**TextLab — User Manual**

**1) What is TextLab?**

TextLab turns large sets of PDFs into analyzable text, lets you consolidate sections with YOUR headings, and then analyze with keywords, topics, semantic search, and summaries — \*\*locally\*\*.

**2) Quick Start**

1. Open \*\*app.py\*\* (Streamlit main page).

2. \*\*Download PDFs\*\* from a CSV/Excel of links (choose link column).

3. \*\*Convert to TXT\*\* (uses best PDF extractor available).

4. \*\*Enter your headings\*\* (e.g., “Findings of the evaluation”) and \*\*map year/country/region\*\* from your data columns.

5. Click \*\*Consolidate\*\* to create a session-wide `consolidated\_df` used by other pages.

6. Explore \*\*Word\*\*, \*\*Document\*\*, and \*\*Summary\*\* pages.

**3) app.py — Download, Convert, Consolidate**

**Download**

• Upload CSV/XLSX, pick link column, set options (threads, timeout, direct-download fixes), and click \*\*Download\*\*.

• You will get a results table and optional ZIP of PDFs.

**Convert PDF→TXT**

• Click \*\*Convert to TXT\*\* after selecting PDFs.

• Uses PyMuPDF → pdfplumber → PyPDF2 in that order.

**Consolidate by User Headings**

• Enter \*your\* section headings (they vary by corpus).

• Map \*\*year / country / region\*\* from your sheet columns.

• Options: lowercase, strip newlines, truncate, \*\*auto-translate to English\*\* (Argos offline).

• Output: `consolidated\_df` saved in session + CSV in temp folder.

**Performance notes**

• Prefer PyMuPDF; increase threads gradually.

• For scanned PDFs, consider adding OCR later (Tesseract).

**4) Word.py — Keyword Analyzer**

**Choose inputs**

• Pick the \*\*text column\*\* (e.g., Findings or full\_text).

• Enter \*\*keywords\*\* (comma or newline separated).

**Match types**

• \*\*Exact\*\*: accent-insensitive, hyphen→space, optional whole-word boundary.

• \*\*AI/Semantic\*\*: expands keywords using SBERT (if available) → RapidFuzz → simple variants.

- Controls: \*\*Semantic vocab n-grams\*\*, \*\*Min phrase count\*\*, \*\*Max expansions\*\*, \*\*Similarity threshold\*\*.

**Outputs**

• \*\*Per-document counts\*\* with `total\_count` and \*\*validation\_snippets\*\* (sentence ±1 around each match).

• Aggregations by \*\*Year\*\* and \*\*Region\*\* with charts.

• Download CSVs for detailed and aggregate counts.

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

**Scopes**

• \*\*Whole documents\*\*: model each doc as a unit.

• \*\*Keyword-centered\*\*: collect sentence windows around your keywords, then model those.

**Models**

• \*\*BERTopic (SBERT)\*\* by default, with size-aware UMAP/HDBSCAN and retries.

• Automatic fallback to \*\*NMF (TF-IDF)\*\* or \*\*LDA (Count)\*\* for small corpora or no embeddings.

**Key controls**

• \*\*k\*\* topics (or reduction target for BERTopic)

• \*\*n-grams\*\* (phrases)

• \*\*min\_df / max\_df\*\* — TextLab clamps to ensure `min\_df < max\_df ≤ 1.0`

• \*\*max\_features\*\* — vocabulary cap for speed

• \*\*BM25 prefilter\*\* (optional) to \*\*focus the corpus\*\* using your keywords and keep Top‑N items

**Diagnostics & Search**

• Topic \*\*semantic quality\*\* (mean cosine-to-centroid).

• \*\*Top representative docs\*\* per topic.

• \*\*Topic×Topic similarity\*\* heatmap.

• \*\*Semantic search\*\* over the corpus.

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

**Summaries**

• \*\*All documents\*\*, \*\*Findings-only\*\*, or \*\*Keyword-focused\*\* summaries.

• Cleans noise first (TOC, dotted leaders, page numbers, Roman numerals, boilerplate headers).

**Chat**

• Private Q&A grounded \*\*only\*\* in local `consolidated\_df`.

• Hybrid retrieval (BM25 + TF‑IDF, optional SBERT rerank).

• Filters by \*\*year / region / country\*\*; similarity gate refuses low-confidence answers.

**7) Tips & Troubleshooting**

• \*\*“max\_df must be ≤ 1.0”\*\* — already clamped internally; if custom code, set `max\_df=0.95` and `min\_df<max\_df`.

• \*\*“Too few items for BERTopic”\*\* — the page auto-adjusts neighbors/cluster sizes and falls back to NMF.

• \*\*No hits in Keyword Analyzer\*\* — lower semantic threshold (0.45), add synonyms, switch to full\_text.

• \*\*Counts vs Snippets\*\* — `total\_count` = all occurrences; `snippet\_count` = number of sentence-windows captured (deduped per doc).

**8) Glossary**

• \*\*BM25\*\*: classic lexical ranking (like a smart TF‑IDF) used for prefilter and search.

• \*\*Embeddings (SBERT)\*\*: numeric vectors that represent meaning; used for semantic matches and BERTopic.

• \*\*LDA/NMF\*\*: classic topic models (bag‑of‑words) — interpretable word lists.

• \*\*TOC\*\*: Table of Contents; removed to reduce noise.

**9) Data & Privacy**

• Everything runs locally; no external API calls during analysis.

• Exports: CSV/ZIP; manage retention per your policy.

**10) Workflow Checklist**

• Download PDFs → Convert to TXT → Enter headings + map year/country/region → Consolidate

• Run Keyword Analyzer (Exact vs Semantic) → Validate with snippets → Export

• Run Document Topics (BERTopic/NMF/LDA) → Inspect diagnostics → Export

• Create Summaries → Use chat for Q&A with filters