NoteSmart: The Content-Based Notes Optimization Application

Caroline Bromberg

The University Of Michigan

College of Engineering Honors Program

Department of Electrical Engineering and Computer Science

EECS 441: Mobile App Development

April 6, 2023

Introduction

Most people need a way to organize their thoughts, whether for academic purposes or personal tasks, and many do this through keeping notes. However, the accumulation of notes over time can lead to disorganization, making it difficult for individuals to locate specific information. The absence of a reliable mainstream application for efficiently organizing and conducting conceptual searches through the content of notes has been a major setback for many.

NoteSmart offers an innovative solution to this challenge. This mobile application allows users to create notes and automatically generates relevant tags by parsing the content. Using these tags, NoteSmart groups together notes with similar content, making it easier for users to evaluate their notes. Instead of having to scroll through a large number of old notes to find specific content, users can search the tags for a certain topic and see all notes that touch on that topic. NoteSmart leverages complex natural language processing algorithms and Non-negative Matrix Factorization to parse the content and generate tags that are relevant and helpful to the user.

To parse the note content and create the relevant tags, NoteSmart uses advanced natural language processing algorithms that provide a semantic analysis of the content. These algorithms identify the relationships between words and phrases in the text, which enables the application to extract essential information from the note content. By identifying the context in which specific words and phrases are used, NoteSmart can generate tags that accurately capture the essence of the note.

Non-negative Matrix Factorization is a machine learning technique that allows the application to look at the data within the notes and generate tags based on that information. This technique is particularly useful in identifying patterns within the note content, which helps in grouping notes with similar content. By utilizing non-negative matrix factorization, NoteSmart provides a comprehensive way to organize notes, ensuring that users can access their notes easily.

The uniqueness of the tag-generation feature in NoteSmart enables users to search for specific notes based on the tags. Users can search for a particular tag, and the application will return all of the user's notes with that tag, thus providing a more efficient way of organizing information. The user interface supports this feature by allowing users to choose a specific tag and see an ordered list of all notes with that tag.

Overall, NoteSmart has several advantages over many other note-taking applications and provides a more intuitive way of organizing notes. The use of natural language processing algorithms and non-negative matrix factorization ensures an accurate way of generating relevant tags, making the application more efficient in organizing notes. NoteSmart represents a significant milestone in the development of note-taking applications, as it combines the storage of user notes with the ability to automatically analyze and organize those notes without needing users to perform any additional actions.

Statement of Purpose and Problems Addressed

The NoteSmart application is an essential tool for organizing thoughts and ideas, with a primary objective to efficiently organize users' notes and make it easier to find information quickly and easily.

I aimed to investigate the level of note organization that is most helpful for users, because both over-generalizing notes and under-generalizing notes can be equally unhelpful. By exploring this, I was able to create a feature that allows users to view and group notes efficiently while still being able to retrieve relevant information. Additionally, I investigated the impact of these automatically-generated content tags on users' abilities to find specific information.

Moreover, different users have different needs and preferences when it comes to tags. Therefore, I further explored whether all users find the tags intuitive or whether different users may find some tags intuitive but not others. With this, I could guarantee the creation of a tagging system tailored to benefit the maximum number of users.

Finally, I evaluated the user interface of the NoteSmart application to ensure its ease and intuitiveness to use. I wanted to guarantee a user-friendly, visually appealing, and easy-to-navigate interface, as no application is useful without a well-thought interface to guide users. In order to evaluate the efficacy of the user interface, I gathered feedback from users in the form of user testing to ensure that the user interface was intuitive and could meet their needs. My hope is that this project will lead to the development of a tool that enhances note-taking productivity and provides a better user experience.

Methods Used

In order to achieve the primary objective of the project, to create an improved version of a notes application that can more efficiently organize users' thoughts and make it easier for them to find the information they need, I used a combination of different methods, algorithms, and investigative techniques. The NoteSmart application uses natural language processing and non-negative matrix factorization techniques to parse note data, create relevant tags, and store them in a database. Additionally, NoteSmart uses the development of a backend and API that can send note data and receive relevant tags. The simplistic and straightforward user interface creates an aesthetic experience that users will enjoy.

Database:

The NoteSmart application stores all tag and note information in a database using Supabase. Supabase is an open-source backend-as-a-service (BaaS) that makes it easy to store, retrieve, and manage data. NoteSmart uses Supabase's database features to store note data and tags in a structured format that can be easily queried and analyzed.

Non-negative Matrix Factorization Technique:

To begin, I used existing natural language processing and matrix factorization techniques to parse note data and create relevant tags. I leveraged natural language processing algorithms to analyze the text within the notes and identify important words and phrases. I then used matrix factorization techniques to identify latent factors within the notes, such as topic clusters, to generate accurate and relevant tags. By utilizing this technique, the NoteSmart application is able to take in any amount of notes with any amount of content within each of them, turn the content into topic clusters, and then assign a tag to that note for each of the topic clusters. This means notes with more variety of information may have more tags generated, while notes that focus solely on one topic may only have one or two tags associated with them.

User Interface:

The aesthetic user interface of NoteSmart was made using Swift. I used user-centered design principles to make the user interface as intuitive and easy to use as possible. NoteSmart also leverages existing design patterns and guidelines to create a visually appealing interface that is consistent with modern design trends. Specifically, NoteSmart's user interface has a simplistic and minimalistic feel to it, while still maintaining all functionality on each screen.

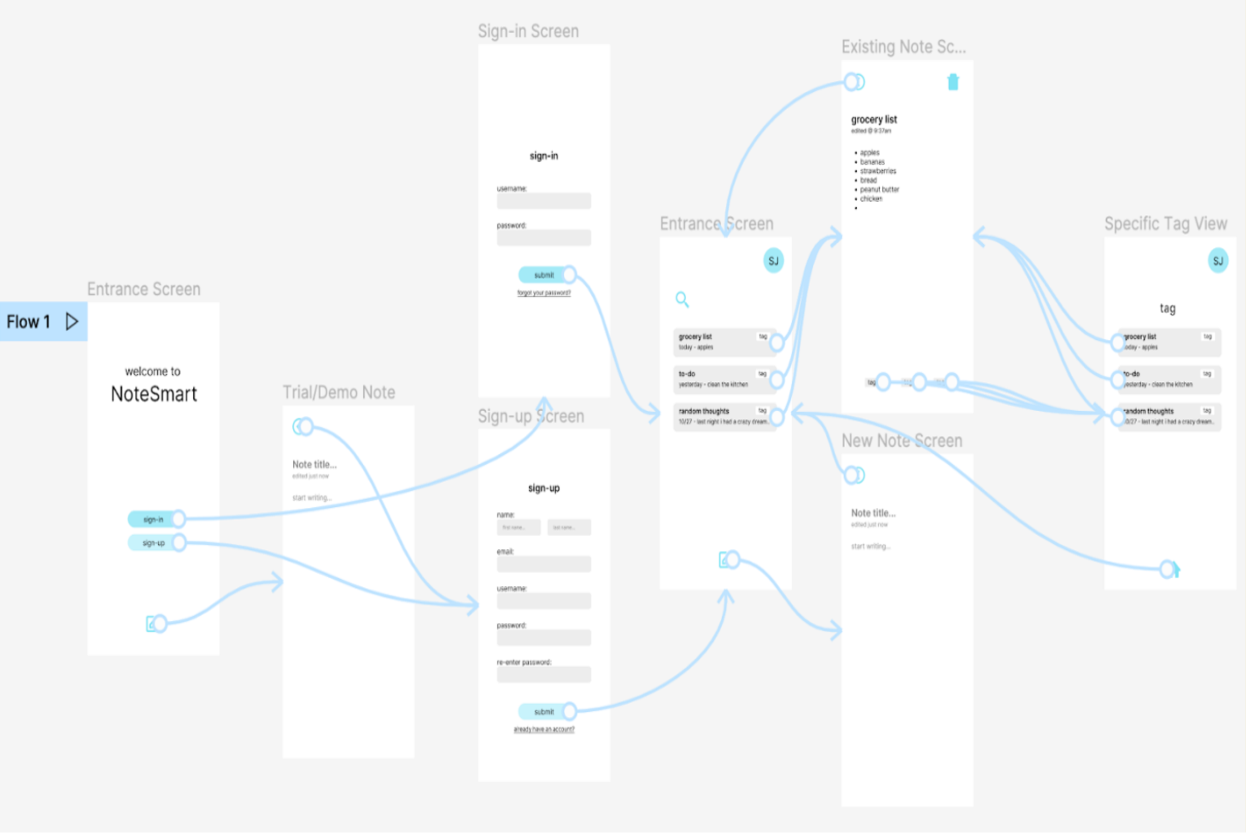


Figure 1: NoteSmart's user interface flow

Backend and API:

While the tag-generation is the primary feature of the app, a backend and application programming interface (API) are necessary in order to handle the application of the algorithms to the notes data, and communicate the data from the database and algorithms to the user interface. The backend and API calls were written in Python, with structured language query (SQL) to communicate with the Supabase database. Communicating directly with the Supabase database allows NoteSmart to easily retrieve and store note data and relevant tags, and the API calls allow NoteSmart to call the Non-negative matrix factorization algorithm on each note that the user writes in the app. The API calls are also responsible for sending information from the database to the user interface and vice versa, which is how NoteSmart can take the content from a note, analyze it to generate tags, and then send the list of generated tags back to the user interface to be presented to users.

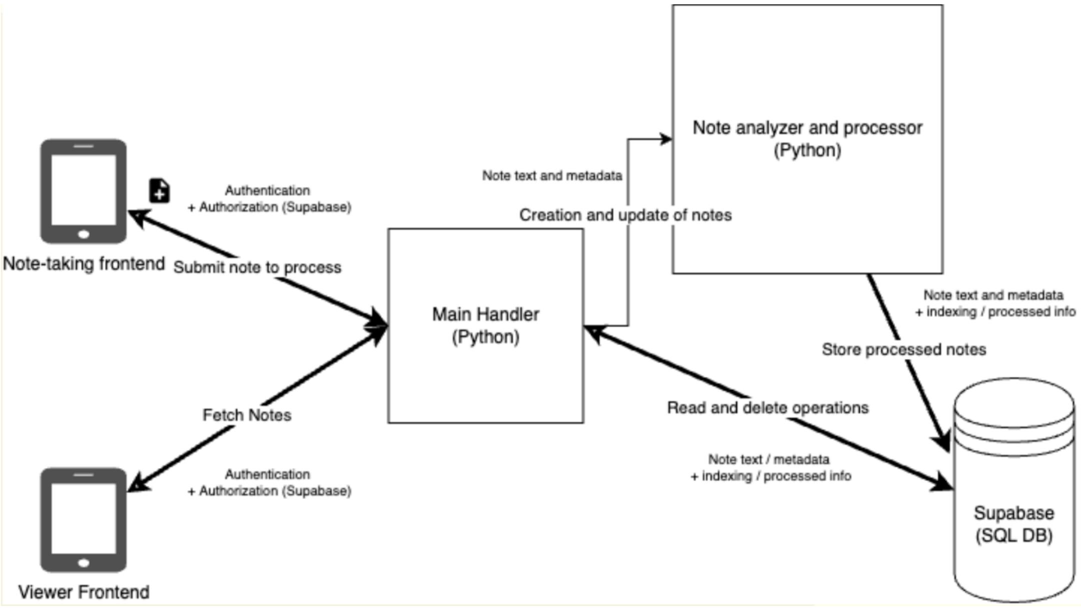
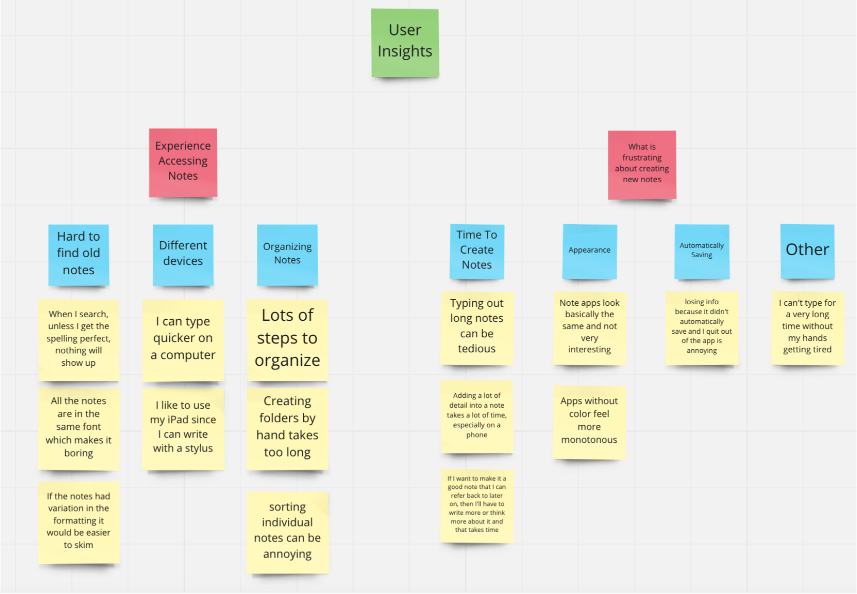


Figure 2: The architecture of the NoteSmart app

User Testing:

To test whether NoteSmart allowed users to complete their desired tasks more quickly than the iPhone default notes app, I recruited a group of participants who regularly use notes apps and provided them with a set of tasks to complete using both NoteSmart and the iPhone notes app. These tasks included searching for a particular note, finding a specific part of a certain note, and organizing notes into categories. I asked the participants to complete the tasks in both apps and timed their performance. I also asked the participants to provide feedback on their experience using both apps, including what makes it difficult to search for specific information on these kinds of apps.



When I search, unless I get the spelling perfect, nothing will show up

All the notes are in the same font which makes it boring

If the notes had variation in the formatting it would be easier to skim

I can type quicker on a computer

I like to use my iPad since I can write with a stylus

Lots of steps to organize

Creating folders by hand takes too long

sorting individual notes can be annoying

Typing out long notes can be tedious

Adding a lot of detail into a note takes a lot of time, especially on a phone

If I want to make it a good note that I can refer back to later on, then I'll have to write more or think more about it and that takes time

Note apps look basically the same and not very interesting

Apps without color feel more monotonous

losing info because it didn't automatically save and I quit out of the app is annoying

I can't type for a very long time without my hands getting tired

Figure 3: User insights on the difficulties of current notes apps

Improving Generated Tags:

To ensure that the generated tags are accurate and relevant, I performed additional research on natural language processing algorithmic techniques. I compared the generated tags of NoteSmart using Non-negative matrix factorization to using Latent Dirichlet Allocation. Latent Dirichlet Allocation is another common topic-analyzing technique, but was not as successful with NoteSmart. This was due to the nature of the types of notes that people commonly create, which tend to be simple and succinct. I found that shorter pieces of text, such as these notes, were better suited for Non-negative Matrix Factorization, while longer pieces of text, such as academic papers, were better suited for Latent Dirichlet Allocation. Discovering this difference between these two common models allowed me to fine-tune NoteSmart's models and ensure I was using the best algorithmic technique.

Results

The user testing showed that the inclusion of the tag-generation feature in NoteSmart had a positive impact on users' overall ability to create and search through their personal notes. Users were able to navigate their notes more efficiently and find specific notes with the exact content they were looking for, even if the notes were created a long time ago.

Users were able to find content in notes with an average of a 20% improved search rate when looking for specific information in the default iPhone application versus NoteSmart. This improvement supports that the tag-generation feature is a valuable addition to the world of note-taking apps.

Furthermore, the user testing identified multiple important design and functionality improvements. These enhancements included displaying all tags associated with a note on the notes homepage for easy navigation and organization, and enabling the ability to search for notes by content instead of only by tag. These changes have helped make NoteSmart a more effective and user-friendly notes application, and improved the overall user experience.

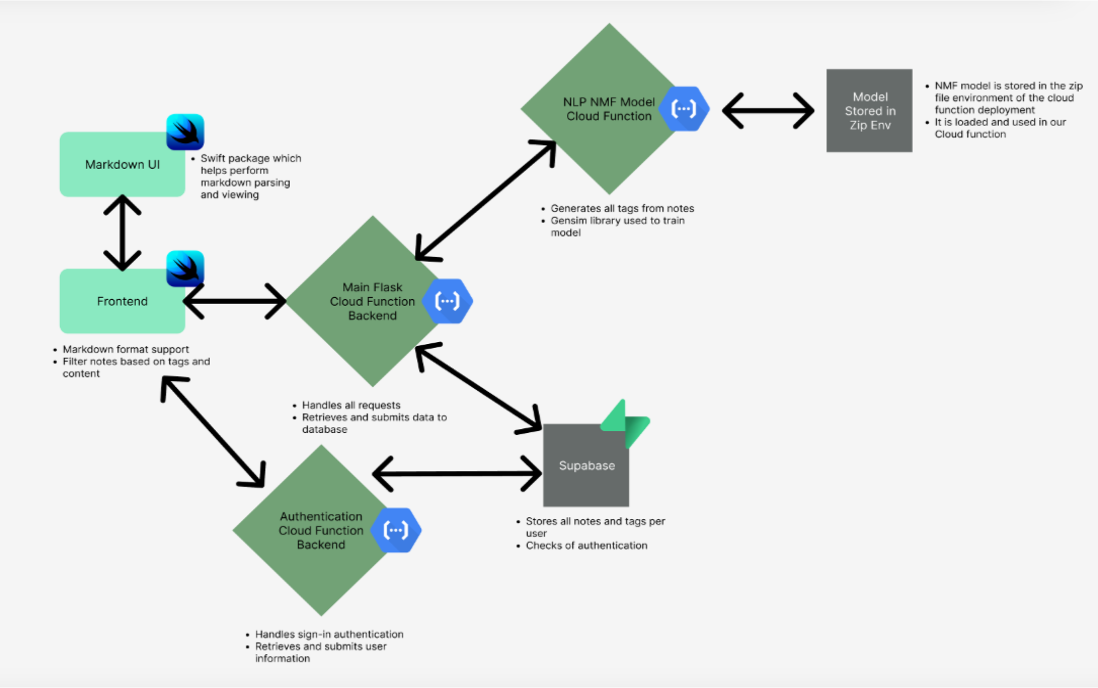

NLP NMF Model
Cloud Function
Model
Stored in
Zip Env
• NMF model is stored in the zip file environment of the cloud function deployment
• It is loaded and used in our Cloud function
Markdown UI
• Swift package which helps perform markdown parsing and viewing
• Generates all tags from notes
• Gensim library used to train model
Frontend
Main Flask
Cloud Function
Backend
• Markdown format support
• Filter notes based on tags and content
• Handles all requests
• Retrieves and submits data to database
Supabase
Authentication
Cloud Function
Backend
• Stores all notes and tags per user
• Checks of authentication
• Handles sign-in authentication
• Retrieves and submits user information

Figure 4: Improved Application Architecture Post-Testing

Discussion

The results of the testing confirmed the hypothesis that the tag-generation feature enhances users' note-taking and searching experiences. The findings indicate that NoteSmart's tag-generation feature is a valuable addition to the app and can significantly improve users' abilities to create and search for personal notes. Additionally, the design and functionality improvements made based on user feedback were found to be effective in enhancing the overall user experience.

However, the user testing had a few limitations. The sample size of the user testing was relatively small, of 10 users. Furthermore, the testing did not evaluate the long-term effects of using the application with the tag-generation feature. Future research could explore the tag-generation feature's long-term impact on note-taking and searching experiences, particularly as users become more accustomed to the app and its functionality.

Conclusion

Overall, the NoteSmart app, specifically the inclusion of the tag-generation feature, can significantly improve users' ability to search through and handle their personal notes. The results showed that users were able to navigate through their notes more quickly and find specific notes with the exact content they were looking for, even if the notes were created a long time ago. With an average of a 20% improved search rate when compared to a notes app that does not include tags these results suggest that tags can significantly improve the searchability and organization of notes, ultimately enhancing users' note-taking experience.

The design and functionality improvements made to the application based on user feedback were also found to be effective in enhancing the user experience. The clarifications made to the tag-generation feature, such as displaying all tags associated with a note and providing the ability to search for notes by tag, made it easier for users to navigate through their notes and find specific information.

Future research could investigate the long-term impact of the feature on users' note-taking and searching experiences, as well as explore additional design and functionality improvements that could further enhance NoteSmart’s usability. By improving the organization and searchability of notes, NoteSmart can enhance users' note-taking experiences, making the application an extremely effective tool for personal organization and productivity.

References

Chawla, R. (2018, June 20). *Topic Modeling with LDA and NMF on the ABC News Headlines dataset*. Medium. https://medium.com/ml2vec/topic-modeling-is-an-unsupervised-learning-approach-to-clustering-documents-to-discover-topics-fdfbf30e27df

Egger, R., & Yu, J. (2022, May 6). *A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts*. frontiersin.org. Retrieved February 6, 2023, from https://www.frontiersin.org/articles/10.3389/fsoc.2022.886498/full

Gensim: topic modelling for humans. (n.d.). https://radimrehurek.com/gensim/models/nmf.html

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011. http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

Welcome to Nimfa — Nimfa 1.3.1 documentation. (n.d.). https://nimfa.biolab.si/

What are the pros and cons of LDA and NMF in topic modeling? Under what situations should we choose LDA or NMF? Is there comparison of tw. . . (n.d.). Quora. https://www.quora.com/What-are-the-pros-and-cons-of-LDA-and-NMF-in-topic-modeling-Under-what-situations-should-we-choose-LDA-or-NMF-Is-there-comparison-of-two-techniques-in-topic-modeling