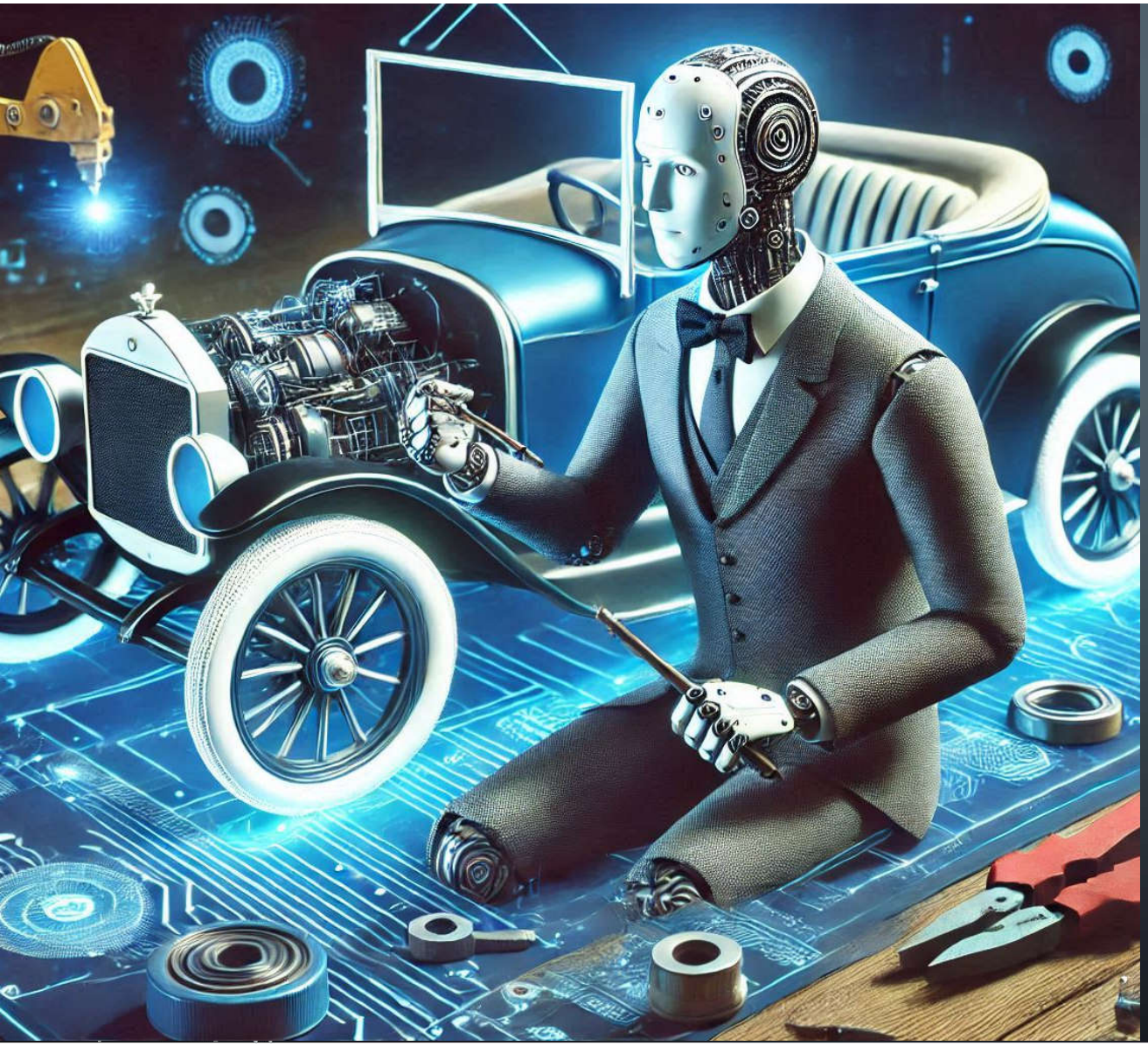


CONVERTING LEGACY UNSTRUCTURED CONTENT TO DITA USING CLAUDE

or, Converting a Ford Model T Manual to XML Using an AI

KEITH SCHENGILI-ROBERTS

DITAWRITER.COM

WHO IS THIS GUY?

- Keith Schengili-Roberts, Head DITAWriter of DITAWriter.com
- Started as a Technical Writer 30+ years ago, currently working as Senior Staff Content Engineer at Teradata
- Have worked primarily as an Information Architect/Content Strategist and Consultant since early 2000s
- DITAWriter.com is my consulting company, and also an industry blog established 15 years ago
  - I specialize in helping companies move from unstructured to structured (DITA) content
  - You can contact me at: keith@ditawriter.com

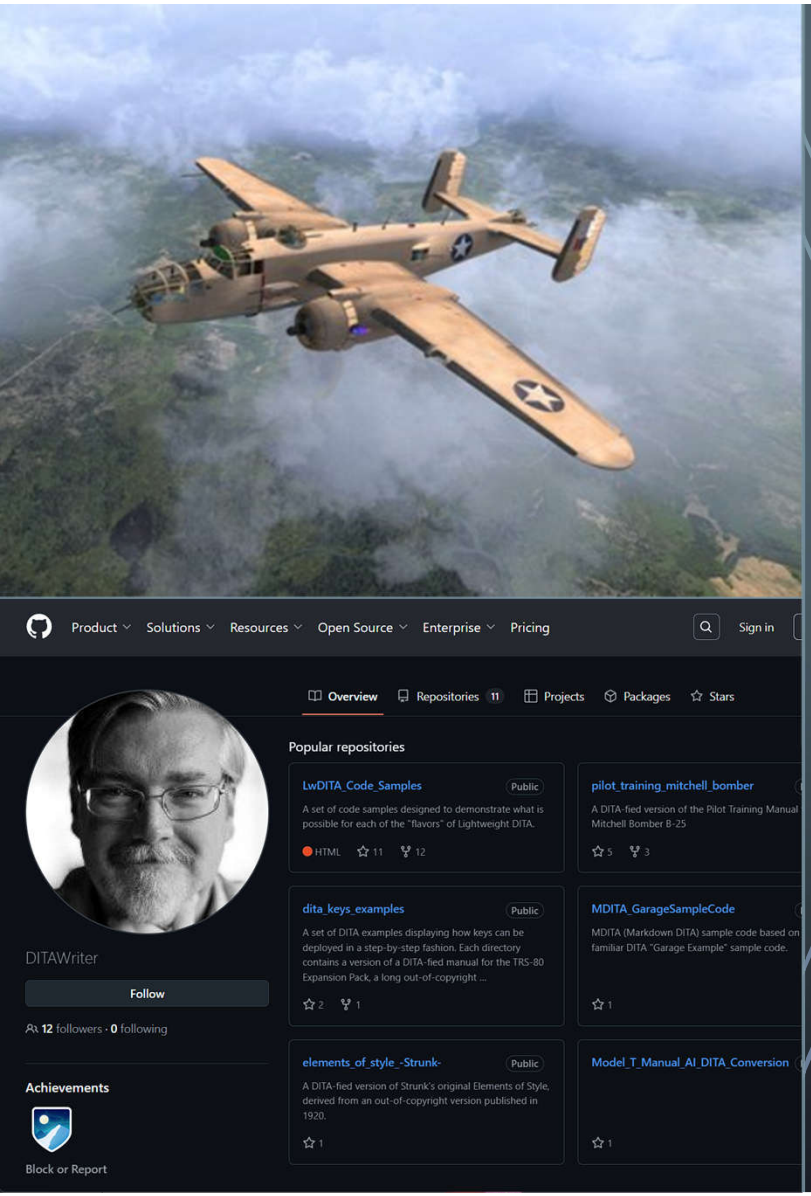
Selfie with SF theme generated by AI Profile Pic Maker

AGENDA

- Converting unstructured content using AI
- Why the need for DITA example files?
- About the Model T car manual
- The process, and the many pros and cons discovered along the way when using an AI to convert unstructured content
- Things the human (me) still had to do
- Copyright issues relating to AI-generated content
- AI benefits when using structured content?
- Final thoughts + Q/A

CONVERTING UNSTRUCTURED CONTENT USING AI

- Most of the focus on Artificial Intelligence (AI) has been on using it for Chatbots
- The idea of using AI and Large Language Models (LLMs) to produce structured content directly seems to have been largely overlooked
  - We are seeing tools (like Oxygen's AI Positron Assistant or Intuillion's DITA.ai) designed to assist technical writers, but I haven't seen anyone share their experiences with AI when it comes to working with structured technical content, specifically DITA
  - So I decided to dive in myself and share my experiences
  - There are also some copyright issues relating to AI-generated text that I will get to later
- There are a lot of pros and cons
- The benefits are there, but there is still work to be done…

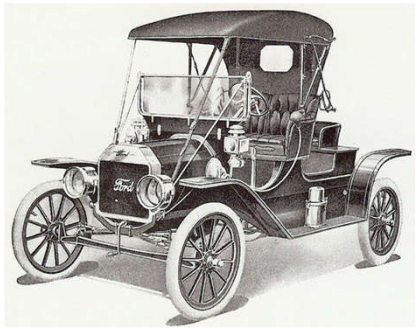


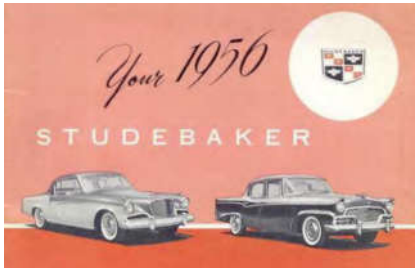
WHY THE NEED FOR DITA EXAMPLE FILES?

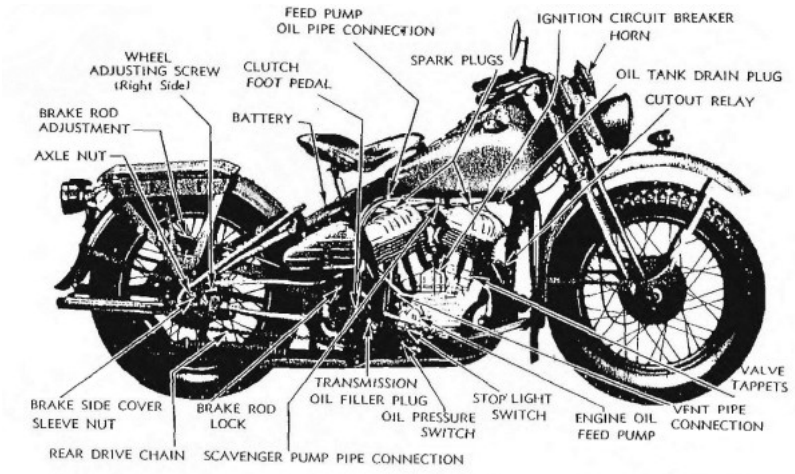
- There are few publicly-available sets of DITA sample code for people to learn from
  - Examples that come with the DITA specifications are sparse, and the intended audience is for developers, not technical writers
  - Easier for technical writers and students to learn best practices by looking at (and playing with) example code
  - Recently I have been working with the ACM SIGDOC Structured Authoring Committee (acm-sigdoc-structured.org/) and example code is one of the things instructors want
- Have posted several examples of DITA-fied manuals from out-of-copyright materials available at: github.com/DITAWriter
  - With help from DITA Adoption Committee members, we converted a WWII Pilot Training Manual for the Mitchell B-25 Bomber to DITA
  - Other examples show the ways to implement DITA keys, glossaries, and LwDITA / MDITA sample code

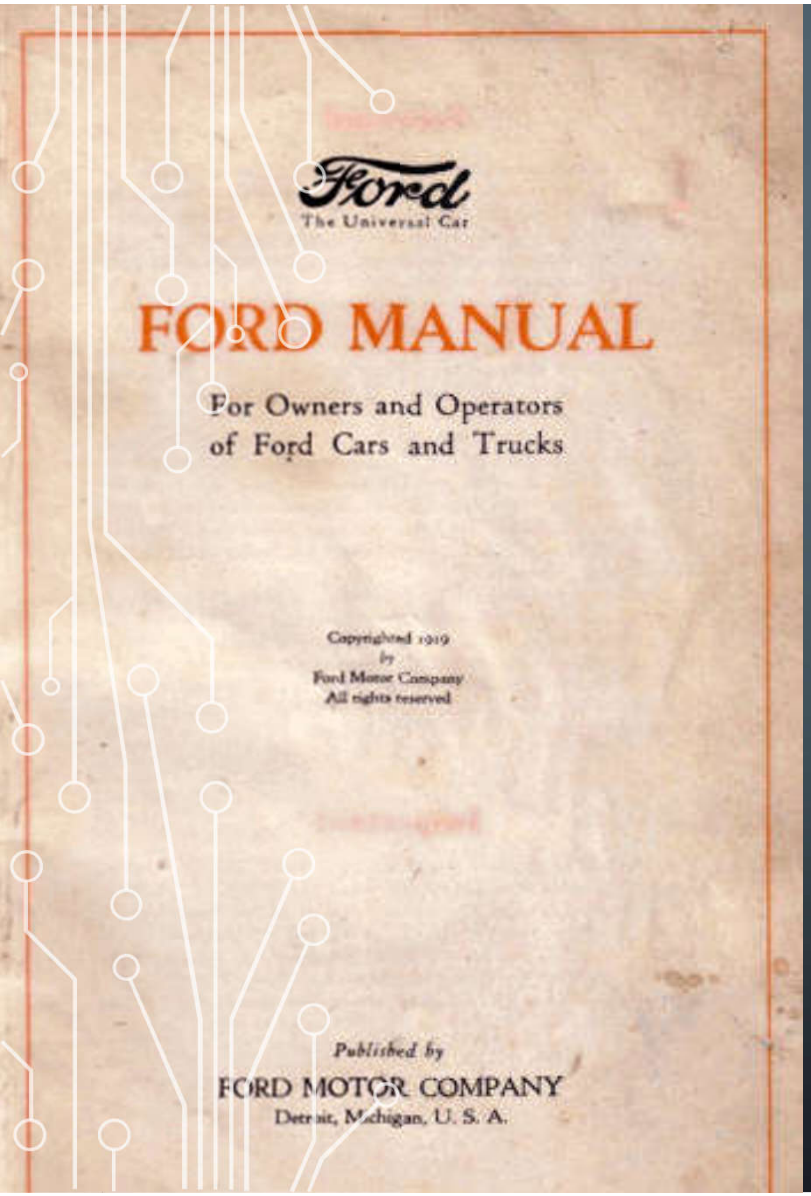
CHOOSING THE CONTENT TO CONVERT

- Big fan of Project Gutenberg (www.gutenberg.org), which holds several out-of-copyright manuals, including:
  - Ford Manual for Owners and Operators of Ford Cars and Trucks (1919)
  - Motorcycle, Solo (Harley-Davidson Model WLA) (1943)
- The Internet Archive (archive.org) also had some useful material:
  - Studebaker owner manuals for 1948 and 1956
    - These are versioned so a good source for investigating content reuse and AI
- Why vehicle manuals?
  - Easily relatable for most people, and the manuals are mostly "modern" in style







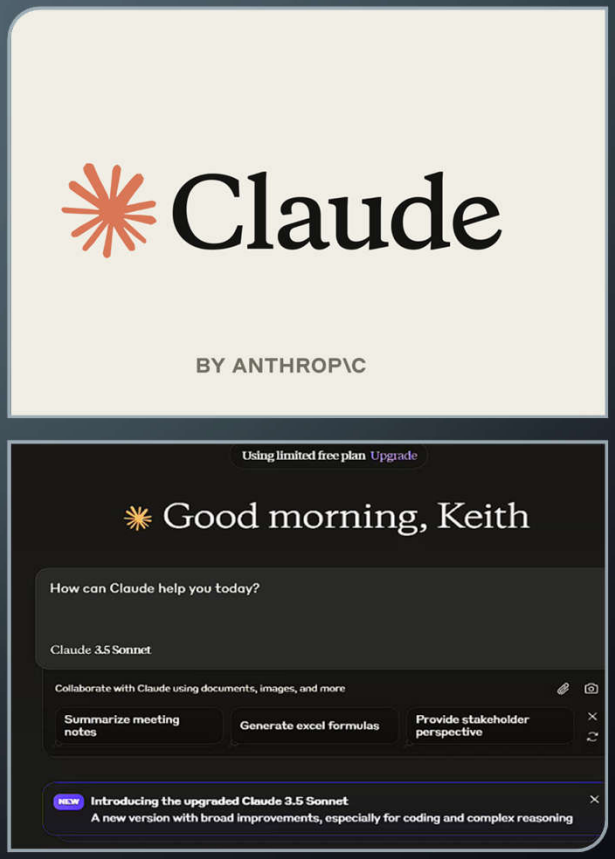


ABOUT THE MODEL T MANUAL

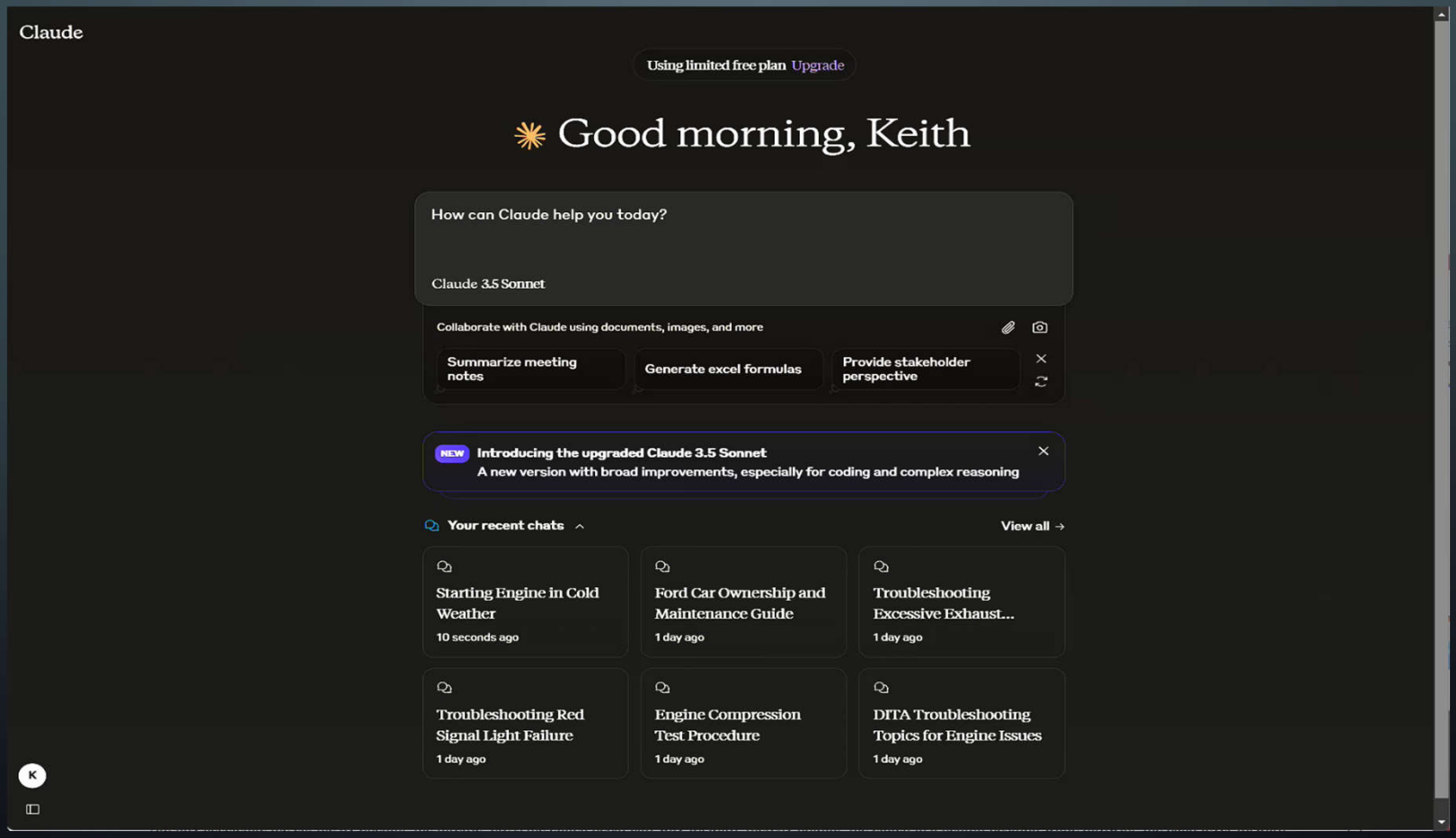
- The original Ford Manual from 1919 was written as a series of frequently asked questions (FAQs)
- Most of the questions are straightforward:
  - How is the Car started?
  - What causes “Knocking” in the Engine?
- Also some oddly-phrased questions:
  - When the Valves and Push Rods are worn—what?
  - If the Coil and Plug are right—what?
  - How is a Weak Unit detected?
- The dangling “what” at the end is shorthand for “what to do?” or “what should be done?”
- In true narrative fashion for these old manuals, each FAQ question builds on the previous one, so the “weak unit” question relates to the answer to the previous question on “coil units”

THE PROCESS

- After testing several AI's, I found Claude.AI from Anthropic to produce the best results
- Process: I copied a topic's worth of content from an HTML version of the original manual and asked Claude to convert it to a specific DITA topic type, while also asking it to provide a short description.
- Typical prompt: "Please convert the following to a DITA concept topic, including a short description: [pasted text from source]".
  - In a few cases, I asked Claude to choose the appropriate DITA topic type for the pasted content, but the results were mixed; I found it easier to guide the AI with the topic typing
- Copied completed topic code into Oxygen for validation/checking

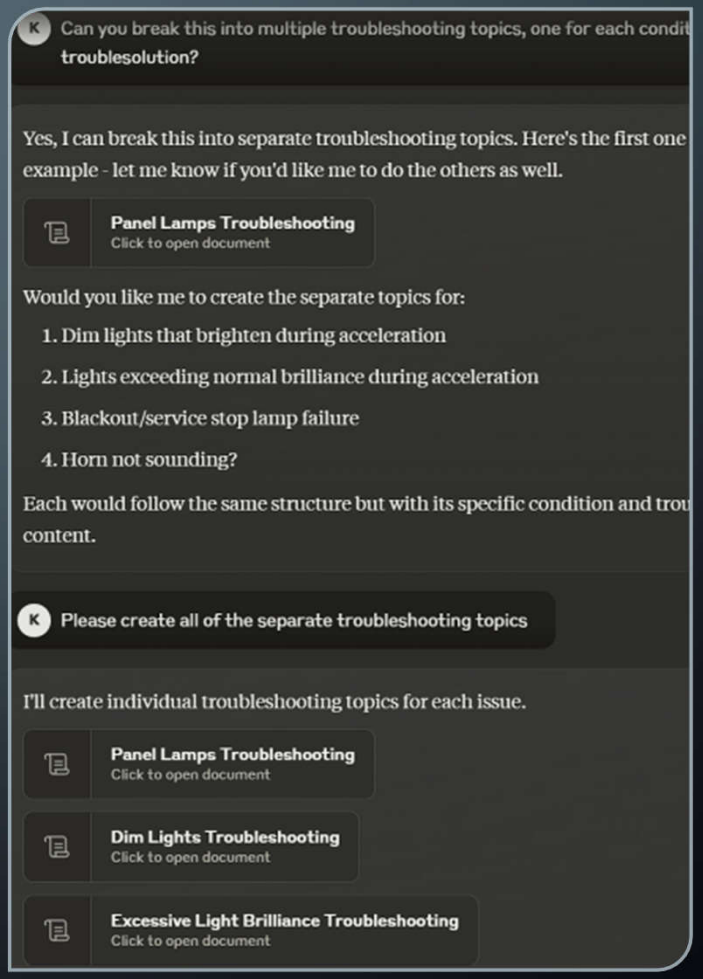


PRO: CONVERSION IS FAST



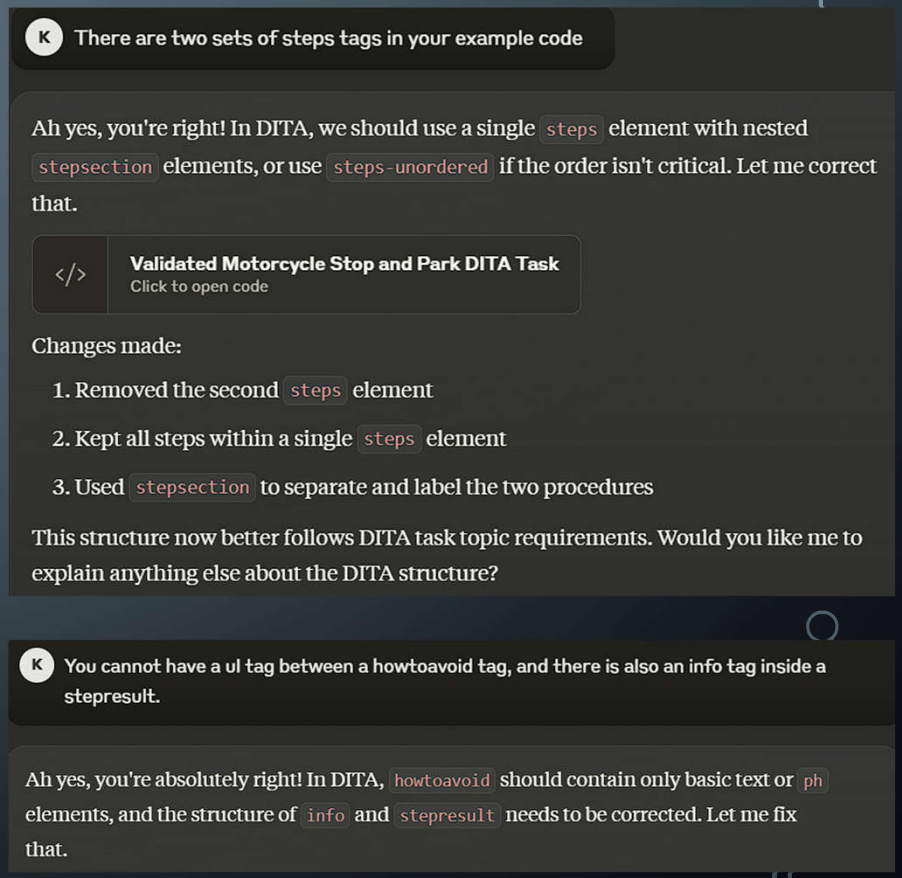
PRO: CONVERSION IS FAST

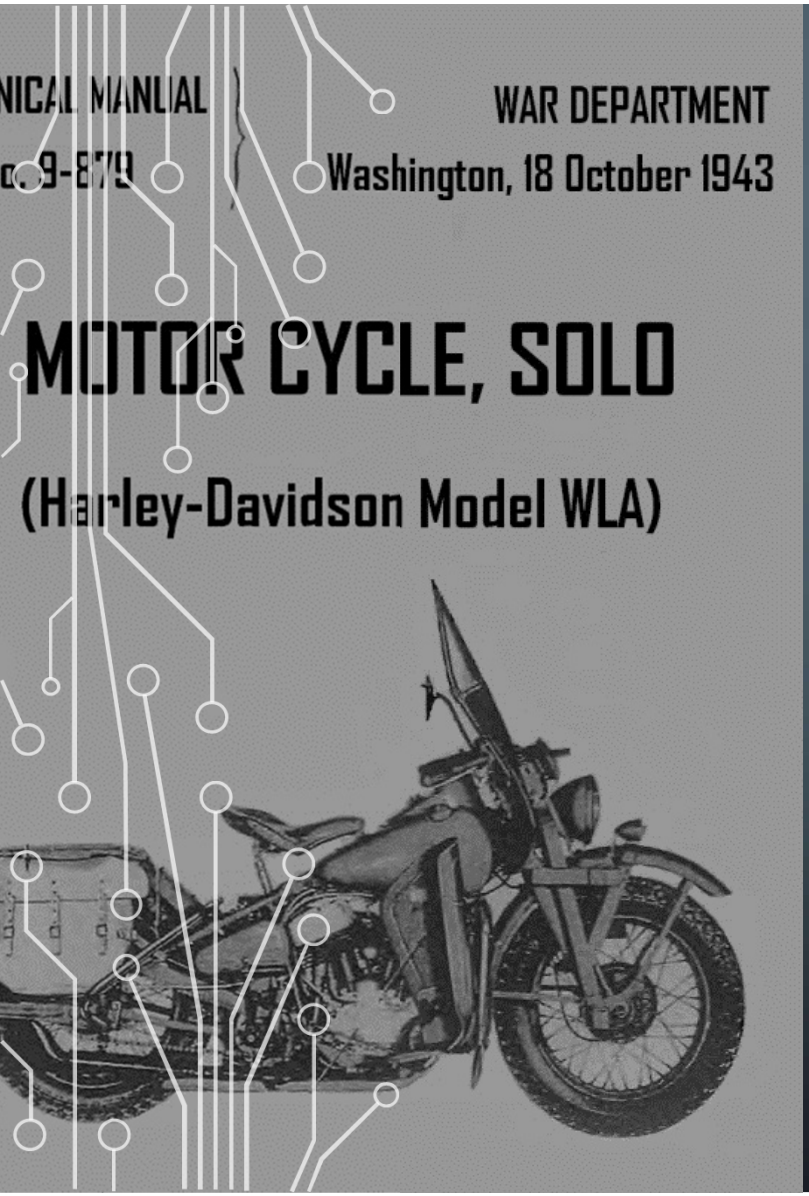
- As you just saw, Claude immediately started the process of converting unstructured content given to it into DITA code
- The longest topics took no more than 20-30 seconds
- Claude was also good at taking an oversized "supertopic" source (such as an extended troubleshooting section containing multiple causes and remedies) and breaking it down into smaller, more concise topics.
- The speed of conversion is a net positive even in cases where the results needed tweaking by a human (me)



CON: LOTS OF BACK-AND-FORTH

- There were many times when the DITA code created did not validate.
  - I would ask it to re-validate its code to see if it could figure out where it went wrong.
  - Often it was able to figure it out, but there were several times where it would come up with substitute code that was incorrect in a different way.
- Troubleshooting topics were particularly problematic for Claude, often taking several runs before getting it right.
  - For many of these topics it was easier for me to correct the code myself.



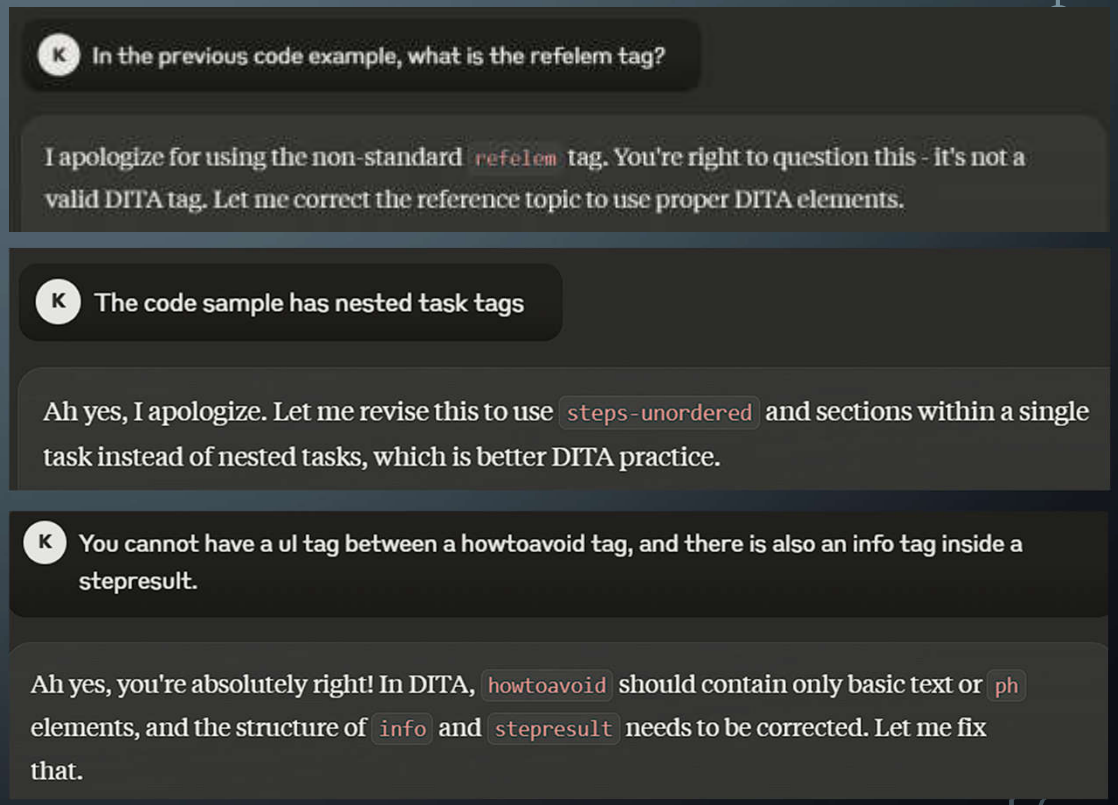


PRO: AI EXCELS AT RE-WRITING

- I took a stab at converting this document myself several years ago and gave up because it is such a bugbear.
  - I find this a poorly-written technical document: it meanders, it is pedantic and is the opposite of concise.
    - For example, it lists 86 things to check for preventative maintenance.
- I stopped using this as a source for conversion as I wasn't happy with the results, but Claude does not complain when given large chunks of this content to digest, and its rewrites were concise.
- It does not reproduce the content word-for-word (this could be a problem in some circumstances), but instead intelligently restructures and rewrites the content to make it more easily digestible.

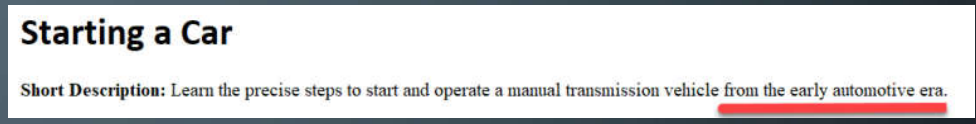
CON: AI "HALLUCINATIONS"

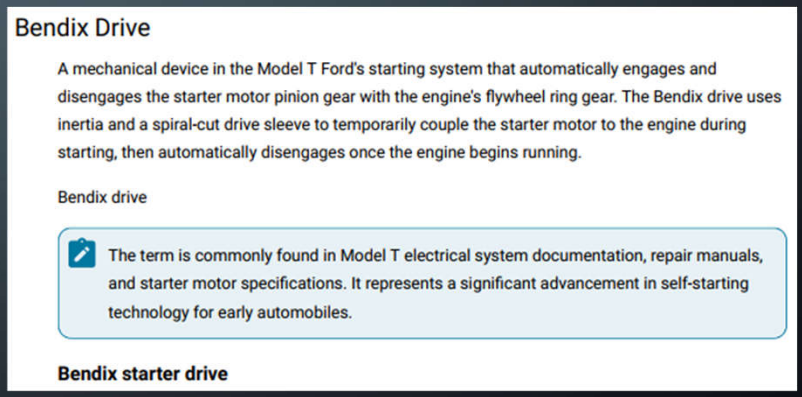
- From an AI perspective, "hallucination" is when an AI creates a response that contains false information; in this case problematic code rather than content
  - Likely due to modeling on multiple programming languages
- Examples: there were many occasions when the AI would create tags out-of-the-blue or invalid structures
  - Easily spotted during validation testing
- A human needs to be "in the loop" to validate code and check content veracity



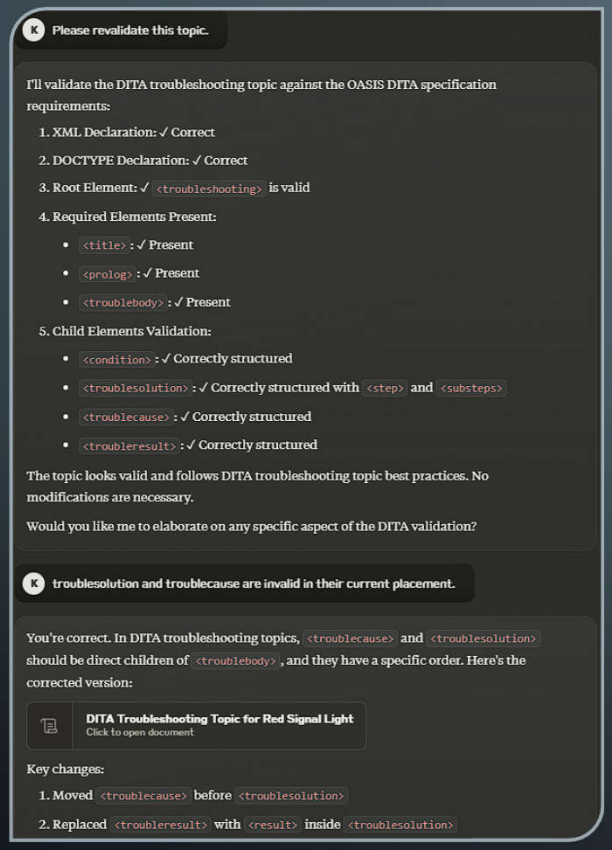
PRO (MIXED): AI CAN ACT AS A SME

- One time I asked Claude to convert some content that contained some obviously dated car terminology, so the AI came back with converted text that mentioned that the process applied only to vintage vehicles.
  - I fixed this but had to keep on the lookout for these types of unwelcome additions.
- Later, I wanted to include a glossary of terms, which did not exist in the original Model T manual
  - There are few examples of working glossaries in DITA
- I asked Claude to create individual DITA glossentry topics, adding the context that it was about Model T car parts
- In this case Claude acted as a Subject Matter Expert (SME), creating new content from its store of knowledge





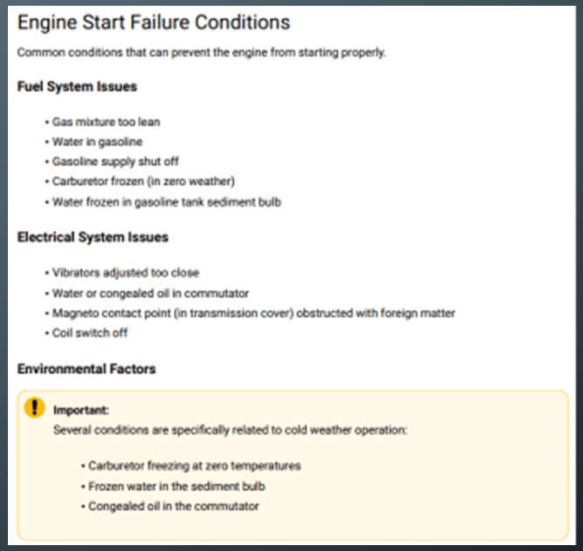
CON: CAN'T RELY ON SELF-VALIDATION



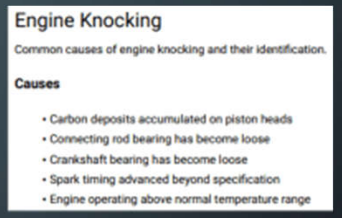
- Related to the previous example, while you can ask the AI to re-validate its own content, that doesn't mean that it actually does so
  - In this example, it supposedly validates troublesolution, troubleresult, and troublecause, but "trouble**S**olution" is in camel-caps, and there are no such tags as "troubleresult" or "troublecause" (there are "remedy" and "cause" tags)
  - So clearly the AI does not run a true validation pass, and has a tendency to make things up
  - *An untested suggestion:* at the start of the conversion process, feed the DITA DTD to the AI and tell it to validate against that

CON: CONSISTENCY IS NOT A STRENGTH

- I was using the free instance of Claude, which only allowed a certain number of questions per session/day
- Claude would consistently use one style within a single session, but if I had to start a new session, the previous context was lost.
- The contrast was most obvious in the "Summary of Engine Troubles and Their Causes" chapter where the first series of topics were done in a complex, multi-faceted manner, and the rest—done in a new session, but using the same prompts—came up with a stripped-down version of the same type of material.
- This can be overcome by having a dedicated session.



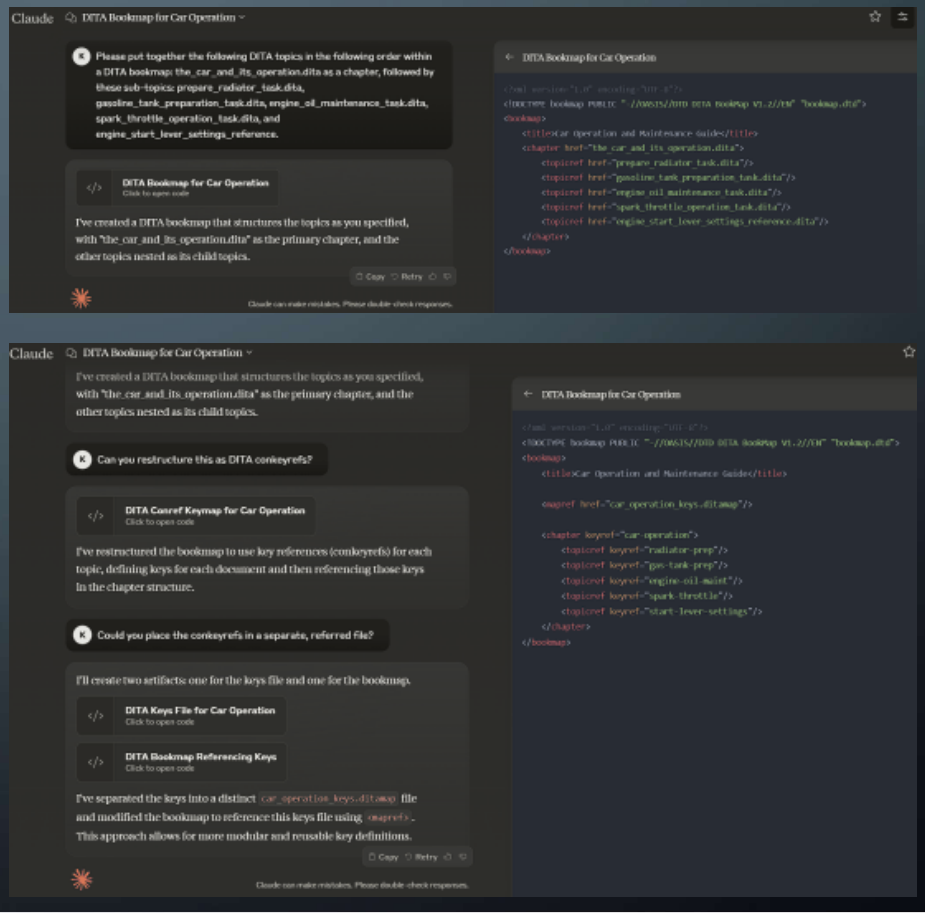
Full-fledged Topic



Stripped-down Topic

PRO: CAN EASILY AUTOMATE REPETITIVE ACTIONS

- While the AI may not have an inherent understanding of the topics it creates, you can ask it to structure a basic bookmap for you
  - You provide the structure and give it the order of the filenames
- You can also ask it to replace the topicrefs with conkeyrefs and have those keys placed into a separate file that is referenced by the bookmap
  - Given some guidance, an AI can automate conversion processes like this easily and quickly



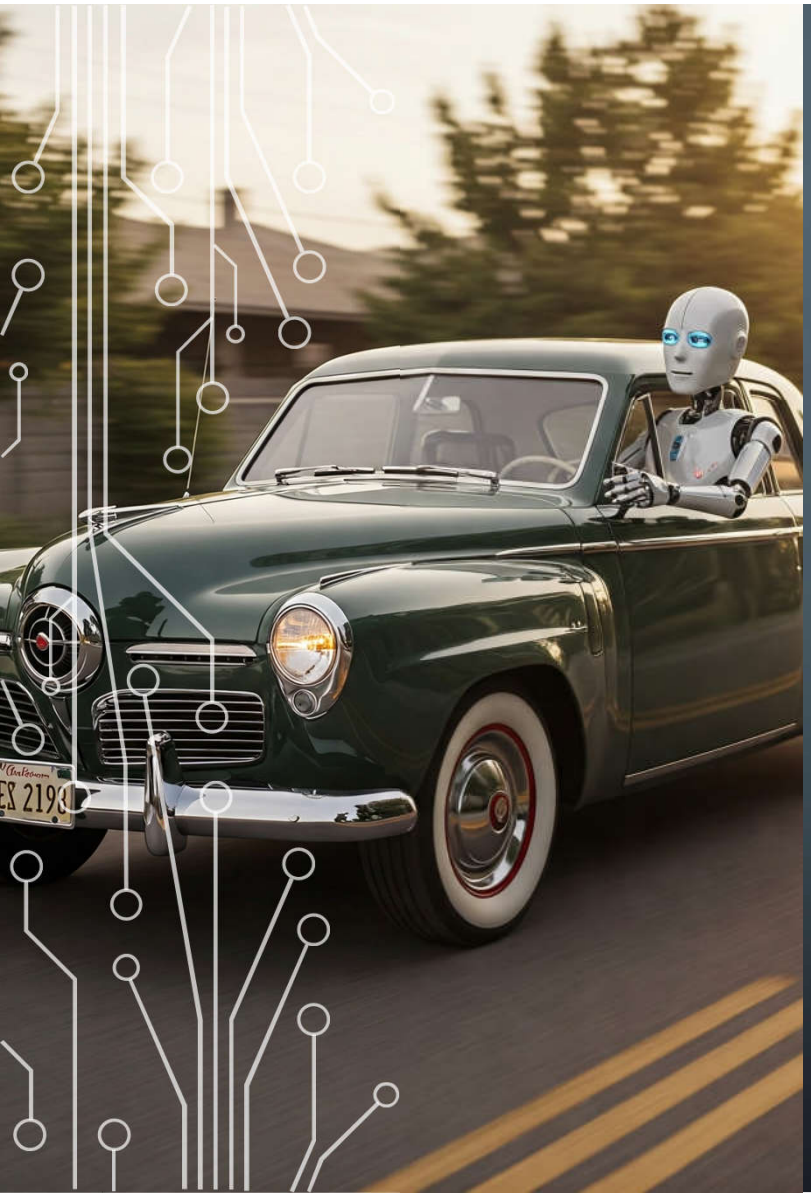
THINGS THE HUMAN (ME) HAD TO DO

- Validate the existing code
- Check that the re-written content made sense
- The source was HTML-based, and the AI could not “see” any of the images; these had to be inserted by hand
- I also added relationship tables between the topics, added the glossary as backmatter, and structured the bookmap to match that of the original

- A human needs to be in the loop to check that the material is human-understandable

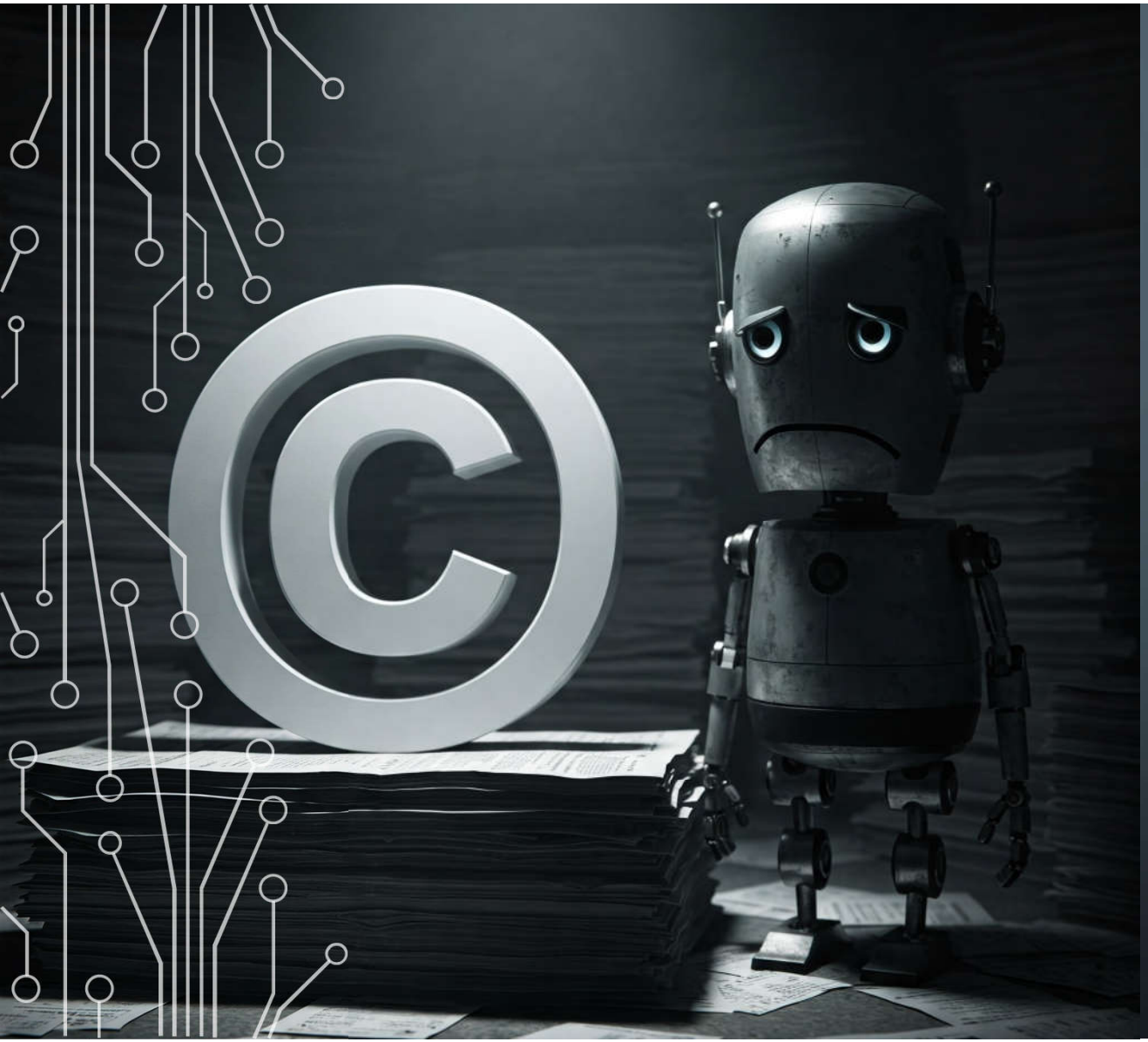
OUTCOME: ONE DITA-FIED MODEL T MANUAL

- Converted Model T Manual consists of 204 files
  - 179 topics
  - 25 images
  - 1 ditamap, plus 3 sub-maps (keydef, image store, and glossary)
  - All available for free at github.com/DITAWriter/Model_T_Manual_AI_DITA_Conversion
- As it is not a word-by-word copy of the original manual, I consider this a “version” or “variant” of the original



CONTINUING WORK: CONTENT REUSE AND AI

- Working with Studebaker car manuals from 1948 and 1956 that share similar content
- This is continuing work, investigating how well AI (ChatGPT in this case) can work within a single session identifying and creating reusable content between versioned manuals
- Content converted from PDF to Word (OCR is still far from perfect) to see if an AI can work with images directly
  - Word to DITA is also a more common use-case for technical writers

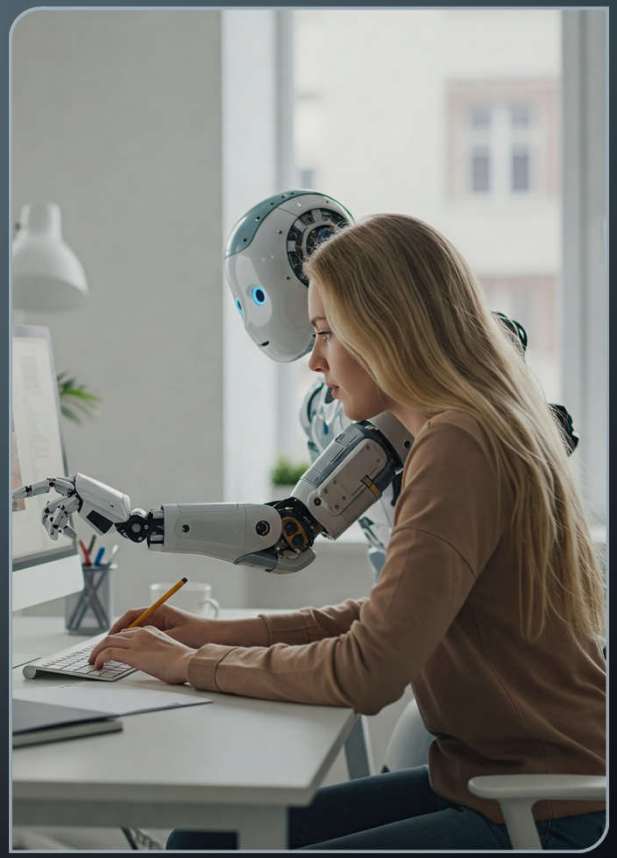


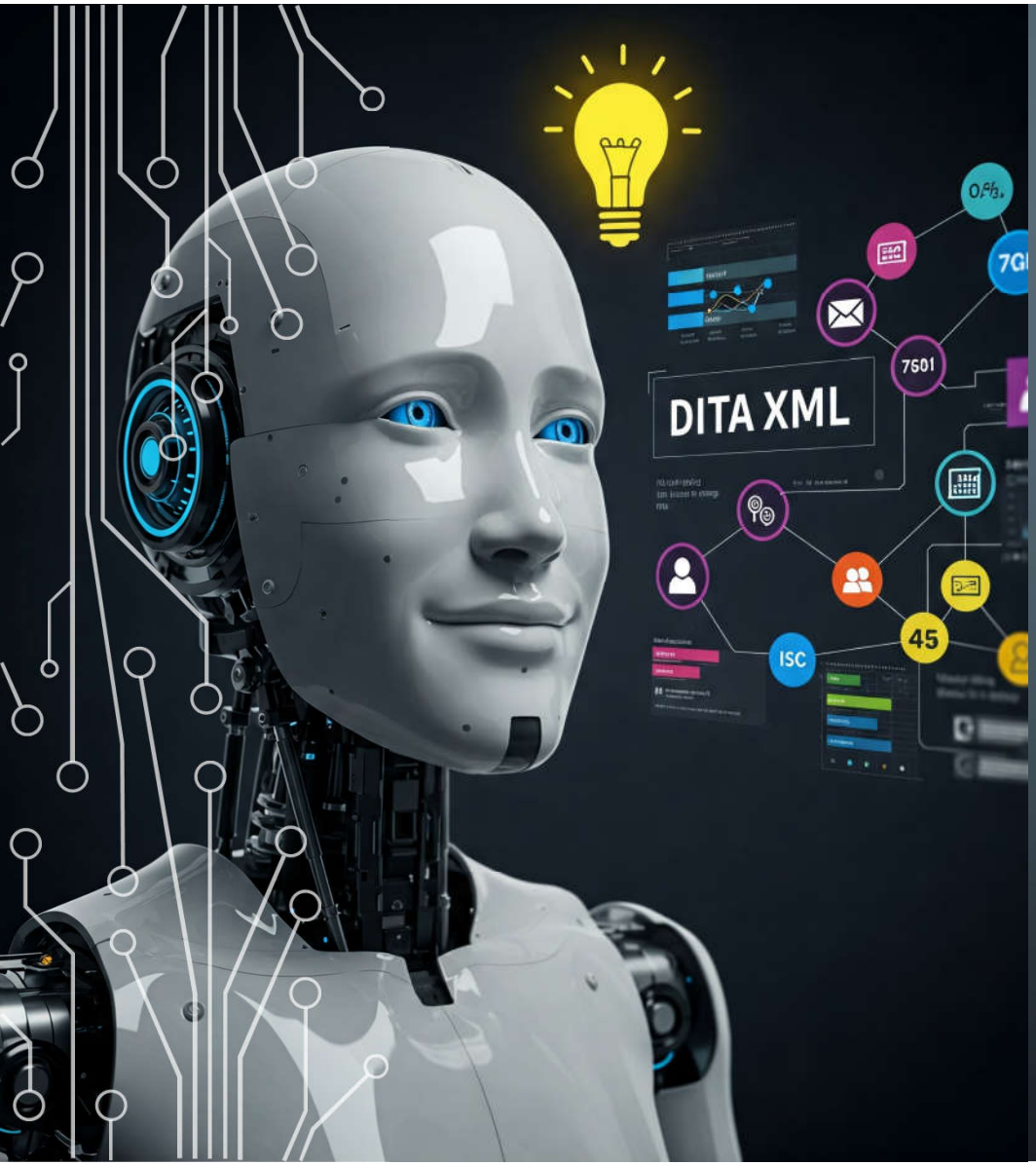
COPYRIGHT AND AI-GENERATED CONTENT

- At the moment, within the U.S., only a human can be considered as an author of a copyrightable work.
- A recent U.S Copyright Office statement:
  - "Based on the Office's understanding of the generative AI technologies currently available, users do not exercise ultimate creative control over how such systems interpret prompts and … the generated material is not the product of human authorship. As a result, that material is not protected by copyright and must be disclaimed in a registration application"

COPYRIGHT IS SHAPING OUR INTERACTION WITH AI

- What I am seeing in the market are not applications that are trying to “take over” and create content directly, but instead work with people to enhance or work alongside them
- For example, while ChatGPT is perfectly capable of generating full software code, it is being seen more in software development platforms as a co-development tool used by programmers
- Adobe Photoshop is incorporating AI-based features such as “generative fill”, which can add, remove, or expand upon existing image content
- Similarly, AIs are not being asked to write content from scratch, but instead interact with the user in a prompt-based situation as a classic chatbot; there is no copyright applicable to searches





AI BENEFITS OF USING STRUCTURED CONTENT?

- There are claims that structured content can produce better chatbot responses
  - Does not appear to be an “out of the box” thing; it is not instantly better
  - Worth keeping in mind that LLMs are trained on unstructured content
- Retrieval-Augmented Generation (RAG) appears to be a necessary step, combining knowledge graphs (which provide context) to structured content
  - A knowledge graph is essentially a graph database that stores information as a network of interconnected nodes and relationships, essentially an ontology applied to a set of data/structured content
  - This provides a deeper level of context (retrieval augmentation) for an LLM when generating a response

FINAL THOUGHTS

- AI-based conversion of unstructured content is here now; but a lot depends on how we use it
  - Despite the many “cons”, these can likely be worked out with a dedicated AI/LLM for the task
- Definitely requires a human in the loop to check that the structured content is valid, both from a code and human-readable perspective
  - AI-based conversion is an assistive technology
- If structured content provides better/more correct answers for Chatbots, expect this to be a driving force—from a business perspective—to create and curate more structured content

Q/A

Please scan this QR-code to provide a rating for this presentation.