# Cross Encoder

1. Cross Encoder is a reranking model for content-based reranking
2. There is a quite collection of pre-trained cross encoders available on sentence transformers

To address the issue of the retrieval giving irrelevant aspects (like providing taste-related text for a smell query), you can adjust your code in the following ways:

1. **Improve Query Embedding**: Ensure that the query embedding captures the aspect-specific context better.
2. **Increase Certainty or Adjust Distance Metrics**: Use certainty or distance adjustments to get more relevant results.
3. **Post-Processing with Cross-Encoder**: Use a cross-encoder model for reranking the results based on relevance to the specific query.

The error message indicates that the client is receiving a 404 status code when trying to access the Weaviate instance. This usually means that the endpoint is incorrect or that the Weaviate server is not running on the specified URL.

Here are the steps to troubleshoot and resolve the issue:

1. Check Weaviate Server URL: Ensure that the Weaviate server URL is correct and that the server is running.
2. Verify Weaviate Endpoint: The endpoint used in the code might be incorrect. Make sure you are using the correct API endpoint for your query.
3. Weaviate Client Initialization: Make sure the Weaviate client is initialized correctly with the right URL.

It looks like your script runs but the scores from the cross-encoder model are unusually low. This might indicate a few things:

Query and Data Mismatch: The queries and data might not align well, leading to poor scores.

Model Issues: The cross-encoder model may not be working as expected.

Data Preprocessing: The data fed into the model might not be in the right format or might need additional preprocessing.

Debugging Steps

Verify the Query and Data Alignment:

Ensure that the queries and the data being retrieved are appropriate and relevant. You might want to print out some examples of the queries and the data to check their alignment.

Check the Cross-Encoder Model:

Verify if the cross-encoder model is loading correctly and functioning as expected. You might want to run some test queries to see if it returns reasonable scores.

Data Preprocessing:

Ensure the data being fed into the model is preprocessed correctly. This includes ensuring text encoding and formatting is appropriate.

The scores generated by models like the cross-encoder/ms-marco-MiniLM-L-12-v2 in the transformers library typically range from -1 to 1 for similarity tasks. The exact range can vary based on the model and task, but generally:

High Positive Scores (close to 1): Indicate strong relevance or similarity.

Scores around 0: Indicate neutrality or moderate relevance.

Negative Scores (close to -1): Indicate low relevance or dissimilarity.

For your reranking task, scores closer to 1 would indicate a high relevance between the query and the text being evaluated, while scores closer to -1 would indicate low relevance. Given that your scores are between -10 and -11, which are outside the typical range, it suggests that there might be an issue with how the model is being used or with the data being passed to it.

Potential Issues and Fixes

Model Output Range:

Verify if the cross-encoder model indeed provides scores in the range of -10 to -11, which would be unusual. This might be due to a specific setup or custom loss function during fine-tuning if applicable.

Data Encoding and Input:

Ensure that the data being input to the model is properly preprocessed. The CrossEncoder model expects pairs of texts in a specific format.

Model Usage:

Double-check that the model is being used correctly in the context of your script.

### Tips for Increasing Scores

* **Increase** certainty: You can experiment with different values for the certainty parameter to filter out less relevant results. However, be cautious as setting it too high might filter out too many results.
* **Optimize Queries**: Ensure your queries are specific and relevant to the aspects you are evaluating.
* **Data Quality**: Clean and preprocess your data to remove noise and irrelevant information.
* **Fine-Tune Model**: Fine-tune the CrossEncoder model on a dataset that closely matches your use case for better performance.

CONCLUSION

If you're still getting low scores after adjusting parameters and refining your model, here are some additional steps you can take to improve the situation:

1. \*\*Data Quality Check\*\*: Ensure that your data is of high quality and relevance to the task at hand. Low-quality or noisy data can negatively impact model performance.

2. \*\*Fine-Tuning\*\*: Consider fine-tuning your model on a domain-specific dataset that closely resembles your task. Fine-tuning allows the model to learn task-specific patterns and can lead to better performance.

3. \*\*Experiment with Different Models\*\*: Try experimenting with different pre-trained models or architectures to see if they perform better for your specific task. Not all models are created equal, and some may perform better than others depending on the task at hand.

4. \*\*Hyperparameter Tuning\*\*: Experiment with different hyperparameters such as learning rate, batch size, and optimization algorithms to see if they have an impact on model performance.

5. \*\*Ensemble Methods\*\*: Consider using ensemble methods to combine predictions from multiple models. Ensemble methods often lead to better performance by leveraging the strengths of different models.

6. \*\*Error Analysis\*\*: Conduct a thorough error analysis to understand why your model is making low-quality predictions. This can help identify areas for improvement and guide your efforts in refining the model.

7. \*\*Feedback Loop\*\*: Continuously collect feedback from your system's users and use it to improve the model over time. This iterative process can lead to gradual improvements in model performance.

By systematically exploring these options and iteratively refining your model, you should be able to improve the quality of predictions and increase the relevance scores you are observing.