Problem: How you evaluate the points mentioned by the participant

We planned to provide two kinds of category base interview questions.

1. LLM base AI generate question
2. Predefined questions

LLM is responsible for generating category-based interview questions on time. The main challenge is how to validate the candidate given answers with the system expect answers.

So, I planned to evaluate the accuracy of the answers by creating a **Scoring System** (0 – 1). For this system we must divide the answers into 6 sections,

1. Key word matching

    - **NLP models** can identify keywords or concepts relevant to each predefined question and determine if the answer addresses the expected topics.
    - **Sentiment analysis** can also help evaluate if the candidate's tone aligns with what would be ideal for the question (e.g., confidence in technical answers, empathy in customer service scenarios).
    - **BERT or Similar Models for Contextual Understanding:** For open-ended questions, models like BERT (Bidirectional Encoder Representations from Transformers) can analyze how closely a candidate's response aligns with the expected context, allowing the platform to rate answers based on coherence and relevance.

2. Benchmark

    - **Predefined Benchmarks:** For each question, create a list of key points or benchmark answers. The system could assign scores based on how many of these points the candidate covers.
    - **Comparing with Sample Answers:** Using a model trained on a large dataset of correct answers (or industry-accepted responses) can help score the answers based on similarity.

3. Coding challenges

    - The system provides predefined and generated company standard coding challenges to do via system provide integrating online IDE
        - Open-source solutions like **CodeMirror**, **ACE Editor**, or **Monaco Editor** (used by Visual Studio Code) [To execute code, you would need a backend that supports running code in multiple languages (like **Judge0** or **Sphere Engine**).]
        - **Leverage Online IDE Platforms with Full Features** (platforms like **Repl.it** or **GitHub Codespaces** provide online IDEs with built-in terminals and debugging tools.)

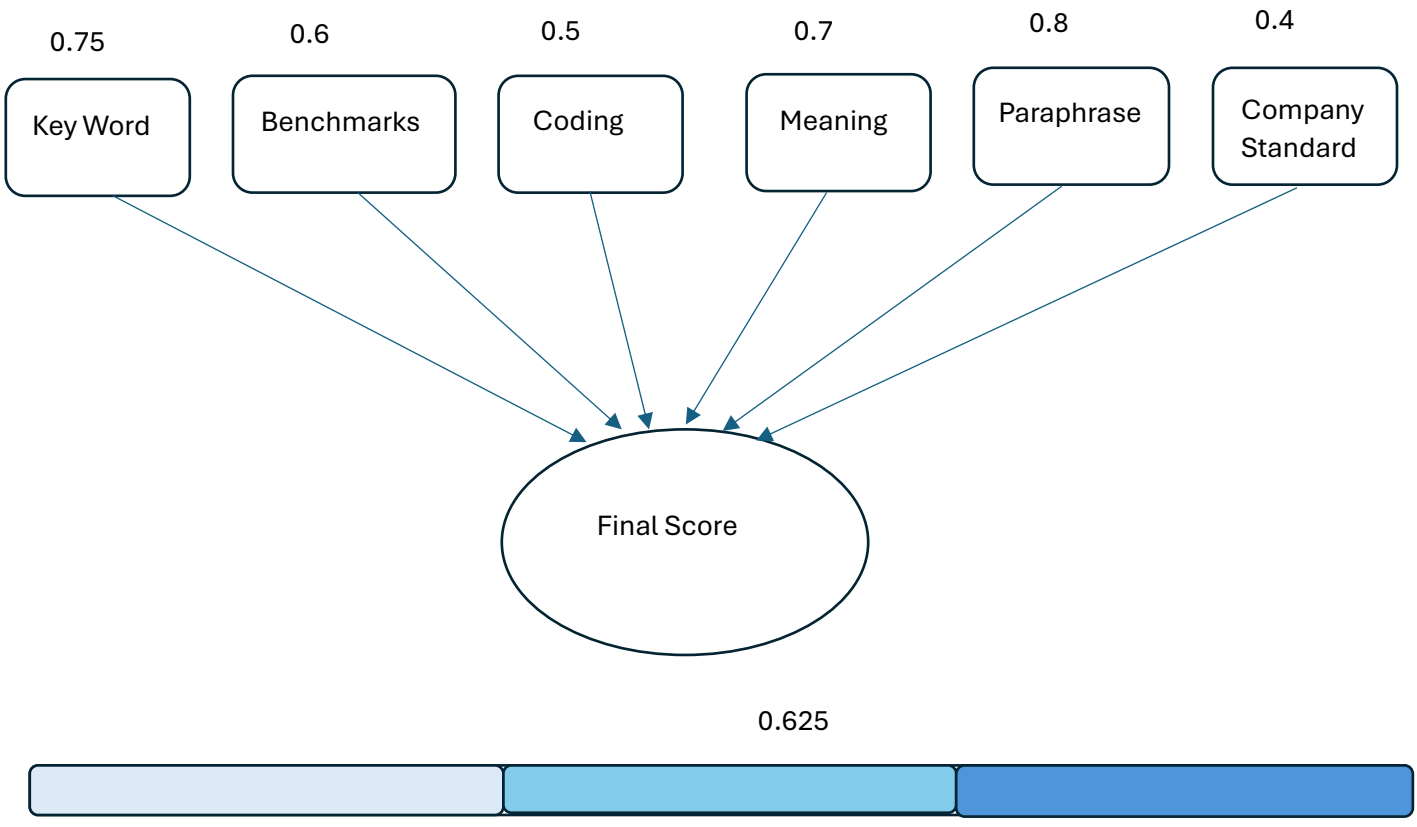
4. Meaning rather than words

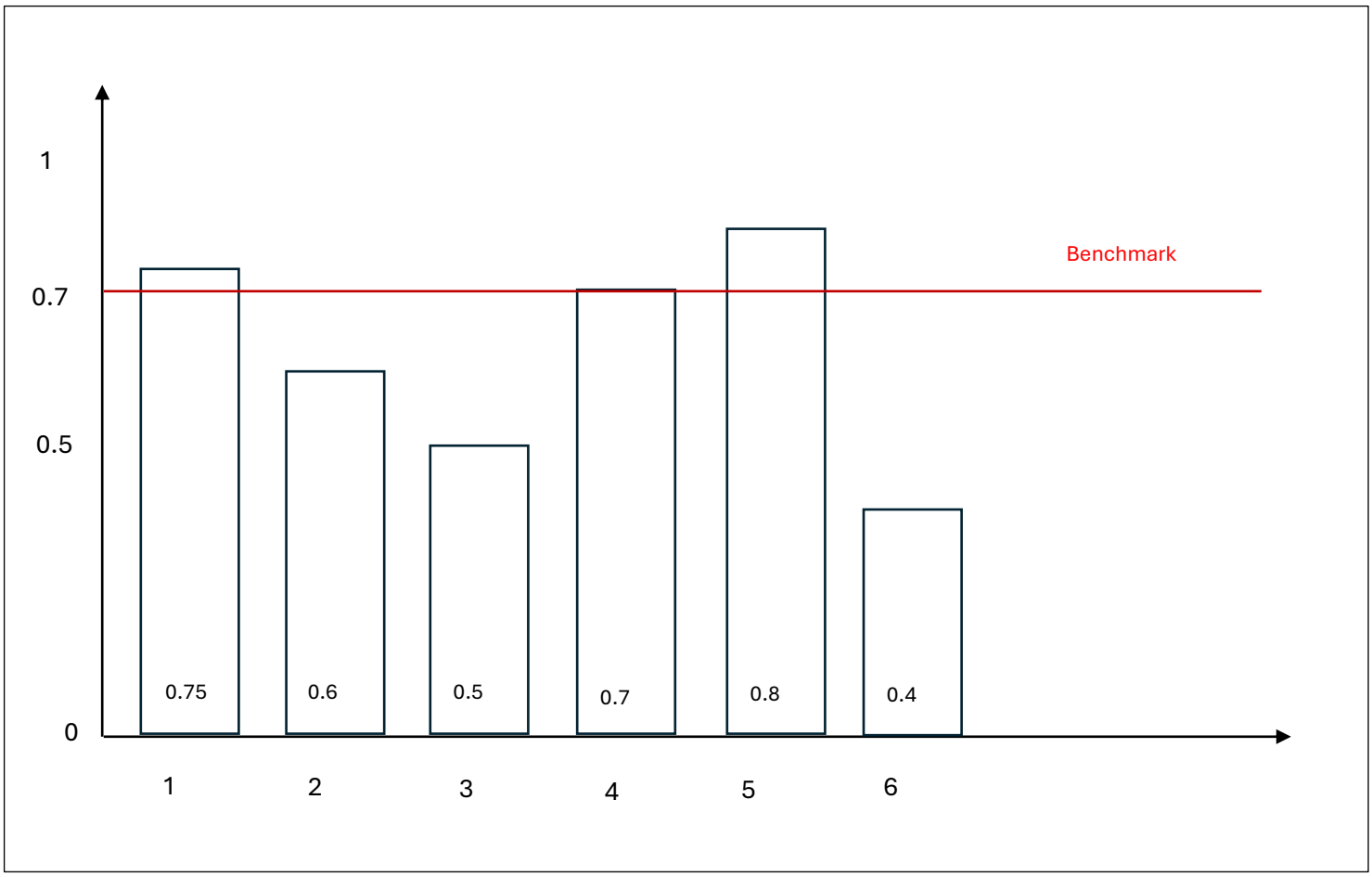
    - **Sentence Embeddings (e.g., Sentence-BERT, Universal Sentence Encoder):** These models can encode entire sentences into vectors that capture their meaning. By comparing the cosine similarity between a candidate's answer embedding and the embedding of an ideal answer, the platform can quantify how closely the meanings align, even with different wording.
    - **Threshold Setting for Similarity Scores:** Define a similarity threshold to determine if an answer is close enough to the ideal. For example, answers with a similarity score above 0.85 (on a 0 to 1 scale) could be considered "accurate," while lower scores indicate room for improvement or off-topic responses.

5. Paraphrase detection

    - **Fine-Tuned Paraphrase Models (e.g., Paraphrase-MiniLM, T5 Paraphrase Model):** Train or use pre-trained models that can detect paraphrases. These models can recognize if two sentences express the same concept despite different wording, providing a higher "paraphrase match" score to answers that, while different in words, match the essence of the ideal response.

6. Company standard
   In here we give scores to candidate consider their,
    - Problem-Solving and Analytical Skills
    - Communication Skills
    - Time Management




1
0.7
0.5
0
Benchmark
0.75
0.6
0.5
0.7
0.8
0.4
1
2
3
4
5
6