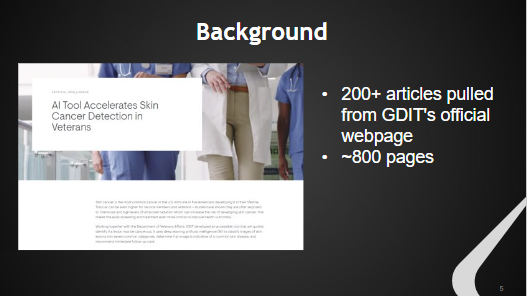


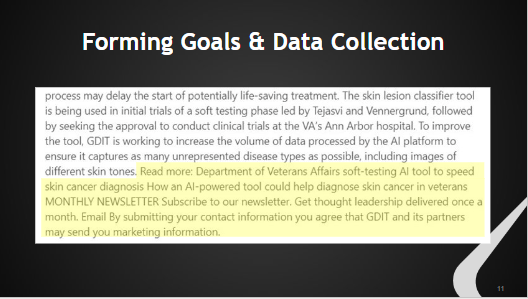
* That’s why we created our brainchild, DISCOVER.
* DISCOVER stands for Data Intelligence and Search Capability Optimizer for Visualizing and Exploring Resources.
* Thank you for joining us today as we embark on this exciting journey with DISCOVER.



* Background, we started with over 200 PDF articles pulled from GDIT's official webpage.
* For context, Praxis is a subsidiary of GDIT.
* Total of nearly 800 pages or 166k words
* Given this data, we had to come up with a creative solution to obtain useful insight and make it accessible to anyone



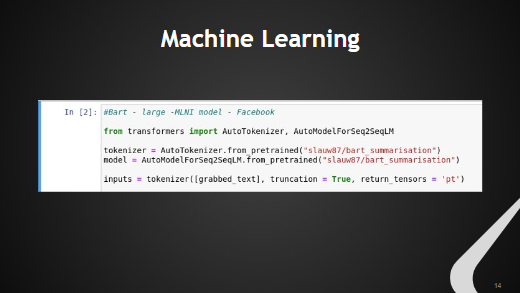
* First we took time to familiarize ourselves with the articles and turn the raw files into a data set.
* In order to convert the files, we used the PyPDF2 python library to read text from the pdfs into csv form.
* We had to form goals of how we could gain insights from this data and process it efficiently.



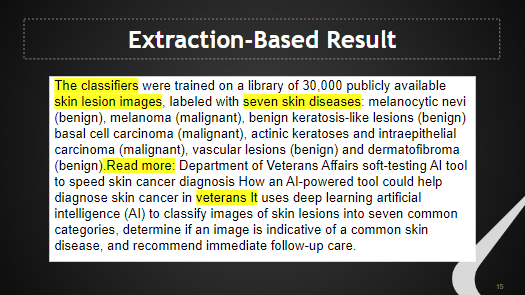
* In this phase, we had to clean up the formatting of the text by removing irrelevant parts.
* Many of the articles had footers like this at the end with related reading and information about a monthly newsletter.
* After we manually found the information that we knew wouldn't be relevant to our analysis, we automated and applied the cleaning to all of our articles in Jupyter Notebooks



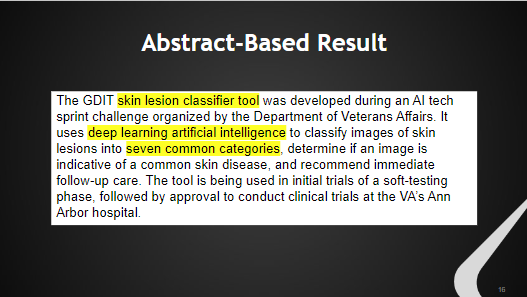
* Then, we used the spacy library along with frequency analysis methods to extract the most important words, phrases, and sentences in articles.
* Finally, we used the pandas library and data frame structure to create a csv with organized columns for all the data and results we had collected so far.



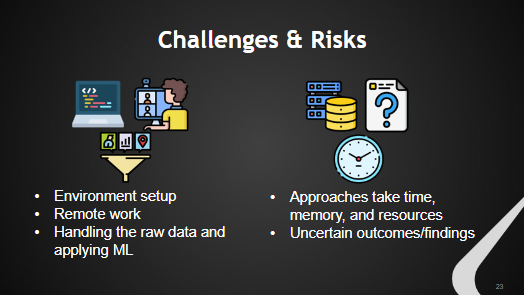
* Our goal was to use machine learning to create abstraction-based summaries whereas before we'd used extraction-based methods.
* We explored pre-trained deep neural networks and found success with a BART large language model, which works by corrupting text with an arbitrary noising function. BART was created by Facebook and announced in 2020. and has become a benchmark natural language interpretation related models



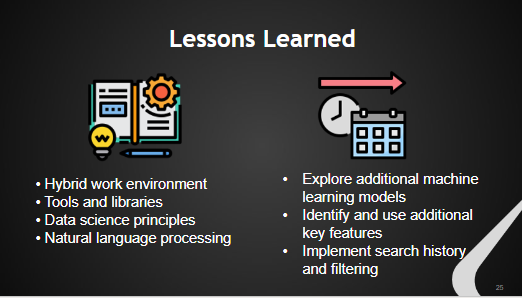
* To illustrate the difference in our summaries, here is the extraction-based summary for the article composed of the sentences that our frequency analysis decided were most important.
* The first half of the summary is very technical and doesn't give much background, but we can see that it's about classifying skin lesion images and diseases.
* The "read more" is confusing – is the whole second half of this summary even about the original article or a related reading?



* Now, this is our abstraction-based summary where ML created sentences rather than just extracting them.
* It didn't give us any grammatical errors and is very clear what the article is about – a skin lesion classifying tool using AI.
* The summary isn't too technical but is still specific enough to tell us they looked into seven different skin diseases.
* It was a much better summary that could have been written by a human



* We faced various challenges and risks while working on this project
* When starting, it was a challenge to set up our environments, which included installing things consistently and deciding what and how we wanted to use different tools for our goals.
* On top of that, we had to navigate working remotely while still using an agile methodology.
* It was important that we communicated and updated each other regularly, especially when dealing with version control and sharing code among multiple people.
* Using Microsoft Teams to chat, GitLab to organize the edits in our code, and daily meetings to discuss updates on our work, we successfully established an efficient and comfortable workflow.
* The data itself was difficult to handle and create goals or applications, so we took our time carefully understanding the articles and patterns we could find before just jumping to a final product.
* The machine learning step was challenging and risky because each model would take time, memory, and resources to research and execute.
* Some approaches took so long that our computers would crash while others gave us results that were worse than non-ML techniques.
* Additionally, the findings/outcomes of these approaches were uncertain until we implemented them and evaluated the results.
* If we ran a model for 2.5 hours just to find that it created summaries that weren't accurate, we would have to go back
* As a result, we had to research models thoroughly to decide if we believed it would be worth the effort to try applying a model to our data.
* These risks could be mitigated with more powerful resources, but this would cost more money.



Lessons Learned

* Hybrid work environment – using agile methodology
* Tools and libraries – got more familiar with all of them
* Data science process and steps
* Explored natural language processing methods and capabilities
* Optimizing the methods for NLP with better run times or storage use

Follow on work/improvements

* Explore additional ML models we could apply and try re-training our current model with the GDIT data included.
* Identify more useful features that could impact querying or grouping articles like important people, dates, or technologies
* Implement a search history and results filtering in the GUI