DALHOUSIE UNIVERSITY

# CSCI 6612 VISUAL ANALYTICS

## PROJECT REPORT

# MetaVeo
# Inside Rajya Sabha: Visualization of Q/A hour


Team Members

Aman Jaiswal (B00857194)
Email: aman.jaiswal@dal.ca

Neelkanth Dabhi (B00848532)
Email: neelkanth@dal.ca

Ruminder Singh (B00854875)
Email: ruminder.singh@dal.ca

# Contents

# Abstract

India is the second most populated country in the world making it the largest democracy. The Rajya Sabha is the upper house of the bicameral parliament of India that held planned sessions to discuss political bills. These sessions serve as an opportunity for the ministers to point out the shortcomings of the government in the form of questions and direct the attention of the government to relevant problems and ultimately help in formulating policies. Through this project, we aim to visualize the various aspects of these questions and their origins. Furthermore, the textual data is leveraged by training a Word2Vec model on question description and visualizing these embeddings, enabling exploration of questions that are closely related. In our knowledge, this is the first attempt to visualize a similar question of this dataset using embeddings. The Dashboard is intended to be used for exploratory data analysis and help the user make informed decisions.

Note: The original proposal was to perform univariate time-series analysis from a user given data and perform forecasting based on seasons. We changed our project to the above topic.

# Introduction

The Rajya Sabha or also known as ‘Council of States’ was formed on 23rd August 1954. Its maximum capacity is 250 out of 12 nominated by the president and the remaining 238 are the representatives of the state and union territories. As Rajya Sabha is a federal chamber, it holds some special powers under the constitution of India. For example, all the bills that are passed in Lok Sabha will be transmitted to the Rajya Sabha for its recommendation. Here all the members from different parties and states debate on the bills mostly in the form of question and answers, and all these questions and answers are made public by the Rajya Sabha. [1]

Discussions and procedures in Rajya Sabha sometimes get complicated to understand for someone who does not know much about how things work in the parliament. This is where visualizations come into the picture to provide the underlying knowledge of the data. We have tried to build a dashboard where a person can get a bigger picture of what is going in the parliament. We are also performing Natural language processing on questions description so that users can filter questions that they are interested in.

# Motivation

This project provides a user with a dashboard to perform exploratory data analysis with a dataset that is yet to be leveraged properly and gain insights into the workings of the Indian government and the major influencers of Indian policymaking. We believe this project can help users to gain some factual basis on how policies are motivated and the ministers who rally for particular matters. This can help the average Indian citizen to have a broader perspective on the interests of ministers and their shift in motivations over the years. Democracy is empowered through information. Ultimately, the aim is to help the curious voters gain a better understanding of the political stance of ministers through interactive visualization and make informed decisions during the election.

# Dataset

The original dataset is available at Kaggle[2] made available by Rajanand Ilangovan. We combined it with additional information taken from Wikipedia. The original dataset contains columns such as the question, answer_date, ministry, question_by, question description, and the answer. In addition to that, we needed information of members of Rajya Sabha for that we acquired data from Wikipedia and the Rajya Sabha website. We created other datasets with information such as member name, party affiliation, state they belong to, and their ministry if any.

# Visualizations

## Persistent Module

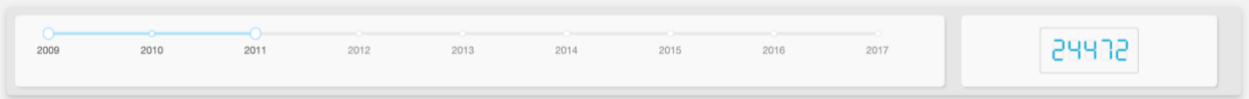


*Figure 1: Persistent Slider*

Since, the dataset spanned multiple years from 2009 to 2017, containing around 89000 samples. we used a year range slider Figure 1, to allow users to focus on the years they are interested in investigating. This year-range slider modifies data across multiple charts and indicates the number of rows present in the selected date range.

## Bar Chart : Total Questions vs Ministries

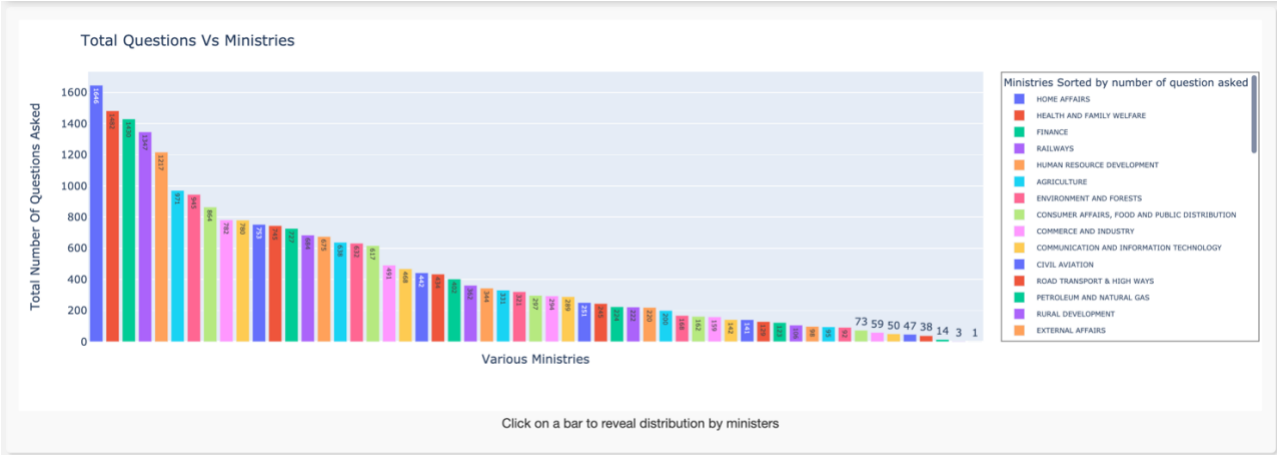


*Figure 2: Total question Vs Ministries.*

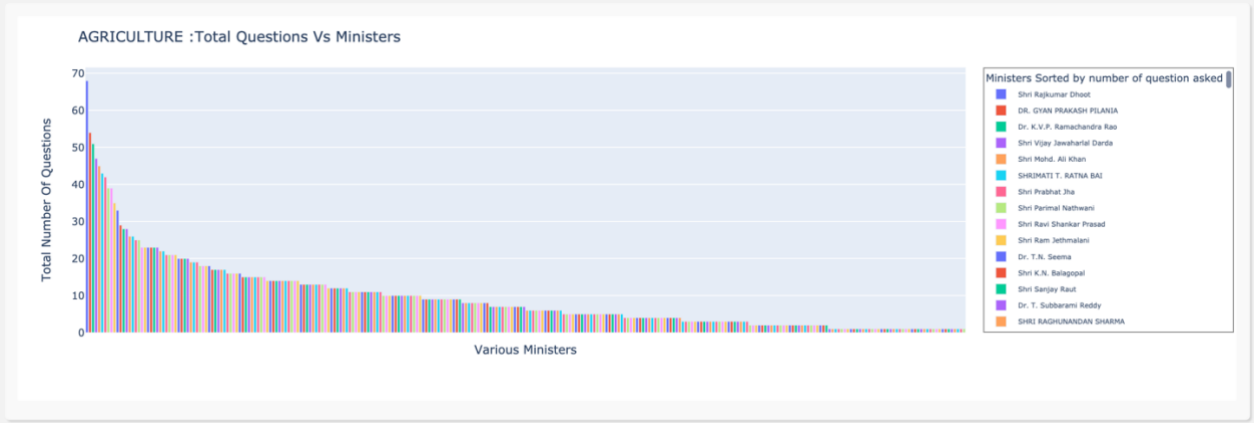


*Figure 3: Total questions vs ministers*

The first visualization in Figure 2, gives an overview of the most popular ministries to the least popular ministries. This gives an idea to user about the which ministries were being asked the most question for the selected period. The bar chart is good for comparing quantities visually. Since we sort the values before plotting it becomes even easier to compare different ministries.

This bar chart can help answer questions like “Which ministry was asked the maximum question in the period 2009- 2012? which was the second maximum?”. It can be immediately observed that Home Affairs was the most popular followed by Health and Family Welfare. The legends on the right-side also aid the visualization to give a sorted list of the bars representing different ministries.

## Bar Chart 2: Total Questions vs Minsters

The Second bar chart aims to give a second level of resolution to the first bar chart. A click event on any of the bars from Figure 1, updates Figure 2 with the distribution of ministers and total questions they asked to the respective ministry in the selected period range. Bar chart again makes it easy to compare different ministers. The title of this chart is dynamic to the name of the clicked bar to remove any ambiguity. The legend is sorted to question counts to make it easy to comprehend which ministers were most active with the selected ministry. This bar chart helps in answering questions like “Which minister posed maximum or minimum questions to the agriculture ministry in the year 2013?”, It can be noted from legends on the right side that Shri Rajkumar dhoot was the most active with agriculture ministry with around 70 questions in the year 2013.

## Histogram: Distribution of question overtime

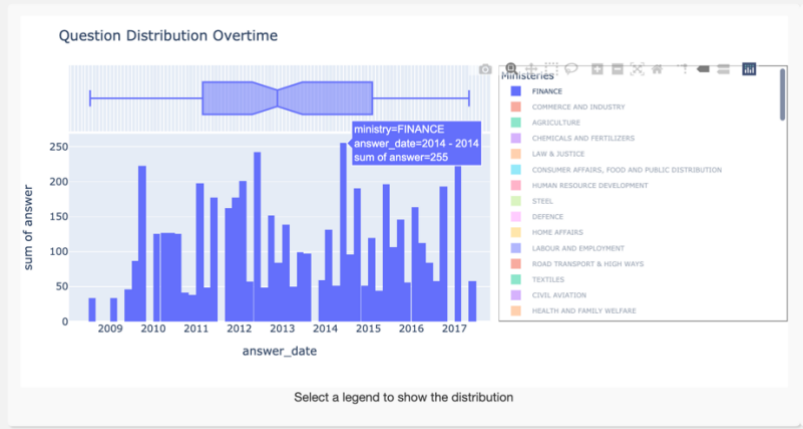


*Figure 4: Question over-time for the finance ministry*

The third figure is a histogram with the Time dimension on the x-axis and a marginal with a box plot. The legends box serves as additional interactivity to only view the ministries we are interested in, all the other legends are turned off by default. This information is important to present to the user to reveal the trends that may exist over time. Figure 4, shows the distribution of questions to the finance ministry. This can help answer questions like “What was the distribution of questions to the finance ministry in the year 2013 compared to 2014”. We can observe from the histogram that after the initial peak to the finance ministry in 2013 the inflow of questions declined

## Histogram: Comparing two ministries

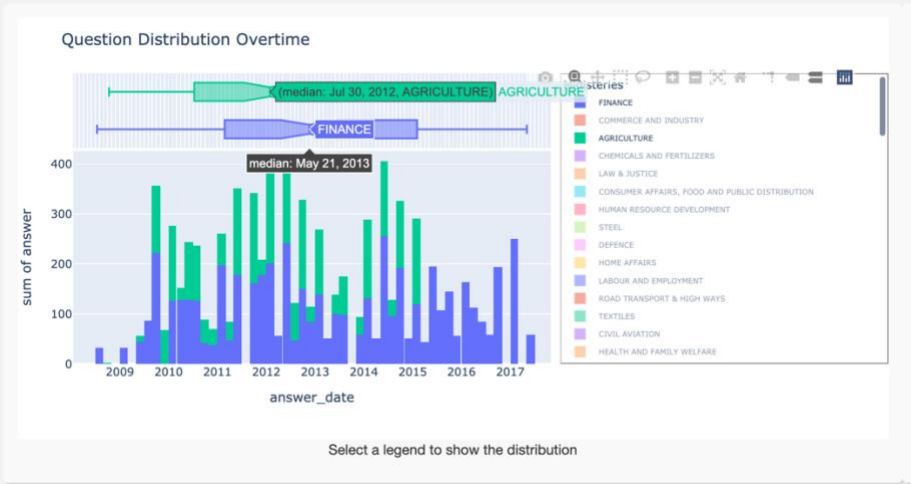


*Figure 5: Comparing two ministries overtime*

It has been the theme of our project to use the legend box for interactivity. In this mode, the same histogram can be used to activate multiple ministries and compare their question distribution overtime. The bars of histograms are stacked with the question count of the selected ministries. The marginal box-plot on the top of the histogram uses the time dimension as its axis, it can be used to compare and find the median dates of the distribution. It can be noted from the above graph that 50% of questions asked to the finance ministry were answered by May 21, 2012, while the same median date for the agriculture ministry is July 30, 2012

## Sunburst Chart: Investigating histogram

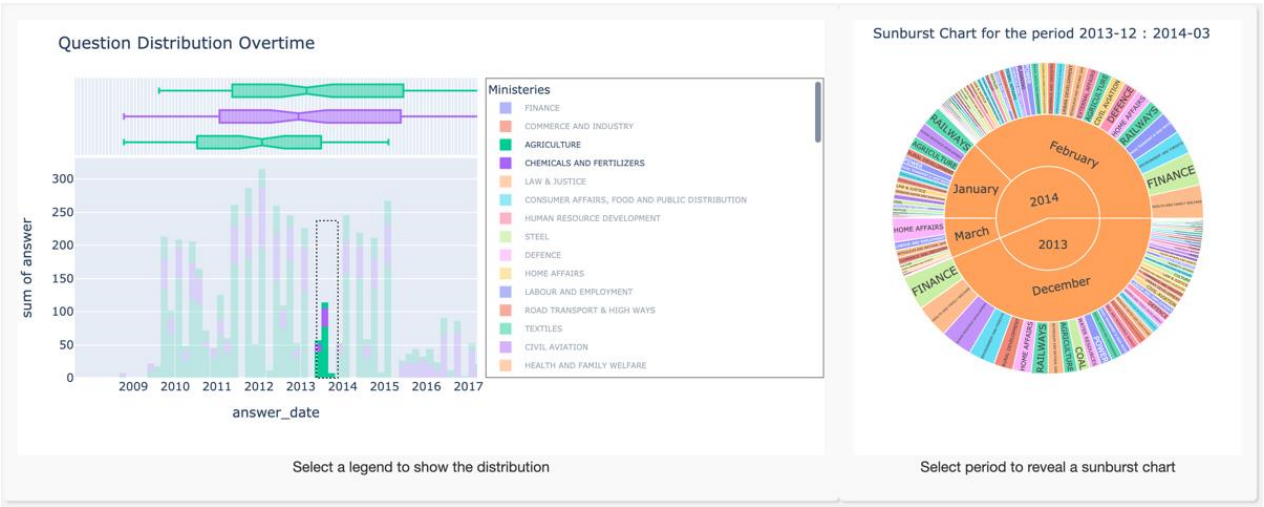


*Figure 6: Sunburst chart for the selected period*

The sunburst chart gives additional context for any sub-date-range that the user might be interested in investigating. The sunburst chart in figure 6, is updated when the box-select tool is used to select a period in figure 5. The title of the sunburst chart is again dynamic to remove any ambiguity. It can be used to immediately see which other ministries were popular in the given range and their distribution over the months and even years depending on the selected range. We used a sunburst chart as it's useful to navigate the hierarchical structure of time dimension (years - months).

**Improvement after presentation**

The month's values were changed to reflect the actual month in a year

## Parallel Coordinate chart: Flow of question from states, parties to ministries

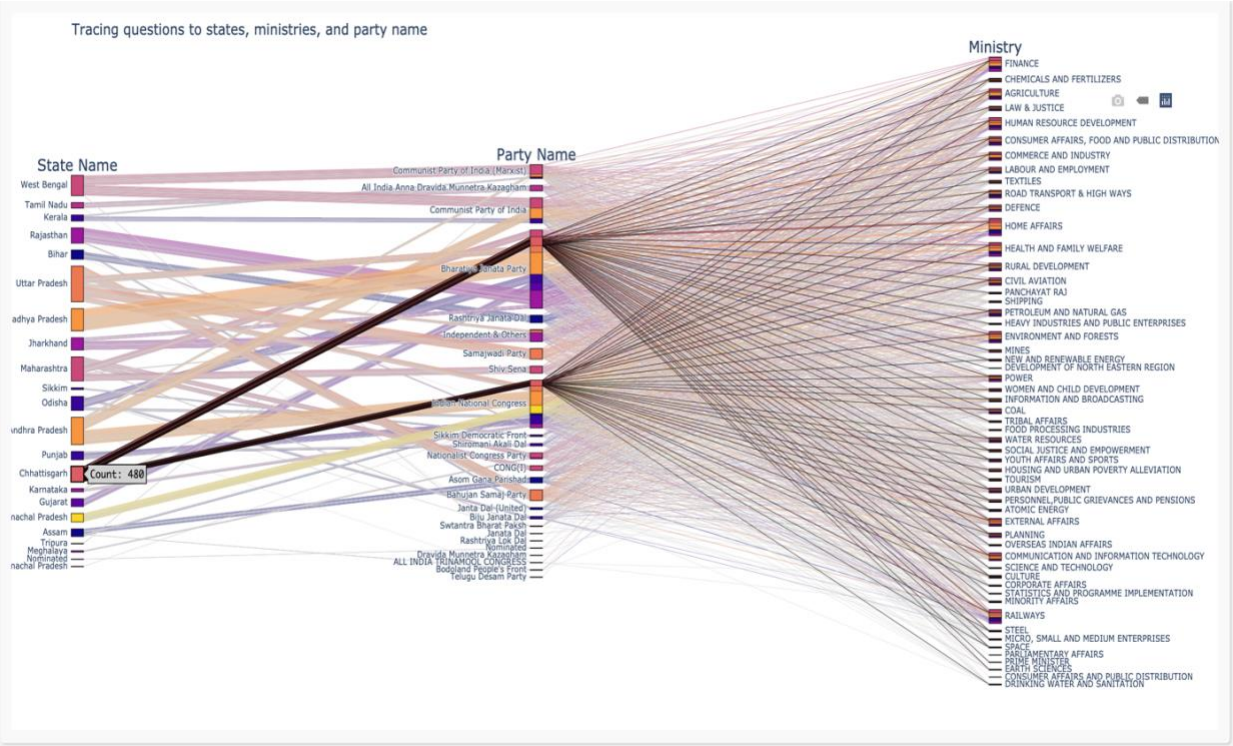


*Figure 7: Tracing questions from states, Parties to Ministries*

Here, we combined the original Kaggle [2] dataset with additional information from [3] to include the origin state and party affiliations of the ministers. The parallel coordinate chart is useful to visualize the flow of questions from different states and parties. The user can hover over any state to view the party affiliations of members coming from that state. Also, the user can hover over any party to check the ministries they influence the most. The above figure 7, shows the minsters from Chhattisgarh were affiliated with the two leading opposition central parties from 2007-2009.

## Heatmap: Ministries vs State

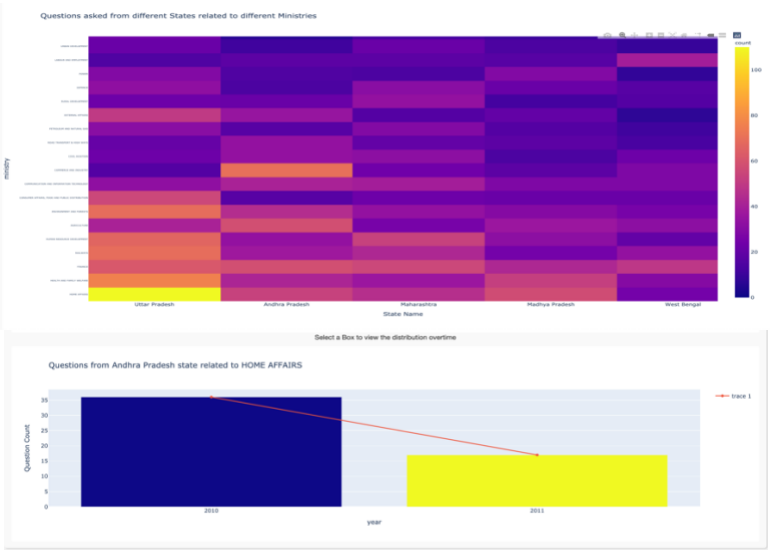


*Figure 8: Heatmap of Ministries vs State, Distribution for AP and Home Affair*

The First heatmap showcases all the ministries against the origin states of ministers. This heatmap is again affected by the year range slider, indicating which state posed the maximum questions to a particular ministry for the given date range. All the different boxes in the heatmap indicate a combination of a state and ministry. Heatmap is useful to view the most popular combinations in a single graph and compare them. We added additional interactivity which gets updated upon a click event on any of these boxes. The click event updates a bar chart to showcase the distribution of questions over the years for the selected combination. Figure 8 shows the distribution of questions from Uttar Pradesh to Home Affairs, it shows a sudden decline in the number of questions in the year 2013.

**Improvement after presentation**

1. The labels of the heat map are sorted with their value counts, this helps to show the proper ordering of data, which is essential in a heatmap.
2. We also added a heatmap to showcase all the ministers and ministries for the selected year range.

## Heatmap: Minister vs ministries

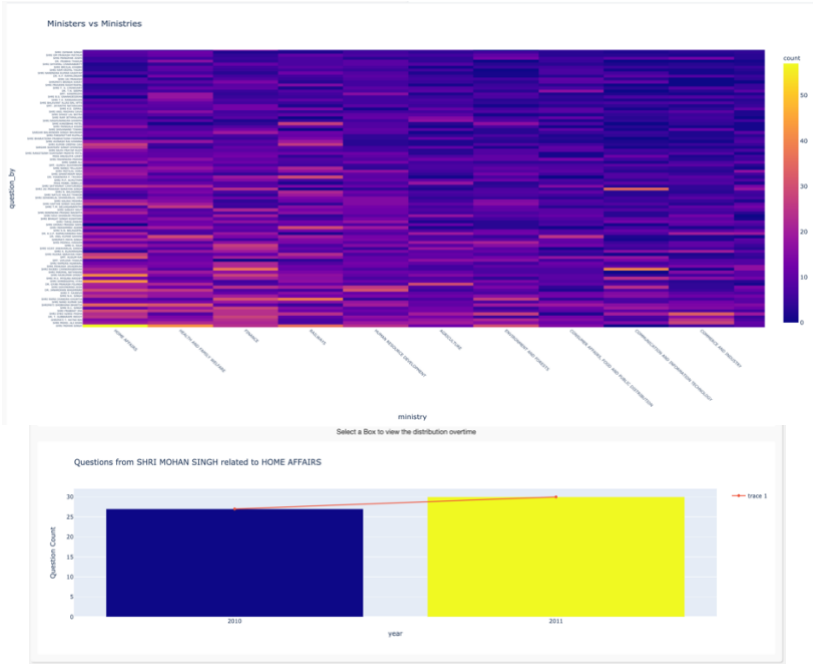


*Figure 9: Heatmap of various ministers against various ministries*

The heatmap on figure 9 gives a broader perspective of the same information available in the first two bar charts in Figure 2,3. Additionally, it allows to immediately focus the attention on the interesting combination of ministers and ministries. A click event on any of the boxes reveals the distribution of years throughout the selected year-range.

## Choropleth Map and Pie Chart

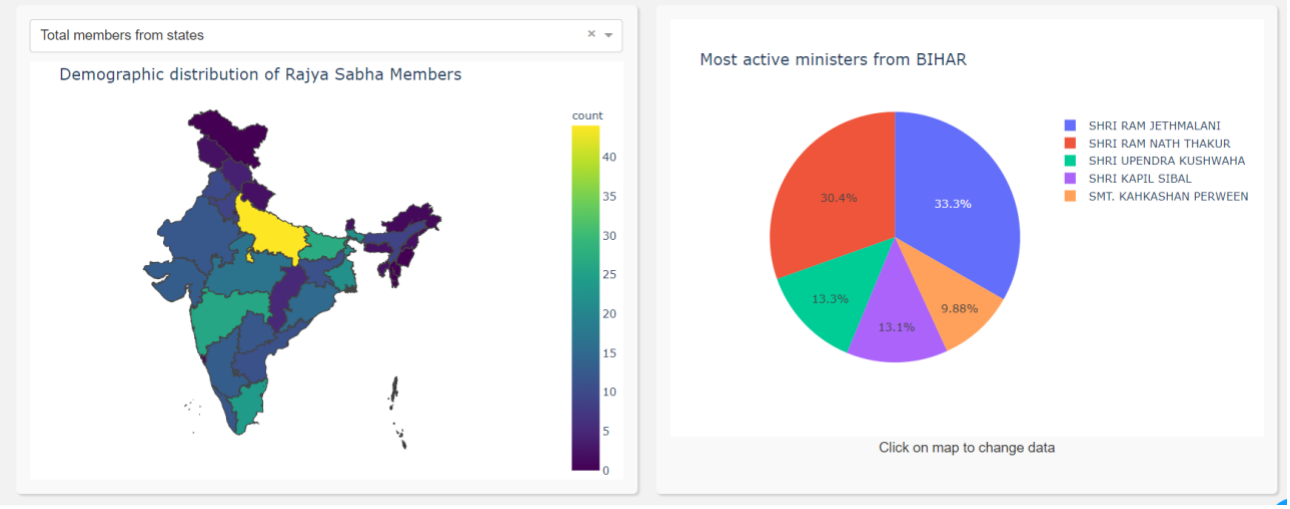


*Figure 8 Members from states*

When we talk about politics, the demographic of politicians becomes important. Here we have provided a choropleth map which shows how many members coming from the particular state, we can see that majority of members come to from the central and western part of the country. On the right-hand side, we have provided a pie chart of the most active members from a particular state. Pie chart changes based on the clicked state in the Choropleth map.

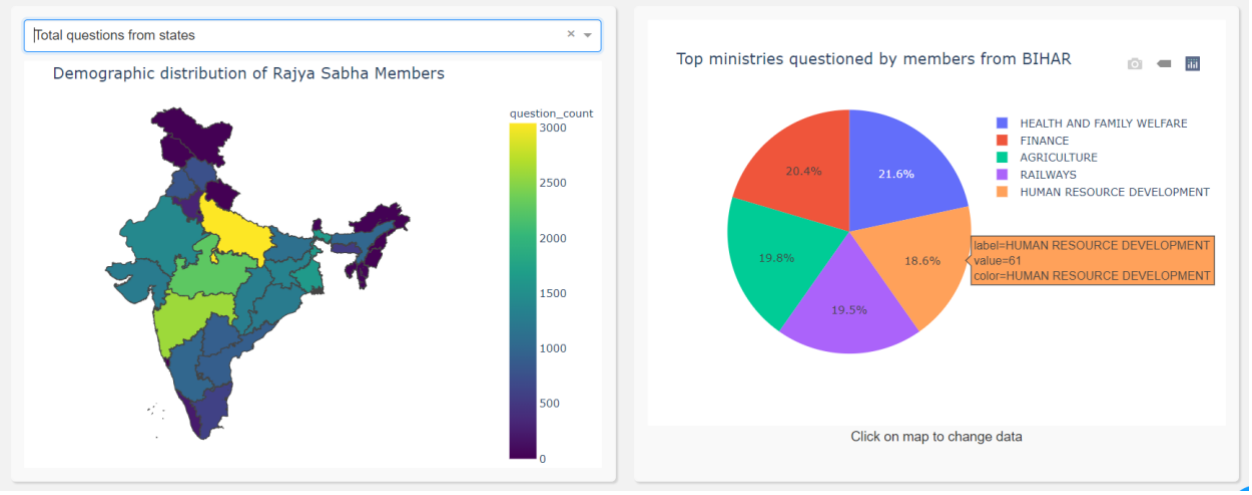


*Figure 9 Questions from state*

In figure 11. We have visualized the number of questions asked to different ministers from the members of a particular state. This type of visualization can help us understand the interests of the state, in the right-side pie chart we have provided Top ministries questioned by members from Bihar. The top 3 ministries are Health, Finance, and Agriculture. This aligns with Bihar's Health and Financial crisis over the past decade.

# Natural Language Processing

The original Kaggle dataset contains a large amount of textual data in the form of question description and answers. There is valuable information in these questions description that needs to be leveraged using proper NLP techniques. Our idea is to vectorize this question description and visualize the embedding space. This space can provide a way to interact and explore these questions in a way that is not overwhelming to the user.

## Approach

There are multiple ways to create embedding for textual data. Recent approaches rely on complicated neural network architectures that preserves semantic relationships [4]. Looking at the novelty of data and our constrained resources we decided to utilize simple architectures that preserve relationships between similar and opposite words. The two algorithms we considered are Word2Vec [5] and Doc2vec. Doc2vec is capable of obtaining corresponding vectors for each sentence. Whereas, Word2Vec provides a vector for each word in the corpus. Word2Vec is a two-layer shallow network that is trained to predict the nearby words in a corpus of text. we used the gensim library [6] to implement and train the models. Before the question description can be used as input for these models, they need to be preprocessed. We used the NLTK [7] library for preprocessing. The Prep-processing steps include lower casing, removing digits, punctuations, stop words, lemmatizing, and tokenizing the words.

```
model_doc.wv.most_similar('rupee')
```

```
[('penny', 0.9314305186271667),
 ('fivefold', 0.923876941204071),
 ('i', 0.9166343808174133),
 ('mgfemmar', 0.9160035848617554),
 ('moreb', 0.9147072434425354),
 ('croreslno', 0.9075325727462769),
 ('pricemep', 0.9030519127845764),
 ('worth', 0.8997265100479126),
 ('tune', 0.8984656929969788),
 ('precariously', 0.8956177234649658)]
```

```
model.wv.most_similar('agriculture')
```

```
[('agricultural', 0.780207097530365),
 ('farm', 0.6822717189788818),
 ('fishery', 0.6765708923339844),
 ('horticulture', 0.6712913513183594),
 ('farming', 0.644605815410614),
 ('dairy', 0.6073001623153687),
 ('msme', 0.5872455835342407),
 ('manufacturingconstruction', 0.569352388381958),
 ('msmes', 0.5652278661727905),
 ('husbandry', 0.5626657009124756)]
```

*Figure 12: a) Similar words to "rupee" in Doc2vec b) Similar words to "agriculture" in word2vec.*

The Word2vec model was trained for 100 iterations with a vector size of 100. Figure 12, shows the most similar words to "rupee" and "agriculture" respectively from the two models. The word2vec model outputs the individual vectors for each word and we require to have an embedding for every question description where each question description contains multiple words. To solve this issue, we used the mean of all the word vectors as the vector for a question description. We used Word2Vec in the final approach for simplicity and reasonable results.

## Visualizing the embeddings

The obtained embeddings cannot be directly used for visualization due to its high dimensionality. We used a combination of dimensionality reduction techniques for our final approach, First, the 100 dimensions of vectors are reduced using PCA. The obtained PCA components are then used as input for T-SNE projections. The benefit of using PCA before T-SNE is that saves computation cost and time which is important while making an interactive visualization. This approach will suppress some noise and speed up the computation of pairwise distances between samples.

## Scatter 3D

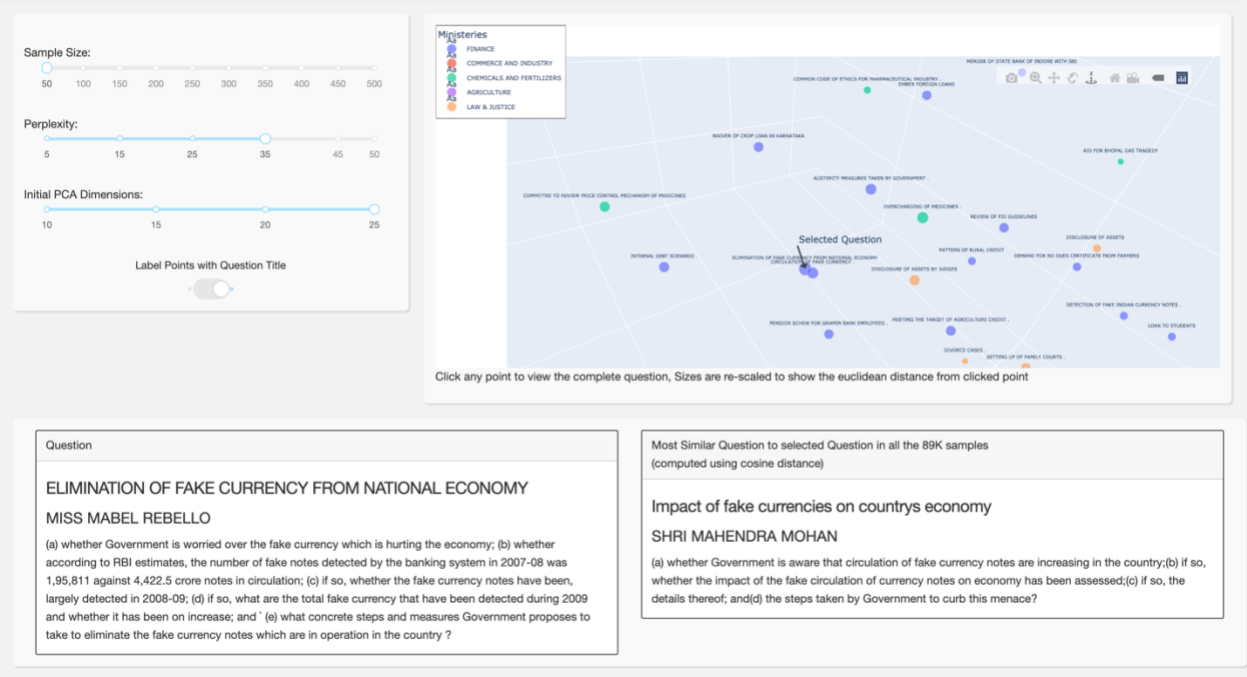


*Figure 13: Question description embedding visualization using scatter 3d*

Scatter 3d is used to visualize the embeddings since it allows us to explore the projection space and offers a better engagement than scatter 2d. The visualization comes with three sliders that control the sample size and the parameters used to obtain the projections of the question vectors. The “initial PCA dimension” slider controls the number of dimensions passed to the T-SNE algorithm as input. The second slider controls the perplexity of the T-SNE algorithm. Usually, higher values of perplexity have a tendency toward clearer shapes in the projections. The scatter 3d plot in Plotly allows for zooming, spanning, and rotating about an axis, etc. Each point in the scatter 3d graph represents a question in our dataset and is annotated with the question title. The idea is to have similar questions closer to each other. A click event happens when a point in the plot, the selected point is annotated, the two cards below are updated and the sizes of points are rescaled. The first card shows the selected question title, who asked the question, and the whole question description itself. The next card shows the same information but for the most similar question (using cosine distance) in all the 89000 samples to the selected one.

**Improvements after presentation**

The limitation of using scatter 3d is that depending on the perspective of the viewer some points may look closer than they are. To overcome this issue, we did two things:

1. The sizes of each of the points in the plot are *re-scaled* with a factor that is computed from the *Euclidean distance* of the points from the selected point. It means the points which are closer to the clicked point would appear bigger than points farther away.
2. The axes of scatter were kept in the final project to help the viewer have a better perspective about which points are closer.

## Evaluation

We used the qualitative method to evaluate our approach. Firstly, to ensure the initial word vectors were a good representation of the meaning of words, we found the nearest words to some specific words using cosine distance. In Figure 12, the nearest words to "Agriculture" are found as "farm", "fishery"," horticulture" etc. The obtained nearest words were good enough representation for our application.
Our final goal is to find questions that are similar in meaning to a specific selected question. We used the cosine distance metric to find such question description. Each question belonged to 100-dimension vectors. The cosine distance is computed for a combination of vectors to find the nearest question vectors. The outcome is again evaluated qualitatively. The following table shows a similar question to some samples.

The results are appropriate for our application.

| Question Title | Most Similar Question Title |
|---|---|
| Elimination of fake currency from the national economy | Impact of fake currencies on the economy |
| Promotion of foreign breed of cows | A decrease in the number of indigenous cows |
| Overcharging of medicines. | The price rise of essential medicines. |
| Purchase of Gold By RBI | Liquidity to Multilateral Institutions by Purchasing Gold |
| Pending court cases. | Setting up of commercial courts. |
| Merger and acquisitions of public sector banks. | Merging of Psbs |

## Optimizations

Apart from using a combination of dimensionality reduction methods, we performed the following optimizations to help with the visualization of vector embedding:

1. We used only 500 samples for visualization in scatter 3d.

2. The nearest vector of these 500 samples is precomputed and their index is stored in a separate data frame. Therefore, a similar question is instantly available after a click event and there's no need to compute the cosine distance against 89000 every time.

# Tools And libraries Used

- Plotly
- Pandas
- Numpy
- NLTK: Preprocessing
- Genism: Word2vec
- Scikit-Learn:  PCA and T-SNE.

# Conclusion

The Rajya Sabha plays an important part in Indian democracy. The Question hour inside Rajya Sabha plays a crucial role in determining the upcoming policies. The project aimed to combine the original Kaggle dataset with additional information from Wikipedia to gain insights that would be beneficial for voters to understand the motivations of ministers. Considering the value of the data, we relied more on visualization rather than using a lot of machine learning to help the analytics. Additionally, we used NLP to leverage a large amount of textual data to visualize the question description. The persistent year range slider updates the most of visualization. All the visualization is selected carefully to add value and make the vast information comprehensible in a visually appealing manner. We tried to provide context to the graphs wherever possible through additional interactivity. In our knowledge, this is the first project to visualize the Rajya Sabha questions description through embeddings obtained from word2Vec and provide an exploratory space to find a similar question and their details.

# Future Work

The project tries to cover a lot of ground about the data and the kind of insights a user can obtain from it. The majority of visualization aims to find relationships between ministers, miniseries and the question description. There are certain improvements that can be made such as:

- Include a continuous stream of data from the Rajya Sabha website and use dynamic storage.
- Use more modern deep learning architecture such as BERT [4] to obtain the embeddings for question description.
- Visualize the embeddings for answers along with the question description.
- Include a circos diagram to establish a relationship between the asking and answering ministers.

# References

[1] "Rajya Sabha Introduction", *Rajyasabha.nic.in*, 2020. [Online]. Available: https://rajyasabha.nic.in/rsnew/about_parliament/rajya_sabha_introduction.asp. [Accessed: 02- Dec- 2020]

[2] "Q & A Discussed in Parliament of India", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/rajanand/rajyasabha. [Accessed: 06- Dec- 2020]

[3] Wikipedia Contributors, "List of Rajya Sabha members from Bihar," *Wikipedia*, Nov. 13, 2020. https://en.wikipedia.org/wiki/List_of_Rajya_Sabha_members_from_Bihar (accessed Dec. 06, 2020).

[4] J. Devlin, M.-W. Chang, K. Lee, K. Google, and A. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2019. [Online]. Available: https://arxiv.org/pdf/1810.04805.pdf.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality." Accessed: Dec. 06, 2020. [Online]. Available: https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

[6] "gensim," *PyPI*, May 04, 2020. https://pypi.org/project/gensim/ (accessed Dec. 06, 2020).

[7]"nltk," *PyPI*, Apr. 12, 2020. https://pypi.org/project/nltk/ (accessed Dec. 06, 2020).