6/5/2025

Polygraph

Detectarea “fake-news” în articole din presă

Militaru Ștefan
Cucu Mihnea
Oprescu Andrei

1.Ideea Proiectului

Scopul principal al proiectului polygraph este construirea unei aplicații care poate furniza date despre veridicitatea și legitimitatea unui articol ce îi pot fi utile unui utilizator. Deși recunoaștem că nu poate fi inventată o „soluție magică” care să spună cu certitudine garantată dacă un articol este complet adevărat, sperăm ca informațiile rezultate din analiza Polygrapgh să îi fie utile oricărui cititor de știri online.

Backlogul taskurilor care ne vor ajuta să ducem proiectul până la capăt este următorul:

1. Analiza textului știrii

- Extrage textul principal al articolului.
- Împarte conținutul în propoziții și analizează structura acestora.

2. Clasificare Fake News vs. Real News

- Algoritmi de machine learning care determină dacă articolul este fake sau real.
- Modele bazate pe NLP care detectează pattern-uri comune în știrile false (ex: exagerări, lipsa surselor, limbaj emoțional).

3. Verificarea surselor

- Compară site-ul articolului cu o bază de date de surse verificate (Media Bias/Fact Check, Open Sources).
- Avertizează utilizatorul dacă sursa este cunoscută pentru știri false.

4. Comparare cu știri de încredere

- Caută subiecte similare pe site-uri de știri verificate și compară faptele prezentate.
- Notifică utilizatorul dacă există diferențe semnificative între versiuni.

5. Analiză detaliată și raport explicativ

- Generează un raport cu argumente pro și contra pentru credibilitatea articolului.
- Explică ce factori au influențat scorul final.
- Evidențiază limbajul emoțional, lipsa surselor și alte trăsături problematice.

6. Interfață Web pentru utilizatori

- Un UI unde utilizatorii pot introduce un link sau un text de articol pentru analiză.
- Oferă un scor de credibilitate (ex: 0% – complet fake, 100% – foarte credibil).

7. Acces la sursele folosite pentru verificare

- Listează toate sursele folosite pentru verificare.
- Oferă link-uri către articolele relevante.
- Explică metodologia de selecție a surselor.

8. Sugestii de surse alternative

- Creează un mecanism care sugerează surse alternative de încredere.
- Prioritizează surse diverse pentru o perspectivă echilibrată.

9. Setări avansate pentru administratori

- Adaugă setări pentru pragurile scorului de credibilitate.
- Permite ajustarea sensibilității algoritmului.

10. Rezumat automatizat al articolelor

- Utilizează NLP pentru a extrage esențialul din articol.
- Afișează principalele puncte sub formă de listă.

11. Istoricul verificărilor

- Creează o pagină unde utilizatorul vede ultimele articole analizate.
- Permite ștergerea manuală a istoricului.

12. Raportarea analizelor incorecte

- Adaugă un buton „Raportează evaluarea” în UI.
- Creează un sistem de feedback pentru ajustarea modelului.

Arhitectura aplicației

Funcționarea generală a aplicației se va realiza după următorii pași:

1. Utilizatorul introduce un link sau un text de articol.
2. Backend-ul preia articolul și îl analizează:

- Tokenizare și curățare text.
- Detectarea sursei și verificarea în baza de date.
- Compararea conținutului cu știri similare.
- Analiza sentimentului și clasificarea fake vs. real.

3. Returnează utilizatorului un scor de credibilitate + argumente.

Tehnologiile utilizate la realizarea aplicației vor fi:

- Python, pentru tokenizarea textului, analiza acestuia, rularea algoritmilor de ML
- Django, pentru interfața cu care va interacționa utilizatorul

Aspecte Tehnice

Analiza și curățarea textului

1. **Extragerea textului**

   Prima etapă a algoritmului presupune extragerea textului din adresa url furnizată de utilizator. Pentru a realiza acest lucru ne vom folosi de librăria newspaper care ne pune la dispoziție diverse tool-uri pentru a extrage și manipula textul unui articol.

2. **Împărțirea în propoziții/cuvinte**

   La finalul acestui pas vom construi și un dicționar care reține numărul de apariții ale fiecărui cuvânt în text.

3. Extragere cuvinte cheie

   În etapa de analiză a textului, am decis și să extragem câteva cuvinte cheie reprezentative care sunt menite să îl ajute pe utilizator să perceapă tonul textului și subiectele discutate de acesta. Pentru a realiza acest lucru ne vom folosi de frecvența cu care apar acești termeni și îi vom selecta pe aceia care apar disproporționat de des. Frecvența aparițiilor termenilor în articolul curent este comparată cu frecvența aparției lor în alte texte din engleză, cu ajutorul datasetului unigram_freq, creditat în bibliografie.

După analiza calitativă a textului, am implementat un algoritm de clasificare automată a știrilor, care combină o componentă de **machine learning** cu o analiză de tip **NLP** (Natural Language Processing) axată pe caracteristici stilistice și de conținut. Scopul este de a determina dacă o știre este **reală** sau **falsă** pe baza textului.

Structura generală a algoritmului

1. **Încărcarea datelor**

   Se utilizează un set de date public (Fake and Real News Dataset), format din două fișiere CSV (True.csv și Fake.csv) care conțin știri reale și false. Acestea sunt combinate într-un singur DataFrame și amestecate aleatoriu pentru a evita eventuale patternuri nedorite.

2. **Preprocesarea textului**

Toate textele sunt transformate în litere mici și sunt eliminate semnele de punctuație, pentru a standardiza informația textuală.

3. **Modelul de clasificare (Machine Learning)**
    - Se folosește un pipeline format din:
        - TfidfVectorizer: transformă textul într-un set de caracteristici numerice pe baza frecvenței termenilor.
        - RandomForestClassifier: un algoritm de învățare automată care construiește mai mulți arbori de decizie și votează rezultatul.
        -
    - Modelul este antrenat pe 80% din date, iar 20% sunt folosite pentru testare. Se afișează un **classification report** care include acuratețea, precizia și scorurile F1.

4. **Predicție pe fișier .txt dat de utilizator**

   După antrenarea modelului, un fișier text extern este analizat. Se face o predicție binară: 1 (știre reală) sau 0 (știre falsă), însoțită de probabilitățile asociate.

5. **Analiza NLP suplimentară**

   Pe lângă predicția oferită de model, sunt adăugate trei tipuri de analize textuale:

    - **Analiza sentimentului**: se folosește biblioteca TextBlob pentru a calcula polaritatea emoțională a textului (între -1 și 1). Știrile foarte emoționale pot fi suspecte.
    - **Detectarea exagerărilor**: se caută cuvinte precum „incredibil”, „niciodată”, „toți”, care sunt adesea asociate cu limbajul senzaționalist.
    - **Verificarea surselor**: se caută expresii vagi precum „experții spun” sau „surse anonime” care pot indica lipsa de credibilitate.
    -

6. **Interpretarea rezultatelor**

   Dacă probabilitatea calculată de model este mai mare de 70%, algoritmul decide cu încredere dacă știrea este reală sau falsă. Dacă nu, se semnalează o predicție incertă.

Etapa de verificare a informației prin comparație cu surse de încredere

Pentru a evalua dacă o știre este conformă cu informațiile disponibile în surse de încredere, am dezvoltat un algoritm care compară conținutul textului analizat cu articole de știri reale obținute de la publicații cunoscute, folosind API-ul de la NewsAPI.org.

Pașii algoritmului

1. **Normalizarea textului (opțională, pregătită pentru extindere)**

   Funcția normalize_text curăță textul de caractere non-alfanumerice și îl convertește în litere mici pentru a facilita compararea ulterioară. Deși nu este utilizată direct în versiunea actuală a codului, această funcție este pregătită pentru etape viitoare de preprocesare.

2. **Căutarea articolelor de încredere**

   Funcția search_trusted_news trimite o interogare către API-ul NewsAPI pentru a căuta articole relevante, în limba engleză, pe baza unui *query* (cuvânt-cheie sau expresie legată de subiectul știrii). Rezultatul este o listă de conținuturi (texte) ale articolelor găsite, care vor fi utilizate pentru comparație.

3. **Calcularea similarității semantice**

   Pentru a compara știrea analizată cu articolele de încredere, funcția compare_with_trusted_news folosește:

   - TfidfVectorizer: transformă textul într-o reprezentare numerică bazată pe frecvența termenilor (TF-IDF), eliminând cuvintele comune care nu oferă relevanță.
   - cosine_similarity: măsoară cât de similar este textul analizat cu fiecare articol de încredere (un scor între 0 și 1). Se extrage similaritatea maximă dintre textul analizat și sursele de referință.
   -

4. **Interpretarea rezultatului**

   Funcția notify_user afișează un mesaj interpretativ în funcție de pragul de similaritate ales (default: 0.7). Dacă similaritatea este ridicată, știrea este considerată compatibilă cu sursele de încredere. Dacă este sub prag, se semnalează existența unor diferențe semnificative, care pot indica informații neverificate sau false.

Analiza sursei și detectarea bias-ului

În ceea ce privește analiza sursei textului ne vom folosi de baza de date de la allsides, secțiunea de Media Bias. Această bază de date clasifică sute de site-uri web/ autori de articole pe axa politică stânga – dreapta, bazându-se pe inputul utilizatorilor și furnizând astfel și un nivel de încredere. Întrucât scopul mai restrâns al proiectului nu justifică accesul prin licență la versiunea up to date online a acestei baze de date, vom folosi o versiune offline stocată în fișierul csv allsides_data.

Sursa articolului este extrasă din url și apoi cu ajutorul librăriei difflib sunt căutați în baza de date termeni similari cu aceasta. Dacă este găsit un termen se furnizează date despre sursa articolului, plasarea acesteia pe axa politică, numărul de review-uri necesare pt clasificarea ei, etc.

Intefața aplicației

Interfața aplicației oferă utilizatorului două posibilități pentru încărcarea articolului, prin text sau prin url. Varianta prin text este una ce oferă mai puține date, întrucât sursa articolului nu poate fi analizată.

După analiza textului (ce poate dura câteva secunde) datele despre articol sunt afișate utilizatorului.

Ce rămâne de implementat

Până la această etapă am finalizat primele 6 cerințe din backlog, deci pentru a finaliza proiectul mai avem de implementat a doua jumătate a acestuia. Va trebui să îmbunătățim funcționalitatea interfeței pentru a permite utilizatorilor să dea feedback, să afle mai multe despre analiza unui anumit articol sau să acceseze alte articole de încredere pe același subiect.

De asemenea va trebui să implementăm și câteva acțiuni pentru administratori care vor putea ajusta parametrii algoritmilor, în funcție de feedbackul dat de utilizatori.

Bibliografie

- Fișierul cu frecvența apariției cuvintelor în engleză a fost preluat de pe Kaggle, de la Rachel Tatman, link: https://www.kaggle.com/datasets/rtatman/english-word-frequency?resource=download
- Versiunea offline a bazei de date allsides este preluată de pe github, https://github.com/favstats/AllSideR/tree/master/data
- Versiunea online a bazei de date, https://www.allsides.com/media-bias

- NewsAPI. (n.d.). *NewsAPI – Get breaking news headlines and search for articles from news sources and blogs*: https://newsapi.org/
- Scikit-learn Developers. (n.d.). *TfidfVectorizer – scikit-learn documentation*: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
- Scikit-learn Developers. (n.d.). *cosine_similarity – scikit-learn documentation*: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html
- Python Software Foundation. (n.d.). *re — Regular expression operations*: https://docs.python.org/3/library/re.html
- Requests Library. (n.d.). *Requests: HTTP for Humans*: https://docs.python-requests.org/en/latest/
- Tatman, R. (n.d.). *English Word Frequency*. Kaggle: https://www.kaggle.com/datasets/rtatman/english-word-frequency?resource=download