SECTION 12: Advanced NLP for Open-Ends

Cleaned Text Dataset

- Table with columns: Respondent_ID, Question_ID, Raw_Text, Cleaned_Text

https://docs.google.com/spreadsheets/d/1ejnuioXFQa2MBZ4P95fUspL6CNPP3AqmDqG_eoeqtSc/edit?gid=605107324#gid=605107324

2. **Embedding Files**
    - Vectors file (numpy or parquet)
    - Mapping file linking each vector to Respondent_ID + Question_ID

https://docs.google.com/spreadsheets/d/1ejnuioXFQa2MBZ4P95fUspL6CNPP3AqmDqG_eoeqtSc/edit?gid=2126872676#gid=2126872676

3. **Processing Notebook**
    - Documented code for cleaning, lemmatization, and embedding steps

https://colab.research.google.com/drive/18H6BxSEtdJiG19SZo6dnLsH2uL0OXTDE?usp=sharing

---

>>>Description

Scope: Prepare all open-ended survey responses for advanced NLP—cleaning, lemmatization, and embedding for use in topic modeling, clustering, and sentiment analysis.

🎯 Objectives

- Clean and normalize free-text from selected survey questions.
- Lemmatize all tokens to base forms using `spaCy`.
- Generate dense Sentence-BERT embeddings.
- Assemble a structured and metadata-tagged corpus for advanced NLP analysis.

🛠️ Analysis Tasks

Task	Details	Method
1. Text Extraction	– Concatenate responses from Q19–Q21, Q25–Q26, Q28–Q29, Q33, Q40–Q41 into a single DataFrame. – Retain columns: `Respondent_ID`, `Question_ID`, `Raw_Text`. – Export for downstream processing.	`python import pandas as pd # Sample load df = pd.read_csv("raw_data.csv") # Select and melt columns questions = ['Q19', 'Q20', 'Q21', 'Q25', 'Q26', 'Q28', 'Q29', 'Q33', 'Q40', 'Q41'] df_text = df.melt(id_vars=['Respondent_ID'], value_vars=questions, var_name='Question_ID', value_name='Raw_Text') df_text.dropna(subset=['Raw_Text'], inplace=True) df_text.to_csv("openends_raw.csv", index=False)`
2. Cleaning & Lemmatization	– Normalize text: lowercase, remove HTML, punctuation, whitespace. – Tokenize and lemmatize with `spaCy`. – Remove stopwords and non-informative tokens. – Document in notebook.	`python import spacy from spacy.lang.en.stop_words import STOP_WORDS import re nlp = spacy.load("en_core_web_sm") def clean_and_lemmatize(text): text = re.sub(r'<.*?>', '', text) text = re.sub(r'[^\w\s]', '', text) text = text.lower().strip() doc = nlp(text) tokens = [token.lemma_ for token in doc if token.lemma_ not in STOP_WORDS and token.is_alpha] return " ".join(tokens) df_text['Cleaned_Text'] = df_text['Raw_Text'].apply(clean_and_l`

		`emmatize) df_text.to_csv("openends_cleaned.csv", index=False)`
3. Embedding Generation	– Use pre-trained `SentenceTransformer` from `sentence-transformers`. – Encode each `Cleaned_Text` into a 768-dim vector. – Save as `.npy` and `.csv` for metadata mapping.	`python from sentence_transformers import SentenceTransformer import numpy as np model = SentenceTransformer('all-MiniLM-L6-v2') embeddings = model.encode(df_text['Cleaned_Text'].tolist(), show_progress_bar=True) np.save('embeddings_vectors.npy', embeddings) df_text[['Respondent_ID', 'Question_ID']].to_csv('embedding_mapping.csv', index=False)`
4. Corpus Assembly	– Join raw, cleaned, embeddings, and respondent metadata. – Validate alignment and row consistency. – Output final NLP-ready DataFrame.	`python df_meta = df[['Respondent_ID', 'Age', 'Gender', 'NCCS', 'Segment']] df_final = df_text.merge(df_meta, on='Respondent_ID', how='left') df_final.to_csv("nlp_ready_dataset.csv", index=False)`

📊 Deliverables

✅ Cleaned Text Dataset

`nlp_ready_dataset.csv`
Columns:

- `Respondent_ID`
- `Question_ID`
- `Raw_Text`
- `Cleaned_Text`

- `Age`, `Gender`, `NCCS`, `Segment`

✅ **Embedding Files**

- `embeddings_vectors.npy` – Dense vector matrix (768 dimensions)
- `embedding_mapping.csv` – Mapping file with `Respondent_ID` + `Question_ID`

✅ **Processing Notebook**

- Contains: cleaning logic, `spaCy` lemmatization pipeline, Sentence-BERT encoding
- Structured for reproducibility and sharing with collaborators

---

✅ **NLP Readiness Report**

- **Cleaning Summary**:
  - HTML stripped, stopwords removed, lemmatization applied
  - Average tokens after cleaning: e.g., 12.4
- **Vocabulary Size After Cleaning**: e.g., 3,452 unique terms
- **Embedding Diagnostic**:
  - Vector length: 768
  - Sample: [0.021, -0.004, …]

✅ **Next-Step Recommendations**

Technique	Tool	Input
Topic Modeling	BERTopic	Cleaned_Text + Embeddings

Clustering	HDBSCAN or KMeans	Embedding Vectors
Sentiment Analysis	VADER/TextBlob/Transformer-based	Cleaned_Text
Emotion Tagging	NRC or DistilBERT fine-tuned	Cleaned_Text