Free Topic: Reading Assistant

Christopher Rock (cmrock2) (Captain)
Zichun Xu (zichunx2)
Kevin Ros (kjros2)

25 October 2020

1 Project Proposal

1.1 Overview

In the early months of 2020, the now named SARS-CoV-2 virus was rapidly spreading across the countries of the Pacific and jumping to new locations in every corner of the globe. The flu pandemic that had long been predicted was happening - except it wasn't the flu. First in China, and soon throughout the world there was talking, writing, researching, and publishing about the virus. This was the first pandemic in the smart-phone era, and the only thing that seemed to spread faster than the virus was information. Governments, medical organizations, companies, and every institution imaginable began pushing out not just information, but also guidelines, rules, and policies.

"Information overload" is something people in today's society are accustomed. Most people develop methods of coping with the huge amount of information available. We filter things through trusted sources, prioritize information that is actionable, and change our mental model of the situation as necessary.

However in the face of a new dangerous situation every piece of information becomes potentially critical. Dynamic situations such as a pandemic require quick reactions to new knowledge. It is our experience that those in leadership and decision-making roles are pushed a large number of documents and expected to be up to date on this rapidly expanding corpus of information. New situations create organizational chaos, and the individual's strategies to limit information overload break down. Trusted sources are more difficult to identify when information comes from many sources simultaneously. And because all information is potentially actionable, all information must seemingly be reviewed.

Initially the challenge is simply to read and understand the documents sent. However the difficulty quickly becomes identifying what is new knowledge. New documents may have significant overlap with prior knowledge. Differences between documents must be reviewed, and often the progression of changes is important.

Information retrieval, text mining, and recommender systems have developed algorithmic strategies to identify and extract useful information from text-based knowledge. The focus of these tools has generally been to pull relevant documents via (search), or push potentially interesting documents (recommender). These techniques can be modified to assist a reader in identifying new useful information.

Our goal is to create a reading assistant tool that allows a user to maintain a collection of "seen" or "read" documents (reflecting the current knowledge of the user) and provides novelty scores based on new documents introduced to the collection. Given a new document, the reading assistant tool will compare the document to all "seen" documents, and provide the user with measures indicating how the new document differs from the document collection. In this way, potentially useful new documents can be efficiently prioritized by the user.

1.2 Project Description

The task of our free topic is to design and implement a reading assistant software tool that helps users determine the novelty of never-before-seen documents based on previously-seen documents. Each user will have a collection of read documents, known to the reading assistant. When the user is provided a new document, the reading assistant will quickly scan the user's read document collection and score the new document (or sections of the new document). This score will reflect how novel the new document is relative to the previously-read documents. Ideally, this will provide the user with a high-level understanding of the importance of the document, allowing the user to better optimize their time. There are many users who would benefit from such a tool. As we discussed in Section 1.1, medical researchers and doctors could use a tool to help sift through and sort the vast amount of information provided during events such as a global pandemic. In academia, researchers could leverage this tool to filter research papers for information relative or novel to their current work. Outside of academia, the general public could use this tool to augment online browsing, as such a tool would allow them to quickly look up previously-read documents and news articles, and interpret new articles in the context of what they have already read. To our knowledge, no such tool currently exits.

We will use the MetaPy toolkit[1] to provide a suite of ranking and evaluation methods for our tool, along with the publically-available CORD-19 Coronavirus document data set[2] as our training and testing data. Aditionally, we will use the Python programming language. To create the tool, we will begin by leveraging our understanding of the BM25 ranking algorithm (which measures document similarity) to construct an "inverted BM25" distance function (which measures document difference).

In order to demonstrate the usefulness of our tool, we will manually score a subset of documents in terms of similarity to a collection of seen documents. In some cases the seen documents will be randomly selected, and in other cases they will all be of a certain topic. Then, we will pass the scored documents to our tool and see if it categorizes the documents in line with our manual scoring. We discuss a rough timeline in Section 1.3

1.3 Workload

We will spend the first 20 hours defining and understanding the project scope. Here, we will begin by defining what it means for two documents to be distinct (or similar). We will also attempt to quantitatively define a distance measure between documents or paragraphs. Additionally, we will define the scope of "seen" and "unseen" documents. That is, we might need to assume that the reader has read many documents for recommendation to be effective (otherwise many documents will be considered novel).

Following this, we will spend the next 20 hours implementing our distance measure using Python and MetaPy. We will likely begin with an inverted version of BM25, but it is difficult to know how well it will work for measuring document difference. Thus, we expect that a significant portion of the 20 hours will be testing out and debugging various implementations, fine-tuning parameters, curating the training and testing documents, and adjusting any initial assumptions in light of new

[1] https://github.com/meta-toolkit/metapy

[2] https://www.semanticscholar.org/cord19

evidence. Once we decide on a specific implementation, we will define various evaluation measures in order to determine the effectiveness of our tool.

The remaining 20 hours will be spent evaluating the tool and tuning any parameters. Given the subjectivity of relevance scores, we will likely need to manually judge documents. For example, this could include randomly choosing a set“already seen” documents, and hand-labeling additional documents as “very similar”, “somewhat similar”, or “not similar” to “already seen” documents. Then, we would see if the tool’s scores corresponded to our similarity classifications.

In the case that we overestimated the time it takes to complete the aforementioned tasks, we will fill the remaining time by making our tool more robust, more user-friendly, or more expansive. This will be accomplished by introducing various similarity score measures (such as word embeddings), a command-line interface, and considering additional data sets, respectively.