

expo QQA 24

MADRID
May 28th,
29th, 30th
2024

expoqa.com

A woman with long blonde hair is wearing a black headset and is seated at a desk in a call center. She is looking at a laptop screen and has her hands on the keyboard. The background is dark, suggesting an office environment at night. The text 'Ensuring Accuracy and Quality in Multilingual Call' is overlaid on the left side of the image in white, bold font.

Ensuring Accuracy and Quality in Multilingual Call

About us



Enrique Sánchez

QA Team Lead

(enrique.sanchez@aircall.io)

Enzo's Father
+10 years in QA
♥ Programming and Testing



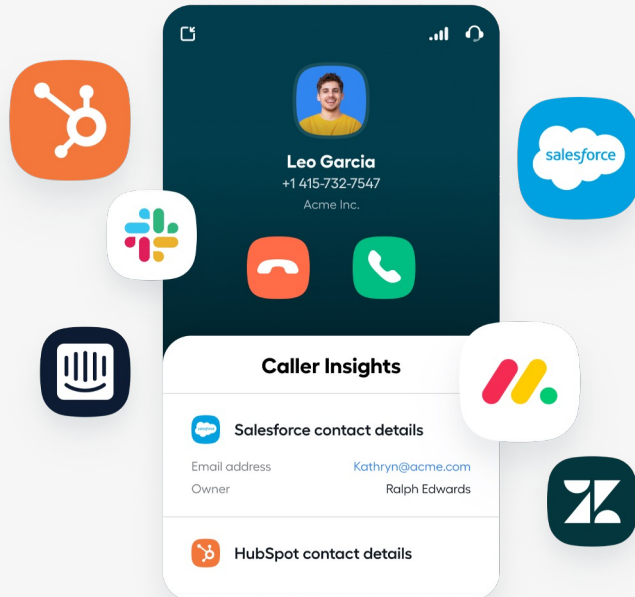
Kevin Perkins

QA AI Specialist

(kevin.perkins@aircall.io)

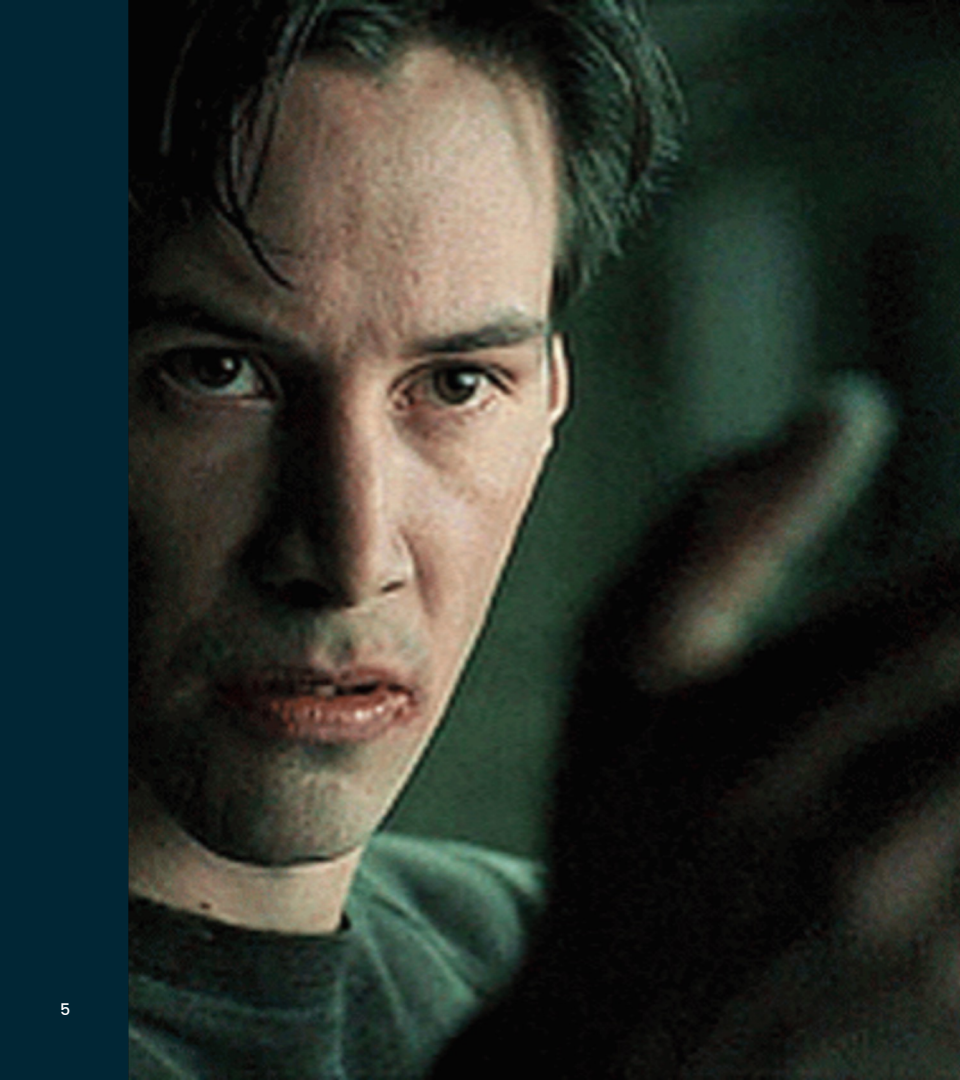
Teo's Father
5 years in QA
Linguist, Bookworm,
Skater, Jersey boy

About Aircall



What is Aircall

- Integrated & smart customer communication platform
- Cloud-based voice and text solution
- Easy to use and reliable
- Integrated with leading CRM and help desk tools:
 - Salesforce, HubSpot, Zendesk...



AI as plugin for our clients

Our Challenge



In our industry, digital transformation – including AI integration – is essential to enhance customer experience, improve team productivity and drive revenue growth. [...]

AI isn't about replacing the human element but enhancing it.

Alan Talanoa,
CTO of Aircall



What the team is working on?



Summarizes conversations

Be able to summarize a conversation in different languages in a few lines.



Identifies key topics

What are the real key topic for the conversation? Can I have a dictionary of topics?



Detects emotions

How the conversation was?
How was the feeling?

AI in Aircall

Conversation Center

The screenshot displays the Aircall Conversation Center interface. At the top, it shows a call log entry: "recibió una llamada de [redacted]". Below this, the interface is divided into several sections:

- Contexto de la llamada (1):** A section titled "Contexto de la llamada" with a sub-section "Resumen" containing a paragraph: "The purpose of the call was for the customer to share positive feedback about the application and express their satisfaction with its features, design, and customer service. They also mentioned their intention to upgrade their client." Below this is a "Ratio de conversación y escucha" chart showing 85% for the user and 30% for the customer.
- Temas clave (3):** A section titled "Temas clave" with a sub-section "Estado de ánimo del cliente" showing a "Positivo" mood. Below this are "Etiquetas" (tags) such as "positive feedback", "application features", "design", "customer service", and "client upgrade".
- Transcripción:** A section titled "Transcripción" showing a 1-minute transcript of the call. The transcript includes the following text: "Okay, so hello, I'm trying to make this call for a test. The idea is to have positive, negative, or neutral moves as the analysis. So yeah, I really like this application. It really helps me with my day-to-day work. It's really helpful and the team is very helpful too. I really have some very good and positive feedback on this application. And yeah, I'm very, very happy with the application overall. and I would like to upgrade my client and because everyone on the company on our side is really happy with the application, with its quality, with its design. The design is very intuitive and very user-friendly." A "Daros tu feedback" button is visible at the bottom right of the transcript.

At the bottom of the interface, there is a playback control bar with a 1.0x speed setting, a play button, and a progress bar showing 0:00 to 02:34.

1 Summarization

2 Emotion detection

3 Key Topics

Our QA Challenge



Accuracy

Ensure the clients receive an accurate response for Summarization, Key Topics...



Shift-Left

Be involve as soon as possible in the team.
Don't start when the model is done



Automation

Automate all the things!



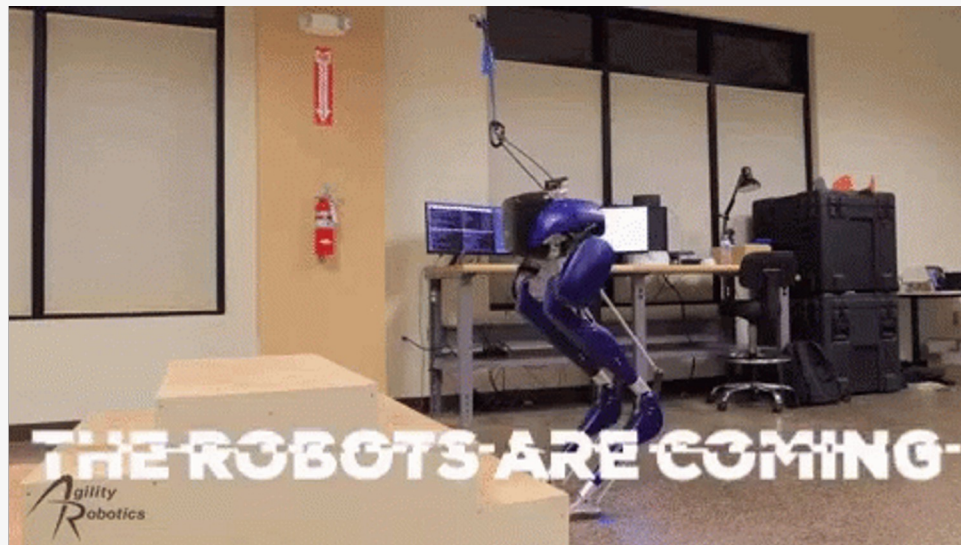
AI is hard to test but possible

Our Testing Approach

Our Testing Approach

Automation All the Way:

Integrated testing in our pipelines, because efficiency is key!



(and we are not afraid of AI)

Our Testing Approach



Prompt Engineering:

What? Crafting clever prompts to feed our AI.

Why? Helps our NLP engineers fine-tune and control the AI magic tricks.



Integration Testing:

What? We test the tools that bring our AI to life.

Why? So our services are as fast as Flash



Model Testing:

What? Rigorous check-ups on our AI's brainwork.

Why? Ensures the model isn't just smart—it's genius.



End-to-End (E2E) Testing:

What? The ultimate trial, from start to finish.

Why? To confirm what users see is spot-on—no surprises and as good as Tony Stark armor!

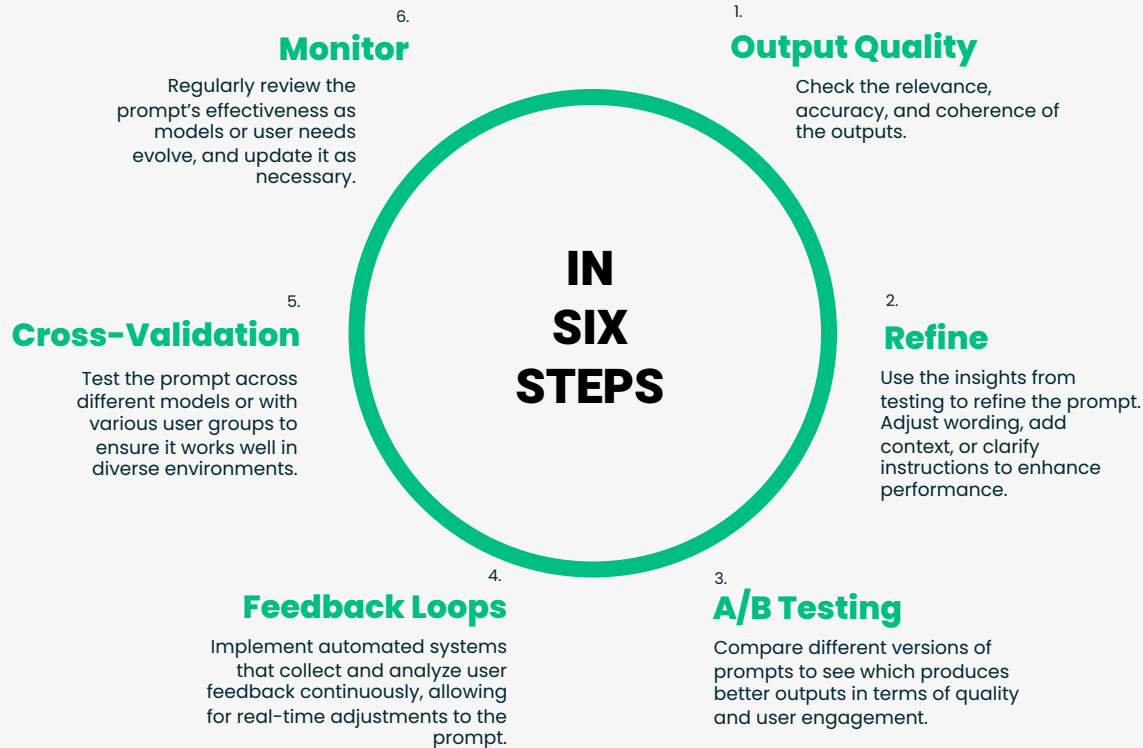


WHAT THE HECK IS GO

Fighting the AI magic

Prompt Testing

Prompt Testing



Example

Prompt Testing

1

Prompt file with multiple prompts to evaluate

2

Simple Test we use to validate what prompt is better under some circumstances

```
transcriptions.py 0.21 KB
Blame Edit Lock Replace Delete

1 prompt_lines_v1 = [
2     *
3     *
4     *
5     *
6     *
7     *
8     #
9     *
10    *
11    *
12    *
13    *
14    *
15    *
16    *
17    *
18    *
19    *
20    *
21    *
22    *
23    *
24    *
25    *
26    *
27    *
28    *
29    prompt_lines_v2 = [
30    *
31    *
32    *
33    *
34    *
35    *
36    *
37    *
38    *
39    *
40    *
41    *
42    *
43    *
44    *
45    *

4 import pytest
5
6 from prompts.criteria.criteria_prompts_summary_key_topic import (
7     criteria as key_topics_criteria,
8 )
9
10 from prompts.summary_key_topics.summary_key_topics import prompts
11 from qa_ai_tools.egvals.prompt_evaluation import prompt_evaluation_in_models
12
13 logger = logging.getLogger(__name__)
14
15 @pytest.mark.parametrize(
16     "model, model_evaluator, prompt_a, prompt_b, prompt_verifier, criteria, file_name",
17     [
18         (
19             "LLM_MODEL",
20             "LLM_MODEL_VERIFIER",
21             prompts["v1_en"],
22             prompts["v2_en"],
23             prompts["verifier"],
24             key_topics_criteria,
25             "conversations/positive_real_awesome.txt",
26         ),
27     ],
28 )
29 def test_evaluate_two_prompts(
30     model, model_evaluator, prompt_a, prompt_b, prompt_verifier, criteria, file_name
31 ):
32     """
33     Test the summary key topics prompt.
34
35     The way to evaluate is using a different model from the one used to generate the prompt.
36     The test evaluates the output of the prompt and check if it is relevant and accurate.
37     The model used will answer with a score of 1 if the output is relevant and accurate, and will add a reasoning for the score.
38
39     The test will fail if the prompt_b is better than the prompt_a
```



Best Detectives in the World
of AI

Model Testing

Model Testing

Output Format

Verifying that outputs adhere strictly to required **specifications**.

- How is the output? Does it fit with the prompt?
- Is using the right language?

Accuracy

Ensuring responses are not just correct, but **precisely** what's needed.

- Is the model understanding the prompt?
- Is the answer related to the prompt?

Relevance

Checking if responses **truly hit the mark**.

- Is the answer right?
- Hallucinations?



Model Testing (deeply)

Output Format

Execution of the model with a **simple request** and validate the output is aligned with the prompt: size, language...

Simple Tests Cases that has to be Green always

Tip: check what the system answer you, sometimes the models add a line at the beginning saying you something like "Here is the response:"

Accuracy

More **complex scenarios**. Fast to analyze by the model.

Analyze the body of the model response to make semantic analysis: the response has key words on it?

The **output should be relevant** to the prompt.

Tip: Do not make too complex scenarios, validate a simple scenario and use another model as evaluator (you can use more than one at the same time)

Relevance

Complex scenarios: mix between **real data and synthetic data**.

Use another model to make harder validations: hallucinations, precision, clarity... in the same way an human expert should do it

Tip: Be careful with the time execution (it can take hours! And cost a lot of money)

Example

Model Testing

1

Definition of the criteria of evaluation for a prompt.

```
criteria_models_evaluation.py 630 B Blame

1 criteria = {
2     "language":
3     "key-topics"
4     "key-topics-
5     "simplicity"
6     "clarity": "
7     "precision":
8     "truthfulness
9     "relevance":
10    "accuracy":
11 }

def test_evaluate_summary(model, model_verifier, criteria, summary_template, file_name):
    """
    Test the summary key topics prompt.

    The way to evaluate is using a different model from the one used to generate the prompt.
    The test evaluates the output of the prompt and check if it is relevant and accurate.
    The model used will answer with a score of 1 if the output is relevant and accurate, and will add a reasoning for the score.
    """

    evaluator = LabeledCriteriaEvalChain.from_llm(llm=model_verifier, criteria=criteria)
    result_relevance = evaluator.evaluate_strings(
        prediction=output, reference=prompt, input=text_to_replace
    )
    logger.info("result_relevance: %s", result_relevance)
    assert result_relevance["score"] == 1, result_relevance
```

2

Simple test case to evaluate the summary. We use Ollama library to help us with the validation.



We need to be fast

Integration Tests

Integration Testing

Models are not isolated

- Input and output from other services matters

Real World Scenarios

- How it works when all the services are active?
- How long does it take?
- What happens if it fails?



Integration Testing (deeply)

Models are not isolated

Our tests cases has to run multiple services and avoid mocking.

We need to validate the output of the model under different scenarios

Do not tests semantically, you only need to be sure the system is resilient

Tip: Make tests where you validate how the different parts of the model send/process data to/from the model, is the output well formatted?

Real World Scenarios

What are your users waiting when they use the scenario?

Tip: Use “real world data” for these tests

Example

Integration Testing

```
@pytest.mark.c3_voice_ai
def test_inbound_call_insights(
    bulk_creation_available_agent_w_twilio,
    external_call_number,
    wait_until_agent_ringing,
    wait_for_insights,
    payload_call_insights_with_calluuid,
    wait_until_call_recorded,
    audio_file,
    status,
    voice_ai_number3,
    record_property,
):
    """
    * Scenario:
    - Trigger an inbound call to an agent
    - The call is automatically recorded
    - Wait for the record to be into WEB

    * Acceptance criteria
    - Transcription related to this call is found in the web DB with the appropriate status
    - Insights are found for the transcription
    """
```

1

```
is_found, transcription, insights_found = wait_for_insights(
    call_uuid=phone.task_id,
    payload=payload_call_insights_with_calluuid(phone.task_id),
)

assert is_found
assert insights_found
assert transcription.get("type") == "CONFERENCE"
assert transcription.get("status") == status
assert transcription.get("insights", {}).get("summary", {}) is not None
assert transcription.get("insights", {}).get("topics", {}) is not None
```

2

1 Simple test in Python using Pytest where we use the services to make a phone call and get the insights information

2 Assertion for the final output for the integration test, in this test we only check the response has valid data



Time to validate all the
technology

E2E Testing

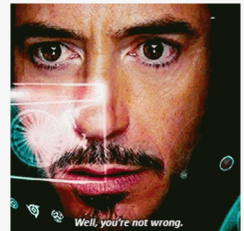
E2E Testing

UI Component Tests

- We test UI elements directly within our team's repository.
- Playwright/Cypress
- Ensure every screen element functions "beautifully" and as expected.
- Running before push to master

Functional Tests

- Conducting comprehensive functional tests across the application.
- Cypress
- To simulate real user scenarios and ensure all features work seamlessly from start to finish
- Running 24/7 in a CI pipeline for different environments



Example

E2E Testing

1 Cypress execution for an E2E test case. We check not only the screen elements, but the requests and the time execution.

2 Slack notification every time one test is executed and it fails

The image shows a composite screenshot illustrating an end-to-end testing workflow. On the left, the Cypress test runner interface is visible, displaying a list of test steps such as `(fetch) POST 202 https://rum-browser-intake-datadoghq.com/api/v2/rum/` and `(xhr) GET 200 https://internal.aircall-staging.com/v3/tags`. A `WindowOpen` command is also present. In the center, a browser window displays the 'Aircall_VoiceAI' application, specifically the 'Conversation Center' page. A search input field is highlighted with a green circle and the number '1'. On the right, the browser's network and console panels are visible, showing various HTTP requests and responses. In the foreground, a Slack notification is overlaid, showing a message from 'Pytest WORKFLOW' at 17:49. The message includes a warning icon, a link to a GitHub repository, and a status bar indicating '1 failed test(s) on run', '29 passed', and '0 skipped'. The notification also shows the channel name '# qa-voice-ai' and the user '@Aurélien Haye'.

AI is hard to test

Lessons Learned



Good

- Be integrated in the team from the beginning*
- Test from the beginning (Prompt generation) and be sure you test the integration of the model*
- Use open source tools*

Don't good

- Flakiness is a real problem. Models are not consistent and you will spend a lot of time with it*
- Prompting is more art than science*
- Integrate all the tools is complex*

Key Points

1

Comprehensive Testing

Utilizing tools like Ollama, Playwright, and Cypress to ensure our AI is accurate, functional, and user-friendly.

2

Integrated Team Approach

Close collaboration within teams for prompt design, data analysis, and testing speeds up error detection and solutions.

3

Holistic Strategy

Our all-encompassing approach provides a robust safety net, ensuring reliability and consistency in AI performance.

A German Shepherd dog is shown in profile, looking towards the left. The dog has a thick, tan coat with a black mask around its eyes and muzzle. The background is a cluttered indoor space, possibly a living room or kitchen, with various items like a green bottle, a wooden chair, and a red cloth hanging in the distance. The lighting is somewhat dim and indoor.

ANY QUESTIONS?

Thanks!



The background of the entire image is a photograph of a large audience seated in a hall, facing a stage. The scene is dimly lit with blue tones. On the stage, there are several large projection screens and a speaker. The audience is diverse in age and appearance, all focused on the front of the room. The overall atmosphere is professional and high-tech.

expo **QA** 24

MADRID
May 28th,
29th, 30th
2024

Thank you for attending

expoqa.com