**Introduction**

In today's digital world, users often interact with search engines, chatbots, and voice assistants using very short or unclear queries. These ambiguous questions are hard for machines to fully understand without asking for more context.

For example, if a user types:

"Obama family tree"

The system must figure out:

Do they want to see a family chart?

Are they asking about ancestry?

Or are they interested in relatives' biographies?

Instead of returning incorrect or incomplete results, the system should ideally reply with a clarifying question, such as:

"What aspect of the Obama family tree are you asking about — lineage, history, or public roles?"

This project focuses on training a Sentence-BERT (SBERT) model to automatically learn this behaviour by fine-tuning it on a dataset of real-world ambiguous queries and their matching clarifying questions.

**Dataset Description**

The training data is a JSON file containing pairs of:

User Queries: which are often vague or ambiguous.

Clarifying Questions: written by humans to resolve the ambiguity.

Example from the dataset:

Clarifying Question: "What aspect of the Obama family tree are you asking about? “This simple but highly relevant dataset helps teach the model to suggest helpful follow-up questions when the user's intent isn’t clear.

**Modeling Approach**

The core model is all-MiniLM-L6-v2, a variant of Sentence-BERT (SBERT) — known for its efficiency in encoding sentences into vector representations that can be compared using cosine similarity.

Training Objective:

To fine-tune this model, the training uses a MultipleNegativesRankingLoss function, which encourages:

Correct query-clarification pairs to be close in embedding space.

Incorrect combinations to be far apart.

This mirrors how humans associate questions and answers: the right follow-up feels contextually close to the original question.

**Implementation Overview**

Data Preparation

Loaded JSON data.

Extracted clean query-clarification pairs.

Saved it into a CSV file for easy loading.

Model Training

Used the Sentence-BERT all-MiniLM-L6-v2 model.

Created input examples using InputExample from the sentence-transformers library.

Loaded training samples into a PyTorch DataLoader.

Fine-tuned using MultipleNegativesRankingLoss for 4 epochs.

A small warmup of 100 steps was added to stabilize the learning rate.

Evaluation

After training, the model was tested using cosine similarity.

A pair of query and clarification was encoded into vectors.

Higher similarity scores indicated that the model correctly learned the semantic relationship.

Example:

The trained model assigned a high similarity score, meaning the model successfully linked the ambiguous query to its clarifying question.

**Results**

The fine-tuned model showed strong performance in identifying which clarifying question matches an ambiguous query, based on its similarity scoring.

This confirms that:

The model learned to distinguish context-specific follow-up questions.

It can generalize to unseen ambiguous queries.

It could be a powerful addition to chatbots, virtual assistants, and intelligent search systems.

**Conclusion**

This project demonstrates how fine-tuning Sentence-BERT with meaningful datasets allows a model to handle ambiguous queries more intelligently. Instead of misinterpreting vague input, the model suggests clarifications — just like a helpful human conversation partner would.

This ability is critical for:

Conversational AI systems.

Search engines.

Voice assistants like Alexa, Siri, and Google Assistant.

Fine-tuning SBERT using MultipleNegativesRankingLoss proves to be an efficient and scalable approach to improving the model's understanding of user intent in real-world applications.

**Future Work**

Dataset Expansion: Adding more diverse and multi-domain query-clarification pairs.

Multilingual Support: Extending the approach to non-English datasets.

Real-time Integration: Deploying the model into a chatbot pipeline for live ambiguity resolution.

**Summary**

This project bridges the gap between how humans communicate and how machines interpret language, helping conversational systems ask the right questions when faced with uncertainty — making human-computer interaction smoother, smarter, and more natural.