Analyse automatique du narratif de corpus

`Shami THIRION SEN`

Lien vers le git du projet

Intro et rappel

Après avoir fait une première analyse du narratif du communisme au premier semestre, l'object du projet actuel est de mettre en pratique l'analyse automatique du narratif avec les outils TAL. Les approches

Le but étant une comparaison de corpus ,

Script et procede

Afin de con

La puissance de TAL !

Semantique textuelle

- construction de corpus contrastif
  - corpus communiste vs. «non-communiste»
- défi temps - disponibilité des données
  - si le tp sem 1 visait le narratifs des temps communiste, (en hind sight) l'obtention du corpus est un défi non négligeable. À l'époque le Bengal est encore loin de la digitalisation des médias.
-

In [1]:
```
import numpy as np
import json
import glob
from pprint import pprint
from collections import defaultdict
import os, math
from utils import * # import des fonctions depuis le fichier utils

import gensim
import gensim.corpora as corpora
from gensim.corpora import Dictionary
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel

#visualisation
import pyLDAvis
import pyLDAvis.gensim

# outils TAL en bengali
import BnLemma as lm
from bnlp import BengaliPOS

import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

In [ ]:
```
```

In [2]:
```
# Bangla POS tagging
bn_pos = BengaliPOS()
from bnlp import BengaliCorpus as corpus
stopwords = corpus.stopwords[:20] + ['ৱি' + 'ঁি']
```

In [3]:
```
def gen_words(texts):
    single_string = [ ' '.join(line) for line in texts]
```

```
    data_words = [gensim.utils.simple_preprocess(text) for text in single_string]
    return data_words
```

In [4]:
```
## BIGRAMMES ET TIGRAMMES
def make_bigrams(texts):
    # print(bigram[doc] for doc in texts)
    return ([bigram[doc] for doc in texts])

def make_trigrams(texts):
    return([trigram[bigram[doc]] for doc in texts])
```

In [5]:
```
# on choisit le nombre de tokens (pour les 3 corpus)
num_tokens = 50000
```

BanglaGanashakti

- Lecture du corpus extrait et nettoyage

In [6]:
```
banglaGanashakti = "corpus/txtFiles/banglaGanashakti.txt"
filtered_ganashakti = read_corpus(banglaGanashakti, num_tokens)
filtered_ganashakti = [list for list in filtered_ganashakti if len(list)>0] # old list_of_docs
```

In [7]:
```
# generation
data_words  = gen_words(filtered_ganashakti)
print(data_words[9][:200])

# copy_this
bigram_phrases = gensim.models.Phrases(filtered_ganashakti, min_count=5, threshold=50)
trigram_phrases = gensim.models.Phrases(bigram_phrases[filtered_ganashakti], threshold=50)

bigram = gensim.models.phrases.Phraser(bigram_phrases)
trigram = gensim.models.phrases.Phraser(trigram_phrases)

data_bigrams = make_bigrams(filtered_ganashakti)
data_bigrams_trigrams = make_trigrams(data_bigrams)

# data_bigrams_trigrams = make_bigrams_trigrams(filtered_ganashakti)
```

```
['আরএসএস', 'অন', 'যবস', 'রহসন']
```

In [8]:
```
print(data_bigrams_trigrams[:10])
```

```
[['বিধান', 'গণতন্ত্র'], ['বিধান', 'প্রস্তাব', 'তন্ত্র', 'শব্দ', 'সরকার', 'বেসরকারি', 'হাত', 'পরিষ্কার'], ['বিয়োজন', 'চেয়ে', 'বিজেপি', 'সাংসদ', 'সরকার', 'মুখ', 'এমন', 'দেশ'], ['সংসদ', 'বেসরকারি', 'বিলটি', 'সংবাদ', 'ক্ষেত্র', 'দেশ', 'পুঁজি', 'প্রবেশ'], ['মাধ্যম', 'প্রতিবেদন', 'স্বনির্ভরতায়', 'প্রশ্ন', 'চিহ্ন', 'পড়া'], ['লেখা', 'বিধান', 'প্রাস্তাবনায়', 'প্রশ্ন', 'মুখ', 'বিচার', 'সুপ্রিমকোর্ট'], ['নিরপেক্ষ', 'শব্দ', 'সংযোজনা', 'দেশ', 'বিচার', 'যুক্তি', 'প্রমাণ', 'পরিবর্তন'], ['মূর্তি', 'খর', 'মন্তব্য', 'দেশ', 'একাত্তর', 'বিশ্বাস', 'ভিত্তি', 'অযোধ্যা'], ['কাশ্মীর', 'লাদাখ', 'হাইকোর্ট', 'বিচার', 'পঙ্কজ', 'মামলা', 'রায়', 'দান', 'অসুবিধে'], ['থাল', 'আরএসএস', 'অনুষ্ঠান', 'হাজির', 'গোটা', 'ব্যবস্হা', 'প্রহসন']]
```

In [9]:
```
### TF-IDF

id2word = Dictionary(data_bigrams_trigrams)
corpus = [id2word.doc2bow(text) for text in filtered_ganashakti]

word = id2word[[9][:1][0]]
print (word)
```

```
এমন
```

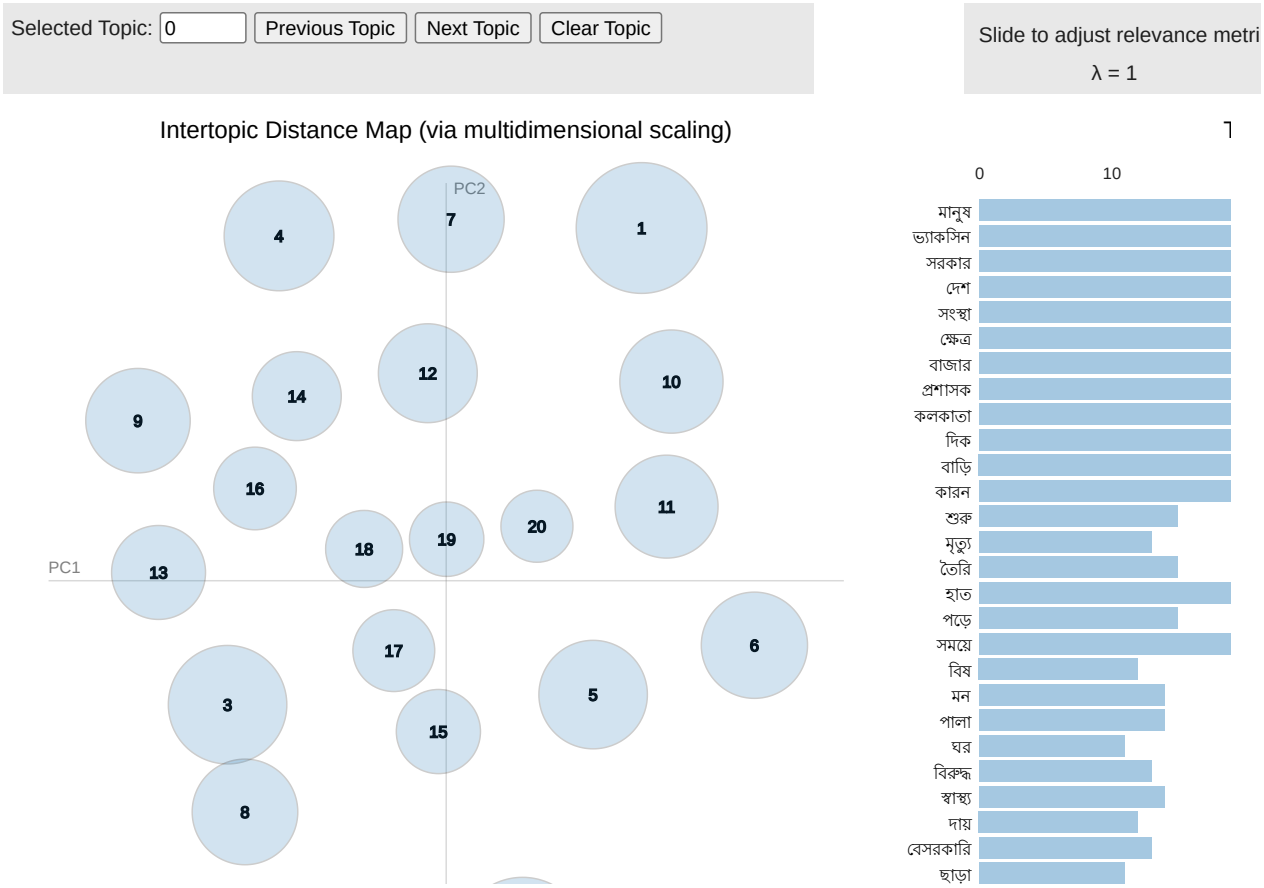
In [10]:
```
corpus=get_corpus(corpus,id2word)
# corpus[:10]
# print(corpus)
```

In [11]:
```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=20,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=200,
                                            passes=10,
                                            alpha="auto")
```

In [12]:
```
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word, mds="mmds", R=30)
```

```
vis
```

Out[12]:



In [13]:
```
pyLDAvis.save_html(vis, 'topic_modeling_ganashaki.html')
```

In [ ]:
```

```

Calcul spécificité

In [14]:
```
max_specificity, min_specificity = get_higest_lowest_specificity(filtered_ganashakti, 10)
```

In [15]:
```
max_specificity, '\n\n', min_specificity
```

Out[15]:
```
({'বছর': 3.981652821276082,
  'দেশ': 2.822366552123779,
  'সংস্থা': 2.4176201267623734,
  'ঘটনা': 1.7034556832544927,
  'বিরোধ': 1.7034556832544927,
  'রাজ্য': 1.7034556832544927,
  'কাগজ': 1.7034556832544927,
  'অভিযোগ': 1.7034556832544927,
  'সভা': 1.4214206099086821,
  'মুখ': 1.3058448579813575},
 '\n\n',
 {'মানুষ': -1.846381145742381,
  'আবেদন': -1.4734260601920461,
  'ভ্যাকসিন': -1.1313767185499328,
  'প্রশাসক': -1.1313767185499328,
  'মঙ্গলবার': -0.9112545540878407,
  'সরকার': -0.7250544362478527,
  'হাইকোর্ট': -0.6995223437085923,
  'তৈরি': -0.6995223437085923,
  'মন': -0.5976209869719409,
  'ভারত': -0.5644679352351545})
```

In [16]:
```
print('Token les plus spécifiques:\n')
pprint(max_specificity)
print('\nToken les moins spécifiques:\n')
pprint(min_specificity)
```

```
Token les plus spécifiques:

{'অভিযোগ': 1.7034556832544927,
 'কাগজ': 1.7034556832544927,
 'ঘটনা': 1.7034556832544927,
 'দেশ': 2.822366552123779,
 'বছর': 3.981652821276082,
 'বিরোধ': 1.7034556832544927,
 'মুখ': 1.3058448579813575,
 'রাজ্য': 1.7034556832544927,
 'সংস্থা': 2.4176201267623734,
 'সভা': 1.4214206099086821}

Token les moins spécifiques:

{'আবেদন': -1.4734260601920461,
 'তৈরি': -0.6995223437085923,
 'প্রশাসক': -1.1313767185499328,
 'ভারত': -0.5644679352351545,
 'ভ্যাকসিন': -1.1313767185499328,
 'মঙ্গলবার': -0.9112545540878407,
 'মন': -0.5976209869719409,
 'মানুষ': -1.846381145742381,
 'সরকার': -0.7250544362478527,
 'হাইকোর্ট': -0.6995223437085923}
```

Bartamanpatrika

In [17]:
```
bartamanpatrika = "corpus/txtFiles/bartamanpatrika.txt"
filtered_bartamanpatrika = read_corpus(bartamanpatrika, num_tokens)
filtered_bartamanpatrika = [list for list in filtered_bartamanpatrika if len(list)>0]
```

In [18]:
```
# generation
data_words  = gen_words(filtered_bartamanpatrika)
# print(data_words[9][:200])

# copy_this
bigram_phrases = gensim.models.Phrases(filtered_bartamanpatrika, min_count=5, threshold=50)
trigram_phrases = gensim.models.Phrases(bigram_phrases[filtered_bartamanpatrika], threshold=50)

bigram = gensim.models.phrases.Phraser(bigram_phrases)
trigram = gensim.models.phrases.Phraser(trigram_phrases)

data_bigrams = make_bigrams(filtered_bartamanpatrika)
data_bigrams_trigrams = make_trigrams(data_bigrams)
```

In [19]:
```
print(data_bigrams_trigrams[:10])
```

```
[['শত', 'নেতৃত্ব', 'ক্রিকেট', 'কনেট্রাল', 'বোর্ড', 'প্রস্তাব', 'গ্রহণ', 'শত', 'টেস্ট', 'অধিনায়ক', 'ম্যাচ', 'মাইলস্টোন', 'ম্যাচ',
'চাওয়া', 'অভিমানী', 'কে'], ['টেস্ট', 'অধিনায়ক', 'কোকিল', 'ইস্তফা', 'পাঁচ', 'ঝুমরাহকেও'], ['হ্যামস্ট্রিংয়ে', 'চোট', 'ওয়েটিং',
'ইন্ডিজের', 'বিরুদ্ধ', 'সিরিজ', 'শর্করা', 'ফেব্রুয়ারি', 'গোড়ায়', 'ভারত'], ['অধিনায়ক', 'অধীনে', 'কাজ'], ['ের', 'ব্যালট',
'ডিও', 'সি', 'ল্যাং', 'বা', 'পায়ে', 'জাদু', 'হার'], ['ভ্যাকসিন', 'নেব', 'নোভাক', 'জকোভিচ', 'প্রশ্ন', 'বিশ্ব'], ['ভার',
'গ্যালারি', 'নোনোভাক', 'লেখা', 'পোস্ট', 'জকোভিচ', 'অস্ট্রেলিয়ান'], ['প্রত্যাখ্যাত', 'বিলাত', 'গোলে', 'সুপার', 'কাপ', 'ঘর',
'রিয়াল', 'মাদ্রাসা', 'রবিবার', 'সৌদি', 'আরব'], ['আইএসএলে', 'রানার', 'মোহন', 'বাগান', 'সুবাদ', 'ব্রিগেড', 'কাপের',
'প্লেঅফে'], ['আইএসএলে', 'সোমবার', 'জাম', 'হায়', 'এফ', 'ম্যাচ']]
```

In [20]:
```
### TF-IDF

id2word = Dictionary(data_bigrams_trigrams)
corpus = [id2word.doc2bow(text) for text in filtered_bartamanpatrika]

word = id2word[[9][:1][0]]
print (word)
```

```
প্রস্তাব
```

In [21]:
```
corpus=get_corpus(corpus,id2word)
```

In [22]:
```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=30,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
```

```
                                    passes=10,
                                    alpha="auto")
```

In [23]:
```
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word, mds="mmds", R=30)
vis
```

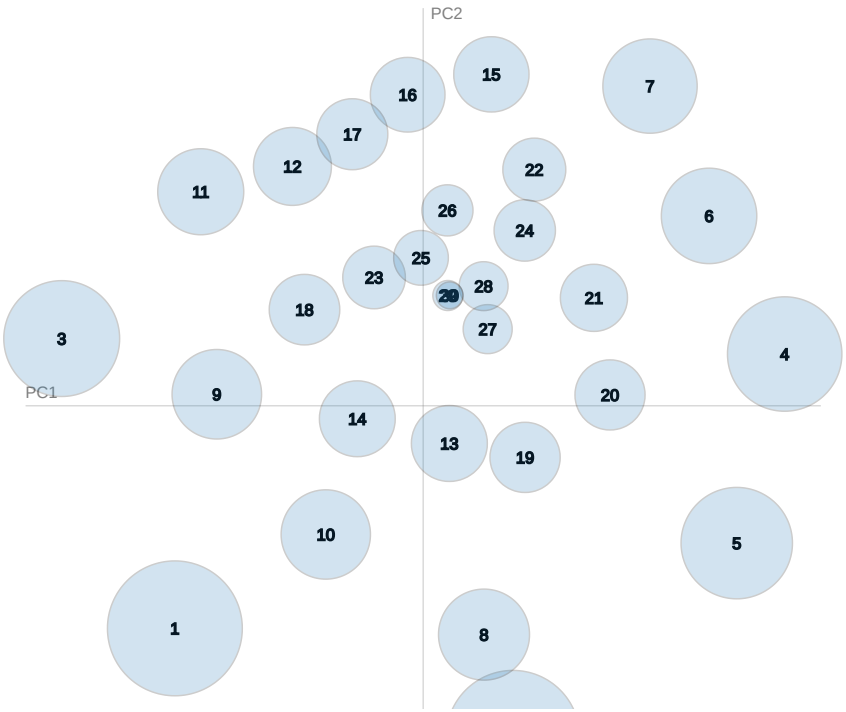
Out[23]:

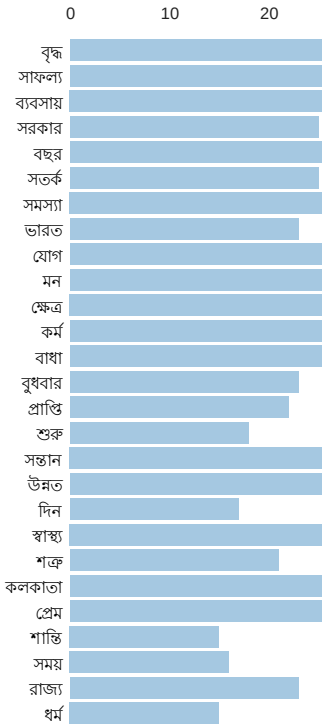
Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metri (2)

λ = 1

Intertopic Distance Map (via multidimensional scaling)





In [24]:
```
pyLDAvis.save_html(vis, 'topic_modeling_bartamanpatrika.html')
```

Calcul de spécificité

In [25]:
```
max_specificity, min_specificity = get_higest_lowest_specificity(filtered_bartamanpatrika, 10)
```

In [26]:
```
print('Token les plus spécifiques:\n')
pprint(max_specificity)
print('\nToken les moins spécifiques:\n')
pprint(min_specificity)
```

```
Token les plus spécifiques:

{'অগ্রসর': 2.213898518699129,
 'অর্থকড়ি': 1.1917822053964344,
 'আনন্দ': 1.194358070023163,
 'উপার্জন': 2.4791153722921853,
 'কাজ': 2.391591541360233,
 'পদ': 1.4933962140349704,
 'বিদ্যা': 2.175673473348266,
 'ব্যবসায়': 4.993389359623521,
 'যোগ': 2.640928523468916,
 'সুখ': 1.194358070023163}

Token les moins spécifiques:

{'কর্মী': -1.2203845358262584,
 'কলকাতা': -2.725720195793848,
 'গৃহে': -1.473846371473666,
 'টাকা': -1.282643850251739,
 'প্রেম': -2.225818409875582,
 'বুধবার': -2.263926350293847,
 'মমতা': -1.2203845358262584,
 'মানুষ': -1.473846371473666,
 'রাজ্জাক': -2.5345546860291215,
 'রাজ্য': -2.263926350293847}
```

Anandabazar

- corpus qui avec 1000 tokens pointe vers cricket et le sujet de sport
- Alors qu'avec 10000 nous obtenons plus de généralité

In [27]:
```
anandabazar = "corpus/txtFiles/anandabazar.txt"
filtered_anandabazar = read_corpus(anandabazar, num_tokens)
filtered_anandabazar = [list for list in filtered_anandabazar if len(list)>0]
```

In [28]:
```
# generation
data_words  = gen_words(filtered_anandabazar)
# print(data_words[9][:200])

# copy_this
bigram_phrases = gensim.models.Phrases(filtered_anandabazar, min_count=5, threshold=50)
trigram_phrases = gensim.models.Phrases(bigram_phrases[filtered_anandabazar], threshold=50)

bigram = gensim.models.phrases.Phraser(bigram_phrases)
trigram = gensim.models.phrases.Phraser(trigram_phrases)

data_bigrams = make_bigrams(filtered_anandabazar)
data_bigrams_trigrams = make_trigrams(data_bigrams)
```

In [29]:
```
print(data_bigrams_trigrams[:10])
```

```
[['১৫টা', 'লোক', 'ঘিরে'], ['ষষ্ঠী', 'সপ্তম', 'তুলনা', 'রবিবার', 'গা', 'রাস্তা', 'গা', 'উত্তর', 'দক্ষিণ', 'নিয়ন্ত্রণ', 'বড়', 'পুজোগুলোয়', 'চোখ', 'পড়া'], ['রাজ্য', 'বিজড়িত', 'ভাঙা', 'দশ', 'রাম', 'সুনীল', 'বনসাই', 'দল', 'নেতৃত্ব', 'পর্যবেক্ষণ', 'মঙ্গল', 'পরিকল্পনা', 'পালন', 'দায়িত্ব'], ['কোটি', 'গুটখায়', 'রঙ', 'শহর'], ['শীর্ষ', 'আদালত', 'মানুষ', 'আস্থা', 'বজায়', 'দায়িত্ব', 'কার'], ['সাল', 'বিধান', 'নির্বচন', 'শুরু', 'বিজড়িত', 'সাল', 'বিতর্ক', 'আরাধন', 'সুকান্ত', 'বারেক', 'পুজোতেই', 'সুলতান'], ['কংগ্রেস', 'প্রস্তাব', 'বাদ', 'শুরু', 'জোট'], ['ডিএ', 'মামলা', 'ধাক্কা', 'খেল', 'রাজ্য', 'সরকার'], ['সোমবার', 'রাত', '৮টা', 'মৃত্যু', 'মৃত', 'পরিবার', 'দেহ', 'নিতে', 'স্বীকার'], ['বারেক', 'চেষ্টায়', 'দিন', 'যোগ', 'সরকার', 'হা', 'অনুমোদন', 'রাহুলপ্রিয়ঙ্কা']]
```

In [30]:
```
### TF-IDF

id2word = Dictionary(data_bigrams_trigrams)
corpus = [id2word.doc2bow(text) for text in filtered_anandabazar]

word = id2word[[9][:1][0]]
print (word)

corpus=get_corpus(corpus,id2word)
# corpus[:10]
```

```
পড়া
```

In [31]:
```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
```

```
                                    num_topics=20,
                                    random_state=100,
                                    update_every=1,
                                    chunksize=200,
                                    passes=10,
                                    alpha="auto")
```

In [32]:
```
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word, mds="mmds", R=30)
vis
```

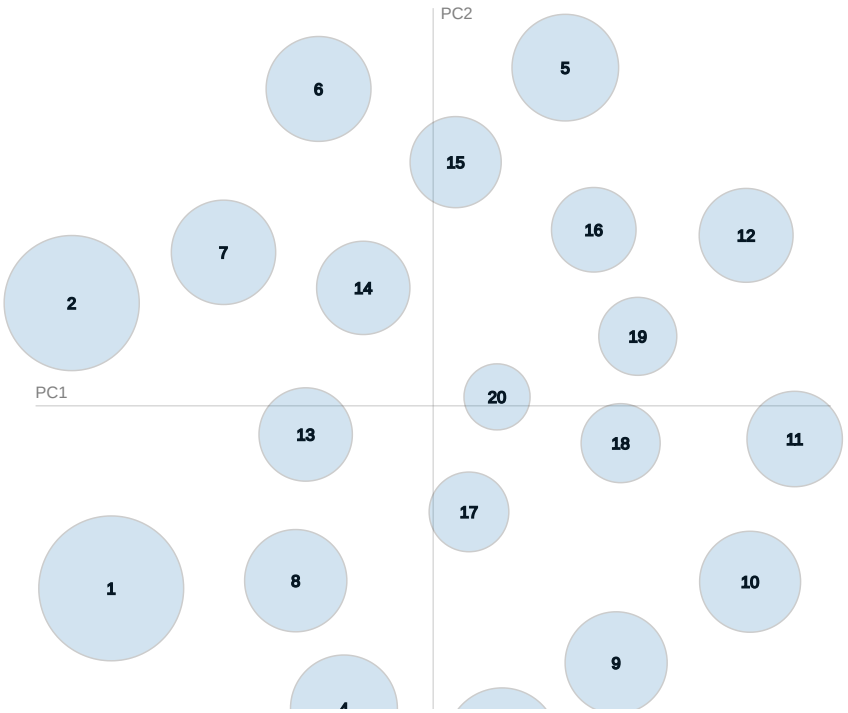
Out[32]:

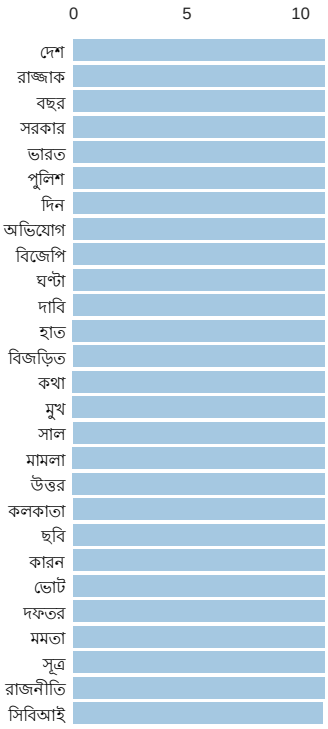
Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metri (2)

λ = 1

Intertopic Distance Map (via multidimensional scaling)





In [33]:
```
pyLDAvis.save_html(vis, 'topic_modeling_anandabazar.html')
```

Calcul de spécificité

In [34]:
```
max_specificity, min_specificity = get_higest_lowest_specificity(filtered_anandabazar, 10)
```

In [35]:
```
print('Token les plus spécifiques:\n')
pprint(max_specificity)
print('\nToken les moins spécifiques:\n')
pprint(min_specificity)
```

```
Token les plus spécifiques:

{'কলকাতা': 1.5778801614714248,
 'গা': 1.4970557220813931,
 'দিন': 1.4548077207324708,
 'দিল্লি': 1.4970557220813931,
 'নেতা': 1.5965450463804507,
 'পশ্চিম': 1.4970557220813931,
 'প্রাক্তন': 1.4970557220813931,
 'ভোট': 3.258295599096833,
 'ম্যাচ': 1.4970557220813931,
 'হাসপাতাল': 1.8719243107077876}

Token les moins spécifiques:

{'ঘণ্টা': -1.9391756661987243,
 'ছবি': -1.211895433809877,
 'টাকা': -1.3875543891087163,
 'দেশ': -1.1934798488373901,
 'ধার': -1.2700287066563456,
 'পাকিস্তান': -1.074699978134664,
 'বিজেপি': -1.5146775696522612,
 'বৃহস্পতিবার': -1.2700287066563456,
 'সোমবার': -1.2700287066563456,
 'হাত': -1.4695469731929143}
```

Conclusion

Après uen .. nous pouvons constater que...

P.S.

Ce rapport se trouve en version `jupyter notebook` dans le dossier `scripts`. Normalement, l'installation des bibliothèques mentionnées dans `requirement.txt` permettent d'executer le notebook afin de reproduire les résultats. (Il se peut que quelques bibiliothèques inutiles soient listés dans `requirement.txt`).

In [ ]: