Assessing Impact of Government Policies on Vulnerable Populations

Arpan Kumar
arpan21020@iiitd.ac.in
2021020

Tony Thomas
tony21360@iiitd.ac.in
2021360

Daksh Pandey
daksh21036@iiitd.ac.in
2021036

Kajol Dwivedi
kajol23041@iiitd.ac.in
MT23041

Riya Dhama
riya23077@iiitd.ac.in
MT23077

Atharv Srivastav
atharv21240@iiitd.ac.in
2021240

ACM Reference Format:
Arpan Kumar, Tony Thomas, Daksh Pandey, Kajol Dwivedi, Riya Dhama, and Atharv Srivastav. 2024. Assessing Impact of Government Policies on Vulnerable Populations. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM STATEMENT

"**Policymakers** need to **assess the impact of their policy on vulnerable populations** because understanding the impact of policies is crucial for determining whether they effectively support vulnerable groups or exacerbate their challenges."

For this, we are developing a chatbot to assess the impact of government policies on vulnerable populations, as it presents a critical need to address disparities and ensure equitable policy outcomes. The project aims to create a conversational system capable of collecting data on government initiatives, understanding their implementation, and assessing effects on vulnerable populations, supporting policymakers and marginalized communities.

2 MOTIVATION

The motivation for developing a chatbot to assess the impact of government policies on vulnerable populations stems from a critical observation: many policies are implemented without adequate evaluation of their effects, particularly on marginalized and vulnerable communities. This oversight can lead to unintended consequences, exacerbating existing disparities and challenges faced by vulnerable populations.

Policymakers often face significant challenges in comprehensively assessing the impact of their policies on marginalized groups. Without a robust understanding of how policies affect vulnerable populations, there is a risk of perpetuating inequities or overlooking critical areas where support is needed the most.

By developing a conversational system equipped to collect data on government initiatives, understand their implementation, and evaluate their effects on vulnerable communities, we aim to bridge this crucial gap in policy evaluation. This chatbot will serve as a valuable tool for policymakers, providing them with actionable insights into the real-world impact of their decisions.

Ultimately, the goal is to empower policymakers to make informed decisions that prioritize equitable outcomes and support marginalized communities. Through innovative technology and data-driven analysis, our project seeks to contribute to more inclusive and effective policymaking, ensuring that policies are not just implemented but positively impact those who need support the most.

3 IMPORTANCE OF THE PROBLEM

In many societies, low-income individuals, ethnic minorities, and other vulnerable groups continue to experience disproportionately negative outcomes as a result of government policies that inadequately consider their unique needs and circumstances. Without effective mechanisms to evaluate policy impacts through an equity lens, these disparities are likely to persist, exacerbating social and economic inequality. Moreover, the COVID-19 pandemic has further highlighted the urgency of addressing systemic inequities, as vulnerable populations have been disproportionately affected by the health, economic, and social impacts of the crisis. The dire need to develop a chatbot to assess the impact of government policies on vulnerable populations is underscored by the persistent and often widening disparities marginalized communities face. Therefore, there is an urgent need to develop innovative tools like chatbots that can provide policymakers with actionable insights to identify and address disparities, ultimately working towards more equitable and inclusive policy outcomes that uplift all members of society.

4 LITERATURE REVIEW

1. Context-based News Articles Retrieval using CLSM

Paper: Link

Information:

According to (Anamalamudi and Reddy 2021), the context-based analysis method implemented can help us analyze the tweets/news and understand the context of a tweet as to why a policy went so bad or so good. The CLSM model extracts contextual features from both queries and documents. It represents queries and documents as sequences of words and utilizes convolutional and max-pooling layers to capture local and global contextual features at the word and sentence levels, respectively.

2. A Government Decision Analytics Framework Based on Citizen Opinion

Paper: Link

Information:

We can use the Bayesian Predictive Process model to predict citizens' opinion for a new proposal. Analyzing sentiments from existing data will make policy makers draft a better policy.

3. Policy Impact and Evaluation

Paper: Link

Information:

Policy impacts can be assessed and differentiated from this work. Both immediate and futuristic impacts are now well defined. Now, we can provide weights for different types of impact and rank them for our information retrieval.

4. The application of artificial intelligence in health policy: a scoping review

Paper: Link

Information: Through comprehensive literature searches and rigorous screening processes, the study identifies and synthesizes relevant articles from 2000 to 2023. The analysis focuses on AI applications within health policy contexts, utilizing Walt and Gilson's policy triangle framework to categorize and understand AI's impact on policy content, process, actors, and context. Findings highlight AI's potential to revolutionize health systems by enabling novel analyses, data collection enhancements, and evidence-based decision-making.

5. Examination of the Synthetic Control Method for Evaluating Health Policies with Multiple Treated Units

Paper: Link

Information: Chapman and Liu introduce the synthetic control method for evaluating health policies like pay-for-performance (P4P). This method constructs a weighted combination of control units to estimate treatment effects by mimicking what the treated group might have experienced without the intervention. In their study of a hospital P4P scheme (Advancing Quality), the synthetic control method yielded different results from the original difference-in-differences (DiD) analysis, suggesting no significant reduction in mortality for incentivized conditions and a significant increase in mortality for non-incentivized conditions. This method underscores the importance of rigorous policy evaluations for understanding impacts on vulnerable populations.

5 NOVELTY

A vast body of literature exists on retrieving information about government policies. On the contrary, there needs to be more literature on methods to assess the impact of government policies. The novelty of this project lies in its innovative use of technology, particularly chatbots, to determine the impact of government policies on vulnerable populations. By leveraging natural language processing and data analytics, the chatbot provides policymakers with real-time, data-driven insights into how policies affect marginalized communities. Additionally, the project incorporates stakeholder engagement and feedback mechanisms to ensure that the voices of those most affected are heard.

6 METHODOLOGY

We are proposing the following Methodology for Chatbot Implementation using an Information Retrieval System integrated with a Language Model

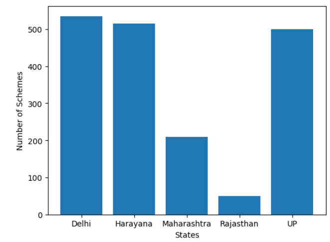


Figure 1: Number of schemes in each state

Preliminary Steps:

(1) We possess a comprehensive dataset concerning government schemes, meticulously organized within a JSON file and a csv file.
(2) API modules have been established to facilitate the retrieval of pertinent tweets from online forums such as Reddit and X.
(3) A pre-trained and fine-tuned chatbot, resembling a RAG-based model, has been successfully implemented.

Collecting the government schemes dataset:

(1) **Website Inspection:** The website https://www.myscheme.gov.in/ was visited to understand its structure and content. We identified that the website categorizes schemes state-wise, providing a list of schemes along with their URLs.
(2) **Data Collection:** State-wise inspection was conducted to gather data on schemes. Raw HTML data of the scheme lists and their URLs were copied for each state.
(3) **Data Scraping:** BeautifulSoup, a Python library, was utilized for web scraping. Details of each scheme were extracted from the scheme URLs. Information such as scheme name, description, eligibility criteria, benefits, and application process were gathered.
(4) **Data Compilation:** The scraped data was organized into a structured format. A JSON file was generated to store the compiled data, ensuring ease of access and future analysis.

Analysis

(1) Graph of Number of Schemes vs State:
(2) Graph of Number of Schemes vs Ministry:
(3) Graph of Number of Stopwords in Each Scheme:

Apify Client Initialization and Apify Actors Used

We explored various methods to incorporate our tweets and Reddit posts but encountered challenges. The Twitter API required a 100 fee, and frameworks like Tweepy and snscrape were unavailable for tweet retrieval. Collecting Reddit posts often caused the Reddit API to fail due to dataset size. After extensive searching, we discovered the APIfy Client and signed up for a trial account to access Reddit posts and tweets.

(1) We created an object instance of an apify client using the Apify User Token and initialized it into a client variable.
(2) Then we used the actors " microworlds/twitter-scraper " for scrapping X tweets and "trudax/reddit-scraper" for scraping

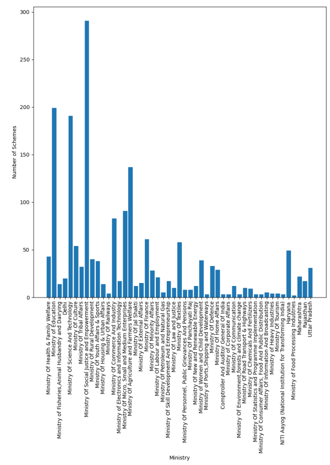
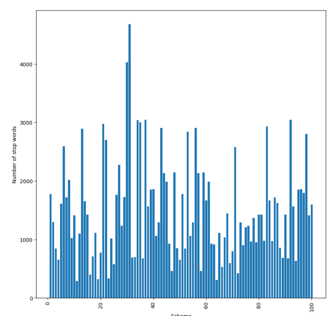


Figure 2: Number of schemes in each ministry



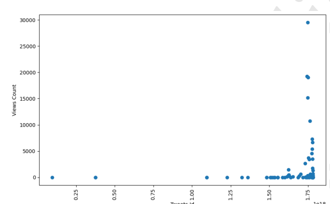


Figure 3: Graph of Number of Schemes vs State

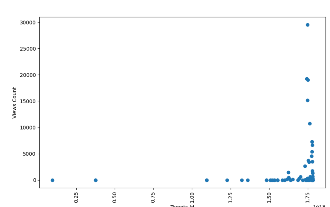


Figure 4: Graph of Number of Schemes vs State

reddit posts.

X Search:

(a) A dictionary raw input is created, which contains various parameters for configuring the X search, such as the maximum number of tweets to retrieve, whether to include user info, etc.
(b) An Apify actor for X web scraping is called with the provided input parameters.
(c) Retrieved tweets are stored in a dictionary tweets, where each key corresponds to a different attribute of a tweet (e.g., id, date, URL, full text, etc.).
(d) Some preprocessing is done to discard irrelevant or erroneous data.
(e) A loop iterates over each retrieved tweet and extracts relevant information, such as date, URL, favourite count, retweet count, etc.
(f) Extracted tweet data is stored in lists within the tweets dictionary.
(g) Finally, the collected tweet data is converted into a pandas DataFrame and saved as a CSV file named "rawtweets.csv".

Reddit Posts Search:

(a) Configuration parameters are set up in the raw input dictionary for Reddit post scraping.
(b) The Apify client calls the trudax/reddit-scraper actor with the configured input parameters (raw input).
(c) Retrieved Reddit posts are processed, and relevant information such as Id, Url, title, date, content, and community name are extracted.
(d) Extracted data is stored in lists within the 'reddit posts' dictionary' for further analysis or export.
(e) Finally, the collected Quora post data is converted into a pandas DataFrame and saved as a CSV file named "rawredditposts.csv".

(3) We then run them for each policy name variant and collect them into our dataset.
(4) We would then preprocess it, drop some empty rows reset the indexing and drop the duplicates accordingly. We chose to keep only those dataset columns which provided important information and were relevant according to our problem statement.
(5) Finally we would save them in csv files for further analysis.

Sentiment Analysis:

A positive sentiment suggests a good response, whereas a negative sentiment suggests an unfavorable one.
Before performing sentiment analysis, a preprocessing step is done.

Preprocessing Tweets and Reddit Posts:
A custom function is defined for preprocessing the text/reviews columns from X and Reddit posts dataset.

(1) **Lowercasing:** Convert all text to lowercase.
(2) **Tokenization:** Split the text into words or tokens.
(3) **Removing Punctuation:** Remove all punctuation marks from the text.
(4) **Removing Stop Words:** Remove commonly occurring words (stop words) that do not add much value to the meaning of the text, such as is, the, at etc.

(5) **Stemming:** Reduce words to their root form, keeping only the word's stem.
(6) **Lemmatization:** Reduce words to their base or dictionary form.

Here, Sentiment analysis is performed using two methods: TextBlob and Transformers pipeline. We also performed Sentiment analysis using roberta-base-go-emotions, which was trained on GoEmotions Dataset from Google for multi-label classfication, reference here https://huggingface.co/SamLowe/roberta-base-go_emotions

For **TextBlob**, A custom function is defined where a polarity score above 0 indicates a positive sentiment, while a score below 0 indicates a negative sentiment.

For the **Transformers** pipeline, A custom function is defined where a function determines the sentiment of the text data using a transformer model that has already been trained. 'POSITIVE' and 'NEGATIVE' are the sentiment labels; a simple majority vote determines which label to apply.

For the **roberta-base-go-emotions**, 'full text' content of the tweets and Reddit posts are fed as input; in cases of more than the maximum sequence length token, we would truncate the sentence to the maximum possible sequence length. Then the model would perform multi-class classification with a class size of 28 different emotions, ranging from joy, sadness, and disappointment. For each class and post, it allows a score, and we then save the results in both csv and pickle module format.

User Query Handling Process:

(1) Upon receiving a user query from the UI, exemplified by inquiries such as "Could you provide more details about the Ramalingaswami Re-Entry Fellowship program?" or "What is the feedback regarding the Ramalingaswami Re-Entry Fellowship?" or "Have the enrolled students received the fellowship under this scheme?"
(2) he system proceeds to identify the pertinent keywords or search terms pertaining to the queried scheme, exelmplified by discerning "Tell me more about the Ramalingaswami Re-Entry Fellowship program?" as the search term.
(3) Utilizing the identified search term, the system systematically scours through available tweets and reddit posts related to the query, aggregating them into distinct files, namely tweets.csv and quora posts.csv.
(4) Subsequently, sentiment analysis is conducted on the textual content of these gathered resources to categorize them as positive or negative reviews, reliable or fake reviews, alongside keyword extraction is also performed to ascertain the presence of rare vocabulary within the corpus.
(5) Following the acquisition of reviews, a term frequency-inverse document frequency (tf-idf) analysis is executed on the dataset, followed by leveraging composite similarity to rank the most relevant results, which includes favourites count and followers count, etc.
(6) The collated information is then seamlessly relayed to the RAG system chatbot, which adeptly formulates an appropriate response tailored to address the user's query.

User Interface

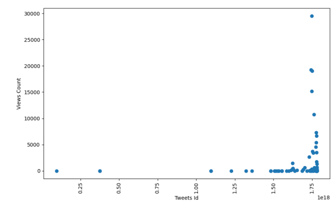


Figure 5: Tweets vs Reviews Count

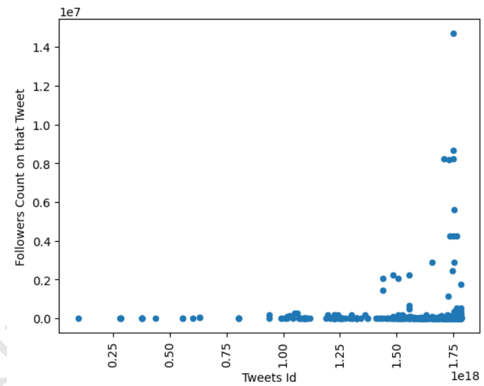


Figure 6: User Followers Count on Tweets

The User Interface consists of an input box that takes the input from the user and presents the output for the query. User Interface is made with the Django framework because Django is python based and our training model is written in python. The chatbot utilizes RAG through LangChain. We tried out many LLMs and transformers by OpenAI, HugginFace to fit LangChain. In the end, we are using Llama3 by Meta as the LLM driving the chatbot. The sentiment analysed tweets from earlier are fed into the LLM by first loading them, vectorizing them, and making a retrieval chain. A verbose prompt template is provided to the LLM to make its ouput more relevant.

7 DATABASE

Government Schemes Data:
We outline the data collection process of the policies from the website https://www.myscheme.gov.in/. The objective was to gather information about various government schemes listed on the website, organize it by state and ministry, and compile the data into a structured format for further analysis.

Since there was no API or predefined dataset to gather this information, we utilized web-scrapping methods manually to find the relevant information regarding the policies. Additionally, our focus was on state policies that predominantly affect vulnerable populations.

X Data:
The dataset consists of structured information from tweets, each represented by attributes including ID, date, URL, full text, favourite count, and retweet count. This data was collected using an Apify

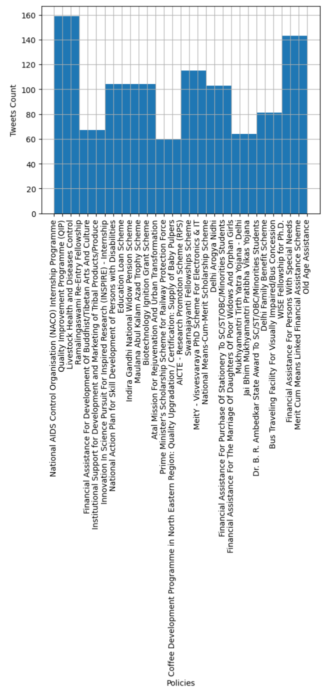


Figure 7: Government Policies vs Tweets Count

actor configured to scrape Twitter based on specific parameters such as the maximum number of tweets and inclusion of user info. Post-collection, the data underwent preprocessing to remove irrelevant entries, and relevant tweet details were then extracted and stored in a structured format. Finally, this data was organized into a pandas DataFrame and saved as "rawtweets.csv" for further analysis.

Reddit Data:
The dataset comprises structured information from Reddit posts, with attributes including Id, Url, title, date, content, and community name. Data collection was facilitated by using the Apify client to invoke the trudax/reddit-scraper actor, configured via the raw input dictionary. After retrieving the posts, I processed and extracted pertinent information, which was then organized into the 'reddit posts' dictionary. This processed data was converted into a pandas DataFrame and subsequently saved as "raw-reddit-posts.csv" for further analysis.

8 EVALUATION

(1) **Go-emotions vs other models for sentiment analysis:** We compared the GoEmotion model with other prevalent models in the field. The GoEmotion model, developed by Google, is specifically designed to detect a wide range of emotions in text ,i.e, 28 sentiments, offering a nuanced understanding beyond basic sentiment polarities (positive, negative, neutral).
For comparative analysis, we assess models like VADER and TextBlob, which focus primarily on polarity detection. Our evaluation metrics include accuracy, precision, recall, and F1-score. The GoEmotion model exhibits superior performance in recognizing complex emotional states due to its fine-grained classification capability. However, it requires more computational resources and larger datasets for training compared to VADER and TextBlob, which are lighter and faster but less nuanced in their sentiment analysis.

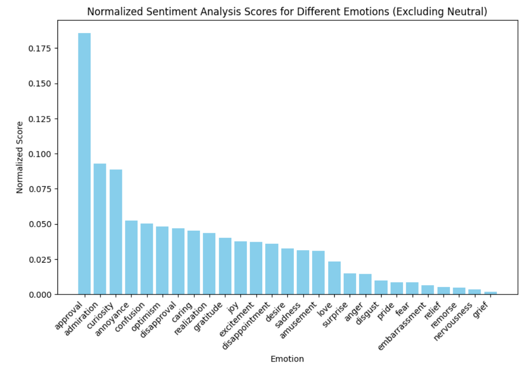


Figure 8: Normalized the sentiment scores for Tweets other than Neutral Emotion

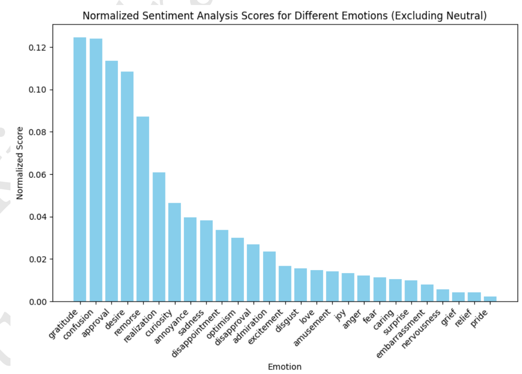


Figure 9: Normalized the sentiment scores for Reddit Posts other than Neutral Emotion

(2) **LAMA vs other models for chatbot:**
In evaluating the LAMA (LAnguage Model Analysis) approach against other models for chatbot applications, several key aspects were considered: the quality of response, contextual understanding, knowledge retention, and user engagement.
(a) **Quality of Response:** LAMA, based on probing language models for factual knowledge, generally provides high-quality responses that are factually accurate and informative. Compared to traditional rule-based systems, LAMA offers more flexible and dynamic responses, adapting to various topics effortlessly. However, it may lag behind specialized fine-tuned models in niche domains where specific expert knowledge or terminology is crucial.

(b) **Contextual Understanding:** LAMA excels in understanding context due to its training on diverse datasets. This capability is superior to simpler models like ELIZA or keyword-based systems that lack deep understanding. However, state-of-the-art models like GPT-3 or BERT might outperform LAMA in nuanced context handling due to their advanced architectures that focus more heavily on context through mechanisms like attention and transformers.

(c) **Knowledge Retention:** LAMA's ability to retain and recall factual information is notable, making it reliable for informative interactions in a chatbot setting. While it performs better than basic models, it can occasionally be outperformed by memory-augmented neural networks or continuously updated models that can learn and store new information over time.

(d) **User Engagement:** In terms of engaging users, LAMA's responses are coherent and contextually appropriate, which sustains user interaction longer than simpler models. Nonetheless, the most advanced conversational agents, which might use reinforcement learning to optimize responses based on user satisfaction, could potentially engage users more effectively by tailoring conversations more strategically to individual preferences.

9 CONTRIBUTIONS

(1) Literature Review: Arpan, Daksh, Tony, Kajol, Riya, Atharv
(2) Data extraction using API and web scraping: Arpan, Tony
(3) Preprocessing and sentiment analysis: Riya, Kajol
(4) Design and development of chatbot: Arpan, Daksh, Tony

10 REFERENCES

(1) Jens Wirdenius, CO-founder at Veloce Network (2017-present) "Twitter monthly active users".
(2) Anamalamudi, Komala, and YCA Padmanabha Reddy. 2021. Context-based News Articles Retrieval using CLSM. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9418018&tag=1.
(3) Rezak, Mohamed A., and Ghada E. Khayat. 2016. (March). https://dl.acm.org/doi/10.1145/2910019.2910093
(4) Ahmed, Wasim. 2015. "A list of tools to capture Twitter data." (January). https://www.researchgate.net/publication/304097418_A_list_of_tools_to_capture_Twitter_data
(5) "Policy Impact and Evaluation." n.d. https://egyankosh.ac.in/bitstream/123456789/19391/1/Unit-27.pdf
(6) The application of artificial intelligence in health policy: a scoping review https://bmchealthservres.biomedcentral.com/articles/10.1186/s12913-023-10462-2
(7) Examination of the Synthetic Control Method for Evaluating Health Policies with Multiple Treated Units https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5111584/
(8) A Government Decision Analytics Framework Based on Citizen Opinion https://dl.acm.org/doi/10.1145/2910019.2910093