Gruparea fișierelor HTML după similaritate vizuală – Alegerea metodei

Contextul și definirea „similarității” din perspectiva utilizatorului

Task-ul propus cere gruparea documentelor HTML care sunt similare din perspectiva unui utilizator ce le deschide în browser. Cu alte cuvinte, paginile web trebuie considerate similare dacă arată asemănător utilizatorului, indiferent de diferențele din codul lor sursă. Primul pas a fost

definirea clară a similarității: ne-am întrebat „Ce înseamnă pentru un utilizator?”. Răspunsul

evident a fost aspectul vizual al paginii – layout-ul, conținutul vizibil și elementele grafice pe care utilizatorul le vede pe ecran. Astfel, am concluzionat că rezultatul randării (cum arată pagina în browser) trebuie să fie baza criteriului de grupare, mai degrabă decât codul HTML brut sau structura DOM.

Am considerat și scenariul invers: „Codul HTML mai contează dacă rezultatul vizual e complet diferit?”. Răspunsul a fost negativ. Două fișiere HTML pot avea cod sursă foarte asemănător (chiar același markup și structură DOM), dar dacă încarcă CSS diferit sau conținut dinamic diferit, pot arăta complet diferit vizual. Invers, pagini cu cod HTML total diferit (de exemplu generat de framework-uri diferite) pot arăta identic în browser. Acest fapt este subliniat și în literatură: structuri DOM identice pot produce output vizual diferit, iar DOM diferit poate genera pagini cu aspect identic . Pentru utilizator,

doar experiența vizuală contează, nu similaritatea la nivel de taguri sau text sursă.

Opțiuni considerate pentru măsurarea similarității paginilor

Pentru a decide metoda optimă de grupare, am analizat mai multe abordări posibile:

Analiza structurii DOM/HTML – Compararea arborelor DOM ale paginilor sau a codului HTML (de exemplu, similaritatea secvențelor de taguri sau a structurii nodurilor). Această abordare ar grupa paginile cu markup asemănător. Am constatat însă că este predispusă la erori: modificări minore în cod pot să nu afecteze deloc aspectul vizual, dar ar face paginile să pară diferite la o comparație de cod. De asemenea, cod aproape identic poate produce pagini cu stilistici diferite (ex. folosind fișiere CSS alternative). Un comparator DOM ar marca fals pagini ca diferite sau similare în mod eronat, deoarece diferențele/asemănările în cod nu garantează diferențe/ asemănări vizuale .

Compararea conținutului textual – Extracția textului din HTML (ignorând tagurile) și măsurarea similarității textuale (prin algoritmi de string similarity, shingling sau SimHash pe cuvinte). Aceasta ar identifica pagini cu același conținut textual. Totuși, două pagini pot conține aceleași texte dar aranjate diferit vizual (sau cu elemente media diferite), ceea ce utilizatorul ar percepe drept pagini distincte. Invers, pagini vizual similare (același șablon) pot avea text diferit. Prin urmare, doar textul nu era un indicator suficient al similarității vizuale.

Analiza imaginii randate (metodă vizuală) – Generarea unui screenshot al paginii și compararea imaginilor rezultate. Aceasta abordează direct perspectiva utilizatorului, întrucât compară efectiv cum arată paginile. Există mai multe moduri de a face comparația imaginilor:

Diferența pixel-cu-pixel: prea strictă – oricărei variații infime de pixeli (ex: un cursor clipind, un antialiasing diferit) i-ar corespunde un rezultat diferit” . Ar produce multe false negative (pagini practic identice vizual marcate ca diferite din cauza unui pixel).

Hash criptografic al imaginii (MD5/SHA): inutil în context – ar grupa doar imagini 100% identice bit cu bit, eșuând să recunoască pagini cu diferențe minore (dimensiuni, capturi la momente diferite, etc.).

Hash perceptual al imaginii (pHash sau similar) – metodă concepută să producă o amprentă

digitală a imaginii care nu se schimbă semnificativ la modificări minore și care este identică pentru imagini pe care omul le percepe ca fiind la fel. Astfel, două capturi de ecran aproape identice vizual vor avea hash perceptual egal sau foarte apropiat, permițând gruparea lor fiabilă.

Aceasta a devenit imediat candidadul principal, fiind o soluție testată pentru gruparea screenshot-urilor similare. De exemplu, în domeniul securității web s-a folosit cu succes pHash pentru a elimina duplicate de site-uri phishing: se calculează hash-ul perceptual al capturilor și se consideră duplicat orice pagină nouă al cărei hash este la distanță Hamming mică de un hash existent.

Computer Vision avansat (CNN pe imagine): utilizarea unei rețele neuronale convoluționale pre- antrenate pentru a extrage vectori de caracteristici vizuale din screenshot-uri, apoi gruparea pe baza acestor vectori. Această metodă poate surprinde caracteristici de nivel înalt (ex. recunoaște că două pagini au ambele un formular de login și scheme de culori similare chiar dacă pixeli diferă). Este o soluție potentă, însă complexă: necesită resurse de calcul sporite și eventual date de antrenament. Pentru scopul nostru – grupare pe baza aspectului – o rețea neurală ar fi overkill inițial. Simplul hash perceptual este mult mai ușor de implementat și a demonstrat eficiență suficientă în practică. (Notăm că pentru rafinament ulterior, o soluție hibridă cu CNN ar putea îmbunătăți cazurile unde hash-ul vizual eșuează, însă am decis că nu este necesară de la început.)

OCR (Reconoscarea Optică a Caracterelor) – extragerea textului din screenshot-uri și compararea paginilor pe baza textului vizibil (independent de stil). Această opțiune ar combina elemente de conținut și vizual, dar are dezavantaje: este complexă (necesită detectarea și citirea tuturor textelor din imagine) și, similar abordării textuale de mai sus, ar ignora componente vizuale non-text (imagini, layout). Am conchis că aspectul vizual global (nu doar textul) este esențial pentru percepția utilizatorului, deci OCR nu adaugă beneficii suficiente comparativ cu hash-ul imaginii în sine.

Justificarea alegerii metodei vizuale (hash pe screenshot)

După evaluarea opțiunilor, am ales soluția bazată pe similaritate vizuală prin hash-ul capturilor de ecran. Aceasta s-a dovedit cea mai aliniată cu perspectiva utilizatorului și cu cerința de a grupa pagini după cum arată, nu după cum sunt codificate intern. Principalele motive pentru această alegere sunt:

Fidelitate față de percepția utilizatorului: Metoda vizuală ia în considerare exact ceea ce vede utilizatorul. Astfel, pagini care arată la fel vor fi grupate împreună chiar dacă au cod HTML diferit (de exemplu, instanțe diferite ale aceluiași șablon sau conținut încărcat dinamic). În schimb, pagini care arată diferit nu vor fi confundate ca similare doar pentru că împart cod asemănător. Acest lucru elimină situațiile problematice în care compararea DOM ar rata sau falsifica similitudini (cod identic dar aspect diferit, sau vice-versa ).

Robustețe la variații minore: Folosind un algoritm de hash perceptual al imaginii, gruparea nu este afectată de variații triviale între două randări ale aceleiași pagini. De exemplu, diferențe

de rezoluție, formate de imagine sau mici decalaje de pixeli nu schimbă hash-ul perceptual semnificativ. Astfel, paginile cu diferențe minore vizual (cum ar fi un alt format de dată, un element dinamic sau un pixel diferit) vor fi în continuare recunoscute ca fiind aceeași grupă vizuală. Acest avantaj al hash-ului perceptual – de a produce aceeași amprentă pentru imagini

„identice pentru ochiul uman” – este exact cerința noastră de similaritate.

Simplicitate și eficiență: Implementarea este relativ simplă și eficientă. Putem folosi biblioteci

existente (de ex. în Python) pentru a genera hash-uri perceptuale. Compararea

hash-urilor (eventual după convertirea lor în valori binare și calculul distanței Hamming) este rapidă și ușor scalabilă. În practică, această metodă a demonstrat deja reducerea drastică a volumului de date de analizat prin gruparea automată a paginilor asemănătoare. De exemplu, un studiu de caz a arătat o reducere de ~75x a numărului de imagini ce trebuie revizuite manual după gruparea prin hash perceptual, comparativ cu lista negrupată. Astfel, ne așteptăm ca și în cazul nostru să eliminăm multe duplicatate sau variații redundante, grupându-le corect.

Rigoare tehnică: Deși este o abordare centrată pe utilizator, soluția este susținută de concepte solide de vision computing. Hash-urile perceptuale aplică transformări (redimensionare, conversie în scala de gri, transformate discrete) și filtrare de frecvențe pentru a extrage esența imaginii. Aceasta asigură un nivel de rigoare – metodologia nu este empirică sau aleatorie, ci se bazează pe algoritmi bine cunoscuți de comparare a imaginilor. În plus, folosirea hash-ului în loc de stocarea integrală a vectorilor de caracteristici sau a imaginilor întregi optimizează spațiul și timpul de comparare.

Pașii de gândire care au condus la soluție

Procesul decizional a parcurs următoarele etape logice:

Interpretarea cerinței – Am clarificat că „similar din perspectiva utilizatorului” = similaritate de aspect vizual. Această interpretare a ghidat direcția tehnică (orientare spre imagine și randare, nu spre cod).

Întrebări cheie – Ne-am pus întrebări precum „Utilizatorul vede codul sau rezultatul?” și „Pot două pagini cu cod diferit să pară la fel?”. Răspunsurile au evidențiat decuplarea dintre cod și vizual, confirmând că trebuie să comparăm output-ul (pagina randată).

Evaluarea abordărilor posibile – Am enumerat potențialele soluții (vezi secțiunea anterioară) și am analizat pentru fiecare în ce măsură îndeplinește criteriul de mai sus. Am ținut cont de avantaje, limitări și de complexitatea implementării fiecărei opțiuni.

Selectarea criteriului principal – Dintre opțiuni, compararea screenshot-urilor a reieșit ca

fiind cea mai directă și fiabilă metodă de a surprinde similaritatea reală a paginilor. Am decis ca acesta va fi modul de a măsura similaritatea între două pagini.

Alegerea algoritmului concret – Pentru a implementa compararea imaginilor, am comparat

variante: diferență de pixeli vs. hash vizual vs. metode AI. Ținând cont de cerința de a grupa pagini aproape identice vizual, am optat pentru hash-ul perceptual ca soluție echilibrată între acuratețe și simplitate. Această decizie a fost susținută atât de raționamentul teoretic (hash-ul perceptual minimizează falsele diferențe), cât și de exemple din practică și literatură care validează eficiența metodei.

Verificarea fezabilității – În final, ne-am asigurat că soluția este fezabilă în contextul dat

(disponibilitate biblioteci, volum de date suportabil, timp de procesare acceptabil). Cum componentele necesare (generator de screenshot, algoritm de hash perceptual) sunt standard și scalabile, alegerea a fost confirmată.

Concluzie

În concluzie, metoda de grupare bazată pe similaritatea vizuală (hash perceptual al capturilor de ecran) a fost aleasă deoarece satisface direct cerința de a identifica pagini web asemănătoare așa cum le percepe utilizatorul. Această abordare oferă un echilibru optim între acuratețe și robustețe, evitând capcanele analizelor la nivel de cod sau text. Soluția este prezentată într-un mod riguros dar accesibil, putând fi inclusă într-un README tehnic pentru evaluatori. Prin aplicarea acestei metode, ne așteptăm să grupăm automat fișierele HTML în seturi de pagini vizual similare, indiferent de variațiile minore de implementare, asigurând un rezultat conform cu perspectiva utilizatorului final.

Surse consultate: Implementări din industrie confirmă eficacitatea abordării prin hash vizual pentru pagini web, evidențiind avantajul că pagini cu diferențe de cod dar aspect identic sunt detectate ca similare , iar funcții de perceptual hashing pot grupa cu succes imagini de pagini aproape identice

vizual. Această metodă este utilizată, de exemplu, la deduplicarea site-urilor phishing cu layout identic dar URL sau cod diferit, ceea ce subliniază aplicabilitatea și fiabilitatea soluției alese în scenarii reale.