Searching for Physical Documents in Archival Repositories

Tokinori Suzuki
Kyushu University
Fukuoka, Japan
tokinori@inf.kyushu-u.ac.jp

Douglas W. Oard
University of Maryland
College Park, MD, USA
oard@umd.edu

Emi Ishita
Kyushu University
Fukuoka, Japan

Yoichi Tomiura
Kyushu University
Fukuoka, Japan

ABSTRACT

Early retrieval systems were used to search physical media (e.g., paper) using manually created metadata. Modern ranked retrieval techniques are far more capable, but they require that content be either born digital or digitized. For physical content, searching metadata remains the state of the art. This paper seeks to change that, using a textual-edge graph neural network to learn relations between items from available metadata and from any content that has been digitized. Results show that substantial improvement over the best prior method can be achieved.

CCS CONCEPTS

• **Information systems → Learning to rank**.

KEYWORDS

Physical information access, Archives, Graph neural network

ACM Reference Format:
Tokinori Suzuki, Douglas W. Oard, Emi Ishita, and Yoichi Tomiura. 2024. Searching for Physical Documents in Archival Repositories. In *Proceedings of 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Searching digital content is such a strong focus of IR research that the digital content assumption is rarely even stated. There are, however, still massive quantities of documents in physical form (e.g., on paper, microfilm, or analog audio or video tape) that are unlikely to be digitized any time soon. For example, only about 2.1% of the estimated 11.9 billion pages held by the US National Archives have been digitized, and at present rates (~121,000 pages/day) full digitization would take centuries [4]. Our focus is on searching that which is not yet digital. The usual approach to this is manually intensive: searchers must identify a repository with useful content, use metadata there to get an idea where to look, learn how those parts of the repository are organized, and look through many boxes. We propose to better automate the process by building graphs from known content and learned relationships, training graph embeddings to represent the content of archival storage containers ("boxes"), along with relations that can be discerned from existing metadata. When a user poses a query, we embed the query and then identify boxes with similar embeddings. The keys to this approach are how we build the graph and our use of Edgeformer, which learns mappings from text to predefined categories (in our case, boxes) that we associate with edges in the graph.

2 RELATED WORK

We describe archival metadata and digitization, summarize work showing sampling content is useful, and introduce Edgeformers.

2.1 Archival Arrangement and Description

The scale of the description task in archives makes it impractical to describe each item individually. Instead, archivists arrange materials in ways expected to facilitate future access and then describe that arrangement [9]. Because the people who created a collection did so in some "original order," archivists make arrangement more efficient by respecting that original order [6]. Hierarchically structured metadata is then used to help searchers navigate the collection. For example, US State Department records are organized by time periods corresponding to changes in the Department's filing system; search within a period follows the arrangement of records from that time. 1960's State Department records were organized by country, date, and a topic hierarchy (e.g., political affairs, economic affairs, or management).[1] Searchers can use this metadata to identify boxes that might contain something that they want to see, but then they must look in those boxes to see what's there.

In recent years, archives have digitized parts of their collections. For reasons of scale and cost, the process has been highly selective, often driven by public interest, or by interests of partners who pay for it. For our experiments we use an extensively digitized set of State Department records on Brazil, but there has been no similarly extensive digitization for records involving (for example) Paraguay.

2.2 The Value of a Few Examples

Our work is motivated by Oard [5], who showed that with a few scanned pages per box it was possible to guess which box might contain what a searcher wished to see. Oard's key result was that

[1]https://www.archives.gov/research/foreign-policy/state-dept/rg-59-central-files/1963-1973

Item	Number	Fraction
Searchable PDF	27,514	
Date metadata	27,511	99.9%
Entity metadata	26,562	96.5%

Table 1: OCR and item-level metadata, 117-box collection.

guesses between five and 10 times better than random selection could be achieved. He interpreted this as indicating that documents could be "known by the company they keep." While intriguing, there are limitations to that work we seek to address. First, we would expect learned embeddings to yield better results than lexical matching. Second, Oard ranked documents separately using metadata and OCR terms and then applied late fusion, but we would expect a single unified model using OCR text and metadata to be better than late fusion. Third, Oard's experiments were based on sampling the same number of scanned documents from every box, but in reality digitization is highly uneven. To address these concerns, we extend Oard's test collection to model uneven digitization,[2] and we recast the task as learning a graph embedding.

2.3 Edgeformer-E

Edgeformers [2] were designed for bipartite graphs in which people (one node type) comment on objects (the other node type) such as products or movies. Different people comment on the same object, and one person may comment on different objects. In such cases, the text must be associated with edges, not with nodes that represent people or objects. One task is to interpret the text associated with an edge to predict a category for that edge (e.g., the person likes or dislikes the object) in a way that leverages the graph. We model our application in an analogous way, with documents as one node type, boxes as the other, the edge text being the text of the document, and the category being the box in which the document will be found.

Several text-aware Graph-Neural Networks (GNNs) have been proposed. One approach is to cascade a pretrained language model and a GNN, which has been used to model nodes [1, 3, 11]. GraphFormers [10] are more closely coupled, with GNNs nested between Transformer layers so node encoding leverages both node text and signals from neighbor nodes. However, these models assume only nodes have associated text. Edgeformers, by contrast, model graphs where text is associated with edges, not nodes [2]. Like Graphformers, Edgeformers are built on Transformers [8]. Edge text is initially embedded using BERT, but in Edgeformer-E the final embeddings are also affected by the nodes to which that edge is connected.[3] Edgeformer-E injects network signals into Transformer encoding using virtual node tokens. Embeddings of virtual node tokens are concatenated to the output embeddings of the text tokens in each layer. After concatenation, the hidden state at each layer represents both the edge's associated text and the two nodes that it connects.

3 METHOD

Here we describe a new test collection and how we model that collection, train Edgeformer-E, and perform inference.

[2]Available at https://github.com/tokinori8/archive-box-search
[3]There is also Edgeformer-N, which predicts node types, but we focus on Edgeformer-E.

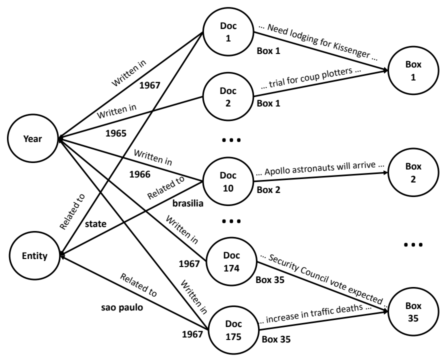

Figure 1: Graph design for the Edgeformer experiments.

3.1 A Larger Test Collection

To support richer experiments, we created a superset of Oard's collection. Following his approach [5], we collected PDF scans and item-level metadata for 27,514 State Department records from the U.S. National Archives and Records Administration (NARA) that had been digitized by the Brown University Library.[4] This includes all documents for which the source was NARA, item-level metadata indicated which box the document was stored in at NARA, and a sufficient number of PDFs from that box were available to permit sampling for training. Our documents together span 117 boxes.

Table 1 shows item-level metadata statistics for the collection. Date metadata indicates the creation date of a document, Entity metadata names people or organizations involved with the document's creation or receipt. As can be seen, some metadata fields were missing for some documents. We extracted the OCR text from the each PDF using python's PyPDF2 library.[5] The number of documents per box varies between 22 and 540, averaging 232.

Our test collection is fully digitized, but we simulate realistic amounts of digitization. In uniform sampling, we randomly select 5 documents per box on which to train, a uniform 2.1% sample, modeling the fact that 2.1% of NARA's pages are digitized. In our second approach, uneven sampling (Section 4.2), we select more documents from some boxes than others, since the amount of digitization can vary considerably from one part of a collection to another.

3.2 Modeling the Collection as a Graph

We model a collection as a textual-edge graph [2], which is a set of nodes, a set of edges, a set of texts assigned to the edges, and a set of categories assigned on the same edges. Our initial model is a bipartite graph, with nodes for each Document and for each Box, with each Document node linked to the node for the Box containing that document. Each edge is labeled using the OCR text for the first page of the corresponding document, truncated to 64 tokens, and the category assigned to an edge was the box in which that document was found. The edges between Document and Box nodes in Figure 1 illustrate this structure, although we note that Edgeformer is blind to node labels, which are shown here only for clarity. We also extend this to a tripartite graph by adding Year and Entity nodes, as shown in Figure 1. Unlike Box nodes, which

[4]https://library.brown.edu/create/openingthearchives/en/
[5]https://pypi.org/project/PyPDF2/

	Edgeformer-E			BM-25		
Graph Model	S@1	S@2	MRR	S@1	S@2	MRR
OCR	13.7 ± 2.7	15.0 ± 3.5	21.8	16.3 ± 0.4	24.5 ± 0.4	25.5
OCR+Year	38.4 ± 2.3	39.4 ± 2.3	42.4			
OCR+Entity	32.3 ± 1.4	35.6 ± 1.4	38.2			
OCR+Year+Entity	**40.8** ± 0.8	**45.7** ± 0.8	50.8			

Table 2: Title metadata query results (percent with standard error), 35-box collection, 5 first-page OCR samples per box.

have one node per box, we created a single metadata node per type. We set the text on each edge from a document to the Year node to "Written in" and the category for that edge to that document's year (extracted from Date metadata). A document has only one date, but it can have metadata specifying many entities (e.g., sender and recipient). Although not shown in Figure 1, we created one edge joining a Document node to the Entity node for each entity in the metadata, setting the edge text in each case to "Related to" and the category to a normalized version of the entity name (initials and last name).[6] Because Edgeformer is blind to node types, we foster consistency by placing Document nodes first when defining edges.

3.3 Graph Learning and Inference

We train Edgeformer-E for edge classification, with inference given two nodes and the edge text to predict the edge category from learned embeddings. The number of node pairs for training depends on the sampled documents and the graph model (Section 4). We used 90% of the sample for training, 10% for validation. Given a query, we then compute its text embedding, use that embedding to perform an approximate nearest neighbor search over known edge category embeddings, and finally a softmax layer ranks edge categories. For models with more than Box categories, we filter out categories not corresponding to Box nodes. We configured Edgeformer-E as in [2], with 12-layer BERT, 768 dimensions,[7] and 64-dimension bottleneck features for node embeddings. However, we changed the size of last softmax layer to the number of edge categories. We used AdamW with learning rate 0.001, early stopping patience 3 epochs, batch size 25, and macro F_1 as the optimization target.

4 RESULTS

Like Oard [5], we evaluate with known item retrieval, first randomly selecting a target box, then a target document (not one used for training). We use Brown's target document title as the query, and we report Success at rank 1 (S@1) and in the top two (S@2), and Mean Reciprocal Rank (MRR). For Edgeformer-E (and per-query BM-25 comparisons) we resample and retrain for each query. For other BM-25 runs we follow Oard and draw 100 queries per sample.[8]

4.1 Uniform Sampling

For comparison with Oard we first use uniform sampling, drawing 5 random first-page samples per box (i.e., the first page of each of 5 documents). Tables 2 and 3 summarize results on Oard's 35-box and our 117-box collection, respectively. Table 2 shows that with title metadata queries Edgeformer-E's overall S@1 is slightly worse

	Edgeformer-E			BM-25		
Graph Model	S@1	S@2	MRR	S@1	S@2	MRR
OCR	5.3 ± 0.9	7.0 ± 0.6	10.3	17.4 ± 0.4	24.5 ± 0.4	24.4
OCR+Year	**23.3** ± 2.8	**24.3** ± 2.8	26.6			
OCR+Entity	18.0 ± 1.0	19.3 ± 0.7	20.5			
OCR+Year+Entity	21.7 ± 1.8	**24.3** ± 0.9	26.2			

Table 3: Title metadata query results (percent with standard error), 117-box collection, 5 first-page OCR samples per box.

in the OCR-only condition than Oard's BM-25 ranking (which also was OCR-only). Manually inspecting ranked lists from 300 queries, we see that BM-25 did better than Edgeformer-E on the 48 queries that requested a "summary" or a "weeka" (a type of summary). On the other 252 queries, Edgeformer-E (38 at rank 1) does better than BM-25 (25 at rank 1), particularly on shorter queries (median query words for correct box at rank 1 was 3 for Edgeformer, 6 for BM-25).

Adding Year or Entity edges to the graph for Edgeformer-E yields very substantial improvements in S@1, and adding both does slightly better by S@1 then the better of the two (Year alone). To see why, we computed per-box MRR by averaging across queries that target a specific box, finding that relatively high MRR (above 0.4) was common for Edgeformer-E OCR+Year (20 of 35 boxes), whereas that was rare for Edgeformer-E OCR (4 boxes) or BM-25 OCR (7 boxes). Looking at the topic codes assigned to folders in those boxes, we see that for 6 of the 7 boxes on which BM-25 OCR achieved an MRR above 0.4, every folder in the box was coded as "General" (e.g., general policy trends, or general political analysis), whereas 14 of the 20 boxes on which Edgeformer-E OCR+Year did that well included at least some folders focused on specific topics (e.g., political parties, local government, or elections). From this we conclude that the supervised learning in Edgeformer-E is able to handle cases involving more specific topics better than can be done by BM-25's unsupervised approach. BM-25 can, however, find boxes in which few-shot learning isn't as representative of a box's full range of content: for 6 of the 7 "General" boxes where BM-25 did well, its MRR exceeded that of Edgeformer-E OCR+Year.

In Table 3, we see that moving from 35 to 117 boxes has little effect on BM-25, but Edgeformer-E results are adversely affected by every measure. This results from a tuning issue with Edgeformer-E training, which stops within 4 epochs on 173 of 300 queries on the 117-box collection, whereas such early stopping is rare on the 35-box collection. When it happens, this results in overfitting to small samples. Adding links to a metadata node (Year or Entity) does help, again yielding improvements over Edgeformer-E OCR by every measure. Moreover, Figure 3 shows we can do even better with only some of the Document to Year edges. S@1 peaks when training with

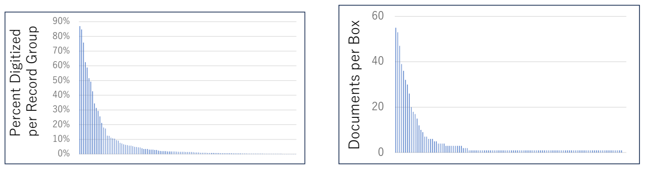

Figure 2: Digitized fraction by Record Group (left) and Uneven sampling distribution for our 117-box collection (right).

[6]We used the same normalization process for people and organizations.
[7]https://huggingface.co/bert-base-uncased
[8]We used Oard's BM-25 code from https://github.com/oard/BoxFinder.

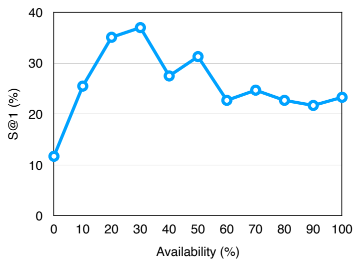

Figure 3: Random Doc-Year link ablation, 117-box collection.

20%–30% of those links, indicating that adding some edges helps learn better embeddings, but that too many edges with the same text (here, "Written in") risks overwhelming the Document to Box edge text from which we are seeing to learn to rank. A limitation of Edgeformer-E is that it has no concept of node type, so it can't learn to give edges between different node types different weights. Here we modulate an edge type's effect with ablation when training, but we will want more direct ways of controlling relative influence.

This effect may also partially explain why adding the sparser Year metadata yields more improvement than does adding the denser Entity metadata. Almost every document has one year, but documents on average have 2.5 links to Entities. Edgeformer-E OCR+Entity, with no random ablation, thus adds even more edges, potentially benefiting from additional relations but at the cost of further deemphasizing the document text that we seek to map to box categories.

To further investigate this effect, we created nested 60, 50 and 40-box collections, all supersets of the 35-box collection. We see consistent decline in S@1 with increasing collection size for Edgeformer-E OCR and OCR+Year, but no decline is evident for BM-25.

Finally, we note that in both tables S@2 is much better than S@1 for BM-25, but Edgeformer-E sees no benefit beyond random selection (~3% for 35-box, ~1% for 117). From this we conclude that our softmax ranking adds little beyond binary classification.

4.2 Uneven Sampling

Our experiments in Section 4.1 were based on uniform sampling, but that is often not the case in real collections. In this section we characterize the distribution of nodes and edges in an actual archive and then use that distribution to sample elements of our collection. Of the 526 NARA Record Groups with estimates for digitized pages,[9] 47 have 2% or more of their content digitized. The left side of Figure 2 shows the digitized fraction for the 117 most digitized NARA record groups (which cover 99.7% of all digitized documents). The right side shows a 117-box distribution of documents per box that approximates that observed distribution, with a mean of five documents per box and a minimum of one document per box.

Known-item queries were again selected with uniform random sampling of box then target document, as in all our experiments. OCR columns in Table 4 show results for Edgeformer-E and BM-25 on uneven samples of our 117-box collection. Edgeformer-E does better with uneven than uniform sampling, whereas BM-25 does worse. For Edgeformer-E, the number of samples for cases where the correct box is at rank 1 is far larger (averaging 4.7 per box) than when the correct box is lower in the ranked list (1.1 per box). As the uniform sampling results in Table 4 show, Edgeformer-E does much better with 10 samples per box than with 5, and uneven sampling produces some boxes with many more than 5 training samples.

[9]No estimates are available for 62 record groups.

	Samples	BM-25	Edgeformer-E	
Sampling	per Box	OCR	OCR	GPT-4
Uneven	mean=5	12.7 ± 0.3	38.0 ± 0.6	**43.3** ± 2.3
Uniform	5	**17.4** ± 0.4	5.3 ± 0.9	4.0 ± 1.2
Uniform	10	21.2 ± 0.4	**49.3** ± 3.7	46.0 ± 3.5

Table 4: S@1, title queries, 117-box collection.

4.3 OCR Rewriting

OCR text poses two difficulties for Edgeformer-E. First, BERT's WordPiece tokenizer is not optimized for text with OCR errors. Second, some documents (e.g., telegrams) start with formatted headers that are not specific to the document's topic. We might address both issues using abstractive summarization. We therefore used GPT-4 to summarize the OCR text for each document and then substituted that text for the OCR in our experiments (without Year or Entity metadata). We told GPT-4 that the text may contain errors, as recommended by OpenAI.[10] Our prompt is: "Please summarize the text below within 60 words. The text is the first page of an archival document scanned by OCR. That may contain some errors. Text: [OCR text]". Inspection of GPT-4 results indicates that the summaries focus well on the document's topic, that their length is typically 80 to 100 words, and that OCR errors have little effect on the summaries. Table 4 shows results for using GPT-4 summaries in place of OCR text for both training and test. Results are similar for uniform samples, but GPT-4 summaries do seem to improve uneven sampling results a bit; the improvement results from 17 additional boxes at rank 1 that each had only a single training sample.

5 CONCLUSION AND FUTURE WORK

We have introduced a new test collection to better characterize searching for physical documents, including uniform and uneven sampling and a broader range of metadata. We performed the first experiments for this task using a graph neural network, obtaining strikingly better results than prior work when a sufficient number of training samples are available. In future work we plan to explore a broader range of text-aware graph models, including models such as Graphformer in which text can be associated with nodes. We also plan to expand the range of graph structures we explore, and to experiment with additional metadata that is available for our new collection. Suzuki et al. [7] have shown that it is also possible to find descriptions of archival content in scholarly literature, and we expect the richer training data obtained in that way could lead to further improvements. Demonstrating that would require a different approach to test collection development, however.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant JP23KK0005.

[10]https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results

REFERENCES

[1] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical Graph Network for Multi-hop Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 8823–8838. https://doi.org/10.18653/v1/2020.emnlp-main.710

[2] Bowen Jin, Yu Zhang, Yu Meng, and Jiawei Han. 2022. Edgeformers: Graph-Empowered Transformers for Representation Learning on Textual-Edge Networks. In *The Eleventh International Conference on Learning Representations*.

[3] Chaozhuo Li, Bochen Pang, Yuming Liu, Hao Sun, Zheng Liu, Xing Xie, Tianqi Yang, Yanling Cui, Liangjie Zhang, and Qi Zhang. 2021. AdsGNN: Behavior-Graph Augmented Relevance Modeling in Sponsored Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 223–232. https://doi.org/10.1145/3404835.3462926

[4] National Archives and Records Administration. 2024. Record Group Explorer Data. Website https://www.archives.gov/findingaid/stat/discovery, visited January 11, 2024.

[5] Douglas W. Oard. 2023. Known by the Company It Keeps: Proximity-Based Indexing for Physical Content in Archival Repositories. In *Linking Theory and Practice of Digital Libraries: 27th International Conference on Theory and Practice of Digital Libraries, TPDL 2023, Zadar, Croatia, September 26-29, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 14241)*. Springer, 17–30. https://doi.org/10.1007/978-3-031-43849-3_3

[6] T. R. Schellenberg. 1961. Archival Principles of Arrangement. *The American Archivist* 24, 1 (1961), 11–24.

[7] Tokinori Suzuki, Douglas W. Oard, Emi Ishita, and Yoichi Tomiura. 2023. Automatically Detecting References from the Scholarly Literature to Records in Archives. In *Leveraging Generative Intelligence in Digital Libraries: Towards Human-Machine Collaboration - 25th International Conference on Asia-Pacific Digital Libraries, ICADL 2023, Taipei, Taiwan, December 4-7, 2023, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14458)*, Dion Hoe-Lian Goh, Shu-Jiun Chen, and Suppawong Tuarob (Eds.). Springer, 100–107. https://doi.org/10.1007/978-981-99-8088-8_9

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[9] Gregory Wiedeman. 2019. The Historical Hazards of Finding Aids. *The American Archivist* 82, 2 (2019), 381–420.

[10] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. 2021. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 28798–28810. https://proceedings.neurips.cc/paper_files/paper/2021/file/f18a6d1cde4b205199de8729a6637b42-Paper.pdf

[11] Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. 2021. TextGNN: Improving Text Encoder via Graph Neural Network in Sponsored Search. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2848–2857. https://doi.org/10.1145/3442381.3449842