TEAM GIMP

Challenge 1B: Methodology Explanation

Approach Overview

Our solution implements a **persona-driven document intelligence system** that extracts and ranks relevant sections from PDF collections based on specific user roles and tasks. The approach combines multilingual natural language processing with semantic similarity scoring to deliver targeted content extraction.

Core Algorithm

1. Dynamic Collection Processing

The system automatically discovers and processes any number of document collections by scanning input directories for valid structures (challenge1b_input.json + PDFs/ folder). This dynamic approach eliminates hardcoded limitations and supports scalable document analysis.

2. Multilingual Language Detection

We implement a multi-stage language detection algorithm that identifies non-Latin scripts (Chinese, Arabic, Japanese) using Unicode ranges, detects Latin-based languages through diacritical marks (Spanish: ñ, á; French: à, ç; German: ä, ü), and defaults to English for ambiguous cases.

3. Intelligent Section Identification

Our section heading detection combines **font analysis** (bold text, larger sizes), **persona patterns** (language-specific regex matching), **general patterns** (numbered sections, title case), and **context awareness** for validation.

4. Relevance Scoring Engine

We employ a four-component scoring system:

- **Semantic Similarity (50%)**: BERT embeddings with cosine similarity between section titles and persona/job descriptions
- **Keyword Matching (25%)**: Persona-specific keywords with language-adapted sets
- **Pattern Matching (25%)**: Regex pattern matching for persona-specific content
- **Content Quality Assessment**: Optional DistilBART summarization for context validation

5. Post-Processing Pipeline

Smart deduplication normalizes titles and removes near-duplicates, ranking sorts sections by relevance scores, and content extraction retrieves 500-800 characters of contextual content following each heading.

Technical Implementation

The solution uses a **dual-model architecture** within the 1GB constraint:

- **BERT-tiny (34MB)**: Embedding generation for semantic similarity
- **DistilBART (600MB)**: Content summarization and quality assessment

Performance optimizations include pre-computed template embeddings, efficient batch processing, and early termination for low-quality candidates.

Docker Execution Steps

Build the Container

docker build --platform linux/amd64 -t challenge1b-multilingual-processor .

Run the Container

docker run --rm -v ${PWD}/input:/app/input:ro -v ${PWD}/output:/app/output --network none challenge1b-multilingual-processor

Key Innovations

1. **Language-Adaptive Processing**: Automatically adjusts patterns based on detected document language
2. **Persona-Specific Extraction**: Tailors content identification to user roles
3. **Dynamic Scalability**: Processes any number of collections without code modifications
4. **Offline Operation**: Uses locally cached models for secure, network-independent processing