

Improve RAG accuracy: integrate user context for better answers

Why the user question is not always good enough

The answer provided by a RAG solution like **Nuclia** will always depend on how precise the question is. The more information you provide, the more accurate the answer will be.

Unfortunately, the user might assume that the system has more context than it actually does. For example, imagine your company is running an online avatar creation studio and you provide the following details about the different subscription plans:

User plans

Our user plans are designed to help you get the most out of our avatar creation studio. We offer three plans: Free, Pro, and Enterprise. Each plan has its own set of features and benefits.

Free plan

Our Free plan is perfect for those who want to try out our avatar creation studio. With this plan, you can create up to three avatars and access a limited set of features:

- Choose from a selection of pre-designed avatars
- Customize the hair color, eye color, and clothing of your avatar
- Download your avatar as a PNG file

Pro plan

Our Pro plan is ideal for individuals who want to create more avatars and access additional features. With this plan, you can create up to 10 avatars and enjoy the following benefits:

- Import your own images to create custom avatars
- Access a wider range of customization options, including facial features and accessories
- Download your avatars as PNG or SVG files

Enterprise plan

Our Enterprise plan is designed for businesses and organizations that want to create a large number of avatars and access advanced features. With this plan, you can create an unlimited number of avatars and take advantage of the following features:

- Use the API to integrate our avatar creation studio with your website or application
- Access premium customization options, such as 3D avatars and animation

The user might ask “Can I import my own images to create an avatar?” without specifying “by the way, I am on the Free plan”. In this case, the system will not be able to provide an accurate answer. The RAG process will probably make a semantic match on the part of the document that mentions the image importation feature. So it might return the answer “Yes, you can import your own images to create custom avatars” without mentioning that this feature is only available on the Pro plan.

The reason this user is not indicating the plan is because the web interface is clearly showing the plan the user is on. The user will assume that the system has this information. It is a common issue when implementing a RAG-based solution: the LLM is very poorly linked to the rest of the data infrastructure, and at the contrary, the user has very high expectations on the system’s awareness (because it is a chatbot, it behaves a bit as a human).

How to solve this issue

To solve this issue, you need to set up a pipeline that will inject the user’s context into the RAG process.

The pipeline will have to:

- Call the API endpoint to get the user's profile.
- Turn it into a valuable context for the RAG process.
- Inject this context into the RAG process to answer the user's question.

It can be done entirely manually using the Nuclia API.

But the Nuclia JS/TS SDK provides a way to do this in a more straightforward way. Instead of writing code to call the API, you can use a more declarative way to define your pipeline steps.

Here is an example of how to do it:

- First, you need to instantiate the Nuclia API object:

```
import { Nuclia } from '@nuclia/core';
```

```
const nucliaApi = new Nuclia({
```

```
  zone: 'YOUR-ZONE',
```

```
  knowledgeBox: 'YOUR-KB-ID',
```

```
});
```

- Then, you can create the pipeline. The first step makes a call to an hypothetical `/user/profile` endpoint. Let's assume it returns the following JSON:

```
{
```

```
  "user": {
```

```
    "plan": "Free"
```

```
    "lastConnection": "2024-08-02"

  },

  "preferences": {

    "language": "en",

    "theme": "dark"

  }

}
```

The step to handle this would look like this:

```
const pipeline =
nucliaApi.knowledgeBox.createAgenticRAGPipeline({

  START: {

    action: {

      type: 'api',

      url: '/user/profile',

      method: 'GET',

      outputs: {

        profile: 'user',

      },

    },

  },

}
```

```
},  
  
  next: [{ stepId: 'translate' }],  
  
},
```

The next field allows you to define the next step in the pipeline. In this case, the next step is `translate`.

- The second step will translate the user profile into a natural language description. It uses a `predict` action to do this. The `query` field is a template string that uses the profile output from the previous step. Most of the steps parameters accept templates string, allowing to pass results from one step to another.

```
translate: {  
  
  action: {  
  
    type: 'predict',  
  
    query: 'The following JSON string describe a user profile.  
Translate in plain english what this JSON string contains. Do not  
mention the JSON string, just provide the user profile in plain  
english speaking in the first person singular as if you were this  
user.\nJSON string: {{translate.profile}}',  
  
    outputs: {  
  
      user_description: {  
  
        type: 'string',  
  
        description: 'The user profile description in natural  
language',
```

```

    },

    },

    },

    next: [{ stepId: 'answer' }],

  },

  • The last step is the one that will call the RAG process. It uses
    the ask action to do this. The user description obtained in the
    previous step is passed as an extra_context parameter.
  answer: {

    action: {

      type: 'ask',

      query: '{{query}}',

      options: {

        extra_context: ['{{START.user_description}}'],

      },

    },

  },

},

});

```

- Finally, you can run the pipeline with a query:

```
const query = 'Can I import my own images to create an avatar?';
```

```
pipeline.run({ query }).subscribe(() => {  
  
    console.log(pipeline.context.answer.results.text);  
  
});
```

And the answer will be:

No, you cannot use external images to create your avatar on the Free plan. This feature is available only with the Pro plan.

Great! You have now a pipeline that can provide user-specific answers.

More complex use cases

In this example, we used a simple user profile to illustrate how to inject user context into the RAG process. In a real-world scenario, you could imagine a lot of different steps to get the user context. For example, you could:

- Use different sources to get the user profile (a CRM, a database, etc.).
- Retrieve previous interactions with the user.
- Use the user's location to provide location-specific answers.
- etc.

Try yourself by creating you own Nuclia account [here](#).