```
In [1]: import pandas as pd
```

```
In [4]: import numpy as np
```

```
In [15]: data = pd.read_csv("people.csv")
```

```
In [23]: data[data["name"] == 'digby morrell']['text'].iloc[0]
```

```
Out[23]: 'digby morrell born 10 october 1979 is a former australian rules footballer who
         played with the kangaroos and carlton in the australian football league aflfrom
         western australia morrell played his early senior football for west perth his 4
         4game senior career for the falcons spanned 19982000 and he was the clubs leadi
         ng goalkicker in 2000 at the age of 21 morrell was recruited to the australian
         football league by the kangaroos football club with its third round selection i
         n the 2001 afl rookie draft as a forward he twice kicked five goals during his
         time with the kangaroos the first was in a losing cause against sydney in 2002
         and the other the following season in a drawn game against brisbaneafter the 20
         03 season morrell was traded along with david teague to the carlton football cl
         ub in exchange for corey mckernan he played 32 games for the blues before being
         delisted at the end of 2005 he continued to play victorian football league vfl
         football with the northern bullants carltons vflaffiliate in 2006 and acted as
         playing assistant coach in 2007 in 2008 he shifted to the box hill hawks before
         retiring from playing at the end of the season from 2009 until 2013 morrell was
         the senior coach of the strathmore football club in the essendon district footb
         all league leading the club to the 2011 premier division premiership since 2014
         he has coached the west coburg football club also in the edflhe currently teach
         es physical education at parade college in melbourne'
```

```
In [8]: np.char.lower(np.array(data["name"]).astype(str))
```

```
Out[8]: array(['digby morrell', 'alfred j. lewy', 'harpdog brown', ...,
               'david cass (footballer)', 'keith elias', 'fawaz damrah'],
              dtype='<U70')
```

```
In [7]: np.array(data["name"]).astype(str)
```

```
Out[7]: array(['Digby Morrell', 'Alfred J. Lewy', 'Harpdog Brown', ...,
               'David Cass (footballer)', 'Keith Elias', 'Fawaz Damrah'],
              dtype='<U70')
```

In [79]:
```
data.head()
```

Out[79]:

	URI	name	text
0	<http://dbpedia.org/resource/Digby_Morrell>	Digby Morrell	digby morrell born 10 october 1979 is a former...
1	<http://dbpedia.org/resource/Alfred_J._Lewy>	Alfred J. Lewy	alfred j lewy aka sandy lewy graduated from un...
2	<http://dbpedia.org/resource/Harpdog_Brown>	Harpdog Brown	harpdog brown is a singer and harmonica player...
3	<http://dbpedia.org/resource/Franz_Rottensteiner>	Franz Rottensteiner	franz rottensteiner born in waidmannsfeld lowe...
4	<http://dbpedia.org/resource/G-Enka>	G-Enka	henry krvits born 30 december 1974 in tallinn ...

In [80]:
```
data.shape
```

Out[80]:
```
(59071, 3)
```

In [ ]:
```
obama = data[data["name"] == 'Barack Obama']
```

In [ ]:
```
obama["text"].iloc[0]
```

In [ ]:
```
import metapy
```

In [ ]:
```
metapy.log_to_stderr()
```

Tokenlize

In [ ]:
```
doc = metapy.index.Document()
doc.content(obama["text"].iloc[0])
```

In [ ]:
```
tok = metapy.analyzers.ICUTokenizer()
```

In [ ]:
```
tok = metapy.analyzers.ICUTokenizer(suppress_tags=True)
tok.set_content(doc.content())
tokens = [token for token in tok]
print tokens
```

Remove Stopword and Stemming

In [ ]:

```
tok = metapy.analyzers.ListFilter(tok, "lemur-stopwords.txt", metapy.analyzers.Li
tok.set_content(doc.content())
tokens = [token for token in tok]
print(tokens)
```

In [ ]:

```
tok = metapy.analyzers.Porter2Filter(tok)
tok.set_content(doc.content())
tokens = [token for token in tok]
print(tokens)
```

Uni-Grams

In [ ]:

```
tok = metapy.analyzers.ICUTokenizer(suppress_tags = True)
tok = metapy.analyzers.LowercaseFilter(tok)
tok.set_content(doc.content())
ana = metapy.analyzers.NGramWordAnalyzer(1, tok)
unigrams = ana.analyze(doc)
print(unigrams)
```

Index prepare

In [ ]:

```
with open("./data/data.dat", "a") as text_file:
    for i in data["name"]:
        try:
            i = i.replace("*","")
            context = str(data[data["name"] == i]["text"].iloc[0])
            text_file.write(" %s \n" % context)
        except:
            skip = True
```

Topic modeling

In [6]:

```
fidx = metapy.index.make_forward_index('people-config.toml')
```

In [ ]:

```
dset = metapy.learn.Dataset(fidx)
```

2 Topic

In [ ]:

```
lda_inf = metapy.topics.LDACollapsedVB(dset, num_topics=2, alpha=1.0, beta=0.01)
lda_inf.run(num_iters=1000)
```

```
In [ ]: lda_inf.save('lda-cvb0')
```

```
In [44]: model = metapy.topics.TopicModel('lda-cvb0')
```

```
In [45]: model.top_k(tid=0)
```

```
Out[45]: [(327105, 0.009785797010899941),
          (283479, 0.0071463531879504116),
          (148007, 0.005922879414277428),
          (350829, 0.0059019653303974615),
          (51910, 0.005900708745190318),
          (376515, 0.005575504069944809),
          (418367, 0.004955867270834303),
          (291712, 0.004627750146476143),
          (160720, 0.004438954237584389),
          (238525, 0.004363857906685078)]
```

```
In [46]: scorer = metapy.topics.BLTermScorer(model)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model.top_k(tid=0, scorer=scorer)]
```

```
Out[46]: [('play', 0.06938529502041213),
          ('music', 0.05124147885954711),
          ('season', 0.04096896697490493),
          ('record', 0.038645531914622167),
          ('film', 0.03617550962295213),
          ('game', 0.030924819548928895),
          ('leagu', 0.029866449415717256),
          ('album', 0.026202557127503023),
          ('championship', 0.020637510275398174),
          ('footbal', 0.02062173657119315)]
```

```
In [47]: [(fidx.term_text(pr[0]), pr[1]) for pr in model.top_k(tid=1, scorer=scorer)]
```

```
Out[47]: [('elect', 0.030420555534287796),
          ('presid', 0.024517339067962884),
          ('research', 0.021354025886095473),
          ('parti', 0.01872255360627406),
          ('polit', 0.017845238657655138),
          ('professor', 0.017500141787774592),
          ('law', 0.017468957221604686),
          ('scienc', 0.01687604853878666),
          ('institut', 0.015306113442293934),
          ('govern', 0.015218302276223228)]
```

```
In [48]: model.topic_distribution(0)
```

```
Out[48]: <metapy.stats.Multinomial {0: 0.497348, 1: 0.502652}>
```

```
In [49]: model.topic_distribution(1)
```

```
Out[49]: <metapy.stats.Multinomial {0: 0.083592, 1: 0.916408}>
```

```
In [50]: model.topic_distribution(1000)
```

```
Out[50]: <metapy.stats.Multinomial {0: 0.983111, 1: 0.016889}>
```

3 Topics

lda_inf = metapy.topics.LDACollapsedVB(dset, num_topics=3, alpha=1.0, beta=0.01) lda_inf.run(num_iters=1000) lda_inf.save('lda-cvb3')

```
In [7]: model3 = metapy.topics.TopicModel('lda-cvb3')
```

```
In [8]: scorer = metapy.topics.BLTermScorer(model3)
        [(fidx.term_text(pr[0]), pr[1]) for pr in model3.top_k(tid=0, scorer=scorer)]
```

```
Out[8]: [('music', 0.09169856792557862),
         ('film', 0.07429887687220926),
         ('album', 0.04527100249155236),
         ('artist', 0.03588293693241695),
         ('band', 0.03239947019198711),
         ('art', 0.03030812769185176),
         ('produc', 0.02796352004903616),
         ('song', 0.02711576987130411),
         ('record', 0.026060539375978443),
         ('festiv', 0.023501562162664413)]
```

```
In [9]: scorer = metapy.topics.BLTermScorer(model3)
        [(fidx.term_text(pr[0]), pr[1]) for pr in model3.top_k(tid=1, scorer=scorer)]
```

```
Out[9]: [('elect', 0.05030631508000383),
         ('presid', 0.03960303917558564),
         ('research', 0.03463397865905549),
         ('law', 0.028550523488983792),
         ('govern', 0.025058447856021182),
         ('parti', 0.02273572285076134),
         ('minist', 0.02253303000505191),
         ('polit', 0.021824747024331792),
         ('committe', 0.021064927035445713),
         ('professor', 0.017727376262357243)]
```

In [10]:
```
scorer = metapy.topics.BLTermScorer(model3)
[(fidx.term_text(pr[0]), pr[1]) for pr in model3.top_k(tid=2, scorer=scorer)]
```

Out[10]:
```
[('season', 0.11557321205151004),
 ('leagu', 0.09364373739693062),
 ('game', 0.08667422925714163),
 ('play', 0.08329153768792841),
 ('team', 0.07488070272419318),
 ('footbal', 0.06297453282344603),
 ('championship', 0.062200003024505485),
 ('coach', 0.06095219214951442),
 ('player', 0.03580290397253716),
 ('finish', 0.0349529629361811)]
```

In [11]:
```
model3.topic_distribution(0)
```

Out[11]:
```
<metapy.stats.Multinomial {0: 0.377516, 1: 0.392771, 2: 0.229713}>
```

In [12]:
```
model3.topic_distribution(1)
```

Out[12]:
```
<metapy.stats.Multinomial {0: 0.157031, 1: 0.811705, 2: 0.031264}>
```

In [13]:
```
model3.topic_distribution(1000)
```

Out[13]:
```
<metapy.stats.Multinomial {0: 0.014301, 1: 0.012469, 2: 0.973230}>
```

4 Topics

lda_inf = metapy.topics.LDACollapsedVB(dset, num_topics=4, alpha=1.0, beta=0.01) lda_inf.run(num_iters=1000) lda_inf.save('lda-cvb4')

In [14]:
```
model4 = metapy.topics.TopicModel('lda-cvb4')
```

In [15]:
```
scorer = metapy.topics.BLTermScorer(model4)
[(fidx.term_text(pr[0]), pr[1]) for pr in model4.top_k(tid=0, scorer=scorer)]
```

Out[15]:
```
[('elect', 0.08897789477214402),
 ('parti', 0.05755335364093145),
 ('presid', 0.046899063344507),
 ('serv', 0.04257288026777284),
 ('law', 0.04187436600652317),
 ('minist', 0.041310148712478956),
 ('govern', 0.033253366189349265),
 ('offic', 0.03213095004751276),
 ('polit', 0.02717720027277465),
 ('court', 0.02674288213731779)]
```

In [16]:
```
scorer = metapy.topics.BLTermScorer(model4)
[(fidx.term_text(pr[0]), pr[1]) for pr in model4.top_k(tid=1, scorer=scorer)]
```

Out[16]:
```
[('book', 0.06829127520010303),
 ('research', 0.06153011016452292),
 ('art', 0.057236676594296654),
 ('publish', 0.05011590611795591),
 ('professor', 0.04353417841973739),
 ('univers', 0.043479649828075395),
 ('scienc', 0.034464923371558376),
 ('institut', 0.03044429738184895),
 ('journal', 0.027489167579242996),
 ('studi', 0.026301905912810952)]
```

In [17]:
```
scorer = metapy.topics.BLTermScorer(model4)
[(fidx.term_text(pr[0]), pr[1]) for pr in model4.top_k(tid=2, scorer=scorer)]
```

Out[17]:
```
[('play', 0.1161384101747569),
 ('season', 0.10976483862334463),
 ('leagu', 0.10840315353909313),
 ('team', 0.09494584780206905),
 ('game', 0.08670567509732785),
 ('footbal', 0.07299426670013531),
 ('championship', 0.07222726176144456),
 ('coach', 0.0707304271838699),
 ('player', 0.047326594909348374),
 ('cup', 0.04037521042506809)]
```

In [18]:
```
scorer = metapy.topics.BLTermScorer(model4)
[(fidx.term_text(pr[0]), pr[1]) for pr in model4.top_k(tid=3, scorer=scorer)]
```

Out[18]:
```
[('music', 0.13553445937561692),
 ('film', 0.10816639735459044),
 ('album', 0.06762199352551734),
 ('record', 0.05125213407484713),
 ('band', 0.04840303229704079),
 ('song', 0.04036783143356343),
 ('produc', 0.039849537157722326),
 ('play', 0.03327065780318017),
 ('perform', 0.03027760984197886),
 ('festiv', 0.029127831224422418)]
```

In [19]:
```
model4.topic_distribution(0)
```

Out[19]:
```
<metapy.stats.Multinomial {0: 0.246415, 1: 0.251515, 2: 0.216952, 3: 0.285118}>
```

In [20]:
```
model4.topic_distribution(1)
```

Out[20]:
```
<metapy.stats.Multinomial {0: 0.053007, 1: 0.808034, 2: 0.048655, 3: 0.090303}>
```

```
In [21]: model4.topic_distribution(1000)
```

```
Out[21]: <metapy.stats.Multinomial {0: 0.011992, 1: 0.014706, 2: 0.960222, 3: 0.013080}>
```

5 Topics

lda_inf = metapy.topics.LDACollapsedVB(dset, num_topics=5, alpha=1.0, beta=0.01) lda_inf.run(num_iters=1000) lda_inf.save('lda-cvb5')

```
In [22]: model5 = metapy.topics.TopicModel('lda-cvb5')
```

```
In [23]: scorer = metapy.topics.BLTermScorer(model5)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model5.top_k(tid=0, scorer=scorer)]
```

```
Out[23]: [('elect', 0.09402805252539857),
          ('parti', 0.061255731541558944),
          ('law', 0.04935793231729857),
          ('minist', 0.04833009845080838),
          ('serv', 0.0461625560452415),
          ('presid', 0.04502619524310325),
          ('offic', 0.03681100643725399),
          ('govern', 0.03355584984320714),
          ('polit', 0.03311922150551379),
          ('court', 0.031563272205838544)]
```

```
In [24]: scorer = metapy.topics.BLTermScorer(model5)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model5.top_k(tid=1, scorer=scorer)]
```

```
Out[24]: [('music', 0.11537995797231959),
          ('de', 0.1056276856532179),
          ('orchestra', 0.0568379965531669),
          ('opera', 0.03400856900748322),
          ('festiv', 0.03345632300655526),
          ('compos', 0.030808396671184294),
          ('perform', 0.029044416140327562),
          ('symphoni', 0.027859874111517163),
          ('la', 0.025826651890129605),
          ('studi', 0.02468028139541567)]
```

```
In [25]: scorer = metapy.topics.BLTermScorer(model5)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model5.top_k(tid=2, scorer=scorer)]
```

```
Out[25]: [('film', 0.10130643791742353),
          ('album', 0.08143328249030457),
          ('music', 0.06924231459357788),
          ('band', 0.059044115369419184),
          ('produc', 0.04569732762297925),
          ('televis', 0.04365450731640906),
          ('record', 0.04331922096896701),
          ('show', 0.03557488466836567),
          ('song', 0.03404206824626779),
          ('play', 0.03105612020944859)]
```

```
In [26]: scorer = metapy.topics.BLTermScorer(model5)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model5.top_k(tid=3, scorer=scorer)]
```

```
Out[26]: [('univers', 0.08358091778449737),
          ('book', 0.07911023180320091),
          ('research', 0.07563531853932073),
          ('professor', 0.04922177748029367),
          ('publish', 0.04646944049200874),
          ('scienc', 0.04630359797973105),
          ('art', 0.04627349679241953),
          ('journal', 0.03370901640501476),
          ('institut', 0.03032858327777639),
          ('studi', 0.02518318034482947)]
```

```
In [27]: scorer = metapy.topics.BLTermScorer(model5)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model5.top_k(tid=4, scorer=scorer)]
```

```
Out[27]: [('season', 0.13858336573129917),
          ('leagu', 0.12240739696593153),
          ('team', 0.10996020392373325),
          ('play', 0.1066592393061887),
          ('game', 0.09697900882056132),
          ('footbal', 0.08221036900352488),
          ('championship', 0.08107853407053334),
          ('coach', 0.07947565393641796),
          ('player', 0.04546182969943403),
          ('cup', 0.04532679535951356)]
```

```
In [28]: model5.topic_distribution(0)
```

```
Out[28]: <metapy.stats.Multinomial {0: 0.219876, 1: 0.118604, 2: 0.241416, 3: 0.218428,
         4: 0.201676}>
```

```
In [29]: model5.topic_distribution(1)
```

```
Out[29]: <metapy.stats.Multinomial {0: 0.035474, 1: 0.010497, 2: 0.132885, 3: 0.780242,
         4: 0.040901}>
```

```
In [30]: model5.topic_distribution(1000)
```

```
Out[30]: <metapy.stats.Multinomial {0: 0.011898, 1: 0.015139, 2: 0.013069, 3: 0.014577,
         4: 0.945317}>
```

6 Topics

lda_inf = metapy.topics.LDACollapsedVB(dset, num_topics=6, alpha=1.0, beta=0.01)
lda_inf.run(num_iters=1000) lda_inf.save('lda-cvb6')

```
In [31]: model6 = metapy.topics.TopicModel('lda-cvb6')
```

```
In [32]: scorer = metapy.topics.BLTermScorer(model6)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model6.top_k(tid=0, scorer=scorer)]
```

```
Out[32]: [('book', 0.09122978988199469),
          ('art', 0.07774735771650901),
          ('publish', 0.062183910688354826),
          ('film', 0.04770481248847509),
          ('writer', 0.04031926333130394),
          ('radio', 0.03679721092047794),
          ('stori', 0.03289908831482361),
          ('show', 0.03170251906795536),
          ('artist', 0.030566021020771435),
          ('write', 0.029884631186846807)]
```

```
In [33]: scorer = metapy.topics.BLTermScorer(model6)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model6.top_k(tid=1, scorer=scorer)]
```

```
Out[33]: [('music', 0.22116702911523822),
          ('album', 0.11056000297534128),
          ('film', 0.09279232056546949),
          ('band', 0.07919865090338994),
          ('record', 0.07679368053585776),
          ('song', 0.06537724753466923),
          ('play', 0.05010209540299292),
          ('releas', 0.043073652081674636),
          ('perform', 0.0430290796444075),
          ('produc', 0.04244285276787855)]
```

In [34]:

```
scorer = metapy.topics.BLTermScorer(model6)
[(fidx.term_text(pr[0]), pr[1]) for pr in model6.top_k(tid=2, scorer=scorer)]
```

Out[34]:

```
[('championship', 0.09848615031690944),
 ('race', 0.07813697195046698),
 ('finish', 0.0633107972733876),
 ('olymp', 0.06216091018002947),
 ('team', 0.05613488836996568),
 ('world', 0.05476068569019832),
 ('tour', 0.04392028448956867),
 ('won', 0.04306734701956039),
 ('compet', 0.0428066772320037),
 ('champion', 0.038972807451263636)]
```

In [35]:

```
scorer = metapy.topics.BLTermScorer(model6)
[(fidx.term_text(pr[0]), pr[1]) for pr in model6.top_k(tid=3, scorer=scorer)]
```

Out[35]:

```
[('univers', 0.13075887413896226),
 ('research', 0.09974359363207341),
 ('professor', 0.08490185067956768),
 ('scienc', 0.058938681151420085),
 ('institut', 0.050356936044012215),
 ('studi', 0.039614597478368374),
 ('technolog', 0.034733736822351924),
 ('presid', 0.030895362134797377),
 ('phd', 0.03059457265874125),
 ('serv', 0.030507887937343714)]
```

In [36]:

```
scorer = metapy.topics.BLTermScorer(model6)
[(fidx.term_text(pr[0]), pr[1]) for pr in model6.top_k(tid=4, scorer=scorer)]
```

Out[36]:

```
[('elect', 0.10896889436264375),
 ('parti', 0.0820565164972183),
 ('serv', 0.06926968052954008),
 ('minist', 0.05759188461525942),
 ('presid', 0.05213667593509719),
 ('law', 0.04666473580216627),
 ('govern', 0.043787950741343146),
 ('offic', 0.04086710878590989),
 ('court', 0.03732182242149534),
 ('democrat', 0.03415061598505187)]
```

```
In [37]: scorer = metapy.topics.BLTermScorer(model6)
         [(fidx.term_text(pr[0]), pr[1]) for pr in model6.top_k(tid=5, scorer=scorer)]
```

```
Out[37]: [('play', 0.19786651966745958),
          ('season', 0.18355921906020667),
          ('leagu', 0.18101800834422105),
          ('game', 0.13428319489899204),
          ('footbal', 0.12117687259699395),
          ('team', 0.10983019769016403),
          ('coach', 0.09574775920276904),
          ('player', 0.0677646616185766),
          ('club', 0.050191573068551154),
          ('basebal', 0.046543849393005925)]
```

```
In [38]: model6.topic_distribution(0)
```

```
Out[38]: <metapy.stats.Multinomial {0: 0.185898, 1: 0.192525, 2: 0.109879, 3: 0.175248,
         4: 0.191798, 5: 0.144652}>
```

```
In [39]: model6.topic_distribution(1)
```

```
Out[39]: <metapy.stats.Multinomial {0: 0.174398, 1: 0.045119, 2: 0.040860, 3: 0.692994,
         4: 0.018971, 5: 0.027657}>
```

```
In [40]: model6.topic_distribution(1000)
```

```
Out[40]: <metapy.stats.Multinomial {0: 0.015666, 1: 0.012595, 2: 0.810583, 3: 0.012592,
         4: 0.011867, 5: 0.136697}>
```

```
In [43]: model6.top_k(1)
```

```
Out[43]: [(283479, 0.01837600704545592),
          (350829, 0.010710558089839957),
          (148007, 0.009966988540180815),
          (320320, 0.009620704413100957),
          (10135, 0.00942867453648292),
          (353600, 0.007780390642744302),
          (327105, 0.007142555191344174),
          (35374, 0.006953586256792378),
          (337812, 0.005946617354246661),
          (395908, 0.005887835221251912)]
```