Article Clustering on the Basis of Content Bias

Shashwat Chandra (13111059)
under the guidance of
Dr. Arnab Bhattacharya

Abstract

There has been a lot of work on classifying articles on the basis of their content. Most search engines are proficient at displaying content relevant to the given search term. However, given a search term, this work aims to cluster relevant articles on the basis of the bias in their content. This work discusses feature-vector generation, clustering, and cleaning of the results of content-bias classification. Different clustering mechanisms are discussed and demonstrated. The results are tested with articles clustered by various users, and show interesting results. This project is my CS697 course project.

1 Introduction

There has been a great deal of work on classifying articles on their basis of their content. Most search engines are proficient at displaying relevant content from a large collection of articles on the basis of a search term. However, within-topic classification is a problem much less studied. This project aims to perform clustering on articles within the same category content-wise on the basis of the bias of the content. A good content-bias classifier can help take a large leap towards building a semantic-web in general, and improving search-result relevance and search-relevance ease-of-use in particular.

For example, given a search term like *'abortion'*, we wish to be able to cluster *pro-life* articles in a different cluster than *pro-choice* articles. Depending on the number of clusters, we also wish to be able to create a separate cluster for *unbiased articles* like news articles. A more general search term like *'security'* should lead to clusters loosely related to *home security*, *national security*, *computer security*, etc.

2 Data Collection

Since this project originally started as a hack for **Yahoo!** Hack-U, the dataset I am using is the **Yahoo! Voices**[1] article repository. I used a webcrawler (marked as a crawler, obeying 'robots.txt') to collect articles on the following keywords:

Keyword	Number of articles
Abortion	986
Alcohol	992
Barack Obama	971
Congress	980
Evolution	1003
Feminism	849
Gun Rights	868
Homosexuality	970
NATO	930
Republican	964
Security	1000
Terrorism	970

The Yahoo Query Language[2], Python (SciPy and SciKit), and Mathematica were used to perform the analysis and generate results.

3 Feature Vectors

The features I looked at are ones that will be able to give us information on article bias, author prejudice, article neutrality, or author emotion. I tried to encapsulate this information using the following features.

3.1 Biased Word Frequency in Article

This is a frequency count of the number of occourences of words that are present in bigoted/prejudiced articles. This list was collected from [3], a semantic reasoning website. The website divides these words in three categories:

- Religious/Moral Terms (like *moral, heathen, unnatural*),
- Judgment Terms (like *unfair, deserve, fault*), and
- Social Terms (like *accuse, shame, appropriate, decency*).

This feature should give us an indication of how prejudicial the author of the article is. Future work may be done to create a list of biased words by performing supervised analysis on a corpus of bigoted articles.

3.2 Wikipedia's Words to Watch

Wikipedia works on the firm belief that there are words and expressions that should be used with care, becuase they may introduce bias[4]. Wikipedia strives to be a neutral

encyclopedia, and articles on Wikipedia generally end up being unbiased, and have neutral points of view[5][6]. The categories of words that Wikipedia says must be used with care are:

- **Words that may introduce bias**
  - Puffery: *legendary, outstanding, renowned, notable, respected, ...*
  - Contentitious labels: *fundamentalist, terrorist, freedom fighter, myth, bigot, ...*
  - Unsupported attributions: *most feel, some people say, many are of the opinion, research has shown, ...*
  - Expressions of Doubt: *supposed, apparent, purported, alleged, ...*
  - Editorializing: *notably, interestingly, essentially, it should be noted, happily, untimely, ...*
- **Expressions that lack precision**
  - Euphemisms: *gave his/her life, an issue with, ethnic cleansing, sightless, ...*
  - Cliches and Idioms: *lion's share, gild the lily, take the plunge, twist of fate, ...*
  - Unspecified space/time references: *recently, formerly, traditionally, somewhere, the big city, ...*
- **Vulgarities, obscenities, and profanities**

This is a frequency count of the number of occourences of such words, leading to an unbiased article indicator.

3.3 Featured Article Count

Yahoo! selects certain articles as Featured Articles, to go on their websites' front pages. These articles are well-written, and well-researched. It is reasonable to expect Yahoo! to select articles that are not controversial to be on the title pages. This attribute shows the number of featured articles written by the author of this article, since that would reflect the trustworthiness of the author.

3.4 Sentiment of Text

Every sentence in the article that is related to the topic is run through an off-the-shelf sentiment analysis tool. The result the tool returns is loosely indicative of the sentiment of the article towards the topic, giving us an emotional indicator. One thing that must be noted about this feature is that sentiment analysis indicates the sentiment towards

the topic, not the direction in which the sentiment is generated, and so this tool cannot tell us whether the author is angry towards the topic, or against the topic. I used AlchemyAPI[7] to perform sentiment analysis on the articles.

3.5 Number of Comments in Current Article

The number of comments generated by an article is generally expected to be indicative of the degree of controversy generated by the article. Thus, this feature is an article-specific indicator of the controversiality of the article.

3.6 Number of Comments in Recent Articles

This is a broader indicator than the one above, and hopefully shows us how controvertial the author is. Thus, this feature is an author-specific indicator of the controversiality of the author.

3.7 Biased word frequency in comments

In the previous attributes related to the comments, I have assumed that an article more-commented on is more controvertial. This attribute actually looks at the content in the comments and looks at the number of biased/bigoted words[3] that appear per comment. This is used as an indicator of the content in the article.

3.8 Topic Synonym/Antonym Count

This attribute looks at the words in the search term and the topic, looks at the synset of these words, and looks for such words in the text. Calculating the value of this feature requires language-specific resources. This count can basically give us an indication of whether the articles are biased towards a particular direction (with respect to the search term) or not. The NLTK library was used for this.

Selecting Features

Looking at all the above indicators, I decided to pick the following attributes:

1. Biased Word Frequency in Article
2. Wikipedia's Words to Watch
3. Featured Article Count
4. Sentiment of Text

5. Number of Comments in Recent Articles

The reasons for picking these attributes, center around the fact that I want this clustering approach to be quick. We want to provide clustering as a service to users, and the waiting time after searching needs to be minimized. I don't wish to use language-intensive resources, because that would slow down the feature-vector generation process, and also be language-specific. Wordlists, as used in the selected features, can be replicated in a new language with ease, however, the degree of language-specific information to calculate synonyms and antonyms for any search term is high. Additionally, given a choice between features like *Number of comments in the current article* and *Number of comments in recent articles*, I am choosing the feature that gives us more information on the article author, since the bias in an article is generally a product of the bias of the author.

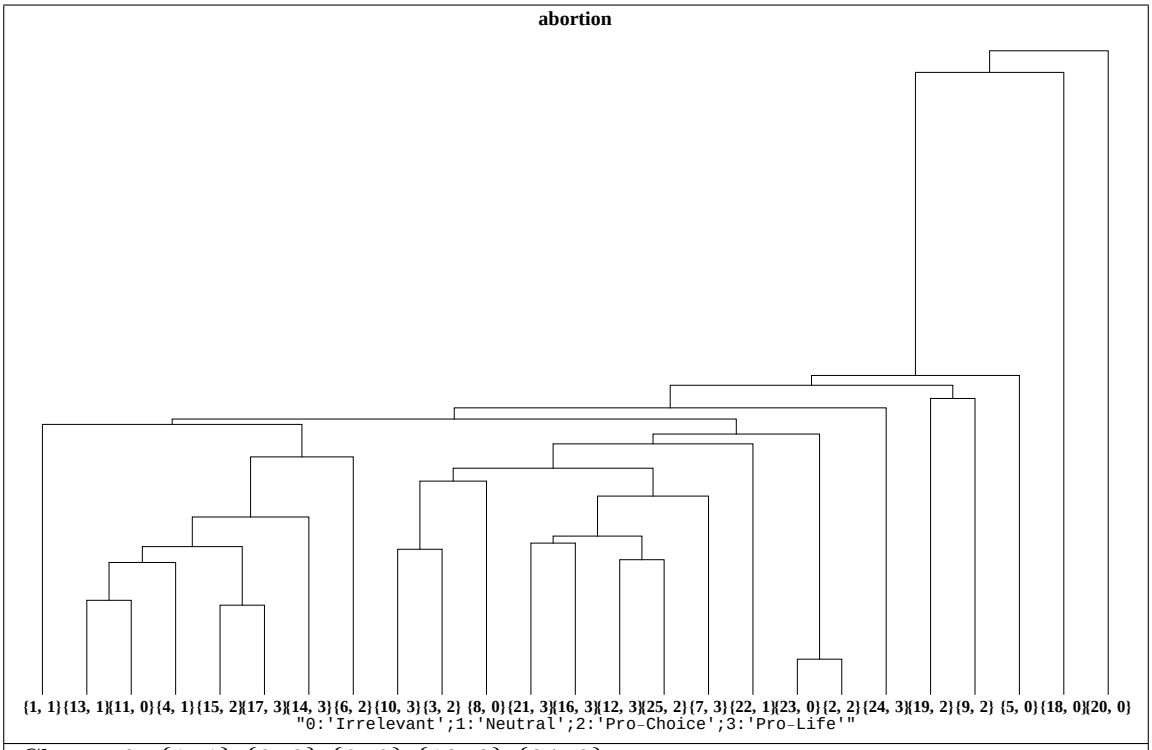
4 Clustering

Two methods of clustering were considered for this project: *K-Means clustering*, and *Hierarchical Clustering*. Intuitively, and for the sake of testing the results, one would expect Hierarchical clustering to be more apt than K-Means. Hierarchical Clustering gives us a better indication of the number of clusters that need to be generated, which is a required input for K-Means. K-Means also performs more poorly with outliers, and outliers are definitely expected in a technique such as this. Hierarchical clustering can be used to detect and remove these outliers. Additionally, hierarchical clustering allows us to see which articles are closer together and use that to judge the accuracy of our features, making it easier to test as well.

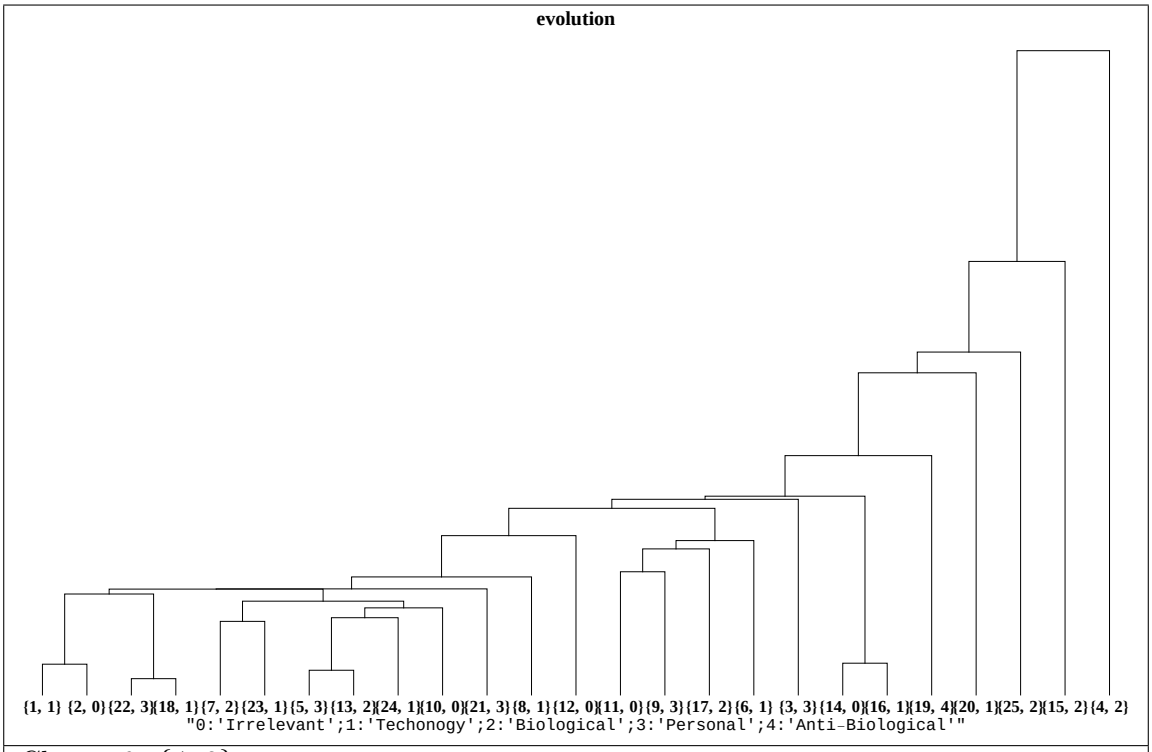
Thus, I expected, and hoped hierarchical clustering would perform better than K-Means. The value of K in K-Means was taken from 3 through 6. For hierarchical clustering, the distance function I have used is *Euclidean Distance*. The linkage function I have used is *Single Linkage*.

5 Results

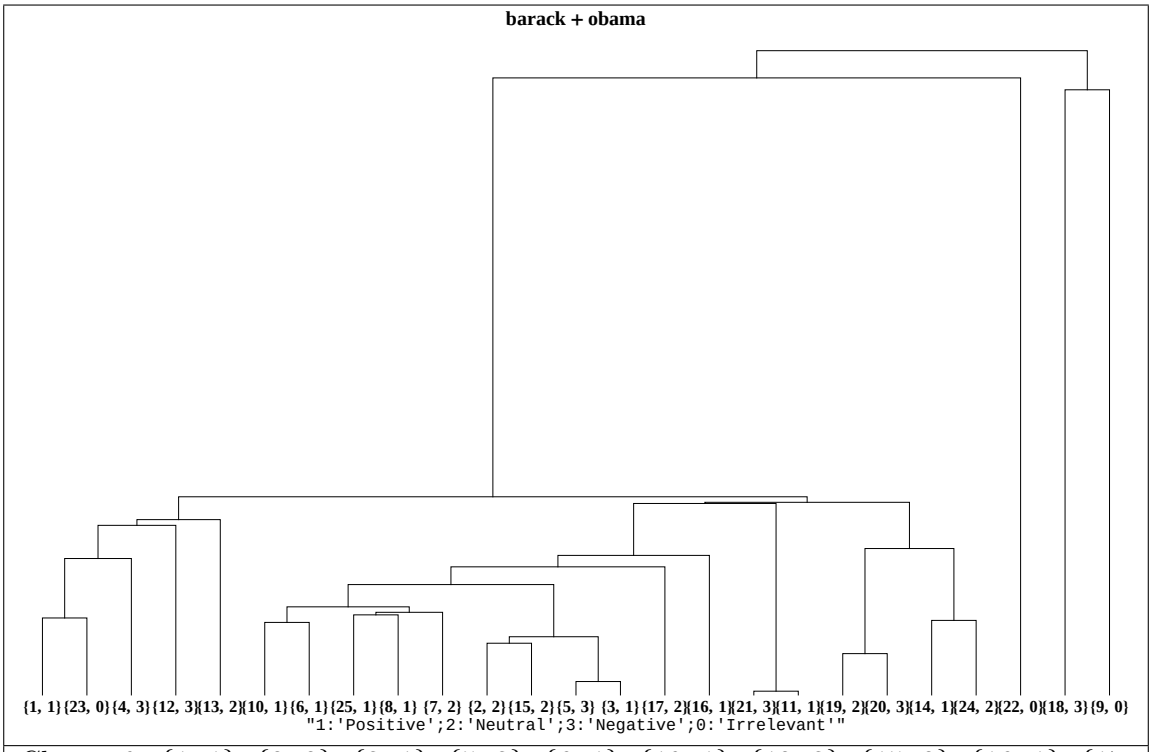
To better understand the results generated via hierarchical clustering, I selected a subset of 25 articles per topic to perform clustering on at the start. These results are displayed as dendrograms below. Each article is represented as a tuple of two numbers: {*article number, cluster number*}. The cluster numbers have been given by human annotators.



Cluster 0: {1, 1} {3, 2} {8, 0} {10, 3} {24, 3}
Cluster 1: {2, 2} {7, 3} {12, 3} {16, 3} {21, 3} {22, 1} {23, 0} {25, 2}
Cluster 2: {20, 0}
Cluster 3: {4, 1} {5, 0} {6, 2} {9, 2} {11, 0} {13, 1} {14, 3} {15, 2} {17, 3} {18, 0} {19, 2}



Cluster 0: {4, 2}
Cluster 1: {6, 1} {9, 3} {11, 0} {17, 2} {19, 4} {20, 1} {25, 2}
Cluster 2: {1, 1} {2, 0} {8, 1} {12, 0} {18, 1} {22, 3}
Cluster 3: {3, 3} {10, 0} {14, 0} {15, 2} {16, 1} {21, 3}
Cluster 4: {5, 3} {7, 2} {13, 2} {23, 1} {24, 1}



Cluster 0: {1, 1}, {2, 2}, {3, 1}, {5, 3}, {6, 1}, {10, 1}, {12, 3}, {15, 2}, {16, 1}, {17, 2}, {25, 1}
Cluster 1: {4, 3}, {7, 2}, {8, 1}, {11, 1}, {13, 2}, {21, 3}, {23, 0}
Cluster 2: {9, 0}, {18, 3}
Cluster 3: {14, 1}, {19, 2}, {20, 3}, {24, 2}
Cluster 4: {22, 0}

We see that K-Means generally performs worse than hierarchical clustering, and we can clearly see the merges, as expected of hierarchical clustering.

6 Cleaning

As can be seen above, there are many cases where an article is very far apart from the others, and is eventually clustered. These articles are clearly outliers for the dataset and feature-set we are using. To remove them, the following algorithm was used:

Calcuate α-mean using the middle $(1 - 2\alpha)$ merging thresholds

if *merging threshold* $<$ (α-mean + δ) **and** *one of the clusters is a single article*

- Remove offending article

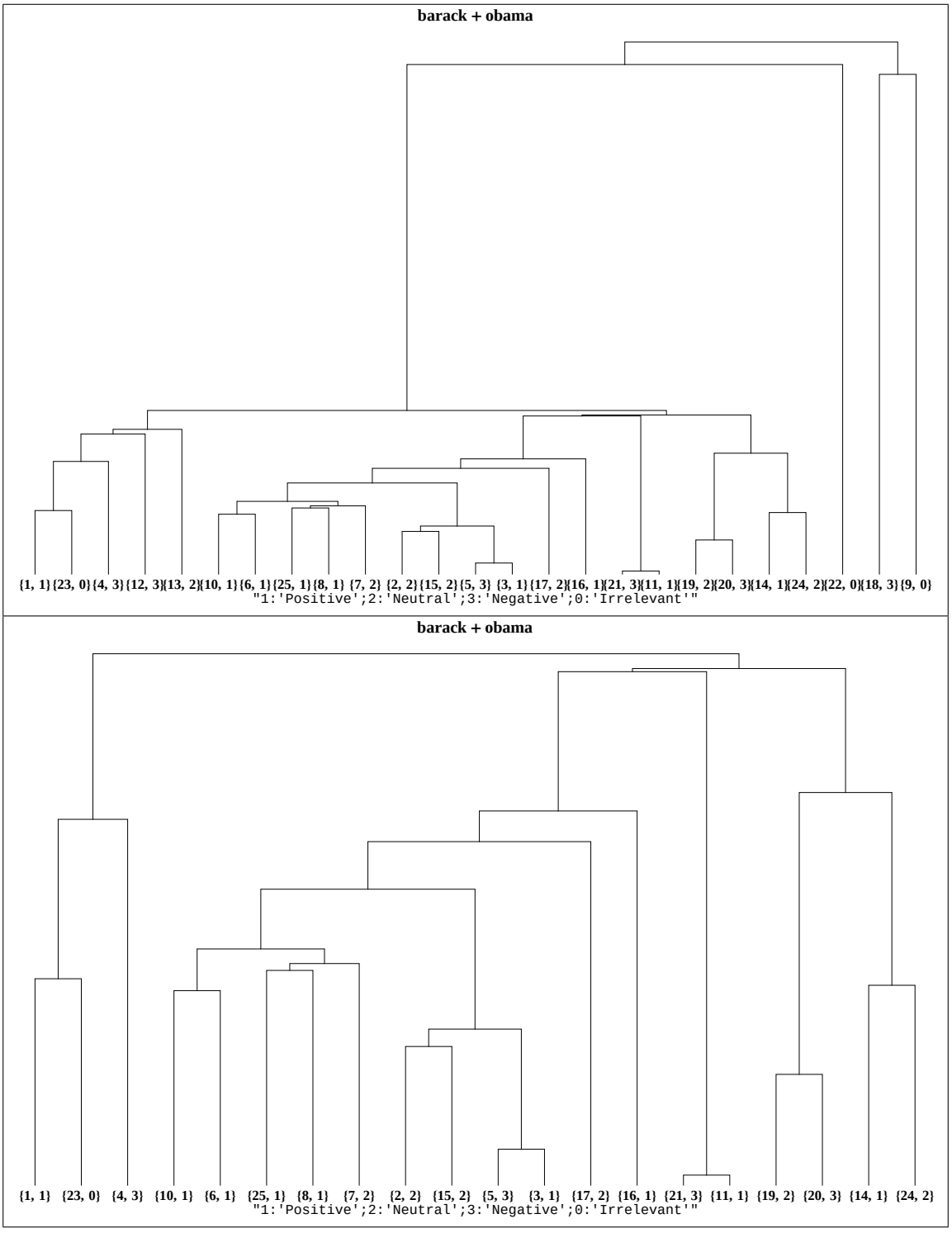
Recalculate clustering

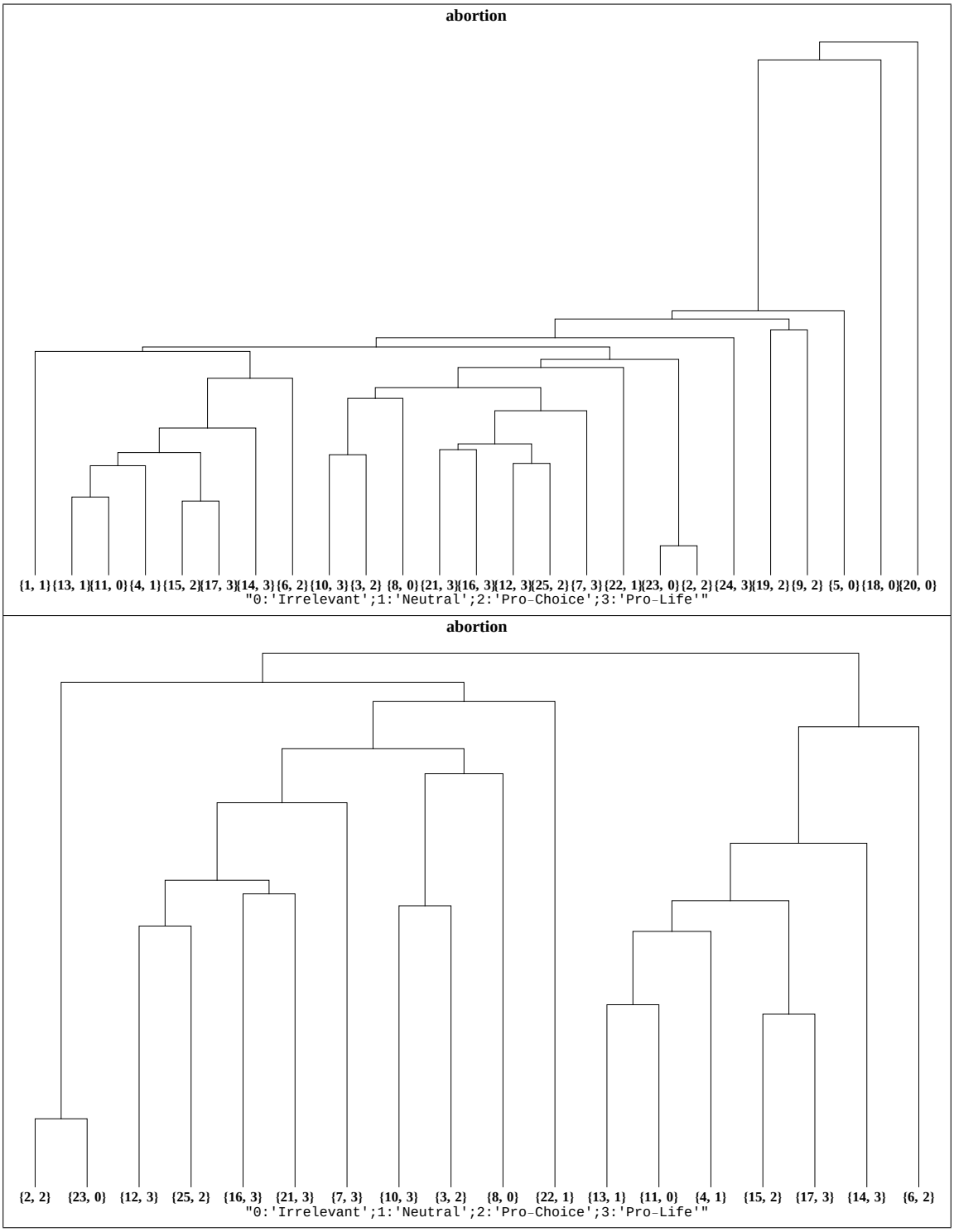
The reason I selected the α-mean technique is so that we can reject the outliers before calculating the cut-off threshold. We are rejecting the α smallest values for calculating the mean, so that these small values do not make the cut-off artificially small, but we are not removing merges with these small values, because small values are preferable, and represent good merges. We only reject the articles leading to large merging thresholds during the pruning process.

A possible alternative that was rejected was using a cut-off related to the harmonic mean of the merging thresholds. A harmonic mean would give lower weights to larger values, and higher weights to small values, leading to a cut-off that would prevent large merging thresholds, but, as mentioned above, this value would be greatly affected by small merging thresholds. This was why I chose a more neutral approach like α-mean.

An α value of 0.2 was used for the following results.

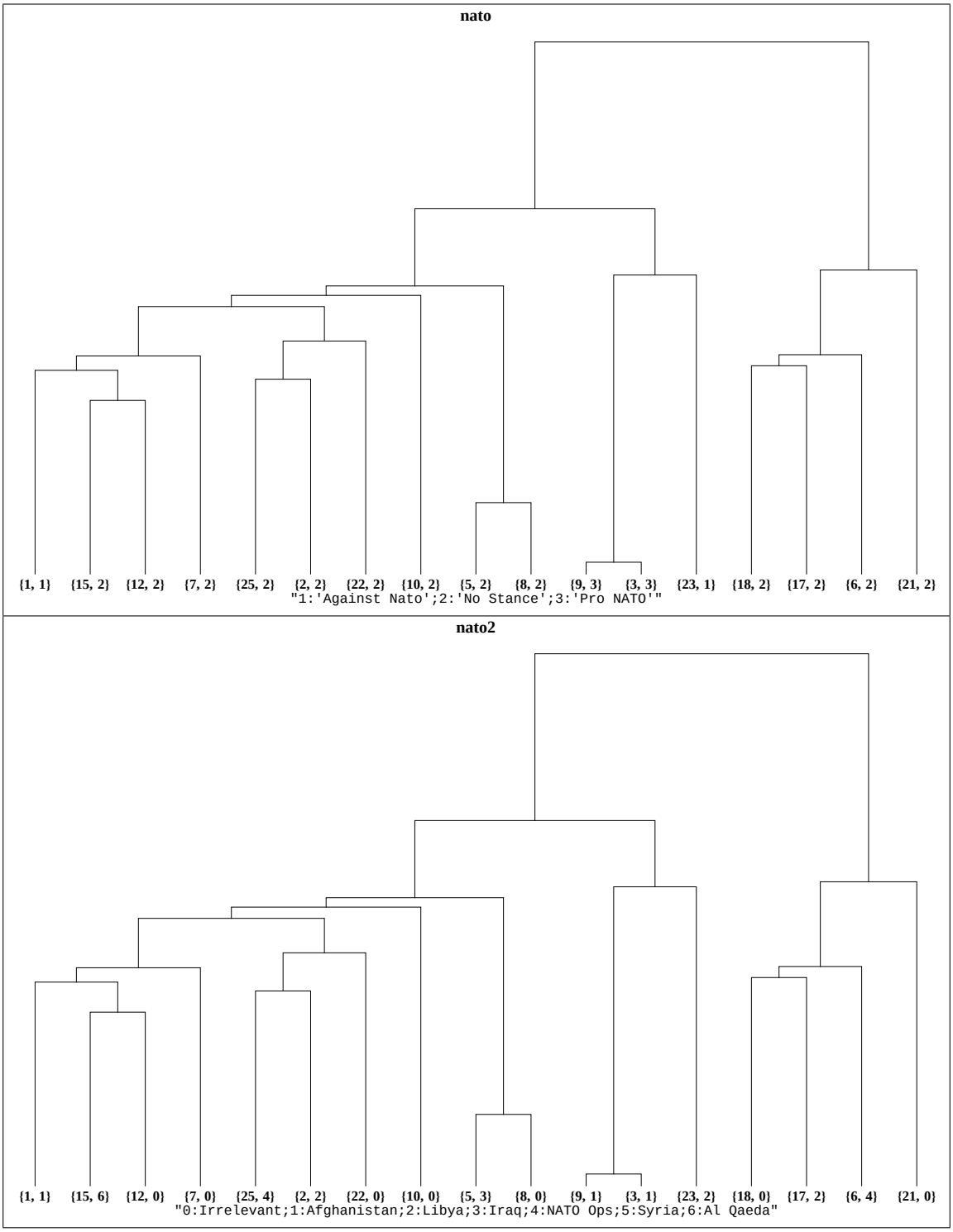
7 Results after Cleaning





We notice that many of the outliers are classified as *irrelevant* and are removed using this approach, leading to a cleaner clustering.

7.1 Inter-annotator agreement



We notice that since our ground-truth clusters are dependant on the type of clusters created by the annotators, we get different possible types of clusters. Our technique performs better on the ideological ground-truth clusters, but poorly on the content-based ground-truth clusters.

8 Conclusion

In conclusion, we can clearly see that the feature vectors, simple as they are, perform quite well when detecting clustering. The outliers detected and removed by the outlier removal algorithm also performs well, although improving the feature vector will enhance this accuracy as well.

We can see that ideological biases are well-detected using these features, but non-ideological/content biases are poorly clustered. This is to be expected, since our features deal with identifying author bias. Neutrality detection is poor, since the clustering algorithm often merges neutral/unbiased articles either with the *pro-topic* articles, or the *anti-topic* articles, for an ideological clustering.

9 Future Work

The first step that needs to be taken is to find a splitting point in the cleaned hierarchical clusters. We need to be able to use this splitting point to actually split the articles into multiple clusters, each representing a different content-bias. We also need to improve the features, or add new features, so as to make the feature-vectors more representative of the articles.

An approach that could solve the poor detection of content biases could be a form of *pyramidal clustering* where we perform a clustering based on the above feature-vector, and after getting different clusters, we sub-cluster then further using a different content-specific feature vector.

10 Acknowledgements

First, and foremost, I would like to thank Dr. Arnab Bhattacharya for his invaluable guidance in all aspects of work in this project. Without his inputs, this project would not be as developed as it is. I would also like to thank the Yahoo! team for their help and access to the Yahoo! Voices articles. I would like to end by thanking Govind Gopakumar, Kushal Yarlagadda, and Anurag Sahay for their help in creating the ground-truth clustering for the articles.

Please note that all the remaining results, for the rest of the search terms, may be found at *http://www.cse.iitk.ac.in/users/chandras/cs697*

References

[1] **Yahoo!** Voices, *http://voices.yahoo.com*

[2] Yahoo! Query Language, *https://developer.yahoo.com/yql/*

[3] Alexander Rohde, **Semantic Reasoning**, *http://semanticreasoning.org/*

[4] Wikipedia: Manual of Style, Words to Watch, *http://en.wikipedia.org/wiki/Wikipedia:Words_to_watch*

[5] Wikipedia: Neutral Point of View, *http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view*

[6] Herzig, Livnat, Alex Nunes, and Batia Snir, **An annotation scheme for automated bias detection in Wikipedia.**, *Proceedings of the 5th Linguistic Annotation Workshop*. Association for Computational Linguistics, 2011.

[7] AlchemyAPI, *http://www.alchemyapi.com*