Evaluations

Methodology

Placeholder Preservation

The key feature is to have the placeholders in square brackets, e.g., `[placeholder]` untouched while the rest of the text is translated to the target language.

There can be 3 types of problems which have the potential to render the outputs unusable for localization purposes:

- the placeholder gets translated
- the placeholder gets sligthly modified, like `[dataStructure]` -> `[DataStructure]`
- totally new placeholder gets introduced

All of them can be identified by the following steps:

1. extract the placeholders via `regular expressions` from the original string
2. extract the placeholders via `regular expressions` from the translated string
3. compare the set of placeholders -> if they differ, one of the above issues has happened

Observations

1. `GPT-4o` and `GPT-4o-mini` sometimes introduces new placeholders. This behavior is inconsistent among languages. If one uses `app/prompts/transaltion_prompt_v1.md`, one might be able to observe it

Results

On the test set, the application managed to preserve all of the placeholders.

Comparing it to Google Translate (which did suprisingly well), we can say that it's remarkable (the dataset consists of 23 elements):

language	Wrong
Hungarian	7
Spanish	7
French	7
German	8
Japanese	6
Arabic	6
Hindi	8
Portugese	7

Term Specificity

It was challenging to come up with metrics for quality in a reference-free context.

I've implemented a `BM25` based quality metrics which aims to quantify how "special" the terms are in the original and in the translated statement:

1. The implemented `BM25` score is calculated for each term both in the original and the translated.
2. The average score for the statement is subtracted from both the original and the translated.
3. For each statement, the empirical cumulative distribution is calculated
4. The maximal distance of the empirical cumulative distribution is calculated. Ideally, it would be 0 because the empirical distributions would be the the same. If one of the distribution hits 1 before the other even takes of from 0 (it means that their distributions don't overlap, i.e., they are totally different) the distance would be 1
5. The "similarity" score for the translations are caalculated as `1 - maximal_distance`

Observations

1. We can't really expect scores close to `1`, because the languages have different structures
2. However, if significant uplift is observed, it indicates that improvement has been done on the specificity of terms used in the translations

Results

Uplift has been observed compared to the Google Translate version except the Hindi language:

- Google Translate:

language	bm25 similarity
Hungarian	0.539884
Spanish	0.487512
French	0.462427
German	0.469132
Japanese	0.458423
Arabic	0.520939
Hindi	0.515892
Portugese	0.534345

- `SafeTranslate` version:

language	bm25 similarity
Hungarian	0.553473
Spanish	0.508901
French	0.503316
German	0.485719
Japanese	0.490617
Arabic	0.532907
Hindi	0.45173

language	bm25 similarity
Portugese	0.543742

Semantic Proximity

In order to see how close the original and the translated meanings are, I've calculated the `cosine similarity` between the embeddings of the original and the translated statements using `OpenAI`'s `text-embedding-3-large` model.

Observations

Surprisingly, even though I feel that they are really close for the languages I speak, the `cosine similarity` was low. The issue might be that concepts in different languages might be represented in vastly different regions of the embedding space. Hence, it might not be a good indicator of translation quality.

Results

- Google Translate:

language	cosine similarity
Hungarian	0.339801
Spanish	0.266178
French	0.267522
German	0.245085
Japanese	0.274572
Arabic	0.354762
Hindi	0.470935
Portugese	0.291396

- `SafeTranslate` version:

language	cosine similarity
Hungarian	0.342938
Spanish	0.306353
French	0.260142
German	0.235501
Japanese	0.268565
Arabic	0.347016
Hindi	0.389017
Portugese	0.269616

Further Study

If I had more time, I would have researched text/speach quality metrics and could have implemented cople of LLM-based evaluation metrics which can be used on-the fly in production too for more reliable quality evaluation.

Conclusion

We can see that my application raises the level of the specificity of terms used in the translation alongside the fact that it achieves the goal of the application which is `placeholder preservation`, where Google Translate falls short