Master Informatique, parcours MALIA-MIASHS

Carnets de note Python pour le cours de Network Analysis for Information Retrieval

Julien Velcin, laboratoire ERIC, Université Lyon 2

partie 3 : modélisation thématique

In [23]:
```
%matplotlib inline

from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
from gensim import corpora, models
#from gensim.models import  Word2Vec
import gensim
import pandas as pd
import numpy as np
from sklearn.preprocessing import normalize
import matplotlib.pyplot as plt
from matplotlib import cm
import os

# en utilisant gensim, on a besoin de préciser l'expression qui permet de "token
#tokenizer = RegexpTokenizer(r'\w+')
```

Preprocessing

On va illustrer l'utilisation du modèle LDA

In [24]:
```
from gensim.utils import simple_preprocess
import pandas

# on charge les données dans un tableau
#df = pandas.read_csv("datasets/huma1.csv", sep="\t")
#df = pandas.read_csv("datasets/dataconf.csv", sep="\t")

# lecture des données pour un fichier texte simple
with open(os.path.join("datasets", "Frank Herbert - Dune.txt")) as f:
    lines = [line.strip() for line in f.readlines()]
doc_set = lines

# colonne qui contient les textes à analyser
#var_texte = 'text'
#var_texte = 'title'
#doc_set = df[var_texte].tolist()


# fonction qui génère les listes de mots (token) à partir des textes
```

```
def sent_to_words(sentences):
    for sentence in sentences:
        yield(simple_preprocess(str(sentence), deacc=True))  # deacc=True removes p

# on construit le corpus
data_words = list(sent_to_words(doc_set))
```

In [25]:
```
# nombre total de documents
ndocs = len(data_words)
print(ndocs)
```

```
8608
```

Dictionnaire et corpus

On ne va pas utiliser la librairie *scikit-learn* cette fois, mais passer directement par la librairie *gensim*.

In [26]:
```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')

def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_word

# on retire les mots-outils
data_words_nostops = remove_stopwords(data_words)
```

In [27]:
```
print(data_words_nostops[:10])
```

```
[['dune'], ['frank', 'herbert'], [], ['copyright'], [], ['book'], ['dune'], [], []
[]]
```

In [28]:
```
# création du dictionnaire
dico = corpora.Dictionary(data_words_nostops)

# ce qui permet par ex. de filtrer le vocabulaire
dico.filter_extremes(no_below=10)

# Create Corpus
texts = data_words_nostops

# matrice Term Document Frequency
corpus = [dico.doc2bow(text) for text in texts]
```

Nous pouvons vérifier que l'entrée souhaitée est une liste de documents qui sont, chacun, représentés comme une liste de tuples (identifiant du mot dans le dictionnaire, TF).

In [29]:
```
print(corpus[0:20])
```

```
[[(0, 1)], [], [], [], [], [(1, 1)], [(0, 1)], [], [], [], [(0, 1), (2, 2), (3, 1),
(4, 1), (5, 1), (6, 2), (7, 1), (8, 3), (9, 1), (10, 1), (11, 2), (12, 1), (13, 1),
(14, 1), (15, 1), (16, 2), (17, 1), (18, 1), (19, 1), (20, 1), (21, 1), (22, 1), (2
3, 1), (24, 2), (25, 1), (26, 3), (27, 2), (28, 1), (29, 1), (30, 1), (31, 1), (32,
2), (33, 1), (34, 2), (35, 1), (36, 1)], [(11, 1), (24, 1), (37, 1), (38, 1), (39,
1)], [], [(2, 1), (40, 1), (41, 1), (42, 1), (43, 1), (44, 1), (45, 1), (46, 1)],
[(7, 1), (47, 1), (48, 1), (49, 1), (50, 1), (51, 1), (52, 1), (53, 1), (54, 1), (5
5, 1), (56, 1), (57, 1), (58, 1), (59, 1), (60, 1), (61, 1)], [(44, 1), (45, 1), (6
2, 1), (63, 1), (64, 1), (65, 1), (66, 1), (67, 1), (68, 1), (69, 1), (70, 1), (71,
1)], [(40, 1), (43, 1), (44, 1), (64, 1), (71, 1), (72, 1), (73, 1), (74, 1), (75,
1), (76, 1), (77, 1), (78, 1), (79, 1), (80, 1), (81, 1), (82, 2), (83, 1), (84, 1),
(85, 1), (86, 1), (87, 1), (88, 1), (89, 1), (90, 1), (91, 1), (92, 1)], [(44, 1),
(71, 1), (82, 1), (93, 1), (94, 1), (95, 1), (96, 1), (97, 1), (98, 1)], [(20, 1),
(43, 1), (45, 1), (48, 1), (99, 1), (100, 1), (101, 1), (102, 1), (103, 1), (104,
1)], [(15, 1), (44, 1), (71, 1), (105, 1), (106, 2), (107, 1)]]
```

In [30]:
```
len(dico)
```

Out[30]: 1809

Apprentissage du modèle

La plupart du temps, il faut fixer le nombre de thématiques souhaitées.

In [31]:
```
ntopics = 50
#ntopics = 20
```

On va utiliser le modèle LDA implémenté dans la librairie *gensim*.

In [32]:
```
# thanks to: https://miningthedetails.com/blog/python/lda/GensimLDA/
from gensim.models.ldamodel import LdaModel

#generate_lda = True
generate_lda = False

# generate LDA model
import logging

if generate_lda:
    print("generate new LDA model")
    logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=l
    ldamodel = LdaModel(corpus, num_topics=ntopics, id2word = dico,
                                            passes=100, random_state=100, per_word_t
    print(ldamodel)
```

On peut sauvegarder le modèle dans un fichier pour le garder en mémoire.

In [33]:
```
temp_file = "models/model_dataconf_" + str(ntopics)

if generate_lda:
    ldamodel.save(temp_file)
else:
```

```
    ldamodel = LdaModel.load(temp_file)
```

In [34]:
```
pwz = ldamodel.get_topics()

print("On peut récupérer la matrice stockant p(w|z):", pwz.shape)

# on peut aussi utiliser ldamodel.get_topic_terms(topicid, topn=n) pour obtenir
# les top n mots via leur identifiant, accompagnés de la proba p(w/z)

#ldamodel.get_topic_terms(1,topn=len(dico))
```

```
On peut récupérer la matrice stockant p(w|z): (50, 1809)
```

In [35]:
```
# show_topics permet d'afficher les mots directement
ldamodel.show_topics(num_topics=ntopics,formatted=False)
```

Out[35]:
```
[(0,
  [('fremen', 0.39152008),
   ('use', 0.05933129),
   ('wish', 0.04528319),
   ('al', 0.0389602),
   ('desert', 0.03285488),
   ('live', 0.028301666),
   ('remember', 0.023284208),
   ('clear', 0.021177702),
   ('one', 0.020621184),
   ('different', 0.018773999)]),
 (1,
  [('duke', 0.20715399),
   ('leto', 0.087413795),
   ('atreides', 0.073362716),
   ('son', 0.07008426),
   ('said', 0.056732424),
   ('alia', 0.04982056),
   ('daughter', 0.03306871),
   ('long', 0.032335553),
   ('one', 0.02794743),
   ('thus', 0.02754769)]),
 (2,
  [('gurney', 0.18998945),
   ('halleck', 0.09835282),
   ('said', 0.06459459),
   ('go', 0.063076265),
   ('called', 0.05651979),
   ('paul', 0.05225922),
   ('made', 0.043874662),
   ('man', 0.030175516),
   ('instant', 0.023096582),
   ('darkness', 0.022464601)]),
 (3,
  [('know', 0.14806195),
   ('said', 0.10589882),
   ('old', 0.09825239),
   ('woman', 0.07482917),
   ('lord', 0.07100542),
   ('yes', 0.061191496),
   ('ah', 0.04418092),
   ('young', 0.034209512),
   ('perhaps', 0.03125972),
   ('one', 0.027654435)]),
 (4,
  [('saw', 0.17431653),
   ('another', 0.1079708),
   ('felt', 0.08253656),
   ('paul', 0.043813605),
   ('thing', 0.038380653),
   ('someone', 0.036623828),
   ('jessica', 0.03478227),
   ('killed', 0.034575544),
   ('watched', 0.03234983),
   ('guards', 0.024033742)]),
 (5,
```

```
 [('mother', 0.2499172),
  ('reverend', 0.098743),
  ('noted', 0.05245002),
  ('attention', 0.051718373),
  ('power', 0.050754163),
  ('often', 0.02744687),
  ('within', 0.025123289),
  ('food', 0.02458133),
  ('jessica', 0.02452121),
  ('produced', 0.022216829)]),
(6,
 [('poison', 0.10300394),
  ('get', 0.06693086),
  ('hold', 0.052068826),
  ('message', 0.042719156),
  ('melange', 0.03934495),
  ('said', 0.035228346),
  ('reason', 0.03232111),
  ('orders', 0.02729153),
  ('produce', 0.02685917),
  ('assassins', 0.02628131)]),
(7,
 [('asked', 0.26652384),
  ('idaho', 0.07488156),
  ('paul', 0.06701268),
  ('basin', 0.064606704),
  ('jessica', 0.0585036),
  ('green', 0.055228002),
  ('wild', 0.037388813),
  ('subtle', 0.033820696),
  ('duncan', 0.029701157),
  ('throat', 0.026203008)]),
(8,
 [('rock', 0.05552254),
  ('light', 0.04187618),
  ('sand', 0.037450355),
  ('moved', 0.03314947),
  ('hulud', 0.027010739),
  ('shai', 0.027010739),
  ('across', 0.024885878),
  ('ahead', 0.023748005),
  ('white', 0.022351231),
  ('pack', 0.021845443)]),
(9,
 [('guild', 0.13922215),
  ('yet', 0.09629041),
  ('imperium', 0.04694776),
  ('human', 0.046749588),
  ('slowly', 0.03803798),
  ('report', 0.03631439),
  ('mean', 0.035089944),
  ('ancient', 0.034857806),
  ('simple', 0.03068954),
  ('best', 0.029148176)]),
(10,
 [('kynes', 0.1901735),
```

```
  ('man', 0.15439048),
  ('said', 0.06626435),
  ('used', 0.054833483),
  ('every', 0.05121806),
  ('kill', 0.03222243),
  ('liet', 0.023382148),
  ('way', 0.02231818),
  ('everything', 0.02079899),
  ('ritual', 0.020643305)]),
(11,
 [('eyes', 0.15014988),
  ('stared', 0.070243515),
  ('paul', 0.061641477),
  ('night', 0.039855924),
  ('feet', 0.039196406),
  ('usul', 0.035300333),
  ('rocks', 0.035215285),
  ('cave', 0.031640723),
  ('weapon', 0.030102286),
  ('man', 0.028910171)]),
(12,
 [('stilgar', 0.16005315),
  ('gesserit', 0.124152415),
  ('bene', 0.124152415),
  ('said', 0.055640686),
  ('count', 0.053947408),
  ('nodded', 0.041490674),
  ('dead', 0.03711738),
  ('fenring', 0.034534775),
  ('paul', 0.03212949),
  ('way', 0.031594284)]),
(13,
 [('say', 0.13123083),
  ('want', 0.064935185),
  ('three', 0.058421325),
  ('haderach', 0.037611097),
  ('kwisatz', 0.037611097),
  ('figure', 0.035037443),
  ('suddenly', 0.034354415),
  ('ring', 0.032839257),
  ('per', 0.030763177),
  ('said', 0.03014509)]),
(14,
 [('turned', 0.15377325),
  ('away', 0.061913818),
  ('beneath', 0.04617186),
  ('paul', 0.043457527),
  ('back', 0.038851023),
  ('lay', 0.034332138),
  ('robe', 0.033794086),
  ('question', 0.030437224),
  ('answer', 0.028789742),
  ('factory', 0.027553469)]),
(15,
 [('within', 0.12079146),
  ('shield', 0.115792155),
```

```
  ('touched', 0.052811593),
  ('field', 0.045379505),
  ('force', 0.041331105),
  ('fighting', 0.04130821),
  ('slow', 0.031256907),
  ('paul', 0.028739836),
  ('shields', 0.028225858),
  ('convention', 0.02400368)]),
(16,
 [('shall', 0.07530684),
  ('arm', 0.06649621),
  ('well', 0.06122491),
  ('upon', 0.048249193),
  ('certain', 0.041403834),
  ('set', 0.03584441),
  ('control', 0.034840427),
  ('said', 0.028866824),
  ('better', 0.02823575),
  ('free', 0.026955582)]),
(17,
 [('room', 0.07471277),
  ('yueh', 0.056084093),
  ('around', 0.03934243),
  ('paul', 0.039013818),
  ('table', 0.03804828),
  ('stood', 0.03416303),
  ('along', 0.030667052),
  ('hall', 0.027272547),
  ('stopped', 0.025716942),
  ('passage', 0.024208717)]),
(18,
 [('could', 0.13298492),
  ('knew', 0.089784525),
  ('training', 0.06192537),
  ('sietch', 0.05764583),
  ('new', 0.052461527),
  ('trained', 0.040432587),
  ('gave', 0.03589431),
  ('school', 0.033736963),
  ('fact', 0.03327423),
  ('one', 0.026005318)]),
(19,
 [('time', 0.18876338),
  ('nothing', 0.07228166),
  ('fear', 0.05561686),
  ('past', 0.051214196),
  ('real', 0.032396667),
  ('spread', 0.030337205),
  ('common', 0.027748667),
  ('effect', 0.026459336),
  ('gone', 0.025573675),
  ('looking', 0.024868153)]),
(20,
 [('first', 0.14794263),
  ('need', 0.08422058),
  ('like', 0.07192972),
```

```
  ('returned', 0.04296963),
  ('entire', 0.042536426),
  ('arrakeen', 0.03754136),
  ('pattern', 0.026617154),
  ('old', 0.024872828),
  ('except', 0.024270514),
  ('time', 0.023078674)]),
(21,
 [('great', 0.109255135),
  ('many', 0.090155095),
  ('things', 0.065794155),
  ('one', 0.06487031),
  ('always', 0.046765085),
  ('already', 0.036172904),
  ('rule', 0.033393335),
  ('order', 0.03266201),
  ('said', 0.026155818),
  ('animal', 0.020401852)]),
(22,
 [('arrakis', 0.2361108),
  ('never', 0.10496085),
  ('told', 0.06406042),
  ('lady', 0.055734463),
  ('person', 0.048116848),
  ('filled', 0.041881867),
  ('known', 0.034880113),
  ('become', 0.027011689),
  ('houses', 0.025671722),
  ('accepted', 0.025291461)]),
(23,
 [('name', 0.10908154),
  ('god', 0.08313867),
  ('world', 0.07303766),
  ('missionaria', 0.03912499),
  ('protectiva', 0.03912499),
  ('legend', 0.038937718),
  ('came', 0.036601163),
  ('brought', 0.036292277),
  ('faced', 0.03527855),
  ('removed', 0.035121955)]),
(24,
 [('rabban', 0.07662282),
  ('child', 0.07149293),
  ('knife', 0.066393815),
  ('troop', 0.061824944),
  ('slave', 0.046868138),
  ('matter', 0.038629983),
  ('plant', 0.03557482),
  ('hangings', 0.028979111),
  ('eye', 0.02778295),
  ('metal', 0.02706986)]),
(25,
 [('emperor', 0.19092879),
  ('sardaukar', 0.11565087),
  ('floor', 0.05338152),
  ('enough', 0.05272819),
```

```
  ('battle', 0.04544667),
  ('understand', 0.04512182),
  ('caught', 0.04104359),
  ('imperial', 0.038640518),
  ('said', 0.028641896),
  ('majesty', 0.026929954)]),
(26,
 [('hawat', 0.2437169),
  ('give', 0.077500366),
  ('tell', 0.07032109),
  ('whispered', 0.069836),
  ('silence', 0.061967067),
  ('said', 0.060754567),
  ('thufir', 0.04743242),
  ('paul', 0.03740274),
  ('beginning', 0.028406613),
  ('anger', 0.0282704)]),
(27,
 [('would', 0.15959162),
  ('come', 0.081279874),
  ('might', 0.059862405),
  ('still', 0.05386671),
  ('could', 0.049363762),
  ('beyond', 0.045764096),
  ('far', 0.044910934),
  ('silent', 0.03256906),
  ('ever', 0.028048769),
  ('kind', 0.025106387)]),
(28,
 [('people', 0.1619574),
  ('thopter', 0.05377284),
  ('dune', 0.045623135),
  ('paul', 0.04025301),
  ('hidden', 0.0375455),
  ('strength', 0.03720771),
  ('five', 0.032819156),
  ('remembered', 0.028475665),
  ('wait', 0.026875647),
  ('dream', 0.025707114)]),
(29,
 [('see', 0.3100301),
  ('feyd', 0.14444727),
  ('rautha', 0.12849264),
  ('maker', 0.057586282),
  ('boy', 0.040199015),
  ('part', 0.032277953),
  ('went', 0.028768046),
  ('found', 0.020856211),
  ('kept', 0.02038998),
  ('upward', 0.018641936)]),
(30,
 [('water', 0.21714847),
  ('spice', 0.13389687),
  ('life', 0.11129377),
  ('little', 0.048089817),
  ('moisture', 0.034343205),
```

```
  ('said', 0.030983191),
  ('ten', 0.02099759),
  ('open', 0.02096758),
  ('inner', 0.01970159),
  ('meters', 0.016941449)]),
(31,
 [('men', 0.18793824),
  ('door', 0.058052193),
  ('two', 0.053999927),
  ('air', 0.03820895),
  ('one', 0.037673946),
  ('guard', 0.029868923),
  ('pressed', 0.029328724),
  ('ledge', 0.026458904),
  ('friend', 0.025432592),
  ('leader', 0.02478685)]),
(32,
 [('house', 0.10341825),
  ('almost', 0.089465365),
  ('call', 0.06752987),
  ('blue', 0.04593871),
  ('grew', 0.034978997),
  ('secundus', 0.034355097),
  ('salusa', 0.034355097),
  ('creature', 0.03243554),
  ('planet', 0.032094117),
  ('turn', 0.031954914)]),
(33,
 [('face', 0.1614681),
  ('held', 0.08341671),
  ('keep', 0.059335694),
  ('sound', 0.05067602),
  ('mouth', 0.038340487),
  ('uncle', 0.03558729),
  ('help', 0.034966707),
  ('watch', 0.03223605),
  ('strange', 0.029975617),
  ('arms', 0.028220711)]),
(34,
 [('paul', 0.106312685),
  ('glanced', 0.08900973),
  ('back', 0.08326162),
  ('good', 0.072035946),
  ('said', 0.07147465),
  ('day', 0.06884489),
  ('blood', 0.036237806),
  ('leave', 0.035924207),
  ('love', 0.032002695),
  ('group', 0.024414087)]),
(35,
 [('word', 0.06841086),
  ('began', 0.058234576),
  ('jihad', 0.05318428),
  ('blade', 0.04210628),
  ('command', 0.03983185),
  ('also', 0.039496444),
```

```
  ('mapes', 0.03519532),
  ('butlerian', 0.033036098),
  ('dry', 0.032688536),
  ('crysknife', 0.03012572)]),
 (36,
  [('beside', 0.064983),
  ('step', 0.053780716),
  ('lips', 0.050606735),
  ('forced', 0.04596928),
  ('times', 0.04507487),
  ('hood', 0.036330454),
  ('knowledge', 0.032825857),
  ('secret', 0.030876935),
  ('cold', 0.029466107),
  ('said', 0.027508616)]),
 (37,
  [('hand', 0.12654708),
  ('looked', 0.095903546),
  ('right', 0.08561679),
  ('left', 0.073751815),
  ('paul', 0.06479004),
  ('without', 0.060826868),
  ('wondered', 0.033846065),
  ('though', 0.031194657),
  ('spoke', 0.029636558),
  ('away', 0.023874078)]),
 (38,
  [('desert', 0.09074755),
  ('took', 0.079913445),
  ('hands', 0.06374237),
  ('deep', 0.06330271),
  ('open', 0.04944181),
  ('worm', 0.046051513),
  ('second', 0.033141635),
  ('last', 0.030333806),
  ('forward', 0.0263377),
  ('breath', 0.023970338)]),
 (39,
  [('muad', 0.2001924),
  ('dib', 0.2001924),
  ('princess', 0.050506797),
  ('done', 0.048964284),
  ('future', 0.047125936),
  ('irulan', 0.039398663),
  ('arrakis', 0.031725932),
  ('words', 0.03125862),
  ('carry', 0.024958812),
  ('consider', 0.022794586)]),
 (40,
  [('sand', 0.13671437),
  ('dust', 0.055911824),
  ('surface', 0.050789136),
  ('across', 0.047704767),
  ('system', 0.04188492),
  ('dunes', 0.03828411),
  ('wind', 0.03617175),
```

```
  ('bible', 0.033219904),
  ('movement', 0.03137355),
  ('toward', 0.026327815)]),
 (41,
  [('must', 0.14934741),
  ('said', 0.060816288),
  ('may', 0.058116272),
  ('make', 0.05455242),
  ('among', 0.05367483),
  ('harkonnens', 0.048813354),
  ('course', 0.037656985),
  ('us', 0.037137102),
  ('thought', 0.03592772),
  ('religion', 0.035897903)]),
 (42,
  [('voice', 0.15205632),
  ('said', 0.08256458),
  ('heard', 0.076858655),
  ('cannot', 0.06855382),
  ('jessica', 0.057900995),
  ('something', 0.055622473),
  ('paul', 0.042244274),
  ('true', 0.037390944),
  ('tone', 0.032773767),
  ('hear', 0.026412904)]),
 (43,
  [('baron', 0.30991906),
  ('said', 0.0787332),
  ('take', 0.06491038),
  ('piter', 0.06337164),
  ('mentat', 0.043625418),
  ('change', 0.040966578),
  ('stop', 0.032754492),
  ('women', 0.02879308),
  ('could', 0.023631535),
  ('ship', 0.02096868)]),
 (44,
  [('side', 0.049000643),
  ('body', 0.03977899),
  ('stillsuit', 0.03936191),
  ('space', 0.036432575),
  ('awareness', 0.03549086),
  ('eyes', 0.028812343),
  ('lifted', 0.02636212),
  ('face', 0.020750437),
  ('studied', 0.020696018),
  ('felt', 0.019998286)]),
 (45,
  [('head', 0.110304356),
  ('years', 0.062038325),
  ('storm', 0.06142045),
  ('shook', 0.043833572),
  ('soon', 0.036718074),
  ('stare', 0.035834733),
  ('snapped', 0.03448375),
  ('bring', 0.032602303),
```

```
  ('talk', 0.03010532),
  ('appeared', 0.028082756)]),
 (46,
  [('thought', 0.17845885),
  ('think', 0.07309521),
  ('father', 0.061636817),
  ('paul', 0.05410372),
  ('let', 0.04916001),
  ('jessica', 0.04410644),
  ('look', 0.044026565),
  ('religious', 0.0415195),
  ('put', 0.03560068),
  ('seen', 0.034135517)]),
 (47,
  [('place', 0.1344542),
  ('planet', 0.12025114),
  ('even', 0.07199668),
  ('much', 0.055267043),
  ('caladan', 0.044730254),
  ('arrakis', 0.04418067),
  ('test', 0.03477705),
  ('full', 0.031960007),
  ('followed', 0.031850494),
  ('show', 0.030024385)]),
 (48,
  [('harkonnen', 0.14870979),
  ('us', 0.14105566),
  ('point', 0.048578393),
  ('behind', 0.04706231),
  ('death', 0.037594385),
  ('half', 0.037526365),
  ('sire', 0.03346051),
  ('one', 0.030724728),
  ('six', 0.02548095),
  ('said', 0.024854934)]),
 (49,
  [('high', 0.08901027),
  ('small', 0.0531295),
  ('presently', 0.040310267),
  ('family', 0.032458954),
  ('major', 0.03198854),
  ('obvious', 0.030293174),
  ('form', 0.029185886),
  ('onto', 0.028794333),
  ('higher', 0.028585242),
  ('permitted', 0.02844581)])]
```

On peut vérifier que les mots ont une probabilité d'appartenir à *plusieurs* thématiques

In [36]:
```
import numpy as np
tab = ldamodel.get_topics()
myword = dico.token2id['sand']
np.where(tab[:,myword]>0.01)
```

Out[36]:
```
(array([ 8, 40]),)
```

In [ ]:
```
# Enable logging for gensim - optional
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=loggi

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

Si on souhaite obtenir p(z|d), il faut réexécuter le modèle sur les données (par ex., le corpus).

Deprecated

Les procédures suivantes fournissent plusieurs "vues" intéressantes sur le modèle. Elles viennent du site *machinelearningplus.com* :

https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

Tout d'abord, on souhaite un tableau qui liste la thématique majoritaire pour chaque document, accompagnée par ses mots les plus probables.

In [ ]:
```
def format_topics_sentences(ldamodel, corpus, texts):
    # Init output
    sent_topics_df = pd.DataFrame()

    # Get main topic in each document
    i=0
    for i, row in enumerate(ldamodel[corpus]):
        #print(row[0])
        row = sorted(row[0], key=lambda x: (x[1]), reverse=True)
        #print(row)
        # Get the Dominant topic, Perc Contribution and Keywords for each document
        for j, (topic_num, prop_topic) in enumerate(row):
            if j == 0:  # => dominant topic
                wp = ldamodel.show_topic(topic_num)
                topic_keywords = ", ".join([word for word, prop in wp])
                sent_topics_df = pd.concat([sent_topics_df, pd.Series([int(topic_nu
            else:
                break
    sent_topics_df.columns = ['Dominant_Topic', 'Perc_Contribution', 'Topic_Keyword

    # Add original text to the end of the output
    contents = pd.Series(texts)
    sent_topics_df = pd.concat([sent_topics_df, contents], axis=1)
    return(sent_topics_df)


df_topic_sents_keywords = format_topics_sentences(ldamodel=ldamodel, corpus=corpus,

# Format
df_dominant_topic = df_topic_sents_keywords.reset_index()
df_dominant_topic.columns = ['Document_No', 'Dominant_Topic', 'Topic_Perc_Contrib',
```

```
# Show
df_dominant_topic.head(10)
```

On peut vouloir obtenir les documents les plus "représentatifs" de chaque thématique (attention, au sens de p(z|d)).

In [ ]:
```
sent_topics_sorteddf = pd.DataFrame()

sent_topics_outdf_grpd = df_topic_sents_keywords.groupby('Dominant_Topic')

for i, grp in sent_topics_outdf_grpd:
    sent_topics_sorteddf = pd.concat([sent_topics_sorteddf,
                                                grp.sort_values(['Perc_Contribution'],
                                               axis=0)

# Reset Index
sent_topics_sorteddf.reset_index(drop=True, inplace=True)

# Format
sent_topics_sorteddf.columns = ['Topic_Num', "Topic_Perc_Contrib", "Keywords", "Tex

# Show
sent_topics_sorteddf
```

Pour finir, le "volume" estimé de documents (en réalité, de mots) couverts par les différentes thématiques.

In [ ]:
```
# Number of Documents for Each Topic
topic_counts = df_topic_sents_keywords['Dominant_Topic'].value_counts()
topic_counts
```

In [ ]:
```

```