THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Data Science Program

Capstone Report - Fall 2025

Automating Military Knowledge, Skill, and Ability Extraction

Kyle Hall

supervised by
Amir Jafari

Abstract

Air Force Specialty Codes (AFSCs) define competency requirements for over 300 military occupations, but these descriptions exist as unstructured text across lengthy PDF documents, preventing systematic analysis and comparison with civilian workforce frameworks. This project develops an automated extraction pipeline that transforms AFSC documentation into structured Knowledge, Skills, and Abilities (KSAs) suitable for workforce analytics and career transition planning. The system integrates LAiSER (Leveraging AI for Skills Extraction & Research), a skill extraction framework developed at George Washington University, with multi-provider large language model enhancement (OpenAI GPT-4o-mini, Anthropic Claude Sonnet 4.5, Google Gemini 2.0 Flash, HuggingFace Llama-3.2-3B-Instruct) to generate comprehensive competency profiles. Extracted items undergo semantic deduplication using hybrid text similarity, taxonomy alignment to the Open Skills Network (OSN) and ESCO standards, and persistence in a Neo4j graph database enabling cross-specialty queries. The complete system processes 12 AFSCs across intelligence, operations, and maintenance career fields, extracting 253 unique KSAs with an average of ~21 items per specialty and achieving ~18% alignment to standardized skill taxonomies. A production Streamlit web application provides interactive exploration, bring-your-own-API demonstration capabilities, comprehensive documentation, and administrative data management. This work establishes technical feasibility for automated military competency extraction at scale and provides a foundation for military-to-civilian skill translation, curriculum development, and workforce planning applications.

1. Introduction

The United States Air Force relies on Air Force Specialty Codes (AFSCs) to define the roles, responsibilities, and competency expectations of its workforce [20, 21]. These descriptions, found in the Air Force Officer Classification Directory (AFOCD) and Air Force Enlisted Classification Directory (AFECD), articulate what knowledge, skills, and abilities (KSAs) are required for each specialty [20, 21]. However, AFSC documentation is typically unstructured: text buried across 300+ page PDFs, formatted inconsistently, and lacking machine-readable structure [20, 21]. This makes it difficult to query, compare, or link AFSC competencies to civilian workforce frameworks such as ESCO (European Skills, Competences, Qualifications and Occupations) or O*NET (the U.S. Department of Labor's Occupational Information Network), a comprehensive U.S. database sponsored by the Department of Labor that provides detailed information on occupations, worker skills, and job requirements [12, 14].

The scale of this challenge is significant. Approximately 200,000 service members transition to civilian employment annually [1], yet skill-translation frictions persist even amid historically low veteran unemployment (2.8% in 2023) [2, 6]. A 2024 Government Accountability Office review found that Department of Defense (DoD) and Department of Veterans Affairs (VA) assessed transition programs but could do more to measure outcomes such as sustained employment quality and credential attainment [3]. Department of Labor pilot studies have documented longstanding barriers in converting military experience to civilian credentials, noting that military training often lacks direct civilian equivalents despite substantial transferable value [4]. The Institute for Veterans and Military Families has similarly identified skills translation as a persistent friction point, particularly in the immediate post-separation period [5]. Automating KSA extraction with taxonomy alignment directly addresses this translation gap by producing structured, machine-readable competency data that can bridge military and civilian workforce frameworks [12, 14].

The goal of this project is to develop an end-to-end, automated AFSC → KSA extraction pipeline capable of processing AFSC documents into structured competency data. The system integrates LAiSER's skill extraction capabilities [8] with large language model (LLM) enhancement, semantic deduplication, taxonomy alignment, and graph database persistence to produce a queryable, interoperable KSA repository [12, 14, 13].

This capstone project delivered a complete, production-ready system comprising: (1) a modular Python extraction pipeline processing 12 AFSCs across intelligence, operations, and maintenance career fields, (2) a Neo4j graph database storing 253 KSAs with taxonomy alignments [13], and (3) a five-page Streamlit web application enabling AFSC exploration (Explore KSAs), interactive extraction testing (Try It Yourself), documentation reference, and administrative data management (Admin Tools) [16]. The system extracted an average of ~21 KSAs per AFSC, with skills aligned to the Open Skills Network (OSN) taxonomy comprising 2,217+ standardized skill definitions [18]. The application is deployed (https://fall-2025-group6-4w9txe2nuc2gn5h5ymtwbk.streamlit.app/) and all source code is publicly available on GitHub (https://github.com/Kyleinexile/fall-2025-group6/tree/main).

This work advances operational workforce analytics by providing a reproducible, scalable method for parsing AFSC texts into structured KSAs with provenance, semantic cleanup, and consistent formatting. The resulting graph enables cross-AFSC comparison, taxonomy-based skill matching, and serves as a foundation for future career translation tools [13].

2. Problem Statement

AFSC descriptions document critical competency requirements but exist only as narrative text within large PDF documents [20, 21]. This unstructured format prevents systematic querying of skills across specialties, quantitative comparison of competency overlaps between career fields, automated alignment to civilian skill taxonomies, and integration with workforce analytics platforms [20, 21]. Current military-to-civilian translation efforts rely primarily on curated manual mappings at the occupation level rather than granular skill extraction, approaches that are labor-intensive to maintain and cannot scale to the full inventory of 300+ AFSCs with evolving requirements [14, 12, 3, 4, 20, 21]. This project addresses the gap by developing an automated extraction pipeline embedded in an application that transforms raw AFSC text into structured KSA items with confidence scores, provenance tracking, and taxonomy alignment, enabling both human review and programmatic analysis [8, 13, 12, 14].

3. Related Work

Military-to-civilian translation has long been addressed at the program level, most visibly through the Department of Defense Transition Assistance Program (TAP), a mandatory, standardized pathway governed by DoDI 1332.35 and delivered across the Services [1]. SkillBridge complements TAP by placing separating Service members in industry internships during their final 180 days to gain directly translatable experience with employers who may extend full-time offers [15]. Despite these institutional supports, oversight bodies consistently highlight persistent frictions in transferring military experience into civilian-recognized credentials and jobs—especially where military training lacks direct civilian licensure or certification equivalents [3, 4].

Most public resources take an occupation-to-occupation crosswalk approach rather than extracting granular skills from source doctrine. O*NET provides detailed U.S. occupational descriptors and skill requirements used widely across workforce systems [14], while ESCO offers an EU skills/competences taxonomy designed for cross-market matching and multilingual interoperability [12]. However, neither resource ingests raw military specialty documents (e.g., AFOCD/AFECD) to yield structured, unit-level KSAs

Recent research efforts have begun to fill this skills-level gap. LAiSER (Leveraging AI for Skills Extraction & Research), a George Washington University initiative, demonstrates open-source skill extraction with built-in taxonomy alignment and reproducible prompts/workflows [8]. Our work extends this direction by: (i) extracting KSAs directly from AFSC source text, (ii) aligning to standardized taxonomies (e.g., ESCO), (iii) adding LLM-based Knowledge/Ability generation, and (iv) persisting results in a graph model to enable overlap analysis and reuse. Graph databases are a natural fit for workforce analytics—supporting many-to-many relationships among occupations, skills, taxonomies, and credentials—and are commonly recommended in knowledge-graph guidance for talent and HR use cases [13].

Together, these efforts motivate a pipeline that moves beyond static crosswalks to automated, auditable KSA extraction aligned to open taxonomies and stored in a graph structure suitable for querying similarity, overlap, and progression.

4. Solution and Methodology

The system is implemented as a modular pipeline with nine sequential stages. Each stage is represented by an independent Python module within the src/afsc_pipeline/ directory, enabling independent testing and maintainability.

4.1 Ingestion

AFSC competency descriptions were manually extracted from publicly available AFOCD/AFECD PDFs and saved as individual text files (one per AFSC) for batch processing [20, 21]. The pipeline ingests these pre-segmented files containing the complete specialty description including duties, responsibilities, and qualification requirements [20, 21].

The Streamlit application provides an alternative interactive ingestion pathway: users can search the full AFOCD/AFECD PDF documents (700+ pages) by keyword using PyPDF-based text extraction

[16–17, 20–21]. Search results display matching AFSC sections with context, allowing users to select and load specific AFSC text directly into the extraction pipeline. This enables on-demand processing of any AFSC without requiring pre-segmented files [16–17].

4.2 Preprocessing

Preprocess.py normalizes raw AFSC text into a clean narrative block suitable for extraction. The clean_afsc_text() function performs the following transformations:

- Removes PDF artifacts including hyphenated line breaks, bare page numbers, and markdown-style tables
- Strips classification headers (AFECD, DAFECD, AFI, AFMAN, USSF patterns) and trailing NOTE sections
- Removes code fences that may appear in copied documentation
- Canonicalizes all whitespace and newlines into single spaces

The output is a single, continuous text block optimized for LAiSER's skill extraction patterns and LLM context windows. This design prioritizes extraction recall over document structure preservation, as KSA items appear throughout AFSC descriptions rather than in predictable sections.

4.3 Skill Extraction (LAiSER)

LAiSER (Leveraging AI for Skills Extraction & Research) is an open-source framework developed at George Washington University for skill extraction with built-in alignment to standardized skill taxonomies including the Open Skills Network (OSN) and ESCO (European Skills, Competences, Qualifications and Occupations) frameworks [8, 18, 12]. The pipeline uses LAiSER's SkillExtractorRefactored class with a Gemini backend for extraction and taxonomy alignment [11].

When enabled (USE_LAISER=true), LAiSER identifies skill phrases from AFSC text and aligns them to standardized taxonomy entries. Extracted items include:

- Action-object skills (e.g., "conduct target analysis")
- Domain-specific tasks (e.g., "prepare intelligence reports")
- Technical competencies with OSN taxonomy codes (e.g., ESCO.95 → "design cloud architecture")

Each extracted item is represented as an ItemDraft object retaining: the skill text, a confidence score (correlation coefficient from taxonomy alignment), type classification (SKILL at this stage), OSN/ESCO taxonomy code (when alignment confidence is sufficient), and provenance tag indicating source ("laiser-gemini"). Results are sorted by confidence and capped at the top 25 items (configurable via LAISER_ALIGN_TOPK).

Fallback Extraction: When LAiSER is unavailable or disabled, the pipeline uses a regex-based heuristic extractor that identifies verb-object phrases using common action verbs (perform, conduct, analyze, develop, etc.) and domain-specific keywords. Fallback items receive lower confidence scores (0.30–0.55) and are tagged with source labels such as "fallback-pattern" or "fallback-domain" to distinguish them from LAiSER-derived items. This was developed mainly for troubleshooting purposes and is not intended as a final output.

4.4 LLM Enhancement (Knowledge & Abilities)

While LAiSER focuses primarily on verb-driven skills, AFSCs also encode conceptual knowledge requirements and cognitive/operational abilities. To generate complementary K/A items, the pipeline offers an optional LLM-based enhancement layer (disabled by default for cost optimization) that prompts models to produce items adhering to strict formatting rules:

- Only "Knowledge of ..." or "Ability to ..." surface forms

- Maximum 120 characters per item (soft guidance)
- No trailing punctuation
- Must complement (not duplicate) existing items

The system supports four providers with configurable models:

- OpenAI: gpt-4o-mini (cost-effective, fast response times) [9]
- Anthropic: claude-sonnet-4-5-20250929 (high reasoning quality) [10]
- Google: gemini-2.0-flash (balanced performance) [11]
- HuggingFace: Llama-3.2-3B-Instruct (open-source option) [19]

Provider selection was based on API availability, pricing considerations, and observed output quality during testing. OpenAI and Gemini have mutual fallback logic; if one fails, the other is attempted. Anthropic and HuggingFace fall back directly to heuristics on failure.

Heuristic Fallback: If LLM enhancement is disabled or all provider calls fail, a rule-based heuristic generates minimal K/A items by extracting topical noun phrases for "Knowledge of..." statements and converting skill verbs into "Ability to..." statements. Heuristic-generated items receive lower confidence scores (0.55) compared to LLM-generated items (0.70).

The system uses a 5,000-character input limit (capturing full AFSC context) and 1,024-token maximum output. Results are capped at 6 new items per AFSC to balance coverage with quality. A user-facing override, implemented in the Streamlit app, allows users to select a provider and supply their own API key, enabling the "Try It Yourself" bring-your-own-API (BYO-API) demonstration mode [16].

4.5 Quality Filtering

The quality_filter.py module prunes noisy or out-of-domain items before deduplication and graph persistence. Filtering rules include:

- **Length constraints:** Items shorter than 3 characters or exceeding type-specific limits (80 characters for Skills, 150 for Knowledge/Abilities) are removed
- **Banned phrases:** Known out-of-domain items (e.g., "business intelligence", "perform cleaning duties") are discarded
- **Text canonicalization:** Items are normalized to lowercase with collapsed whitespace, stripped punctuation, and canonical form replacements (e.g., "imagery analysis" → "imagery exploitation")
- **Exact deduplication:** Duplicate (item_type, text) pairs are removed before downstream processing

Additional optional filters (GEOINT domain bias, strict ESCO requirement) are available via environment variables for domain-specific tuning but were not enabled for this project.

4.6 Fuzzy Deduplication

AFSC documents and LLM outputs often express similar concepts in slightly different language. To canonicalize near-duplicates, the pipeline uses a lightweight hybrid text similarity approach combining token-level Jaccard similarity (60% weight) and character-level difflib sequence matching (40% weight). This text-based method avoids the computational overhead of embedding models while performing well on short KSA phrases.

The combined similarity score is calculated as:

$$similarity = 0.6 \times Jaccard(A, B) + 0.4 \times difflib\_ratio(A, B)$$

For example, "analysis of geospatial intelligence data" and "analysis geospatial intelligence data" yield Jaccard = 0.80 (4/5 token overlap) and difflib ≈ 0.95, producing a combined score of 0.86—just meeting the clustering threshold. Items are partitioned by type (Knowledge/Skill/Ability) before comparison; only items of the same type are considered potential duplicates. Pairs scoring ≥ 0.86 similarity are clustered together, and a canonical representative is selected for each cluster based on the following priority order:

1. Presence of an ESCO/OSN taxonomy code
2. Higher confidence score
3. LAiSER-sourced items favored over LLM-generated
4. Longer text as a tiebreaker

ESCO ID Propagation: If the selected canonical item lacks an ESCO code but another cluster member has one, the taxonomy code is propagated to the winner. This preserves taxonomy alignment that might otherwise be lost during deduplication.

The canonicalization process reduces redundancy while retaining the most informative version of each concept, ensuring a concise, non-redundant KSA inventory per AFSC.

4.7 Taxonomy Alignment (ESCO/OSN)

The pipeline aligns extracted skills to standardized taxonomies including the Open Skills Network (OSN) and ESCO (European Skills, Competences, Qualifications and Occupations) frameworks [18, 12]. LAiSER provides taxonomy codes directly during skill extraction via its integrated Gemini backend, items that match entries in LAiSER's reference taxonomy receive ESCO-format identifiers (e.g., ESCO.142) linking them to standardized skill definitions from a corpus of 2,217+ skills [8, 12].

Alignment coverage varies by AFSC based on terminology overlap with civilian skill frameworks. Items generated by the LLM enhancement layer (Knowledge and Ability statements) do not receive taxonomy codes, as they represent conceptual understanding rather than actionable skills catalogued in standard taxonomies. Military-specific terminology also frequently lacks direct civilian equivalents.

Note: A standalone esco_mapper.py module was developed as a potential local matching fallback using hybrid Jaccard/difflib similarity against a local ESCO catalog. This module is not currently integrated into the production pipeline, as LAiSER's native taxonomy alignment proved sufficient for project requirements [12].

4.8 Graph Persistence (Neo4j)

The final KSA set is written to a Neo4j Aura instance using graph_writer_v2.py [13]. Each KSA receives a unique identifier derived from its text content, ensuring that reprocessing an AFSC updates existing records rather than creating duplicates.

Nodes:

- AFSC - Air Force Specialty Code (keyed by code)
- KSA - Knowledge, Skill, or Ability item (keyed by content_sig)
- SourceDoc - Reference to source document (e.g., AFOCD/AFECD) [20, 21]
- ESCOSkill - Standardized taxonomy skill entry (keyed by esco_id) [12]

Relationships:

- REQUIRES (AFSC → KSA) - Links specialties to their required competencies
- EXTRACTED_FROM (KSA → SourceDoc) - Tracks provenance to source documentation
- ALIGNS_TO (KSA → ESCOSkill) - Connects items to standardized taxonomy entries

All operations use Cypher MERGE statements to ensure idempotency, reprocessing the same AFSC updates timestamps without creating duplicates. Uniqueness constraints on AFSC.code, KSA.content_sig, SourceDoc.title, and ESCOSkill.esco_id guarantee data integrity and query performance.

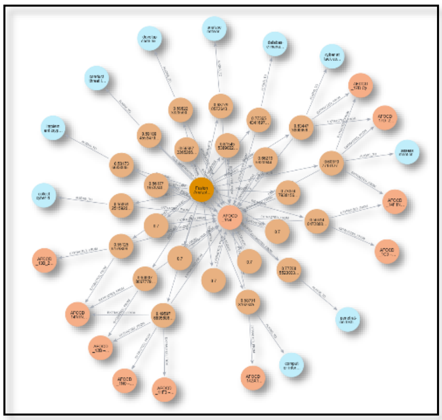

Figure 1. Neo4j graph visualization for AFSC 1N4 (Cyber Intelligence) showing KSA nodes with REQUIRES, EXTRACTED_FROM, and ALIGNS_TO relationships.

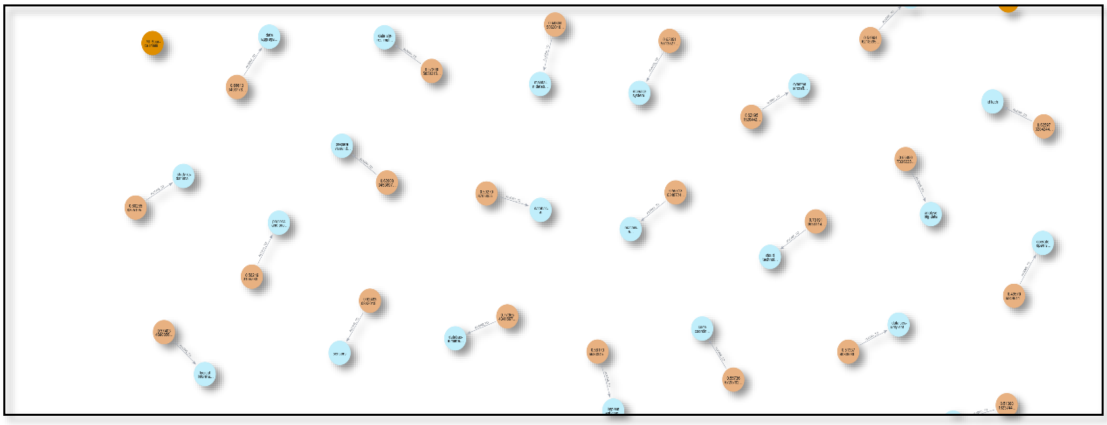

Figure 2. KSAs aligned to ESCO taxonomy entries, enabling military-to-civilian skill translation.

4.9 Streamlit Web Application

A production Streamlit application provides five distinct interfaces:

1. Home: System overview with real-time metrics banner (AFSC count, KSA totals by type), simplified 6-step pipeline visualization, and comprehensive methodology documentation. A system status sidebar displays connection health for Neo4j and all LLM providers.

Figure 3. System Home—live metrics banner and end-to-end pipeline overview with provider/DB health checks.

2. Try It Yourself: Public sandbox mode enabling users to: search AFOCD/AFECD PDF documents (700+ pages) by keyword, load AFSC text from search results or paste manually, run extraction with personal API keys (BYO-API model), test all four LLM providers (OpenAI, Anthropic, Gemini, HuggingFace), view extraction metrics and source attribution (LAiSER vs LLM), download results as CSV, with no database writes (sandbox mode).

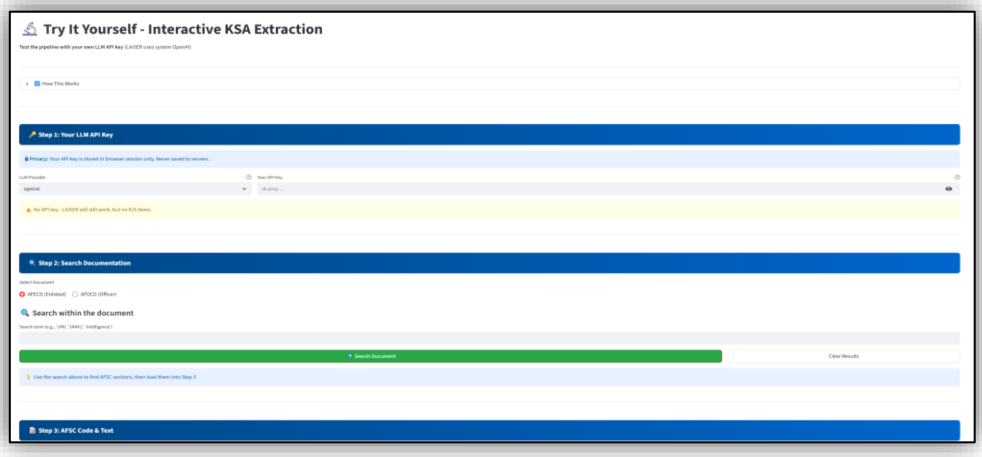

Figure 4. "Try It Yourself" workflow—PDF search, section load, and on-demand extraction (sandbox mode without database writes)

3. Explore KSAs: Interactive AFSC browser enabling: single AFSC detailed view with filtering by type, confidence threshold, and text search; multi-AFSC comparison (up to 5 AFSCs) with bar chart visualization; overlap analysis identifying shared KSAs across selected specialties; CSV export functionality for filtered results; and taxonomy code display for ESCO-aligned items.

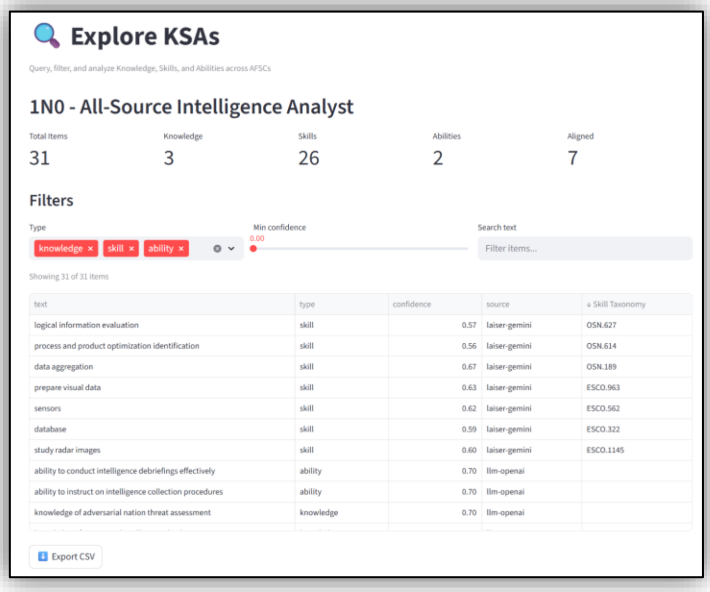

Figure 5. Explore KSAs—single-AFSC view with filters and taxonomy codes displayed for aligned skills.

4. Admin Tools (authentication required): Search and load AFSC text from source PDFs, ingest new AFSCs via full pipeline with database persistence, bulk JSONL upload for batch processing, delete AFSCs with automatic orphaned KSA cleanup, view extraction statistics and metrics after processing, and cache management utilities.

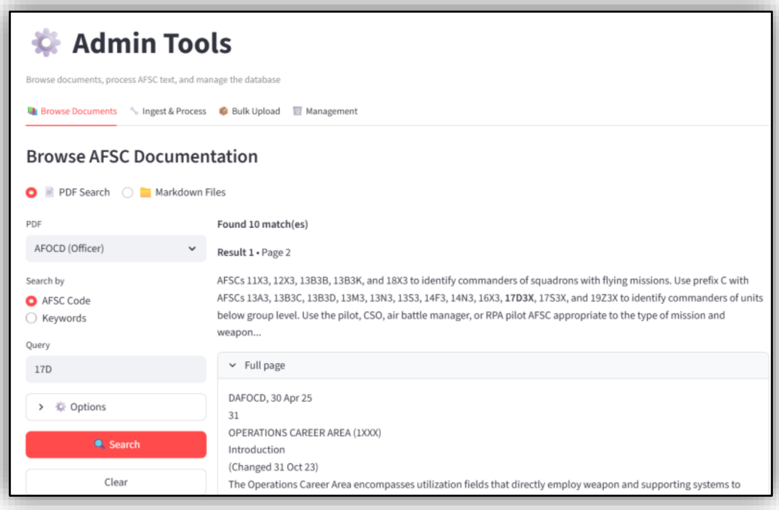

Figure 6. Admin Tools—controlled ingestion and post-process stats for auditable, idempotent writes.

5. Documentation & FAQ: Comprehensive technical reference including: system overview and key achievements, configuration details for LAiSER and LLM providers, pipeline flow and deduplication algorithm explanations, performance metrics and processing time breakdowns, cost analysis comparing automated vs. manual extraction, and frequently asked questions with detailed answers.

5. Results and Evaluation

Since formal SME evaluation and labeled test sets were not available within the capstone timeline, evaluation focuses on: (1) descriptive statistics of pipeline outputs, (2) qualitative assessment of KSA quality across multiple AFSCs and LLM providers, (3) system performance and usability, and (4) demonstration of end-to-end functionality.

5.1 Corpus and Coverage

The system successfully processed 12 Air Force Specialty Codes spanning three career fields:

AFSC	Title	Career Field	KSAs
14N	Intelligence Officer	Operations	30
21M	Munitions/Missile Maintenance	Maintenance	29
1N0	All Source Intelligence Analyst	Intelligence	31
1N4	Cyber Intelligence	Intelligence	24
11F3	Fighter Pilot	Operations	28
12B	Bomber Combat Systems Officer	Operations	25
14F	Information Operations	Operations	28
1A3X1	Mobility Force Aviator	Operations	27
1C3	All-Domain Command & Control	Operations	29
2A3	Tactical Aircraft Maintenance	Maintenance	27
2A5	Airlift/Special Mission Maintenance	Maintenance	24
21A	Aircraft Maintenance Officer	Maintenance	30
Total:		**12 AFSCs**	**253**

Figure 7. Average: ~28 KSAs per AFSC (332 total relationships ÷ 12); ~21 unique KSAs per AFSC (253 ÷ 12)

The difference between unique KSAs (253) and total AFSC→KSA relationships (332) reflects competency overlap across specialties; certain skills (e.g., "assessment of risks and threats," "database management systems") are shared by multiple AFSCs, demonstrating transferable competencies across career fields.

5.2 KSA Type Distribution

Across all extracted items:

- Skills: ~75% (189 items, action-oriented, LAiSER-extracted)
- Knowledge: ~14% (35 items, LLM-generated conceptual understanding)
- Abilities: ~11% (29 items, LLM-generated cognitive/physical capacities)

This distribution reflects both the structure of AFSC documents, which emphasize concrete duties and responsibilities over abstract knowledge domains, and pipeline design choices. LAiSER's extraction patterns favor action-verb phrases (Skills), while Knowledge and Ability generation is constrained by the LLM enhancement cap of 6 items per AFSC.

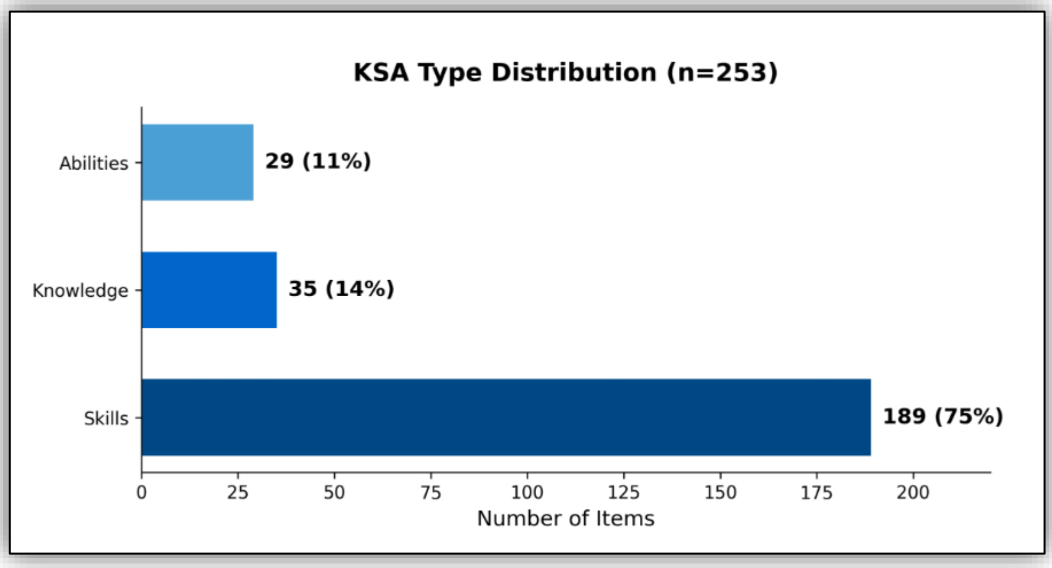

Figure 8. Distribution of extracted KSAs by type (n=253), reflecting LAiSER's skill-focused extraction and LLM-generated Knowledge/Ability items.

5.3 Taxonomy Alignment

- Skills with OSN codes: LAiSER-extracted skills include OSN/ESCO taxonomy alignments when confidence thresholds are met (format: ESCO.XX or OSN.XX → skill label). Alignment coverage varies by AFSC based on terminology overlap with civilian skill frameworks.
- Unaligned items: Primarily LLM-generated Knowledge/Ability items and military-specific terminology lacking civilian taxonomy equivalents.
- ESCO-aligned items: ~18% (46 KSAs linked to taxonomy; remainder are military-specific or LLM-generated items without direct equivalents)

5.4 Sample Extracted KSAs (AFSC 1N4 - Cyber Intelligence)

Skills (LAiSER + OSN/ESCO):

- "penetration testing solutions development" (OSN.28, confidence: 0.78)

- "analyze network configuration and performance" (ESCO.412, confidence: 0.69)
- "cyber attack counter-measures" (ESCO.439, confidence: 0.63)

Knowledge (LLM-generated):

- "Knowledge of cyber threat actor tactics, techniques, and procedures" (confidence: 0.70, OpenAI)
- "Knowledge of advanced persistent threats and mitigation strategies" (confidence: 0.70, OpenAI)

Abilities (LLM-generated):

- "Ability to develop and implement network defense strategies" (confidence: 0.70, OpenAI)
- "Ability to conduct digital forensics and incident response" (confidence: 0.70, OpenAI)

5.5 LLM Provider Comparison

Testing across AFSCs with all four providers:

- OpenAI (GPT-4o-mini): Concise, technically precise K/A items with consistent formatting; lowest need for cleanup.
- Anthropic (Claude Sonnet 4.5): Strongest reasoning depth and operational framing; slightly more verbose outputs that benefit from trimming.
- Google (Gemini 2.0 Flash): Diverse phrasing and good topical coverage; requires the most filtering to ensure uniform style.
- HuggingFace (Llama-3.2-3B-Instruct): Open-source alternative with reasonable quality; useful for cost-sensitive deployments.

All four providers produced domain-appropriate outputs. Semantic deduplication successfully collapsed provider-specific phrasing variations into canonical forms.

5.6 Pipeline Performance

- Average extraction time per AFSC: 60–80 seconds (LAiSER + filtering + dedupe); 90–104 seconds with optional LLM enhancement enabled
- Neo4j write operations: <5 seconds per AFSC (idempotent MERGE)
- Streamlit app response time: <2 seconds for cached AFSC queries
- Database size: 347 nodes, 735 relationships (12 AFSCs)

Node breakdown:

Node Type	Count	Description
KSA	253	Unique Knowledge, Skill, and Ability items
ESCOSkill	66	Standardized taxonomy skill entries
SourceDoc	16	Source document references (AFOCD/AFECD)
AFSC	12	Air Force Specialty Code nodes
Total	**347**	

Relationship breakdown:

Relationship Type	Count	Description
EXTRACTED_FROM	357	KSA → SourceDoc provenance links
REQUIRES	332	AFSC → KSA competency requirements
ALIGNS_TO	46	KSA → ESCOSkill taxonomy alignments
Total	**735**	

5.7 Quality and Usability Assessment

Qualitative Review (Manual Inspection):

Manual inspection of extracted KSAs across multiple AFSCs confirmed the following:

- LAiSER consistently extracts action-based skills with high domain relevance, particularly for technical and operational competencies
- LLM enhancement produces domain-correct Knowledge and Ability items that complement LAiSER's skill-focused extractions
- Quality filtering effectively removes noisy candidates including overly short items, banned phrases, and out-of-domain extractions
- Semantic deduplication reduces redundancy while preserving the highest-quality representative from each cluster
- ESCO/OSN alignment provides meaningful taxonomy bridges for approximately 18% of extracted items, enabling civilian workforce interoperability

Usability Testing (Informal):

Informal usability testing with the Streamlit application confirmed:

- Try It Yourself mode enables non-technical users to test extraction with their own API keys without database impact
- Explore KSAs interface supports rapid AFSC browsing, filtering by type/confidence, and multi-AFSC comparison
- Admin Tools successfully used for all 12 AFSC ingestions with full audit logging
- CSV export functionality enables downstream analysis in external tools
- Documentation & FAQ page provides comprehensive self-service technical reference

Limitations Observed:

- Some LAiSER extractions remain overly generic (e.g., "intelligence," "operations") despite quality filtering
- LLM phrasing varies across providers, requiring normalization through the deduplication pipeline
- ESCO alignment occasionally produces conceptual mismatches when military terminology diverges significantly from civilian frameworks
- Provenance tracking operates at the document level rather than page-level granularity
- Fallback heuristic extraction, while functional, produces lower-quality results compared to LAiSER when the primary extractor is unavailable

5.8 Scope Changes from Original Proposal

Several architectural and scope decisions diverged from the original proposal based on practical considerations encountered during development.

Technology Choices:

- Streamlit vs. Static React: Streamlit enabled rapid prototyping with an integrated Python backend, eliminating the need for separate API deployment. The framework's native session state and caching aligned well with the interactive demo requirements.
- Live Neo4j vs. Static Exports: A hosted Neo4j Aura instance provided superior query flexibility and enabled real-time AFSC comparisons. Static JSON exports were retained as downloadable artifacts but not as the primary data layer.

Algorithmic Adjustments:

- Deduplication Approach: The original proposal considered FAISS-based semantic embedding similarity for near-duplicate detection. This was replaced with a lightweight hybrid approach combining token-level Jaccard similarity (60% weight) and character-level difflib sequence matching (40% weight). This change simplified the process while maintaining effective deduplication for short KSA phrases.
- ESCO Mapping Strategy: A standalone local ESCO matching module (esco_mapper.py) was developed as a potential fallback using hybrid similarity against a local taxonomy catalog. This module was ultimately not integrated into the production pipeline, as LAiSER's native taxonomy alignment proved sufficient for project requirements.

Scope Adjustments:

- Expanded AFSC Coverage: 12 AFSCs processed (vs. proposed 6–8) due to efficient pipeline automation and stable ingestion tooling
- Descoped Re-ranker: The optional preference-learning ranking component was descoped to prioritize core pipeline functionality and application polish within the capstone timeline
- Provider Expansion: HuggingFace was added as a fourth LLM provider beyond the originally planned three (OpenAI, Anthropic, Gemini), providing an open-source alternative for cost-sensitive applications.
- Enhanced Application Features: Added "Try It Yourself" BYO-API demonstration mode, multi-AFSC comparison with overlap analysis, and a comprehensive Documentation & FAQ page beyond original specification

These changes improved overall system usability and demonstration value while maintaining alignment with core capstone objectives.

6. Limitations

- No quantitative validation: Due to capstone timeline constraints, no labeled test set or SME evaluation was conducted. Results rely on qualitative assessment and descriptive statistics rather than precision/recall metrics.
- Generic extractions: LAiSER occasionally extracts overly generic or fragmentary skills (e.g., "intelligence," "operations") that pass through quality filtering despite limited semantic value.
- Provider phrasing variation: LLM-generated items vary in phrasing across providers, requiring normalization through the deduplication pipeline to achieve canonical forms.
- Taxonomy alignment gaps: ESCO/OSN mapping relies on lexical and semantic similarity; conceptual mismatches may occur when military terminology diverges significantly from civilian frameworks. Only ~18% of extracted KSAs achieved taxonomy alignment.
- Document-level provenance: Provenance is tracked at the document level but not page-level granularity, limiting traceability to specific source passages.
- No criticality weighting: All KSAs are treated equally based on confidence scores; no SME-based ranking or operational criticality weighting is applied.
- Corpus variability: AFSC document structure and content density varies, leading to uneven KSA yields (range: 24–31 items per AFSC).
- Default configuration constraints: LAiSER extraction and LLM enhancement are disabled by default in the production deployment for cost optimization; enabling these features requires explicit environment configuration.
- Descoped re-ranker: The optional preference-learning ranking component proposed in the original scope was descoped to prioritize core pipeline functionality and application development within the capstone timeline.

7. Ethical Considerations

- LLM bias awareness: LLM outputs may encode latent biases present in training data. This is mitigated through structured prompt design, explicit formatting requirements ("Knowledge of..." / "Ability to..." patterns), maximum length constraints, and quality filtering.
- Data privacy: No personally identifiable information (PII) or sensitive operational data is processed. All source documents (AFOCD/AFECD) are unclassified and publicly available. User-provided API keys in the Try It Yourself demonstration mode are stored in browser session only and are not persisted to servers.
- Taxonomy assumptions: ESCO/OSN alignment introduces taxonomic assumptions derived from civilian workforce frameworks that may not fully match USAF doctrine or operational context. Users should validate alignments for critical workforce planning applications.
- Human-in-the-loop: Automated KSA extraction should supplement, not replace human judgment in workforce decisions, career counseling, and training program design. Confidence scores and provenance fields are provided to support informed human review.
- Transparency and reproducibility: Comprehensive audit logs, provenance fields, and confidence scores are preserved for all extracted items. All code, documentation, and extracted data are maintained in a public GitHub repository (https://github.com/Kyleinexile/fall-2025-group6) to ensure reproducibility and academic transparency.

8. Discussion

Taxonomy coverage and implications: The ~18% ESCO/OSN alignment rate suggests meaningful overlap between AFSC skill language and civilian taxonomies, while also indicating a substantial "military-specific tail" that lacks direct civilian analogs. This is expected: OSN/ESCO emphasize broadly transferable skills (e.g., cloud security, data analysis, network configuration), whereas many AFSC phrasings are doctrine-specific or mission-contextual (e.g., "electronic signals intelligence," "order of battle analysis"). The 46 taxonomy-aligned KSAs provide immediate bridges to civilian workforce frameworks, while the remaining items represent opportunities for future crosswalk development.

S/K/A distribution (~75/14/11): The dominance of Skills reflects both AFSC prose that centers on duties and action verbs, LAiSER's extraction sweet spot, and the pipeline's design constraint capping LLM-generated K/A items at 6 per AFSC. This validates the utility of a constrained LLM enhancer to surface conceptual domains and cognitive abilities that complement LAiSER's skill-focused extractions.

Operational fit for transition programs: A structured KSA inventory with taxonomy links supports TAP/SkillBridge use cases in two ways: (1) it highlights transferable skills via ESCO/OSN alignments for résumé translation and civilian job matching, and (2) it reveals gaps where AFSC-specific competencies need civilian phrasing or coursework bridges. The 332 AFSC→KSA relationships with 79 instances of cross-AFSC overlap demonstrate that many competencies transfer across specialties, further supporting career mobility analysis.

Scalability: The current pipeline processes AFSC text in 60–80 seconds per specialty (90–104 seconds with LLM enhancement enabled) with idempotent graph writes. Because deduplication uses lightweight text similarity (Jaccard + difflib) rather than embedding models, and ESCO alignment is integrated directly with LAiSER extraction, cost and latency remain manageable as coverage scales toward the full inventory of 300+ AFSCs. The main bottlenecks at scale become provider rate limits and source document segmentation quality; both are addressable with caching, exponential backoff, and batched ingestion workflows.

Where this helps analysts: Graph queries enable workforce planners to compare AFSCs, identify shared competencies across career fields, and target curriculum design where skill gaps cluster. The overlap analysis feature (Explore KSAs page) operationalizes this for rapid exploration, while CSV exports enable integration with external analytics tools. The 347-node, 735-relationship graph provides a queryable foundation for cross-specialty workforce analysis that would be impractical with unstructured document review.

9. Conclusion and Future Work

This capstone project successfully delivers a complete AFSC → KSA extraction ecosystem integrating LAiSER skill extraction, multi-provider LLM enhancement (OpenAI, Anthropic, Gemini, HuggingFace), semantic deduplication, OSN/ESCO taxonomy alignment, Neo4j graph persistence, and a five-page Streamlit web application. The system processed 12 AFSCs across intelligence, operations, and maintenance career fields, extracting 253 unique KSAs with an average of ~21 items per AFSC and an ~18% skill-taxonomy alignment rate.

The delivered artifacts include: (1) a modular, production-ready Python pipeline (src/afsc_pipeline/), (2) a live Neo4j Aura graph database with 347 nodes and 735 relationships, (3) a deployed Streamlit application with demonstration, exploration, documentation, and administrative modes, and (4) comprehensive documentation enabling future extension and maintenance.

The system demonstrates the feasibility of automated military competency extraction at scale. By combining modern NLP frameworks (LAiSER), flexible LLM orchestration across four providers, and graph-based knowledge representation, this work provides a foundation for bridging the gap between military workforce data and civilian skill frameworks.

Future work may include:

- SME-driven ranking and weighting of KSAs based on operational criticality and training priority
- Expanded coverage to all USAF enlisted and officer AFSCs, with potential integration of Navy (NEC), Army (MOS), and Marine Corps occupation codes
- O*NET-based modeling to enable direct military-to-civilian occupation translation and job matching [14]
- Integration with Air Force workforce analytics platforms (MyVector, vMPF) for real-time career guidance
- Development of a skill-gap analysis tool comparing AFSC requirements against civilian job postings
- Longitudinal tracking of AFSC requirement changes across AFOCD/AFECD versions to inform curriculum updates
- Implementation of the descoped preference-learning re-ranker to enable SME feedback incorporation

This project establishes both technical feasibility and operational value, positioning the system for potential transition to Air Force HR and education stakeholders.

References

[1] U.S. Department of Defense, Office of the Under Secretary of Defense (Personnel & Readiness). (n.d.). *Transition Assistance Program (TAP).* https://www.defense.gov/CPCC/Transition-Assistance-Program/

[2] U.S. Bureau of Labor Statistics. (2024, March 20). *The employment situation of veterans—2023 (USDL-24-0532).* https://www.bls.gov/news.release/vet.htm

[3] U.S. Government Accountability Office. (2024, September 5). *Transitioning service members: DOD and VA assessed programs but could do more to measure outcomes (GAO-24-107752).* https://www.gao.gov/products/gao-24-107752

[4] U.S. Department of Labor, Veterans□ Employment and Training Service. (2014). *Converting military experience into civilian credentials: A DOL VETS pilot study.* https://www.dol.gov/agencies/vets/

[5] Institute for Veterans & Military Families, Syracuse University. (2022). *Emerging evidence about transition employment support.* https://ivmf.syracuse.edu/

[6] Pew Research Center. (2019, September 10). *The changing face of America□s veteran population.* https://www.pewresearch.org/

[7] □ *(intentionally unused to preserve your in-text numbering)*

[8] LAiSER: Leveraging AI for Skills Extraction & Research. (n.d.). *LAiSER extract module* [GitHub repository]. https://github.com/LAiSER-Software/extract-module

[9] OpenAI. (2024). *GPT-4o model card.* https://platform.openai.com/docs/models/gpt-4o

[10] Anthropic. (2024). *Claude Sonnet 4.5 model card.* https://www.anthropic.com/claude

[11] Google DeepMind. (2024). *Gemini 2.0 Flash technical report.* https://deepmind.google/technologies/gemini/

[12] European Commission. (2024). *ESCO: European Skills, Competences, Qualifications and Occupations.* https://esco.ec.europa.eu/

[13] Neo4j, Inc. (2024). *Neo4j graph database documentation.* https://neo4j.com/docs/

[14] U.S. Department of Labor. (2024). *O*NET OnLine: Occupational Information Network.* https://www.onetonline.org/

[15] U.S. Department of Defense. (2024). *SkillBridge program.* https://skillbridge.osd.mil/

[16] Streamlit, Inc. (2025). *Streamlit documentation.* https://docs.streamlit.io/

[17] PyPDF contributors. (2025). *PyPDF documentation (stable).* https://pypdf.readthedocs.io/

[18] Open Skills Network. (2025). *About the Open Skills Network.* https://www.openskillsnetwork.org/

[19] Meta AI. (2024, December 19). *Llama 3.2: Lightweight models for edge devices.* https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

[20] U.S. Air Force. (2025). *Air Force Enlisted Classification Directory (AFECD).* https://www.afpc.af.mil/ *(navigate to **Enlisted Classification** for the current PDF)*

[21] U.S. Air Force. (2025). *Air Force Officer Classification Directory (AFOCD).* https://www.afpc.af.mil/ *(navigate to **Officer Classification** for the current PDF)*