# Summarize text from large documents

Ever have a large document that you don’t have time to read and want a summary so you can determine if it’s worth investing the time? The Summarize Text API in Cognitive Services was designed to extract key sentences in a document as they relate to the topic so you can quickly understand what the document is about.

Overview

## Diagram Description automatically generated

## The document is first uploaded to a SharePoint library

* Power Automate will OCR and extract the text of the document
* The text is then sent to the summarization API via an Azure Function.
* The resulting summary text is then updated in a multi-line text field in the SharePoint document library where it is indexed and searchable.

## Setup

In your Azure subscription you’ll need to configure the [Cognitive Services Text Analytics](https://portal.azure.com/#create/Microsoft.CognitiveServicesTextAnalytics) feature. This will setup all the necessary components you need, specifically the Endpoint to invoke the service and key to authenticate properly. I created a new resource group on Azure to contain all the components we’ll need.

Graphical user interface, text, application

Description automatically generated

*Endpoint API needed to call the service*

Graphical user interface, application

Description automatically generated

*Key needed when calling the service*

## Power Automate

The core of solution resides in Power Automate. Activities can be configured to process the document and OCR and extract the text needed to summarize the document.

Graphical user interface, application, Teams

Description automatically generated

*PowerAutomate core activities*

Text, icon

Description automatically generatedTo accomplish there are 2 main steps that need to be configured in Power Automate:

1. **Recognize Text In Image** – use AI Builder in Power Automate to train a model that will “Extract all the text in photos and PDF Documents (OCR)”. This will extract text in PDF (Image and text) documents which is what we see the most.
2. **HTTP call to an Azure function** – The summarization API does not have a user friendly interface so I created an Azure function that will conduct all the processing necessary and returns the summary of the document.

## Putting it all together

Now that we have the results from AI Builder, we can send this to the Azure Function for processing.

## Send the contents of the file to AI Builder

Graphical user interface, application

Description automatically generated

## Extract the OCRed text

The result of the AI Builder activity is a JSON string that needs to be parsed. Ultimately what were looking for is the “text” attribute in the JSON file. This will get appended to our variable as we loop through the entire JSON output.

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

*Loop through the JSON file extracting the “text” attribute*

## Deploy the Azure function

As part of this solution, I have created an Azure function that will need to be deployed to the resource group created earlier. Download the Visual Studio solution from [GitHub](https://github.com/Spucelik/SummarizeDocumentAzureFunction) and update the AzureKeyCredential with a valid Key from the [Cognitive Services Text Analytics](https://portal.azure.com/#create/Microsoft.CognitiveServicesTextAnalytics) feature you created earlier.

A screenshot of a computer

Description automatically generated with medium confidence

Build and deploy the Azure function to the resource group created above. This will provide you with the endpoint URL need to make the HTTP call in Power Automate.

Graphical user interface, text, application

Description automatically generated

## Call the Azure Function

Now that you have the Azure function deployed, we have everything configured to send the extracted text to the summarization API for processing.

Graphical user interface, text, application

Description automatically generated

Once you have the results back, update the library.

Graphical user interface, text, application, email

Description automatically generated

The resulting text is then displayed in the library for users to view and search.

Graphical user interface, application, email

Description automatically generated