**TextLab — Technical Documentation**

Architecture and developer notes for app.py, Document.py, Word.py, and Summary.py.

---

**System Overview**

Goal: Convert and consolidate evaluation PDFs, then enable keyword analytics, topic modeling, semantic search, and summarization — locally.

Core modules

• app.py: downloader; PDF→TXT; consolidation by user headings; region/year/country mapping; session persistence.

• Word.py: Keyword Analyzer — exact & semantic; counts; aggregations; validation snippets.

• Document.py: Topic modeling (BERTopic/NMF/LDA); semantic diagnostics; search.

• Summary.py: Noise-aware summaries; private retrieval chat.

Data flow

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

TXT + user headings + meta columns → consolidated\_df (session)

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

Session keys

• consolidated\_df (primary) or consolidated (fallback).

---

**Dependencies**

Core: streamlit, pandas, numpy, scikit-learn, altair, requests

PDF: pymupdf, pdfplumber, PyPDF2

Language: langdetect, argostranslate (optional)

Semantic: sentence-transformers, rank-bm25, rapidfuzz

Topics: bertopic, umap-learn, hdbscan

---

**app.py**

Downloader: filename sniffing (Content-Disposition, RFC 5987), direct-download transforms (Drive/Dropbox/SharePoint/Box/GitHub), landing-page sniffing, threads with backoff.

PDF→TXT: PyMuPDF → pdfplumber → PyPDF2; dedup filenames.

Consolidation: user-entered regex headings; meta mapping; region override; cleaning/truncation; Argos translate; CSV export + session state.

Performance: PyMuPDF preferred; 256KB chunks; retries; ZIP option.

---

**Word.py**

Input: any text column from consolidated\_df.

Exact: accent-insensitive, hyphen→space, whole-word option.

AI/Semantic: SBERT → RapidFuzz → inflection fallback.

Semantic vocab: 1–3-grams with min\_count/max\_vocab.

Outputs: per-doc counts, validation snippets (sentence ±1), aggregates by year/region; charts; CSV downloads.

---

**Document.py**

Model selection: BERTopic (default) with size-aware UMAP/HDBSCAN; fallback NMF/LDA.

Safety clamps: ensure 0 < min\_df < max\_df ≤ 1 for all vectorizers; adapt neighbors/cluster sizes to corpus; retries then fallback.

Scopes: Whole documents or Keyword-centered (sentence windows).

Diagnostics: centroid cosine quality, representative docs, topic similarity heatmap, semantic search.

---

**Summary.py**

Summaries: overall / findings / keyword-focused with TOC/heading noise removal.

Chatbot: BM25 + TF-IDF; optional SBERT; thresholded refusal; filters by year/region/country; local-only data.

---

**Extensibility**

• OCR fallback (Tesseract).

• Heading presets per project.

• Embedding model registry.

• GPU acceleration and docx/pptx export pipelines.