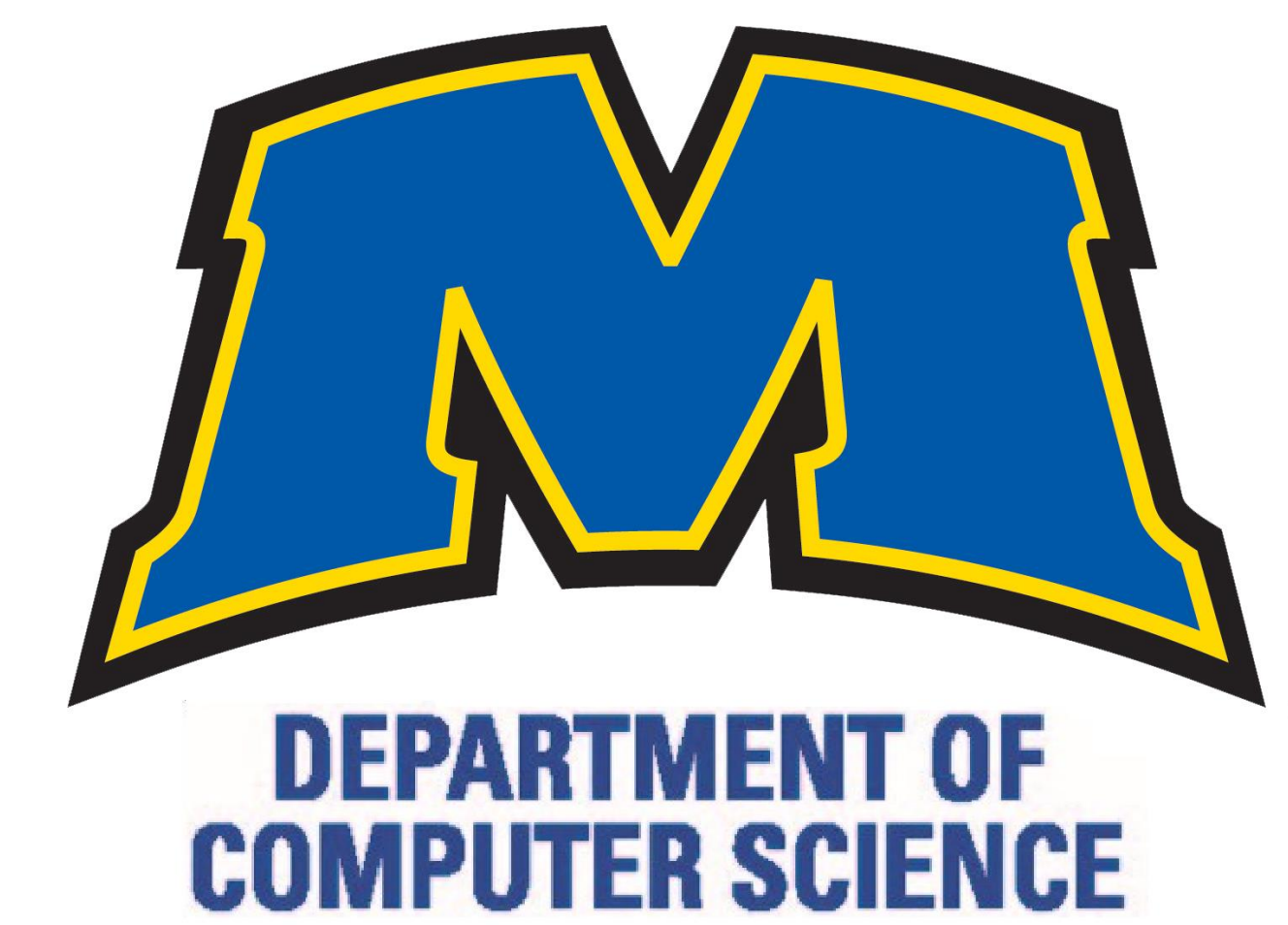




# Political Orientation Detection & Machine Learning



Kenneth May  
Advisor: Dr. Ashraf Aly  
Department of Computer Science

## MOTIVATION

Motivation-  
The motivation for this project comes from the apparent visibly deepening political divide in the United States between right and left-leaning individuals; "conservatives" and "liberals". This rift is pervasive, and can be seen throughout nearly all forms of media – ranging all the way from comedy to sports.  
In today's climate it seems that people of all sides of the political spectrum are pulling away from one another, focusing on their disagreements rather than trying to find workable compromises.

This is largely due to **confirmation bias**, a psychological term that describes a person's tendency to make generalizations based on information they agree with.

Rather than put forth the intellectual effort required debate the issues at hand, people are content to only listen to others who share their perspectives.



## HYPOTHESIS

I expect to see that after a baseline of politically biased articles are scanned and their most common words determined that the algorithm described will correctly predict the basic political leanings of the contents of new articles scanned.

To compare the new articles and buzzword counts, a variety of methods will be used, such as:

- Article percentage of buzzwords ( $\frac{\# \text{ buzzwords used in Article}}{\text{total \# of words in Article}}$ )
- Word bag percentage of buzzwords ( $\frac{\# \text{ buzzwords used in Article}}{\text{total \# of Buzzwords}}$ )
- Right vs. Left-Wing buzzword counts
- A combination of the above.

As the project is still in progress, this hypothesis has not yet been tested, though progress with the rest of the algorithm is in full-swing.

## RESULTS

1	comey:19	1	comey:10
2	investigation:15	2	law:8
3	clinton:7	3	president:7
4	people:6	4	more:7
5	trump:6	5	james:6
6	russia:6	6	over:5
7	fbi:6	7	election:5
8	campaign:5	8	clinton:4
9	transparency:4	9	comeys:4
10	tapper:4	10	justice:4
11	comeys:3	11	undermining:3
12	right:3	12	former:3
13	under:3	13	himself:3
14	reid:3	14	trumps:3
15	book:3	15	reputation:3
16	partisans:3	16	others:3
17	time:3	17	crimes:3
18	president:3	18	rule:3
19	congress:3	19	democratic:3
20	russian:3	20	americas:3

Fig. X. The first 20 lines output by the program after presenting it one cnn.com opinion article (left) and one foxnews.com opinion article, both about former FBI director James Comey.

## INTRODUCTION

Introduction-

- Confirmation Bias – the tendency to process information by looking for information that is consistent with one's existing beliefs.
- This is the basic idea behind the algorithms in applications such as Facebook and Google – they determine what a user likes and reinforce it.
- My goal is to do the opposite – find out what a user likes and give them the opposite.
- A key aspect of this project is algorithm development and data analysis.
- The implementation of a "bag-of-words" algorithm is used in this project to design an application that will allow users to enter an article, then be told how that article lines up politically.
- A baseline of 30-50 articles from left and right-wing news sites will be used.
- This will serve as the initial baseline from which the rest of the algorithm will be built.

## EXAMPLE FUNCTIONS

```
public static void mostCommonWords(List<String> listofStrings, List<String> stopWords, String stopWordsString, Map<String, Integer> articleMap, String newFilename) throws IOException {
    Writer writer = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(new File(newFilename)), "UTF-8"));
    int occurrences;
    boolean stopWordsBool;

    // Loop through the list
    for(String str : listofStrings){
        stopWordsBool = stopWords.contains(str);
        occurrences = Collections.frequency(listofStrings, str);

        // If the word occurs more than once and is not in the stop words list, add it to map
        if (!stopWordsBool){
            articleMap.put(str, occurrences);
        }
    }

    List<String> sorted = sortedByFrequency(articleMap);
    writeToFile(newFilename, stopWordsString, articleMap);

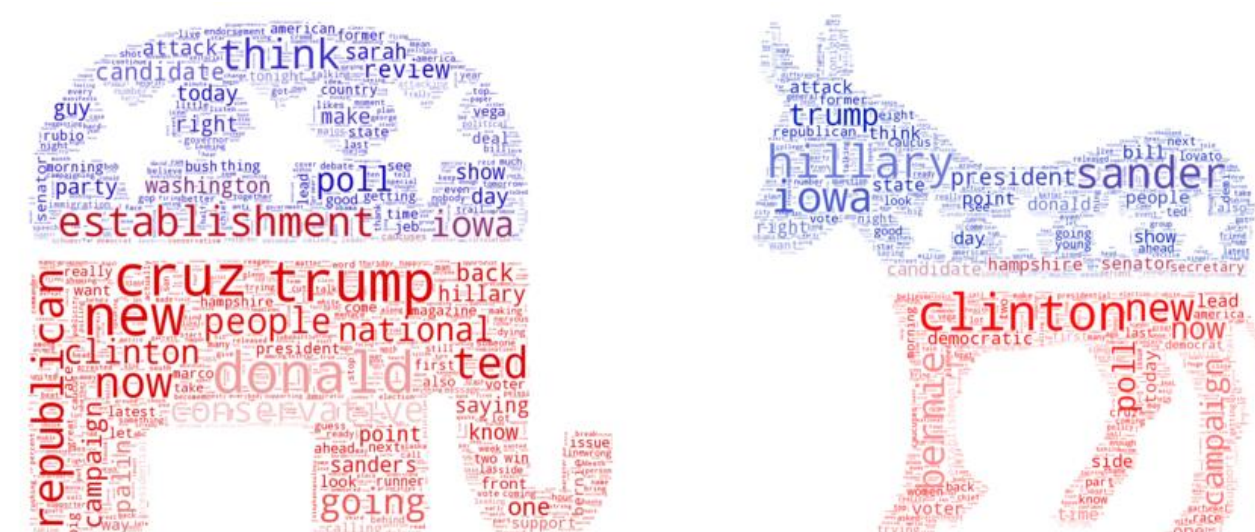
    writer.flush();
    writer.close();
}
```

Fig. "mostCommonWords" function. Takes two lists of strings (one that contains the article's words and one that contains "stop words"), two strings (a string of stop words and the file name of the output file) and a Map of the article and determines the most common words that appear in the article. It then calls a method to write the appropriate information to the output file.

## METHOD

The main method used in this project is a "bag-of-words" method. This method involves using word association to compare the words of the article to. Essentially, this method compares the text of an article to a bag of buzz words, in order to determine how biased an article is, based on how frequently the politically-charged buzzwords are used.

Versions of the bag-of-words method have been used in similar capacities, most of which were intended to assist researchers in determining the political leanings of Twitter users.



This method will instead be used on news-related articles, and as such requires a more varied list of buzz words, since "Tweets" are limited by a strict number of characters rather than whole words.

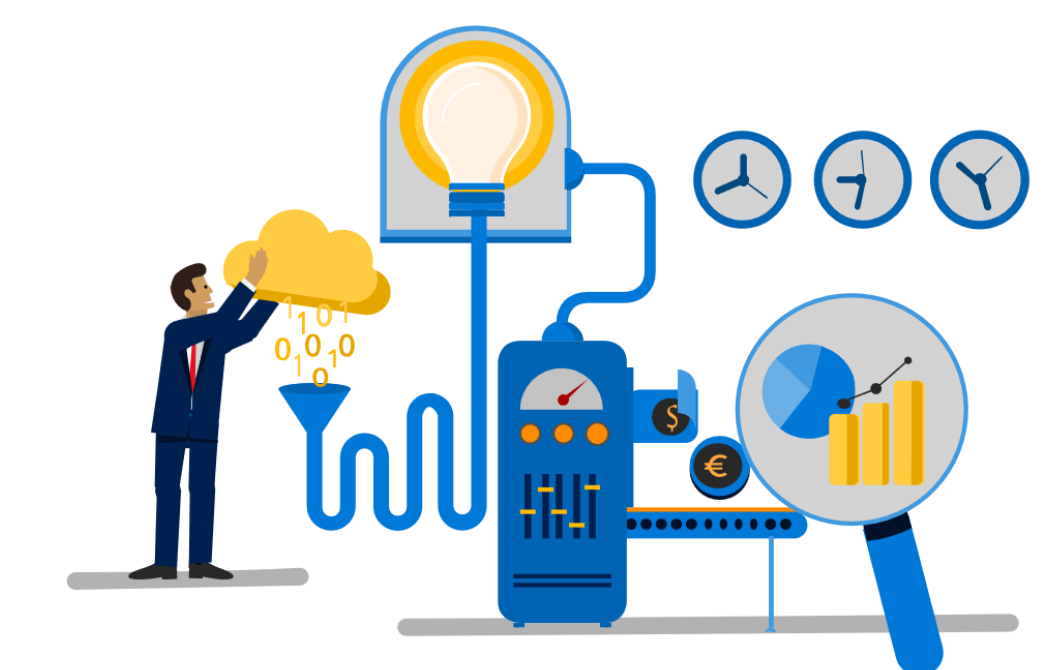
- The first step is to create a baseline: Scan in the text of a politically-aligned article and determine the most common words for that article. Each word and its frequency count are added to an output file, which will be the buzzword reference file later on.
- After a baseline is determined, the new article will be scanned to determine how often buzzwords from the right- and left-wing buzzword files are used, in order to gauge the political leanings of said article.

```
public static void writeToFile(String fileOutName, String stopWords, Map<String, Integer> frequencyMap) throws IOException, FileNotFoundException {
    List<String> sorted = sortedByFrequency(frequencyMap);
    Writer writer = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(new File(fileOutName)), "UTF-8"));
    for (String s: sorted){
        if(!stopWords.contains(s)){
            writer.write(s + " " + frequencyMap.get(s) + System.LineSeparator()); // write words from sorted to a file
        }
    }
    writer.flush();
    writer.close();
}
```

Fig. "writeToFile" – the method that writes the appropriate output to the output file. This method is called in the previous figure, and takes two strings (the file name of the output file and a string of stop words) and a Map of the words/associated frequencies of that word as parameters.

## FUTURE WORK

- Some aspects of the bag-of-words approach are not ideal.
  - I plan on focusing on researching and building some other algorithms that may detect bias using more than just singular words, and instead incorporating things such as phrases, source and author.
- Some other useful features to add to this project would be things like:
  - A user account system
  - User political alignment test / selector
  - Expanded bias spectrum (rather than just left/right)
  - Database
  - Website
  - Mobile application
  - Recommended reading
  - Article archive / current news
- Using these features and more, I plan to create a comprehensive application that will poll a user to determine their political leanings, recommend articles to them that do not align with their political point of view and manage the articles in the system such that newly scanned articles are recommended sooner (as they are conceivably more relevant), delete articles after they get too old and constantly refine itself to keep up with changing jargon.



## REFERENCES

- Confirmation Bias - Definition, Background, History, & Facts." Encyclopedia Britannica.
- Conover, Michael D., et al. "Predicting the political alignment of twitter users." Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. IEEE, 2011.
- Hilton, Steve "Steve Hilton: Without A Comey Prosecution, There's No Hope For Healing America". 2018. Fox News. <http://www.foxnews.com/opinion/2018/04/19/steve-hilton-without-comey-prosecution-there-s-no-hope-for-healing-america.html>.
- Iyyer, Mohit, et al. "Political ideology detection using recursive neural networks." Proceedings of the Association for Computational Linguistics. 2014.
- Lizza, Ryan. "James Comey's Unforgivable Error" 2018. CNN. <https://www.cnn.com/2018/04/19/opinions/comeys-unforgivable-error-lizza/index.html>.
- Pennacchiotti, Marco, and Ana-Maria Popescu. "A Machine Learning Approach to Twitter User Classification." Icwsm 11.1 (2011): 281-288.