Unified Hybrid NLP System – Detailed Workflow

Step 1: User Input Interface

Goal: Accept natural language bibliometric queries (e.g., “Summarize research trends in urban mobility using c
Tools: Chatbot UI (Streamlit, Gradio, or React + Flask), Optional: NER/Intent Detection using spaCy or Transf

Step 2: Literature Retrieval & Data Collection

Goal: Collect high-quality metadata + full-text papers
Sources: OpenAlex, Semantic Scholar, CORE, ArXiv, IEEE/Scopus
Tools: requests, openalex, pandas, wget, GROBID, pdfminer.six

Step 3: Preprocessing & Metadata Enrichment

Goal: Prepare data for analysis and storage
What to do: Extract metadata, clean text, enrich using NLP
Tools: KeyBERT, BERTopic, SciSpacy

Step 4: Knowledge Graph Construction

Goal: Build a Neo4j knowledge graph of literature relationships
Entities: Papers, Authors, Institutions, Topics, Venues, Citations
Tools: Neo4j, py2neo, Neo4j Bloom, D3.js, Gephi

Step 5: Semantic Embeddings + Vector DB

Goal: Enable semantic search using vector similarity
Steps: Use sentence-transformers (MiniLM), store in FAISS/Pinecone
Query: Convert query to embedding $\rightarrow$ cosine similarity $\rightarrow$ top-N results

Step 6: Clustering for Bibliometric Insight

Goal: Group papers by semantic or bibliometric similarity
Steps: GMM or HDBSCAN on embeddings, label clusters
Tools: scikit-learn, hdbscan, matplotlib, plotly

Step 7: Hybrid Text Generation (BERT + GPT)

Goal: Generate high-quality bibliometric summaries
Architecture: BERT Encoder + GPT Decoder (per hybrid transformer paper)
Datasets: ScisummNet, PubMed summaries, ArXiv summaries

Step 8: Bibliometric Metrics & Insights

Goal: Provide structured analytics + graphs
Metrics: h-index, g-index, citation count, PageRank, keyword evolution
Tools: pandas, matplotlib, networkx, pyvis, seaborn, plotly

Step 9: Result Presentation

Formats: Natural language (chatbot), interactive graphs, exportable PDF/CSV

Tools: Flask, Streamlit, Gradio, ReportLab, WeasyPrint, DataTables.js