Methodology

This project employed a text analysis approach to assess the degree of thematic alignment between selected sessions of the **Global Solutions Summit (GSS) 2025** and the **T20 Communiqué**. Specifically, we compared the content of action points derived from four sustainability-focused GSS sessions with policy recommendations articulated in the T20 document. The analysis proceeded through a multi-step methodology grounded in natural language processing (NLP) and semantic similarity techniques.

1. Document Collection and Preparation

Two primary source documents were analyzed: (i) a summary of action points from the GSS 2025, and (ii) the final version of the T20 Communiqué. Both files were provided in .docx format and parsed using the python-docx library to extract the full text.

2. Text Segmentation

The extracted documents were segmented into analytically relevant units:

- From the GSS document, **action points** were extracted using a regular expression pattern that identified items marked with checkboxes ([ ]).
- From the T20 Communiqué, **recommendation blocks** were extracted using a pattern based on numerical section headings (e.g., 1.1.), capturing complete recommendation texts.

Both sets of texts were further segmented at the sentence level using the sent_tokenize function from the Natural Language Toolkit (NLTK), resulting in a corpus of individual sentences for fine-grained analysis.

3. Text Preprocessing

Each sentence was preprocessed to standardize and normalize the content:

- All characters were converted to lowercase.
- Punctuation, special characters, and line breaks were removed using regular expressions.

4. Sentence Embedding

To generate dense semantic representations, each preprocessed sentence was embedded using the **Sentence-BERT** model (all-MiniLM-L6-v2) implemented via the sentence-transformers library. This model is optimized for capturing sentence-level meaning and enables effective comparison of short texts in vector space.

5. Semantic Similarity Computation

We computed **pairwise cosine similarity** scores between all action sentences and recommendation sentences using the cosine_similarity function from scikit-learn. This produced a similarity matrix in which each cell represents the semantic proximity between an action point sentence and a T20 recommendation sentence.

6. Sentence Matching and Scoring

For each action sentence, the most semantically similar recommendation sentence was identified based on the highest cosine similarity score. The resulting matches were compiled into a structured table that included: the original action sentence, the corresponding recommendation sentence, their respective document indices, and the computed similarity score (rounded to three decimal places).

8. Output and Visualization

The final dataset was organized as a pandas DataFrame, sorted in descending order by similarity score. Results were exported in both Excel (.xlsx) and HTML formats to facilitate review, sharing, and further quantitative or qualitative analysis.

Tools

The analysis was conducted in Python, using the following libraries and tools:

- **Text processing**: nltk, re, spacy, docx
- **Semantic modeling**: sentence-transformers (Sentence-BERT)
- **Similarity computation**: scikit-learn
- **Data handling and export**: pandas, numpy