LitStudy

A Python package for literature reviews

Topici Speciale în Ingineria Software, grupa 506

Popescu Paullo-Robertto-Karloss

Horceag Andrei

Pasăre Roxana-Francisca

De ce acest articol?

- Analiza literaturii este dificilă din cauza numărului mare de articole.
- Cercetătorii au nevoie de statistici și vizualizări generate automat.

Motivație: Să facem aceste funcționalități accesibile printr-o interfață web simplă.

Cadru contextual și Obiectiv

Problema

- *litstudy* e util, dar greu de folosit fără Python.
- Lipsește o interfață simplă pentru analiză bibliografică.

Scopul

Să oferim o aplicație web care:

- încarcă un fișier .bib
- folosește litstudy
- afișează statisticile și rețelele într-un dashboard

Info LitStudy

- Articol s fost publicat în **SoftwareX,** în anul **2022.**
- Autorii: *Stijn Heldens, Alessio Sclocco, Henk Dreuning , Ben van Werkhoven, Pieter Hijma, Jason Maassen şi Rob V. van Nieuwpoort* propun un pachet Python dedicat analizelor bibliografice.
- Lucrarea se concentrează pe automatizarea procesului de explorare a literaturii științifice.
- Contribuie la simplificarea și structurarea revizuirilor de tip *mapping review.*

Context

- Numărul publicațiilor științifice crește rapid.
- Analiza manuală a literaturii devine dificilă și consumatoare de timp.
- Cercetătorii au nevoie de instrumente automate și replicabile.

Scop

- Prezentarea unui pachet Python (**litstudy**) pentru analiză bibliografică automată.
- Oferă un flux complet:

import → procesare → analize → vizualizări → rețele.

Problema abordată

- Revizuirile literaturii necesită:
  - colectare date
  - filtrare
  - statistici
  - vizualizări
  - rețele de autori/citări
- Majoritatea soluțiilor sunt fragmentate sau necesită mult cod personalizat.

Fundament teoretic

- LitStudy este un framework Python pentru explorarea colecțiilor de articole.
- Include funcții pentru:
  - import (.bib, API-uri)
  - curățare și filtrare
  - statistici
  - grafice
  - rețele de colaborare

Sursele de date suportate

- BibTeX (.bib)
- CrossRef
- Semantic Scholar
- Scopus
- Surse extensibile prin plugin-uri

Funcționalități principale

- Analiză bibliografică (autori, ani, jurnale)
- Statistici descriptive
- Vizualizări (grafice temporale, top autori)
- Rețele de:
  - co-autorat
  - citare

Rețele bibliografice

- Construite cu *networkx*.
- Identifică:
  - colaborări
  - grupuri de autori
  - conexiuni puternice

Topic Modeling

- Folosește:
  - LDA (model probabilistic)
  - NMF (metodă matematică de descompunere)
- Extrage teme comune în articole.
- Oferă o perspectivă asupra direcțiilor de cercetare.

Studiul de caz din articol

- Aplicat pe domeniul:
  - programare paralelă
  - GPU computing

- Au demonstrat:
  - evoluția numărului de articole
  - colaborările dintre autori
  - grupuri tematice identificate automat

Concluziile autorilor

- LitStudy simplifică analiza colecțiilor bibliografice.
- Este un instrument extensibil și reproductibil.
- Util în etapa exploratorie a unei revizuiri a literaturii.

Puncte forte

- Open-source și integrabil în aplicații.
- Automatizează sarcini consumatoare de timp.
- Suportă multiple surse de date.
- Rețelele de autori/citări sunt foarte utile.
- Perfect pentru *mapping reviews*.

Puncte slabe

- Necesită cunoștințe de Python → inaccesibil pentru utilizatori non-tehnici.
- Vizualizările sunt simple, nu interactive.
- Nu acoperă partea calitativă a unei revizuiri sistematice.
- Depinde de calitatea metadatelor din fișiere .bib.
- Unele funcții sunt slab documentate.

Relevanța pentru proiectul nostru

- Oferă funcționalitățile de analiză de care avem nevoie.
- Noi aducem un **dashboard web accesibil** tuturor.
- Eliminăm bariera Python.
- Facem analiza literaturii rapidă, vizuală și ușor de folosit.

Structura dataset-ului

- Conține ~100 articole științifice.
- Coloane principale: *title, authors, year, modality, task, source, link.*
- Datele provin din PubMed și au fost centralizate într-un singur fișier CSV.

Categorii în CSV

- **Modality:** CT, MRI, PET/CT, Ultrasound, General imaging.
- **Task:** detection, classification, segmentation, prognosis, review, analysis.
- Categoriile permit filtrare ușoară în aplicația finală.

Arhitectura Aplicației

1. **Input Layer:** Utilizator -> Upload CSV/BibTeX sau Interogare DBLP API.
2. **Processing Layer (Backend):**
   - *Data Cleaning:* **normalize_documents()** (Reparare date lipsă).
   - *Core Engine:* **litstudy** library.
3. *NLP Engine:* **scikit-learn** (TF-IDF & NMF).
4. **Presentation Layer (Frontend):** Streamlit (Dashboard, Tabs, Widgets).
5. **Output Layer:** Raport PDF (fpdf), Export CSV, Grafice .png.

Tehnologii Utilizate

- **Limbaj:** Python **3.12.10** (Versiune specifică pentru stabilitate).
- **Core Framework: Streamlit** (pentru interfața web rapidă).
- **Bibliometrie: litstudy** (motorul principal).
- **Analiză de Date & NLP:**
  - **pandas** (manipulare tabele).
  - **scikit-learn** (Modelare NMF).
  - **networkx** (Generare grafuri).
- **Rapoarte: fpdf** (Generare documente PDF programatic).

Gestionarea Robustă a Datelor

- **Provocare:** Fișierele CSV/BibTeX reale au adesea date lipsă (ex: lipsește Jurnalul sau Anul).
- **Soluția Noastră (Cod):** Funcția **normalize_documents**.
  - Detectează câmpuri **NaN** sau lipsă.
  - Completează automat sursa din alte metadate (**publisher**, **booktitle**).
  - Standardizează formatele de text (**clean_text**) pentru a preveni erori la generarea PDF.
- **Caching:** Integrare cu DBLP Cache (fișierele **.dblp** din proiect) pentru interogări rapide.

Interfața cu Utilizatorul (UI/UX)

- **Structură Modulară:** Utilizarea **st.tabs** pentru a separa logica:
  1. Statistici
  2. NLP (Topic Modeling)
  3. Rețele
  4. Export
- **Filtrare Dinamică (Sidebar):**
  Orice modificare (An, Autor, Cuvânt cheie) actualizează instantaneu *toate* graficele din aplicație folosind **st.session_state**.
- **Feedback Vizual:** Bare de progres și mesaje de stare (Success/Error/Warning).

Modelare de Subiecte (Topic Modeling)

- **De ce?** Pentru a înțelege *despre ce* sunt articolele fără a le citi pe toate.
- **Implementare:**
  - Am înlocuit abordarea simplă cu **NMF (Non-negative Matrix Factorization)** via **scikit-learn**.
  - **Vectorizare:** TF-IDF (Term Frequency-Inverse Document Frequency) pentru a elimina cuvintele irelevante (stop words).
- **Rezultat:** Extragerea automată a clusterelor de cuvinte (ex: Topic 1: "Deep, Learning, Neural", Topic 2: "MRI, Scan, Image").

Vizualizarea Rețelelor Academice

- **Funcționalitate:** Identificarea grupurilor de cercetători care publică frecvent împreună.
- **Tehnic:**
  - Construcția grafului folosind **litstudy.build_coauthor_network**.
  - Vizualizare interactivă integrată în Streamlit prin componenta **components.html**.
- **Util:** Ajută la identificarea liderilor de opinie și a "bulelor" de cercetare izolată.

Rapoarte automate PDF/CSV

- **Inovație:** Majoritatea tool-urilor doar *arată* datele. LitStudy le *livrează*.

1. Raport PDF Automat (class PDFReport):

- Capturează starea curentă a graficelor (figuri matplotlib).
- Le inserează într-un document formatat, alături de titluri și descrieri generate automat.
- **Rezultat:** Un document PDF complet cu toate statisticile generate.

2. Export Date Brute (CSV):

- Permite descărcarea dataset-ului după curățare și filtrare.
- **Scop:** Asigură **reproductibilitatea** cercetării și permite analize externe ulterioare.

Motorul NLP

```
# Transformam textul in numere, eliminand cuvintele comune (stop words)
tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2, stop_words='english')
tfidf = tfidf_vectorizer.fit_transform(text_data)
feature_names = tfidf_vectorizer.get_feature_names_out()
wip
# 3. ANTRENARE MODEL NMF
nmf_model = NMF(n_components=num_topics, random_state=42, init='nndsvd')
nmf_model.fit(tfidf)
topics_data = []
for topic_idx, topic in enumerate(nmf_model.components_):
    top_indices = topic.argsort()[:-11:-1]
    top_w = [feature_names[i] for i in top_indices]
    topics_data.append(f"Topic {topic_idx + 1}: {', '.join(top_w)}")
st.session_state['nmf_topics'] = topics_data
```

- **Vectorizare Inteligentă:**
  - Folosim TfidfVectorizer pentru a transforma textul în vectori numerici.
  - Filtrăm automat "zgomotul" (stop words) și cuvintele care apar prea des sau prea rar.
- **Algoritmul NMF:**
  - Am implementat **Non-negative Matrix Factorization** via scikit-learn.
  - Este o metodă nesupervizată care detectează automat teme comune în titluri și abstracte.
- **Optimizare Streamlit:**
  - Utilizarea st.session_state este critică: permite persistența datelor (topicurile generate) între reîncărcările paginii, oferind o experiență fluidă utilizatorului.

Provocări întâmpinate

- **Provocare 1:** Diacriticele în PDF.
  - *Soluție:* Funcția clean_text care encodează textul în latin-1 compatibil cu FPDF.
- **Provocare 2:** Performanța la dataset-uri mari.
  - *Soluție:* Optimizare prin filtrare în pași și caching local al rezultatelor DBLP.
- **Provocare 3:** Integrarea graficelor interactive în PDF static.
  - *Soluție:* Salvarea figurilor ca imagini temporare (tempfile) doar pentru momentul generării raportului.

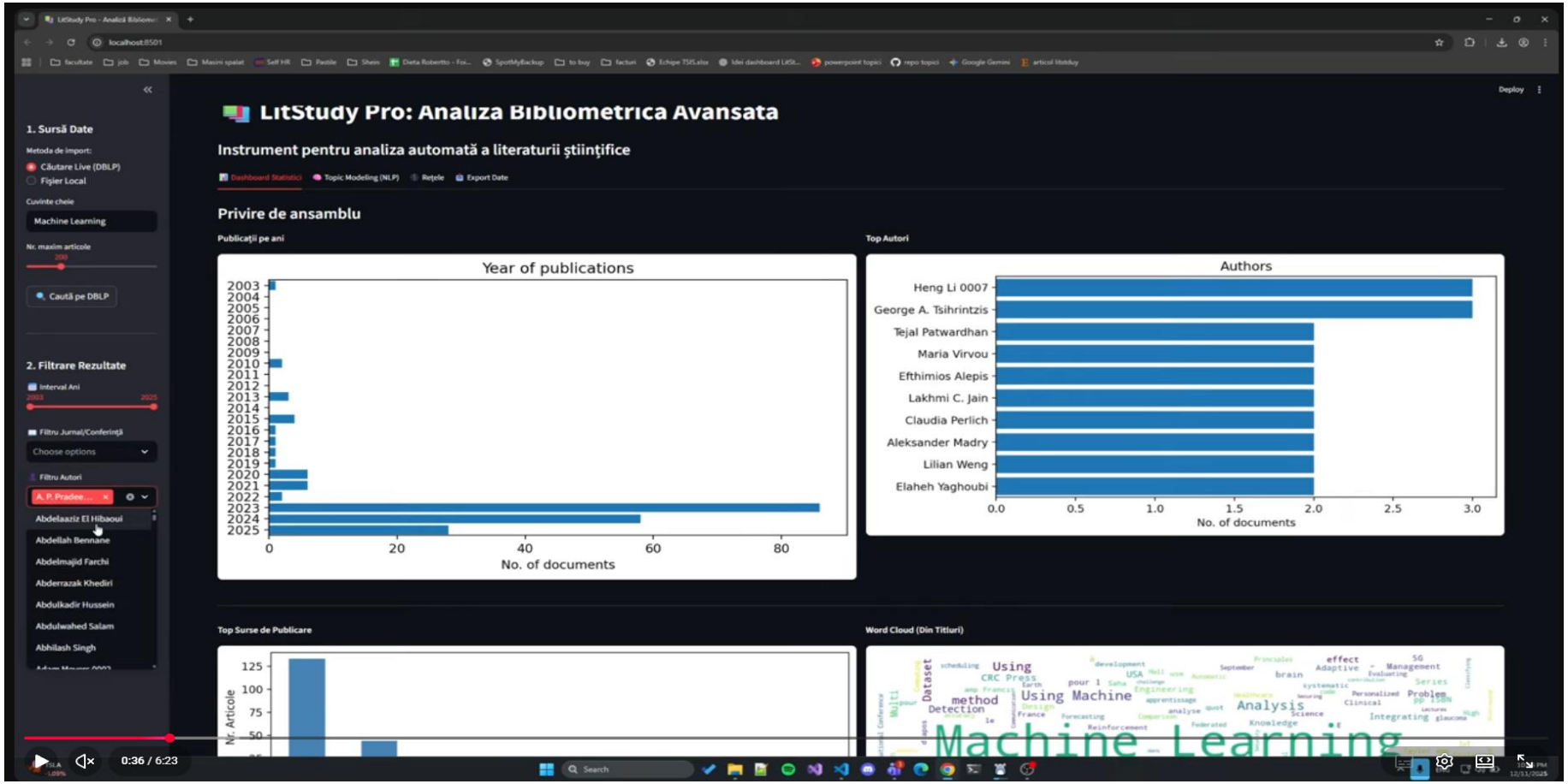
Analiza Dataset-ului Medical CSV

- **Dataset:** Articole despre imagistică medicală (CT/MRI).
- **Ce a descoperit LitStudy?**
  - *Tendință:* Creștere masivă a articolelor despre "Machine Learning" după 2018.
  - *Topicuri:* Separare clară între "Segmentation" (tehnic) și "Prognosis" (clinic).
  - *Autori:* Identificarea a 2 grupuri majore de cercetare care nu colaborează între ele.

Îmbunătățiri Posibile

- **NLP Modern:** Trecerea de la NMF la **BERT/Transformers** pentru înțelegere semantică (contextuală).
- **Semantic Search:** Căutare bazată pe înțeles ("AI for heart" să găsească și "Deep learning for cardiac"), nu doar pe cuvinte exacte.
- **Surse Noi:** Integrarea OpenAlex (sursă gratuită, alternativă la Scopus).

Demo



https://youtu.be/fNvYmaUuIEA

Concluzii

- Am reușit reproducerea funcționalităților din articolul original.
- Am transformat un script de cercetare într-un **produs software utilizabil**.
- Soluția este **Open Source**, **Modulară** și **Extensibilă**.

Mulțumim pentru atenție!

- **Github Repository:** *https://github.com/FranciscaPasare28/TSS*