In [1]:
```
import numpy as np
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.stop_words import ENGLISH_STOP_WORDS

newsgroups_train = fetch_20newsgroups(subset='train', remove=('headers
vectorizer = CountVectorizer(lowercase=True, stop_words=ENGLISH_STOP_W
                             binary=True,min_df=10, max_df=.04)
X_train = vectorizer.fit_transform(newsgroups_train.data)

X_train.toarray()
docs,words=X_train.nonzero()
print(X_train.shape) #(11314, 10299)
print(len(docs)) #it is N=480590
```

```
(11314, 10299)
480590
```

In [2]:
```
import numpy as np
from tqdm import tqdm
M=11314 #amount of texts
W=10299 #amount of different words
N=480590#total amount of words in the corpus
K=20 #amount of tags
NITER=50


def lda(X_train,alpha,beta,NITER):
    X=X_train.toarray()
    docs,words=X.nonzero() #print(len(docs)) #it is N=480590
    z=[np.random.choice(K) for i in range(N)] #randomly assign tags to

    n_dk=np.zeros(M*K).reshape(M,K) #amount of words in document d ass
    n_kw=np.zeros(K*W).reshape(K,W) #amount of times word w was assign
    n_k=np.zeros(K) #total amount of words assigned to tag k

    for doc, word, cur_z in zip(docs, words, z):
        n_dk[doc, cur_z] += 1
        n_kw[cur_z, word] += 1
        n_k[cur_z] += 1

    for tek_iter in tqdm(range(NITER)):
        for i in range(N):
            n_dk[docs[i],z[i]]-=1
            n_kw[z[i],words[i]]-=1
            n_k[z[i]]-=1
            p = (n_dk[docs[i], :] + alpha) * (n_kw[:, words[i]] + beta
            z[i] = np.random.choice(np.arange(K), p=p / p.sum())

            n_dk[docs[i],z[i]]+=1
            n_kw[z[i],words[i]]+=1
            n_k[z[i]]+=1
    return  n_kw

alpha=np.ones(K)
beta=np.ones(W)


print(alpha[0:10])
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [3]:
```
n_kw=lda(X_train,alpha,beta,NITER)
#print(n_kw)
```

```
100%|██████████| 50/50 [22:01<00:00, 26.38s/it]
```

In [4]:
```
n_kw_sorted=np.argsort(n_kw, axis=1)
#print(vectorizer.vocabulary_.get('car'))
#print(type(vectorizer.vocabulary_))
```

```
my_dict={}
for item in vectorizer.vocabulary_:
    my_dict[vectorizer.vocabulary_[item]]=item

for k in range(K):
    print("tag=",k,end=" ")
    for j in range(10):
        print(my_dict[n_kw_sorted[k,W-1-j]],end=" ")
    print("\n")
```

```
tag= 0 posting news article appreciate newsgroup posted posts reply
btw doubt

tag= 1 current small sorry goes circuit sound usually oh company out
put

tag= 2 key chip phone clipper public encryption keys law security se
cure

tag= 3 war israel jews killed israeli rights land history children m
ilitary

tag= 4 12 11 13 18 16 14 17 25 24 23

tag= 5 file files window code ftp version application graphics runni
ng user

tag= 6 note process single usually reading cases level check unless
result

tag= 7 anybody sorry advance simple figure company reply stuff exact
ly sort

tag= 8 address sounds hear tom small matter wondering nice yeah repl
y

tag= 9 jesus christian bible christians religion man christ word chu
rch saying

tag= 10 soon gun cause guns control gordon banks medical surrender p
itt

tag= 11 card computer video pc memory disk mac monitor hi board

tag= 12 stuff deleted add mention interesting worth cheers flame rea
ding wondering

tag= 13 car bike cars sell price condition sale engine buy nice

tag= 14 game team games play season win players league teams hockey

tag= 15 hi mr 24 40 al 14 id ah mi ad
```

```
tag= 16 went days told saw left started home came took wouldn

tag= 17 guess stuff interesting sorry stay folks anybody hand yeah a
gree

tag= 18 money pay clinton public states federal american care genera
l country

tag= 19 space research nasa university science earth center technolo
gy systems low
```

In [ ]: