ArXiv Recommender System: Executive Summary

Introduction and problem statement:

The arXiv is a free, online repository of over two million academic STEM pre-prints and is a crucial tool for finding and sharing research. Despite its importance, users must routinely manually sort through broad subject categories (e.g. “mathematical physics”) to find papers of interest. In light of this, we asked the following question: is there a smarter way for a researcher to find new articles relevant to them?

Most automated recommendation systems like those of Spotify, Amazon, Netflix, etc. use their database of user past activity to improve their recommendations. Since the arXiv does not have accounts, we can only recommend papers based on their content. Therefore, we settled on the following concrete problems:

1. **Given a library and a query, yield the 5 most similar articles to the query.**
2. **What topics are present? Can we label the topic of a new query?**

There are two main difficulties present. First, recommendations and topic labels can only be evaluated subjectively by users. Second, the size of the library is limited by time and computational resources. A smaller library limits the possible recommendations.

Method:

- Selected 20,000 recent articles from the 5 most commonly occurring and co-occurring arXiv subjects as our library.
- Used NLP techniques to vectorize the library and query papers.
- Found the nearest neighbors (cosine similarity) of the query vector.
- Used clustering techniques to identify documents sharing a topic

Pitfalls:

- Content in abstracts often obscured as LaTeX math which cannot be parsed by state-of-the-art vectorizers (sentence transformers)
- Cosine similarity is a less effective measure of content similarity for sparse representations
- Clustering algorithms work poorly on sparse representations.

Results and Next steps:

We created multiple models based on different text vectorization schemes (bag of words, word2vec, doc2vec, sentence transformers) and created a separate test set to evaluate the recommendations each model generated. (See the README file for details). We built a front-end app for the best performing model. It allows the user to input a paper by arXiv ID and returns the 5 most similar articles as well as top keywords for the topics found in the query and recommended articles. This is supported by two different BERTopic clustering models. Currently these models are limited in accuracy, unable to categorize about one third of our library. We hope to improve this in the future and leverage them to obtain more precise recommendations from larger libraries.