In [1]:
```
import numpy as np
import pandas as pd
import stop_words
import os

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.utils import shuffle
```

In [2]:
```
root_folder = './newsdataset/20news-bydate-train'
```

In [3]:
```
# Load some selected topics

docs = []

for doc in os.listdir(root_folder + '/comp.windows.x'):
    if '.' not in doc:
        with open(root_folder + '/comp.windows.x/' + doc, 'r') as f:
            docs.append(f.read())

for doc in os.listdir(root_folder + '/soc.religion.christian'):
    if '.' not in doc:
        with open(root_folder + '/soc.religion.christian/' + doc, 'r') as f:
            docs.append(f.read())
```

In [4]:
```
docs = shuffle(docs)
```

In [5]:
```
max_tokens = 1000

vectorizer = CountVectorizer(ngram_range=(1, 1), max_features=max_tokens, stop_words=stop_words.get_st
op_words('en'))
X = vectorizer.fit_transform(docs)
```

In [6]:
```
lda_model = LatentDirichletAllocation(n_components=2, learning_method='batch')
lda_model.fit(X)
```

Out[6]:
```
LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
             evaluate_every=-1, learning_decay=0.7,
             learning_method='batch', learning_offset=10.0,
             max_doc_update_iter=100, max_iter=10, mean_change_tol=0.001,
             n_components=2, n_jobs=1, n_topics=None, perp_tol=0.1,
             random_state=None, topic_word_prior=None,
             total_samples=1000000.0, verbose=0)
```

In [7]:
```
df = pd.DataFrame(lda_model.exp_dirichlet_component_ , columns = vectorizer.get_feature_names()).T
```

In [8]:
```
keywords = []
for i in range(df.columns.size):
    keywords.append(df[i].sort_values(ascending=False).iloc[:10].index.tolist())
```

In [9]:
```
pd.DataFrame(keywords, index = list(map(lambda x: 'topic_%d' % x, range(2))))
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9
topic_0	god	edu	can	one	will	people	subject	re	jesus	lines
topic_1	edu	com	can	window	subject	lines	file	organization	use	mit