# NLP Homework 1: Corpus Analysis

Emma Ozias

## 1. Dataset

I chose the “ag_news” dataset from hugging face. My dataset contains news headlines rather than documents. Each headline falls into one of four different categories: World, Sports, Business, or Science/Technology. I thought this would be interesting to analyze to see which words are most commonly associated with each of the four labels. I enjoy following current events, so I have read many news headlines which is why I chose this dataset. I chose to use only 7000 headlines from the train subset of “ag_news”

```
Total headlines in each category:
Label 0 (World) count: 1861
Label 1 (Sports) count: 1683
Label 2 (Business) count: 1636
Label 3 (Science/Technology) count: 1820


Total word count in each category:
Label 0 (World) count: 47904
Label 1 (Sports) count: 40178
Label 2 (Business) count: 41867
Label 3 (Science/Technology) count: 45318


Average word count in each headline by category:
Label 0 (World) count: 25.740999462654486
Label 1 (Sports) count: 23.872846108140227
Label 2 (Business) count: 25.591075794621027
Label 3 (Science/Technology) count: 24.9
```

## 2. Methodology

Outline of my code: load dataset -> split dataset into a list of headlines and a list of labels -> preprocess each headline -> convert to bag of words format -> count how many times each word is seen in a specific label and how many times it was seen in all other labels -> calculate log likelihoods and perform Naïve Bayes -> output the top 10 words from each label -> perform LDA and print out 20 topics -> determine the top 3 topics for each of the labels.

More details (including packages and libraries used):

- Preprocessing
    - Make all text lowercase
    - Remove punctuation
    - Lemmatize the text
    - Remove stop words
    - Remove numbers
    - Remove other common and unimportant words such as “reuters”, “washingtonpost”, “ap”, “new”, “say”, as well as others
- NLTK library to tokenize the headlines when making my bag of words model
- Collections library to create data structures for storing of each word in a specific label, and the count of the same word in other labels
- Math library for computing the log likelihoods
- Gensim library and corpora package for the LDA part of my code
- Finally, I used Chat GPT to help me with determining the top three topics for each of the four labels

One important decision that I made while writing my code was to remove numbers during preprocessing. Before I did this, I found that numbers appeared in my list of the top 10 words for some of my labels.

# 3. Results and Analysis

```
Top 10 words for label 0 (World):
('sadr', 8.783397214635759)
('shrine', 8.700282307925253)
('sharon', 8.264357992332378)
('gaza', 7.966313132845052)
('burundi', 7.783991576051097)
('wound', 7.750090024375416)
('blockade', 7.750090024375416)
('kathmandu', 7.696980199061468)
('ariel', 7.678631060393271)
('palestinian', 7.621472646553322)


Top 10 words for label 1 (Sports):
('hewitt', 7.387593402134678)
('invitational', 7.307550694461142)
('mets', 7.158018960490178)
('woods', 7.125229137667187)
('patterson', 7.125229137667187)
('nl', 7.091327585991506)
('pitcher', 7.019868622009361)
('agassi', 7.019868622009361)
('angels', 6.902085586352977)
('swimmer', 6.859525971934182)


Top 10 words for label 2 (Business):
('aspx', 8.606458055763673)
('fullquote', 8.606458055763673)
('quickinfo', 8.606458055763673)
('mortgage', 7.446691343942232)
('qantas', 7.391121492787422)
('mutual', 7.362133955914169)
('parmalat', 7.2369708129601635)
('hare', 7.203069261284482)
('delta', 7.131610297302338)
('treasury', 7.013827261645954)


Top 10 words for label 3 (Science/Technology):
('nasa', 8.179401082785272)
('planet', 7.474954346971395)
('newsfactor', 7.328350872779519)
('micro', 7.301682625697358)
('saturn', 7.2742836515092435)
('mozilla', 7.156500615852861)
('spacecraft', 7.156500615852861)
('cassini', 7.0919620947152895)
('amd', 7.0919620947152895)
('browser', 7.058060543039607)
```

The image above is printed out after naïve bayes is completed. This is a list of the top 10 words for each category. I think this is very accurate for each category. For example, in the "World" category, the names of political and world leaders are listed as well as "gaza" and "palestinian". These words are extremely prevalent in the news right now due to the Israel and Palestine conflict.

```
Topic: 0
Words: 0.009*"quot" + 0.006*"two" + 0.006*"us" + 0.004*"iraq" + 0.004*"internet" + 0.004*"broadband" + 0.003*"company" + 0.003*"cost" + 0.003*"first" + 0.003*"year"

Topic: 1
Words: 0.006*"get" + 0.005*"hamm" + 0.004*"two" + 0.004*"afp" + 0.004*"gold" + 0.004*"back" + 0.004*"ple" + 0.003*"athens" + 0.003*"warn" + 0.003*"google"

Topic: 2
Words: 0.006*"first" + 0.005*"profit" + 0.005*"price" + 0.005*"oil" + 0.004*"iraq" + 0.004*"company" + 0.004*"friday" + 0.004*"yesterday" + 0.003*"time" + 0.003*"british"

Topic: 3
Words: 0.007*"gold" + 0.006*"us" + 0.006*"win" + 0.005*"rule" + 0.004*"make" + 0.004*"first" + 0.004*"charley" + 0.004*"hurricane" + 0.003*"take" + 0.003*"china"

Topic: 4
Words: 0.021*"najaf" + 0.012*"al" + 0.012*"shrine" + 0.011*"iraq" + 0.010*"sadr" + 0.009*"cleric" + 0.007*"iraqi" + 0.006*"plan" + 0.006*"shiite" + 0.006*"holy"

Topic: 5
Words: 0.016*"win" + 0.014*"athens" + 0.011*"olympic" + 0.010*"gold" + 0.008*"women" + 0.008*"phelps" + 0.007*"men" + 0.006*"team" + 0.006*"saturday" + 0.006*"meter"

Topic: 6
Words: 0.012*"stock" + 0.009*"google" + 0.009*"athens" + 0.007*"www" + 0.007*"http" + 0.007*"target" + 0.007*"ticker" + 0.007*"com" + 0.007*"investor" + 0.007*"aspx"

Topic: 7
Words: 0.014*"oil" + 0.013*"stock" + 0.013*"google" + 0.012*"price" + 0.010*"share" + 0.010*"company" + 0.006*"us" + 0.006*"market" + 0.005*"crude" + 0.005*"public"

Topic: 8
Words: 0.007*"us" + 0.004*"two" + 0.004*"court" + 0.004*"share" + 0.004*"world" + 0.004*"quot" + 0.004*"nortel" + 0.003*"game" + 0.003*"unite" + 0.003*"percent"

Topic: 9
Words: 0.007*"make" + 0.007*"game" + 0.006*"oil" + 0.005*"price" + 0.005*"bank" + 0.004*"security" + 0.004*"stock" + 0.004*"report" + 0.004*"find" + 0.003*"first"

Topic: 10
Words: 0.009*"price" + 0.006*"us" + 0.005*"win" + 0.005*"two" + 0.004*"afp" + 0.004*"year" + 0.004*"aug" + 0.004*"one" + 0.003*"take" + 0.003*"unite"

Topic: 11
Words: 0.008*"rebel" + 0.007*"chavez" + 0.007*"cital" + 0.006*"venezuela" + 0.006*"nepal" + 0.006*"president" + 0.005*"kathmandu" + 0.005*"blockade" + 0.005*"maoist" + 0.005*"people"

Topic: 12
Words: 0.010*"world" + 0.009*"gold" + 0.008*"olympic" + 0.007*"medal" + 0.007*"win" + 0.006*"quot" + 0.006*"athens" + 0.005*"record" + 0.004*"two" + 0.004*"game"

Topic: 13
Words: 0.007*"oil" + 0.007*"us" + 0.005*"price" + 0.005*"million" + 0.005*"stock" + 0.005*"week" + 0.004*"company" + 0.004*"year" + 0.004*"bank" + 0.004*"first"

Topic: 14
Words: 0.005*"year" + 0.004*"south" + 0.004*"thursday" + 0.004*"inc" + 0.004*"second" + 0.004*"quot" + 0.003*"game" + 0.003*"get" + 0.003*"com" + 0.003*"africa"

Topic: 15
Words: 0.008*"lead" + 0.006*"run" + 0.006*"hit" + 0.005*"two" + 0.004*"face" + 0.004*"game" + 0.004*"plan" + 0.004*"homer" + 0.003*"saturday" + 0.003*"one"

Topic: 16
Words: 0.008*"plan" + 0.007*"attack" + 0.006*"kill" + 0.005*"us" + 0.005*"quot" + 0.005*"one" + 0.004*"state" + 0.004*"unite" + 0.004*"rally" + 0.004*"pakistan"

Topic: 17
Words: 0.007*"microsoft" + 0.006*"xp" + 0.006*"update" + 0.006*"security" + 0.006*"windows" + 0.005*"two" + 0.005*"find" + 0.004*"win" + 0.004*"iraq" + 0.004*"attack"

Topic: 18
Words: 0.013*"athens" + 0.012*"gold" + 0.011*"olympic" + 0.009*"win" + 0.007*"game" + 0.007*"first" + 0.006*"medal" + 0.005*"batteries" + 0.005*"com" + 0.005*"ple"

Topic: 19
Words: 0.004*"company" + 0.004*"first" + 0.004*"team" + 0.004*"game" + 0.004*"three" + 0.004*"world" + 0.004*"sox" + 0.003*"second" + 0.003*"thursday" + 0.003*"profit"
```

The image above displays 20 topics from the completion of LDA. I notice that most of the words in each category are part of the same category, or often used together. For example, the words: profit, price, and oil in topic 2 are often seen together in a news headline. However, I do not understand the relation between some of the words in the same category. For example, I do not understand how internet and broadband are related to us and iraq (two countries) in topic 0.

```
Top three topics for each label:
Label 0 (World): 4, 16, 11
Label 1 (Sports): 5, 18, 15
Label 2 (Business): 7, 6, 13
Label 3 (Science/Technology): 19, 7, 18
```

This image shows the top three topics for each label. I am not surprised by these results. Each of the topics fit extremely well in the labels that they are listed under. For example, topics 4, 16, and 11 are listed as the top three topic matches to the World category. All of these topics include the names of world leaders or political figures, and a place in the

world (iraq, us, nepal, etc.). Also, these topics include words in the top 10 lists for their respective categories.

## 4. Discussion

I learned that each category in my dataset has words that are specific to that category. When using the Naïve Bayes model of probability, you calculate the likelihood that a specific word is in one label minus the likelihood that that word is in any of the other labels. This allows you to lower the probability of common words from the top 10 in each dataset leaving you with a highly specialized top 10 words list. For example, any of the top 10 words in the World category are very unlikely to be found in the other 3 categories. Additionally, the LDA model is able to put some of these highly specialized words into the same topic. Then, when identifying the top three topics for each category, I saw that each of these top topics included some of the top 10 words identified by the Naïve Bayes model.

I learned a lot from this assignment. I started out struggling, so I went to office hours to receive help from Professor Wilson. My first challenge was figuring out how to breakup and preprocess my dataset. I wrote a preprocessing function but could not pass the dataset to my function due to incompatible types. I needed to pass a string, but I was trying to pass the dataset object. Once I made it past this challenge, I was able to preprocess my text and create the bag of words model. Then, I attempted to implement the Naïve Bayes, but found myself struggling again. Professor Wilson helped me get past that by breaking down each step of the Naïve Bayes section. I found lots of great information on LDA online, and I was able to implement that. For the last part of my code, I struggled again. I spent multiple hours trying to determine the top three topics for each label. Eventually, Chat GPT was able to help me get past that. Overall, the biggest thing that I learned from the completion of this assignment is that asking for help is not a bad thing. I have been embarrassed to ask for help in other classes that I have taken in the past. Also, I learned a lot about Python in this assignment. This is only my second ever Python assignment, and I did not know anything about the collections library. I did end up using the collections library to build two different data structures. I can use that library again in the future. Additionally, I learned how to use gensim to perform LDA which could be helpful in the future for this class as well.