Training Developer Feedback on AI for Revision of Content (ARC)

Benjamin D. Nye, Jose-Luis Ambite, Joel Mathew, Mark G. Core,
Daniel Auerbach, Dilan R. Ramirez, Joel Walsh
University of Southern California
Los Angeles, CA
nye@ict.usc.edu, ambite@isi.edu, joel@isi.edu, core@ict.usc.edu,
auerbach@ict.usc.edu, dramirez@ict.usc.edu, joelawalsh@gmail.com

ABSTRACT

Due to rapid technological evolution and dynamic operational challenges, the ability to train military personnel effectively and quickly is a strategic necessity. However, a single change in Army doctrine requires changes across many courses and training resources, with some estimates indicating several hundred man-years of required updates accumulated. The AI-Assisted Revision of Content (ARC) tool uses AI to analyze doctrine and training documents (lessons, slides), to suggest which sections of training need to be updated. ARC was designed with three main capabilities: 1) Ingestion: Specialized scraping of documents ingested (Field Manuals, Army Doctrine Publications, lesson plans, presentations), which extracts passages, page references, and section structure, where possible; 2) Normalized Hybrid Search: Analyzing document-to-document alignment and similarity comparisons, using hybrid search with both sentence transformers and keyword embeddings (Okapi BM25), with a novel best-match normalization technique; 3) Change Analysis: Specialized user interfaces to enable comparing training documents, both in pairs (e.g., previous vs. latest doctrine) and triads (e.g., a slide deck referenced against its older vs. current doctrine source). Guided alpha testing collected formative input from seven Army training Centers of Excellence (CoE's), which determined design priorities to refine ARC capabilities. Results from hands-on beta tests with four Army Centers were highly positive, strongly agreeing that ARC would "increase productivity" (average rating 5.5 on a 6 point scale). However, CoE's reported differences in their needs, such as their typical changes (e.g., reference updates vs. explaining new concepts) or detailed auditing (e.g., safety-critical instructions from a technical manual). Ongoing research on ARC is exploring annotating and suggesting changes in editable documents that soldiers use every day (e.g., SharePoint, Word). Long term, strong demand by training centers suggests that sandboxes and rapid transition infrastructure should be implemented to support AI ecosystems for content development.

ABOUT THE AUTHORS

Benjamin Nye, Ph.D. is Director of Learning Science at the USC Institute for Creative Technologies (ICT). Dr. Nye's research has been recognized for excellence in intelligent tutoring systems (First Place ONR ITS STEM Grand Challenge), cognitive agents (BRIMS 2012 best paper), realistic behavior in training simulations (Federal Virtual Worlds Challenge), and machine learning for adaptive systems (OMEGA, I/ITSEC 2021 Best Paper Overall, with Eduworks). Nye's research is on scalable learning technologies and design principles that promote learning. His recent work emphasizes AI tools for instructors and content developers to use AI tools that enable them to rapidly update content and to create AI-enabled learning experiences (i.e., AI-human teams to generate AI tutoring).

Jose-Luis Ambite, Ph.D. is an Associate Research Professor of Computer Science, and a Principal Scientist at the Information Sciences Institute. He is an expert in data integration, including query rewriting under constraints, learning schema mappings, entity linkage, and information extraction. He received the Best Research Paper award at the International Semantic Web Conference in 2012. In the "L2K2R2: Learn to Read to Know, Know to Learn to Read" project, funded under the DARPA Big Mechanism program, he developed neural entity extraction and normalization methods from the biomedical literature. Within the NIH Big Data to Knowledge (BD2K) Training Coordination Center, Dr Ambite led the development of the educational resource discovery index, which indexed ~12,000 resources, and of techniques to automatically discover, model, organize learning resources for biomedical big data. His research interests include data integration, databases, knowledge representation, semantic web, semantic similarity, biomedical data science, and federated learning.

Joel Mathew is a Research Engineer at the USC Information Sciences Institute in the AI Division. He has worked extensively in developing various neural models for addressing tasks in Natural Language Processing (NLP). He developed techniques to combat spear-phishing in the DARPA ASED (Active Social Engineering Defense) program and co-developed techniques to catch textual inconsistencies for extremely low-resource languages for Bible Translation as part of the Greek Room project. His active interests include working on machine translation, automated quality assessment for translations and building tools for supporting Bible translation into low-resource languages.

Mark G. Core, Ph.D. is a Research Scientist in ICT's Learning Science, researching topics such as authoring tools, natural language processing, virtual reality and data analytics. He has over 15 years of experience in developing and evaluating virtual role players for learning, such as BiLAT (training of bilateral negotiation, winner of a 2008 Army Modeling and Simulation Award), and the Standard Patient Studio (training of medical interviewing, winner of multiple awards including 2016 Best Government Game, I/ITSEC). He has also published research on machine learning pipelines for automated analyses, including frameworks such as SLATS (Semi-Supervised Learning for Assessment of Teams in Simulations) and RACR (Rapid Adaptive Content Registry).

Daniel Auerbach is the senior Research Programmer for the USC ICT Learning Sciences group. He specializes in intelligent systems frameworks, particularly for service-oriented and agent-based frameworks. He received a B.A. in Linguistics and Computer Science from Cornell University.

Dilan R. Ramirez is a Research Programmer at USC ICT in the Learning Science group, with a focus on full-stack intelligent systems such as MentorPal virtual mentors and the Rapid Adaptive Content Registry (RACR) system. He is also engaged with the University of Texas at El Paso as a research assistant contributing to spatial visualizations and web applications. Dilan graduated from the University of Texas at El Paso with a BS in Computer Science.

Joel Walsh is a postdoctoral fellow at at the USC Institute for Creative Technologies (ICT). Dr. Walsh's research involves leveraging multimodal generative models for teaching and learning.

Training Developer Feedback on AI for Revision of Content (ARC)

Benjamin D. Nye, Jose-Luis Ambite, Joel Mathew, Mark G. Core,
Daniel Auerbach, Dilan R. Ramirez, Joel Walsh

University of Southern California
Los Angeles, CA
nye@ict.usc.edu, ambite@isi.edu, joel@isi.edu, core@ict.usc.edu,
auerbach@ict.usc.edu, dramirez@ict.usc.edu, joelawalsh@gmail.com

INTRODUCTION

Military services are fundamentally "learning organizations" that engage in not only training, but also lessons learned about organizational and business processes. This includes both smaller updates (e.g., changes to technical details) and larger doctrinal shifts. An example of this methodology to identify gaps in current Army capabilities is DOTmLPF-P analysis, which provides a comprehensive evaluation of **D**octrine (principles that guide combat and other operations), **O**rganization (how to organize to fight), **T**raining (how to prepare to fight), **M**ateriel (necessary equipment), **L**eadership and education (how we prepare our leaders to lead the fight), **P**ersonnel (availability of qualified people for operations), **F**acilities (infrastructure and physical resources), and **P**olicy (national security, international relations, and other factors impacting military operations; Rainey, 2024). The outcomes of this analysis are change recommendations, which are implemented throughout the Army and incorporated into training.

As doctrine and manuals evolve, course curricula and resources must be updated. Identifying the materials that need to be changed and updating these materials requires extensive effort from the curriculum developers, which subtracts from effort that could be devoted to the development of new resources or tools. This task is particularly challenging since training materials encompass text documents, spreadsheets, slide presentations, and videos of different formats; and the changes vary in scope and granularity. However, recent advances in machine learning and artificial intelligence (AI) offer promising methods which may significantly reduce the effort of curricula revision by quickly identifying sections of training relevant to each specific change and suggesting updates (Kelly & Smith, 2024).

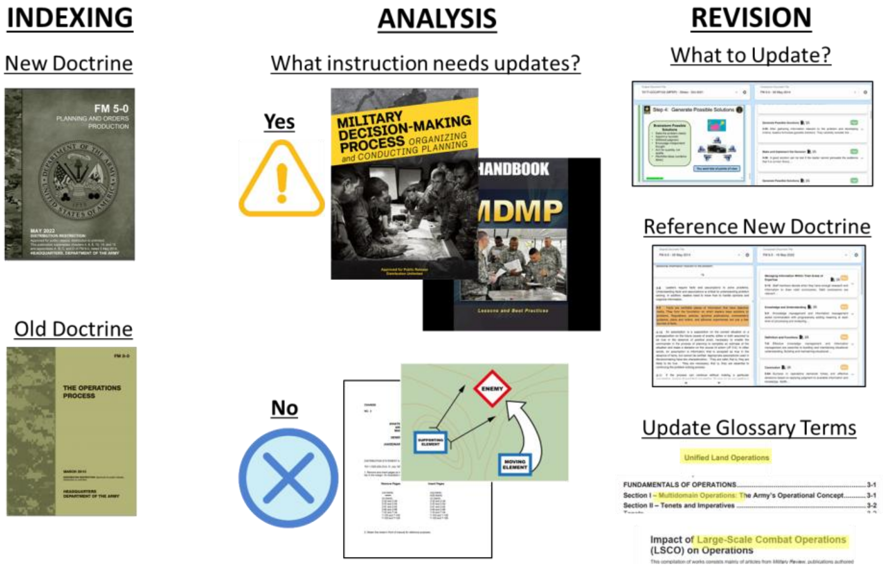

INDEXING
New Doctrine
FM 5-0
PLANNING AND ORDERS PRODUCTION
Old Doctrine
THE OPERATIONS PROCESS
ANALYSIS
What instruction needs updates?
Yes
MILITARY DECISION-MAKING PROCESS ORGANIZING and CONDUCTING PLANNING
HANDBOOK
MDMP
No
ENEMY
SUPPORTING ELEMENT
MOVING ELEMENT
REVISION
What to Update?
Reference New Doctrine
Update Glossary Terms
Unified Land Operations
FUNDAMENTALS OF OPERATIONS
Section I – Multidomain Operations: The Army's Operational Concept
Section II – Tenets and Imperatives
Impact of Large-Scale Combat Operations (LSCO) on Operations

Figure 1. Stages for ARC Content Revision Workflow

The AI for Revision of Content (ARC) project is primarily focused on updating content for Army training and education. Given the scale of Army learning programs, this involves multiple related problems (see Fig. 1):

1) *Indexing (What Content)*: Identify and represent meaningful changes to doctrine or other source materials;
2) *Analysis (Where to Update)*: Find lesson materials are affected by changes, and where in those documents;
3) *Revision (How to Update)*: Update content more quickly and/or more reliably to use new doctrine or practices.

Research on ARC started with an emphasis on the second challenge (Analysis): finding and triaging what needs to be updated, often out of thousands of pages of lessons and slides. Based on this foundation, we also researched tools for the other problems (Indexing and Revision) ARC does not directly focus on generating entirely new lessons or slides, so our research complements ongoing research that uses AI to build new content (AFIT, 2025).

This paper describes the rationale for ARC, technological advances needed for ARC capabilities, and the results from cycles of iterative user testing designed to identify needs, prioritize features, and evaluate existing ARC prototype. Testing and revision was conducted across more than one year, starting with a series of alpha tests conducted via MS Teams (guided testing to show capabilities) and then a second series of hands-on beta tests conducted with six Army training centers. User acceptance surveys across centers show positive ratings by Army training developers for ARC, but also demonstrate that training centers need a workbench of tools, rather than any single AI tool, to address their broad range of needs. Contributions from this work span both technical innovations (e.g., normalization of hybrid search results) and empirical (e.g., identifying key differences between training communities that require different kinds of AI assistance).

BACKGROUND

ARC combines classical natural language processing (NLP) technologies with newer large language model (LLM) capabilities. Classical NLP and LLMs have complementary advantages: classical techniques tend to be faster and can run efficiently across a large space of documents, while LLMs can follow task descriptions (prompts) to conduct complex analyses without requiring a specially-trained model. As a result, our research has combined both approaches, sometimes for the same task. For example, to parse metadata and passages out of documents, we relied on fast traditional parser for PDFs in a commonly used format (e.g., FM's), but can use an LLM to assist parsing a PDF with an unfamiliar format. Combined approaches are increasingly popular for ETL (Extract, Transform, Load, see: unstructured.io, PyMuPDF, etc; Adhikari, N. S., & Agarwal, 2024). Given that certain Army documents are highly structured (e.g., Field Manuals, TDC lesson plan exports), there are advantages for speed, reliability, and computation to try to use traditional parsers before heavier LLM approaches.

Semantic Search. Large-scale semantic search and comparison relies on traditional high-dimensional numerical vector representations of semantics called embeddings (Ambite et al., 2019; Camacho-Collados & Pilehvar, 2018; Liu et al., 2020). Embeddings are computed in a self-supervised fashion from very large corpora using pretext tasks, such as masked word completion and next-sentence inference, using transformer architectures (Vaswani et al., 2017). Embeddings can capture the semantics of words (Devlin et al., 2019; Wang et al., 2020), sentences (Reimers & Gurevych, 2019), paragraphs (Li et al., 2015), or documents (Dai et al., 2015), abstracting away from specific words or tokens in the text. A major advantage of text embeddings is that they can find similar textual units using vector similarity through nearest-neighbor algorithms (Fakhraei et al., 2019; Aumüller et al., 2020) and quickly find related text based on this semantic similarity (Chandrasekaran & Mago, 2021). While LLMs can produce embeddings, these are slower to compute and not necessarily better than earlier embedding models (e.g., SentenceTransformers, BERT). Advances in search frameworks have added "hybrid search" which combines semantic embeddings with other search indices (e.g., keyword-based), to balance information from different types of similarity criteria (Xian et al., 2024).

LLM Analysis. LLM models have enabled new capabilities in three areas: content adaptation (revision), content generation (drafting), and content assessment (reviewing). As noted earlier, content adaptation and assessment are relevant to the current work. A recent review of LLM use by educators reports that K-12 teachers primarily used tools like ChatGPT to extend (add), revise, and trim down content (Karataş, F., Eriçok, B., & Tanrikulu, 2025). While experienced teachers were more likely to modify materials more extensively, there were no significant differences between grade levels, which suggests that the training developer may influence LLM usage more than the specific content. However, given the significant differences between K-12 and military training, the type of training content may influence how (or if) AI is appropriate. In particular, LLM hallucinations occasionally still introduce errors or unsupported statements (Tonmoy et al., 2024), which is problematic for safety-critical training or other precise but hard-to-validate content.

Reviewing and assessing content has been highly active since the release of ChatGPT. While prior NLP models needed to be trained by machine learning experts to check content against a new rubric, LLM instructions with N-shot prompts have shown useful results across a variety of grading and metadata tagging tasks (Asthana et al., 2023; Wu et al., 2025). In military training, we have observed collaborators developing prompt collections which include analyses to screen that learning objectives meet specific requirements for format and verbs (Fortuna, in progress). Verification of training content against source materials or against formatting requirements could be particularly valuable for military training, by reducing delays caused by multiple cycles between human reviewers (e.g., small changes that take a long time due to finalize). However, reviewing content has primarily focused on rubrics for a single document. For ARC, we are primarily interested in comparing two or more documents, which has not been studied significantly with LLMs.

LLM Policies and Access. From 2023 to the present, military policies for LLMs have evolved, with two trends. Initially, military policies were reticent to permit commercial LLM models, meaning that open source models were prioritized (e.g., Mistral and LLaMA; Jiang, et al., 2024; Touvron et al., 2023). As a result, research on ARC first prioritized engaging with models such as the Army TracLM, which has trained and tuned models on corpora of Army documents (see: https://huggingface.co/TRAC-MTRY). Similarly, early access for systems such as the CamoGPT API offered a hosted LLM endpoint (CALL, 2025). As policies expanded access to commercial models (Vought, 2025), Army-approved commercial LLMs were tested on ARC problems, such as OpenAI GPT-4o-gov and Claude 3.5-gov through AskSage (OpenAI, 2024; Anthropic, 2024 www.asksage.ai). While not the focus of this paper (as these were not compared systematically), our impression was that models trained with Army documents showed possible advantages for generation or applying rubrics specific to the Army. However, commercial LLMs offer substantial advantages for throughput (e.g., running requests in parallel) and context window size (models able to handle large documents). Ideally, specialized military models could scale in a secure cloud comparable to general-purpose LLMs, so that systems such as ARC can pick the best tool for each task.

The second major trend has been small LLMs, including models small enough to run on a laptop CPU (e.g., LLaMA 3.2 1B; https://huggingface.co/meta-llama/Llama-3.2-1B). While these models are much less powerful, they can be run locally for security and for when network connections are unreliable. Testing with very-small models indicates that they are not useful for analyzing content meaning, but can be used for tasks that otherwise use brittle pattern-matching, such as extracting the title or date from a document whose format is unfamiliar. Small, specialized models may be valuable in the future, or general purpose variants may be available through built-in LLMs packaged with the operating system (e.g., Google Nano on AndroidOS, CoPilot on Windows). Policies and security standards must continue to evolve to leverage the full continuum of AI capabilities, to reduce servers and to speed up responses.

TECHNICAL APPROACH

ARC was designed to address three core questions:

- How can AI and ML be used to rapidly identify content affected by changes to doctrine?
- How can AI and ML increase diligence and consistency with glossaries, references, and other linkages?
- What technologies and user interfaces can effectively suggest simple updates for training documents?

ARC addressed these by developing three capabilities (Fig 1.): Indexing, Analysis, and Revision. Each one poses distinct research problems, where we prioritized general issues (e.g., how to quantify the amount of change between versions of a slide?) compared to incremental ones (e.g., how to extract data from a new PDF format?).

Indexing: Document Parsing and Representation

We developed a pipeline to parse common Army publications (e.g. Field Manuals, Army Doctrine Publications, lesson plans, presentations) into a structured format for content at a known unit of granularity (e.g. paragraph-level text, slide content) and associated metadata (e.g. page number, paragraph ID, etc.). Input documents may be in PDF, PPTX or DOCX formats which are then parsed using associated content extraction tools. Off-the-shelf parsers, however, introduce various errors (especially for PDFs) so we run postprocessing based on heuristics. We track page and slide numbers, as well as figure and table captions. Each document is split at an appropriate level of granularity (e.g., sentences, paragraphs, sections, chapters), to enable analysis at the right level of abstraction.

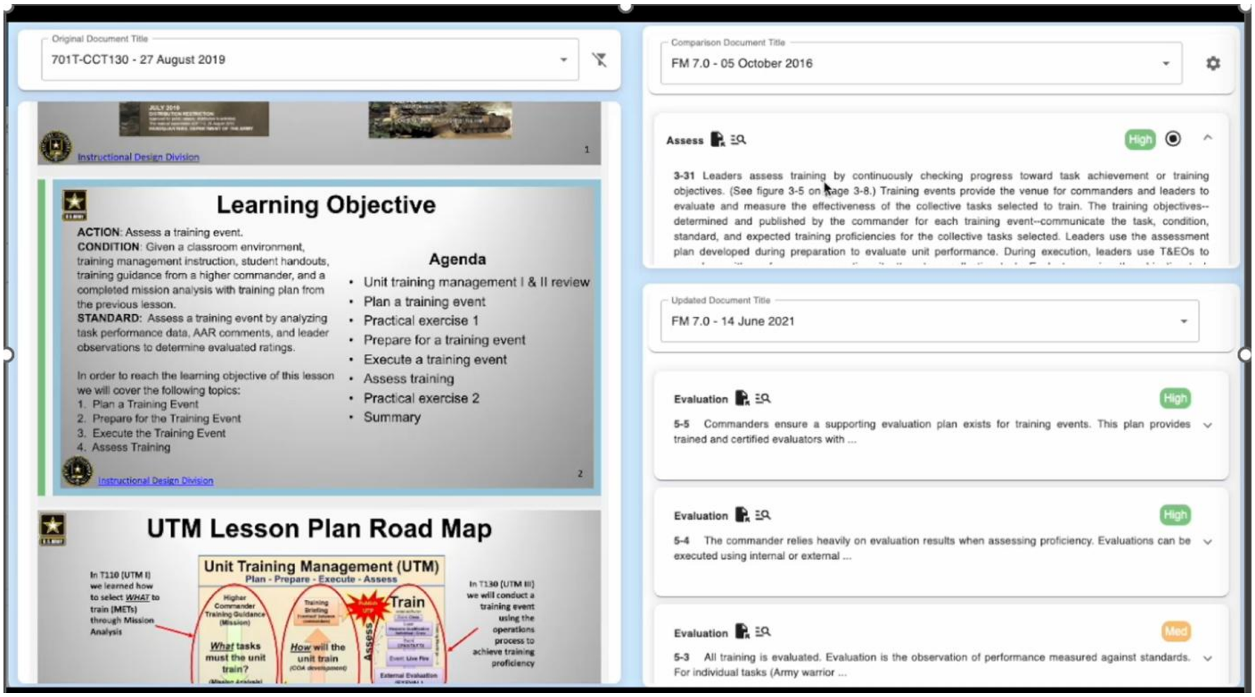
When ingesting a document, the system will infer or allow manual entry of document metadata. As shown in Table 1 (left), this metadata distinguishes between the "Document Type" (e.g., PDF vs. Docx) and the "Resource Type" (e.g., Doctrine vs. Lesson Plan). This is because while parsers care about the file document type, end-users want to filter by

Resource Type. The document name and version date are also stored as metadata. An optional edition field may also be used to specify a more specific version than the date alone (e.g., v1 vs. v2). Based on discussions with Army CoE training experts, we inventoried common resources used when updating training. While we focused on traditional materials (e.g., doctrine, manuals), these are not exhaustive, particularly for rapidly changing content (e.g., new UAV tactics observed in public web posts). The most prevalent resources (core resource types) were prioritized:

- *Resource Types (tested)*: Doctrine (e.g., FM, APD), Lesson Plans (e.g., TDC, docx), Presentations (pptx).
- *Other Types (not yet tested):* Memos, newsletters (lessons learned), info papers, white papers, academic publications, technical manuals, technical papers (e.g., cybersecurity), spreadsheets, outlines for instructors.

Table 1: Extracted Data from PDF File (Left: document metadata. Right: extracted unit)

name: ADP 3.0, *version*: 31 July 2019, *resourceType*: Doctrine, processedOn: Dec 03, 2024 at 22:49, *aircoeeDocId*: adp-3.0-31-july-2019,	*type*: TEXT_PARAGRAPH, *pageNumber*: 11, *yCoordinate*: 177.49, *pageLabel*: 1-1, *chapterNumber*: 1, *chapterName*: MILITARY OPERATIONS, *sectionHeader*: MILITARY OPERATIONS, *content*: This chapter discusses military operations, … readiness., *text*: This chapter discusses military operations, …readiness. MILITARY OPERATIONS



Original Document Title
701T-CCT130 - 27 August 2019

Instructional Design Division 1

Learning Objective

ACTION: Assess a training event.
CONDITION: Given a classroom environment, training management instruction, student handouts, training guidance from a higher commander, and a completed mission analysis with training plan from the previous lesson.
STANDARD: Assess a training event by analyzing task performance data, AAR comments, and leader observations to determine evaluated ratings.

In order to reach the learning objective of this lesson we will cover the following topics:
1. Plan a Training Event
2. Prepare for the Training Event
3. Execute the Training Event
4. Assess Training

Agenda
- Unit training management I & II review
- Plan a training event
- Practical exercise 1
- Prepare for a training event
- Execute a training event
- Assess training
- Practical exercise 2
- Summary

Instructional Design Division 2

UTM Lesson Plan Road Map

In T110 (UTM I) we learned how to select *WHAT* to train (METs) through Mission Analysis

In T130 (UTM III) we will conduct a training event using the operations process to achieve training proficiency

Unit Training Management (UTM)
Plan - Prepare - Execute - Assess
Higher Commander Training Guidance (Mission)
What tasks must the unit train?
Training Briefing
How will the unit train
(COA development)
Assess
Train
Event: Live Fire
External Evaluation

Comparison Document Title
FM 7.0 - 05 October 2016

Assess High

3-31 Leaders assess training by continuously checking progress toward task achievement or training objectives. (See figure 3-5 on page 3-8.) Training events provide the venue for commanders and leaders to evaluate and measure the effectiveness of the collective tasks selected to train. The training objectives--determined and published by the commander for each training event--communicate the task, condition, standard, and expected training proficiencies for the collective tasks selected. Leaders use the assessment plan developed during preparation to evaluate unit performance. During execution, leaders use T&EOs to

Updated Document Title
FM 7.0 - 14 June 2021

Evaluation High
5-5 Commanders ensure a supporting evaluation plan exists for training events. This plan provides trained and certified evaluators with ...

Evaluation High
5-4 The commander relies heavily on evaluation results when assessing proficiency. Evaluations can be executed using internal or external ...

Evaluation Med
5-3 All training is evaluated. Evaluation is the observation of performance measured against standards. For individual tasks (Army warrior ...

Figure 2: ARC Comparison View - Slide Deck (left) compared to Relevant Doctrine (right)

Army PDF documents for doctrine have a standardized look and feel, such as clear chapter names, a table of contents, and page numbering. However, due to their PDF format, documents that look similar visually often have irregularities in their machine-readable structure (consistent content, inconsistent parsing). On the converse, libraries for MS Office documents (docx, pptx) parse their XML structure reliably, with the caveat that documents tend to have a less regular structure (consistent parsing, inconsistent content). This means that ARC often ingests PDF documents very cleanly, but a small percentage fail due to structural issues and need to be parsed using more general (and less tidy) methods. By comparison, DOCX files seldom have useful structure (e.g., no table of contents or section tags), so content is split based on heuristics (e.g., consolidate small chunks, split up too-large chunks). Surprisingly, PPTX text was easiest to parse: a single slide is an appropriate text length, and by analyzing the positions of text boxes it is possible to join them to approximate the order that a human would read them. The main challenge for slide decks is missing metadata: slides often lack release dates or version numbers, requiring manual input of this data.

As shown in Table 1 (right), each unit is stored in a database with a specific type. A typical search over a document focuses on the Text Paragraph as the unit, as well as other text-heavy units such as a Slide or a Table unit. A Figure (i.e., caption and related text) is also a unit. Chapters and Sections are also stored, but ignored for search purposes. However, when known, all searchable units have index fields for their chapter and section. Moreover, while the "content" field stores the raw text, the content used by the search algorithm considers both the text passage and its chapter and section ("MILITARY OPERATIONS" in the "text" field of Table 1, right). The page is stored by the document page (e.g., PDF reader number) and also by the label shown on the page, if available (e.g., 5-1). These are often quite different. Finally, to assist with sorting the text, the y-index (vertical location) of the text block is stored. This allows quick review of text in the order it shows up on the page, from top to bottom. Without this, it would be hard to sort text from a PDF which does not store text blocks in the order they are displayed (fairly common).

Not all data fields are required for each text unit stored by ARC, with the most critical being the Type, Content (for displaying to the user), Text (for hybrid search), and Page Number (to enable opening the source document to the page for that unit). All documents uploaded to ARC save a PDF equivalent, used for quick previews in a web-browser. As shown in Fig. 2, each unit can be shown as a series of cards either as as search results or presented in-sequence to reconstitute the document. The section title (e.g., "Assess" for the card at the top of Fig. 2, right) is followed by an icon to open the full PDF file to the given page (file icon) and a second icon which quickly shows an HTML preview.

Semantic Match Analysis: Identity-Max Normalization for Hybrid Search

Each unit of text from a document is indexed in OpenSearch, a search engine which enables efficient retrieval of text queries (https://github.com/opensearch-project/OpenSearch). OpenSearch finds relevant matches between passages from any two documents and for queries searching across a broad set of documents. In ARC, OpenSearch is configured for *hybrid search* which combines both a traditional lexical (Okapi BM25) method and a transformer embedding model (SentenceTransformers; all-MiniLM-L6-v2). Hybrid search balances retrieving semantically similar results while not neglecting the keyword-based similarity, which may be important for new acronyms or terms not understood by the semantic model and for favoring matches which use exact matches for terms. Hybrid search scores combine of the lexical and embedding search scores (weighting the semantic search method higher than the lexical search (L2 normalized; Harmonic Mean weighted as 0.7 and 0.3).

The hybrid search produces a match score and ranking for the results, which were generally satisfactory when rated by SME's (i.e., top results were typically the best-match from the set of possible matches in a document). However, search scores are not normalized: while they produce a rank-order, a "good match" score might be 0.28 for one search but only 0.126 for another. Worse, due to optimized ranking methods (e.g., approximate nearest-neighbor), the range of scores are not consistent even for the same query and best-match. For example, passage A may return result B with a score of 0.05 when searching over all documents, but B with score of 0.2 when searching across a subset (e.g.. a single target document and its units). This poses a problem, because ARC aims to not just find the best match but it also categorizes results with user-defined thresholds into high/medium/low matches, to help find areas of documents that changed between versions.

We developed a novel normalization technique we will call *Identity-Max Normalization (im_norm)*, which is simple to implement and should be useful for a variety of systems. We empirically established that when given a filter criteria (a search space with specific target documents), OpenSearch consistently produced a stable, positive upper bound for the range of scores for the results. The key idea is that if we obtain the upper bound for a filter criteria (a max norm), this may be used to normalize all the scores in the results and then users can set predictable thresholds to partition the results by match quality score (in a range from 0-1). The problem then is reduced to finding the upper bound score for a given filter criteria (search space). Intuitively, an exact match should always be the best match, so we artificially add the exact query document (the "identity" result) into the search space for every query. As expected, this would make OpenSearch match the query document to itself as the top match and assign it a theoretical maximum score for that filter criteria. This top score is then used to normalize all the remaining matches to produce stable ranges for the scores using the formula: $im\_norm_i = max(0, 100*(raw\_score_i /identity\_max\_score))$

The raw_score_i refers to the unnormalized score set by OpenSearch for any search result i and the *identity_max_score* refers to the score of the top result which is always the query document itself. Adjustments were cases where scores fell below zero, but in practice the maximum was above 0 and results below zero appeared to be batch matches, regardless of the search space. While additional systematic testing of this normalization technique would be valuable, in practice this method produced highly intuitive scores, with over 90% typically being a very close match and 80% being a good match (very similar content or a revision of the same passage), for passages that were paragraph-length. Additional research explored replacing the default semantic model with an Army-specific small LLM (TracLM 7b; Ruiz & Sell, 2024), which our team converted to an embedding model using LLM2Vec (BehnamGhader et al., 2024). However, due to the limitations for indexing documents on a local device (e.g., laptop), we did not use the Army-specific LLM for user testing. Further research is needed to compare results from Army fine-tuned embeddings.

Change Analysis: Triaging Potential Areas to Update

Specialized user interfaces to enable comparing training documents, both in pairs (e.g., previous vs. latest doctrine) and triads (e.g., a slide deck against its older vs. current doctrine source). Analyses are run at the passage-level and document-level.

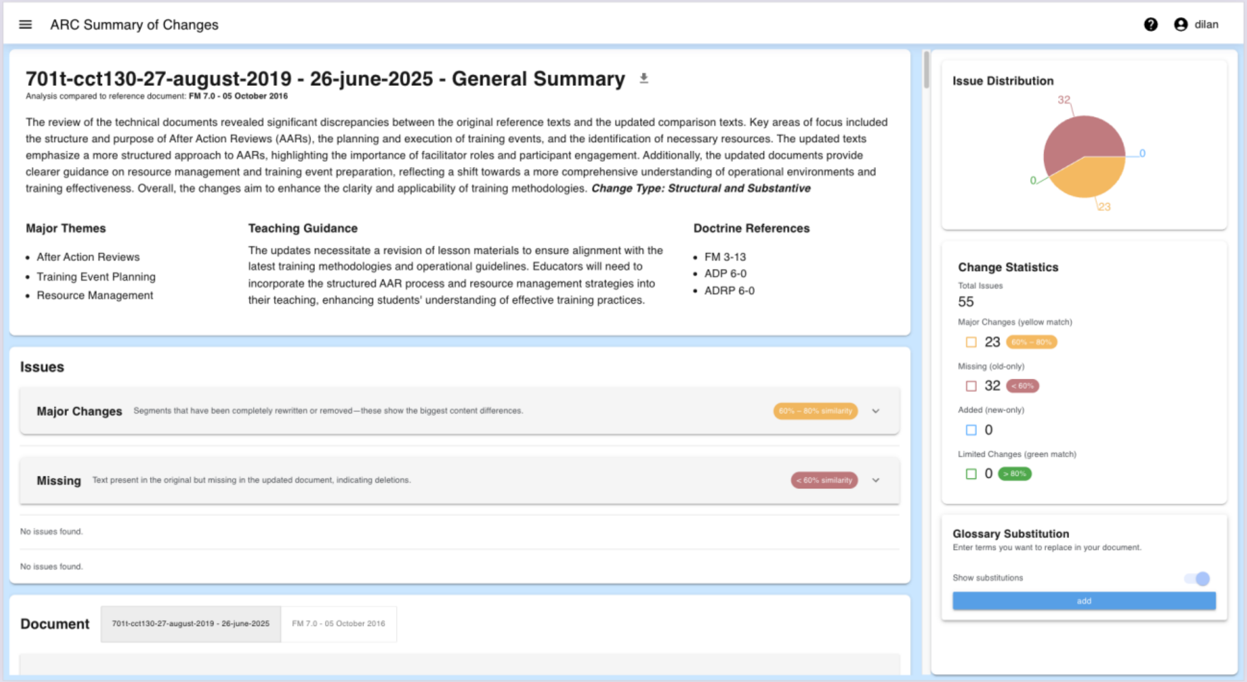
Passage Comparison. For any pair of two passages, ARC can analyze them in three ways: semantic (as detailed above), text differences (word changes), and LLM difference prompts. Semantic passage analysis estimates the change in *meaning* of a passage, in terms of its relevance to another passage (i.e., talks about similar things). This is used extensively by ARC to color-code and filter changes. A normalized, consistent scoring system means that users can create thresholds to return only passages with slight changes, for example (e.g., not an exact match, but still a high match). In Fig. 2, the results of these categories can be seen in two places. First, all matches on the right are tagged with their relevance (green=High, yellow=Medium, red=Low). Second, the current slide has a green line to the left, indicating its relevance to the best-match from the doctrine document. Highlights can also be turned on for the whole document, so that a slide deck, doctrine, or lesson plan shows all passages with a color-coding highlight. It can also be used to filter the whole document preview, so that only passages with certain levels of matches are displayed to review. These are intended to help draw attention to areas which are no longer included in new doctrine, content that has moved to other doctrine documents, and content that was significantly revised between doctrine versions. Text differences can be enabled to display a "*diff*" between passages, displaying a view similar to track-changes in a document draft. Diffs are used to drill-down to see word-by-word what changed between two passages, ideally ones with at least a medium match score (e.g., close enough to meaningfully review).

The LLM Difference Prompt performs a deeper analysis of the *impact* of the differences between two passages. The prompt describes criteria for change categories, then generates a list of changes which are each flagged with their main category. High impact changes are those categorized as either a change in Action (recommended action or behavior) or Evaluation (change in assessment or judgement). These are the most important, because they detect issues that the semantic comparison might miss. For example, a long medical passage that lists symptoms then concludes "Drug X is the recommended treatment" versus "Drug X is no longer the recommended treatment."). Semantically, these passages are genuinely similar. However, an LLM can flag these subtle changes. Medium impact changes include changes to Claims (smaller points) and Fact Changes (different evidence), Minor changes include Additions which do not fit other categories and Phrasing changes. As LLM prompts are resource intensive and documents may be long (e.g., hundreds of pages), these are only performed when the user initiates this analysis. The Summary of Changes panel (Fig 3) also uses an LLM to summarize overall differences across substantially changed matches. A button is available to open a custom prompt window as well, so that the user can compare passages with their own prompt.

Document-Level Comparison. Building on passage analyses, ARC analyzes whole documents in two panels:

1) Comparison Panel (Fig 2): Shows the main document on the left panel, and either one or two documents on the right panel. The main document can be scrolled to view all text and highlights can be enabled to show the semantic score for the best match from the comparison document on the right. Passages can be filtered by match quality and/or by text keywords. The right hand panel also includes a search capability to look for the best matches across all documents, which a content developer might use to find a replacement reference for content which has moved to a new doctrine owner (e.g., the definition of a term moved from FM 6-0 2014 to FM 5-0 2022).

2) Summary Panel (Fig. 3): Shows an initial summary of changes, generated by an LLM analyzing a sample of partially matching sections (prioritizing Medium matches, which often represent significantly-revised text). A summary of matches is shown as a pie chart (upper right), and matches can be browsed in an Issues list, where they can be analyzed using the LLM Comparison or other LLM calls (currently AskSage gpt-4o-min-gov, but flexible across models). The "Document" section allows browsing the document, tagged with comments for each issue. On the lower right panel, a content developer can modify a Glossary Substitution table, which performs a global search-and-replace on the Document. Updated Document text can be saved to a docx file, with comments and glossary substitutions applied.



ARC Summary of Changes

dilan

701t-cct130-27-august-2019 - 26-june-2025 - General Summary

Analysis compared to reference document: **FM 7.0 - 05 October 2016**

The review of the technical documents revealed significant discrepancies between the original reference texts and the updated comparison texts. Key areas of focus included the structure and purpose of After Action Reviews (AARs), the planning and execution of training events, and the identification of necessary resources. The updated texts emphasize a more structured approach to AARs, highlighting the importance of facilitator roles and participant engagement. Additionally, the updated documents provide clearer guidance on resource management and training event preparation, reflecting a shift towards a more comprehensive understanding of operational environments and training effectiveness. Overall, the changes aim to enhance the clarity and applicability of training methodologies. ***Change Type: Structural and Substantive***

Major Themes

- After Action Reviews
- Training Event Planning
- Resource Management

Teaching Guidance

The updates necessitate a revision of lesson materials to ensure alignment with the latest training methodologies and operational guidelines. Educators will need to incorporate the structured AAR process and resource management strategies into their teaching, enhancing students' understanding of effective training practices.

Doctrine References

- FM 3-13
- ADP 6-0
- ADRP 6-0

Issues

Major Changes Segments that have been completely rewritten or removed—these show the biggest content differences. 60% – 80% similarity

Missing Text present in the original but missing in the updated document, indicating deletions. < 60% similarity

No issues found.

No issues found.

Document 701t-cct130-27-august-2019 - 26-june-2025 | FM 7.0 - 05 October 2016

Issue Distribution

Change Statistics

Total Issues
55

Major Changes (yellow match)
□ 23 60% – 80%

Missing (old-only)
□ 32 < 60%

Added (new-only)
□ 0

Limited Changes (green match)
□ 0 > 80%

Glossary Substitution

Enter terms you want to replace in your document.

Show substitutions

Figure 3: Summary of Changes analyzes slide deck aligned to older field manual vs. newer version

Both document-level analyses can also be evaluated in three modes:

- *Two-Document*: Compares two documents, where semantic matches are determined by how well a passage has a match in a given document. Typically used to compare two versions of the same document (e.g., reviewing changes to revised doctrine). It can also be used with a slide deck or lesson to quickly review associated doctrine.
- *Multi-Document - Doc vs. Multiple Refs*: Compares the main document against two reference documents. A use case was to see results from two current doctrine documents selected, to review materials most relevant to the current slides. Color coding for matches can be based on either documents' best match. Future versions will expand the number of files, which is not limited by the algorithms but requires additional UI design and tests.
- *Multi-Document - Doc vs. Before/Afte*r: A second multi-document mode is to compare a lesson or slide deck against different versions of the same doctrine. For example, when a slide deck was authored with an older reference but must be updated to the new materials.

When in Before/After mode, the default shading and issue detection is done using a two-stage semantic match. First, each document passage searches for the best match in the old doctrine. Then, the best match from the old doctrine is used as a query to find the best match in the new doctrine. Issues are rated based on the *decline* in match quality from the original reference. This is because there is no problem if a slide with no doctrinal content also has no doctrinal content in the new reference (slide content might be a warm up, a "Questions" slide, or other non-doctrinal content). The issues are most important to flag when a slide had a good match in old doctrine, but there are substantial changes or no equivalent in the new doctrine (e.g., did the content move? was it outdated?). A smaller, related issue is when comparable content exists but the location has changed (e.g., different chapter, requiring a citation update).

USER STUDY DESIGN

ARC employed a design-based research, where Army training centers[1] and SME's were engaged across three phases:
1) *Familiarization*: Initial engagements that included virtual slide deck briefings, presentations at the 2024 Army University Learning Symposium, and question-answer sessions to gather priorities and needs;
2) *Alpha Testing and Corpus Collection*: Seven interested training centers opted-in to participate, and most shared sample training materials which were loaded into ARC. Guided testing was conducted over virtual teleconferences, where the initial ARC prototype capabilities were shared on sample documents from the corpus. Attendees could ask questions and direct research staff to perform certain actions in the system to test it. Voluntary surveys for user

[1] Note: In this paper, we will refer to participating groups as "centers" for ease of reading, but the engagement included grain sizes that included centers, schools, and agencies.

acceptance and feedback were available following each session, which were also used as formative evaluation indicators for the progress on different components (Kirkpatrick & Kirkpatrick, 2006; Venkatesh et al., 2012).
3) *Beta Testing and System Revision:* Hands-on beta testing of ARC across six Army training centers, consisting of one or more sessions where a set of SME's tested ARC on test machines set up for this purpose. Documents relevant to each center were pre-loaded into ARC, and in some cases were also added by SME's for testing. As with Alpha testing, a voluntary survey was available to evaluate the system and provide feedback.

Revisions to ARC were conducted following Familiarization (Summer 2024) and after Alpha testing (Fall 2024) to form the first hands-on prototype. After this time, ARC was revised based on feedback between Beta test site visit trips, leading to three additional cycles of revisions and capability development (Dec 2024, Feb and Mar 2024, April 2024). Survey feedback and research team notes were consolidated into a unified spreadsheet where qualitative themes were grouped to determine the prevalence of requests for features and/or reported issues. As a result, user tests at different sites included additional features and bugfixes, which were included in Beta tests as soon as possible to maximize feedback (i.e., prioritizing capabilities to review, rather than polish). Most notably, LLMs were not user-tested until the final two centers, as the project was reviewing evolving guidance on allowable LLMs.

The survey used for this research was adapted from a version of the Unified Theory of Acceptance and Use of Technology (UTAUT; Venkatesh et al., 2012) that was previously used by our team in the Rapid Adaptive Content Registry project (RACR; Nye et al., 2022). This survey presents screenshots of different major parts of the system, including the Login, Home area (file list), Comparison View, and Summary View. SME's rated questions on Ease of Use ("I found the <x> easy to use"), Performance Expectancy ("Using <x> will increase my productivity."), Attitudes ("Using <x> is a good idea."), and Intent to Use ("How often would you expect to use the system, if you were working on such courses regularly (e.g., at least one per month)?"). For the system overall and for each section, three open response questions asked: "Were there any bugs that you saw?", "What would you like best about <x>?", and "What would you change about <x>?" Users were also asked "How could you see yourself using the system?" to determine what use-cases were most relevant. Open-responses were categorized and sorted by frequency.

Beta testing received a much higher number of responses than Alpha testing (N=6 for Alpha, N=29 for Beta). In part this was due to an overall higher number of active testers: while Alpha test sessions typically included 4 to 8 attendees, but often only one or two would actively guide the process. At most sites, more than one Beta test session was conducted (e.g., a morning and an afternoon), where test machines were either used individually (4 participants per session) or in pairs (e.g., alternating testing vs. observing, up to 8 in a session). While exact estimates of survey response rates are not possible, the research team estimates that approximately 20% of Alpha test attendees completed a survey versus over 80% of Beta testers. Across Beta test prototypes, there were 9 responses for the first revision, 6 for the second revision, and 14 for the final major version. Responses were anonymous, but optionally users could select all job roles that apply to them with respect to content. Across the two phases, participants identified their roles as: Course Developer (Alpha 50% vs. Beta 79%), Instructor (50% vs. 48%), Education Specialist (67% vs. 14%), IT / Software Developer (0% vs. 3%), Project Manager (33% vs. 21%), Data Analytics (17% vs. 10%), Quality Assurance (33% vs. 14%), and Researcher (33% vs. 21%).

RESULTS AND DISCUSSION

User ratings for ARC were positive, ranging from 4.48 to 5.67, where 6 is the highest rating. Ratings were higher for Alpha testing, but this is not statistically significant due to the smaller number of responses. In general, Alpha testers did not need to figure out the user interface directly, so their ratings for ease of use (Effort Expectancy) would be expected to be higher. Despite being positive overall, open response comments still reported that the system needed to put key high-value features up-front. For example, tools in the Summary Panel (Fig 3) were often more useful, but users often started by browsing documents using the Comparison View (Fig. 2). This was in part due to rapid design iterations: newer features were often more helpful, but were listed last in the options. Across both study phases, the highest ratings were that ARC was a good idea- users reported strong demand for capabilities to revise content faster and reliably, particularly in the face of limited staff and rapidly-changing operations.

Given that the Beta test data offers stronger insights into the system components, only results from the Beta tests will be presented for the remainder of the results section. Intentions to use ARC were high. In response to "How often would you expect to use the system, if you were working on such courses regularly (e.g., at least one per month)?": 34.4% (10 users) reported "Every course that I made", 20.7% (6) "Most of the time", 17.2% (5) "About half the time", 17.2% (5) "Occasionally (1 in 3 courses)", and only 10.3% (3) selected "Rarely (once or twice a year)" or "Never." This means that 72.4% (21) reported that they would use ARC at least half the time when working to revise a course. While Beta testers appeared to consistently complete the survey, it is feasible that they might be less likely to report that they would use ARC. However, even if that were true for all such non-respondents, the percentage who would use ARC half the time would still be expected to be above 60% (as we are confident that well over 80% of hands-on testers completed the survey, given the number of seats available in each session).

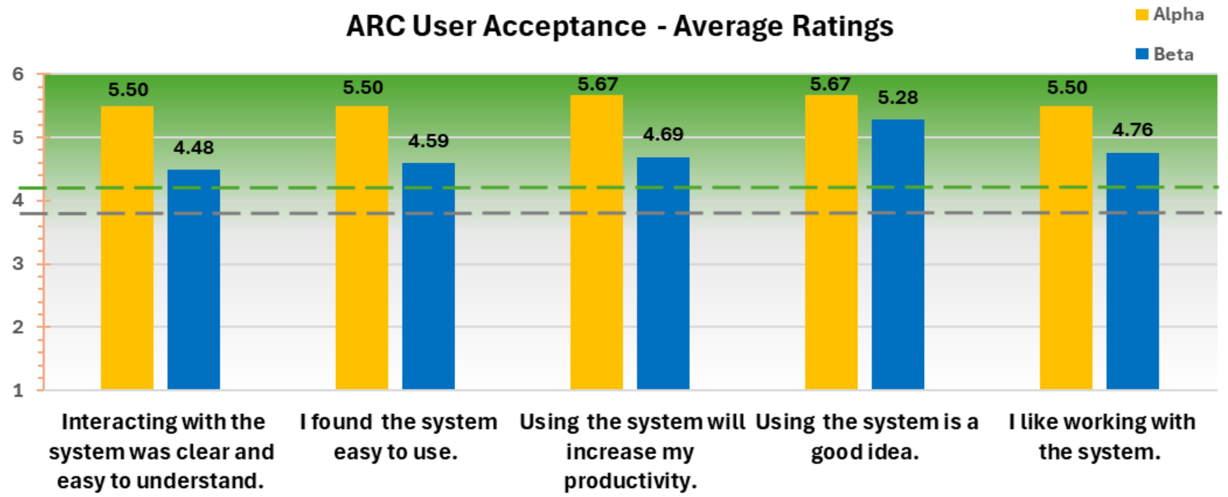


Figure 4: User acceptance ratings from Alpha tests (N=6, yellow) and Beta tests (N=29, blue). Scale from 1=Completely Disagree, 2=Disagree, 3=Slightly Disagree, 4=Slightly Agree, 5=Agree, 6=Strongly Agree. Scores above 4.2 (green line) are consistently positive, while below 3.8 (grey line) are equivocal or negative.

To consider the features that users found most valuable, Table 2 lists the two questions asked for every component and subcomponent (*Clear and easy to understand* and *Good idea*). As not all subsystems were fully developed across all Beta tests (e.g., the Summary Panel), some components are not included in the table, but will be considered in the open response results. As with the system overall, ratings were positive. However, some components have substantial differences between the value of the feature overall versus the ease of use. The Document List, Importing, and Preview panel show somewhat lower ratings for ease of use, but still comparable (e.g., rated that they were useful but could use some polish). The main issue for the Document List was that the search list was too picky (needed looser matching to find documents by name). For Importing, users raised issues that certain documents had suboptimal chunking so passages were not well-formed and handling of tables should be improved. Preview panels were rated as easy to use, but some users noted that they would already open documents in a second screen and they would not need a preview.

Table 2. Ratings of major ARC subsystems from Beta test responses (in the form "mean (variance)")

	Document List	Import & Indexing	Comparison Panel	Highlighting Whole Doc	Preview Panel
Clear and Easy	4.55 (1.16)	4.68 (1.02)	4.34 (1.32)	4.62 (1.40)	4.59 (1.45)
Good Idea	5.14 (1.22)	5.18 (0.78)	5.48 (0.72)	4.28 (1.62)	4.97 (1.38)

However, the Comparison Panel shows a wide difference: it is rated the most valuable, but also with the most issues. As this is where match results are observed, separate survey questions asked about the match quality. Users rated the "The best match for each passage was typically correct" as 4.38 (1.42) and "The best match for each passage typically had the correct color" as 4.41 (1.52). Open responses indicate that the most common match issue was over-rating lower quality matches, which might be possible to improve with improved semantic models (e.g., tuned on Army data). Despite this, users typically found the comparison panel useful but had many (sometimes opposite) suggestions for improvement. On the converse, highlighting the whole document was rated slightly higher for usability than for its value as a feature. Ratings for this feature decreased for Beta tests on Revision 3 (final revision), likely because the new Summary Panel was expanded- many users reported that they would be less likely to browse the whole highlighted document if the summary panel could give them a more targeted set of issues.

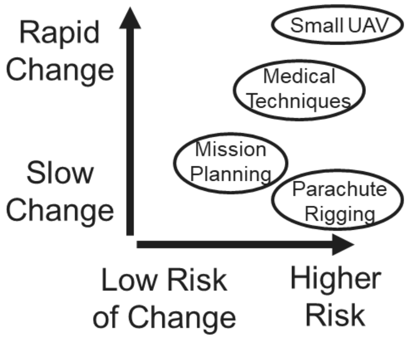


Figure 5. Risk of Change vs. Speed of Change for Training

Among open response data, themes for ARC strengths (what did you like best?) found three themes. Users reported it was faster to find and update content from references ("cuts down on tracking references. Ease and speed of the system will boast efficiency."; "Its ability to search rapidly."). Analyzing and comparing documents for changes was a second theme ("Easy to analyze differences between documents"; "comparison ability"). Finally, the Summary Panel was noted in the later revision ("ability to summarize outcomes"). This indicates that ARC was useful for activities the system was primarily designed for: to rapidly find changes in documents, then to update lessons which are impacted.

ARC limitations fell into two distinct categories: areas that ARC can improve (e.g., simplified workflows, optimizing match results) versus areas that ARC was not currently designed to address (a mismatch in capabilities vs. needs). One most-wanted feature for users was a large, pre-loaded repositories of documents (""database of all officially published doctrine", "Access to repositories (CAR, TDC, APD, JEL+, CJCS, CATs, ATN)"). Multiple users also wanted a sync and notification functionality to automatically add new versions as soon as they are released. The lack of a live master repository and API for documents is a recognized issue for the Army, which affects not just ARC but a wide range of AI efforts (e.g., AI2C and AskSage chat agents). For users who gave the lowest ratings, the second issue was more relevant: some training developers were focused primarily on generating lesson content, rather than analyzing it. Among all testers who rated "increase my productivity" negatively, 3 out of 4 were primarily interested in generating lessons. These SME's reported deep knowledge of existing doctrine, so they seldom directly compared new references versus older ones ("have it generate information. Simply comparing information has little value.", "something that grabs the old lesson plan and builds the new one"). While ARC can be complementary to lesson generation, it was not the main focus of this phase.

Themes in response to "How could you see yourself using the system?" were the most interesting, as they differed substantially by the users' self-reported roles. Responses reflected two qualitatively different types users: Single-Course Developers (e.g., Instructors) and Multi-Course Developers / Curriculum Developers (e.g., Instructional Systems Specialists). Single course developers presented use cases which we will term *Bottom-up course revision*, as they regularly start by updating their slide decks (e.g., for the upcoming week), then revise lesson plans sometime later to reflect changes. They focused on rapid updates to reflect new information ("compare powerpoint to doctrine to further expand on key bullet points"; "keeping a pulse on current changes to regulation and updating the lessons"). Multi-Course Developers focused on *Top-down course revision* and their primary use case for ARC was to analyze and triage many documents ("review/update all 500+ lesson plans annually", "I would use, but it is better suited for my Developers to build, review and update doctrine."). Overall, different users want to do different things with ARC, based on their role. Instructors were focused on rapid changes to individual lessons and slides, including updates days before they present the slides. Instructional developer specialists wanted to use and expand capabilities to rapidly overview changes across many lessons.

CONCLUSIONS AND FUTURE DIRECTIONS

While not directly reflected in the data, Beta tests and discussions across six distinct training communities showed that their needs for AI tools were strongly influenced by two aspects of the training material: Risk of Change and Speed of Change. Fig. 5 shows a notional model of where different types of training topics appeared to fall on these criteria. Risk of Change indicates a clear added risk for making a change to the training (which is not necessarily the same as the risk of the domain itself). For example, mission planning and leadership courses train soldiers to be decisive and adaptive in a changing mission- a mistake on a single slide is likely to be outweighed by the benefits of new, relevant case studies and also mitigated by the fact that Army leaders collaborate to make plans. On the converse, a small error in parachute rigging can directly lead to deaths. Speed of change represents how quickly knowledge must be updated to be useful. For example, while new parachutes are designed infrequently, new small unmanned aerial vehicles (UAVs) and tactics emerge multiple times per month in modern conflicts. As a result, even if changes have non-trivial risk, the risk of failing to update content is likely to be higher. These differences mean that ARC and similar platforms must support content developers with very different needs. AI to revise safety-critical training, such as parachute rigging, should prioritize verification and high reliability, such as helping human experts triple-check content despite saving time. On the converse, evolving topics such as UAVs should have AI that can share the most-recent lessons learned and feeds of recent events. Other domains, such as mission planning or medical techniques, require reviewing changes in references and then updating related training, which is where the ARC project started.

Research with training centers identified multiple ways AI could be better integrated with training updates. However, advances in Army and DoD infrastructure for innovation are necessary to unlock additional avenues for AI for developing and revising training. Policies and sandboxes are needed to pilot AI inside the kind of tools that soldiers already use. In meetings across multiple centers, it was generally agreed that SharePoint and Office tools were a de-facto standard for document editing and management. Moreover, in a connected project (the AI Army Writing Enhancement Tool, or AWE), we demonstrated that it is technologically straightforward to integrate AI tools using the SharePoint Framework (SPFx) and similar systems. If ARC integrated with SharePoint, we could analyze folders of documents, apply AI analyses or changes to live-editable lesson plans, and similar capabilities to expand its impact. Integrating with common tools would also enable the DoD to grow an AI workbench for training development. However, current policies block "pilot scale" integration with widely-used Army tools behind months (or even years) of approvals. In our research, we explored setting up our own SharePoint sandbox, which was successful for testing AWE, but is not very useful for pilot testing because Army credentials could not be used to log in. As such, ARC was tested on standalone machines. However, with different DoD policies tests could have been conducted with ARC connections to SharePoint. As the Army and broader training moves toward AI, sandboxes are needed that offer a low-friction path to build and test for operational use cases.

Finally, research on ARC is continuing with an additional major revision based on the most recent Beta test in May 2025. This revision requires users to select a role that aligns to the single-course developer vs. multi-course developer division, so that we can provide them a Home screen that emphasizes the tools that are most relevant to their needs. We are also enabling functionality such as the data can be stored for use after an analysis, with the main use-case being that a change list can be stored for different document versions, and then that change list can be used to categorize issues (e.g., filter to only see changes related to a change in a specific concept, such as Large Scale Combat Operations). We are also developing training materials and guides for ARC, which is part of our broader goal of training instructors and course developers on ways to use AI (Core et al., 2025; Luckin et al., 2022). This dovetails with ongoing research at ICT for AI generation of high-quality active and interactive tutoring, which may be relevant to developing or revising assessments. Guides will also suggest complementary tools to generate and verify training content, ranging from AI prompt guides to specialized frameworks (both in-house and broader DoD initiatives).

ACKNOWLEDGEMENTS

The project or effort depicted was or is sponsored by the U.S. Government under contract number W911NF-14-D-0005, as part of the USC ICT UARC and the AI Research Center of Excellence in Education (AIRCOEE). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. We would also like to greatly thank LTC Elvin Fortuna, lead SME for ARC: his extensive knowledge and insight into the curriculum design and update process was critical to this work. We would also like to thank all Army centers who provided extensive feedback and corpus of materials used to test ARC, including the Fires Center of Excellence (FCoE), Medical Center of Excellence (MEDCOE), Quartermaster Center and School, Transportation School, Ordnance Corps School, DTRA and Army University. We appreciate your strong engagement and hope this research informs the future of tools to support your training mission.

REFERENCES

Adhikari, N. S., & Agarwal, S. (2024). A Comparative Study of PDF Parsing Tools Across Diverse Document Categories. *arXiv preprint arXiv:2410.09871.*

AFIT Faculty Learning Community (2025). *AFIT Generative AI Teaching Guidebook.* Wright-Patterson AFB, AFIT. Retrieved June 25, 2025 from: https://scholar.afit.edu/docs/140.

Anthropic. (2024). *Claude 3.5*. www.anthropic.com/

Ambite, J. L., Fierro, L., Gordon, J., Burns, G. A., Geigl, F., Lerman, K., & Horn, J. D. V. (2019). BD2K Training Coordinating Center's ERuDIte: The Educational Resource Discovery Index for Data Science. *IEEE Transactions on Emerging Topics in Computing*. https://doi.org/10.1109/TETC.2019.2903466

Asthana, S., Arif, T., & Thompson, K. C. (2023). Field experiences and reflections on using LLMs to generate comprehensive lecture metadata. In *NeurIPS'23 workshop on generative AI for education (GAIED)*.

Aumüller, M., Bernhardsson, E., & Faithfull, A. (2020). ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, *87*, 101374. https://doi.org/10.1016/j.is.2019.02.006

BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., & Reddy, S. (2024). Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961.*

Camacho-Collados, J., & Pilehvar, M. T. (2018). From Word To Sense Embeddings: A Survey on Vector Representations of Meaning. *Journal of Artificial Intelligence Research*, *63*, 743–788. https://doi.org/10.1613/jair.1.11259

CALL - Center for Army Lessons Learned (2025). *Enhancing Military Operational Effectiveness through the Integration of CAMO and NIPR GPT. 25-958.* Retrieved Jun 26. 2025 from: https://api.army.mil/e2/c/downloads/2025/03/07/840ed7cf/25-958-enhancing-military-operational-effectiveness-through-the-integration-of-camo-and-nipr-gpt.pdf

Chandrasekaran, D., & Mago, V. (2021). Evolution of Semantic Similarity—A Survey. *ACM Computing Surveys*, *54*(2), 41:1-41:37. https://doi.org/10.1145/3440755

Core, M., Nye, B., Carr, K., Li, S., Shiel, A., Auerbach, D., ... & Swartout, W. (2025, May). Usability and Preferences for a Personalized Adaptive Learning System for AI Upskilling. In *Proceedings of Florida AI Research Society (FLAIRS) 2025 Conference, 38,* 1-7. AAAI Press.

Dai, A. M., Olah, C., & Le, Q. V. (2015). *Document Embedding with Paragraph Vectors* (arXiv:1507.07998). arXiv. https://doi.org/10.48550/arXiv.1507.07998

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the NAACL: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Fakhraei, S., Mathew, J., & Ambite, J. L. (2019). NSEEN: Neural Semantic Embedding for Entity Normalization. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases.*

Fortuna, E., (in progress). Prompt set for learning product development. *CamoGPT Workspace.*

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., ... & Sayed, W. E. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088.*

Karataş, F., Eriçok, B., & Tanrikulu, L. (2025). Reshaping curriculum adaptation in the age of artificial intelligence: Mapping teachers' AI-driven curriculum adaptation patterns. *British Educational Research Journal, 51*(1), 154-180.

Kelly, P., & Smith, H. (2024, May). How to think about integrating generative AI in professional military education. *Military Review, 1*, 1-8.

Kirkpatrick, D., & Kirkpatrick, J. (2006). *Evaluating training programs: The four levels*. Berrett-Koehler Publishers.

Li, J., Luong, T., & Jurafsky, D. (2015). A Hierarchical Neural Autoencoder for Paragraphs and Documents. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1106–1115. https://doi.org/10.3115/v1/P15-1107

Liu, Q., Kusner, M. J., & Blunsom, P. (2020, March 16). *A Survey on Contextual Embeddings*. ArXiv.Org. https://arxiv.org/abs/2003.07278v2

Luckin, R., Cukurova, M., Kent, C., & du Boulay, B. (2022). Empowering educators to be AI-ready. *Computers and Education: Artificial Intelligence, 3*, 100076.

Nye, B. D., Jain, A., Ramirez, D. R., Core, M. G., & Swartout, W. (2022). Designing a rapid adaptive content registry (RACR) for adaptive learning. In *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2022*. NTSA.

OpenAI (2024). GPT-4o mini: advancing cost-efficient intelligence. Retrieved June 27, 2025 from: *https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/*

Rainey, J.E. (2024, August). Continuous Transformation. *Military Review 1*, 1-5.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Ruiz, D. C., & Sell, J. (2024). Fine-Tuning and Evaluating Open-Source Large Language Models for the Army Domain. *arXiv preprint arXiv:2410.20297.*

Tonmoy, S. M. T. I., Zaman, S. M., Jain, V., Rani, A., Rawte, V., Chadha, A., & Das, A. (2024). A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313, 6.*

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models* (arXiv:2302.13971). *arXiv.* https://doi.org/10.48550/arXiv.2302.13971

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 425-478.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*. papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91 fbd053c1c4a845aa-Abstract.html

Vought, R. (2025). *Accelerating Federal Use of AI through Innovation, Governance, and Public Trust. Memorandum M-25-21.* Retrieved June 15, 2025 from: https://www.whitehouse.gov/wp-content/uploads/2025/02/ M-25-21-Accelerating-Federal-Use-of-AI-through-Innovation-Governance-and-Public-Trust.pdf

Wang, S., Zhou, W., & Jiang, C. (2020). A survey of word embeddings based on deep learning. *Computing*, *102*(3), 717–740. https://doi.org/10.1007/s00607-019-00768-7

Wu, X., Saraf, P. P., Lee, G., Latif, E., Liu, N., & Zhai, X. (2025). Unveiling scoring processes: Dissecting the differences between LLMs and human graders in automatic scoring. *Technology, Knowledge and Learning*, 1-16.

Xian, J., Teofili, T., Pradeep, R., & Lin, J. (2024). Vector search with OpenAI embeddings: Lucene is all you need. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (pp. 1090-1093).