Automatska ekstrakcija podatka iz naučne literature pomoću velikih jezičkih modela

Od PoC-a do reproducibinog open-source pipeline-a

Zoran Stojanović

Maj 2026.

ITN SANU

Webinar Srpske mreže za reproducibilnost istraživanja

RSORN

Институт техничких наука САНУ

Agenda

- Motivacija
- RAG arhitektura
- Tech stack - lično iskustvo
- MP Toxicology - PoC pipeline
- research_intel - production pipeline
- Evidence mapping i reproducibilnost
- Demo i zaključak

Lista skraćenica

Algoritmi i infrastruktura

- **LLM** - Large Language Model
- **RAG** - Retrieval-Augmented Generation (Generisanje dopunjeno pretraživanjem)
- **PoC** - Proof of Concept
- **API** - Application Programming Interface
- **DOI** - Digital Object Identifier
- **JSON** - JavaScript Object Notation
- **JATS** - Journal Article Tag Suite
- **FAISS** - Facebook AI Similarity Search
- **GROBID** - GeneRation Of BIbliographic Data
- **NCBI** - National Center for Biotechnology Information
- **DB** - Database (baza podataka)
- **ACID** - Atomicity, Consistency, Isolation, Durability (Atomičnost, Konzistentnost, Izolacija, Trajnost)

Biohemija (oksidativni stres)

- **MDA** - Malondialdehid
- **SOD** - Superoksid-dizmutaza
- **CAT** - Katalaza
- **GSH** - Glutation

Polimeri mikroplastike

- **MP** - Mikroplastika
- **PET** - Polietilen-tereftalat
- **PS** - Polistiren
- **PE** - Polietilen
- **PVC** - Polivinil-hlorid

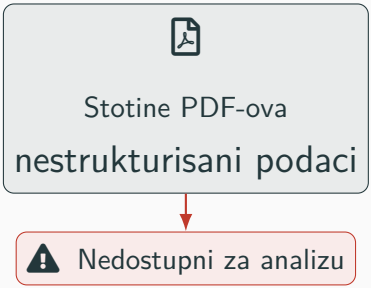
Rečnik pojmova

- **chunking** - segmentacija teksta na isečke
- **domain shema** - domenska šema (struktura podataka za oblast)
- **ekstrakcija** - izdvajanje strukturisanih podataka
- **embedding** - vektorska reprezentacija teksta
- **fine-tuning** - fino podešavanje modela
- **flash cards** - kartice za učenje
- **ingestion** - unos i obrada podataka
- **open-source** - otvoreni kod
- **output** - izlaz / izlazni podaci
- **parsing** - raščlanjivanje teksta
- **pipeline** - tok obrade podataka
- **provenance** - poreklo / istorijat podatka
- **query** - upit
- **recall** - odziv (mera kvaliteta pretrage)
- **repair loop** - petlja za automatsku korekciju
- **retrieval** - preuzimanje relevantnih isečaka
- **semantic search** - semantička pretraga
- **storage** - skladište podataka
- **sumarizacija** - sažimanje teksta
- **vendor lock-in** - zavisnost od jednog dobavljača / tehnologije
- **workflow** - tok / redosled radnih koraka
- **Multilevel deterministic cache** - keš na više nivoa sa SHA256 ključevima (embeddings, ekstrakcije, repair-i)
- **per-source throttle** - kontrola brzine zahteva posebno za svaki izvor (PubMed, arXiv, S2, CrossRef)

Motivacija

Problem: Obim naučne literature se uvećava brže nego što uspevamo da je ispratimo

- PubMed indeksira $>$**1.5 milion** novih radova godišnje
- Toksikologija mikroplastike: **stotine studija** sa različitim protokolima, vrstama, dozama, biomarkerima
- Ručna ekstrakcija podataka: spora, skupa, podložna greškama
- Rezultat: **podaci ostaju zarobljeni** u PDF-ovima



Reproducibinost: podaci moraju biti proverljivi

Centralni problem
Ako ne možemo da pratimo **koji dokument**, **koji odeljak** i **koji citat** stoji iza ekstrahovanog podatka - ekstrakcija nije reproducibilna.

Šta nam treba:

- Strukturisani izlaz sa **istorijom podatka - evidence provenance** (članak → paragraf → citat)
- Verifikabilnost: svaki podatak može da se prati do izvora
- Skalabilnost: stotine radova, ne samo jedan
- Otvoreni alati i ponovljivi workflow (proces obrade)

Šta već postoji - komercijalni istraživački alati

Alat	Šta radi	Ograničenje
NotebookLM	Sažeci, audio podkasti; Data Tables → Google Sheets (dec. 2025.)	Ručni upload; šema kroz chat, nije programska
Elicit	API (mart 2026.); PRISMA 2020.; ekstrakcija iz 20.000 radova; 95%+ tačnost	Nema trajnog DB; bez custom domain šeme
Consensus	Deep Search: strukturisani izveštaj; Consensus Meter (% naučnog slaganja)	Bez programske šeme; nema statističke analize
SciSpace	Ekstrakcija tabela/figura → CSV/Excel; SLR agent (PRISMA)	CSV samo; bez custom JSON; bez trajnog DB
Scholarcy	Ekstrakcija nalaza; JSON output (API); flash cards, BibTeX	JSON šema fiksna (Scholarcy-jeva); API zasebno plaćen
ChatPDF / Perplexity	Q&A po PDF; tabele iz više radova; Academic mode (Semantic Scholar)	5 - 10% halucinacija; nema trajnog storage-a ni custom šeme

Zajednički problem: black-box alati - nema prilagođene šeme, nema reproducibilnosti.

Zašto custom pipeline?

Komercijalni alati

- Generički izlaz, bez domain logike
- Session-based: podaci se ne čuvaju trajno
- Nema statističke analize, samo sažeci
- Zatvoreni, ne mogu se automatizovati end-to-end
- Pretplata, ograničene API kvote

Custom RAG pipeline

- JSON/Pydantic šema prilagođena domenu
- PostgreSQL + Qdrant: trajni, strukturisani storage
- Python statistika: prava analiza nad podacima
- Chunk $\rightarrow$ article $\rightarrow$ quote traceability
- Open-source, ponovljiv, bez vendor lock-in

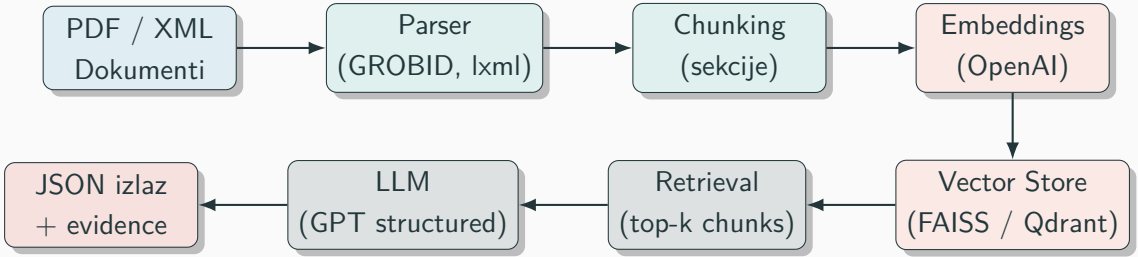
Naš pipeline vs komercijalni alati

Kriterijum	research_intel	SciSpace	Elicit	NotebookLM
Akvizicija	Automatska - PubMed, arXiv, S2, CrossRef	UI pretraga + ručni upload	API (mart 2026) + ručni upload	Potpuno ručno
Šema	JSON/Pydantic, programski prilagođena	CSV/Excel kolone; bez custom JSON	Strukturisane kolone; bez custom JSON	Data Tables (→ Sheets); bez prog. šeme
Storage	PostgreSQL + Qdrant, trajno (ACID)	Session, CSV export	Session, CSV/BibTeX	Google Sheets; bez DB
Analiza	Python + statistički modeli (SciPy, Pandas)	Nema	Tekstualna sinteza	Tekstualni sažeci
Izveštavanje	Tekst + citati + grafikoni	CSV/Excel tabela	PRISMA izveštaj sa citatima	Sažeci + Data Tables
Automatizacija	End-to-end (jedan upit → gotov izlaz)	Klik-po-klik	Semi-auto (PRISMA workflow)	Ručni upload + chat

Scholarcy (JSON API; šema fiksna) i Perplexity (tabele; halucinacije 5 - 10%) - detalji u slajdu "Šta već postoji".

RAG arhitektura

Retrieval-Augmented Generation (RAG) - pregled



RAG kombinuje **pretragu** relevantnih delova dokumenta sa **generacijom** strukturisanog odgovora - LLM "vidi" samo relevantne odeljke.

Zašto RAG, a ne fine-tuning?

Fine-tuning

- Skupo, zahteva labeled data
- Teško ažurirati kada dolaze novi radovi
- Halucinacije ostaju ako model "pamti" pogrešno
- Nema reference na izvor

RAG ✅

- Radi sa existing modelima
- Novi radovi = nova ingestion, bez re-treninga
- Grounded: model citira šta je pročitao
- Provenance je prirodna posledica arhitekture

Structured output + schema validation

Problem: LLM vraća slobodan tekst - kako da dobijemo tabelu?

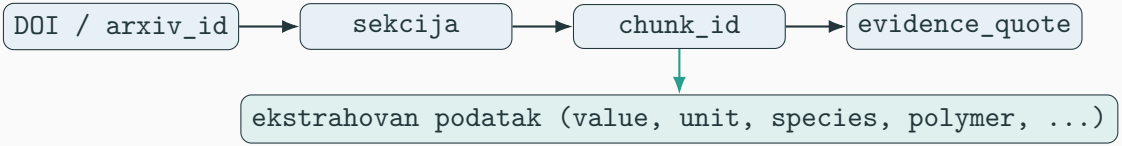
Rešenje: JSON shema + repair loop

```
{
  "parameter": "MDA",          # biomarker
  "value": 2.34,
  "unit": "nmol/mg protein",
  "polymer": "PET",
  "evidence_quote": "MDA levels increased...",
  "chunk_id": "doc_042_sec_3"  # provenance
}
```

Ako LLM vrati nevažeći JSON → **repair poziv** sa greškom validacije.

Evidence mapping - traceability

Šta je evidence mapping?
Svaki ekstrahovan podatak nosi **lanac porekla**: članak → sekcija → paragraf → citat (quote).



Reviewer može da **proveri svaki red tabele** - bez black-boxa.

Tech stack - lično iskustvo

Put: od PoC-a do production pipeline-a

	MP Toxicology (PoC)	**research_intel (prod)**
Jezik	Python 3.10 / Jupyter	Python 3.12 + asyncio
PDF parser	scipdf (GROBID wrapper)	PyMuPDF + GROBID 0.8
XML/HTML	nije obradjeno	lxml (JATS) + BeautifulSoup4
Embeddings	OpenAI text-emb-3-large	OpenAI / local HuggingFace
Vector store	**FAISS** (in-memory)	**Qdrant** (Docker, persistent)
Relaciona DB	SQLite (cache)	**PostgreSQL 16** + SQLAlchemy
LLM	GPT (structured output)	GPT (structured output)
Akvizicija	lokalni PDF-ovi	PubMed, arXiv, S2, CrossRef + lokalni PDF-ovi
Orkestracija	Jupyter notebook	Async pipeline + Alembic

Šta sam naučio - alati i odluke

Alati koji su se pokazali

- **GROBID** - ML parser; bolji od regex na naučnim PDF-ovima
- **Qdrant** vs FAISS: persistent store neophodan van Jupytera
- **pydantic** za config i validaciju - štedi na bug-ovima
- **tenacity** za retry logiku prema API-jima
- **asyncpg** - dramatično ubrzanje ingestion-a

Šta bih uradio drugačije

- Početi sa PostgreSQL odmah, ne SQLite
- Schema-first pristup od dana 1
- Repair loop je neophodan - LLM češće greši u JSON-u
- JATS XML (PubMed Central) je **mnogo čistiji** od PDF-a
- Testovi za parsere pre nego za ekstrakciju

Python biblioteke i baze podataka

Python biblioteke

- **pydantic** - validacija JSON šeme i konfiguracije modela
- **tenacity** - automatski retry sa exponential backoff
- **asyncpg** - async PostgreSQL driver (brzi batch ingestion)
- **SQLAlchemy** - ORM + async DB pristup (C3)
- **Alembic** - verzionisane migracije DB šeme
- **PyMuPDF** - parsiranje i čitanje PDF-ova
- **lxml** - parsiranje JATS XML (PubMed Central)
- **BeautifulSoup4** - parsiranje HTML (arXiv, publisher sajtovi)
- **openai** - OpenAI SDK: embeddings + structured output
- **requests** - HTTP pozivi prema akvizicionim API-jima
- **pandas** - tabele i statistička analiza ekstrahovanih podataka

Baze podataka

- **PostgreSQL 16** - relaciona baza, ACID transakcije; trajni storage za članke, dokumente i chunkove (production)
- **Qdrant** - vektorska DB (Docker); filtered similarity search, persistent vektorski indeks (production)
- **SQLite** - laka lokalna DB; keš embeddinga i ekstrakcija u PoC-u
- **FAISS** - in-memory vektorski indeks (PoC); bez persistencije → zamenjen Qdrant-om

MP Toxicology - PoC pipeline

MP Toxicology RAG Extraction - kontekst

Problem:

- Toksikologija mikroplastike: **heterogeni podaci** (vrste, polimeri, doze, biomarkeri)
- Cilj: automatski popuniti tabelu iz stotina PDF studija
- Fokus: biomarkeri (ALT, AST, triglycerides, glucose, MDA, SOD, CAT, GSH, ...)

Dataset:

- $\sim$90 recenziranih radova (lokalni PDF-ovi)
- Životinjski modeli: zebra fish, miševi, pacovi
- Polimeri: PET, PS, PE, PVC, itd.

JSON red:

species: *D. rerio*
polymer: PET
dose: 100 mg/L
parameter: MDA
value: 2.34
unit: nmol/mg
evidence: "..."

PoC pipeline - arhitektura

1. **Ingestion:** PDF → **GROBID** (Docker) → strukturisani XML tekst
2. **Chunking:** podela po sekcijama (Methods, Results) sa `textacy`
3. **Embeddings:** `text-embedding-3-large` → **FAISS** index (keširan u SQLite da se ne ponavljaju API pozivi)
4. **Retrieval:** za svaki biomarker upit → top-k relevantnih chunk-ova
5. **Extraction:** GPT structured output po JSON shemi
6. **Repair:** ako JSON validacija padne → drugi LLM poziv sa greškom
7. **Output:** pandas DataFrame → CSV / Excel

Ključna odluka

Multilevel deterministic cache (jedna SQLite baza 3 vrste ključa)

SHA256 ključevi: embeddings, ekstrakcije i popravke (repairs)

reproducibiilnost i ušteda na API troškovima.

PoC pipeline - šta je radilo, šta nije

Radilo dobro

- GROBID znatno bolji od direktnog PyMuPDF na naučnim PDF-ovima
- Biomarker regex reranking iza embedding-a poboljšao recall
- Keš eliminisao 90%+ API poziva pri ponovnom pokretanju
- Evidence quote u svakom redu $\rightarrow$ laka verifikacija

Ograničenja PoC-a

- FAISS in-memory: gubi se pri restartu
- Jupyter nije dobar za velike batch-eve
- Nema standardizacije jedinica (nmol/mg vs µmol/g)
- Ručni unos PDF-ova - nema akvizicije

Demo - pokretanje ekstrakcije

```
# 1. Embed upit za biomarker
query = "MDA malondialdehyde oxidative stress"
q_emb = openai_embed(query)
# 2. Retrieve relevantne chunkove
chunks = faiss_index.search(q_emb, k=8)
# 3. LLM ekstrakcija po JSON shemi
result = gpt_extract(chunks, schema=ToxicologyRecord)
# 4. Repair ako validacija padne
if not validate(result):
    result = gpt_repair(result, error=validation_error(result))
df.loc[len(df)] = result  # 5. Snimi
```

(Live demo posle ovog bloka)

research_intel - production pipeline

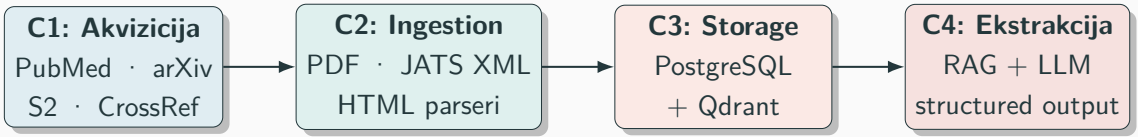
research_intel - motivacija za novi pristup

Zašto novi projekat?
PoC je pokazao da pristup radi - ali nije skalabilno: ručno prikupljanje PDF-ova, in-memory store, sekvencijalno izvršavanje. **research_intel** je open-source rešenje koje radi sve automatski.

Ciljevi:

- Automatska akvizicija radova (PubMed, arXiv, Semantic Scholar, CrossRef)
- Modularan ingestion za PDF, JATS XML, HTML
- Persistent storage: PostgreSQL + Qdrant
- Ponovljiv, testabilan, proširiv

Arhitektura - 4 komponente



- **C1–C3** su završeni i testirani
- **C4** (ekstrakcija) - u aktivnom razvoju; PoC MP pipeline dokazuje koncept
- **Planirano za C4:** ekstrakcija podataka iz tabela, grafikona i slika; ekstrakcija koda (kada je dostupan u radu)

C1: Akvizicija - multi-source

4 izvora, jedan API sloj:

- **PubMed** (NCBI E-utilities) - biomedicinska literatura
- **arXiv** - preprinti, CS, fizika, bioinformatika
- **Semantic Scholar** - semantička pretraga, citati
- **CrossRef** - DOI metapodaci, open access status
- **Lokalni fajlovi** - PDF/XML/HTML sa diska lokalno
- **Rate limiting:** `tenacity` retry + per-source throttle
- **Deduplication:** po DOI pre upisa u PostgreSQL

Šta se čuva:

- DOI, naslov, autori, abstract
- Journal, godina, open access flag
- PDF URL (Unpaywall)
- Raw metadata (JSON)

C2 + C3: Ingestion i Storage

C2 - modularan parser:

- **PDF** → PyMuPDF heuristika + GROBID za strukture
- **JATS XML** → lxml (PubMed Central; najčistiji format)
- **HTML** → BeautifulSoup4 (arXiv, publisher sites)
- Section-aware chunking sa provenance

C3 - dva tipa storage-a:

- **PostgreSQL 16** (SQLAlchemy async): članci, dokumenti, chunkovi
- **Qdrant** (Docker): vektorski indeks, filter po metapodacima
- **Alembic** migracije za šemu
- Async: `asyncpg` + `asyncio`

*JATS XML iz PubMed Central-a je **daleko čistiji** od PDF-a - uvek koristiti kada je dostupan.*

Evidence mapping i reproducibilnost

Šta znači reproducibiilnost ovde?

Naučna reproducibiilnost zahteva da drugi istraživač može:

1. Pokrenuti isti pipeline na istim dokumentima
2. Dobiti isti (ili ekvivalentan) strukturisani izlaz
3. Pratiti svaki podatak do **izvornog citata**
4. Verifikovati bez pristupa originalnom LLM pozivu

Naš odgovor:

- ✅ Open-source kod
- ✅ Fiksirane verzije modela

  `"gpt-4o-2024-08-06"`
- ✅ Keš API odgovora
- ✅ Evidence quote u svakom redu
- ✅ chunk_id $\rightarrow$ full provenance chain

Evidence mapping u praksi

DOI	Vrsta	Parametar	Vrednost	Evidence quote
10.1016/...	D. rerio	MDA	2.34 nmol/mg	"MDA levels were significantly elevated..."
10.1016/...	Mus mus.	SOD	18.2 U/mg	"SOD activity decreased by 34%..."
10.3389/...	Rattus	CAT	5.1 nmol/mg	"Catalase activity was reduced..."

Svaki red tablice može da se **verifikuje direktno u PDF-u** bez pokretanja modela.

Šire implikacije - open science

- **Meta-analize:** strukturisani podaci iz stotina studija dostupni za statističku analizu
- **Living reviews:** novi radovi se dodaju akvizicijom, pipeline se ponovo pokreće
- **Interoperabilnost:** JSON/CSV izlaz kompatibilan sa standardnim alatima
- **Demokratizacija:** mali timovi mogu raditi meta-analize koje su ranije zahtevale decenije ručnog rada

Open-source
research_intel pipeline je projektovan kao javni alat - prilagodljiv za bilo koji naučni domen, ne samo toksikologiju.

Demo i zaključak

Live demo - šta ćemo pokazati

1. **Pokretanje pipeline-a** na jednom naučnom PDF-u
2. **GROBID parsiranje** $\rightarrow$ strukturisani tekst po sekcijama
3. **Embedding + retrieval** $\rightarrow$ relevantni chunk-ovi za biomarker upit
4. **LLM ekstrakcija** $\rightarrow$ JSON red sa evidence quote-om
5. **Repair loop** $\rightarrow$ automatska korekcija nevažećeg JSON-a
6. **DataFrame output** $\rightarrow$ tabela sa provenance kolonama

Napomena: demo koristi MP_toxicology_RAG_extraction notebook (lokalno).

Zaključak

- RAG + structured output = **skalabilna, proverljiva ekstrakcija** iz naučne literature
- Evidence provenance je **arhitekturalno rešenje** za reproducibilnost, ne naknadna misao
- Put od PoC-a (Jupyter + FAISS) do production-a (async + Qdrant + PostgreSQL) je iterativan
- GROBID + JATS XML su **ključni enableri** za kvalitet parsiranja
- Otvorena pitanja: standardizacija jedinica, evaluacija tačnosti, multi-domen proširivost
- **Orkestracija agenata:** greška u ranom koraku multiplikuje se kroz pipeline - potrebne su feedback petlje i self-healing između agenata
- **Gubitak konteksta (Search):** top-k može izostaviti ključan chunk; trenutno PoC šalje do $\sim$40 ranked chunkova po radu (4 upita $\times$ 10, deduplikovano)

PoC demo je dostupan

`https://github.com/zorankiki/MP_toxicology_RAG_extraction`

`research_intel` production pipeline - u izradi, otvoreni za saradnju.

Hvala na pažnji!

Pitanja?

zoran.stojanovic@itn.sanu.ac.rs

PoC demo (GitHub)

Institut (web)