FigExplorer: A System for Retrieval and Exploration of Figures from Collections of Research Articles

Saar Kuzi
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA
skuzi2@illinois.edu

ChengXiang Zhai
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA
czhai@illinois.edu

Yin Tian
CRRC Academy
Beijing, China
ty@crrc.tech

Haichuan Tang
CRRC Academy
Beijing, China
thc@crrc.tech

ABSTRACT

In this paper, we present `FigExplorer`, a novel general system that supports the retrieval and exploration of research article figures. Specifically, `FigExplorer` can support 1) figure retrieval using keyword queries, 2) exploration of related figures of a given figure, 3) exploration of a figure topic using the citation network, and 4) search result re-ranking using an example figure. The different functions were implemented using either classical IR models or neural network-based figure embeddings. Finally, the system was designed to facilitate the collection of user data for training and test purposes and it is flexible enough such that it can be extended to include new functions and algorithms. As an open-source system, `FigExplorer` can help advance the research, evaluation, and development of applications in this area.

ACM Reference format:
Saar Kuzi, ChengXiang Zhai, Yin Tian, and Haichuan Tang. 2020. FigExplorer: A System for Retrieval and Exploration of Figures from Collections of Research Articles. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, July 25–30, 2020 (SIGIR '20),* 4 pages.
https://doi.org/10.1145/3397271.3401400

1 INTRODUCTION

Figures in research articles are often very useful for both understanding the methods used in an article and digesting the experimental findings. Thus, it would be useful to have systems that can support the exploration of research article figures. Such systems can help researchers digest knowledge buried in the literature quickly, thus accelerating scientific discovery and technology innovation. In the most basic mode of exploration, users can retrieve figures using keyword queries. To this end, several systems for figure retrieval were developed [4, 6, 7]. Using retrieval only, however, may not be sufficient for the effective exploration of research figure collections. Furthermore, the development of figure mining and retrieval algorithms is currently impeded due to the lack of test collections and training data. To overcome this issue, simulated evaluation can be used [5] but its effectiveness is limited.

`FigExplorer` (http://figuresearch.web.illinois.edu) is a novel general system for retrieval and exploration of figure collections. First, `FigExplorer` allows the user to directly retrieve figures, extracted from research articles, using keyword queries. Then, using each retrieved figure as a seed, the user can further explore the collection and refine the search. Specifically, the user can view *related figures* from other articles or from the same article. The user can also explore the general topic of the figure by viewing *clusters of figures* constructed from the citation network of the figure. Finally, the user can select any figure to be used for *re-ranking* (refining) the result list. The different exploration functions are implemented using *figure embeddings* that were learned from the text data representing the figure with a neural network. To train the model, we use a weak supervision approach which leverages the citation connections between articles to generate training data. For the keyword retrieval part, the system includes the implementation of multiple ways to represent figures with text data, which the user can experiment with, and several ranking algorithms.

`FigExplorer` can also support the *annotation* of relevance judgments by allowing a user to make such judgments on the retrieved (related) figures using radio buttons. *Implicit feedback*, which can be used for model training and evaluation, can also be easily collected based on user actions in the system.

Finally, `FigExplorer` is an open-source toolkit which was designed to facilitate future research on the topic.[1] Specifically, the toolkit can be used by other researchers for 1) learning about the state-of-the-art figure retrieval methods and new functions supported by embeddings, 2) building test collections, 3) building their applications, and 4) developing and testing new models.

2 RELATED SYSTEMS

Several figure search engines were previously developed for the bio-medical domain [4, 6, 7, 10]. The BioText engine [4] enables the search of figures using keyword queries. In another system [10], figure search was also implemented with some basic exploration capabilities. The SLIF system [7] also performed figure retrieval but proposed a topic model approach to browse the result list. Finally,

[1] http://github.com/saarku/fig-explorer

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401400

the FigSearch retrieval system [6] was tailored for the use-case of gene-related figures. Compared to previous systems, `FigExplorer` is the first open-source general figure retrieval and exploration system. Specifically, the system includes several novel functions, which utilize embeddings, that can be used to perform exploration of the figure collection. Furthermore, the system is general enough to facilitate future research on the topic.

Other systems focused on the extraction, summarization and indexing of figures [1, 2, 8, 9]. In our system, the focus is on the search and exploration of figure collections, while we use existing tools for the extraction part. Thus, `FigExplorer` is complementary with these existing systems and they can be combined.

3 SYSTEM FUNCTIONS

`FigExplorer` was implemented as a Web application and its main page is presented in Figure 1. The most basic mode of exploration in the system is the retrieval of figures using a keyword query. The search process begins with the user typing a query in the text box and then selecting the search configuration (i.e., collection, model, and figure fields); in the case where the user does not specify the configuration, default settings are used. After issuing the query, the user is presented with a result list of 10 figures; a screen-shot of a part of the search result page is presented in Figure 2. The user can continue to view the next 10 figures (or go back) by clicking on the "Next" ("Previous") button at the end of the page. For each result figure, its caption is presented which also serves as a hyperlink to the figure's article. Below the caption, the user can indicate whether the figure is relevant to the query by clicking on the corresponding radio button. Next to the relevance radio buttons, there are buttons that can be used for performing further exploration of the collection. Finally, the user can view the image file of the figure (clicking on the image will enlarge it for a better view) and a short textual summary of the figure (snippet) which was automatically extracted from the article; the keywords of the query and the explicit mentions of the figure in the article are boldfaced in the snippet.

Further exploration of the collection can be performed using buttons that appear for each figure. (Some buttons may not appear for some figures due to the lack of data for them.) The output of each exploration function is presented below the figure and can be removed by either pressing the button again or by using the "Minimize" button. A screen-shot of a single figure, after two exploration functions were used, is presented in Figure 3. The exploration functions include: 1) "Paper Info": allowing the user to view information about the article containing a figure such as the title and the abstract (this function can help the user to put the figure in context easily). 2) "Same Paper Figures": displaying other figures in the same article. 3) "Related Figures": presenting the user figures that are semantically related to the target figure (using figure embeddings). 4) "Citation Clusters": enabling a broader exploration of the topic of the figure by clustering the citation network of the figure and presenting representative figures for each cluster (this function is also powered with embeddings). 5) "Re-rank using this figure": re-ranking the existing result list using the embedding-based representation of the specific figure; this option exists also for the related figures and the figures of the citation clusters.

`FigExplorer` also logs information about user actions. This information includes 1) clicks on a caption of either a result figure or a related figure, 2) clicks on the relevant/not-relevant button, and 3) events of issuing a query or of using any of the exploration functions. The logging of such information can be useful for creating a test collection for different tasks such as figure retrieval (relevance between a query and a figure) and figure relatedness prediction (relevance between two figures).

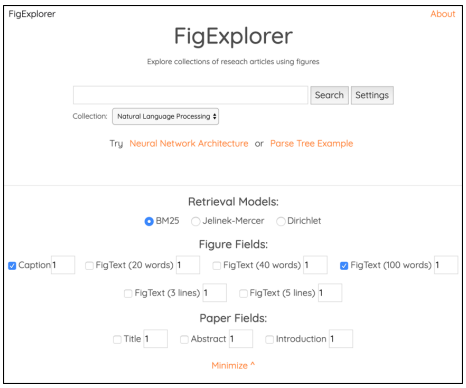

Figure 1: The main page of `FigExplorer`.

4 IMPLEMENTATION

The high-level architecture of `FigExplorer` is presented in Figure 4. `FigExplorer` is an open-source toolkit that was designed to be flexible enough for future extensions. Next, we describe the front-end and back-end of the system.

4.1 Front-end

`FigExplorer` is a Web application that is written in JavaScript using jQuery and Ajax. The front-end has three main functionalities. 1) User Interface: obtains the user input and sends it to the server. The input usually includes the query and the search settings in the case of retrieval and the figure ID in the case of an exploration function. 2) Search Results: receives the search results (or an exploration function output) from the server and presents them in the browser. 3) Logger: collects user interactions from the browser.

4.2 Back-end

The back-end of the system is composed of the following components. 1) A Back-end Server: written in Python using the Flask library (flask.pocoo.org). 2) A Search Engine Server: written in Java using the Lucene library (lucene.apache.org). 3) A Machine Learning (ML) Server: written in Python. 4) Figures Index: a Lucene-based inverted index. 5) Image Files repository. 6) Search Log.

The back-end server forms a bridge between the front-end and the search engine server. The main task of the back-end server is to handle search and exploration function requests from the client. In the case of a keyword search, the back-end server communicates with the search engine server to get the result list. The search engine server is also used to get information regarding the figure's article (abstract and title) and other figures in the same article. In

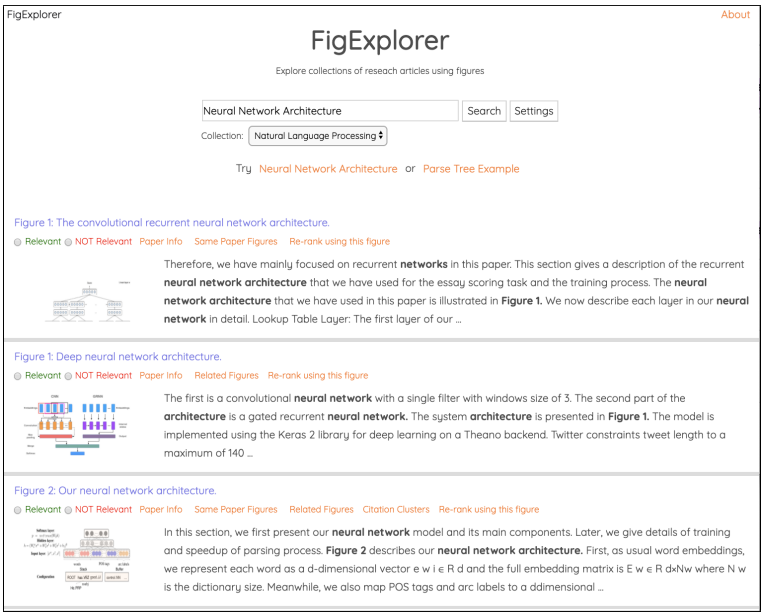

Figure 2: Search result page example.

Figure 3: Example of a single search result after two exploration functions were used.

the case of some of the exploration functions ("Related Figures" and "Citation Clusters"), the results are obtained from the ML server. Once the results are returned (from either server), the back-end server performs some post-processing of the content and sends it to the browser. Some basic computations are also performed in the back-end server including 1) bold-facing of the query terms in the figure snippet, 2) fetching the image files of the figures from the image files repository, and 3) updating the search log.

The main task of the search engine server is to perform retrieval using a keyword query. The search engine uses an index that stores the figures in the collection using textual information. Specifically, a figure is represented using multiple textual fields including its caption, text in the article that explicitly discusses the figure, and some of the article's text (the title, abstract, and introduction). The index also stores, for every figure, the image file directory which can be used to get an image from the repository. To score a figure with respect to a query, the linear interpolation of the scores of the query in the textual fields is computed; the weights in the interpolation can be controlled at the settings of the system (see Figure 1). The search engine server and the back-end server communicate using the py4j library (py4j.org).

The ML Server is responsible for learning the embedding-based representation of figures and utilizing it for the different exploration functions. Embedding-based representation of figures is learned using the caption of the figure as well as text in the article that directly describes the figure (both are concatenated). The textual information of a figure is fed into an LSTM network whose output serves as a vector representation for the figure. We use the binary

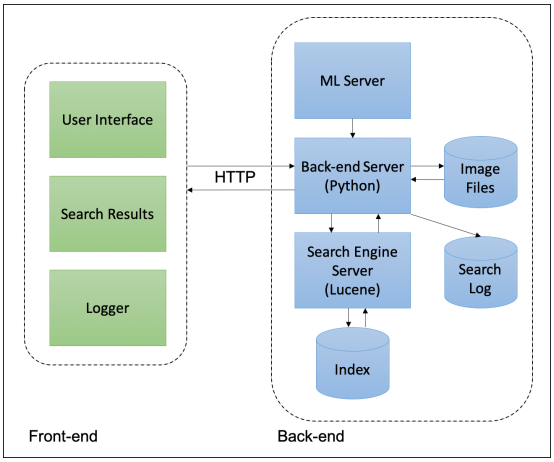

Figure 4: System Architecture.

Cross-Entropy loss function (i.e., the goal is to predict whether two figures are related or not). Weak supervision is used to build the training data by assuming that two figures in citing articles or in the same article are related; negative examples are obtained using random sampling. The model is learned by leveraging all figures in the collection using the TensorFlow library (tensorflow.org).

The outputs for the "Related Figures" and "Citation Clusters" functions are computed in an offline manner, using the embedding vectors, and are stored for fast serving. To find related figures, for each figure in the collection, a KNN search is performed to find the most similar figures using the cosine similarity. To perform the citation-based clustering for a figure, a citation network of a figure is constructed first. To do that, all figures in articles that have a citation relation with the article of the seed figure (a direct connection or a connection through a third article) are included. Then, K-means clustering is performed using the embedding vectors. Finally, a representative figure is selected for each cluster based on the distance to the cluster's mean. The "Re-rank using this figure" function is performed in an online manner. First, 100 figures are retrieved using the keyword query. Then, these figures are re-ranked based on their similarity with the seed figure in the embedding space.

5 DATA SETS

To build a collection of figures, we follow the next steps: 1) Obtaining a set of research articles. 2) Extracting the figures from the PDF files using the PdfFigures toolkit [3]. 3) Extracting the full text of the articles using the Grobid toolkit (github.com/kermitt2/grobid). 4) Processing the full text of the documents to extract the textual fields for a figure. 5) Indexing the figures using the textual fields. Currently, this is an offline pipeline that runs separately from the system. In future work, we plan to integrate this pipeline in the system to support automated figure crawling and indexing. Our demo currently supports two data sets. 1) Natural Language Processing: 73,409 figures that were extracted from 40,367 articles whose copyright belongs to ACL up to October 2018 (aclweb.org/anthology). 2) Mechanical Engineering: 9,712 figures that were extracted from 1,377 articles on bearing failures. This data set was created to explore the potential use of `FigExplorer` for supporting mechanical failure diagnosis where the analyst may conveniently retrieve figure plots showing typical vibration signal patterns for any hypothesized failure of a bearing (e.g., outer ring face) which can help finalize a diagnosis.

6 DEMONSTRATION SCENARIOS

We plan to demonstrate the following functions of `FigExplorer`:
1. Sample applications of figure search: We will use a set of queries to demonstrate at least the following applications: 1) finding figures illustrating technical approaches, 2) finding experimental results, and 3) finding illustrations of an example.
2. Exploratory search: We will create a set of topics and use the system to illustrate an exploratory search process. For example, a process of creating a literature review of a new topic using figures.
3. Exploration functions: We will showcase examples in which a keyword query is not a sufficient tool for exploration and exploration functions can be used to improve the process.
4. Comparison of different figure ranking algorithms: The system allows a user to easily configure the choices of the retrieval methods. We will vary the configurations to compare different ways to represent figures with text data and different ranking algorithms.
5. Collection of relevance judgments: We plan to collect users' queries and relevance judgments for building a test collection.

ACKNOWLEDGMENTS

This work was supported in part by a research grant from the CRRC Overseas Research Center at the University of Illinois at Urbana-Champaign. We are grateful to Jiafeng Guo for his help with preparation of the Mechanical Engineering collection.

REFERENCES

[1] Sumit Bhatia and Prasenjit Mitra. 2012. Summarizing figures, tables, and algorithms in scientific publications to augment search results. *ACM Transactions on Information Systems (TOIS)* 30, 1 (2012), 1–24.
[2] Sagnik Ray Choudhury, Suppawong Tuarob, Prasenjit Mitra, Lior Rokach, Andi Kirk, Silvia Szep, Donald Pellegrino, Sue Jones, and Clyde Lee Giles. 2013. A figure search engine architecture for a chemistry digital library. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. 369–370.
[3] Christopher Clark and Santosh Divvala. 2015. Looking Beyond Text: Extracting Figures, Tables, and Captions from Computer Science Papers. (2015).
[4] Marti A Hearst, Anna Divoli, Harendra Guturu, Alex Ksikes, Preslav Nakov, Michael A Wooldridge, and Jerry Ye. 2007. BioText Search Engine: beyond abstract search. *Bioinformatics* 23, 16 (2007), 2196–2197.
[5] Saar Kuzi and ChengXiang Zhai. 2019. Figure Retrieval from Collections of Research Articles. In *European Conference on Information Retrieval*. Springer, 696–710.
[6] Fang Liu, Tor-Kristian Jenssen, Vegard Nygaard, John Sack, and Eivind Hovig. 2004. FigSearch: a figure legend indexing and classification system. *Bioinformatics* 20, 16 (2004), 2880–2882.
[7] Abdul-Saboor Sheikh, Amr Ahmed, Andrew Arnold, Luis Pedro Coelho, Joshua Kangas, Eric P Xing, William Cohen, and Robert F Murphy. 2009. *Structured literature image finder: Open source software for extracting and disseminating information from text and figures in biomedical literature.* Technical Report. Carnegie Mellon University School of Computer Science, Pittsburgh, USA, CMU-CB-09-101.
[8] Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. FigureSeer: Parsing result-figures in research papers. In *European Conference on Computer Vision*. Springer, 664–680.
[9] Pradeep B Teregowda, Madian Khabsa, and Clyde L Giles. 2012. A system for indexing tables, algorithms and figures. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*. 343–344.
[10] Hong Yu, Feifan Liu, and Balaji Polepalli Ramesh. 2010. Automatic figure ranking and user interfacing for intelligent figure search. *PLoS One* 5, 10 (2010), e12983.