Solutions for Large Excel Data Pipeline (600+ Files/Year)

■■ Problem Summary

1. Copilot Studio fails or times out with large Excel files due to runtime limits.
2. It’s not designed for bulk data ingestion (600+ files with multiple sheets).
3. Mixed data types (numbers + strings) cause schema and memory issues.
4. Large-scale Excel ingestion needs an ETL pipeline, not conversational AI tools.

■ Option 1: Azure Data Factory + Data Lake + Synapse/Fabric

Use Azure Data Factory pipelines to read, transform, and load Excel files into Azure Data Lake. Then analyze using Synapse or Fabric. Ideal for enterprise-scale ingestion.

Benefits: Handles large volumes, supports incremental ingestion, integrates with AI Foundry.

■ Option 2: Python / Pandas Automation

Use a Python ETL script to process all Excel files locally or in Azure. This is fast to prototype and can merge all sheets into a single Parquet file.

import pandas as pd, glob all_data = [] for file in glob.glob("data/*.xlsx"): xls = pd.ExcelFile(file) for s in xls.sheet_names: df = pd.read_excel(file, sheet_name=s, dtype=str) df['source_file'] = file; df['sheet_name'] = s all_data.append(df) combined = pd.concat(all_data, ignore_index=True) combined.to_parquet("cleaned_data.parquet")

■ Option 3: Microsoft Fabric

Fabric unifies Data Factory, Data Lake, Power BI, and AI. Use Dataflows Gen2 to ingest and normalize all Excel files directly into a Lakehouse.

■ Option 4: LangChain + Azure Blob for AI Chatbot

If the goal is to query Excel data, upload processed files to Azure Blob, use LangChain’s loaders to generate embeddings, and connect via Azure AI Foundry.

■ Suggested Architecture

[Excel files → ADF/Python ETL → Data Lake/Synapse → AI Search + OpenAI → AI Foundry Chatbot]

■ Recommendation

1. Move all raw Excel files to Azure Blob Storage.

2. Use Azure Data Factory or Python to clean and combine into Parquet/CSV.
3. Connect the cleaned dataset to Azure AI Search in AI Foundry for RAG-based querying.