Extractive Search for Analysis of Biomedical Texts

Daniel Clothiaux
Bioplx
Boulder, Colorado, USA
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
dclothiaux@bioplx.com

Ravi Starzl
Bioplx
Boulder, Colorado, USA
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
rstarzl@bioplx.com

ABSTRACT

Extractive search has been used to create datasets matching queries and syntactic patterns, but less attention has been paid on what to do with those datasets. We present a two-stage system targeted towards biomedical texts. First, it creates custom datasets using a powerful mix of keyword and syntactic matching. We then return lists of related words, provide semantic search, train a large language model, a synthetic data based QA model, a summarization model over those results, and so on. These are then used in downstream biomedical work.

CCS CONCEPTS

• **Information systems** → **Document filtering**; **Question answering**; **Expert search**.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Daniel Clothiaux and Ravi Starzl. 2022. Extractive Search for Analysis of Biomedical Texts . In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3477495.3536328

1 INTRODUCTION

In many user facing search engines, the goal is to present the user with a small number of select documents. However, when searching biomedical corpuses, it is often necessary to search for more comprehensive results, such as all unique interactions with a protein. In this case, just the best ten results are not enough, it is necessary to do more in depth analysis of the full set of documents. Instead, we use extractive search [5] to return the full set of documents matching a keyword or syntactic search query, with various tweaks on their model. We then throw everything we can think of at this set of documents, which we call a cube.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3536328

2 OVERVIEW

We create a paper database consisting of all of the documents in the Semantic Search database [1] that are tagged as "biology", "medicine", or "chemistry" combined with the set of full papers from Pubmed Open Access. We also create a database of internal documents.

Our system consists of two modules. The first is an extractive search module, which pulls a set of documents matching the user query from the above databases. The second is an analysis module, which presents a variety of tools to help the user understand the cube.

2.1 Extractive Search

Our extractive search module follows that of Taub-Tabib [5], with keyword, semantic, and syntactic search. In this domain, exact matching is often extremely useful, as when a user searches for a given gene or protien, they often want matches of that gene or protein, and not fuzzy matches of words that occur in a similar context. Syntactic search extends that concept from keyword matching to matching syntactic patterns in a parse tree, with chosen words held constant.

Dependency parses are created of each sentence in the databases, with entities tagged using the BERN 2 parser [4], chosen as our search experts and biologists preferred its ontologies to competitors. These are stored in a syntactically aware index using Odinson [6]. Search queries are parsed, with a light markup similar to [3] indicating which words are held constant, which words can vary, and what entities they must match. For example, the query "^RBD $attaches to the :ACE2 receptor", indicates that sentences should be found with synonyms matching RBD, in this case RBD and "reception binding domain", the wildcard ACE2 representing any gene or protein (as that is ACE2's entity), and finally the word "attaches" with the proper dependency relationships with RBD and and ACE2. In the case of acronyms, "^" matches the expanded acronym, otherwise we take words matching the top-n nearest word vectors.

This search returns a list of documents matching the query, for the user to see and adjust their query in response to the results. They can freeze the query and run further queries against the subset of matches to the first query, or if the results match a minimum size (if the user only finds two abstracts, its not really worth doing anything more with), can create a cube over the results of the search.

2.2 Analysis

Once we have a cube created, we create a suite of tools to help the user analyze the cube. First, we we mark the metadata of the

documents in the index that they are in the appropriate cube. We also create dense embeddings for each document in the database, and similarly mark the documents in the cube. This allows the cube to be further queried, and allows fuzzy semantic search over a set of documents matching precise criteria.

Secondly, in the case of syntactic search, we return a list of entities matching the wildcard, sorted by frequency, saving a CSV for further downstream analysis. Clicking each entity returns the list of documents where that entity was a match.

Finally, we create a suite of neural tools to use to help users explore and understand the database. We finetune a large language model, train a neural QA model including both question and answer generation, and a summarization model over the cube.

For the large language model, we finetune GPT-J 6B [7], as scientific texts make a large portion of its training data, on the cube, as well as a summarization system. For the QA model, as this cube almost certainly doesn't exist in a QA database anywhere, we finetune BART to generate answer-question pairs from a document following [2]. Once we have the QA generation model, we use it to generate a synthetic set of questions and answers from the cube. This synthetic dataset is then used to train a QA model.

We run various statistics over the cube as well, counting occurrences of other biomedically relevant terms outside the syntactic search, such as which other genes or diseases were the most common overall. We use these frequency counts to create a TF-IDF based recommendation system for both terms and papers outside the cube.

2.3 Industrial Applications and Future Work

Once we have the cube, we can use the suite of analysis tools to help us better understand the areas of interest. In the past, we have successfully created datasets related to Covid, such as the interactions with the ACE2 proteins touched on above, and for general search to help deal with the deluge of Covid related papers. These have mostly been used as a sanity check: did our searchers pull all of the relevant papers? So we pulled all the papers that could be relevant, constructing cubes from the union of multiple searches, with full expansions of abbreviations, then checked against what we had found.

We also created a cube relating to Bioplx's main product, including genes indB and sbnA, which are used to bind iron in low iron environments (such as in the blood), and the toxin sprA1, which is used as a kill switch to kill the bacteria in low iron environments the above genes activate in. Not being a biologist, I used this cube to help better understand what the Bioplx system was doing. This shows that the process of extractive search followed by a wide variety of tools can be used with great effect for both experts and nonexperts in biomedicine.

At the moment, we created a basic search interface for this, which can search and create cubes. However, the downstream systems we apply to the cube are currently fragmented, and require using terminal and running a bunch of scripts, which restricts their use. Our initial experiments persuaded us of the usefulness of them, so the next step is to create a better UI that anyone can easily use. We will integrate them more tightly as well, such as by integrating search results with the language model and QA system.

We will also consider experimenting with few shot learning and adaptive search techniques, to allow people to easily construct their own ontologies, and not be dependent on the entity classes the parser defines for them. We also want to expand the synthetic data system we used to train the QA system, to train a human-in-the-loop QA bot, that can reason over multiple steps with the user involved.

3 PRESENTERS

BioPlx is an advanced microbiomics company developing non-antibiotic methods for the control of infectious disease, including multiple-drug resistant "superbugs".

The company uses targeted microbiological and immunological agents to decolonize (eliminate) undesirable or pathogenic organisms, and then to recolonize with safe replacement organisms that durably occupy the same microbiomic niches, and so prevent pathogen recurrence.

Daniel Clothiaux is a PhD student at Carnegie Mellon Univserity, currently in absentia while working at Bioplx. Starting in natural language processing, he then previously worked on image generation, focusing on handwriting, and conducted a major survey of English handwriting before shifting back to NLP. His advisor, Ravi Starzl, is the cofounder and CEO of Bioplx. Currently directing the Starzl Lab, he also co-directed the Biotechnology, Innovation, and Computation Program at the Language Technologies Institute and Carnegie Mellon.

ACKNOWLEDGMENTS

To Tim Starzl, Samantha Kent, Mariam Crow, Scott Clark, and Jeanette Rimby for help with what they thought would be most useful for searching biomedical texts. Extra thanks go to Scott Clark for many discussions on what would be most useful for him in his work as a search expert.

REFERENCES

[1] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262* (2018).

[2] Siamak Shakeri, Cicero Nogueira dos Santos, Henry Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. *arXiv preprint arXiv:2010.06028* (2020).

[3] Micah Shlain, Hillel Taub-Tabib, Shoval Sadde, and Yoav Goldberg. 2020. Syntactic search by example. *arXiv preprint arXiv:2006.03010* (2020).

[4] Mujeen Sung, Minbyul Jeong, Yonghwa Choi, Donghyeon Kim, Jinhyuk Lee, and Jaewoo Kang. 2022. BERN2: an advanced neural biomedical namedentity recognition and normalization tool. (2022). arXiv:2201.02080 [cs.CL]

[5] Hillel Taub-Tabib, Micah Shlain, Shoval Sadde, Dan Lahav, Matan Eyal, Yaara Cohen, and Yoav Goldberg. 2020. Interactive extractive search over biomedical corpora. *arXiv preprint arXiv:2006.04148* (2020).

[6] Marco A Valenzuela-Escárcega, Gus Hahn-Powell, and Dane Bell. 2020. Odinson: A fast rule-based information extraction framework. In *Proceedings of the 12th Language Resources and Evaluation Conference*. 2183–2191.

[7] Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.