**TextLab — Technical Documentation**

Detailed architecture and developer notes for app.py, Document.py, Word.py, and Summary.py.

---

**1) System Overview**

\*\*Goal:\*\* Convert and consolidate large sets of evaluation PDFs, then enable keyword analytics, topic modeling, semantic search, and summarization — all locally.

\*\*Core modules\*\*

• \*\*app.py\*\*: Bulk PDF downloader; PDF→TXT conversion; consolidation by user‑entered headings; region/year/country mapping from source columns; session state persistence.

• \*\*Word.py\*\*: Keyword Analyzer — exact and semantic matching; counts per document; aggregations; validation snippets.

• \*\*Document.py\*\*: Topic modeling (BERTopic / NMF / LDA) with safe parameter clamps; semantic diagnostics (centroids, heatmaps, semantic search).

• \*\*Summary.py\*\*: Noise‑aware summaries (removes TOC/headers/indices); evaluation‑focused chat grounded only in consolidated data.

\*\*Data flow\*\*

CSV/Excel links → app.py (download PDFs) → app.py (PDF→TXT)

TXT + user headings + meta columns (year/region/country) → consolidated\_df (in session)

consolidated\_df → Word.py / Document.py / Summary.py

\*\*Session keys\*\*

• st.session\_state['consolidated\_df'] (primary) or 'consolidated' (fallback): main dataframe used across pages.

• Optional: st.session\_state['region\_overrides\_by\_file'], 'region\_overrides\_by\_country' for assignments.

---

**2) Dependencies**

• \*\*Core\*\*: streamlit, pandas, numpy, scikit‑learn, altair, requests

• \*\*PDF extraction\*\*: pymupdf (preferred), pdfplumber, PyPDF2

• \*\*Language\*\*: langdetect, argostranslate (optional offline translation)

• \*\*Semantic\*\*: sentence‑transformers (SBERT), rank‑bm25, rapidfuzz (optional)

• \*\*Topics\*\*: bertopic, umap‑learn, hdbscan

> All algorithms have graceful fallbacks. If SBERT/BERTopic are absent, the app uses TF‑IDF/NMF.

---

**3) app.py — Implementation Notes**

\*\*Bulk downloader\*\*

• Robust filename sniffing (Content‑Disposition; RFC 5987; URL fallback) and direct‑download URL transforms (Drive/Dropbox/SharePoint/Box/GitHub).

• HTML landing page sniffing for embedded PDF URLs; one‑hop follow.

• ThreadPoolExecutor for parallel downloads with backoff & progress.

\*\*PDF→TXT\*\*

• Backend order: PyMuPDF → pdfplumber → PyPDF2.

• Writes .txt to temp folder; de‑duplicates filenames.

\*\*Consolidation\*\*

• Headings are user‑entered per run; regex matching extracts section text.

• Country/year/region mapped from source columns (not guessed), with optional manual region override.

• Cleaning options: lowercase, newline stripping, truncation per section; auto‑translate to EN via Argos when available.

• Consolidated CSV saved in temp dir and st.session\_state['consolidated\_df'].

\*\*Performance\*\*

• Prefer PyMuPDF; chunk size 256KB; max retries 2; progress UI; ZIP option.

---

**4) Word.py — Keyword Analyzer**

• Input: any text column from consolidated\_df (e.g., findings or full\_text).

• Modes:

- Exact (accent‑insensitive, hyphen→space, whole‑word option)

- AI/Semantic: SBERT (preferred) → RapidFuzz → simple inflections fallback

• Semantic vocab built from the corpus (1‑3‑grams) with min\_count and max\_vocab to curb noise.

• Outputs:

- Per‑document counts for each keyword + total\_count

- Validation snippets: sentence with match ±1 sentence

- Aggregations by Year and Region; Altair bars; downloads for detailed and aggregate CSVs

---

**5) Document.py — Topics & Semantics**

\*\*Model selection\*\*

• Default BERTopic (SBERT) with size‑aware UMAP/HDBSCAN and retries.

• Automatic fallback to NMF (TF‑IDF) or LDA (Count) when corpora are tiny or embeddings unavailable.

\*\*Safety clamps\*\*

• safe\_df\_cutoffs(n\_items, min\_df\_int, max\_df\_pct) ensures 0 < min\_df < max\_df ≤ 1 for vectorizers, preventing max\_df=1.001/min\_df≥max\_df errors.

• HDBSCAN parameters adapt to corpus size (min\_cluster\_size, min\_samples, n\_neighbors).

\*\*Scopes\*\*

• Whole documents — treat each doc as one item.

• Keyword‑centered — extract sentence‑window snippets around user keywords and model those.

\*\*Max features (vocabulary cap)\*\*

• Definition: upper limit on vocabulary size after filtering; controls matrix width in NMF/LDA.

• When to increase: topics feel generic; using 3‑grams; multilingual corpus; higher k.

• When to decrease: memory/slow runs; noisy topics (combine with higher min\_df or lower max\_df).

• Practical presets: small (≤200 items): 5k–12k; medium (200–2k): 10k–20k; large (2k+ or 1–3‑grams): 20k–40k.

• Note: Current slider applies to NMF/LDA. BERTopic uses its own CountVectorizer; we can wire the same cap there if needed.

\*\*Diagnostics\*\*

• Topic semantic quality: cosine to centroid (mean/median/std) and top representative documents.

• Topic×Topic similarity heatmap (cosine of centroids).

• Semantic search (SBERT→TF‑IDF fallback).

---

**6) Summary.py — Summaries & Private Chat**

\*\*Summaries\*\*

• Three modes: overall corpus, evaluation findings only, keyword‑focused.

• Pre‑clean text to remove TOC/index/heading noise (roman numerals, page numbers, dotted leaders, etc.).

\*\*Chatbot\*\*

• Retrieval: BM25 + TF‑IDF hybrid; optional SBERT rerank.

• Scoping filters by year/region/country.

• Refuses answers if similarity below a user threshold to avoid hallucinations; only uses local data.

---

**7) Data Structures**

\*\*consolidated\_df (columns)\*\*

• filename, country, year, unicef\_region (or region)

• One or more user‑named section columns (e.g., Findings of the evaluation, Conclusions, etc.)

• Optional full\_text if included

\*\*keyword analyzer output\*\*

• One row per document: filename, year, region, count per keyword, total\_count and a validation\_snippets column.

\*\*topics output\*\*

• topics\_df: topic, name, count

• doc\_table: filename, year, region, text, clean, topic, topic\_label, topic\_confidence

---

**8) Extensibility**

• OCR fallback: integrate Tesseract/ocrmypdf when extract\_text returns empty.

• More languages: plug spaCy or fastText lang detect; add Argos models.

• Custom headings: load/save heading presets per project.

• Model registry: allow pluggable embedding models (e5-small, mpnet, etc.).

---

**9) Testing & QA**

• Unit tests for: URL transform, filename parsing, section extraction regex, safe\_df\_cutoffs, cleaning (TOC removal), BM25+TF‑IDF retrieval.

• Golden sample set: 5–10 PDFs with known headings and multilingual content.

---

**10) Security & Privacy**

• No network calls during analysis steps; downloads are explicit via user input.

• All exports are local CSV/ZIP; admins should define retention policies.

---

**11) Known Limitations**

• Scanned PDFs need OCR.

• BERTopic quality depends on corpus size and domain homogeneity.

• Semantic expansion can over‑match if threshold is too low; expose threshold in UI (already done).

---

**12) Release & Operations**

• Pin critical libs in requirements.txt for reproducibility.

• Use streamlit cache clear if UI shows stale state.

• For speed on large corpora, consider CPU‑optimized wheels or GPU (PyTorch) for SBERT.