# AFSC → KSA Graph Explorer: An Automated Pipeline for Translating Air Force Specialty Code Narratives into Structured Competencies

Kyle Hall
*Data Science Program, George Washington University*
*Washington, DC, USA*
*Kyle.hall@gwu.edu*

***Abstract***—The United States Air Force relies on Air Force Specialty Codes (AFSCs) to document the roles, responsibilities, and competency expectations of its workforce. AFSC descriptions appear as long form, unstructured text within the Air Force Officer Classification Directory (AFOCD) and the Air Force Enlisted Classification Directory (AFECD), which limits their direct use in workforce analytics and civilian translation. This paper presents an end-to-end pipeline that parses AFSC text into structured knowledge, skills, and abilities (KSA) items with provenance and optional taxonomy alignment. The system combines LAiSER based skill extraction with a multi-provider LLM enhancement layer, quality filtering, lightweight fuzzy deduplication, and graph persistence in Neo4j. A Streamlit application exposes a public interface for exploration and an administrative ingest workflow. Results across 12 AFSCs yielded 332 KSA instances that deduplicated to 253 unique items, with 79 shared across multiple specialties. The resulting graph contains 347 nodes and 735 relationships, with 46 items (18.2%) receiving ESCO taxonomy alignment, enabling overlap analysis and downstream civilian matching while illuminating a domain specific tail that lacks direct civilian analogs.

***Keywords***—AFSC, KSA extraction, skill taxonomy, Neo4j, NLP, military transition, ESCO, workforce analytics

## I. Introduction

The United States Air Force relies on Air Force Specialty Codes (AFSCs) to define the roles, responsibilities, and competency expectations of its workforce. These descriptions, found in the Air Force Officer Classification Directory (AFOCD) and Air Force Enlisted Classification Directory (AFECD), articulate the knowledge, skills, and abilities (KSAs) required for each specialty [1], [2]. However, AFSC documentation is typically unstructured text, buried across 300+ page PDFs, formatted inconsistently, and lacking machine-readable structure, which makes it difficult to query, compare, or link competencies to civilian frameworks such as ESCO or O*NET [3], [4].

The scale of the translation challenge is significant. Approximately 200,000 service members transition to civilian employment annually [5], yet skills translation frictions persist even amid historically low veteran unemployment (2.8% in 2023) [6]. Oversight assessments note that while programs such as TAP and SkillBridge provide structured support, outcome measurement and credential conversion remain uneven [7], [8]. Automating KSA extraction with taxonomy alignment directly addresses this gap by producing structured, machine-readable competency data that can bridge military and civilian workforce frameworks.

This work contributes: (1) an end-to-end automated KSA extraction pipeline processing raw AFSC text, (2) integration of LAiSER skill extraction with multi-provider LLM enhancement for knowledge and ability generation, (3) a graph-based knowledge representation with ESCO taxonomy alignment stored in Neo4j, and (4) a publicly accessible Streamlit demonstration application enabling AFSC exploration and overlap analysis.

## II. Problem Statement

AFSC descriptions document critical competency requirements but exist only as narrative text within large PDF documents. This unstructured format prevents systematic querying of skills across specialties, quantitative comparison of competency overlaps, automated alignment to civilian taxonomies, and integration with analytics platforms. Current translation efforts rely largely on curated occupation level crosswalks rather than granular skill extraction and are labor intensive to maintain for 300+ AFSCs with evolving requirements. This project addresses the gap by transforming AFSC text into structured KSA items with confidence scores, provenance tracking, and taxonomy alignment, enabling both human review and programmatic analysis.

## III. Related Work

Military-to-civilian translation is addressed programmatically through the Department of Defense Transition Assistance Program (TAP) [5] and DoD SkillBridge [8], but persistent frictions in converting military experience into civilian recognized credentials and jobs have been noted by DOL and GAO [7], [9]. Most public resources adopt occupation-to-occupation crosswalks; O*NET provides detailed U.S. occupational descriptors and skill requirements [4], while ESCO offers an EU skills and competences taxonomy for multilingual interoperability [3]. However, neither resource ingests raw military specialty documents to yield structured KSAs.

LAiSER (Leveraging AI for Skills Extraction & Research) demonstrates open-source skill extraction with taxonomy alignment [10]. Our work extends this direction by: (i) extracting KSAs directly from AFSC source text, (ii) aligning to standardized taxonomies, (iii) adding LLM based knowledge/ability generation, and (iv) persisting results in a graph model suitable for overlap analysis and reuse. Graph databases natively represent many-to-many relationships

among occupations, skills, taxonomies, and credentials, and are widely used in knowledge graph talent analytics [11].

## IV. Data and Sources

Primary sources are AFOCD and AFECD PDFs obtained from the U.S. Air Force Personnel Center [1], [2]. These documents collectively span over 600 pages and describe 300+ distinct AFSCs across officer and enlisted career fields. For this study, 12 AFSCs were selected to ensure balanced representation across three career domains: intelligence (1N0, 1N4, 14N, 14F), operations (11F3, 12B, 1A3X1, 1C3), and maintenance (21M, 2A3, 2A5, 21A).

For batch processing, AFSC competency descriptions were manually segmented from source PDFs into per-AFSC text files. The Streamlit application also exposes an interactive pathway using PyPDF [12] to search full PDFs and load text into the pipeline. Civilian taxonomies are referenced through ESCO [3] and the Open Skills Network [13] for alignment and comparison.

## V. Methods

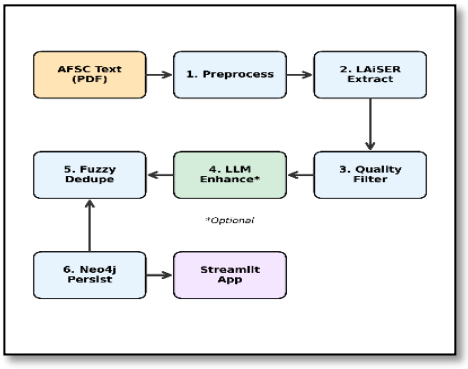


*Fig. 1. Pipeline architecture: preprocessing through Neo4j persistence and Streamlit exposure.*

### *A. Document Ingestion*

AFSC competency descriptions were manually segmented from AFOCD/AFECD into per-AFSC text files for batch processing. The Streamlit application provides an alternative interactive ingestion pathway, allowing users to search full PDFs by AFSC code and load relevant sections into the pipeline [12].

### *B. Preprocessing*

The preprocessing module normalizes raw AFSC text by removing PDF artifacts (hyphenation, page numbers, tables), stripping headers/footers, removing code fences, and canonicalizing whitespace. The output is a single continuous narrative optimized for extraction.

### *C. Skill Extraction (LAiSER)*

LAiSER (Leveraging AI for Skills Extraction & Research) provides skill extraction with integrated taxonomy alignment [10]. The pipeline uses the SkillExtractorRefactored class with a Gemini backend [14], returning up to 25 skills per AFSC with confidence scores and, when available, ESCO aligned identifiers [3]. When LAiSER is unavailable, a heuristic fallback produces lower confidence items tagged with source provenance.

### *D. LLM Enhancement (Knowledge & Abilities)*

To complement verb centric skills with conceptual knowledge and abilities, the pipeline includes an optional LLM enhancement layer. Prompts enforce strict surface forms ("Knowledge of…", "Ability to…") and limit output to 6 items per type per AFSC to control costs and maintain quality. Supported providers include OpenAI GPT-4o-mini [15], Anthropic Claude Sonnet 4.5 [16], Google Gemini 2.0 Flash [14], and Meta Llama 3.2-3B-Instruct via HuggingFace [17].

### *E. Quality Filtering*

The quality filter prunes out-of-domain or malformed items using length limits (8–200 characters), banned phrase detection, surface form canonicalization, and exact deduplication before downstream processing.

### *F. Fuzzy Deduplication*

*0.6 × token-level Jaccard + 0.4 × character-level difflib ratio*

Near duplicates are identified using a hybrid similarity metric, with a threshold of 0.86. Items are clustered by type (knowledge/skill/ability), with winner selection prioritizing: (1) ESCO presence, (2) higher confidence, (3) LAiSER source, and (4) longer length. ESCO IDs propagate to canonical winners where appropriate.

### *G. Taxonomy Alignment*

Skills aligned by LAiSER carry ESCO format identifiers linking to standardized definitions [3]. Knowledge and Ability items generated by LLMs are generally not assigned taxonomy codes due to their conceptual nature.

### *H. Graph Persistence (Neo4j)*

KSA items and AFSC nodes are written to Neo4j [11] with idempotent MERGE operations, ensuring upserts keyed by content signatures. The schema includes three relationship types: REQUIRES (AFSC → KSA), EXTRACTED_FROM (KSA → SourceDoc), and ALIGNS_TO (KSA → ESCO).

### *I. Streamlit Web Application*

A production Streamlit application [18] provides five functional pages: Home (overview and status), Try It Yourself (sandbox with BYO-API keys), Explore KSAs (filtering and CSV export), Admin Tools (bulk ingestion and cleanup), and Documentation & FAQ.

## VI. Results

### *A. Corpus Coverage*

The pipeline processed 12 AFSCs representing three career domains. Table I summarizes extraction results: intelligence AFSCs (1N0, 1N4, 14N, 14F) yielded 113 KSAs; operations AFSCs (11F3, 12B, 1A3X1, 1C3) yielded 109 KSAs; and maintenance AFSCs (21M, 2A3, 2A5, 21A) yielded 110 KSAs. The 332 total instances deduplicated to 253 unique items, with 79 items (23.8%) shared across multiple specialties.

| AFSC | Title | Career Field | KSAs |
|---|---|---|---|
| 14N | Intelligence Officer | Intelligence | 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 | Intelligence | 28 |
| 1A3X1 | Mobility Force Aviator | Operations | 27 |
| 1C3 | All-Domain C2 | Operations | 29 |
| 2A3 | Tactical Aircraft Maintenance | Maintenance | 27 |
| 2A5 | Airlift/Special Mission Maintenance | Maintenance | 24 |
| 21A | Aircraft Maintenance Officer | Maintenance | 30 |
| **Total:** | **KSA** | | **332** |
| **Total:** | **Unique KSA (deduplication)** | | **253** |

*[TABLE I: AFSC Extraction Results - 12 AFSCs, 332 instances → 253 unique KSAs after deduplication]*

### *B. Deduplication and Cross-AFSC Overlap*

The 332 per-AFSC KSA instances reduced to 253 unique items after fuzzy deduplication, indicating that 79 KSAs (23.8% of instances) appear across multiple specialties. These shared competencies represent transferable skills, for example, "analyze intelligence data" appears across 1N0, 1N4, and 14N, while "coordinate maintenance operations" spans 21M, 2A3, 2A5, and 21A.

### *C. KSA Type Distribution*

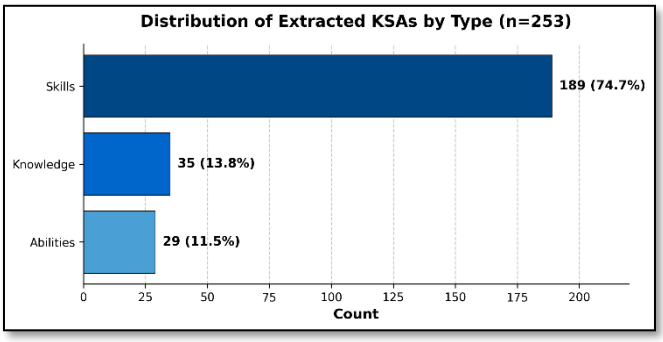


*Fig. 2. Distribution of extracted KSAs by type*

Of 253 unique KSAs: Skills comprise 189 (74.7%), Knowledge 35 (13.8%), and Abilities 29 (11.5%). This distribution reflects both AFSC document structure, which emphasizes concrete duties over abstract knowledge, and the pipeline's LLM enhancement cap of 6 Knowledge/Ability items per AFSC.

### *D. Taxonomy Alignment*

Of 253 unique KSAs, 46 (18.2%) received ESCO taxonomy alignment through LAiSER. Aligned items predominantly represent transferable competencies with direct civilian analogs. The remaining 207 items (81.8%) constitute a military specific tail requiring either manual taxonomy extension or transitional phrasing for civilian translation.

### *E. Graph Statistics*

The resulting Neo4j graph contains 347 nodes (12 AFSC, 253 KSA, 66 ESCO reference, 16 source document) and 735 relationships (332 REQUIRES, 357 EXTRACTED_FROM, 46 ALIGNS_TO).

### *F. Pipeline Performance*

End-to-end processing time per AFSC: 60–80 seconds without LLM enhancement, 90–104 seconds with enhancement enabled.

## VII. Discussion

Taxonomy coverage at 18.2% suggests both immediate bridges to civilian frameworks and opportunities to craft transitional phrasing for military-specific competencies. The dominance of Skills (74.7%) 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 Knowledge/Ability items at 6 per AFSC.

The 79 cross-AFSC overlaps (23.8% of total instances) demonstrate the graph's utility for identifying transferable competencies. Structured KSAs with taxonomy links support programs like TAP and SkillBridge by: (1) surfacing immediately transferable skills for resume translation and job matching, and (2) revealing gap areas where additional credentialing may bridge military specific competencies to civilian requirements.

## VIII. Limitations

Several limitations constrain current results: (1) Sample size: 12 AFSCs represent approximately 4% of the 300+ total inventory; (2) PDF text quality: complex tables may produce artifacts; (3) Taxonomy coverage: ESCO alignment captures only 18.2% of items; (4) Evaluation methodology: without SME labeled ground truth, precision/recall metrics cannot be computed; (5) LLM variability: different providers produce varying output quality.

## IX. Future Work

Future priorities include: expanded corpus processing all 300+ AFSCs, SME validation for ground truth labeling, fine tuning for military terminology, dual ESCO/O*NET alignment, active learning feedback loops, and credential mapping to civilian certifications.

## X. Conclusion

This work demonstrates a reproducible, scalable pipeline that transforms AFSC narrative text into structured KSAs suitable for analytics and civilian alignment. Results across 12 AFSCs yielded 332 KSA instances that deduplicated to 253 unique items, with 79 shared across multiple specialties and 46 (18.2%) receiving ESCO taxonomy alignment. The resulting graph of 347 nodes and 735 relationships enables overlap analysis, taxonomy based matching, and operational insights for transition and workforce planning. Source code: https://github.com/Kyleinexile/fall-2025-group6


## Acknowledgment

The author thanks the LAiSER team and George Washington University Data Science faculty for guidance and feedback during development.

## REFERENCES


[1] U.S. Air Force. (2025). Air Force Officer Classification Directory (AFOCD). https://www.afpc.af.mil/

[2] U.S. Air Force. (2025). Air Force Enlisted Classification Directory (AFECD). https://www.afpc.af.mil/

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

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

[5] U.S. Department of Defense. (n.d.). Transition Assistance Program (TAP). https://www.defense.gov/CPCC/Transition-Assistance-Program/

[6] U.S. Bureau of Labor Statistics. (2024). The employment situation of veterans—2023. https://www.bls.gov/news.release/vet.htm

[7] U.S. Government Accountability Office. (2024). Transitioning service members (GAO-24-107752). https://www.gao.gov/products/gao-24-107752

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

[9] U.S. Department of Labor, VETS. (2014). Pilot study translating military skills. https://www.dol.gov/agencies/vets/

[10] LAiSER. (n.d.). LAiSER extract module [GitHub]. https://github.com/LAiSER-Software/extract-module

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

[12] PyPDF contributors. (2025). PyPDF documentation. https://pypdf.readthedocs.io/

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

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

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

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

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

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