CS288 HW5 Writeup

Will Lavanakul

1 Custom Example Selection

My custom example selection utilizes the Levenshtein distance instead of cos similarity on vectors. Specifically, we directly measure the single character edit distance between the given utterance and training sample utterances, and take the lowest valued training samples.

This distance measure is effective since it has a good tradeoff between work similarity and utterance structure. If a prompt is very similar in words, but different in structure, it will still have a low edit distance. On the other hand, if the words are not as similar but still have a very similar structure (ie only one word needs to be changed) it will still have a low edit distance. This tradeoff provides samples with similar samples as well as similar words which should improve performance.

I achieved an accuracy of 0.575.

2 Model comparison

Below are the comparisons of each model for the utterance 'what river runs through m0'.

- Random selection: intersection (river , traverse_1 (m0)). Accuracy: 0.325.
- Cos similarity: answer (intersection (river , loc_1 (m0))). Accuracy: 0.625.
- Levenshtein distance: answer (intersection (river , through_2 (m0))). Accuracy: 0.575.

For random selection, the structure is already wrong as there is no 'answer'. It also contains 'traverse' as part of the parse. The cos similarity is much better as it has a correct parse. The Levenshtein distance is also quite good as the only error it has is including 'through' in the parse. This is due to the prompt containing 'through'. I think that cos similarity selects better examples as it leverages the embedding models knowledge, which has seen much more examples. The edit distance only relies on a predefined measure for strings which might not be as powerful as a language model that has actually seen language data. If there is no access to an embedding model, the edit distance can be a good substitute as the accuracy was not far off (-5%).