NTCIR-18: Proceedings of the 18th NTCIR Conference on Evaluation ofInformation Access Technologies, June 10-13, 2025, Tokyo, Japan
DOI: https://doi.org/10.20736/0002002089

KASYS at the NTCIR-18 SUSHI Task

Haruki Fujimaki
University of Tsukuba
Japan
s2313638@u.tsukuba.ac.jp

Makoto P. Kato*
University of Tsukuba
Japan
mpkato@acm.org

Abstract

This paper describes the KASYS team's participation in the NTCIR-18 SUSHI Task by presenting a multi-level metadata aggregation and retrieval approach for Subtask A, which focuses on retrieving undigitized historical materials with sparse item-level metadata. Our system leverages the hierarchical organization of the data—comprising Box, Folder, and Item levels—by aggregating metadata from lower to higher levels and applying two search strategies ("Merge" and "Each"). We evaluate traditional BM25 alongside dense retrieval models (E5 and ColBERT) without fine-tuning, and hyperparameter optimization using Optuna is employed to determine the optimal weight for each level. Although our multi-level score aggregation strategy was designed to exploit the hierarchical structure of the data, it did not yield a significant performance improvement over a simpler BM25 baseline. Future work will explore improved preprocessing of noisy metadata, hybrid retrieval methods combining BM25 with dense re-ranking, and model fine-tuning to further enhance performance in searching undigitized archival collections.

Keywords

hierarchical metadata aggregation, multi-level retrieval, undigitized archive retrieval

Team Name

KASYS

Subtasks

Subtask A

1 Introduction

The KASYS team participated in subtask A of the NTCIR-18 SUSHI Task. This paper describes our approach to addressing the challenges of this task and discusses the official results.

The SUSHI (Searching Unseen Sources for Historical Information) task aims to retrieve undigitized materials, such as historical documents, handwritten notes, and legacy records, from large-scale archives. A key challenge is that these materials often lack item-level metadata, making traditional keyword search ineffective. General search approaches are incapable of assigning scores to non-existent items, rendering them unsuitable for such scenarios. This paper presents a novel approach that focuses on utilizing available metadata at different hierarchical levels (Box, Folder, and Item) and strategically combining their search scores, without relying on model fine-tuning.

*Also with National Institute of Informatics.

2 Related Work

A long line of IR research has explored how to exploit multiple document fields when scoring documents. Zaragoza and Robertson's BM25F shows that treating each document field individually and then merging the evidence with learned weights yields higher effectiveness for structured documents [5]. More recently, Yates et al. proposed the Neural Document Field (NDF) ranker, which feeds heterogeneous fields (short, long, and multi-instance) into a unified neural architecture and employs field-level dropout to handle missing values, achieving significant gains over classical learning-to-rank baselines [9]. Our system extends this philosophy in two dimensions: (i) it treats the archive's hierarchical levels (Box, Folder, and Item) as an additional set of "fields", and (ii) at each level it still performs separate searches over multiple metadata attributes (e.g., title, OCR, and file name) before normalizing and fusing the resulting scores. By keeping the fusion step lightweight and parameter-free, the method retains the interpretability of BM25F-style weighting while requiring fewer weight parameters than BM25F, and we therefore considered it suitable for scenarios where task-specific retraining is infeasible or metadata are severely sparse.

3 Methods

We developed and applied the following methods to address the challenges of the SUSHI task. We focused on a multi-level search and aggregation strategy, leveraging the hierarchical structure of the archive data.

3.1 Retrieval System Architecture

The SUSHI task data is organized hierarchically: Boxes contain Folders, and Folders contain Items. Metadata is provided at each level, but item-level metadata is often sparse or missing. Our submitted runs were generated by a system that performs searches at the Box, Folder, and Item levels, using their respective metadata. These search results are then aggregated to compute a final score for each folder, which is the unit of ranking for subtask A.

3.1.1 Metadata Preprocessing. We utilized the provided metadata files for Box, Folder, and Item levels. For Items, we used only the specified metadata fields. Crucially, to enhance the searchability of higher-level containers, we performed metadata aggregation. For Folders, we incorporated aggregated metadata from all Items belonging to that Folder. This involved consolidating relevant metadata fields from all Items within a Folder into a single, combined metadata representation for the Folder. Similarly, for Boxes, we aggregated metadata from all Folders within each Box. During aggregation, duplicate values within each metadata field were removed, retaining only unique values, which were then concatenated using commas (","). This aggregation process aims to improve

search accuracy at higher levels by compensating for the lack of direct item-level metadata and enriching the contextual information available at the Folder and Box levels. Item metadata also included OCR-extracted text data of the document, obtained using all pages of the provided PDFs. Folder labels were generated from metadata using the sample code provided with the test collection. Specifically, the code uses 1965 titles when available and defaults to 1963 titles otherwise, without incorporating scope notes.

3.1.2 Multi-level Search and Scoring. Searches were conducted on the metadata of Boxes, Folders, and Items using specific fields. We employed two search patterns: "Each" and "Merge". In the "Each" pattern, each specified metadata field was searched independently. In the "Merge" pattern, the values of the specified fields were concatenated with spaces, forming a single, combined search query. We experimented with different retrieval models: BM25, E5 (both base[1] and large[2] versions), and ColBERT.[3] Section 3.2 provides details on these models. BM25 serves as a widely used traditional baseline, while E5 and ColBERT are relatively new dense retrieval models that offer potential advantages in capturing semantic similarity.

For each query component ($q \in Q = \{q_t, q_d, q_n\}$ corresponding to Title, Description, and Narrative), separate searches were performed at each level (level $\in$ {Box, Folder, Item}). Let $S_{\text{level}}(q, d)$ be the raw relevance score obtained by the retrieval model for a document d (Box, Folder, or Item) at a given level for query component $q \in Q$. In each search, we applied a top-k cutoff, meaning that only the highest-scoring k results were retained. Let $TopK_{\text{level}}(q)$ be the set of top-k documents for query component q at level level. The scores from each search result were then normalized to ensure the sum of scores for the top-k results equaled 1:

$$\mathrm{S}_{\text{level}}(q, d) = \frac{s_{\text{level}}(q, d)}{\sum_{d' \in \mathrm{TopK}_{\text{level}}(q)} s_{\text{level}}(q, d')} \tag{1}$$

We applied this normalization step to ensure that each query component and each search level contributed equally to the final score. The final score for each folder was computed by averaging the normalized scores across Box, Folder, and Item level searches, using optimized weights.

3.1.3 Score Aggregation and Ranking. After calculating the normalized search scores at the Box, Folder, and Item levels for each query component, we first averaged these normalized scores across the three query components (Title, Description, Narrative) for each document D at each level:

$$\bar{\mathrm{S}}_{\text{level}}(Q, d) = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{S}_{\text{level}}(q, d) \tag{2}$$

We then aggregated these average scores at the Folder level by applying specific weights w_B, w_F, w_I to each level's score. These weights were treated as hyperparameters and optimized ($w_B = 0.1$, $w_F = 0.6$, and $w_I = 0.3$). The final score for a given Folder F, denoted as $\mathrm{S}_{\text{Final}}(F)$, was calculated as follows:

$$\begin{aligned} \mathrm{S}_{\text{final}}(Q, F) = & w_F \cdot \bar{\mathrm{S}}_{\text{Folder}}(Q, F) \\ & + w_I \cdot \sum_{I \in \text{ItemsIn}(F)} \bar{\mathrm{S}}_{\text{Item}}(Q, I) \\ & + w_B \cdot \bar{\mathrm{S}}_{\text{Box}}(Q, \text{BoxOf}(F)) \end{aligned} \tag{3}$$

where $\text{ItemsIn}(F)$ represents the set of Items belonging to Folder F, and $\text{BoxOf}(F)$ denotes the Box containing Folder F. In this process, the Folder-level average score ($\bar{\mathrm{S}}_{\text{Folder}}(Q, F)$) was used directly with its weight w_F. The weighted average Item-level scores ($\bar{\mathrm{S}}_{\text{Item}}(Q, I)$) for all Items within the Folder were summed and added. Similarly, the weighted average Box-level score ($\bar{\mathrm{S}}_{\text{Box}}(Q, \text{BoxOf}(F))$) of the containing Box was added. This aggregation method reflects the hierarchical structure of the data, ensuring that information from each level appropriately influences the final ranking. The final Folder scores, resulting from this aggregation (Equation 3), were then used to generate the submitted runs by ranking Folders in descending order of their scores.

3.2 Retrieval Models

This section describes the retrieval models used in our system. We chose to use BM25, E5, and ColBERT. For E5 and ColBERT, no fine-tuning was applied.

3.2.1 E5. In recent years, bidirectional encoder representations have rapidly advanced in Natural Language Processing (NLP), significantly improving the accuracy of search systems. Representative examples include BERT and Sentence-BERT. These methods enable similarity evaluation and relevance calculation by mapping the semantic features of words and sentences into a high-dimensional space using a bidirectional understanding of context. Building upon this, Wang et al. proposed E5 (Embeddings from bidirectional Encoder representations) [8]. E5 inherits the strengths of previous models while achieving more efficient parameter design and computational optimization. It particularly excels in enhancing the generalization of representations through contrastive pre-training on large-scale, unlabeled web text pair datasets. Furthermore, fine-tuning on labeled datasets (NLI, MS-MARCO [4], and NQ [3]) leads to high retrieval performance. To effectively capture asymmetric similarity, E5 employs a strategy of adding prompts as prefixes, clearly distinguishing between queries and passages. This is particularly relevant in search tasks where the query and document have different structures and objectives.

3.2.2 ColBERT. ColBERT, proposed by Khattab et al. [2], differs from traditional single-vector dense retrieval models that capture the overall context of the entire input in one vector. Instead, ColBERT calculates individual embeddings for each token in the text and employs a late interaction mechanism. This approach enables detailed matching between queries and documents at a granular, token level. Conventional methods, which compress the entire input into a single high-dimensional vector, may not fully reflect subtle differences in synonyms and context-dependent meanings. In contrast, ColBERT, leveraging large-scale pre-trained models like BERT, first generates independent embeddings for each token in both queries and documents. Then, during retrieval, it calculates the similarity between each token in the query and all tokens

[1] https://huggingface.co/intfloat/e5-base-v2
[2] https://huggingface.co/intfloat/e5-large-v2
[3] https://huggingface.co/colbert-ir/colbertv2.0

in the document, selecting the document token with the highest similarity for each query token. The relevance score is the sum of these maximum similarities. This comprehensive utilization of local semantic information, which is difficult to capture with single holistic representations, leads to improved retrieval accuracy.

Furthermore, ColBERTv2 [6], an evolution of ColBERT, aims to further enhance efficiency and accuracy while maintaining the original model's powerful token-level matching mechanism. ColBERTv2 uses techniques such as embedding space optimization, token representation compression, and normalization to generate more compact and expressive feature representations.

4 Runs

We submitted five runs for SUSHI Subtask A. Table 1 provides an overview of these runs. All runs were generated using the retrieval algorithm described in Section 3. In four runs (KASYS-1, 2, 3, and 4) the "Merge" pattern was applied—merging specified metadata fields into a single query and using BM25, E5-base, E5-large, and ColBERT as the retrieval models, respectively. In contrast, KASYS-5 used the "Each" pattern, in which each metadata field was searched separately using BM25. For both patterns, BM25 was configured with parameters $k_1 = 0.9$ and $b = 0.4$, which were not tuned.

Hyperparameters were optimized on the Dry Run test collection with the objective of maximizing nDCG@5, using the Optuna [1] library with a Tree-structured Parzen Estimator (TPE) sampler. The optimization targeted the following elements:

- **Metadata fields:** The specific fields to be searched at each level:
  - **Box:** `Brown Box Name`, `Item folder label`, `Item ocr`, `Item NARA File Name`, `Item NARA Title`
  - **Folder:** `Item Brown Title`, `Item Folder Label`
  - **Item:** `Folder Label`, `OCR`, `NARA File Name`, `NARA Title`
- **Top-k value:** The maximum number of top-scoring results to consider in score calculation (optimized to 50).
- **Aggregation weights:** The contribution of scores from Box, Folder, and Item levels (optimized to 0.1, 0.6, and 0.3, respectively).

5 Experiments

This section presents and discusses the official results of the SUSHI Task Subtask A for our submitted runs. Tables 2 and 3 show the folder and box ranking results for each run, respectively.

The runs using BM25 showed the best performance among our submissions. Specifically, KASYS-1 (BM25 with the "Merge" pattern) achieved the highest nDCG@5 score of 0.203 for folder ranking and 0.261 for box ranking. The dense retrieval models (runs KASYS-2, KASYS-3, and KASYS-4, using E5-base, E5-large, and ColBERT respectively) did not perform as well as BM25. This is likely because our search system concatenates several metadata values, which are primarily composed of words or short phrases, into a single string. This concatenation may introduce noise and disrupt the contextual information that dense retrieval models rely on. Furthermore, the lack of fine-tuning on the SUSHI data may have limited the ability of these models to adapt to the specific characteristics of the archive. In light of these findings, a promising direction for future work would be to explore a hybrid approach, where BM25 is used for initial retrieval and dense retrieval models are employed for re-ranking. This combined strategy is generally expected to enhance overall performance.

Furthermore, comparing the two BM25 configurations, KASYS-1 (using the "Merge" pattern) outperformed KASYS-5 (using the "Each" pattern). This suggests that merging fields before searching was more effective than searching fields individually in this setup. We hypothesize that there may be limitations on the effective parameters available for scoring at each level when fields are treated independently ("Each" pattern). This effect might be particularly pronounced for BM25 due to its reliance on term matching, where merging fields could provide a denser representation for matching compared to potentially sparse individual fields.

Comparing KASYS-1 and TerrierBaseline-TDN (a baseline provided by the organizers using BM25 with item titles, OCR, and folder labels, and a query combining Title, Description, and Narrative), we observe mixed results. While KASYS-1 achieved a slightly higher Success@1 score for folder ranking (0.356 vs 0.333), TerrierBaseline-TDN performed slightly better on Success@1 for box ranking (0.422 vs 0.378). nDCG@5 scores were comparable for both folder and box ranking. These results suggest that our multi-level score aggregation strategy did not provide a significant advantage over the simpler baseline approach in this specific case.

Several factors may have contributed to these results. One key challenge is the presence of noisy, duplicated, and missing metadata at different levels. For instance, the OCR information aggregated at the Box and Folder levels often contains errors originating from the Item-level OCR, potentially degrading the quality of searches at those levels. This issue affects not only KASYS-1 but also runs using E5 and ColBERT. Future work should explore methods for cleaning and filtering the OCR data, such as applying OCR error correction techniques or selectively using OCR text based on confidence scores. Additionally, the utilization of semantically less meaningful metadata items, such as file names and IDs, may require different strategies, such as incorporating entity linking or leveraging external knowledge graphs.

6 Conclusions

For the SUSHI Subtask A, focused on searching for undigitized materials in large archives with limited item-level metadata, we conducted retrieval experiments leveraging metadata at different hierarchical levels (Box, Folder, and Item) without fine-tuning retrieval models. Our results demonstrate that, while the traditional BM25 model achieved the best performance among our submitted runs, our multi-level score aggregation strategy, combined with hyperparameter optimization, did not significantly outperform a simpler BM25 baseline. This highlights the challenges of searching undigitized content with sparse and noisy metadata.

This study contributes to the understanding of effective retrieval strategies for undigitized archives by exploring the potential of no fine-tuning models and multi-level metadata aggregation. Future work will focus on several key areas: 1) Developing robust preprocessing techniques to handle noisy and missing metadata, particularly OCR data; 2) Investigating hybrid retrieval approaches that combine the strengths of BM25 and dense retrieval models,

Table 1: KASYS runs.

Run name	Description
KASYS-1	Integrated search of target items by level using BM25
KASYS-2	Integrated search of target items by level using E5-base
KASYS-3	Integrated search of target items by level using E5-large
KASYS-4	Integrated search of target items by level using ColBERT
KASYS-5	Separate search of target items by level using BM25

Table 2: Subtask A folder ranking results of KASYS runs.

Run name	nDCG@5	MAP	MRR	Success@1
TerrierBaseline-TDN	0.203	0.132	0.417	0.333
KASYS-1	0.203	0.129	0.417	0.356
KASYS-2	0.059	0.073	0.14	0.067
KASYS-3	0.081	0.082	0.165	0.067
KASYS-4	0.12	0.102	0.264	0.178
KASYS-5	0.131	0.094	0.272	0.178

Table 3: Subtask A box ranking results of KASYS runs.

Run name	nDCG@5	MAP	MRR	Success@1
TerrierBaseline-TDN	0.266	0.25	0.526	0.422
KASYS-1	0.261	0.228	0.5	0.378
KASYS-2	0.072	0.159	0.2	0.067
KASYS-3	0.113	0.171	0.22	0.067
KASYS-4	0.199	0.224	0.369	0.222
KASYS-5	0.226	0.198	0.372	0.2

such as re-ranking or late fusion; 3) Exploring methods for incorporating external knowledge and entity linking to enhance the semantic understanding of metadata; and 4) Evaluating the effectiveness of fine-tuning retrieval models on the SUSHI data to further improve performance.

Acknowledgments

This work was supported by Japan Society for the Promotion of Science KAKENHI Grant Number JP23K28090.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[2] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3397271.3401075

[3] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (08 2019), 453–466. https://doi.org/10.1162/tacl_a_00276 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00276/1923288/tacl_a_00276.pdf

[4] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. (November 2016). https://www.microsoft.com/en-us/research/publication/ms-marco-human-generated-machine-reading-comprehension-dataset/

[5] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple BM25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management* (Washington, D.C., USA) *(CIKM '04)*. Association for Computing Machinery, New York, NY, USA, 42–49. https://doi.org/10.1145/1031171.1031181

[6] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 3715–3734. https://doi.org/10.18653/v1/2022.naacl-main.272

[7] Tokinori Suzuki, Douglas Oard, Shashank Bhardwaj, Emi Ishita, and Yoichi Tomiura. 2025. NTCIR-18 SUSHI Pilot Task Overview. In *In Proceedings of the 18th NTCIR Conference on Evaluation of Information Access Technologies, June 10-13, 2025, Tokyo, Japan.*

[8] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. arXiv:2212.03533 [cs.CL] https://arxiv.org/abs/2212.03533

[9] Hamed Zamani, Bhaskar Mitra, Xia Song, Nick Craswell, and Saurabh Tiwary. 2018. Neural Ranking Models with Multiple Document Fields. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 700–708. https://doi.org/10.1145/3159652.3159730