PROBLEMS AND SOLUTIONS FOR DAY 4

This conversation chronicled a series of cascading dependency errors encountered while trying to install a complex set of NLP, RAG, and UI libraries. The core issue was the **failure of low-level C-extensions to compile** in the Windows environment, which blocked the installation of major packages like spaCy and ChromaDB.

Here is a summary of the issues encountered and the proposed (or attempted) resolutions:

| ❌ Issue Encountered | 📦 Affected Libraries | Root Cause Identified | ✅ Resolution Strategy |
| --- | --- | --- | --- |
| **No module named 'srsly.ujson.ujson'** | spaCy, Presidio, medSpaCy | Corrupted or failed compile of the srsly C-extension. | Manual pip install --upgrade srsly and clean reinstallation of NLP libraries. |
| **No module named 'pydantic\_core.\_pydantic\_core'** | ChromaDB, LangChain, Haystack-AI | Corrupted or failed compile of the pydantic-core C-extension. | Manual uninstall and upgrade/reinstall of pydantic, pydantic-core, and chromadb. |
| **No module named 'cymem.cymem'** | spaCy (Low-level dependency) | Failed compile of the cymem C-extension. | Manual uninstall and reinstall of cymem and related C-extension libraries (preshed). |
| **No module named 'murmurhash.mrmr'** | spaCy (Low-level dependency) | Failed compile of the murmurhash C-extension. | The cumulative effect pointed to a severe **C++ compiler visibility issue**. |
| **Failed to build thinc** | spaCy (Core dependency) | **Definitive confirmation of C++ compiler failure.** The core spaCy dependency could not compile its C-extensions. | **1. Install/Verify Microsoft Visual C++ Build Tools.** **2. Destroy and recreate the virtual environment.** |
| **'C:\Program' is not recognized...** | Attempting to fix the compiler visibility issue. | Missing quotes when executing a Windows batch file with spaces in the path. | Correct execution of the vcvars64.bat script using double quotes around the full path: "%VS\_PATH%\vcvars64.bat". |

Export to Sheets

**Final Resolution Path**

The conversation established that the core problem was the Python environment's inability to find or utilize the necessary C++ compiler on Windows, resulting in the failure to build high-performance components.

The complete solution proposed was:

1. **System Prerequisite:** Install/Verify the **Microsoft Visual C++ Build Tools** ("Desktop development with C++" workload).
2. **Environment Fix (Crucial):** Run the **vcvars64.bat** script *outside* the virtual environment to correctly set the compiler PATH variables for the current terminal session:

Bash

"C:\Path\to\Visual Studio\vcvars64.bat"

1. **Clean Slate:** Destroy the old virtual environment and create a brand-new one.
2. **Strategic Install:** Reinstall all packages in smaller, logical groups to prevent large-scale dependency failures.
3. **Finalize:** Download the necessary spaCy language model (en\_core\_web\_sm).