Import necessary dependencies

In [7]:
```
import pandas
from matplotlib import pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
import numpy
from sklearn.feature_selection import chi2
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from collections import Counter
import re
import sqlite3
```

Read in the data

In [8]:
```
train_data = pandas.read_csv("./data/train.csv", header=None)
train_data.head()
```

Out[8]:

	0	1	2
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...

to# saveecessarydatabase data

In [ ]:
```
db = sqlite3.connect('newsclassifier.db')
cat_list = pandas.read_csv('./data/classes.txt', header=None)
cat_list.head()
cat_list.to_sql("category_list", db, if_exists='replace')
```

Data Cleaning

In [9]:
```
train_data.columns = ['category', 'headline', 'content']
train_data.head()
```

Out[9]:

	category	headline	content
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...

Sample 1000 rows

In [4]:
```
train_data_sample = train_data.sample(n = 1000, replace = False, random_state = 123)
train_data_sample.head()
```

Out[4]:

	category	headline	content
30870	4	US Stocks Higher, Helped by Ford Outlook (Reut...	Reuters - U.S. stocks opened higher on Friday\...
7738	1	Judge wants speed on Abu Ghraib evidence	A military judge today warned the US governmen...
25351	2	Sting Pound Lynx Early	Charlotte opens the game with a WNBA-record 21...
74308	4	Cassini snapshots murky moon Titan	The Cassini probe got the first close-up photo...
88346	1	Farewell Yasser Arafat	GAZA CITY, 12 November 2004 - The world will b...

Clean HTML code & news sources from headline

In [5]:
```
import re

def clean(x):
    x = re.sub(r'(&[A-Za-z]+)|\(.*\)', '', x)
    return str(x)

for i, row in train_data_sample.iterrows():
    train_data_sample.at[i, "headline"] = clean(row.headline)
```

Clean news sources from content

In [6]:
```
sources_data = pandas.read_csv("./data/news_sources_clean_v1.csv")

def remove_sources(x):
    x = str(x)
    # print('X OUTSIDE OF LOOP:' + x)
    for i, source in sources_data.iterrows():
        source_list_string = str(sources_data.at[i, 'list'])
        #print('source_list_string:' + source_list_string)
        source_list_stripped = source_list_string.strip()
        #print('source_list_stripped:' + source_list_stripped)

        if source_list_stripped in x:
            # print('x at this point:' + x)
            # print('source_list_stripped:' + source_list_stripped)
            # print('row number: ' + str(i))

            #this doesn't work
            x = x.replace(source_list_stripped, '')
            #regex_expression = re.compile(source.list)
            #x = re.sub(regex_expression, '', x)
    return x

for i, row in train_data_sample.iterrows():
    train_data_sample.at[i, "content_cleaned"] = remove_sources(row.content)
```

Save the new dataframe with cleaned headline and content to database

In [ ]:
```
train_data_sample.to_sql('train_data_sample', db, if_exists='replace')
```

Make a CountVector (Bag of words)

In [7]:
```
# create a CountVectorizer from raw data, with options to clean it
cv = CountVectorizer(min_df = 2, lowercase = True, token_pattern=r'(?u)\b[A-Za-z]{2,}\b',
                         strip_accents = 'ascii', ngram_range = (1, 1),
                         stop_words = 'english')
cv_matrix = cv.fit_transform(train_data_sample.headline).toarray()

# get all unique words in the corpus
vocab = cv.get_feature_names()

# produce a dataframe including the feature names
cv_matrix_df = pandas.DataFrame(cv_matrix, columns=vocab)
```

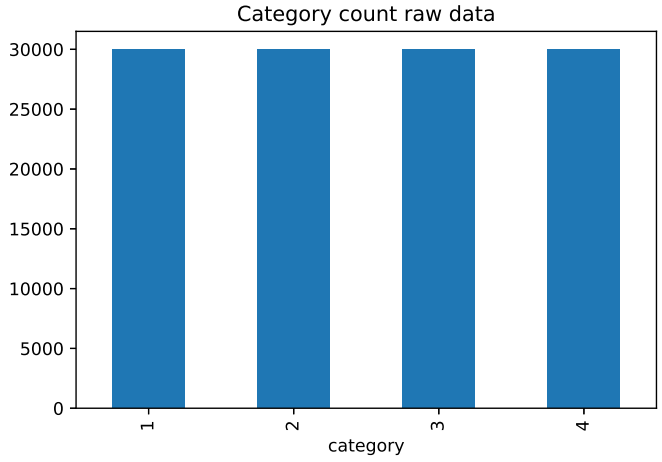
Out[7]:
```
'======================================================================'
```

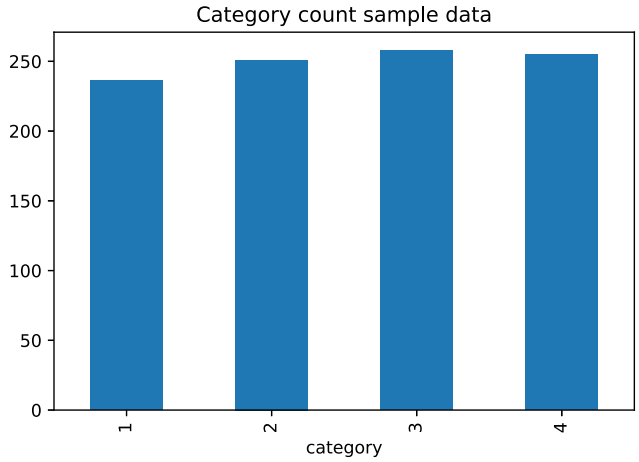
Save the bag of words

In [ ]:
```
cv_matrix_df.to_sql('headline_bagofwords', db, if_exists='replace')
```

Data Exploration

In [8]:
```
# bar plot of the count of unique things in each category
train_data.groupby('category').headline.count().plot.bar(ylim = 0)
plt.title("Category count raw data")
plt.show()
train_data_sample.groupby('category').headline.count().plot.bar(ylim = 0)
plt.title("Category count sample data")
plt.show()
```





The number of unique documents in each category

In [9]:
```
print(pandas.DataFrame(train_data_sample.groupby(['category']).count()))
```

```
          headline  content  content_cleaned
category
1              236      236              236
2              251      251              251
3              258      258              258
4              255      255              255
```

The count of observations and features

In [10]:
```
print("There are {} observations and {} features in this dataset. \n".\
    format(cv_matrix_df.shape[0],cv_matrix_df.shape[1]))
```

```
There are 1000 observations and 893 features in this dataset.
```

A description of the categories

In [11]:
```
categories = train_data_sample.groupby("category")
categories.describe().head()
```

Out[11]:

	headline				content				content_cleaned			
	count	unique	top	freq	count	unique	top	freq	count	unique	top	freq
category												
1	236	236	Bush's Convention Tops Kerry's in Primetime Po...	1	236	235	TAIPEI (Reuters) - The pro-independence party...	2	236	235	TAIPEI (Reuters) - The pro-independence party...	2
2	251	251	Edwards banned from Games	1	251	251	ISTANBUL, Turkey -- Striker Andriy Shevchenko ...	1	251	251	NEW YORK (Reuters) - Lamar Odom supplemented ...	1
3	258	258	Consumer Sentiment Improves in November	1	258	258	The Congress-led UPA government decided on Wed...	1	258	258	The Congress-led UPA government decided on Wed...	1
4	255	255	Arguments conclude in evolution sticker trial	1	255	255	com September 14, 2004, 4:00 AM PT. With the e...	1	255	255	AP - The on Thursday filed the first case in ...	1

WordCloud/TagCloud of the top words in the headlines

In [12]:
```
# prepare the dictionary to be used in wordcloud
word_count_dict = {}
for word in vocab:
    word_count_dict[word] = int(sum(cv_matrix_df.loc[:, word]))
```

In [13]:
```
# generate a word cloud image with top 100 words and 80% horizontal:
wordcloud = WordCloud(max_words=100, prefer_horizontal=0.8, background_color='white').\
            generate_from_frequencies(word_count_dict)

# display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Plots of the data

Bar plot of the top word counts

In [18]:
```
from collections import Counter

counter = Counter(word_count_dict)

freq_df = pandas.DataFrame.from_records(counter.most_common(20),
                                        columns=['Top 20 words', 'Frequency'])
freq_df.plot(kind='bar', x='Top 20 words');
```