schema : https://schema.org/
mty : https://www.meetology.onto/
sc : http://rdfs.org/sioc/spec/#term_
wdt : https://www.wikidata.org/wiki/Property:
xsd : https://www.w3.org/2001/XMLSchema#
cff : https://www.w3.org/2005/Incubator/ssn/ssnx/cf/cf-feature#
seas : http://w3id.org/seas/

format JSON-LD

https://www.w3.org/2018/jsonld-cg-reports/json-ld/
Stuart J. Chalk, (2016). "SciData: a data model and ontology for semantic representation of scientific data", Journal of Cheminformatics, 8 (1), 1.

ENRICHMENT

"group": {
"group_topics": [...],
"group_city": “New York",
"group_country": "us",
"group_id": 7764982,
"group_name": “USA Divers",
"group_lon": -73.99,
"group_urlname": "BeachDivers",
"group_state": "FL",
"group_lat": 40.69}

query geonames docker
indexed with ElasticSearch

```
payload = '{"query": {"bool": \
                    {"minimum_should_match": 1,"should": \
                     [{"match": {"name": {"query":"'+name+'"}}}, \
                      {"match": {"alternatenames":"'+name+'"}}, \
                      {"match": {"asciiname": {"query":"'+name+'"} }}], \
                     "filter": [{"term": {"fclass":"A" }}, \
                               {"geo_distance" : \
                                {"distance" :"5km","location" :{"lat" : '+str(lat)+', "lon" : '+str(lon)+'}}}]}}}}'
```

messages enriched
with geonameid &
population

messages enriched with
geonameid & population

query wikidata endpoint
using SPARQLWrapper

```
query = """SELECT ?city ?pop ?area ?elevation WHERE {
  ?city wdt:P1566 """+'"'+str(geonameid)+'"'+""".
  OPTIONAL { ?city wdt:P1082 ?pop. }
  OPTIONAL { ?city wdt:P2046 ?area. }
  OPTIONAL { ?city wdt:P2044 ?elevation. }
}
limit 1"""
```

WIKIDATA

geographical data
(elevation & area)

enriched messages with
geographical data

query OpenWeather API
using temporal and spatial information

OpenWeather

```
if message['startDate'] <= time.time()*1000 + 388800000 and  message['startDate'] >= time.time()*1000 + 10800000:
    message['weather'] = {"@type":"Forecast"} # init section
    message['weather']['forecastDate'] = dt.utcnow().strftime('%Y-%m-%d %H:%M:%S+00')
    try:
        obs_fs = owm.three_hours_forecast_at_coords(message['organizer']['geo']['latitude'],
                                                    message['organizer']['geo']['longitude'])
        w = obs_fs.get_weather_at(str(dt.fromtimestamp(message['startDate']/1000))+'+00')
```

enriched message with
weather forecast

enriched message with
weather forecast

query Meetup rest API
in order to get events description

```
request = requests.get("http://api.meetup.com/2/events", params = params)

data = request.json()
#print(data)
desc = data["results"][0]["description"]
soup = BeautifulSoup(desc, "lxml")
clean = soup.get_text()
message['description'] = clean
message['@context']['description']['@language'] = detect(clean)
```

end of enrichment pipeline

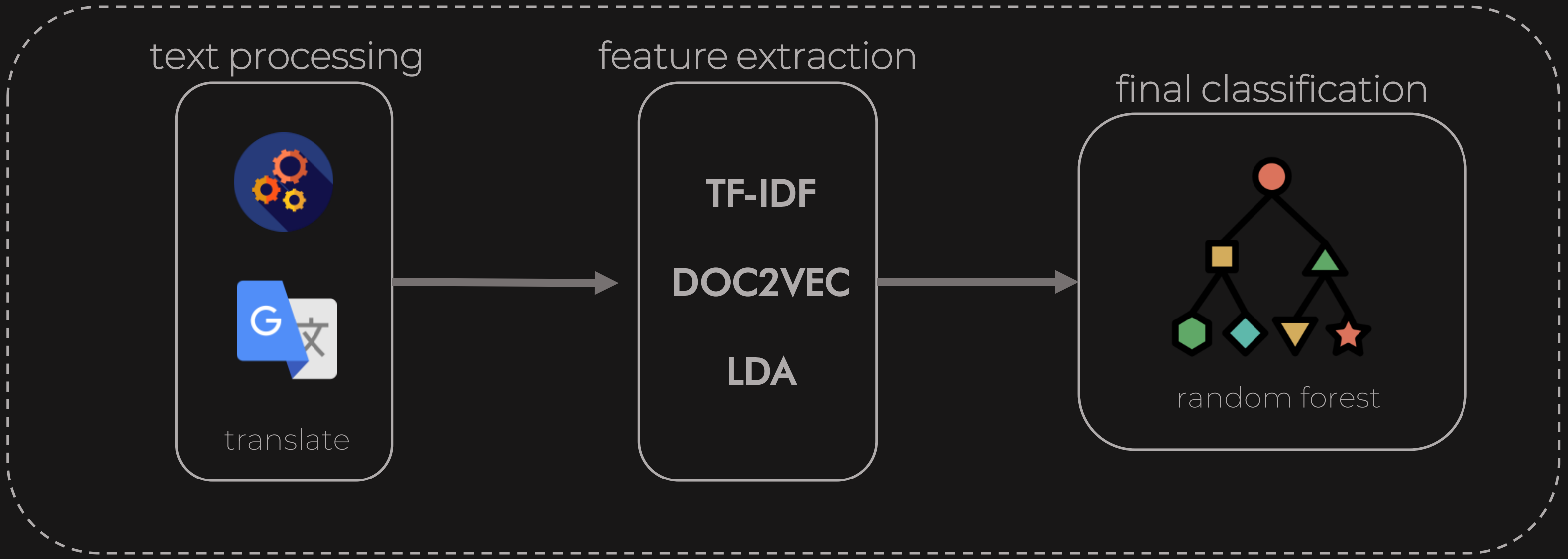
ArangoDB as storage
for enriched messages parsed into JSON-LD

TOPIC GENERALIZATION

GENERAL STRATEGY

from event descriptions predict the event category

(macro-topic)



TEXT 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

stemmization

- SnowballStemmer (multilingual)
- PorterStemmer (English)

A. Uysal and S. Gunal, (2014). "The impact of preprocessing on text classification", Information Processing & Management, 50 (1), 104-112.

FIRST APPROACH: BAG OF WORDS

feature extraction by tf-idf vectorization

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

where

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

classification

multiclass problem

33 classes, some of them slightly unbalanced

performance measure (in terms of average F-measure)

- initially (no cleansing) ~55% (still better than a zero rule!)
- after text preprocessing ~68%
- after some tricks ~75% (more on this later!)

issues

td-idf method requires a collection for the vectorization to work well → not suitable for prediction on the fly

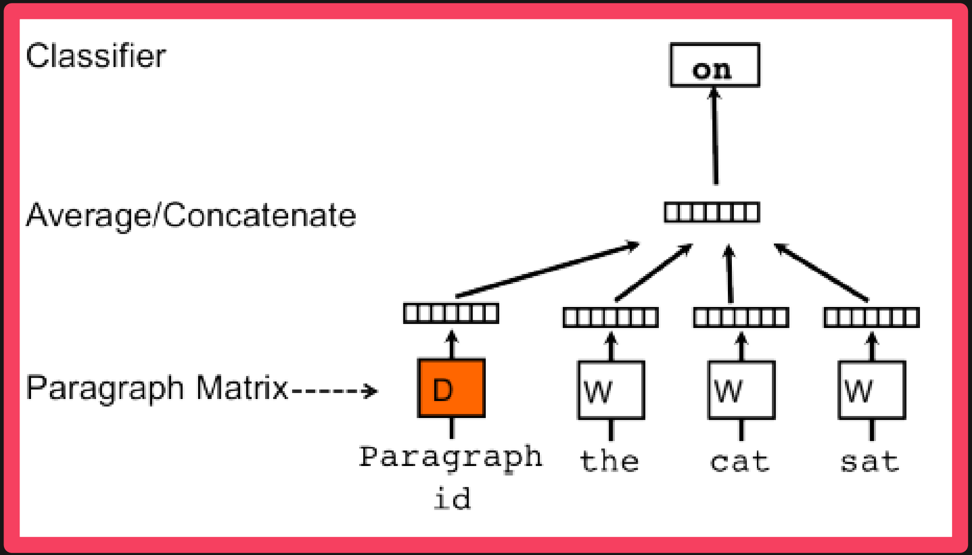
D. Xue and F. Li, (2015). "Research of Text Categorization Model based on Random Forests", IEEE International Conference on Computational Intelligence & Communication Technology, 173-176.

PARAGRAPH VECTOR (Doc2Vec)

feature extraction: doc vectorization

- train a (gensim) doc2vec model (10 epochs, 300-dim vec)
- dbow and dm concatenation

	precision	recall	f1-score	support
LGBT	0.99	0.35	0.52	463
alternative lifestyle	0.97	0.78	0.86	223
book clubs	0.68	0.45	0.54	388
career business	0.70	0.83	0.76	4580
cars motorcycles	0.98	0.50	0.66	362
community environment	0.82	0.46	0.59	944
dancing	0.82	0.81	0.82	1090
education learning	0.79	0.48	0.60	1003
fashion beauty	0.71	0.40	0.51	81
fine arts culture	0.82	0.60	0.69	1027
fitness	0.79	0.69	0.74	2269
food drink	0.75	0.56	0.64	1424
games	0.85	0.82	0.83	1469
health wellbeing	0.71	0.81	0.75	4532
hobbies crafts	0.80	0.60	0.68	515
language ethnic identity	0.82	0.66	0.73	1773
movements politics	0.83	0.78	0.80	752
movies film	0.75	0.45	0.56	539
music	0.79	0.61	0.69	895
new age spirituality	0.78	0.74	0.76	3001
outdoors adventure	0.74	0.86	0.79	5385
paranormal	0.60	0.13	0.21	23
parents family	0.89	0.50	0.64	640
pets animals	0.92	0.45	0.61	312
photography	0.95	0.76	0.84	584
religion beliefs	0.84	0.49	0.62	777
sci-fi fantasy	0.86	0.32	0.47	287
singles	0.61	0.32	0.42	844
socializing	0.47	0.73	0.57	5125
sports recreation	0.84	0.83	0.84	2470
support	0.93	0.38	0.54	431
tech	0.82	0.87	0.84	4340
writing	0.90	0.65	0.76	387
accuracy			0.72	48935
macro avg	0.80	0.60	0.66	48935
weighted avg	0.75	0.72	0.72	48935



performances (avg F-measure)

- initially → ~ 60%
- after processing → ~ 68%
- after some tricks → ~ 72% (more on this later!)

Le Q. and Mikolov T. , (2014). "Distributed Representations of Sentences and Documents", Proceedings of the 31st International Conference on Machine Learning, in PMLR, 32(2), 1188-1196.

PARAGRAPH VECTOR (Doc2Vec)

PROS

predict on the fly
semantically aware

CONS

a lot of data is needed

(as well as computational time once data is granted)

overfitting in case of small/medium dataset

Le Q. and Mikolov T. , (2014). "Distributed Representations of Sentences and Documents", Proceedings of the 31st International Conference on Machine Learning, in PMLR, 32(2), 1188-1196.

LATENT DIRICHLET ALLOCATION (LDA)

unsupervised method for topic modeling & topic extraction

- generative model (three level bayesian model)

high level idea

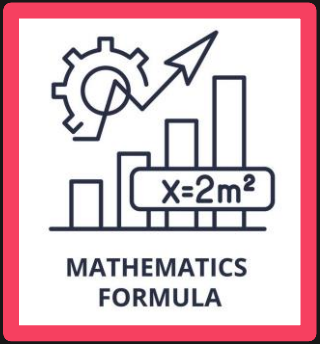
assume your texts comes from a latent-topics generated distribution, try to infer the distribution parameters (thus, the latent topics)

(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 → ~ 5/7% performance gain by only stripping ~20 badwords (the previously introduced “trick”)

David M. Blei, Andrew Y. Ng, Michael I. Jordan, (2003). "Latent dirichlet allocation", The Journal of Machine Learning Research, 3, 993-1022.

LATENT DIRICHLET ALLOCATION (LDA)

unsupervised method for topic m

- generative model (three l

high level idea

assume your texts comes from a laten
infer the distribution parameters (thu

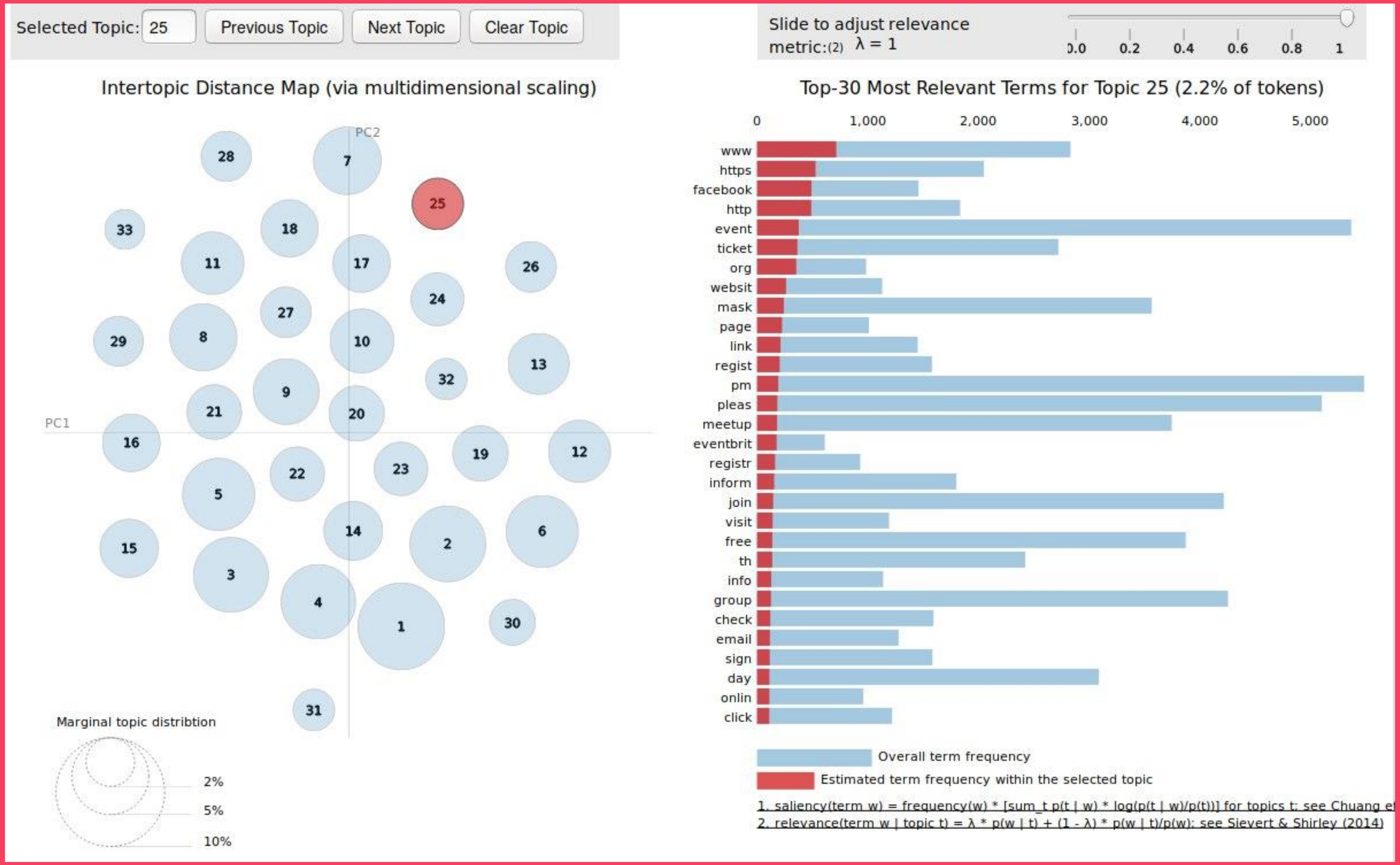
(almost) technically

- "LDA takes de Finetti theorem
- compute the probability distr

  exploiting the m

- use Bayesian inference to ob

  exploiting variati



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 → ~ 5/7% performance gain by only stripping ~20 badwords (the previously introduced "trick")

David M. Blei, Andrew Y. Ng, Michael I. Jordan, (2003). "Latent dirichlet allocation", The Journal of Machine Learning Research, 3, 993-1022.

RDF PARSING

2 RDF parsing
distinct strategies
(directly implemented in python)

rdflib
new plugin for JSON-LD

PROS
easy to use, good performance

CONS
doesn't allow to use @base attribute

pyld
for parsing into RDF

PROS
allow to use @base attribute

CONS
not fully compatible with rdflib

SPARQL query example

```
result_more_query = rdf_lib.query(
    """
        SELECT ?name ?cat ?country
        WHERE{
            ?event schema:name ?name;
            schema:organizer ?org;
            schema:category ?cat FILTER ( str(?cat) = "tech" ).
            ?org schema:geo ?geo.
            ?geo mty:countryname ?country  FILTER ( str(?country) = "gb" ).
        }
    """
)

for row in result_more_query:
    print("%s ---Has Category---> %s --Country-->%s" % row)
```

```
#NottsTest - TBA ---Has Category---> tech --Country-->gb
Google AI Workshop: Machine learning with Tensorflow ---Has Category---> tech --Country-->gb
Hands-on Meetup: Create a design system in React with Styled System -TICKET only ---Has Category---> tech --Country-->gb
Reading SEO- On-Page Technical, Power of Information for SEO & Internal Search  ---Has Category---> tech --Country-->gb
Emily Jiang- On Stage Hacking: Build 12-Factor Microservices in an Hour ---Has Category---> tech --Country-->gb
Community meetup ---Has Category---> tech --Country-->gb
Agile HR Massive Morning Meetup | London | Fri 13th Sept ---Has Category---> tech --Country-->gb
 Marc Gravell - gRPC in .NET - Hosted by IRESS ---Has Category---> tech --Country-->gb
July Gophers ---Has Category---> tech --Country-->gb
Meet Amazing Data Women - Summer Social! ---Has Category---> tech --Country-->gb
Keynote by Angela Yu on Why I'm Building My Next App in Flutter ---Has Category---> tech --Country-->gb
London #PowerBI user group with Will Thompson ---Has Category---> tech --Country-->gb
Foolproof design lab  ---Has Category---> tech --Country-->gb
Cyber Nottingham - September Meetup ---Has Category---> tech --Country-->gb
Agile team development at Co-op ---Has Category---> tech --Country-->gb
Introduction to Time-Series ---Has Category---> tech --Country-->gb
OWASP Birmingham Chapter Meetup July 2019 ---Has Category---> tech --Country-->gb
London Scala Workshop: Zainab Ali - Run scalac, run! ---Has Category---> tech --Country-->gb
Hi-tech for high profit: Digital twins, geospatial data and property investment. ---Has Category---> tech --Country-->gb
Wireframing & prototyping using Axure ---Has Category---> tech --Country-->gb
```

CONCLUSIONS

KEY POINTS

- acquires streaming data from Meetup (RSVP messages)
- attaches a semantic structure (by ontology definition)
- integrates data with useful information
- stores the result on ArangoDB
- creates an RDF graph for SPARQL query

IMPROVEMENTS & FUTURE WORK

data from other services,
more enrichment

in case use business
solution for API

improve classification
performance

this work might be the skelethon of a future project in which after collecting data through the illustrated pipeline, tries to predict (linear regression) the event participation on spatial and temporal bases

THANK YOU

REFERENCES

- https://www.w3.org/2018/jsonld-cg-reports/json-ld/

- Stuart J. Chalk, (2016). "SciData: a data model and ontology for semantic representation of scientific data", Journal of Cheminformatics, 8 (1), 1.

- Alper Kursat Uysal and Serkan Gunal, (2014). "The impact of preprocessing on text classification", Information Processing & Management, 50 (1), 104-112, ISSN 0306-4573.

- D. Xue and F. Li, (2015). "Research of Text Categorization Model based on Random Forests", 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, 173-176.

- David M. Blei, Andrew Y. Ng, Michael I. Jordan, (2003). "Latent dirichlet allocation", The Journal of Machine Learning Research, 3, 993-1022.

- Le Q. and Mikolov T. , (2014). "Distributed Representations of Sentences and Documents", Proceedings of the 31st International Conference on Machine Learning, in PMLR, 32(2), 1188-1196.

Thanks to Vanessa Grass for the meetup labelled data

Repository available on gitlab (complete) and github (reduced).

(exchangeability)

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

By de Finetti's theorem:

$$p(\mathbf{w}, \mathbf{z}) = \int p(\theta) \left(\prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n) \right) d\theta$$

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

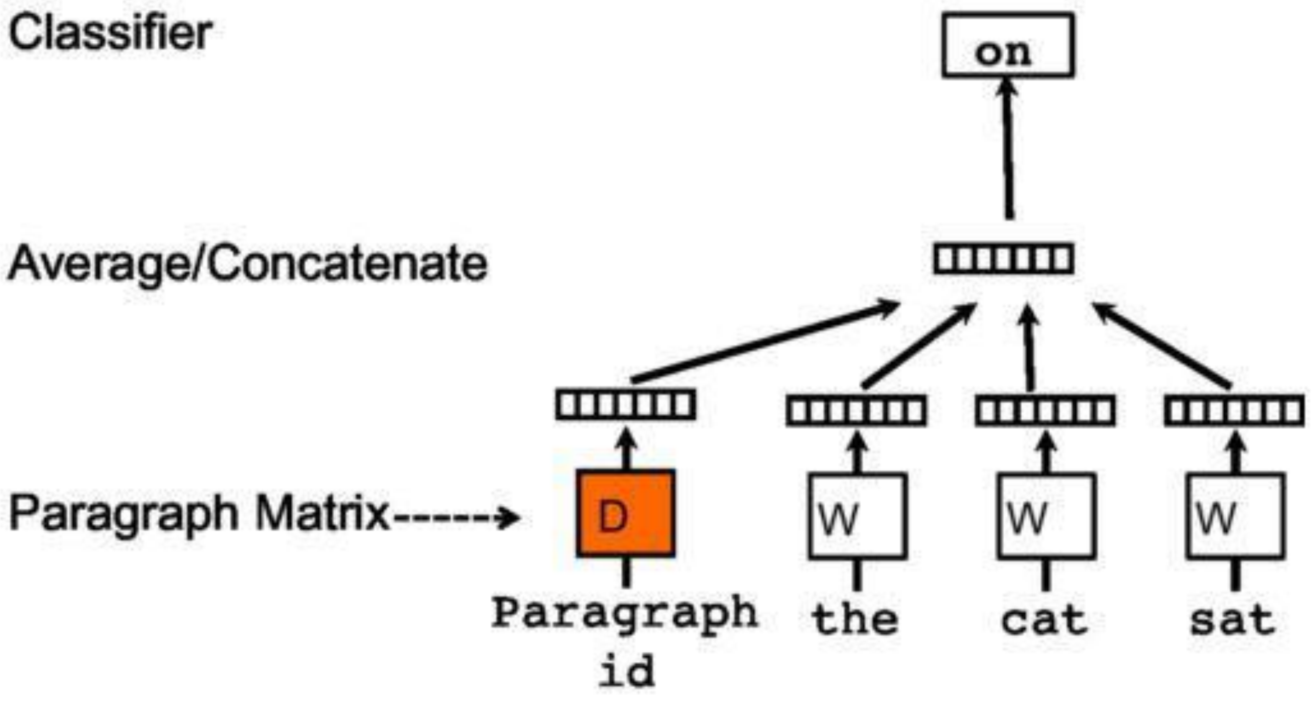
probability of the entire corpus

Inference:

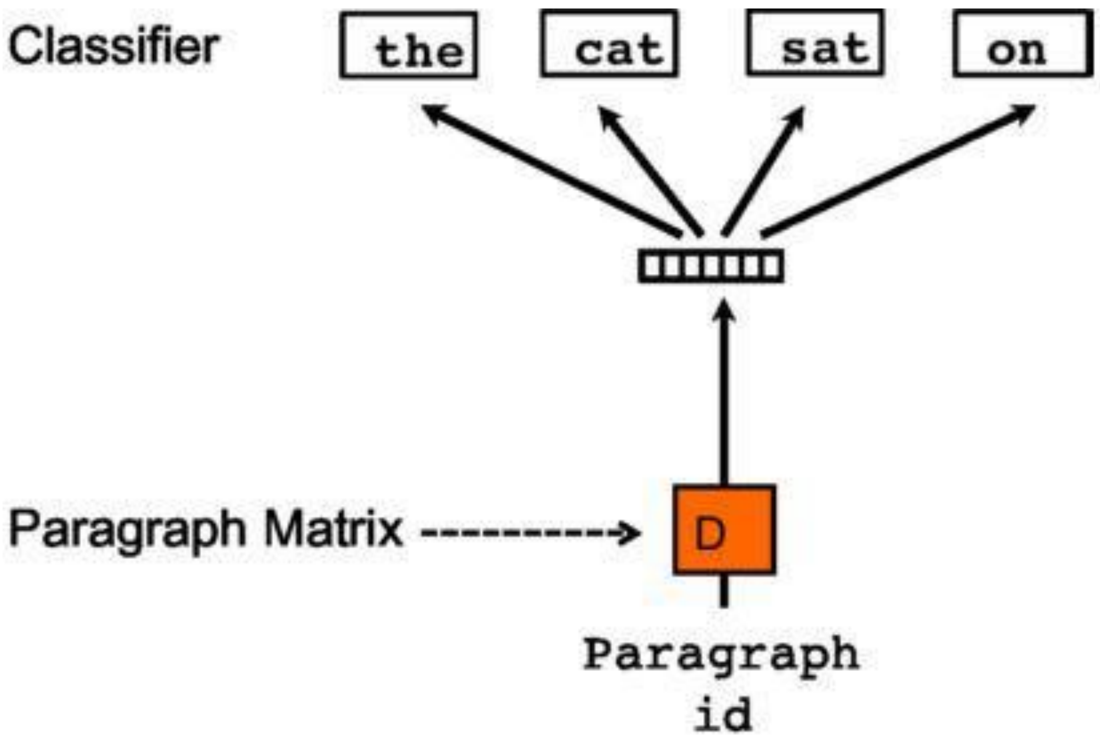
$$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)}{p(\mathbf{w} \mid \alpha, \beta)}$$

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

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



Paragraph vector: distributed memory



Paragraph vector: distributed bag of words