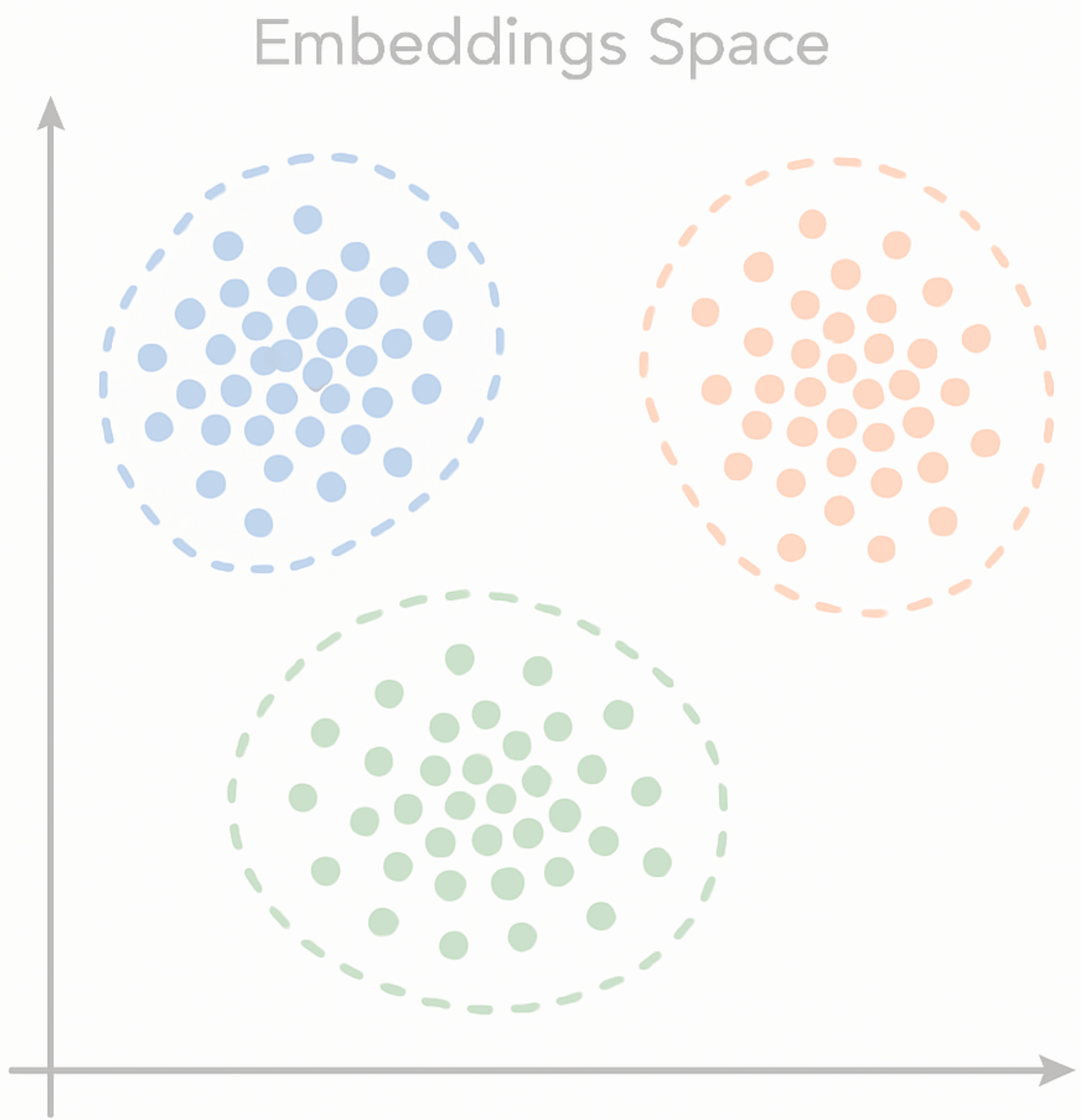
Lezione n.5

Laboratorio su Clustering e Topic Modeling con modelli embeddings e riduzione della dimensionalità

Luigi Di Caro

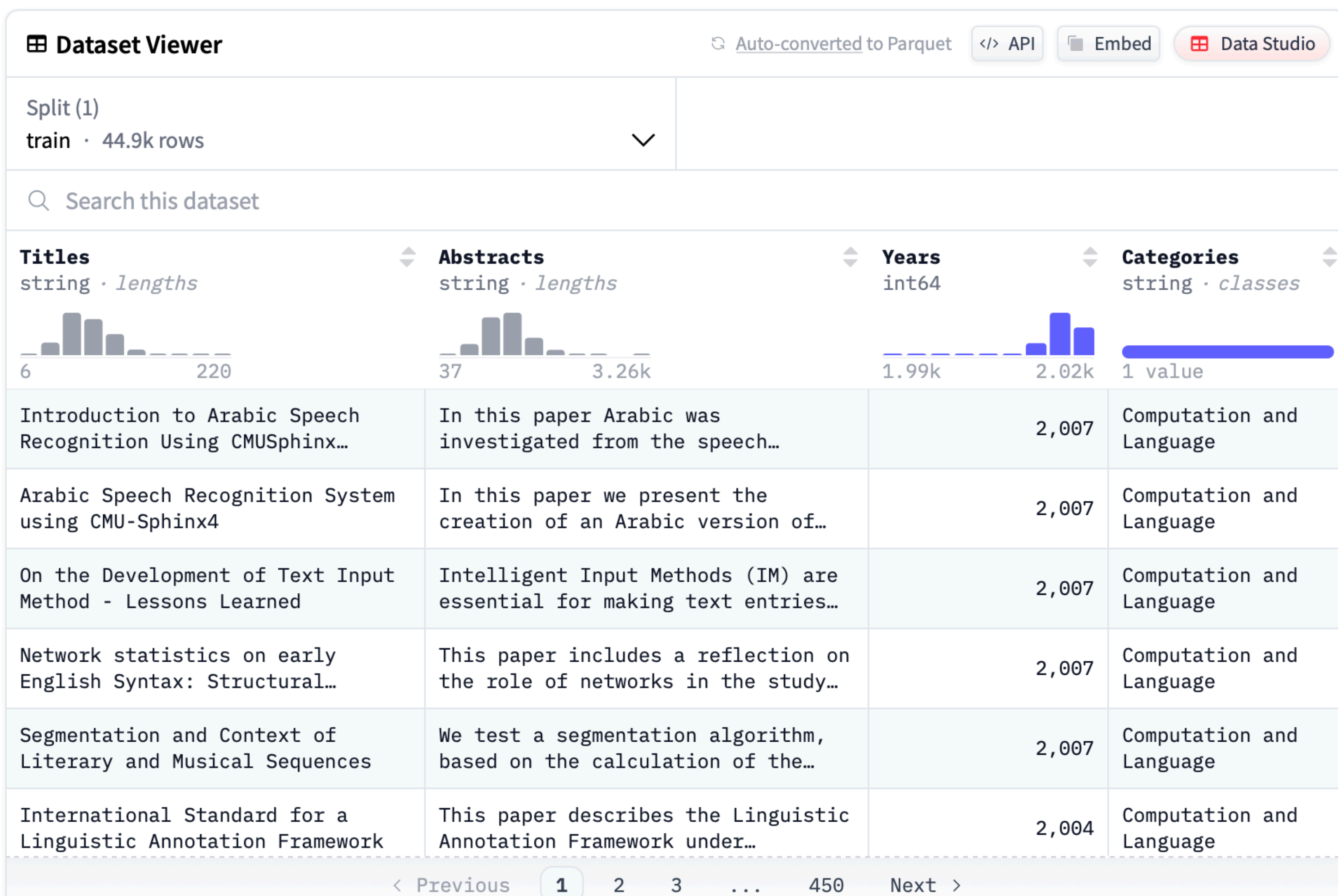
Panoramica

- Clustering con embeddings
- Topic Modeling con BERTopic



Dataset

- Come dataset utilizzeremo circa 45K articoli scientifici presi da un open repository, **ArXiv**
  - Link: https://huggingface.co/datasets/MaartenGr/arxiv_nlp



Dataset Viewer

Auto-converted to Parquet | API | Embed | Data Studio

Split (1)
train · 44.9k rows

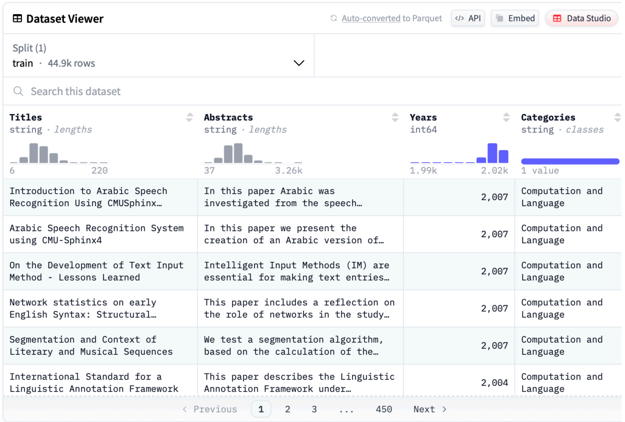
Search this dataset

Titles string · lengths 6 – 220	Abstracts string · lengths 37 – 3.26k	Years int64 1.99k – 2.02k	Categories string · classes 1 value
Introduction to Arabic Speech Recognition Using CMUSphinx…	In this paper Arabic was investigated from the speech…	2,007	Computation and Language
Arabic Speech Recognition System using CMU-Sphinx4	In this paper we present the creation of an Arabic version of…	2,007	Computation and Language
On the Development of Text Input Method - Lessons Learned	Intelligent Input Methods (IM) are essential for making text entries…	2,007	Computation and Language
Network statistics on early English Syntax: Structural…	This paper includes a reflection on the role of networks in the study…	2,007	Computation and Language
Segmentation and Context of Literary and Musical Sequences	We test a segmentation algorithm, based on the calculation of the…	2,007	Computation and Language
International Standard for a Linguistic Annotation Framework	This paper describes the Linguistic Annotation Framework under…	2,004	Computation and Language

‹ Previous 1 2 3 ... 450 Next ›

Dataset

- Come dataset utilizzeremo circa 45K articoli scientifici presi da un open repository, **ArXiv**
  - Link: https://huggingface.co/datasets/MaartenGr/arxiv_nlp



Titles (string · lengths, 6–220)	Abstracts (string · lengths, 37–3.26k)	Years (int64, 1.99k–2.02k)	Categories (string · classes, 1 value)
Introduction to Arabic Speech Recognition Using CMUSphinx…	In this paper Arabic was investigated from the speech…	2,007	Computation and Language
Arabic Speech Recognition System using CMU-Sphinx4	In this paper we present the creation of an Arabic version of…	2,007	Computation and Language
On the Development of Text Input Method - Lessons Learned	Intelligent Input Methods (IM) are essential for making text entries…	2,007	Computation and Language
Network statistics on early English Syntax: Structural…	This paper includes a reflection on the role of networks in the study…	2,007	Computation and Language
Segmentation and Context of Literary and Musical Sequences	We test a segmentation algorithm, based on the calculation of the…	2,007	Computation and Language
International Standard for a Linguistic Annotation Framework	This paper describes the Linguistic Annotation Framework under…	2,004	Computation and Language

< Previous 1 2 3 ... 450 Next >

```
from datasets import load_dataset
dataset = load_dataset(“maartengr/arxiv_nlp")["train"]

abstracts = dataset["Abstracts"]
titles = dataset["Titles"]
```

```
titles[1:10]


[‘Arabic Speech Recognition System using CMU-Sphinx4',
 'On the Development of Text Input Method - Lessons Learned',
 'Network statistics on early English Syntax: Structural criteria',
 'Segmentation and Context of Literary and Musical Sequences',
 'International Standard for a Linguistic Annotation Framework',
 'A Formal Model of Dictionary Structure and Content',
 'Practical Approach to Knowledge-based Question Answering with Natural\n  Language Understanding and Advanced Reasoning',
 'Learning Probabilistic Models of Word Sense Disambiguation',
 'Learning Phonotactics Using ILP']
```

Tipica pipeline per text clustering

- Convertire i documenti in input ad embeddings
- Ridurre la dimensionalità degli embeddings
- Raggruppare embeddings simili con un algoritmo di clustering

Tipica pipeline per text clustering

- Convertire i documenti in input ad embeddings **[fase 1]**
  - Possiamo utilizzare centinaia di modelli di embeddings possibili
  - Proviamo ad utilizzare questo:
    - General Text Embeddings (GTE) model
      - https://huggingface.co/thenlper/gte-small

Tipica pipeline per text clustering

- Convertire i documenti in input ad embeddings **[fase 1]**
    - https://huggingface.co/thenlper/gte-small

```
from sentence_transformers import SentenceTransformer

model = SentenceTransformer("thenlper/gte-small")

embeddings = model.encode(abstracts, show_progress_bar=True)

embeddings.shape
```

Tipica pipeline per text clustering

- Ridurre la dimensionalità degli embeddings **[fase 2**

```
from umap import UMAP

umap_model = UMAP(n_components=5, min_dist=0.0, metric="cosine", random_state=42)

reduced_embeddings = umap_model.fit_transform(embeddings)
```

  - Un numero di componenti da 5 a 10 va spesso bene per gestire anche grandi dimensionalità di embeddings
  - La distanza minima tra embeddings = 0 crea cluster tipicamente compatti
  - Sempre meglio <u>cosine</u> piuttosto che <u>euclidean</u> con alta dimensionalità
  - random_state -> replicabilità ma senza parallelismo (più lento)

Tipica pipeline per text clustering

- Raggruppare embeddings simili con un algoritmo di clustering **[Fase 3]**
    - Useremo l'algoritmo HDBSCAN

```
from hdbscan import HDBSCAN

hdbscan_model = HDBSCAN(min_cluster_size=50, metric="euclidean",
   cluster_selection_method="eom").fit(reduced_embeddings)

clusters = hdbscan_model.labels_

len(set(clusters))
```

Tipica pipeline per text clustering

- Diamo un’occhiata ai risultati

```
import numpy as np

cluster = 0
for index in np.where(clusters==cluster)[0][:3]:
  print(abstracts[index][:300] + “… \n”)
```

```
A computer model of "a sense of humour" suggested previously
[arXiv:0711.2058,0711.2061], relating the humorous effect with a specific
malfunction in information processing, is given in somewhat different
exposition. Psychological aspects of humour are elaborated more thoroughly. The
mechanism of …

  Computer model of a "sense of humour" suggested previously [arXiv:0711.2058,
0711.2061, 0711.2270] is raised to the level of a realistic algorithm.
…

  Large scale surveys of public mood are costly and often impractical to
perform. However, the web is awash with material indicative of public mood such
as blogs, emails, and web queries. Inexpensive content analysis on such
extensive corpora can be used to assess public mood fluctuations. The work
…
```

Tipica pipeline per text clustering

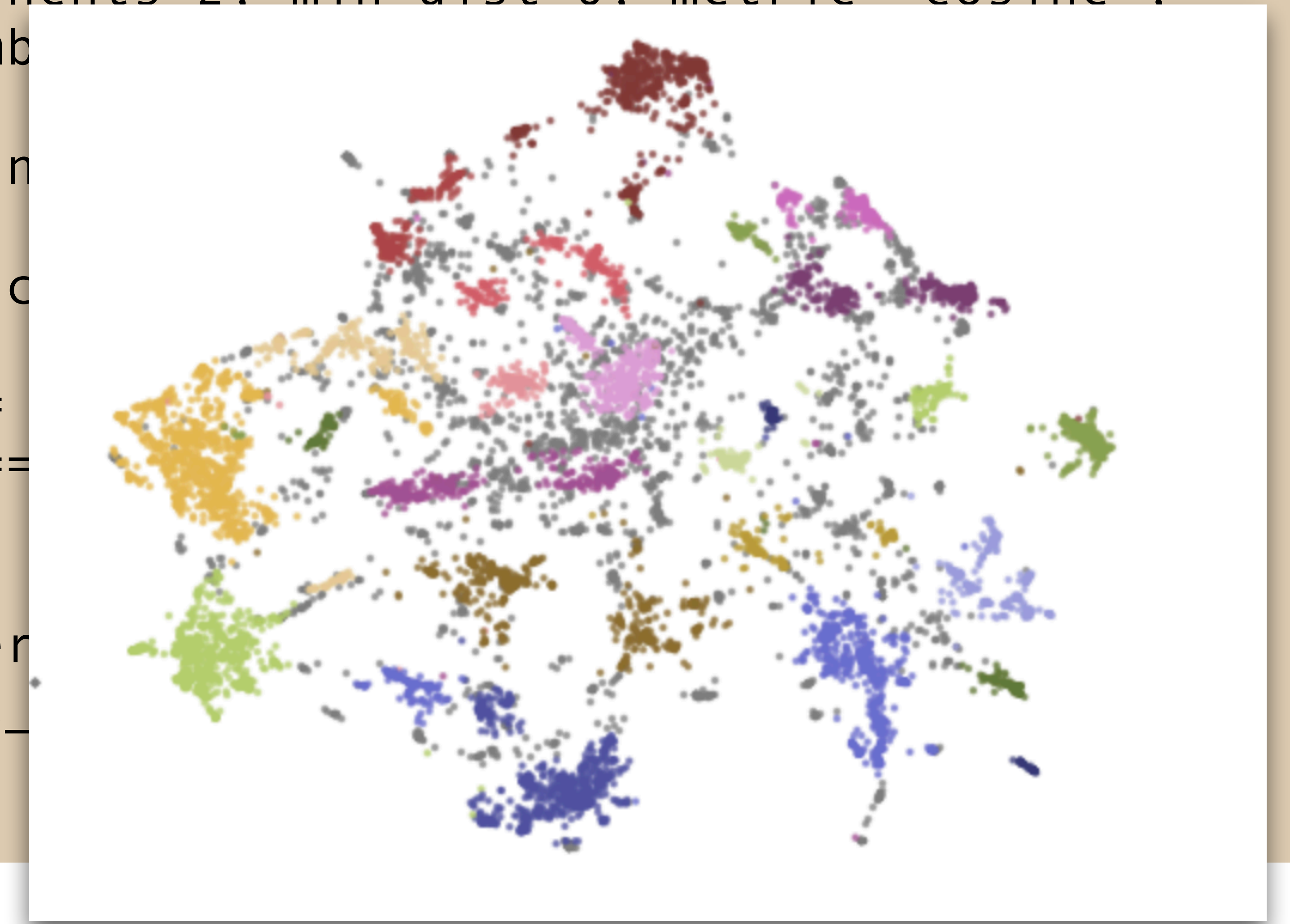
- Plot (clusters colorati, outliers in grigio)

```
import pandas as pd

reduced_embeddings = UMAP(n_components=2, min_dist=0, metric="cosine",
random_state=42).fit_transform(embeddings)

df = pd.DataFrame(reduced_embeddings, columns=["x", "y"])
df["title"] = titles[:1000]
df["cluster"] = [str(c) for c in clusters]

to_plot_df = df.loc[df.cluster != "-1", :]
outliers_df = df.loc[df.cluster == "-1", :]

import matplotlib.pyplot as plt
plt.scatter(outliers_df.x, outliers_df.y, alpha=0.6, s=2, c="grey")
plt.scatter(to_plot_df.x, to_plot_df.y, c=to_plot_df.cluster.astype(int),
alpha=0.6, s=2, cmap="tab20b")
plt.axis("off")
```

Tipica pipeline per text clustering

- Plot

```
import pandas as pd

reduced_embeddings = UMAP(n_components=2, min_dist=0, metric="cosine",
random_state=42).fit_transform(emb

df = pd.DataFrame(reduced_embeddin
df["title"] = titles[:1000]
df["cluster"] = [str(c) for c in c

to_plot_df = df.loc[df.cluster !=
outliers_df = df.loc[df.cluster ==

import matplotlib.pyplot as plt
plt.scatter(outliers_df.x, outlier
plt.scatter(to_plot_df.x, to_plot_
alpha=0.6, s=2, cmap="tab20b")
plt.axis("off")
```



Tipica pipeline per text clustering

- Diamo un’occhiata ai risultati

```
import numpy as np

cluster = 0
for index in np.where(clusters==cluster)[0][:3]:
  print(abstracts[index][:300] + “… \n”)
```

```
A computer model of "a sense of humour" suggested previously
[arXiv:0711.2058,0711.2061], relating the humorous effect with a specific
malfunction in information processing, is given in somewhat different
exposition. Psychological aspects of humour are elaborated more thoroughly. The
mechanism of …

  Computer model of a "sense of humour" suggested previously [arXiv:0711.2058,
0711.2061, 0711.2270] is raised to the level of a realistic algorithm.
…

  Large scale surveys of public mood are costly and often impractical to
perform. However, the web is awash with material indicative of public mood such
as blogs, emails, and web queries. Inexpensive content analysis on such
extensive corpora can be used to assess public mood fluctuations. The work
…
```

Topic Modeling: BERTopic

- Approccio basato su due componenti:
  - La prima è una pipeline di text clustering, come quella appena vista
    - Embeddings —> Dim. reduction —> Clustering
  - La seconda è il calcolo di una distribuzione di parole nei vari clusters (invece che genericamente in un corpus)
    - Da tf-idf a cluster-tf-idf

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

Topic Modeling: BERTopic

- Creazione del modello BERTopic

```
pip install bertopic

from bertopic import BERTopic

topic_model = BERTopic(
    embedding_model = model,
    umap_model = umap_model,
    hdbscan_model = hdbscan_model,
    verbose=True).fit(abstracts, embeddings)

topic_model.get_topic_info()

topic_model.get_topic(4)
```

```
topic_model.get_topic(4)
```

```
[('question', np.float64(0.04115600726035278)),
 ('questions', np.float64(0.030107294123957042)),
 ('answer', np.float64(0.029326524708882088)),
 ('answering', np.float64(0.024693077277456826)),
 ('the', np.float64(0.0224706801064735)),
 ('to', np.float64(0.020122045426418232)),
 ('qa', np.float64(0.019740165691257625)),
 ('and', np.float64(0.018451482996570852)),
 ('of', np.float64(0.017544512242 8602)),
 ('comprehension', np.float64(0.017399126383423693))]
```

Topic Modeling: BERTopic

- Interrogazione del modello BERTopic

```
topic_model.find_topics("…")
```

```
topic_model.find_topics("question answering")
```

```
([4, 2, -1, 29, 8],
 [np.float32(0.9301015),
  np.float32(0.8814262),
  np.float32(0.8811931),
  np.float32(0.8782772),
  np.float32(0.87549454)])
```

Topic Modeling: BERTopic

- Visualizzazione dei topic

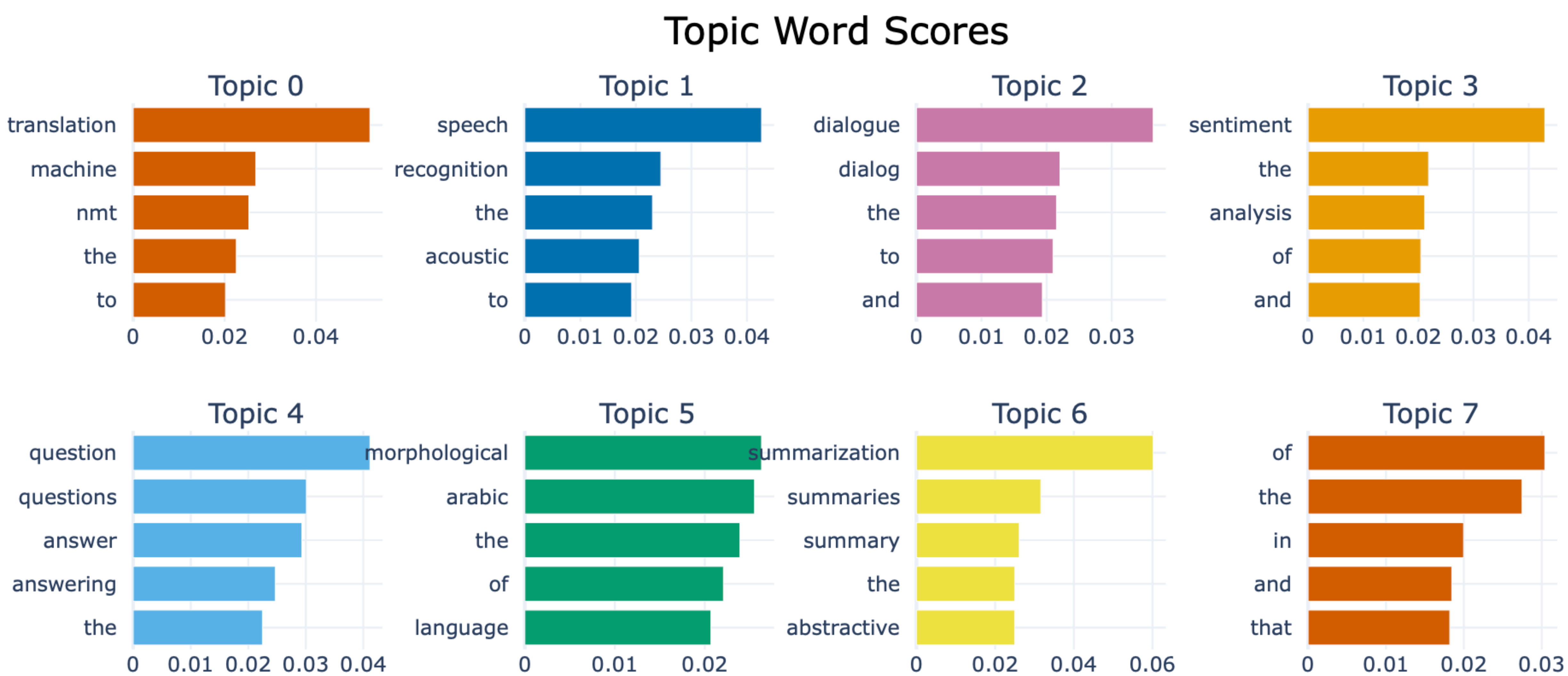
```
fig = topic_model.visualize_documents(
 titles,
 reduced_embeddings = reduced_embeddings,
 width = 1200,
 hide_annotations = True)

fig.update_layout(font = dict(size = 16))

topic_model.visualize_barchart()
```

```
topic_model.visualize_hierarchy()
```

```
topic_model.visualize_heatmap(n_clusters = 30)
```

Topic Modeling: BERTopic

- Esempio di visualizzazione



Topic Modeling: BERTopic

- Molte altre funzioni…

Method	Code
Fit the model	`.fit(docs)`
Fit the model and predict documents	`.fit_transform(docs)`
Predict new documents	`.transform([new_doc])`
Access single topic	`.get_topic(topic=12)`
Access all topics	`.get_topics()`
Get topic freq	`.get_topic_freq()`
Get all topic information	`.get_topic_info()`
Get all document information	`.get_document_info(docs)`
Get representative docs per topic	`.get_representative_docs()`
Update topic representation	`.update_topics(docs, n_gram_range=(1, 3))`
Generate topic labels	`.generate_topic_labels()`
Set topic labels	`.set_topic_labels(my_custom_labels)`
Merge topics	`.merge_topics(docs, topics_to_merge)`

[BERTopic: Neural topic modeling with a class-based TF-IDF procedure]
M Grootendorst - arXiv preprint arXiv:2203.05794, 2022

Lab n.4 - Clustering e Topic Modeling

- Esercizio di laboratorio di base n.4
  - Provare ad utilizzare una pipeline simile (potete variare anche i modelli) con **un nuovo dataset**
  - Il dataset deve essere “consistente”, ad es. >10k testi

Method	Code
Fit the model	`.fit(docs)`
Fit the model and predict documents	`.fit_transform(docs)`
Predict new documents	`.transform([new_doc])`
Access single topic	`.get_topic(topic=12)`
Access all topics	`.get_topics()`
Get topic freq	`.get_topic_freq()`
Get all topic information	`.get_topic_info()`
Get all document information	`.get_document_info(docs)`
Get representative docs per topic	`.get_representative_docs()`
Update topic representation	`.update_topics(docs, n_gram_range=(1, 3))`
Generate topic labels	`.generate_topic_labels()`
Set topic labels	`.set_topic_labels(my_custom_labels)`
Merge topics	`.merge_topics(docs, topics_to_merge)`