Topic Segmentation & Thread Summarisation for Code-Mixed Conversations

Akira Nair, Aneesh Edara, Prednya Ramesh, Ricky Raup

Motivation:

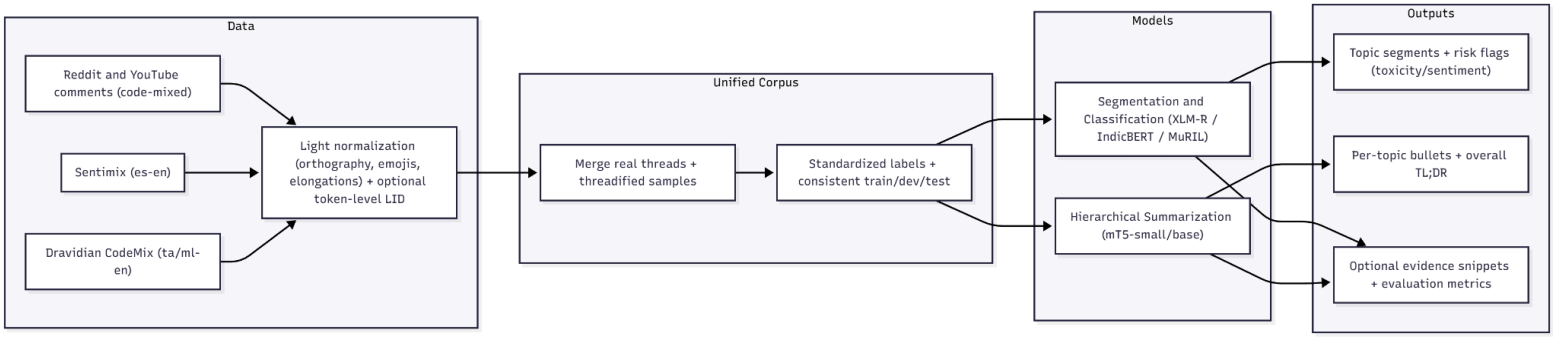
While large language models (LLMs) have developed the capacity to process and understand text in numerous spoken languages, it is unclear how well they perform in code-mixed and code-switched language use cases, where speakers alternate between two or more languages in a single paragraph or sentence (e.g. Spanglish (Spanish and English), Khichidi (Hindi and English)). Examples of code-mixed text are quite ubiquitous in online settings, particularly on multilingual social media platforms. We hope to fine-tune existing LLMs to better process code-mixed languages and evaluate our models' ability for topic segmentation and thread summarisation. By improving LLMs' understanding of code-mixed language, we strive to strengthen content moderation in cross-lingual settings and enable more equitable participation in online discourse for communities that routinely blend languages.

Plan:

We are going to start by pooling together multiple code-mixed datasets, including real Reddit, YouTube comment threads, Sentimix (Spanish-English sentiment), Dravidian CodeMix (Tamil/Malayalam English), and HASOC (Hindi-English hate speech). After we pool this data, we will proceed to create a unified benchmark corpus with standardised labels and a consistent train/test/dev split. We will then pre-process it so that our text data is normalised because some of these datasets have slightly messy or different variations of text formatting.

We will also do language token annotations, marking where switches happen between languages so that our models can pick up on this. Some models that we are interested in fine-tuning are mBERT, XLM-RoBERTa, IndicBERT, and MuRIL. These models are all multilingual transformers that have been pre-trained on text input from multiple languages. For example, MuRIL was pre-trained on 17 different Indian languages. We hope that the multilingual training of these models gives them some kind of advantage when it comes to using text with multiple languages mixed. The main tasks that we will approach are sentence-level tasks like sentiment analysis and hate speech detection, and we want to do some form of summarization as well. We are also interested in other tasks at the token level, like language identification and part-of-speech tagging, but we have not made a final decision yet on whether we will target many tasks broadly, or one/two tasks in order to maximise performance on those specifically.

Illustrative example:



Data in:
Code-mixed comment threads from Reddit and YouTube, plus small code-mixed datasets (Sentimix, Dravidian CodeMix).

Clean it up:
Light text cleanup (fix messy spelling/emoji/elongations). Optionally tag each token with its language to mark switch points.

Build one dataset:
Combine real threads with “threadified” samples and create standard train/dev/test splits by thread (to avoid leakage).

Train two things:

- **Segmentation & classification:** (XLM-R / IndicBERT / MuRIL): find topic shifts and detect sentiment/toxicity.
- **Hierarchical summarization**:(mT5-small/base): make short per-topic bullets and an overall TL;DR for the whole thread.

Outputs:

- A topic map of the conversation (where it changes subject).
- Per-topic bullets + an overall TL;DR.
- Risk flags (e.g., toxic subthreads) and optional evidence snippets.
- Evaluation scores so we know it’s working (ROUGE, Pk/WindowDiff, F1).