DocInsight — Software Requirements Specification (SRS) v0.2

Project title  
DocInsight — Semantic + Stylometric Document Intelligence

1. Executive summary (short & concrete)  
DocInsight is a focused, research-driven prototype that detects semantic paraphrase and authorship/style anomalies in academic documents. Unlike generic lexical-match systems, DocInsight combines domain-adapted sentence embeddings, a two-stage retrieval + reranker, and a stylometric evidence ensemble to deliver sentence-level, explainable flags. The project is engineered to produce measurable improvements over baseline semantic-only systems and to generate reproducible evaluation results suitable for a conference submission.

2. Problem statement — Where current tools (e.g., Turnitin) fall short  
- Lexical / n-gram dependence: Many commercial detectors rely heavily on string matching and chunk overlap; this misses paraphrased or semantically equivalent content.  
- Coarse granularity: Most systems return document-level or chunk-level percentages without sentence-level rationales or source snippets that a reviewer can verify.  
- Weak semantic & paraphrase sensitivity: Off-the-shelf lexical matching produces low recall for reworded content produced by paraphrase tools or LLMs.  
- Limited stylometry/AI-detection integration: Stylometric signals (writing-style shifts, function-word patterns) are rarely integrated as corroborating evidence; AI-detection modules often give coarse or unreliable scores.  
- Explainability & reviewer trust: Black-box scores reduce reviewer confidence and make it hard to act on flagged items.

These limitations create real academic risk: paraphrased or AI-assisted submissions can pass lexical checks while still being academically dishonest.

3. Our concrete contributions — what DocInsight will do differently (and measurably)  
1. Domain-adapted semantic embeddings — fine-tune SBERT on an academic paraphrase curriculum (PAWS + Quora + synthetic adversarial paraphrases) so embeddings capture paraphrase signals typical for essays and assignments.  
2. Two-stage retrieval + reranking — use SBERT + FAISS for fast candidate retrieval, then apply a cross-encoder (BERT/SRoBERTa) to rerank top-k candidates for higher precision (reduces false positives).  
3. Stylometric ensemble evidence — extract per-sentence/section stylometric features (TextStat, POS distributions, function-word frequencies, TTR) and train an ensemble that signals anomalous sections; fuse with semantic score using a calibrated logistic model.  
4. Adversarial augmentation & robustness training — generate paraphrase adversaries (back-translation, LLM paraphrase loop) and include them in fine-tuning to improve robustness to paraphrase attacks.  
5. Sentence-level explainability & provenance — each flagged sentence will include: matched source snippet(s), semantic score, reranker score, stylometry note, and a short template rationale the reviewer can read in <10s.  
6. Citation & common-knowledge handling — rule-based filters and context-aware down-weighting ensure citations and commonplace facts don’t generate false positives.  
7. Public mini-benchmark — we will create and publish a small academic paraphrase benchmark (synthesized + human-curated) to quantify improvements and for reproducibility.

4. Scope   
In scope:  
- Parser for .pdf and .docx.  
- Sentence-level semantic search pipeline: SBERT (fine-tuneable) → FAISS → cross-encoder reranker.  
- Stylometry feature extractor + anomaly scorer (trainable).  
- Evidence fusion engine + explainable report (PDF/JSON).  
- Streamlit demo UI showing end-to-end output and examples for the panel.

Out of scope:  
- Building or matching Turnitin’s enterprise corpus.  
- Large-scale productionization or marketplace deployment.  
- Full forensic/authentication features beyond stylometry.

5. System design — exact components & how they interact (concrete)  
1. Parsing & preprocessing  
 - Tools: PyMuPDF (pdf), python-docx/docx2txt (docx), spaCy/NLTK for sentence splitting.  
 - Preprocessing: Remove / tag reference section (detect “References”, “Bibliography”), normalize citations, strip boilerplate.  
2. Embedding generator (bi-encoder)  
 - Base: sentence-transformers/paraphrase-mpnet-base-v2 or all-MiniLM-L6-v2 (dev chooses trade-off).  
 - Fine-tuning: contrastive or MultipleNegativeRanking loss on curated paraphrase pairs (3k–10k).  
 - Output: 384–768 dimensional L2-normalized vectors.  
3. Fast retrieval (FAISS)  
 - Index: IndexFlatIP or IVFFlat (for small/medium demo).  
 - Retrieve: top-k candidates per sentence (k=5 default).  
4. Cross-encoder reranker  
 - Model: bert-base-uncased cross-encoder (sentence-pair classifier).  
 - Function: rerank top-k by cross-encoder score to improve precision for top result.  
5. Stylometry extractor & classifier  
 - Features: avg sentence length, sentence-length variance, Flesch reading ease, TTR, function-word frequencies, POS tag distribution ratios, punctuation density.  
 - Model: lightweight classifier (RandomForest / XGBoost) or anomaly detector (isolation forest) trained on human vs LLM sections and intra-doc shifts.  
6. Evidence fusion engine  
 - Input: semantic cosine, cross-encoder score, stylometry anomaly score, citation-flag boolean, lexical-overlap ratio.  
 - Fusion: calibrated logistic regressor (interpretable weights) that outputs an evidence score and a recommended action (info/flag/high-confidence).  
7. Report generator  
 - Outputs: PDF (highlighted sentences + matched snippet side-by-side), JSON (machine-readable), and Streamlit interactive view.

6. Training & data plan (specific and practical)  
Base datasets: PAWS, Quora QP, public paraphrase datasets.  
Synthetic augmentation: use back-translation (e.g., EN→FR→EN) and an LLM prompt loop to create multiple paraphrases per sentence (aim for ~5 paraphrases per seed sentence).  
Stylometry training data: generate LLM outputs for prompt templates similar to student assignments; collect or synthesize human-written essays (public domain or anonymized examples).  
Fine-tuning targets & sizes:  
- SBERT fine-tune: start with 3k–10k paraphrase pairs, 3–5 epochs (monitor dev F1).  
- Cross-encoder: train on top-k pairs labeled paraphrase/non-paraphrase (1k–5k pairs).  
- Stylometry classifier: train on 5k–10k labeled sections if available (can augment synthetically).

Compute: fine-tuning possible on Colab/GPU for prototyping; use compact models to keep training time short (1–3 hours typical for small sets).

7. Evaluation & success criteria (concrete targets)  
- Paraphrase detection (per-sentence): target F1 = 0.78–0.85 on held-out academic-paraphrase set (we will report precision at 0.85 recall points).  
- Stylometry anomaly detection: ROC-AUC target = 0.72–0.85 for section-level detection of LLM vs human and intra-doc shifts.  
- Explainability usability: human reviewer agreement ≥ 0.7 on flagged items in a pilot (5–10 graders).  
- False-positive control: keep false-flag rate on common-knowledge sentences < 5% (via whitelist and scoring weights).

We will report confidence intervals and significance tests when comparing against baselines.

9. Implementation guidance (practical tips)  
- SBERT choice: start with all-MiniLM-L6-v2 for speed; switch to paraphrase-mpnet-base-v2 when fine-tuning quality is needed.  
- Reranking policy: only run cross-encoder on top-5 retrieved candidates to keep demo latency acceptable.  
- Stylometry thresholds: initial heuristics (e.g., variance jump > 30% across sections) are acceptable for demo; replace heuristics with trained classifier later.  
- Citation handling: detect reference section headers and ignore these blocks; inline citation tokens (e.g., (Smith, 2020)) are removed or down-weighted before embedding.  
- Deployment for demo: Streamlit on local machine or Colab; export report as PDF for panel use.

10. Risks, mitigations & ethical notes  
- Risk: Overfitting to synthetic paraphrases — mitigate with varied augmentation, and keep human-curated validation set.  
- Risk: Bias against non-native writing — mitigate by including diverse human essays in training and using stylometry only as supporting evidence.  
- Ethical note: DocInsight provides evidence, not final judgment. Human reviewers retain authority and can override system flags.

11. References (select — for slide citation)  
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT.  
- Google Research. PAWS (Paraphrase Adversaries).  
- Johnson, J., et al. (2021). FAISS: Billion-scale similarity search.  
- Recent stylometry and LLM-detection literature (2023–2025).