

NLP Assignment: Text Analysis and Topic Discovery

Introduction

This assignment focuses on Natural Language Processing (NLP) techniques to analyze and interpret unstructured text data. The primary goal is to transform raw text into meaningful insights by applying preprocessing, feature extraction, word embeddings, and topic discovery methods.

Objective

The dataset used includes diverse text samples such as earthquake-related sentences, e-commerce reviews (Amazon & Flipkart), and watch brand descriptions (Japanese brands). This variety ensures a broad evaluation of NLP techniques across different domains.

The assignment demonstrates a complete NLP pipeline:

- Text Cleaning & Tokenization to prepare raw text.
- TF-IDF Analysis to extract the most important words in each document.
- Word2Vec Embeddings to capture semantic relationships and similarities between words.
- Latent Dirichlet Allocation (LDA) for uncovering hidden topics and assigning them to documents.

Methodology

Text Preprocessing

Before applying NLP techniques, the raw text documents were cleaned and prepared using the following steps:

1) **Lowercasing:** Converted all text to lowercase

2) **Removing Punctuation & Numbers:** Used regular expressions (`re.sub`) to eliminate punctuation marks, digits, and special characters, leaving only alphabetic words.

3)**Tokenization:** Split sentences into individual words (tokens) using NLTK's `word_tokenize`.

4)**Stopword Removal:** Removed common English stopwords that do not contribute meaningful information. Like in, and, an, etc...

TF-IDF (Term Frequency-Inverse Document Frequency)

1)Preprocessed the dataset by converting text to lowercase, removing punctuation, and eliminating stopwords.

2)Transformed the cleaned text into numerical vectors using **TF-IDF Vectorizer**.

3)Extracted the **top 10 words with highest TF-IDF scores** for each document to identify the most important terms.

Word2Vec Embeddings

1)Trained a Word2Vec model on the tokenized dataset to learn word embeddings.

2)Generated vector representations for each word, capturing contextual and semantic meaning.

3)Queried the model to find the 5 most similar words for chosen keywords (e.g., *earthquakes*, *Amazon*).

4)Reduced embeddings to 2D using PCA and visualized them in a scatter plot for interpretation.

Topic Modeling (LDA – Latent Dirichlet Allocation)

1)Converted documents into a Bag-of-Words (BoW) representation using a dictionary and corpus.

2)Trained an LDA model with 4 topics, each representing a hidden theme in the dataset.

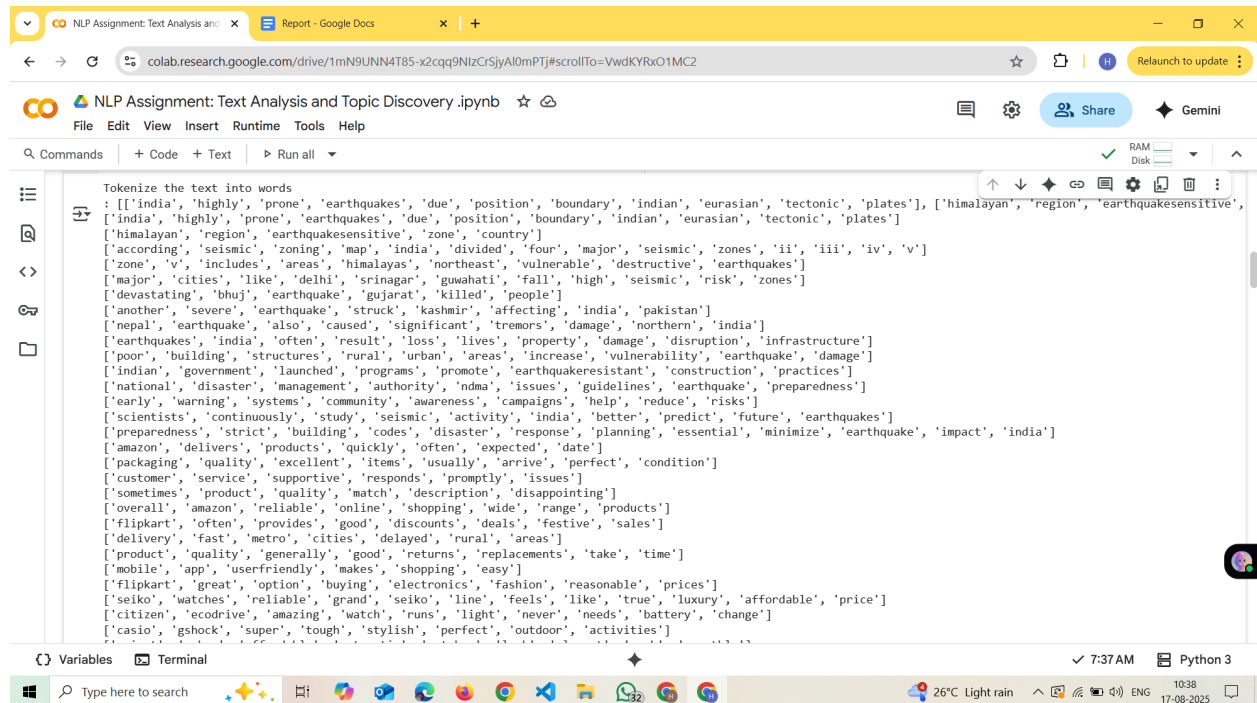
3)Extracted the top 5 keywords per topic to define topic meaning.

4)Assigned the most relevant topic to each document based on topic probability distribution.

5)Used pyLDAvis for an interactive visualization of topics and their relationships.

Results & Observations

Text Preprocessing: In text preprocessing, we observed that raw sentences were cleaned and transformed into standardized tokens by removing punctuation, stopwords, and converting to lowercase.

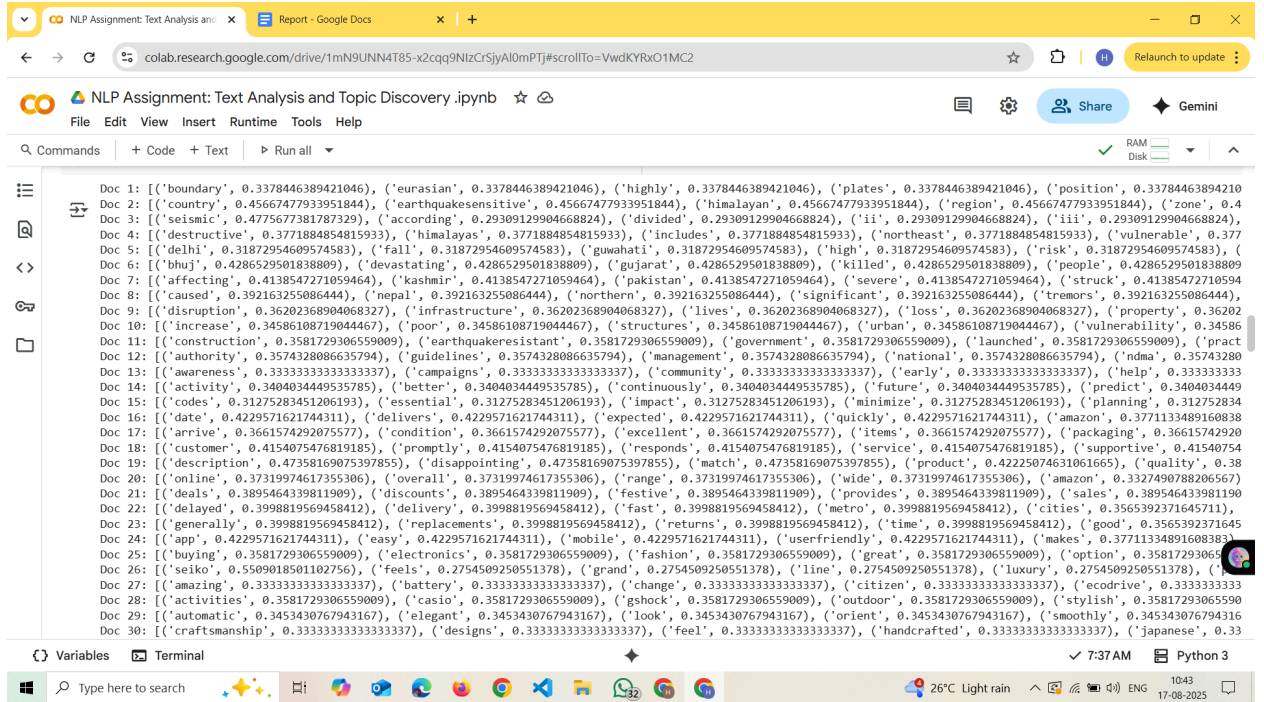


The screenshot shows a Google Colab notebook titled "NLP Assignment: Text Analysis and Topic Discovery.ipynb". The code cell is titled "Tokenize the text into words" and contains a list of tokenized words from various text sources. The tokens are organized into several groups, each representing a different category of text. The tokens are as follows:

```
['india', 'highly', 'prone', 'earthquakes', 'due', 'position', 'boundary', 'indian', 'eurasian', 'tectonic', 'plates'], ['himalayan', 'region', 'earthquakesensitive', 'zone', 'country']
['according', 'seismic', 'zoning', 'map', 'india', 'divided', 'four', 'major', 'seismic', 'zones', 'ii', 'iii', 'iv', 'v']
['zone', 'v', 'includes', 'areas', 'himalayas', 'northeast', 'vulnerable', 'destructive', 'earthquakes']
['major', 'cities', 'like', 'delhi', 'srinagar', 'guwahati', 'fall', 'high', 'seismic', 'risk', 'zones']
['devastating', 'bhuj', 'earthquake', 'gujarat', 'killed', 'people']
['another', 'severe', 'earthquake', 'struck', 'kashmir', 'affecting', 'india', 'pakistan']
['nepal', 'earthquake', 'also', 'caused', 'significant', 'tremors', 'damage', 'northern', 'india']
['earthquakes', 'india', 'often', 'result', 'loss', 'lives', 'property', 'damage', 'disruption', 'infrastructure']
['poor', 'building', 'structures', 'rural', 'urban', 'areas', 'increase', 'vulnerability', 'earthquake', 'damage']
['indian', 'government', 'launched', 'programs', 'promote', 'earthquakeresistant', 'construction', 'practices']
['national', 'disaster', 'management', 'authority', 'ndma', 'issues', 'guidelines', 'earthquake', 'preparedness']
['early', 'warning', 'systems', 'community', 'awareness', 'campaigns', 'help', 'reduce', 'risks']
['scientists', 'continuously', 'study', 'seismic', 'activity', 'india', 'better', 'predict', 'future', 'earthquakes']
['preparedness', 'strict', 'building', 'codes', 'disaster', 'response', 'planning', 'essential', 'minimize', 'earthquake', 'impact', 'india']
['amazon', 'delivers', 'products', 'quickly', 'often', 'expected', 'date']
['packaging', 'quality', 'excellent', 'items', 'usually', 'arrive', 'perfect', 'condition']
['customer', 'service', 'supportive', 'responds', 'promptly', 'issues']
['sometimes', 'product', 'quality', 'match', 'description', 'disappointing']
['overall', 'amazon', 'reliable', 'online', 'shopping', 'wide', 'range', 'products']
['flipkart', 'often', 'provides', 'good', 'discounts', 'deals', 'festive', 'sales']
['delivery', 'fast', 'metro', 'cities', 'delayed', 'rural', 'areas']
['product', 'quality', 'generally', 'good', 'returns', 'replacements', 'take', 'time']
['mobile', 'app', 'userfriendly', 'makes', 'shopping', 'easy']
['flipkart', 'great', 'option', 'buying', 'electronics', 'fashion', 'reasonable', 'prices']
['seiko', 'watches', 'reliable', 'grand', 'seiko', 'line', 'feels', 'like', 'true', 'luxury', 'affordable', 'price']
['citizen', 'ecodrive', 'amazing', 'watch', 'runs', 'light', 'never', 'needs', 'battery', 'change']
['casio', 'gshock', 'super', 'tough', 'stylish', 'perfect', 'outdoor', 'activities']
```

TF-IDF Analysis:

1. The TF-IDF scores highlighted distinctive words in each document.
2. For earthquake-related texts, words like *earthquakes*, *seismic*, *zone*, *damage*, and *preparedness* appeared with higher scores.
3. In e-commerce reviews, terms such as *delivery*, *quality*, *reliable*, *discounts*, and *returns* were significant.
4. For watch brands, words like *Seiko*, *Casio*, *Citizen*, *luxury*, and *automatic* stood out.
5. Observation: TF-IDF successfully identified domain-specific important words in each text category.



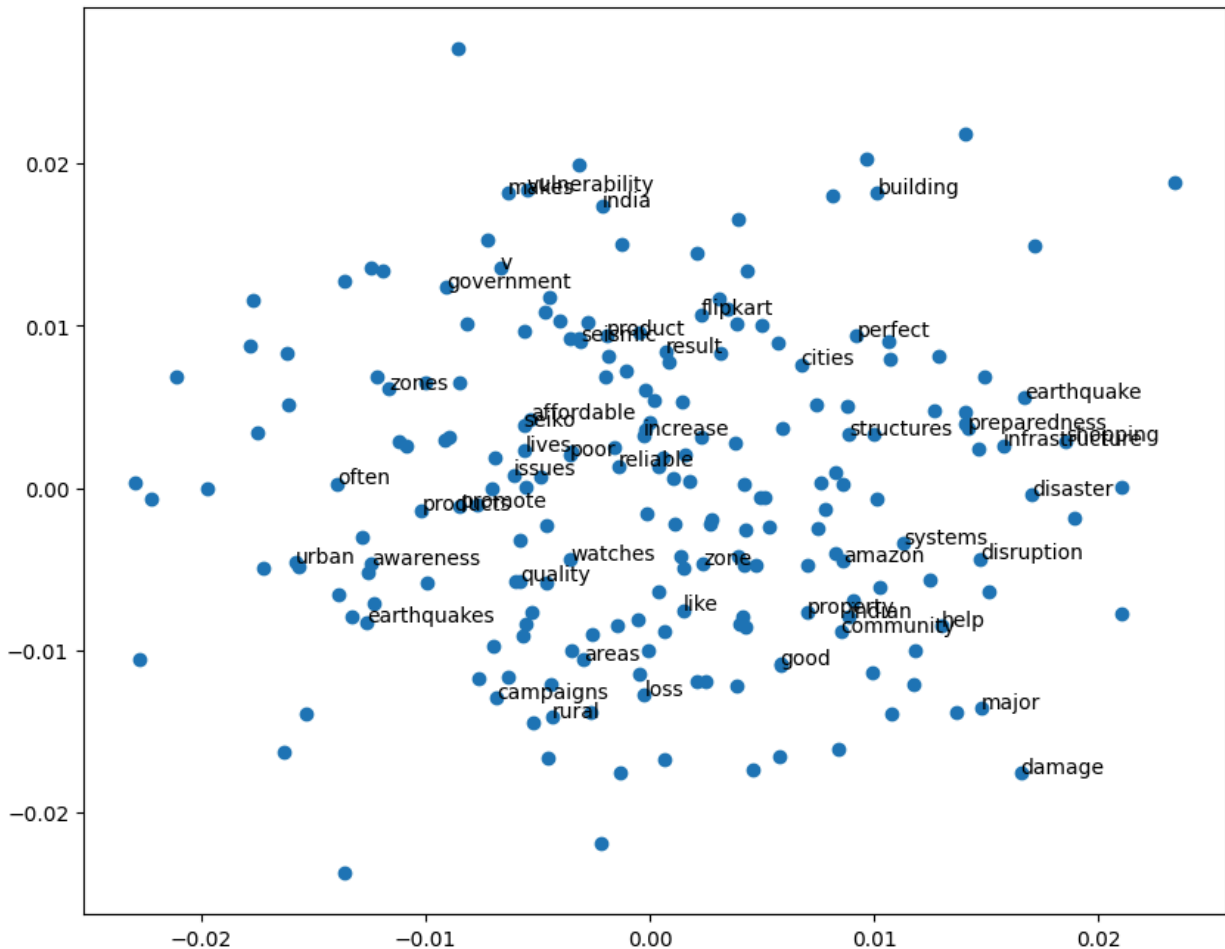
Word2Vec Embeddings:

The model captured related semantic relationships between words.

Example: The word *earthquakes* showed similarity with *seismic*, *tremors*, *damage*, *disaster*, and *preparedness*. And The word *Amazon* showed closeness to *shopping*, *delivery*, *products*, *reliability*, and *quality*.

PCA visualization revealed clusters of related words, where earthquake terms were grouped, and shopping-related terms formed another cluster.

Observation: Word2Vec effectively learned contextual similarities in the dataset.



Topic Modeling (LDA)

The LDA model extracted 4 main topics, each defined by its top keywords:

Topic 1: *earthquake, seismic, damage, India, preparedness* (Natural disasters)

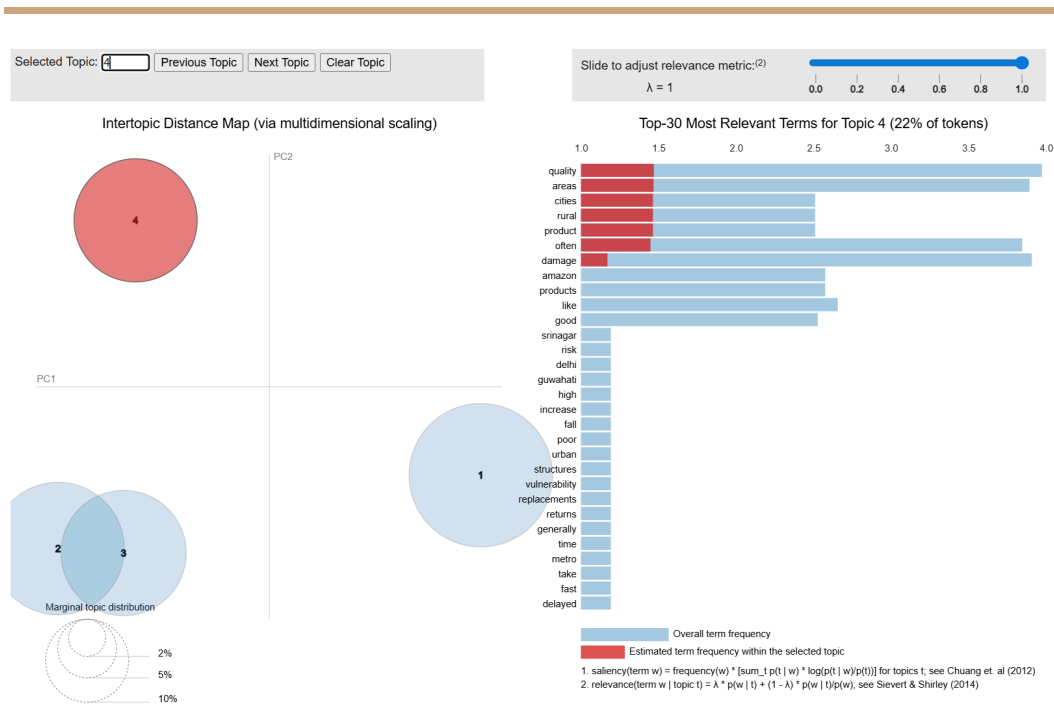
Topic 2: *Amazon, delivery, quality, shopping, products* (E-commerce: Amazon)

Topic 3: *Flipkart, discounts, returns, deals, app* (E-commerce: Flipkart)

Topic 4: *Seiko, Casio, Citizen, Orient, luxury* (Watches & brands)

Each document was assigned to the most relevant topic.

Observation: The model grouped documents into coherent themes that aligned well with the actual categories of data.



Conclusion: This assignment demonstrated how different NLP techniques can be applied to analyze diverse text data and extract meaningful insights. Through text preprocessing, the raw text was standardized and simplified for analysis. TF-IDF highlighted the most important words in each document, Word2Vec revealed semantic relationships between words, and LDA topic modeling uncovered hidden themes and grouped documents into coherent categories. Overall, the project shows how combining statistical, embedding-based, and probabilistic methods can provide a comprehensive understanding of textual information.