Structuring Unstructured Data with LLMs: DSPy Practical Assignment

Why This Matters

Unstructured data (text, PDFs, web content) makes up **80-90% of data**, yet it's unusable for analysis without costly manual processing. **LLMs solve this by acting as "structured data compilers"** – converting messy text into clean, query able formats (entities, relations, graphs). This assignment tests your ability to:

- · **Bridge theory and practice** (real-world data ≠ textbook examples)
- · **Handle LLM uncertainty** (confidence loops, error resilience)
- · **Build production-ready pipelines** (not just one-off scripts)

The provided DSPy codebase demonstrates **exactly what modern data engineering teams need**:

- 1. **Entity extraction** → Turning text into typed objects (e.g., "pelletized frass" → Drug)
- 2. **Intelligent deduplication** → Solving real-world noise (e.g., "PB IC", "pea-barley intercrop", "pea-barley intercrops" → 1 entity)
- 3. **Knowledge graph generation** → Creating visual, query able relationships (Mermaid diagrams)

Code Walkthrough (Key Concepts for Applicants)

Code Sample: https://colab.research.google.com/drive/1b-w8EBRn_bRCysFnnBNUXCqg05KR8VsX?usp=sharing

1. Entity Extraction (ExtractEntities Signature)

```
class EntityWithAttr(BaseModel):

entity: str = Field(description="the named entity")

attr_type: str = Field(description="semantic type (e.g. Drug, Disease)")
```

```
class ExtractEntities(dspy.Signature):

paragraph: str = dspy.InputField()

entities: List[EntityWithAttr] = dspy.OutputField()
```

- · **Why it's clever**: Uses Pydantic to **force structured outputs** from LLMs. No more regex parsing of free-text responses!
- · **Your takeaway**: Always define *exactly* what the LLM should output. DSPy validates responses against your schema.

2. Deduplication with Confidence Loops

```
def deduplicate_with_lm(items, batch_size=10, target_confidence=0.9):

while True:

pred = dedup_predictor(items=batch)

if pred.confidence >= target_confidence: # Critical safety check!

return pred.deduplicated
```

- · **Why it's clever**: LLMs hallucinate. This loop **self-corrects** until confidence ≥ 90%.
- · **Your takeaway**: Never trust a single LLM call. *Always* add validation loops for critical tasks.

3. Mermaid Graph Generation

```
def triples_to_mermaid(triples, entity_list):

# Only allows entities from our deduplicated list as nodes

entity_set = {e.strip().lower() for e in entity_list}

...
```

```
lines.append(f" {_clean(src)} -- {lbl} --> {_clean(dst)}")
```

- · **Why it's clever**: Prevents "garbage nodes" by **strictly enforcing entity validity**.
- · **Your takeaway**: Output formats must be *robust* – real data breaks naive assumptions.
- · Mermaid link

Your Assignment (Due in 72 Hours)

Task

Scrape **10 URLs** (provided below), process their text using the DSPy pipeline, and deliver:

- 1. **10 Mermaid diagrams** (one per URL) visualizing key relationships.
- 2. **A structured CSV** with columns: link, tag, tag_type.
- 3. **A Colab notebook** showing your full implementation.

URLs to Scrape

- 1. https://en.wikipedia.org/wiki/Sustainable_agriculture

- 2. https://www.nature.com/articles/d41586-025-03353-5
- 3. https://www.sciencedirect.com/science/article/pii/S1043661820315152
- 4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10457221/
- 5. https://www.fao.org/3/y4671e/y4671e06.htm
- 6. https://www.medscape.com/viewarticle/time-reconsider-tramadol-chronic-pain-2025a1000ria
- 7. https://www.sciencedirect.com/science/article/pii/S0378378220307088
- 8. https://www.frontiersin.org/news/2025/09/01/rectangle-telescope-finding-habitable-planets
- 9. https://www.medscape.com/viewarticle/second-dose-boosts-shingles-protection-adults-aged-65-years-2025a1000ro7
- 10. https://www.theguardian.com/global-development/2025/oct/13/astro-ambassadors-stargazers-himalayas-hanle-ladakh-india

Assignment Deliverables

1. Mermaid Diagrams (10 total)

- · Save as mermaid_{i}.md (e.g., mermaid_1.md)
- · **Must include**:
    - o Valid Mermaid syntax (test in Mermaid Live Editor)
    - o Only entities from your deduplicated list as nodes
    - o Edge labels trimmed to 40 chars (as in example)

2. Structured CSV (tags.csv)

link	tag	tag_type
https://...	sustainable agriculture	Concept
https://...	nitrogen uptake	Process
...	...	...

Rules:

- · tag: Exact entity string (e.g., "pea-barley intercrop", not "intercrop")
- · tag_type: Semantic category (e.g., Crop, Process, Measurement)
- · **No duplicates per URL** (use your deduplication logic!)

3. Colab Notebook

- · **Must include**:
    - o Full code with **comments explaining key steps**
    - o Output csv

How to get Free LLM API key to use with DSPY?

- 1) Follow steps using your own account to get Free LLM api keys from here: https://scribehow.com/viewer/Sign_Up_for_Longcat_API_Platform__9sYiobPNS0OnXzyxKHu4zg?add_to_team_with_invite=True&sharer_domain=gmail.com&sharer_id=e0b8270f-e494-45b1-b41a-c6adf9f11845
- 2) You might run into limits, so you can ask to increase your limits- https://scribehow.com/viewer/Request_More_LongCat_API_Quota__0kdRFlLmTdKCulL67qz_rA?add_to_team_with_invite=True&sharer_domain=gmail.com&sharer_id=e0b8270f-e494-45b1-b41a-c6adf9f11845