Scalable Hybrid Retrieval for Natural Language Email Queries

Calder Katyal
Yale University
calder.katyal@yale.edu

Jess Yatvitskiy
Yale University
jess.yatvitskiy@yale.edu

Abstract

Most email retrieval systems still rely on basic keyword matching, often failing when queries lack shared terms or use natural language. These systems also overlook entities like senders or dates unless formatted rigidly. We introduce a scalable hybrid retrieval framework that combines sparse keyword search with dense semantic retrieval. Our approach expands queries into diverse variants, embeds them using a pretrained model, and fuses rankings to improve robustness. On a public email dataset, our method produces consistent, semantically relevant results across diverse queries and outperforms either component alone.[1]

1 Background

Information retrieval in the context of emails presents unique challenges due to the diverse and unstructured nature of email content, varying lengths, and implicit contextual information. Traditional information retrieval systems have evolved from simple keyword matching to more sophisticated semantic understanding techniques.

Keyword-based retrieval. Classical information retrieval has relied heavily on lexical matching techniques such as BM25 [13], which extends the TF-IDF framework by incorporating document length normalization and term saturation. While effective for explicit keyword queries, these approaches struggle with vocabulary mismatch problems and lack semantic understanding [6]. In email contexts specifically, lexical retrieval faces additional challenges as email communications often contain jargon, abbreviations, and implicit references that require contextual knowledge to interpret correctly.

Neural retrieval models. With advances in deep learning, dense retrieval models have emerged as powerful alternatives to traditional sparse retrieval methods. Models like Sentence-BERT [12] and DPR [7] map text to continuous vector spaces where semantic similarity can be measured more effectively. These approaches have demonstrated significant improvements over lexical methods on various retrieval benchmarks [15, 10]. Yet dense models alone can miss critical exact-match cues—like project codes or sender names—so their recall gains do not always translate into precision on noisy, domain-specific email corpora.

Natural language understanding for structured queries. Converting natural language queries into structured representations has been explored through text-to-SQL frameworks such as Seq2SQL [19] and SQLNet [16]. These approaches interpret user intent and translate it into formal query languages that can precisely filter and retrieve information. Recent transformer-based models such as BART [9] and T5 [11] have further improved natural language understanding capabilities, enabling more accurate translation of complex queries into structured forms.

[1] Code accessible at `https://github.com/calderkatyal/CPSC-477-Final-Project`.

Hybrid retrieval systems. Recognizing the complementary strengths of lexical and semantic approaches, hybrid retrieval systems combine multiple retrieval methods to achieve better performance. Late fusion techniques such as Reciprocal Rank Fusion (RRF) [2] and CombSUM [4] merge rankings from different retrieval systems without requiring score normalization. More sophisticated approaches like CLEAR [5] and ColBERT [8] integrate lexical and semantic signals at different stages of the retrieval pipeline. In the context of email search, hybrid approaches are particularly valuable as they can leverage both keyword matching for precision and semantic understanding for recall [17].

Domain adaptation and personalization. Email search is inherently personal, with relevance often depending on user-specific contexts and relationships. Recent work has explored personalized retrieval models that adapt to individual users' search patterns and preferences [18, 1]. Techniques such as few-shot learning [14] and meta-learning [3] have shown promise for adapting general-purpose retrieval models to domain-specific applications with limited labeled data, which is particularly relevant for email search where privacy concerns often limit data availability.

Our work builds upon these foundations by integrating advanced keyword-based retrieval with semantic search methods specifically optimized for email queries. We propose a novel approach that dynamically balances lexical and semantic signals based on query characteristics, addressing the unique challenges of natural language email search in a scalable manner.

2 Method

Given a query, we combine an improved sparse keyword retrieval method with a semantic search algorithm. For the keyword search, we first preprocess the query and reconstruct it into an Elasticsearch DSL query, then we perform the search using ElasticSearch's BM25 ranking model. For our semantic search algorithm, we perform three steps sequentially: query expansion, dense retrieval, and rank fusion. After obtaining the rankings from the keyword and semantic searches, we interpolate the scores to produce the final rankings. A schematic is found in Fig. 1.

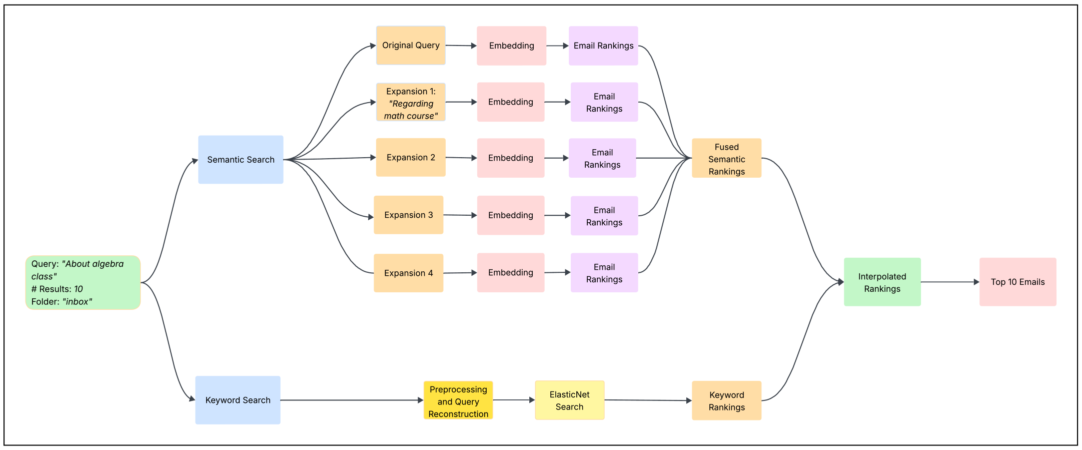

Figure 1: Email search system schematic.

2.1 Semantic Search

Query expansion. Because a single phrasing seldom covers all relevant surface forms, we generate paraphrases of the user query with the fine-tuned BART model `eugenesiow/bart-paraphrase`, using CPU and standard float32 representation. Diverse paraphrases are generated using stochastic decoding with top-k and nucleus sampling, and duplicates are dropped. The original query is retained as one variant. We seek a maximum of $n = 5$ variants, including the original query.

Dense retrieval. Every email (subject + body) is pre-embedded with the 1.5B-parameter `infly/inf-retriever-v1-1.5b` model and stored in a FAISS index for efficient maximum-inner-product search. Embeddings for the emails were generated using a single NVIDIA A100 GPU using

BF16 precision. Embeddings for each variant of the user query (including the source query) are computed on the fly using the same model and precision with CPU. Cosine similarity ranks the semantic relevance of each email to every query variant. The model is both lightweight and SOTA on the MTEB benchmark [10] for retrieval.

Rank fusion. The m query variants each yield their own ranked list of emails via dense retrieval. Since all lists are derived from semantic similarity using paraphrased queries, we treat them as equally trustworthy and aggregate them using Reciprocal Rank Fusion (RRF) [2]. RRF rewards documents that appear across multiple rankings, especially if they rank high in those lists. For the formula and details, see Appendix A.

2.2 Keyword Search

Query parsing and reconstruction. Most email search systems fail to identify senders or dates unless queries follow rigid formats. To address this, we use spaCy to extract sender mentions (e.g., "from Alice") and map names to known aliases (e.g., "Alice" → `alice@corp.com`). While our dataset provided this mapping, it could also be inferred by linking names and email addresses across messages. Date mentions (e.g., "last July") are extracted using dateparser and standardized. We then remove stop words and reassemble these components into an Elasticsearch query, specifying that if sender or date information is present, emails should receive a boost to matching fields. The final query is scored over a pre-built Elasticsearch index.

Keyword Ranking. Elasticsearch returns BM25 scores based on term frequency, inverse document frequency, and length normalization. Each query term contributes additively to the total document score, adjusted by its importance (via IDF), frequency, and document length. For the full scoring formula and parameters, see Appendix B.

2.3 Score Interpolation

We first min-max normalize the semantic and keyword scores to yield scores S_i and K_i for each email. The final score for email i is given by the interpolation

$$C_i = \lambda(L) \cdot S_i + (1 - \lambda(L)) \cdot K_i \tag{1}$$

where L is the length of the source query and $\lambda(L) \in [0.25, 0.75]$ is a monotonically increasing function of L. Specifically, we choose $\lambda(L) = 0.25 + 0.5 \cdot \big((1 + e^{-0.9(L-4)})^{-1} - (1 + e^{2.7})^{-1}\big)$, which has the property that $\lambda(1) = 0.25$, $\lambda(4) = 0.5$, and $\lambda(k)$ asymptotically approaches 0.75. The intuition is that a longer query will contain more semantic information and will naturally benefit more from semantic search, while a shorter query is most likely composed of keywords.

3 Experiments

We tested our system with various queries to analyze the performance of the keyword, semantic, and hybrid retrieval methods. For assessing quality of results, we began with manually reviewing the top scored emails. We then employed modified versions of well-known consistency metrics to measure the consistency of the results our system produced across semantically similar queries.

Dataset We used the Clinton Email Dataset[2], which includes metadata and content for thousands of declassified emails. We cleaned headers and punctuation, extracted aliases from sender fields, normalized subjects and bodies, and parsed timestamps. Using alias-to-person mappings, we split the corpus into 4,706 received and 1,579 sent emails based on whether Hillary Clinton was the sender or a recipient. Emails with missing content were removed. The processed data was saved in structured form for retrieval experiments.

Objective measures. In the absence of labeled data, we must design new measures to assess ranking consistency. To evaluate our method, we form two semantic groups, each composed of four queries. For the first group, the queries were semantically similar but differ in lexical content; for the second

[2] Data accessible at `https://www.kaggle.com/datasets/kaggle/hillary-clinton-emails/data`.

group, the queries are semantically different. To measure consistency, we use two metrics: weighted Kendall's W, and weighted pairwise MSE, both discussed in Appendix C. Both metrics return scalars in [0,1]; a Kendall's W of 1.0 and MSE of 0.0 corresponds to perfect consistency in the semantic group. We test our hybrid method, as well as the individual semantic and keyword components. The specifics of the experiment are given in Appendix D; results are found in 1.

Table 1: Consistency across semantically similar and semantically different queries.

Semantic Group	Method	Weighted Kendall's W	Weighted Pairwise MSE
Similar	Hybrid	0.91 ± 0.07	0.13 ± 0.04
	Semantic	0.98 ± 0.01	0.12 ± 0.07
	Keyword	0.21 ± 0.01	0.38 ± 0.06
Different	Hybrid	0.17 ± 0.07	0.43 ± 0.10
	Semantic	0.34 ± 0.11	0.51 ± 0.03
	Keyword	0.19 ± 0.02	0.44 ± 0.09

Our hybrid method thus has roughly the same consistency as the semantic component alone. The results indicate that the poorer performance of keyword search in capturing semantic meaning does not significantly impair our hybrid scoring, implying that we can benefit from the aforementioned unique advantages of keyword search without too strongly compromising our semantic capabilities.

Subjective analysis. Semantic search generally returned highly relevant emails, particularly for abstract or open-ended queries. Keyword search was less consistent: its effectiveness depended on whether the query terms appeared verbatim in the relevant emails. For example, it performed well on "Expressing gratitude or praise" but poorly on "Making plans to talk." By contrast, it excelled when queries included names, acronyms, or uncommon phrases that appeared in the dataset.

As expected, hybrid search performed well when both individual methods did. It also benefited when semantic results contained key terms, allowing keyword scores to reinforce them. However, when the two methods diverged—due to typos, unusual synonyms, or dataset-specific language—hybrid results were often mixed, with some relevant emails appearing alongside irrelevant ones.

4 Limitations and Future Work

Our hybrid system performs well across diverse queries, but it struggles when semantic and keyword rankings diverge—such as in cases with typos, uncommon phrasing, or domain-specific terms—often leading to noisy top results. A natural next step is to make interpolation weights adaptive to query features beyond length, such as query perplexity.

Regarding semantic search, email chains often provide more context than individual emails; it may be worth creating embeddings for them instead. Senders and receivers also add context, so we might generate embeddings for people based on their associated emails and incorporate these into email embeddings. Addressing domain-specific words, we might preprocess the email set to find unusual words and predict their meaning based on surrounding context, generating embeddings for them that can be incorporated into the query embeddings.

We might also fine-tune our model through contrastive learning or by curating a small set of labeled data. Moreover, our system currently runs offline. Extending it to an online setting and incorporating user-specific context remain important directions for future work.

5 Conclusion

We introduced a scalable hybrid retrieval system that effectively combines keyword and semantic signals to improve natural language email search. By fusing semantic variants and interpolating semantic and keyword scores, our approach captures both exact matches and nuanced meaning. Experiments demonstrate that the hybrid method leverages the unique strengths of both semantic and keyword search, providing a strong foundation for more adaptive and personalized retrieval systems in real-world email search.

References

[1] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 645–654, 2017.

[2] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584836. doi: 10.1145/1571941.1572114. URL `https://doi.org/10.1145/1571941.1572114`.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017.

[4] Edward A Fox and Joseph A Shaw. Combination of multiple searches. In *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 243–252, 1993.

[5] Tianyu Gao, Xingcheng Yao, and Danqi Chen. CLEAR: Contrastive learning for sentence representation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5939–5951, 2021.

[6] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 55–64, 2016.

[7] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781, 2020.

[8] Omar Khattab and Matei Zaharia. 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*, pages 39–48, 2020.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[10] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023. URL `https://arxiv.org/abs/2210.07316`.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[12] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, 2019.

[13] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[14] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[15] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems*, volume 34, pages 6010–6022, 2021.

[16] Xiaojun Xu, Chang Liu, and Dawn Song. SQLNet: Generating structured queries from natural language without reinforcement learning. In *International Conference on Learning Representations*, 2017.

[17] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. Critically examining the neural hype: Weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1129–1132, 2019.

[18] Hamed Zamani, Michael Bendersky, Xuanhui Wang, and Mingyang Zhang. Situational context for ranking in personal search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1531–1540, 2017.

[19] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2SQL: Generating structured queries from natural language using reinforcement learning. In *International Conference on Learning Representations*, 2017.

Appendix

A RRF Scoring Formula

Given a document d, its Reciprocal Rank Fusion (RRF) score is computed as:

$$\mathrm{RRF}(d) = \sum_{i=1}^{n} \frac{1}{k + r_{i,d}} \tag{2}$$

where $r_{i,d}$ is the 0-based rank position of document d in the i-th ranking, and k is a constant (we use $k = 60$) that dampens the impact of lower-ranked results.

B BM25 Scoring Formula

BM25 ranks documents by balancing three factors: how often query terms appear in a document (term frequency), how rare those terms are in the entire corpus (inverse document frequency), and how long the document is (length normalization). This helps prioritize documents that are both relevant and concise.

The full BM25 score for a document D given a query with terms $q_1, q_2, \ldots, q_n$ is computed as:

$$\sum_{i=1}^{n} \mathrm{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{\text{fieldLen}}{\text{avgFieldLen}}\right)} \tag{3}$$

where $f(q_i, D)$ is the frequency of query term q_i in document D, $\mathrm{fieldLen}$ is the length of the document, and $\mathrm{avgFieldLen}$ is the average length of all documents in the corpus. The parameters k_1 and b control term frequency scaling and length normalization, respectively (commonly set to $k_1 = 1.2, b = 0.75$). Finally, $\mathrm{IDF}(q_i)$ is the inverse document frequency of the term, which increases the weight of rarer terms.

C Evaluation Metrics

The two objective measures we use to assess the accuracy of our method are discussed below.

Weighted Kendall's W. To evaluate full-rank agreement, we adapt Kendall's coefficient of concordance with exponential weighting. For each email d, we collect its rank across the m rankings and compute the variance. We weight each email based on its best (lowest) rank:

$$w_d = \exp\left(-\frac{\min_i r_{i,d}}{\tau}\right) \tag{4}$$

where $\tau = 20$ is a decay constant. The final score is computed by normalizing the weighted average variance:

$$W = 1 - \frac{\sum_d w_d \cdot \mathrm{Var}_d}{\sum_d w_d \cdot \frac{N^2-1}{12}} \tag{5}$$

where N is the number of ranked items. The result is a scalar in [0,1], where higher values of W indicate stronger agreement across the rankings, particularly among high-confidence results.

Weighted Pairwise MSE. While Kendall's W measures agreement in rank order, we also evaluate how consistently the system assigns scores. For each email and each pair of rankings, we compute the squared difference in scores, weighted by the higher-ranked appearance using the same weighting scheme as in Weighted Kendall's W:

$$\text{Weighted-MSE} = \frac{1}{Z} \sum_{i<j} \sum_{d} \exp\left(-\frac{\min(r_{i,d}, r_{j,d})}{\tau}\right) \cdot (s_{i,d} - s_{j,d})^2 \tag{6}$$

where $s_{i,d}$ is the score assigned to email d in the i-th ranking, and Z is the total sum of weights and acts as a normalization constant. This penalizes score instability, especially for top-ranked emails. The result is a scalar in [0,1], where a lower value corresponds to greater score agreement.

D Query Sets by Semantic Group

To assess consistency, we formed two groups: one for semantically similar queries, and one for semantically different queries. Each group contains three sets of four queries, which we use to assess consistency.

Semantically Similar Sets

Set 1. Query 1: Expressing gratitude or praise
Query 2: Offering thanks or compliments
Query 3: Conveying gratefulness or approval
Query 4: Demonstrating approval or gratitude

Set 2. Query 1: Reports of bombings and other crises
Query 2: News about bombings and other emergencies
Query 3: Accounts of explosions and related disasters
Query 4: Reports concerning explosive attacks and other calamities

Set 3. Query 1: Will you be able to talk soon?
Query 2: Do you have time to chat?
Query 3: Can we have a call?
Query 4: Can we touch base?

Semantically Different Sets

Set 4. Query 1: Reports of bombings and other crises
Query 2: Expressing gratitude or praise
Query 3: Will you be able to talk soon?
Query 4: Making plans for future improvements

Set 5. Query 1: Asking for updates on a situation
Query 2: Providing someone else's opinion
Query 3: Major developments or crises
Query 4: Can we have a call?

Set 6. Query 1: Regarding Syria or Ukraine
Query 2: Expressing gratitude or praise
Query 3: Making plans to talk
Query 4: Sharing a memo or an article