Cydex: Neural Search Infrastructure for the Scholarly Literature

Shane Ding, Edwin Zhang, and **Jimmy Lin**
David R. Cheriton School of Computer Science
University of Waterloo

Abstract

Cydex is a platform that provides neural search infrastructure for domain-specific scholarly literature. The platform represents an abstraction of Covidex, our recently developed full-stack open-source search engine for the COVID-19 Open Research Dataset (CORD-19) from AI2. While Covidex takes advantage of the latest best practices for keyword search using the popular Lucene search library as well as state-of-the-art neural ranking models using T5, parts of the system were hard coded to only work with CORD-19. This paper describes our efforts to generalize Covidex into Cydex, which can be applied to scholarly literature in different domains. By decoupling corpus-specific configurations from the frontend implementation, we are able to demonstrate the generality of Cydex on two very different corpora: the ACL Anthology and a collection of hydrology abstracts. Our platform is entirely open source and available at cydex.ai.

1 Introduction

The ongoing worldwide COVID-19 pandemic has brought about a resurgence of interest in natural language analysis applied to the scientific, particularly biomedical, literature. This has been catalyzed by the availability of corpora, such as the COVID-19 Open Research Dataset (CORD-19) curated by the Allen Institute for AI (Wang et al., 2020), as well as substantial efforts in the creation of evaluation resources, such as the TREC-COVID challenge (Voorhees et al., 2020; Roberts et al., 2020), which has constructed a test collection on information needs related to COVID-19.

Our project builds on Covidex, a full-stack open-source neural search engine for CORD-19 that includes three main capabilities: basic keyword search, neural ranking models, and a faceted search and browsing interface. An early description of our project can be found in Zhang et al. (2020a) and an updated paper appears in this workshop (Zhang et al., 2020b). Despite the successes of Covidex, parts of the system remain hard-coded to work only with CORD-19. We saw an opportunity to develop our code base into general neural search infrastructure that can be deployed on different corpora of scholarly articles.

The contribution of this work is the execution of this vision: We introduce Cydex, which abstracts Covidex to provide neural search capabilities to scholarly literature in different domains. Throughout this paper, we use the terms "domain" and "corpus" interchangeably, as the most precise way to define a particular domain is by a collection of texts that capture the domain. Our platform is demonstrated on the ACL Anthology and a collection of abstracts in the hydrology domain. All of the components described in this paper are open source.

2 From Covidex to Cydex

This section begins with a description of Covidex, and then describes how we have adapted and generalized each layer in its stack to create Cydex.

2.1 The Covidex Stack

Covidex, which has been available online for searching CORD-19 since late-March 2020, runs a stack comprised of three layers, all of which are open source:

Anserini/Pyserini. Anserini[1] is an information retrieval toolkit (Yang et al., 2018) built on the popular open-source Lucene search library, which is widely deployed in industry to power production search applications. As Lucene is implemented in Java, our tools are designed to run on the Java Virtual Machine (JVM). However, Python is the main language for PyTorch (Paszke et al., 2019) and

[1] http://anserini.io/

Proceedings of the First Workshop on Scholarly Document Processing, pages 168–173

https://doi.org/10.18653/v1/P17

TensorFlow (Abadi et al., 2016), the two most popular neural network toolkits today, and more broadly, Python has emerged as the language of choice for applied machine learning today in part due to its diverse and mature ecosystem. Pyserini (Yilmaz et al., 2020)[2] bridges the gap between the JVM and Python by providing a Python interface to Anserini. Together, Anserini and Pyserini provide basic keyword search capabilities to arbitrary corpora, which include tools to fetch raw document texts as well as utilities to access various term statistics.

PyGaggle. As part of Covidex, Zhang et al. (2020b) built PyGaggle,[3] a Python library for neural text ranking designed to work with Pyserini. Although the application of BERT (Devlin et al., 2019) to text ranking is well known (Nogueira and Cho, 2019), PyGaggle was designed to showcase models that adopt a novel sequence-to-sequence formulation for ranking (Nogueira et al., 2020b), specifically using T5 (Raffel et al., 2020). Deployed as a relevance classifier that reranks BM25 results from Pyserini, the model is fed a query q and each candidate document d in turn. The model is fine-tuned to produce either "true" or "false" depending on whether the document is relevant or not to the query. At inference time, a softmax is applied to the logits of the "true" and "false" tokens, and the resulting probability of the "true" token is used as the relevance score of d. Candidate documents are then reranked using their relevance scores.

Given the lack of COVID-19 training data when the model was initially developed, the T5 ranker was fine-tuned on the popular MS MARCO passage dataset (Bajaj et al., 2018) and directly applied to CORD-19 content. In other words, the ranker was deployed in a zero-shot setting. Additionally, PyGaggle implements an unsupervised sentence highlighting technique using BioBERT (Lee et al., 2020), as described in Zhang et al. (2020a). The key takeaway here is that the current implementation of PyGaggle is completely domain agnostic.

Covidex. To be precise, Covidex is a faceted search and browsing interface built on top of PyGaggle and Anserini/Pyserini, but in this paper, we use it to refer to the entire search engine for convenience. The Covidex layer is comprised of two major components:

The first is a REST API that exposes the keyword search capabilities implemented in Pyserini and the neural ranking capabilities implemented in PyGaggle. This is accomplished by wrapping both into a single API endpoint. Using this endpoint, clients can directly submit search requests containing their queries. The API service is built and deployed using the FastAPI Python web framework, selected for speed and ease of use.[4]

The second of these components is the search and browsing interface itself, which is built with the React JavaScript library[5] to support the use of modular, declarative components and to take advantage of its vast ecosystem. The interface implements a search bar that calls the API service (described above) and renders the results, also providing support for faceted browsing. This feature includes several filters, such as a slider for numeric ranges and a multi-select filter for fields such as the author and publication venue.

2.2 Abstractions for Cydex

At the outset, our goal for Cydex was to minimize the effort required for developers to set up their own Covidex instance on custom corpora.

At the bottom layer, since Anserini/Pyserini was already designed as a general-purpose search toolkit, the only major change required there was the addition of corpus-specific ingesters. For Cydex, after considering a number of options, we settled on an implementation of an ingester that can parse collections of bibtex records to generate Lucene indexes. As it is common to augment bibtex records with abstracts (as is the case with the ACL Anthology), this provides a general-purpose solution that encompasses many scholarly corpora. We specifically decided not to support indexing full text at present for two reasons: (1) due to copyright restrictions, full-text articles are not commonly available, and (2) experimental results from TREC-COVID suggest that abstracts alone achieve reasonable search effectiveness.

The next layer up in the stack, PyGaggle, required no changes at all, since the current neural ranking models are deployed in a zero-shot manner and thus contain no domain-specific knowledge. Of course, there are active research efforts in domain adaptation, both for search tasks (MacAvaney et al., 2020) as well as general NLP tasks (Gururangan et al., 2020), but these efforts are beyond the scope of this paper.

[2] http://pyserini.io/
[3] http://pygaggle.ai/
[4] https://fastapi.tiangolo.com/
[5] https://reactjs.org/

Most of our effort was spent in the top layer of the stack, on the Covidex search and browsing interface itself. In particular, we had to refactor hard-coded values in Covidex into a general configuration framework. To provide a generalized implementation, we abstracted dataset-specific details from all components. Customization is accomplished through two developer-supplied schemas, one ingested by the API service and the other by the interface itself.

The schema for the API service specifies the fields that the system should retrieve from the underlying Lucene index, along with the data type of the field, the default value of the field, and whether the field contains multiple values or not. In addition, since Covidex supports multiple "search verticals" (for example, representing two closely related sub-corpora), the schema also contains a `search_vertical` key, which indicates to Cydex exactly what should be searched. The example below illustrates the API service schema for the ACL Anthology:

```
{
    "document_fields": {
        "abstract_html": {
            "type": "str",
            "default": "None",
            "field_size": "single"
        },
        ...
    },
    "search_vertical": {
        "ACL": "ACL Anthology"
    }
}
```

To enable customization of the actual search and browsing interface, another schema is used to declare the facets, indicating which fields should be filtered and the type of filter that should be applied. The example below is how one such filtering setup may look like for the ACL Anthology:

```
{
    "publish_time": {
        "type": "slider",
        "displayText": "Publish Time"
    },
    "authors": {
        "type": "selection",
        "displayText": "Authors"
    },
    ...
}
```

By implementing the abstraction with schemas, we allow for the dynamic generation of both the service API and user interface components via configurations specified by the developer.

Cydex optionally allows the developer to implement a presentation component to display custom search result layouts. This allows developers to customize how they want to display information about the search results, for example, to specify fonts, font sizes, and the format of the citation. This design allows the developer to focus on rendering, while leaving the configuration and implementation details to Cydex itself.

3 Case Studies

To demonstrate the effectiveness of abstractions within Cydex, we have built and deployed custom instances on two corpora:

- The ACL Anthology, which should already be familiar to readers. The version we used has 57K articles and contains details such as the publication venue and special interest groups (SIGs). We directly index the bibtex files generated as part of the anthology; these records include abstracts, to the extent that they are available.

- A corpus of hydrology abstracts, originally compiled for topic modeling analysis by Rahman et al. (2020). The corpus contains 42K articles from six peer-reviewed hydrology journals such as Hydrology and Earth System Sciences, the Journal of Hydrometeorology, and Water Resources Research. The corpus comprises bibtex records that have been augmented with the abstract texts.

Although Cydex has been verified to work with both corpora, here we focus on the ACL Anthology. A screenshot is shown in Figure 1.

4 Evaluation and Discussion

The focus of our efforts, and the contribution of this work, is search infrastructure to support information access to scholarly literature. However, it is certainly fair to inquire about the quality of the search results produced by Covidex (and Cydex by extension, since there have been no changes to the T5 ranking models in PyGaggle).

We have yet to conduct formal domain-specific evaluations, either in the ACL or hydrology contexts, and can only point to Covidex results based on participation in the multi-round TREC-COVID challenge (Voorhees et al., 2020; Roberts et al., 2020). As detailed in Zhang et al. (2020b), the Covidex team submitted the best automatic runs

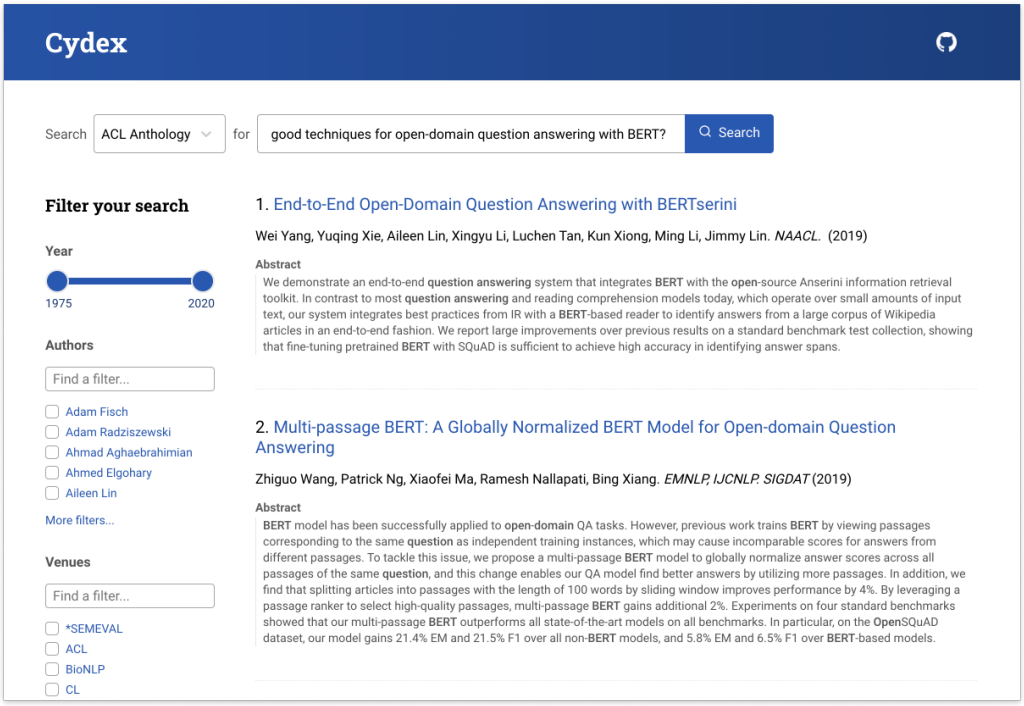

Figure 1: The Cydex search interface on top of the ACL Anthology. On the left, we display all configured facets based on the provided schema, while the right side displays search results using a customizable layout component.

in the final two rounds using a combination of techniques that included ranking with T5. Note that the Covidex ranking model operated in a zero-shot setting, and there is independent evidence that transformer-based ranking models have powerful cross-domain relevance transfer capabilities (Yilmaz et al., 2019; MacAvaney et al., 2020). Thus, we hope that the ranking models would generalize to the NLP and hydrology domains as well.

Despite the lack of domain-specific evaluation data, we performed our own informal evaluation of Cydex on the ACL Anthology. Our evaluation can be characterized as informal "hallway usability testing", primarily as a sanity check. We asked four colleagues (who were not the co-authors) to compare the top five results of five different natural language questions between Cydex and the ACL Anthology's current site-specific Google search. All of our colleagues are familiar with the NLP literature. We came up with the test set of five natural language questions based on our own interests.

Using a standard evaluation methodology, the human assessors were presented side-by-side results (from the two systems) and asked which one (the left system or the right system) they liked better. The identity of the two systems were blinded and randomized, so the assessors had no way to determine the source of the results. All four assessors preferred the results of Cydex for at least three out of the five test questions. Needless to say, there are not enough assessors or questions in this simple evaluation to draw any meaningful conclusions, although these results suggest that our system does not appear to be obviously worse than Google. We consider this to be an encouraging outcome, given that Google presents a high bar in terms of search quality, and we have only begun to build Cydex.

More broadly, however, there are obvious questions worth addressing about the premise of our endeavor: Why do we even need Cydex? Why do we need another search engine to the scholarly literature? Isn't Google Scholar sufficient?

While undoubtedly valuable and certainly the most widely used search engine for the scholarly literature, it would be far-fetched to think that any research project can displace Google Scholar. However, a failure by academic researchers to also engage in the space would be implicitly ceding this important intellectual territory to a commercial search engine that is, at its core, not transparent and controlled by a single entity that may not necessarily act in the best interest of scholars.

Our project does not aspire to be a comprehensive guide to the literature like Google Scholar or AI2's Semantic Scholar. Rather, our niche is domain-specific "verticals" that are manageable

from the scale perspective, yet sufficiently large to support interesting analyses. Our two illustrative case studies of the ACL Anthology and hydrology abstracts fit this bill exactly: our software stack can run with only modest resources, and both these corpora are sufficiently rich and self-contained to answer interesting questions—see, for example, recent analyses of the ACL Anthology by Mohammad (2020) and the work by Rahman et al. (2020) on the collection of hydrology abstracts.

5 Ongoing Work and Conclusions

There are two main directions we are currently pursuing as part of ongoing work on Cydex. The first is expansion of the platform to new domains, in particular examining the scalability of our underlying infrastructure. Both the ACL Anthology and the corpus of hydrology abstracts are tiny by modern standards; even CORD-19, with around 300K articles (as of October 2020), is small compared to many information retrieval test collections. A worthwhile target would be the recently released Kaggle arXiv dataset,[6] which contains over 1.7M preprints, in areas ranging from physics to the many subdisciplines of computer science.

The effectiveness of our ranking algorithms is naturally another area of interest, although the development of neural ranking models is orthogonal to the infrastructure that we present here. Our group's latest work on a sequence-to-sequence ranking formulation using T5 is discussed in Nogueira et al. (2020b). While we seek to further improve these core models, in the context of Cydex we are more interested in the end-to-end user experience, where ranking is just one (albeit important) aspect. Beyond features such as faceted browsing and the highlighting of relevant content (already implemented), the platform could benefit from the integration of additional capabilities such as citation recommendation (Bhagavatula et al., 2018; Nogueira et al., 2020a) and related article browsing (Smucker and Allan, 2006; Lin and Wilbur, 2007). These features all contribute to users' perception of system quality and must be evaluated holistically, for example, via user studies.

Cydex represents the starting point of a platform for building information access capabilities for domain-specific scholarly literature, powering our own explorations into scientific literature analysis. We hope that our open-source approach provides infrastructure that may be useful for other researchers as well.

[6] https://www.kaggle.com/Cornell-University/arxiv

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, pages 265–283.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3*.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 238–251.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jimmy Lin and W. John Wilbur. 2007. PubMed related articles: A probabilistic topic-based model for content similarity. *BMC Bioinformatics*, 8:423.

Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2020. SLEDGE: A simple yet effective baseline for coronavirus scientific knowledge search. *arXiv:2005.02365*.

Saif M. Mohammad. 2020. NLP Scholar: An interactive visual explorer for natural language processing literature. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 232–255.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Kyunghyun Cho, and Jimmy Lin. 2020a. Navigation-based candidate expansion and pretrained language models for citation recommendation. *Scientometrics*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020b. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of EMNLP*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Mashrekur Rahman, Jonathan M. Frame, Jimmy Lin, and Grey Nearing. 2020. Hidden stories: Topic modeling in hydrology literature. *EarthArXiv*.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R. Hersh. 2020. TREC-COVID: rationale and structure of an information retrieval shared task for COVID-19. *Journal of the American Medical Informatics Association*.

Mark D. Smucker and James Allan. 2006. FindSimilar: Similarity browsing as a search tool. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 461–468.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2020. TREC-COVID: Constructing a pandemic information retrieval test collection. *SIGIR Forum*, 54(1):1–12.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 Open Research Dataset. *arXiv:2004.10706*.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: reproducible ranking baselines using Lucene. *Journal of Data and Information Quality*, 10(4):Article 16.

Zeynep Akkalyoncu Yilmaz, Charles L. A. Clarke, and Jimmy Lin. 2020. A lightweight environment for learning experimental IR research practices. In *Proceedings of the 43rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*, pages 2113–2116.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3481–3487.

Edwin Zhang, Nikhil Gupta, Rodrigo Nogueira, Kyunghyun Cho, and Jimmy Lin. 2020a. Rapidly deploying a neural search engine for the COVID-19 Open Research Dataset: Preliminary thoughts and lessons learned. *arXiv:2004.05125*.

Edwin Zhang, Nikhil Gupta, Raphael Tang, Xiao Han, Ronak Pradeep, Kuang Lu, Yue Zhang, Rodrigo Nogueira, Kyunghyun Cho, Hui Fang, and Jimmy Lin. 2020b. Covidex: Neural ranking models and keyword search infrastructure for the COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on Scholarly Document Processing (SDP 2020)*.