PROJECT 4: “DrugScan & Summarise”: Detect Drug Mentions in Clinical PDFs and Produce a Drug-Centric Digest

1 Business scenario

Your pharmacology team reviews dozens of open-access journal articles every week. They need a one-page brief that:

1. **Lists every drug** discussed in the paper.
2. Provides a **short fact** (indication, mechanism, or warning) for each drug.
3. Gives a **four-sentence abstract** that focuses on how the drugs were used or evaluated.

2 Learning objectives

- Use **scispaCy** (or a lightly-fine-tuned spaCy model) for **drug NER** in noisy PDF text.
- Compare **static embedding summarisation** (TextRank) with a **transformer abstractive model** (BART-base or T5).
- Query a **public drug knowledge file** (DrugBank Open Data CSV or openFDA labels) to enrich the output.
- Package results in a simple **Streamlit dashboard** or command-line script.

3 Data & resource links

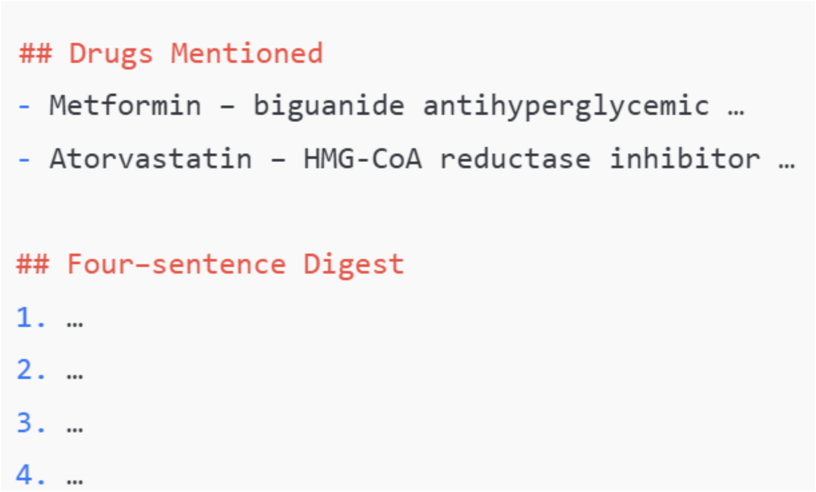
What you need	Where to get it
Sample PDFs – three pharmacology articles *(download the PDF tab on each page)*	• Metformin RCT in metabolic syndrome (PMC ID: **PMC8560579**) PMC • Atorvastatin lipid-lowering review (PMC ID: **PMC6464917**) PMC • Safety of ibuprofen vs paracetamol (PMC ID: **PMC3099387**) PMC
Larger pool for experimentation	PubMed Central Open-Access subset download page PMC
Drug NER model	scispaCy (en_ner_bc5cdr_md) on GitHub GitHub
Drug facts	*Either* DrugBank Open Data CSV (requires free academic sign-up) DrugBank *or* openFDA drug-label downloads OpenFDA

(If campus firewall blocks these, using open wifi, download the three PDFs and a mini-CSV with 10 drug fact rows in the starter repo.)

4 Core tasks

1. **PDF text extraction**
    - Use **PyMuPDF** (fitz) or pdfminer.six; strip headers/footers and de-hyphenate.
2. **Drug NER**
    - Run scispaCy model → collect (drug_surface, char_span).
    - Optional: add a **rule-based matcher** for dosage patterns (“mg”, “IU”).
3. **Quick fact look-up**
    - Normalize surface forms to lower-case.
    - String-match against the “name” and “synonyms” columns in DrugBank CSV (or openFDA JSON).
    - Return a one-line summary field (e.g., *“Metformin – biguanide antihyperglycemic for type 2 diabetes”*).
4. **Focused summarisation**
    - Extract all sentences that contain ≥1 drug; concatenate into a mini-document.
    - Produce:
        - **Extractive baseline** – TextRank top-4 sentences.
        - **Abstractive system** – BART-base (pre-trained) max length = 4 sentences.
    - Compare with ROUGE-L against the article’s own abstract (drug-filtered).
5. **Output formatting**
    - JSON or Markdown:

    Markdown example:



```
## Drugs Mentioned
- Metformin – biguanide antihyperglycemic …
- Atorvastatin – HMG-CoA reductase inhibitor …

## Four-sentence Digest
1. …
2. …
3. …
4. …
```

6. **Optional Streamlit mini-app**
   - File-uploader → spinner → shows the above Markdown with drug names highlighted.

5 Enhancement ideas (for extra credit)

- **Fine-tune** BART on 500 random PubMed abstracts vs full bodies for domain style.
- Add **confidence score filtering** for NER and summariser (drop < 0.3).
- Display drug mentions overlaid on the PDF text (PyMuPDF page widget).
- Export the digest as a **PDF** or **HTML** report for email.

6 Expected deliverables

1. **Notebook** (DrugScan.ipynb) with all code, figures, and commentary.
2. data/ folder with sample PDFs and drug fact file.
3. output/ folder containing the JSON/Markdown digests.
4. (If built) app.py Streamlit script + screenshot.
5. README.md with environment setup and run commands.
6. Presentation as per template