Table of Contents

1. Initial Model Selection & Training Methodology 2
2. Data Sources & Dataset Preparation 3
3. Metrics for Accuracy & Performance 4
4. Detecting & Handling Model Drift 5
5. Flowchart: 6

CALL TRANSCRIPTION SUMMARIZATION SERVICE.

The goal is to create and manage an AI service that automatically creates brief summaries from Arabic and English audio recordings of CS calls.

---

1. Initial Model Selection & Training Methodology

**Why fine-tuning instead of starting from scratch:

- We use a powerful "generalist" that has already been trained and make it more specific to call summaries (fine tuning it)
- It's easier to run and cheaper to maintain also it's faster.

1. **For English Transcripts:**
    - **Base Model**: To begin, we will begin with a model that has been developed particularly for the purpose of summarization, such as BART or PEGASUS. These models are already quite good at understanding and organizing texts in a clear way.
    - **Utilizing** Hugging Face Transformers will be our framework of choice. All of these pre-trained models and the tools that allow us to fine-tune them are easily accessible to us by this well-known library that is free to use.

2. **For Arabic Transcripts:**
    - Since Arabic is a grammatically rich language (meaning that words change form often), we need a model that is designed for it.
    - **Base Model**: We'll use a pre-trained Arabic model like AraBART (an Arabic version of BART) or a huge multilingual model like mT5 that knows several languages, including Arabic.
    - Again, we'll utilize Hugging Face Transformers, as it supports these models.

3. **Training Method (Fine-tuning):**
    - We gather a dataset of sample transcripts and human-written descriptions.
    - We display these samples to our selected pre-trained model.
    - The model modifies its internal parameters to grow better at providing summaries that look like our human-written examples.
    - We utilize GPUs since they're aimed at performing huge parallel calculations. Training and fine-tuning language models involve dividing very large matrix many times. GPUs have thousands of cores that can conduct these tasks at once, making training quicker and more efficient than CPUs.

---

2. Data Sources & Dataset Preparation

Teaching the AI with Examples

The most important part of this whole project is the data we use to teach the AI. If we give it clear and good data, it will learn to perform a good job.

1. **Data Sources:**

We need a lot of examples of a call transcript and its perfect summary. We'll get these from a few places:

- **Past Calls (This is the best option):**
  - ❖ The ideal situation is to use the company's old call transcripts. If human agents already wrote summaries for these calls, that's perfect.
  - ❖ Before we use any of this, we must use a tool to find and remove any personal customer information. This means names, phone numbers, addresses, and anything else that identifies a person. This is for keeping customer data safe.

- **Public Datasets:**
  - ❖ There are free collections of data online that can be used. For English, there are datasets full of summarized chat discussions.
  - ❖ For Arabic, we can find similar sets built for Arabic writing.
  - ❖ This helps the AI get a general idea of what summarization is before we focus it directly on our calls.

- **Generating our own data:**
  - ❖ We can generate data from GPT or any ai to generate similar data cases to our case

2. **Prepare and Clean the Data:**

- **Formatting with Pandas:**

We will use Pandas. We will write a simple script that gathers all the transcripts and their summaries, then arrange them properly into a table with two columns: one for the transcription and one for the summary. This clean table is then saved as a CSV or JSON file, which is the right shape for training the AI.

- **Cleaning Text with Python and TextBlob**

To clean up the messy texts, we will use another Python script. This tool will use a library like TextBlob to automatically clean the text. It will remove filler words like "um" and "ah," fix common writing mistakes.

- **Anonymizing Data with Presidio or Comprehend**

To protect personal information, we will use a specific tool for anonymization. We can choose Microsoft Presidio (a free, open-source tool) . These tools understand context and find private information like names, phone numbers etc. in both English and Arabic. Our script will send each recording to this tool, which will return a safe version where all private information is changed with placeholders like [NAME] or [PHONE_NUMBER].

- **Splitting Data with Scikit-Learn**

To divide our data for training and testing, we will use the train_test_split function from the Scikit-learn library in Python. To randomly shuffle our entire dataset and splits it into three parts: 80% for training the model, 10% for confirming its progress during training, and 10% for final testing.

---

3. Metrics for Accuracy & Performance

1. **Checking Summary Quality with ROUGE Scores**

What ROUGE is:

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It's just a set of automatic scores used to check how similar an AI summary is to a human-written summary. It's like:

- How many words match?
- How many short phrases match?
- How much of the sentence structure is the same?

The main ROUGE scores:

- ROUGE-1 → Looks at single words.

  Example: If the human summary says "package delivery confirmed" and the AI says "delivery confirmed," then "delivery" and "confirmed" count as overlaps.

- ROUGE-2 → Looks at pairs of words.

  Example: If both summaries include "package delivery" as a phrase, that's a match.

- ROUGE-L → Looks at the longest common sequence of words in order.

  Example: If both have the phrase "confirmed by the agent," it rewards that because the words appear in the same order. **(medium, n.d.)**

Why we use it:

- It's a fast, automatic way to get a score for summaries.

- Higher ROUGE scores mean the AI summary is closer in structure to the human reference.
- It’s not perfect (it doesn’t check meaning deeply), but it’s a good quick check before human reviewers.

2. **Evaluation**

The ROUGE score is helpful, but it's not perfect. The most important test is always human judgment. Every week, we need to review a random sample of the AI's summaries. we will score each one from 1 to 5 on three things:

- Does the summary read smoothly and make logical sense?
- Are all the facts correct? Did the AI make anything up or get something wrong?
- Is the summary to the point, or is it too long and repetitive?
  This human feedback is our best testing.

3. **Measuring Speed with Latency and Throughput**

A accurate summary is useless if it takes too long. We need to measure the speed of our service with two key numbers:

- **Latency:** This is the delay from the moment we send a transcription to the AI until the moment we get the summary back. For a good user experience, this needs to be under 5-10 seconds.
- **Throughput:** This tells us how many transcripts the AI can process at once. It's the number of reports finished per minute. This is crucial for knowing if our system can handle busy times or a quick surge in traffic without slowing down or crashing.

---

4. Detecting & Handling Model Drift

1. **What is Model Drift?**

Model drift is what happens when the AI's ability slowly gets worse over time. This isn't because the AI is not working good, but because the maybe the data is changing. For example, buyers might start asking about a new product, using new slang, or following new procedures that the AI wasn't originally taught on. The AI starts to see new types of talks it doesn't understand as well. **(IBM, n.d.)**

2. **How We Detect Drift**

We have two main ways to spot when the AI is starting to struggle:

- **Track Performance Scores:** We will continuously watch the AI's ROUGE scores and human evaluation rates. If we see these scores start to drop strongly and stay low, it's a big red flag that the AI is no longer working as well as it used to.
- **Monitor the Inputs:** We will also keep an eye on the incoming call records themselves. A sudden change in the length of calls or the topics being talked can be an early warning sign that the nature of the calls is changing and the AI might soon be in over its head.

3. **How We Handle Drift**

- Collect Changes Automatically: We will build a simple system where human workers can easily fix any bad reports the AI makes. These updated transcript-summary pairs are immediately saved into a special dataset.

- Schedule Regular Re-Training: Every 3 to 6 months, or quickly if we discover major drift, we will retrain or refine tune the AI. We take its original training data and mix in all the new corrected cases we've gathered.

5. Flowchart: