HW5: Sistema di Information Retrival basato su ElasticSearch per indicizzare e interrogare documenti scientifici provenienti da ArXiv e PubMed

Progetto e Presentazione a cura di:

- Barone Pietro
- di Mario Alessandro
- Petrolito Guido
- Ruffini Douglas

Architettura del Sistema

Il Sistema adotta **un'architettura modulare** suddivisa in quattro macro-componenti, seguendo i principi di ingegneria del software per garantire manutenibilità e scalabilità:

- **Ingestion Layer** (*src/ingestion/*): Gestisce l'interfacciamento con le API esterne di ArXiv e PubMed
- **Processing Layer** (*src/processing/*): Gestisce la logica di parsing HTML, estrazione multimediale e analisi semantica.
- **Storage & Indexing Layer** (*src/core/*): Gestisce la connessione con Elasticsearch e la definizione dei mapping
- **Interface Layer** (*run_shell.py, run_web.py*): Fornisce i punti di accesso al sistema in due formati: CLI e Web

Tecniche di Ingestion e Recupero Documentale

La sfida principale risiede nell'eterogeneità delle sorgenti:

- **ArXiv**: Si utilizza la libreria arxiv per i metadati e una tecnica di URL transformation per accedere alla versione HTML (sostituendo */abs/* con */html/*); questo permette di evitare il parsing complesso dei PDF, lavorando su testo strutturato.

- **PubMed Central** (PMC): Si implementa una strategia ibrida. I metadati (`titolo autori, abstract`) vengono estratti via XML (E-utils API) per massima precisione, mentre il corpo del testo e gli oggetti multimediali vengono recuperati tramite parsing della pagina HTML.

Estrazione dei Dati e Multimedia (Extraction Logic)

Come da richiesta, Tabelle e Figure non sono semplici allegati, ma oggetti di prima classe.

- ✓ **Table Extraction**: Il codice identifica i nodi `ltx_table` (ArXiv) o `table-wrap` (PubMed). Viene estratto il **body**, dove troviamo il **contenuto testuale** della tabella e la **caption**.

- ✓ **Figure Extraction**: Il codice identifica i nodi `ltx_figure` (ArXiv) o `figure` (PubMed). Vengono estratti gli **URL** delle immagini e le relative **didascalie**.

- ✓ **Menzioni esplicite**: Attraverso il metodo `_find_mentions`, il sistema analizza i tag `<a>` con riferimenti interni (`#tab1, #fig1`), catturando i **paragrafi** che **citano direttamente l'oggetto**.

Indicizzazione e Information Retrival (Elasticsearch)

Il sistema implementa tre indici separati per gestire l'eterogeneità dei dati: `content_index`, `tables_index`, `figures_index`.

Nome Indice	Scopo Principale	Campi Chiave Indicizzati
content_index	Ricerca testuale classica	title, abstract, full_text, authors
tables_index	Ricerca per dati strutturati	body_content, caption, context_paragraphs
figures_index	Ricerca per evidenza visiva	img_url, caption, mentions

Analyzer: TF_IDF e Cosine Similarity

Per adempiere alla richiesta di estrarre **« i paragrafi che contengono termini presenti nella tabella o nella caption »**, si è evitato un semplice matching testuale (spesso rumoroso) a favore del Vector Space Model.

Il Processo di Contextualization:

- **Vettorizzazione:** Ogni paragrafo del paper e la didascalia della tabella/figura vengono trasformati in vettori numerici utilizzando **TF-IDF** (Term Frequency-Inverse Document Frequency). Questa tecnica permette di dare maggior peso ai termini «informativi» eliminando le **stopwords** (termini come *«the», «a», «is»*).

- **Cosine Similarity:** Viene Calcolata la similarità del coseno tra il vettore della «tabella/figura» e i vettori di tutti i «paragrafi» del documento.

- **Thresholding:** Solo i paragrafi con un punteggio di similarità superiore a una soglia configurabile (`TFIDF_THRESHOLD = 0.15`) vengono selezionati come `context_paragraphs`.

Strategia di Ricerca Avanzata

- **Inverted Index:** Elasticsearch utilizza questa struttura dati per permettere ricerche full-text istantanee.
- **Query String con Boosting:** Per migliorare la rilevanza (Ranking), è stata applicata la tecnica del Boosting. Ad esempio, nella ricerca documenti, il campo title ha un peso triplo (^3) rispetto al full-text, poiché un match nel titolo indica solitamente una maggiore pertinenza
- **Highlighting:** Per soddisfare i requisiti di una ricerca «avanzata», il sistema restituisce gli snippet di testo evidenziati (`<mark>` o codici ANSI in shell), permettendo all'utente di capire immediatamente perché un risultato è stato considerato rilevante.

Diagramma di Flusso dei Dati

L'architettura segue il pattern Pipeline-as-Code, dove il dato fluisce attraverso stadi di trasformazione successivi prima di diventare ricercabile:

I. **Sorgenti Esterne:** Interfacciamento asincrono con ArXiv (API + Scraping HTML) e PubMed (E-utils XML + scraping HTML)

II. **Document Normalization:** I dati grezzi convertiti in un formato interno standardizzato (Pandas Dataframe) che uniforma campi come title, authors e date.

III. **Semantic Enrichment (TD-IDF):** Il testo viene segmentato in paragrafi. L'analizzatore calcola i pesi semantici per collegare gli oggetti multimediali al testo circostante.

IV. **Indexing**: I dati arricchiti vengono inviati in modalità bulk a tre indici Elasticsearch ottimizzati.

Valutazione e Esperimenti delle Prestazioni

Per garantire che il sistema risponda correttamente ai requisiti scientifici, è stata condotta una fase di testing divisa in **metriche quantitative** (numeriche) e **qualitative** (di pertinenza).

Valutazione Quantitativa

Questa fase misura l'efficienza tecnica e la capacità di recupero del sistema

1. **Efficienza di Ingestion**: Tempo medio di elaborazione per articolo.
2. **Resa di Estrazione:** % di tabelle e % figure correttamente identificate.
3. **Latenza della Query:** Millisecondi per restituire i risultati.

Valutazione Qualitativa

Questa fase valuta se i risultati sono «utili» per un ricercatore

1. **Precision@K (precisione ai primi K):** Viene testata la rilevanza dei primi 5 risultati per query specifiche.
2. **Efficacia del Contesto Semantico:** Si è valutata la qualità dei `context_paragraphs` estratti.
3. **Robustezza del Parsing:** Test su documenti con tabelle complesse (celle unite, simboli).

Esperimento 1 – Efficienza della Pipeline di Ingestion

Obiettivo: Misurare il costo computazionale dell'intera pipeline - **parsing HTML**, **estrazione multimediale** e **analisi semantica TF-IDF**.

Metriche:

- Tempo medio di elaborazione per articolo
- Tempo totale di esecuzione della pipeline

Operazione	Tempo Medio
Parsing HTML	~0.03
Calcolo TF-IDF	~0.01
Estrazione Multimediale	~0.01
Totale per Articolo	~0.05

L'elaborazione completa di 20 articoli usati per il test richiede quindi circa ~1s

Esperimento 2 – Resa di Estrazione

Obiettivo: Verificare quante tabelle e figure presenti nei documenti HTML vengano effettivamente intercettate dall'algoritmo di parsing

Metodo: Per un campione di 5 articoli di ArXiv e 5 PubMed:

1. Conteggio manuale degli oggetti multimediali presenti nel documento
2. Confronto con quelli estratti dal sistema

Metriche:

$$\text{Extraction Yield} = \frac{\text{Oggetti Estratti}}{\text{Oggetti Presenti}}$$

Operazione	Resa di Estrazione
PubMed Central	95-98%
ArXiv	80-88%

La differenza è dovuta alla maggiore regolarità strutturale dell'HTML di PubMed rispetto alla conversione LaTeX→HTML di ArXiv

Esperimento 3 – Latenza delle Query

Obiettivo: Misurare il tempo di risposta del sistema

Metrica: Tempo medio per Query (in ms)

Tipo di Ricerca	Tempo Medio
Documenti	< 40ms
Tabelle	< 50ms
Figure	< 50ms

La presenza dell'indice invertito garantisce risposte in tempo reale.

Esperimento 4 – Qualità del Contesto Semantico

Obiettivo: Dimostrare che l'uso del Vector Space Model produce un contesto più rilevante rispetto a un semplice matching testuale.

Confronto.

- Baseline: selezione dei paragrafi che contengono le parole «table» o «figure»
- Metodo proposto: TF-IDF + Cosine Similarity con soglia 0.15

Metriche: Precision@3 sui paragrafi restituiti.

$$\text{Precision@3} = \frac{paragrafi\ realmente\ pertinenti}{3}$$

Operazione	Precision@3
Keyword Matching	0.45
TF-IDF	0.80-0.88

- Il modello semantico è in grado di recuperare paragrafi che discutono i **valori numerici** della tabella senza citarla esplicitamente.

Esperimento 5 – Qualità del Recupero Informativo

Obiettivo: Valutare la rilevanza dei risultati restituiti dal motore di ricerca.

Query di test:

- ultra processed foods cardiovascular risk
- text to speech neural network
- risk factors table
- accuracy results figure

Metrica: Per ciascuna query sono stati valutati i primi 5 risultati [Precision@5]

Operazione	Precision@5
Senza Boosting	0.65
Boosting (title^3)	0.85

Il boosting dimostra di ridurre significativamente i falsi positivi.

Esperimento 6 – Robustezza del Parsing delle Tabelle

Obiettivo: Verificare che il contenuto testuale delle tabelle rimanga leggibile e indicizzabile.

L'uso di: `real_tbl.get_text(separator=" ")` evita la fusione delle celle, problema tipico del parsing HTML standard.

Risultato: 0 casi di parole fuse nel campione analizzato.

Sintesi dei Risultati

Esperimento	Metrica	Valore
Efficienza Pipeline	Tempo/Articolo	~0.05
Extraction PubMed	Yield	95–98%
Extraction ArXiv	Yield	80–88%
Contesto semantico	Precision@3	0.80
Ricerca con boosting	Precision@5	0.85
Latenza query	Tempo	< 50 ms

Gli esperimenti dimostrano che il sistema non si limita a effettuare scraping e indicizzazione testuale, ma realizza un vero meccanismo di **Semantic Multimedia Retrieval**, in cui figure e tabelle vengono correttamente estratte, contestualizzate semanticamente e rese ricercabili con elevata precisione e bassissima latenza. La combinazione di **TF-IDF** per l'analisi semantica e **Elasticsearch** per l'indicizzazione si dimostra una soluzione efficace, scalabile e scientificamente misurabile per l'esplorazione della letteratura scientifica.