COVID-19 Randomized Controlled Trial (RCT) Summarization

Amanda Steigman, Han Xu, Lin Jiang,
Swapnav Deka, Suman Tripathy

Faculty Mentor: Chunhua Weng
PhD Mentor: Tian Kang

Statement of Problem

- **Information Overload**
  - Hundreds of thousands of medical research papers are released every year, and it is impossible for doctors to keep up with all of them
    - i.e. PubMed, a major biomedical literature database, currently has over twelve million citations and adds around 40,000 more every month
- **Inadequate Health Literacy**
  - National guidelines regarding patient oriented materials dictate the reading level should not be above a fifth-grader's
  - Less technical summaries of medical papers would make relevant info more accessible to the average person

Project Goal

- Our main goal is to make it easier for people without medical background to quickly grasp the essential ideas from a large medical corpus

- This can be achieved through two methodologies:
    1. To turn one complex abstract into a concise and understandable summary
    2. To combine and compare multiple related abstracts at the same time

Project Deliverables

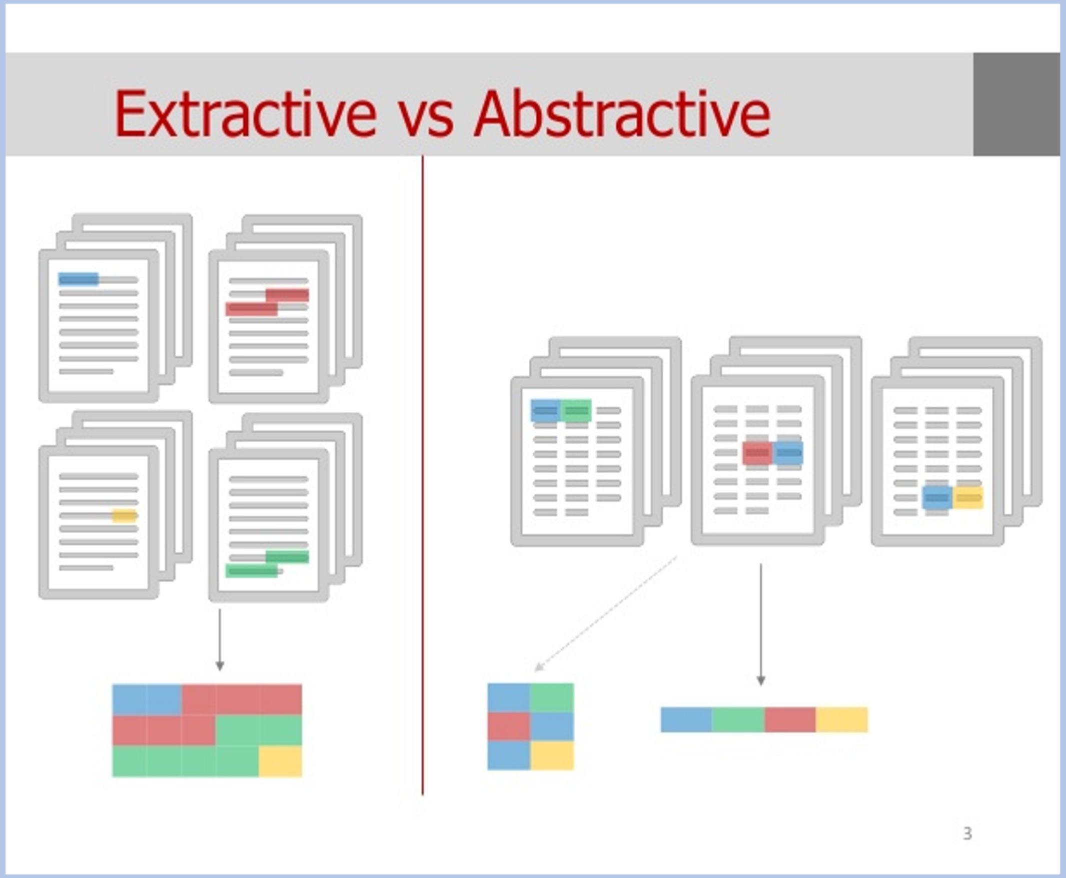
- **Single Document Summarizer**
  - Provide a high-level readable summary that preserves the main takeaways and most pertinent information from the original literature
- **Multi Document Summarizer**
  - MDS using keyword search
    - After a user search for a keyword, this tool can summarize multiple retrieved abstracts into one comprehensive summary
  - MDS using all abstracts
    - This tool outputs a global view of what topics this corpus covers
    - Users can define these topics by looking at the most relevant words, and then multiple abstracts within one topic get summarized

Single Document Summarizer

Text Summarization Methods

- **Extractive Summarization**
  - Extracts words and word phrases from the original text to create a summary

- **Abstractive Summarization**
  - Learns an internal language representation to generate more human-like summaries, paraphrasing the intent of the original text



Attempt at the Abstractive Method

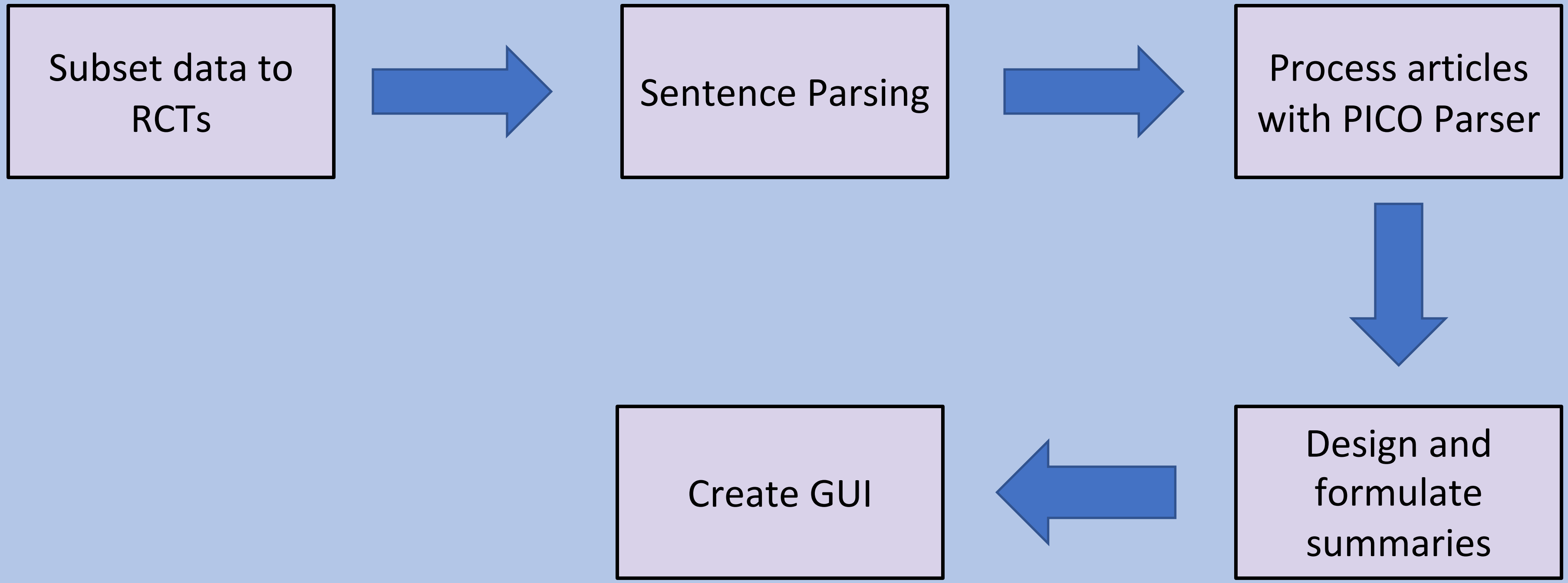
- Abstractive summarization would better fit our goal, since it allows for more flexible and natural sounding summaries
  - In practice, however, it is difficult to achieve good results with abstractive summarization, and the process is much more computationally expensive

- Our team attempted to fit a **sequence to sequence model**, which is abstractive
  - Neural language model that takes in one sequence as input and derives from it, using an encoder-decoder Recurrent Neural Network (RNN) framework, another sequence
  - Required too much memory to run on personal computers and cost of purchasing a Virtual Machine with GPU was ~$3000/month

Pivoting to the Extractive Method

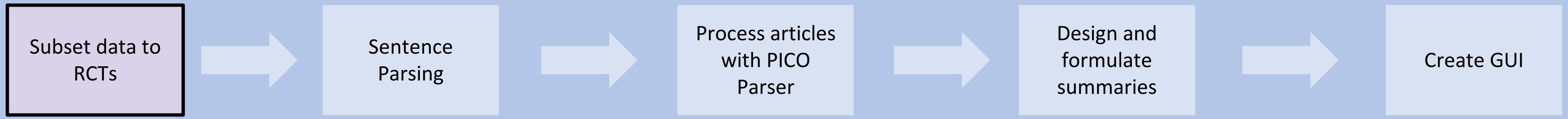
- Our team then changed strategies to use an extractive summarization method known as the **template method**
  - A systematic summarization technique with the goal of outputting a structured, tabular view of medical texts using relevant information of interest taken directly from the abstracts
  - easy to implement - once we extracted the necessary elements from the abstracts, we just placed them properly in our template summary

Template Method Workflow



Subset Data to RCTs

- Original LitCovid dataset contained around 24,000 abstracts
- Our team subsetted this to only include RCTs
  - RCT abstracts are more structured than other clinical trial reports, so they are a good starting point for extractive document summarization
  - Contain section headers for background, objective, methods, results, and conclusion, so it is easier to parse relevant information for summaries
- Final sample included ~3,800 abstracts



Process Articles with PICO Parser

- PICO elements:
    - (P)opulation / Participant
    - (I)ntervention
    - (C)omparison
    - (O)utcome

- PICO Parser outputs include two sections for each sentence:
    - “Evidence Elements”: store the PICO elements extracted from this sentence
    - “Evidence Propositions”: statements composed of extracted PICO elements


```
        {
            "Section": "RESULTS",
            "Text": "The odds ratio of a favourable outcome with hydroxychloroquine was 1.11 ( 95 CI 0.72 to 1.69 ) ( p=0.20 ) .",
            "Evidence Elements": {
                "Participant": [],
                "Intervention": [
                    {
                        "term": "hydroxychloroquine",
                        "negation": false,
                        "UMLS": "",
                        "start": 44,
                        "end": 62
                    }
                ],
                "Outcome": [
                    {
                        "term": "odds ratio of a favourable outcome",
                        "negation": false,
                        "UMLS": "",
                        "start": 4,
                        "end": 38
                    }
                ]
            },
            "Evidence Propositions": [
                {
                    "Intervention": "hydroxychloroquine",
                    "Observation": "1.11",
                    "Outcome": "odds ratio of a favourable outcome",
                    "Count": ""
                }
```


Example output of PICO parser

Design and Formulate Summaries

- Based on extracted PICO elements, designed a few templates that allowed for flexibility of having none, one, or multiple Participants, Interventions, and Outcomes
- We used elements extracted from the Objective and Conclusion sections, as they were the most relevant to our summaries
  - i.e. Results section was too technical, Discussion section did not provide useful info on the PIO of the study
- Extracted summary consists of background information, parsed PICO elements, and concluding sentence

Similarity Metric

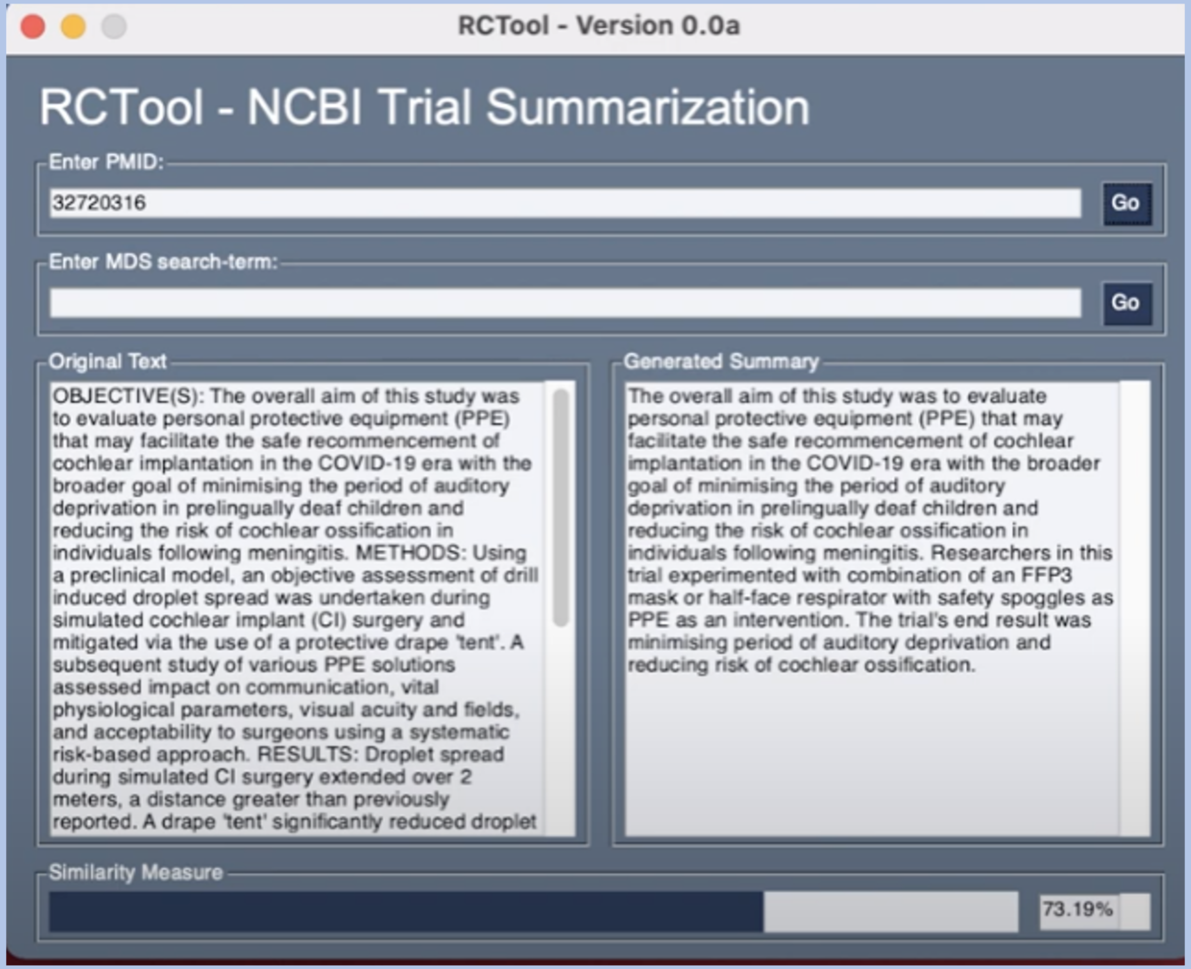
- After generating a summary for an abstract, it's important to know how accurate it is
- Used Bert Sentence encoder to create embeddings for every abstract and summary
- Then use cosine similarity between Bert vectors for an abstract and summary
  - Below, we visualize a "poor" summary and a "great" summary

Researchers in this trial experimented with without as an intervention. CLINICAL RELEVANCE.

The overall aim of this study was to evaluate personal protective equipment (PPE) that may facilitate the safe recommencement of cochlear implantation in the COVID-19 era with the broader goal of minimising the period of auditory deprivation in prelingually deaf children and reducing the risk of cochlear ossification in individuals following meningitis. Researchers in this trial experimented with combination of an FFP3 mask or half-face respirator with safety spoggles as PPE as an intervention. The trial's end result was minimising period of auditory deprivation and reducing risk of cochlear ossification.

Subset data to RCTs → Sentence Parsing → Process articles with PICO Parser → Design and formulate summaries → Create GUI

RCTool: RCT Summarization GUI Demo



Subset data to RCTs → Sentence Parsing → Process articles with PICO Parser → Design and formulate summaries → Create GUI

Multi Document Summarizer

Document Retrieval Using Keyword

- All PICO terms from the COVID documents were clustered into categories based on Jaccard similarity

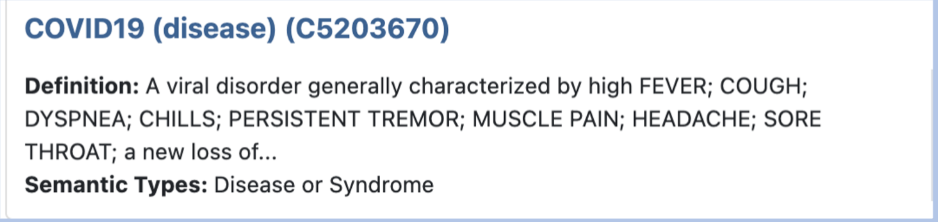
e.g. topic = "Covid-19"
relevant terms = ['coronavirus', 'covid', 'covid-19', 'Coronavirus', 'COVID-19', 'COVID19', 'CORONAVIRUS', 'COVID-2019', 'COVID', 'Corona Virus Disease-19', 'corona virus', 'corona disease' …. etc.]

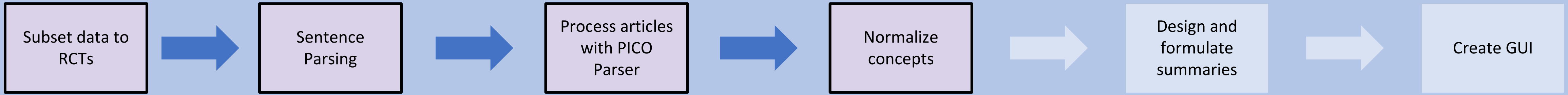
- Each document is tied to all topics whose PICO terms it contains
- When user inputs a keyword, it gets tied to a topic also, and all relevant documents containing those topics are retrieved

Unified Medical Language System (UMLS)

- The UMLS is a set of programmable files and software that allow the interoperability of computer systems by standardizing many health and biomedical vocabularies
  - Using UMLS makes it feasible to develop a search-term system that can map an array of terms to a single UMLS code for more efficient retrieval of information

- The end result is that linking terms and codes across doctors, pharmacies, and insurance companies allows for the coordination of patient care, among other things





Multi-Document Comparison

- Input files: outputs of search-term system
- Output: a tree graph
- Goal: compare the different PICO elements for different input abstracts

- From the visualization:
  - The users can have a general view of the different focuses and results of articles on topics they care about
  - It’s easier to compare abstracts with converging or diverging outcomes for the participant/drug under the same intervention

Multi-Document Comparison Example

- Each leaf represents one article
  - First layer represents (P)opulation / Participant, second layer represents (I)ntervention, third layer represents (O)utcome, fourth layer represents observation of Outcome, last layer shows the pmid of articles
  - **Example:**
    - Search term: ‘rheumatic disease’, search type: ‘P’

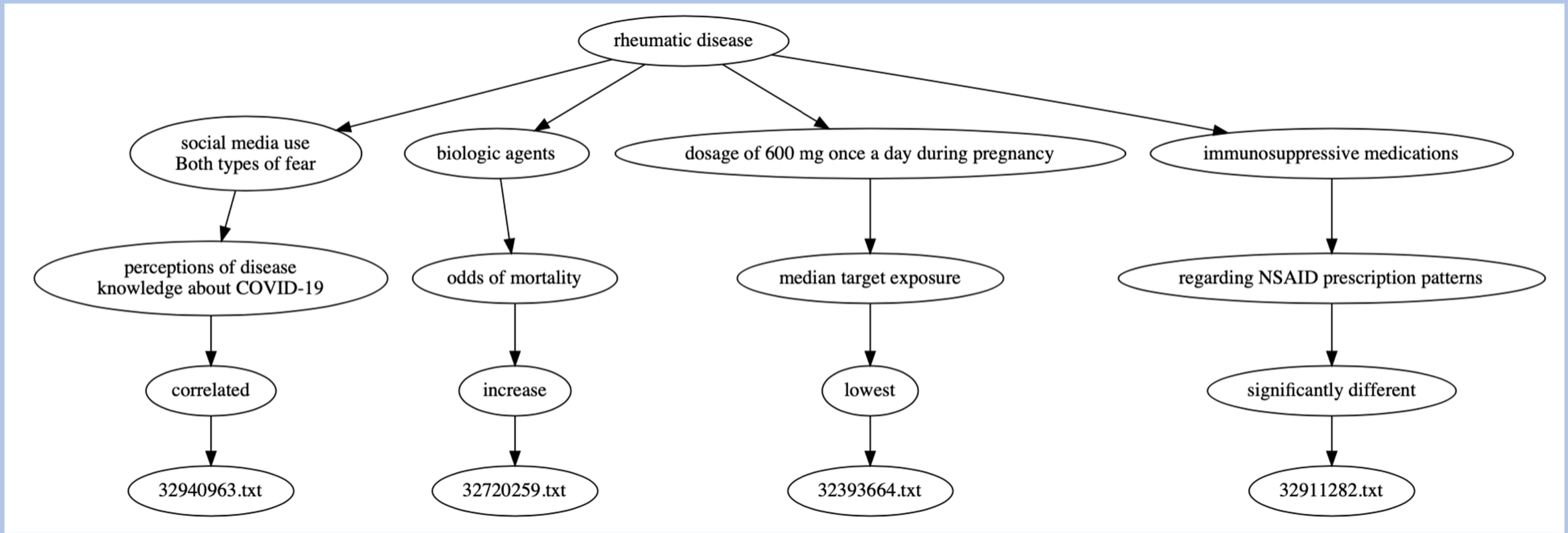
P

I

O

Observation

Doc_ID

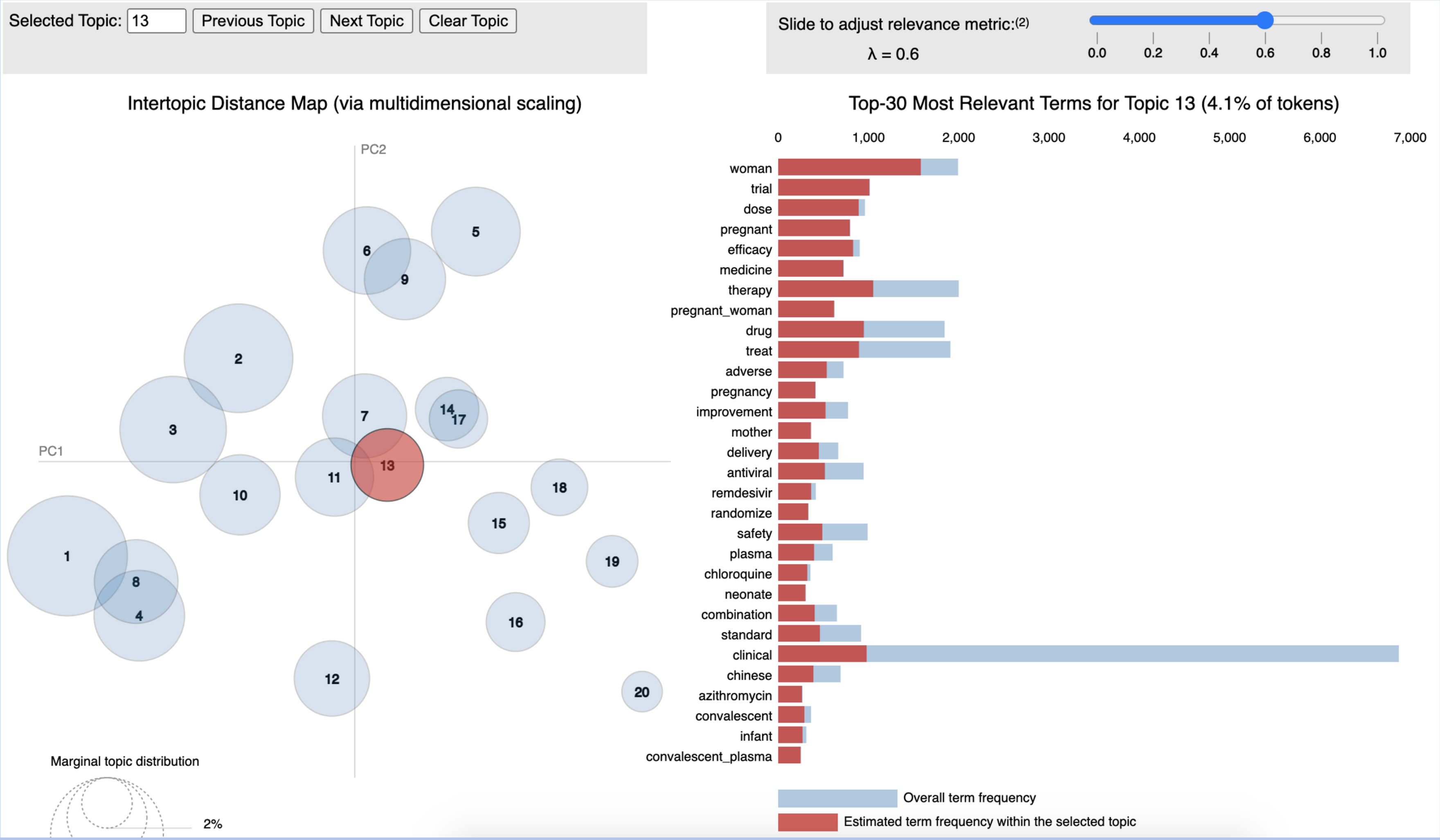


Example Graph

Multi-Document Topic Modeling

Topic Modeling

- Global method to get overview of areas that input documents cover
- Original abstracts were clustered into multiple topics
- Topic can be defined based on the most relevant words

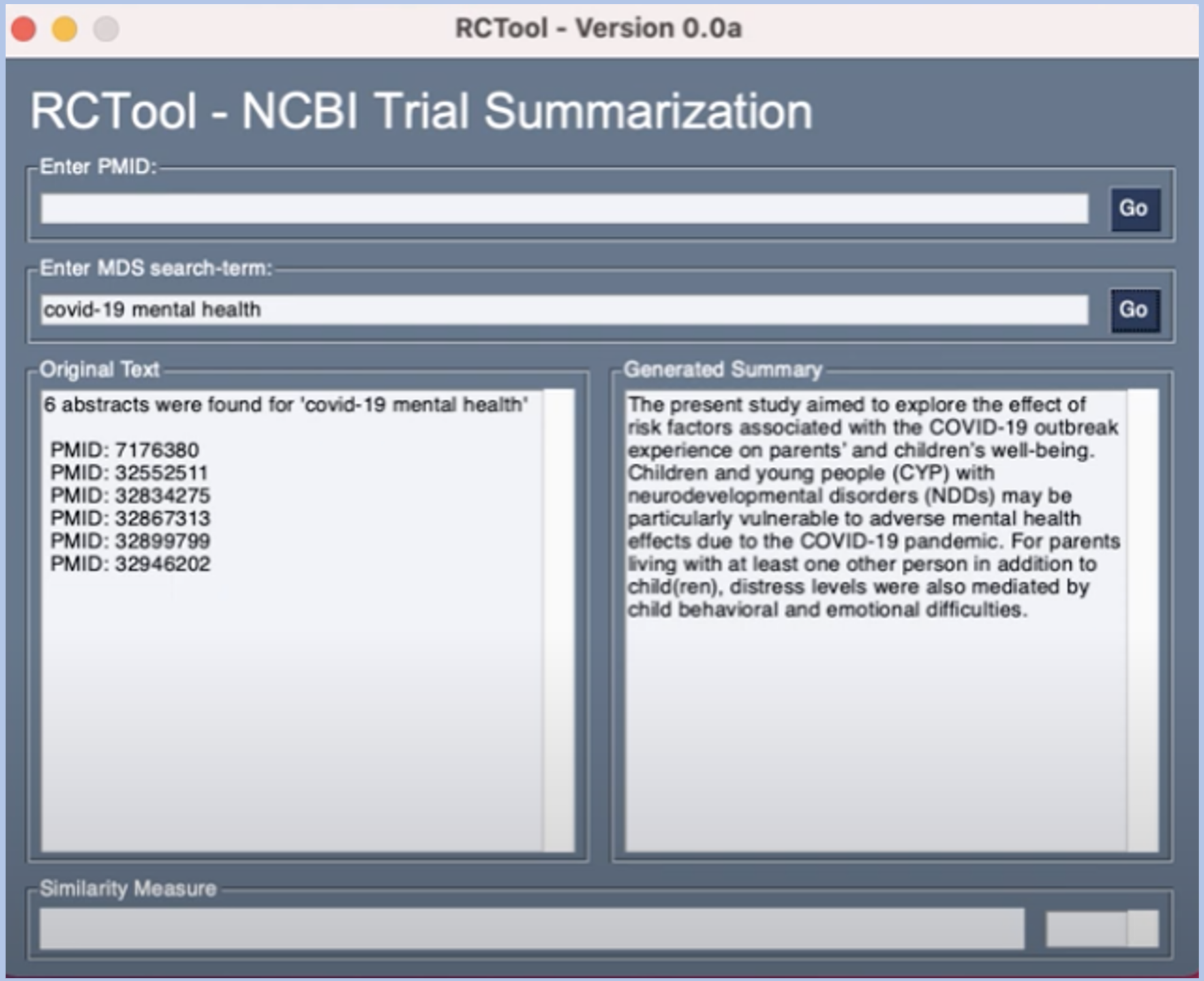


Multi-Document Summarization (MDS)

- We used an open-source package for MDS called Potara (https://github.com/sildar/potara)
- This system performs multi-document abstractive summarization via sentence fusion and Integer Linear Programming (ILP)
- The system includes three steps:
  - **Sentence clustering**: regroup similar sentences into clusters
  - **Sentence fusion**: build a directed word graph from clusters and fused sentence are obtained by finding commonly used paths in the graph
  - **Sentence selection**: use concept-based ILP to extracts sentences that cover as many important concepts as possible while ensuring the summary length is within a given constraint

- Multi-Document Summarization + Search-term
  - For users who have interest in particular topics
- Multi-Document Summarization + Topic modeling
  - Use the output of hierarchical topic modeling for better results
  - For users who don’t have topics of particular interests

RCTool: MDS Summarization Demo



Thank you!

Any Questions?

Appendix

Future Work

- **For template method:**
  - Improve templates to deal with different word tenses
  - Choose wisely when there are redundant PICO elements
  - Make sure that intro and conclusion sentences that we pull from the abstract itself are meaningful
- **For multi-document summarization:**
  - Manual reviews show that current performance of MDS has room for improvement
  - Modify existing MDS tool to improve the system’s performance with medical articles
  - Implement document retrieval using UMLS normalized terms instead of Jaccard Similarity for a more standardized approach

Template Method: Limitations

- The PICO parser doesn’t always give consistent results about PICO elements or evidence proposition, and a lot of files have nothing extracted out of them
- There’s a variety of RCT files, and it’s difficult to come up with a fixed template that fits all
- It’s difficult and subjective to merge “similar” PICO elements together

- These are all things that can be **improved** in future work!