**UNSUPERVISED STANCE DETECTION**

Submitted in partial fulfillment of the requirements for the award of degree

**Of**

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE**

**AND ENGINEERING**

**Submitted by**

**M Prathyusha (N171211),**

**MD Iqbal (N170973),**

**Sk Fakrunnisa (N170543)**

Under the esteemed guidance

**Of**

**Mrs. Bhavani (Assistant Professor)**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES** **NUZVID, ELURU(DIST-521202)**

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES NUZVID, ELURU (521202),**

**ANDHRA PRADESH**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DECLARATION CERTIFICATE**

This is to certify that the project entitled, “UNSUPERVISED STANCE DETECTION” is submitted by M Prathyusha (N171211), MD Iqbal (N170973), Sk Fakrunnisa (N170543) to the department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, Nuzvid for the submission of major project report in IV year B.Tech in Computer Science and Engineering is a bonafide work carried out under supervision and guidance during the academic year 2022.

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**   **- - - - - - - - - - - - - - - - - - - - - - - - - -**

|  |
| --- |
| **Mr. S. Chiranjeevi** **Mrs.S.Bhavani** |
| Assistant Professor Assistant Professor |
| **Head of the Department**  **Project Supervisor** |
| Dept. of CSE Dept. of CSE |
| RGUKT, Nuzvid RGUKT, Nuzvid |
|  |

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES NUZVID, ELURU (521202),**

**ANDHRA PRADESH**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DISSERTATION APPROVAL CERTIFICATE**

This is to certify that the project entitled, “UNSUPERVISED STANCE DETECTION” is submitted by M Prathyusha (N171211) ,MD Iqbal (N170973), Sk Fakrunnisa (N170543) to the department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, Nuzvid for the award of Bachelor of Technology in Computer Science and Engineering, has been accepted by the external examiners and that these students have successfully defended the project viva in the Viva-Voce examination held today.

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**   **- - - - - - - - - - - - - - - - - - - - - - - - -**

|  |
| --- |
| **Mr. S. Chiranjeevi** **Mrs.S.Bhavani** |
| Assistant Professor Assistant Professor |
| **Head of the Department**  **Project Supervisor** |
| Dept. of CSE Dept. of CSE |
| RGUKT, Nuzvid RGUKT, Nuzvid |
|  |

|  |
| --- |
|  |

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES NUZVID, ELURU (521202),**

**ANDHRA PRADESH**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ACKNOWLEDGMENT**

We would like to acknowledge and express our gratitude to the following people for their magnificent support in sharing their wisdom and knowledge to our successful completion of the final year project.

We would like to express our profound gratitude and deep regards to our guide **Mrs.S.Bhavani, Assistant professor**, Dept. Of Computer science and Engineering for her exemplary guidance, monitoring and constant encouragement throughout the B.Tech course. We shall always cherish the time spent with her during the course of this work due to the valuable knowledge gained in the field of reliability engineering.

We express gratitude to **Mr.S.Chiranjeevi, Head of the Department**(CSE) and other faculty members for being a source of inspiration and constant encouragement which helped us in completing the project successfully.

We are thankful to the director, **Prof. G.V.R SRINIVASARAO** for giving this opportunity to do this project.

Our sincere thanks to all the batch mates of 2017 CSE, who have made our stay at RGUKT-NUZVID, a memorable one.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Table Of Contents**

**Chapter-1**

1. Abstract………………………………………………………..

**Chapter-2**

1. Introduction……………………………………………………......

**Chapter-3**

1. Related work…………………………………………………...

**Chapter -4**

1. Proposed method……………………………………………….

**Chapter -5**

1. Implementation ………………………………………………..

**Chapter - 6**

1. Results…………………………………………………………

**Chapter -7**

1. Discussions…………………………………………………….

**Chapter - 8**

1. Conclusion …………………………………………………….

**Chapter - 9**

1. Future Enhancement……………………………………………

**Chapter - 10**

1. Bibliography………………………………………………………

**List of the figures**

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Figure Name** | **Page Number** |
| Fig1 | Proposed methodology flowchart | 13 |
| Fig2 | Implementation flowchart | 17 |
| Fig3 | PCA plot of create debate data | 18 |
| Fig4.1.1 | PCA plot for convinceme data with k-means for texts | 20 |
| Fig4.1.2 | PCA plot for convinceme data with agglomerative for texts | 20 |
| Fig4.2.1 | PCA plot for convinceme data with k-means for discussions | 21 |
| Fig4.2.2 | PCA plot for convinceme data with agglomerative for discussions | 21 |
| Fig4.3.1 | PCA plot for createdebate data with k-means for texts | 22 |
| Fig4.3.2 | PCA plot for createdebate data with agglomerative for texts | 22 |
| Fig4.4 | PCA plot for createdebate data with agglomerative for discussions | 23 |
| Fig4.5.1 | PCA plot for Argumentative data with k-means classifier | 23 |
| Fig4.5.2 | PCA plot for Argumentative data with Agglomerative classifier | 24 |

**List of tables**

|  |  |  |
| --- | --- | --- |
| **Table Number** | **Table Name** | **Page no.** |
| Table 1 | Silhoutte scores of argumentative dataset  Using different classifiers | 19 |
| Table 2 | Silhoutte scores of convinceme dataset  Using different classifiers | 19 |
| Table 3 | Silhoutte scores of createdebate dataset  Using different classifiers | 20 |

|  |
| --- |
| **Chapter-1** |
| **ABSTRACT** |

Social media platforms have become an integral place for online consultation for users to discuss, debate and form opinions. In this project, we propose an unsupervised method for detecting argumentative stances related to a topic. To address this limitation, we propose a topic-agnostic approach that focuses on commonly encountered classes of arguments, particularly arguments from results. This is done by extracting the effects to which the claims relate and proposing means of inferring whether the effects are favor or against results. The focus of our work is the taxonomy of attitudes trained for discussion of topics.

|  |
| --- |
| **Chapter-2** |
| **INTRODUCTION** |

In the context of decision making, it is important to compare the positive and negative effects resulting from possible decisions. In fact, a common form of reasoning is to agree or disagree with something because of its possible consequences. In this project, we tackle the classic stance detection problem and pay special attention to such arguments. Stance recognition (also called stance classification) is the task of determining whether a text agrees, disagrees, or is irrelevant to a particular topic. This problem is related to opinion mining, but whereas opinion mining focuses on the polarity of emotions explicitly expressed by the text, stance detection focuses on the position the text takes with respect to general topics. intended to make a decision.

Online discussions are usually laid out in a tree structure. Assuming claim E is placed at the root of the tree, each subsequent node is a direct response to the previous node. This tree structure can be transformed into an interaction network G. where the nodes of G are speakers and the edges correspond to interactions. Edges can be weighted to reflect the strength of interaction between a particular pair of speakers.

In this project, we propose a new approach to stance detection. Our method is unsupervised, domain-independent, and computationally efficient. The premise of our approach is that naturally occurring conversational structures in many online discussion forums and social platforms can be used for stance detection

|  |
| --- |
| **Chapter-3** |
| **RELATED WORK** |

**STEM - UNSUPERVISED STRUCTURAL EMBEDDING FOR STANCE DETECTION**

As per Ref[8], online discussions tend to unfold in a tree structure. This tree structure can be converted into an interaction network G. The interaction network G was computed into the 2-core of the graph (By taking strongly connected vertices). Next, computing the embedding of the 2-core graph using SDP of max cut. Now, rounding the vectors into Random hyperplane or Clustering. Then labels propagate greedily in graph 2-core graph.

Network Formation:

In an interaction network with respect to Ref[8], nodes correspond to speakers, and an edge E(u,v) between two nodes (speakers) u and v indicates a direct interaction between the two. Edges can be weighted to indicate the strength of the interaction (that is, how many times user u replied and how many times user v was quoted).

1. core Reduction:

To compute the 2-core graph, as per Ref[8], iteratively remove the nodes with degree less than two in the remaining graph until there are no such nodes.

Semi Definite Program (SDP):

An automated approach to predicting software errors using machine learning techniques. Because of a number of factors, interest in semidefinite programming, a relatively new area of optimization, is expanding. Semidefinite programming problems can be used to represent or approximate a variety of real-world operations research and combinatorial optimization issues. SDPs are employed in the setting of linear matrix inequalities in automatic control theory. SDPs are actually a specific example of cone programming, and interior point methods are effective for solving them. All linear program and (convex) quadratic program can be described as SDPs, and the solutions to polynomial optimization issues can be roughly determined using hierarchies of SDPs. Complex system optimization has been done using semidefinite programming. Some quantum query complexity issues have recently been expressed in terms of semidefinite program.

Max Cut:

For graphs, the largest cut is the cut whose size is at least as large as the other cuts. That is, in Ref[8] it divides the vertices of the graph into two complementary sets S and T such that the number of edges between S and T is as large as possible. Finding such a cut is known as the maximum cut problem.

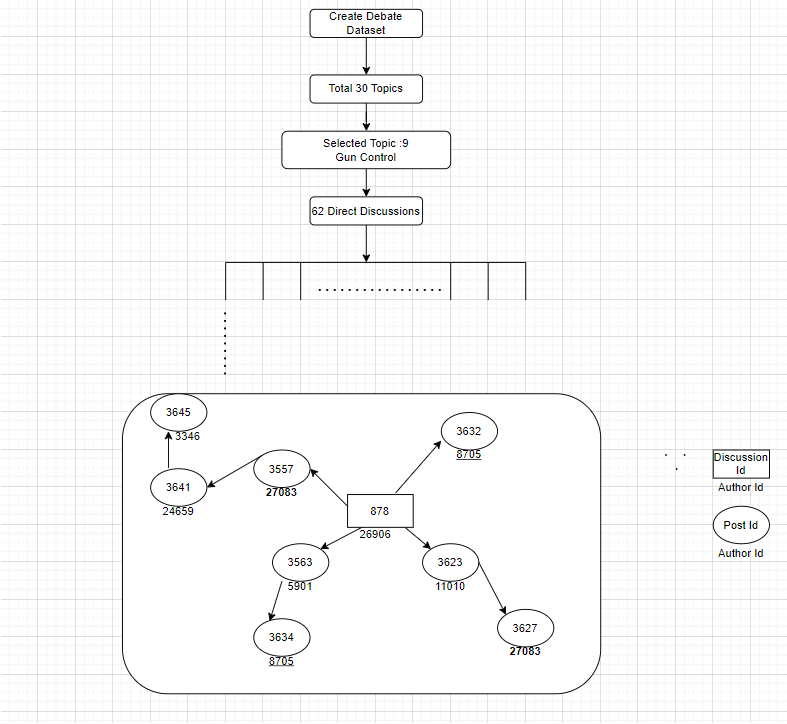
|  |
| --- |
| **Chapter-4** |
| **PROPOSED METHODOLODY** |

The main aim of our project is to detect stance of argumentative data set. Our task is to detect the stance that the claim has with respect to the topic. Statements such as the claim or topic usually express a positive (favorable) or negative (unfavorable) position to a concept that we call the target. As such, the target is a phrase that belongs to the statement. Our solution starts by first determining the stance of the claim and of the topic towards their respective targets.

Here we give the argumentative data to the unsupervised classifiers such as k-means clustering algorithm, DBSCAN Clustering algorithm, Agglomerative Hierarchical Clustering. It trains the data and cluster into positive and negative stances.

Here, we considered the datasets of Create-debate and Convinceme. These datasets are converted from sql format to excel format using python code. The data which was extracted in the form of excel consists of many sheets such as author, dataset metadata, discussion, discussion stance, discussion topic, markup, post, quote, text, topic and topic stance. Create-debate dataset consist of 3,929 text fields and 3,050 posts which are discussions made by the authors or people.

**Architecture of Proposed Methodology**

****

**Fig1:** Architecture of the proposed methodology considering create-debate dataset.

Create-debate dataset consists of total 30 topics like abortion, gay marriage, school uniforms, existence of God, etc. Here this dataset consists all the discussions about only gun control. On the topic gun control there are 62 direct discussion sets. This discussion set consists of a statement with the related topic and a network of posts which are directly or indirectly replied to the main discussion statement.

Here we observe same author can participate in n number of discussions and can reply multiple times. In the above flowchart the authors with author ID 8705 and 27083 had more than one post with different post ID’s which are replies to different authors.

Each topic has individual sets of stances. For the topic gun control the stances are classified into five types. They are unknown, prefers strict gun control, opposes strict gun control, undecided, other. Remaining topics had stances around 3 to 6. In the topic gun control we also observe discussion stance for the 62 discussion topics which range from 2 to 10.

|  |
| --- |
| **Chapter-5** |
| **IMPLEMENTATION** |

In our approach we taken the data of an argumentative statements which are train.tsv and test.tsv, create-debate and convinceme . The train.tsv and test.tsv data set consists of the columns with Argument-Id, Conclusion, Stance and Premise. As we are following unsupervised machine learning concept, we drop the stance column in both train and test data set. The scaled train data is given for training with k-means clustering algorithm, DBSCAN Clustering algorithm and Agglomerative Clustering Algorithm to predict the stance of scaled test data.

In the convinceme data set we extracted the excel data from the sql data. From this excel format of data we took the sheets named text and discussion stance. Considering text id and text from the text sheet we trained the model with unsupervised classifiers, for clustering the data we considered into two or three clusters based on classifiers. The classifiers used are K-means, DBSCAN, Agglomerative clustering algorithms.

Also, from the convinceme dataset we taken the discussion stance sheet which consists of the discussion stances for the each individual discussion topic set around two. This data is also given for training with multiple classifiers for clustering the data into two or three clusters based on the classifier.

Here we observe k-means and agglomerative data is classified into two clusters and DBSCAN algorithm classifies into 3 or more clusters.

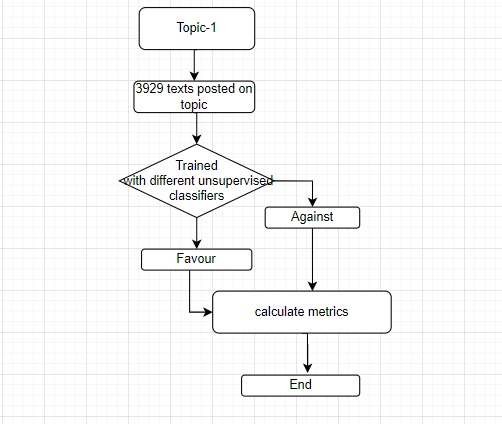
In the create-debate data set also we extracted the excel data from the sql data. From this excel format of data we took the sheets named text and discussion stance. Considering text id and text from the text sheet we trained the model with unsupervised classifiers, for clustering the data we considered into two or three clusters based on classifiers. The classifiers used are K-means, DBSCAN, Agglomerative clustering algorithms.

Also, from the create-debate dataset we taken the discussions stance sheet which consists of the discussion stances for each individual discussion topic set which range from two to ten. This data is also given for training with multiple classifiers for clustering the data into two or three clusters based on the classifier.

Here we observe k-means and agglomerative data is classified into two clusters and DBSCAN algorithm classifies into 3 or more clusters.

The classifiers output the data into two different stances which are positive(favor) and negative(against) stances. Here, the results are displayed in the form of a scatter plot where each point in the plot describes the stance of the given statements in the data set. Assumed as the blue colored plots refer to the positive stance(favor) and the orange color plots refer to the negative stance(against).

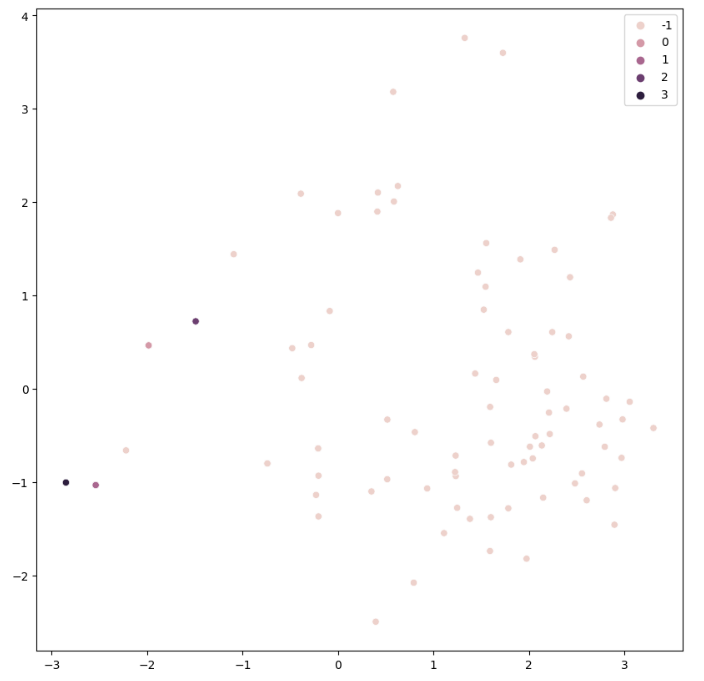
**Flowchart of the implementation process**



**Fig2:** Flowchart of the implementation approach.

In the above figure, topic refers to different topic with respect to the dataset we considered. For create-debate dataset the topic is gun control and all the discussions and texts are based on gun control. For convinceme dataset the topic is abortion, and all the discussions and texts are based on abortion. Unsupervised classifiers used in the implementation are k-means, DBSCAN, and agglomerative clustering algorithms.

These classifiers divide the data into two clusters positive and negative irrespective of the stances of the dataset. The metrics used to calculate the model fitness of the classifier are silhouette score which is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation) and PCA plots which is a plot that shows clusters of samples based on their similarity.



**Fig3:** PCA plot of the create-debate discussion data clustered using DBSCAN classifier.

**Chapter-6**

**RESULTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Classifier** | **Silhouette Score** |
| 1. | K-means | 0.79 |
| 2. | DBSCAN | 0.89 |
| 3. | Agglomerative | 0.81 |

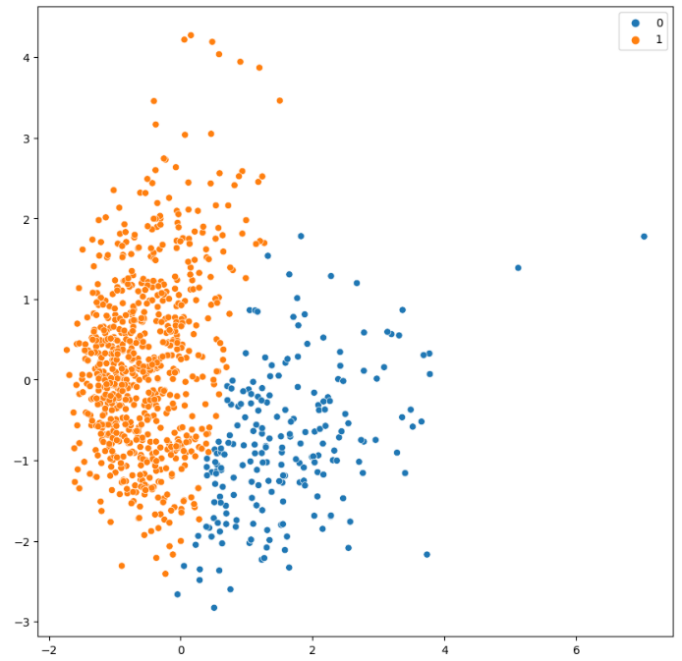
**Table1:** Accuracy for the dataset train.tsv and test.tsv on social topics is trained with above mentioned classifiers and the result is calculated through silhoutte score.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Data** | **Quantity** | **K-means** | **DBSCAN** | **Agglomerative** |
| 1 | #texts | 68197 | 0.17 | 0.46 | 0.13 |
| 2 | #discussions | 5412 | 0.13 | 0.06 | 0.12 |

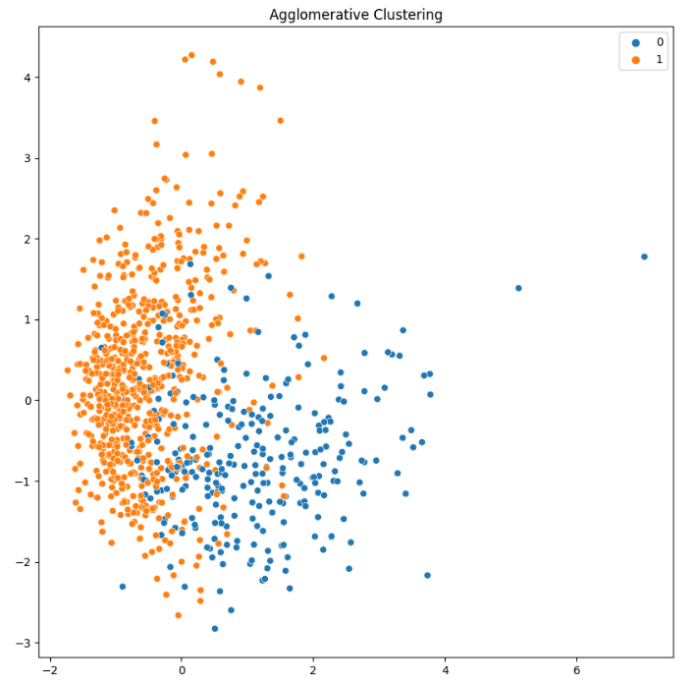
**Table2:** Silhoutte score is calculated for the convinceme dataset considering texts and discussions from the convinceme dataset which is trained with above mentioned classifiers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Data** | **Quantity** | **K-means** | **DBSCAN** | **Agglomerative** |
| 1 | #texts | 3929 | 0.11 | -0.06 | 0.07 |
| 2 | #discussions | 137 | 0.27 | 0.35 | 0.25 |

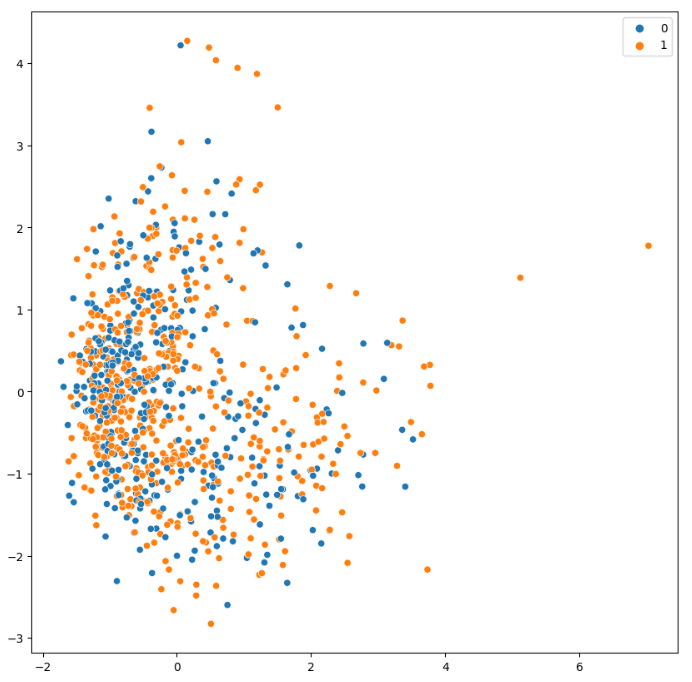
**Table3:** Silhoutte score is calculated for the create-debate dataset considering texts and discussions from the create-debate dataset which is trained with above mentioned classifiers.

****

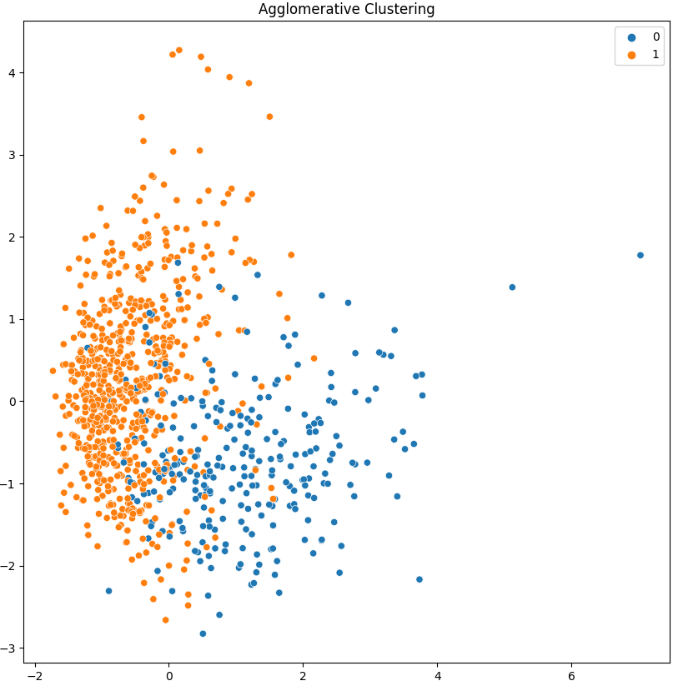
**Fig4.1.1:** PCA plot for Convinceme dataset with K-means classifier for texts.

****

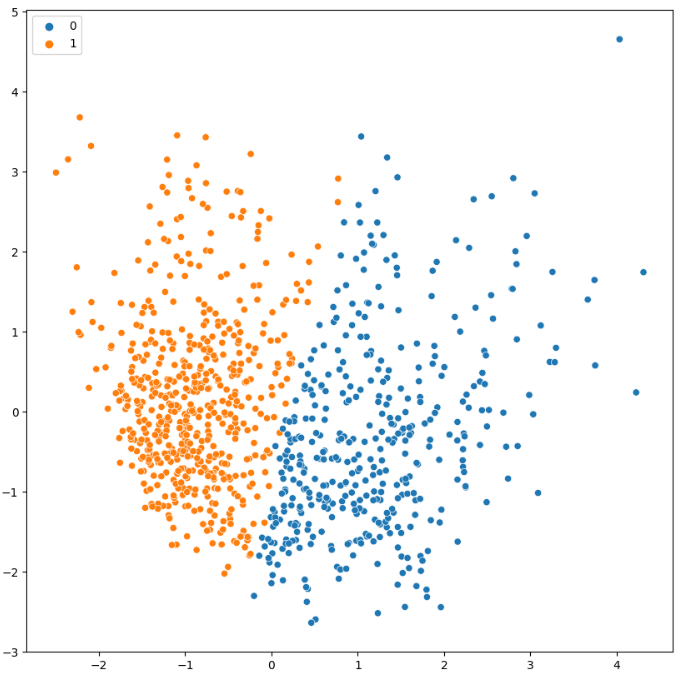
**Fig4.1.2:** PCA plot for Convinceme dataset with Agglomerative classifier for texts.

****

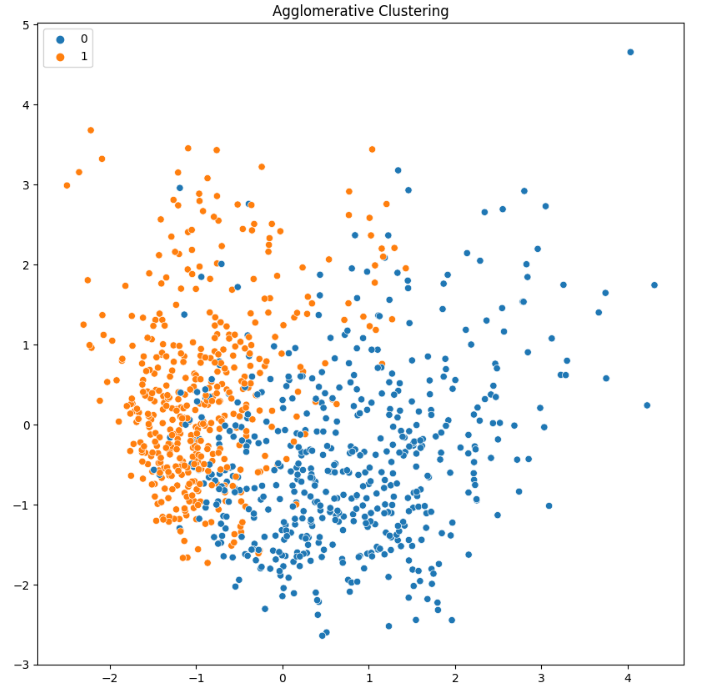
**Fig4.2.1:** PCA plot for Convinceme dataset with K-means classifier for discussions.

****

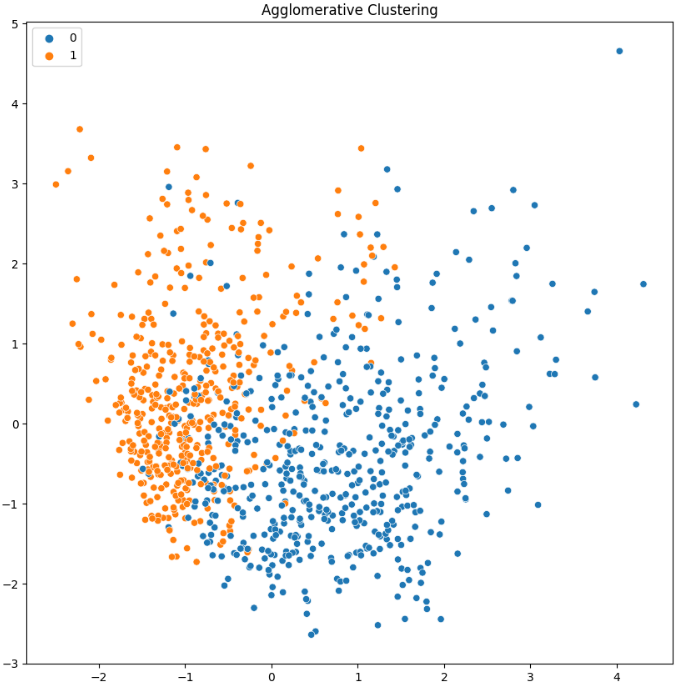
**Fig4.2.2:** PCA plot for Convinceme dataset with Agglomerative classifier for discussions.



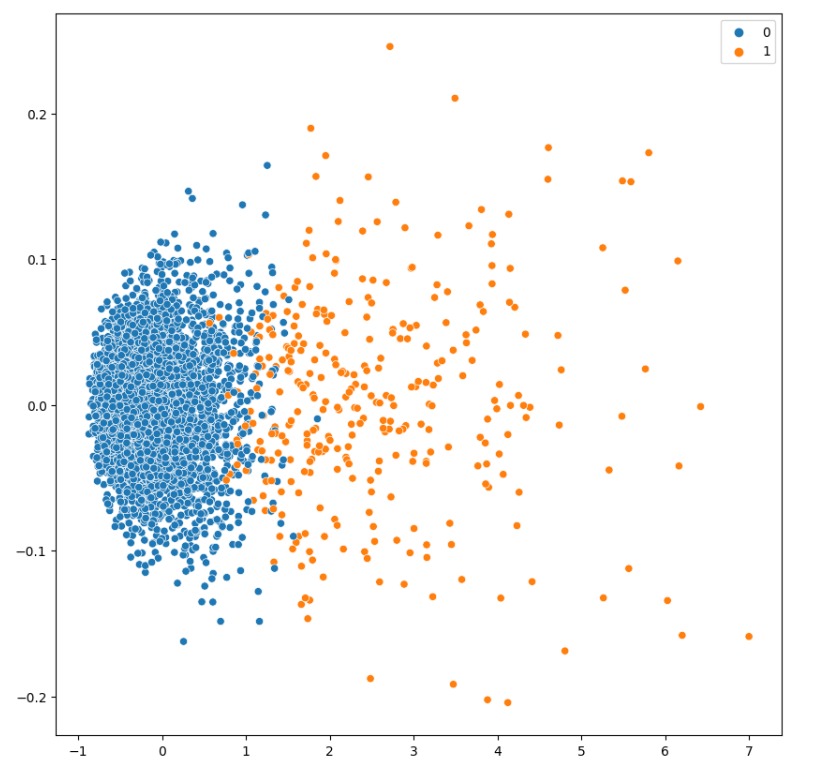
**Fig4.3.1:** PCA plot for Create-debate dataset with k-means classifier for texts.



**Fig4.3.2:** PCA plot for Create-debate dataset with Agglomerative classifier for texts



**Fig4.4:** PCA plot for Create-debate dataset with Agglomerative classifier for discussions.



**Fig4.5.1:** PCA plot for Argumentative dataset with k-means classifier.



**Fig4.5.2:** PCA plot for Argumentative dataset with Agglomerative classifier.

**Chapter-7**

**DISCUSSIONS**

**K-Means Clustering:**

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled data set into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled data set on its own without the need for any training.

To calculate the similarity, we can use the Euclidean distance, Manhattan distance or Hamming Distance. In this algorithm we choose k number of random data points as initial centroids. Until the cluster center stabilizes allocate each point in D to the nearest of Kth centroids and compute centroid for the cluster using all points in the cluster.

**DBSCAN Clustering Algorithm:**

Clusters are dense regions in the data space, separated by regions of the lower density of points. The **DBSCAN algorithm** is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

Key parameters:

**eps:** The distance that specifies the neighborhoods. Two points are considered to be neighbors if the distance between them are less than or equal to eps.

**MinPts:** Minimum number of data points to define a cluster.

**Core point:** A point is a core point if there are at least minPts number of points (including the point itself) in its surrounding area with radius eps.

**Border point:** A point is a border point if it is reachable from a core point and there are less than minPts number of points within its surrounding area.

**Outlier:** A point is an outlier if it is not a core point and not reachable from any core points.

**DBSCAN Algorithm working :**

1. Start.

2. Set D = {end point of each segment}

3. Detect all the core points in D which have: Eps>MinPts.

4. Join core points in the cluster.

5. Detect all the borders which have Eps<MinPts and are neighbors of core points.

6. Attach border points to core points.

7. Mark other points in D as noise/outliers.

8. End.

**Agglomerative Clustering Algorithm:**

The **agglomerative clustering** is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It’s also known as AGNES*(*Agglomerative Nesting*).* The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named dendrogram.

**Agglomerative Clustering Algorithm working:**

1. Compute the proximity matrix using a particular distance metric.

2. Each data point is assigned to a cluster.

3.  Merge the clusters based on a metric for the similarity between clusters.

4. Update the distance matrix.

5. Determining where to cut the hierarchical tree into clusters. This creates a partition of the data.

**Limitations:**

Limitations we faced in this approach are analyzing the huge data extracted from the online discussions which consists of network information in individual sheets. In the dataset some statements will be difficult to categorize whether they are supporting or opposing the post given by other authors. For example, if “Gun control should be banned” is a statement and if the reply post is like “Yes” or “No” system cannot categorize whether the author is supporting or opposing the statement.

|  |
| --- |
| **Chapter-8** |
| **CONCLUSION** |

We proposed a fully unsupervised method to detect the stance of arguments from consequences in online debates. The method exploits grammatical dependencies and lexicons to identify effect words and their impact. We observed the partition of positive (favor) and negative (against) stance of the argumentative data set in two separate clusters by using the unsupervised classification clustering algorithms like K-Means clustering algorithm, DBSCAN clustering and Agglomerative Clustering. The results are calculated by Silhouette score which is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation) and also given PCA plots for each classifier with respect to the datasets.

For the datasets which are used in our project, the clusters formed using k-means clustering algorithm are clustered well as two unique clusters separately. In agglomerative clustering we observed the clustering as scattered. In DBSCAN clustering it categorized into three and more clusters which is difficult to identify the taxonomy. By calculating another metric silhouette score we observed that k-means is behaving well for the datasets which are used in our approach.

|  |
| --- |
| **Chapter-9** |
| **FUTURE ENHANCEMENT** |

We have presented an effective unsupervised method for

identifying clusters which have similar stances with respect to controversial topics. After selecting the best clustering algorithm, further we can convert the data into graphical networks. We can also take the data from the social media platforms (e.g: twitter) through API’s and form the network for stance detection which is an automated way for knowing the perspective of the user.

**Chapter-10**

**BIBLIOGRAPHY**

1. Abbott, Rob, Brian Ecker, Pranav Anand, and Marilyn Walker. "Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it." In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 4445-4452. 2016.

2. Allaway, Emily, and Kathleen McKeown. "Zero-shot stance detection: a dataset and model using generalized topic representations." arXiv preprint arXiv:2010.03640 (2020).

3. Darwish, Kareem, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. "Unsupervised user stance detection on twitter." In Proceedings of the International AAAI Conference on Web and Social Media, vol. 14, pp. 141-152. 2020.

4. Hardalov, Momchil, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. "Few-shot cross-lingual stance detection with sentiment-based pre-training." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 10, pp. 10729-10737. 2022.

5. Joseph, Kenneth, Lisa Friedland, William Hobbs, Oren Tsur, and David Lazer. "Constance: Modeling annotation contexts to improve stance classification." arXiv preprint arXiv:1708.06309 (2017).

6. Kobbe, Jonathan, Ioana Hulpuș, and Heiner Stuckenschmidt. "Unsupervised stance detection for arguments from consequences." In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp. 50-60. 2020.

7. Küçük, D. and Can, F., 2020. Stance detection: A survey. ACM Computing Surveys (CSUR), 53(1), pp.1-37..

8. Pick, Ron Korenblum, Vladyslav Kozhukhov, Dan Vilenchik, and Oren Tsur. "STEM: Unsupervised STructural EMbedding for Stance Detection." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 10, pp. 11174-11182. 2022.

9. Seese, D. 1990. Groetschel, M., L. Lovasz, A. Schrijver: Geometric Algorithms and Combinatorial Optimization. (Algorithms and Combinatorics. Eds.: R. L. Graham, B. Korte, L. Lovasz. Vol. 2), Springer-Verlag 1988, XII, 362 pp., 23 Figs., DM 148,-. ISBN 3–540–13624-X. Biometrical Journal, 32(8): 930–930.

10. Sobhani, P.; Inkpen, D.; and Zhu, X. 2017. A dataset for multi-target stance detection. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 551– 557.

11. Somasundaran, S.; and Wiebe, J. 2010. Recognizing stances in ideological on-line debates. In Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text, 116–124.

12. Sridhar, D.; Foulds, J.; Huang, B.; Getoor, L.; and Walker, M. 2015. Joint Models of Disagreement and Stance in Online Debate. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 116–125. Beijing, China: Association for Computational Linguistics.

13. Sun, Q.; Wang, Z.; Zhu, Q.; and Zhou, G. 2018. Stance detection with hierarchical attention network. In Proceedings of the 27th International Conference on Computational Linguistics, 2399–2409.

14. Tsang, S. J. 2020. Issue stance and perceived journalistic motives explain divergent audience perceptions of fake news. Journalism, 1464884920926002.

15. Tyagi, A.; and Carley, K. M. 2020. Divide in Vaccine Belief in COVID-19 Conversations: Implications for Immunization Plans. medRxiv.

16. Walker, M.; Anand, P.; Abbott, R.; and Grant, R. 2012a. Stance Classification using Dialogic Properties of Persuasion. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 592–596. Montreal, Canada: ´ Association for Computational Linguistics.

17. Walker, M. A.; Tree, J. E. F.; Anand, P.; Abbott, R.; and King, J. 2012b. A Corpus for Research on Deliberation and Debate. In LREC, volume 12, 812–817. Istanbul, Turkey.

18. Yin, J.; Narang, N.; Thomas, P.; and Paris, C. 2012. Unifying Local and Global Agreement and Disagreement Classification in Online Debates. In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, 61–69. Jeju, Korea: Association for Computational Linguistics.

19. Zakharov, S.; Hadar, O.; Hakak, T.; Grossman, D.; Kolikant, Y. B.-D.; and Tsur, O. 2021. Discourse Parsing for Contentious, Non-Convergent Online Discussions. In Proceedings of the International AAAI Conference on Web and Social Media, volume 15, 853–864.

20. Zubiaga, A.; Kochkina, E.; Liakata, M.; Procter, R.; and Lukasik, M. 2016. Stance Classification in Rumours as a Sequential Task Exploiting the Tree Structure of Social Media Conversations. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2438–2448