

# Amélioration du RAG avec Azure Document Intelligence et le format Markdown

## Introduction

Dans la [partie précédente](#), nous avons constaté une limitation majeure lors de la mise en œuvre du RAG en effectuant un split direct du PDF : les tableaux ne sont pas pris en compte (voir captures d'écran ci-dessous). Par conséquent, il est impossible pour le modèle de répondre aux questions qui s'y réfèrent.

### Cost Comparison

Contoso Electronics deducts the employee's portion of the healthcare cost from each paycheck. This means that the cost of the health insurance will be spread out over the course of the year, rather than being paid in one lump sum. The employee's portion of the cost will be calculated based on the selected health plan and the number of people covered by the insurance. The table below shows a cost comparison between the different health plans offered by Contoso Electronics:

	Employee's cost per paycheck	
	Northwind Standard	Northwind Health Plus
Employee Only	\$45.00	\$55.00
Employee +1	\$65.00	\$71.00
Employee +2 or more	\$78.00	\$89.00

```
await AskQuestionAsync("What is the employee's cost per paycheck?");
```

Question: What is the employee's cost per paycheck?

Answer: INFO NOT FOUND

Sources:

- employee\_handbook.pdf - default/employee\_handbook.pdf/5cdbb1aa86084ce1b3054b87832d4451 [Friday, April 5, 2024]
- Benefit\_Options.pdf - default/Benefit\_Options.pdf/08fd84ff01624f4d84e9a6715a1b25e0 [Friday, April 5, 2024]
- role\_library.pdf - default/role\_library.pdf/d8f22330d164409b8b00e1972f8fe7d4 [Friday, April 5, 2024]

### 1. Solution : Azure Document Intelligence

Pour résoudre ce problème, nous avons introduit [Azure Document Intelligence](#), un service d'intelligence artificielle qui effectue l'OCR des documents PDF et extrait le contenu sous forme de texte structuré. Ce service permet non seulement d'extraire les tableaux, mais aussi de conserver la structure et la sémantique du document d'origine.

L'laC a été mis à jour pour inclure la ressource Azure Document Intelligence:

- nouveau module : [document-intelligence.bicep](#)
- mise à jour du [main.bicep](#) pour inclure le module.

## 2. Conversion en format Markdown

Azure Document Intelligence convertit le contenu des PDF en format Markdown, un langage qui permet de structurer le texte et d'ajouter de la sémantique. Cette conversion offre plusieurs avantages :

- Prise en compte des tableaux :** Les tableaux sont correctement extraits et convertis en format Markdown, ce qui permet au modèle RAG de les traiter et d'y répondre.
- Sémantique améliorée :** La structure du Markdown fournit une sémantique supplémentaire au modèle de langage (LLM), facilitant ainsi la compréhension du contexte et l'amélioration de la pertinence des réponses.
- Compatibilité :** Le format Markdown est largement pris en charge par les outils et les plateformes de traitement de texte, ce qui facilite son intégration dans les pipelines de traitement du langage naturel.

Azure Document Intelligence dispose d'un studio en ligne qui permet de visualiser et de convertir le contenu des documents PDF en Markdown. C'est très pratique pour visualiser le résultat de l'OCR et vérifier la qualité de la conversion.

The screenshot shows the Azure Document Intelligence studio interface. On the left, there's a sidebar with options to 'Drag & drop file here or Browse for files or Fetch from URL'. A file named 'Benefit\_Options.pdf' is uploaded. The central area displays the document content, including a 'Cost Comparison' section with a table. The right-hand pane shows the converted Markdown output, which includes the same table structure. The interface also has tabs for 'Markdown', 'Text', 'Selection marks', 'Tables', and 'Figures'.

Employee's cost per paycheck		
	Northwind Standard	Northwind Health Plus
Employee Only	\$45.00	\$55.00
Employee +1	\$65.00	\$71.00
Employee +2 or more	\$78.00	\$89.00

## 3. Intégration du Markdown dans le RAG

Pour intégrer le Markdown dans le RAG, nous avons adapté le pipeline d'indexation des documents. Dans un premier temps, on effectue l'OCR des documents PDF avec Azure Document Intelligence pour obtenir le contenu en Markdown. Ensuite, on indexe ce contenu dans la base vectorielle pour permettre au modèle RAG de le récupérer et de générer des réponses.

Le code complet de l'intégration du Markdown dans le RAG est disponible dans le fichier [1-getting-started-with-ocr.ipynb](#).

Voici la partie du code C# qui illustre cette intégration :

- Initialisation du client Azure Document Intelligence

```
var docIntelEndpoint = env["AZURE_DOCUMENT_INTELLIGENCE_ENDPOINT"];
var credential = new DefaultAzureCredential();
```

```
var docIntelClient = new DocumentIntelligenceClient(new Uri(docIntelEndpoint),
credential);
```

- Conversion du contenu des documents PDF en Markdown

```
string folderPath = "../data";
string[] pdfFiles = Directory.GetFiles(folderPath, "*.pdf",
SearchOption.AllDirectories);

foreach (var pdfFile in pdfFiles)
{
    var markdownFilePath = $"{pdfFile}.md";
    if (File.Exists(markdownFilePath))
    {
        Console.WriteLine($"Skipping {pdfFile} because it already has a markdown
file");
        return;
    }

    try
    {
        using var fileStream = File.OpenRead(pdfFile);
        var binaryData = BinaryData.FromStream(fileStream);
        var analyzeRequest = new AnalyzeDocumentContent
        {
            Base64Source = binaryData
        };
        var result = await docIntelClient.AnalyzeDocumentAsync(waitUntil:
WaitUntil.Completed, "prebuilt-layout", analyzeRequest: analyzeRequest,
outputContentFormat: ContentFormat.Markdown);
        var markdownContent = result.Value.Content;
        await File.WriteAllTextAsync(markdownFilePath, markdownContent);
        Console.WriteLine($"Created: {markdownFilePath}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Erreur lors du traitement OCR de {pdfFile}:
{ex.Message}");
    }
}
```

- Indexation du contenu Markdown dans la base vectorielle

```
string[] markdownFiles = Directory.GetFiles(folderPath, "*.md",
SearchOption.AllDirectories);
foreach (string filePath in markdownFiles)
{
    string fileName = Path.GetFileName(filePath);
    string fullPath = Path.GetFullPath(filePath);
```

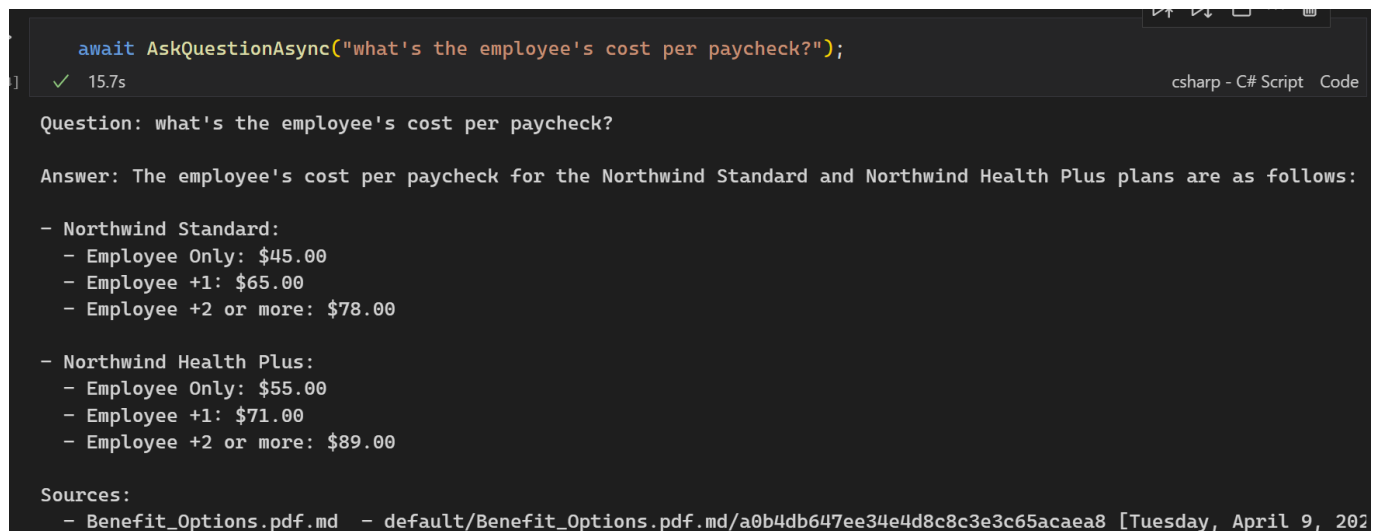
```
await memory.ImportDocumentAsync(fullPath, documentId: fileName);

Console.WriteLine("Successfully imported File Name: " + fileName);
}
```

#### 4. Résultats et conclusions

---

Après avoir intégré Azure Document Intelligence et le format Markdown dans le pipeline du RAG, nous avons constaté une nette amélioration de la performance du modèle. Les tableaux sont correctement pris en compte et le modèle est capable de répondre aux questions qui s'y réfèrent.



```
await AskQuestionAsync("what's the employee's cost per paycheck?");
```

✓ 15.7s      csharp - C# Script   Code

Question: what's the employee's cost per paycheck?

Answer: The employee's cost per paycheck for the Northwind Standard and Northwind Health Plus plans are as follows:

- Northwind Standard:
  - Employee Only: \$45.00
  - Employee +1: \$65.00
  - Employee +2 or more: \$78.00
- Northwind Health Plus:
  - Employee Only: \$55.00
  - Employee +1: \$71.00
  - Employee +2 or more: \$89.00

Sources:

- Benefit\_Options.pdf.md - default/Benefit\_Options.pdf.md/a0b4db647ee34e4d8c8c3e3c65acaea8 [Tuesday, April 9, 2024]

En résumé, l'intégration d'Azure Document Intelligence et l'utilisation du format Markdown ont permis d'améliorer la performance du RAG en prenant en compte les tableaux et en fournissant une sémantique supplémentaire au modèle de langage. Cette approche a démontré son efficacité et nous allons l'appliquer dans la suite de ce projet.