Ensemble Retrieval Strategies for an Improved NAICS Search Engine in the U.S. Census Bureau

Cameron Milne, Yezzi Angi Lee, Taylor Wilson, Hector Ferronato
Reveal Global Consulting, 8115 Maple Lawn Blvd, Fulton, MD 20759

Abstract
Large Language Models (LLMs) have achieved significant gains in performance over classical approaches in Information Retrieval (IR) systems. Semantic search, a concept involving a contextual understanding of a query, significantly outperforms its keyword predecessors in recent academic literature. This paper examines five research questions critical to an effective multi-stage retrieval pipeline: [1] how much better can dense embeddings perform over sparse embeddings? [2] what combination of context windows in an ensemble approach can maximize first-candidate generation? [3] among top performers in ensemble approaches, which approach can generate the highest accuracy scores when we increase *k?* [4] can a fast-reranking algorithm complement first-candidate generation to further boost accuracy? [5] can a cache of seen-before queries help boost performance where the corpus fails? This project designs a novel search engine for the North American Industry Classification System (NAICS) using semantic retrieval models and achieves an accuracy of 86.87% on synthetic data, a 45.1% improvement over a keyword search. The strategies for bootstrapping data, selecting language models, and using open-source technologies within this project can serve as reference material for future LLM applications in the U.S. Census Bureau.

Keywords: Large Language Models (LLMs), Semantic Search, Information Retrieval, Natural Language Processing (NLP), Federal Statistics

1. Introduction

The North American Industry Classification System (NAICS) is a standard developed jointly by the United States, Canada, and Mexico to better categorize and analyze data around their economies (United States Census Bureau 2019). NAICS was developed by the Office of Management and Budget (OMB) in 1997 to replace the Standard Industrial Classification (SIC) system with innovative classification methodologies for the purpose of supporting interoperability of economic classification between the U.S., Canada, and Mexico. Since 1997, NAICS has evolved through five-year review cycles to 1,012 unique codes within 20 sectors that group establishments by similarity in processes used to produce goods or services. Examples range from *Soybean Farming* (111110) comprising establishments primarily engaged in growing or producing soybeans to *Used Merchandise Retailers* (459510) which includes establishments retailing used merchandise, antiques, and secondhand goods. Common definitions for industries and economic sectors such as these have helped the U.S., Canada, and Mexico to understand economic inputs and outputs, industrial performance, productivity, unit labor costs, and employment growth in North America.

The U.S. Census Bureau offers a NAICS search engine[1] on its website that employs string-matching: an information retrieval (IR) technique that identifies exact matches between a query and a corpus. This approach is supported by Bureau analysts who maintain detailed documentation of descriptions, illustrative examples, and cross-references that evolve with economic terminology. However, string-matching itself constrains user experience by failing to provide relevant documents when queries are lengthy or complex. As a result, Census classification analysts report 15-25% of

[1] https://www.census.gov/naics/

their work hours are dedicated to assisting users over email and phone, impressing the need for more advanced tools to efficiently manage user queries.

Recent progress in instruction fine-tuned large language models (LLMs) is driving fast-paced advances in IR systems by transitioning from lexical search strategies to semantic search. Semantic search, where "semantic" meaning is given consideration through the underlying language model's self-attention mechanism, is more effective at handling ambiguity and delivering targeted results. Relatedly, the increasing availability of powerful language models on platforms like Hugging Face, inexpensive vector databases to replace NoSQL databases with faster indexing and querying, and evolving research around retrieval-augmented generation (RAG) have made development of customized search engines easier. These advancements present an opportunity for enhancing the retrieval tools available for analysts.

This paper introduces a NAICS search engine built on semantic retrieval models. Our project seeks to answer a few questions related to search: [RQ1] How much better can dense embeddings perform over sparse embeddings? [RQ2] What combination of context windows in an ensemble approach can maximize first-candidate generation? [RQ3] Among top performers in ensemble approaches, which approach can generate the highest accuracy scores when we increase *k?* [RQ4] Can a fast-reranking algorithm complement first-candidate generation to further boost accuracy? [RQ5] Can a cache of seen-before queries boost performance where the corpus fails? By performing a series of experiments, we draw conclusions on critical elements of a multi-stage retrieval pipeline for this use-case and provide a roadmap for improvement through data augmentation.

This project utilizes open-source, state-of-the-art technologies in machine learning and natural language processing, advancing the Federal government's efforts to embrace artificial intelligence (AI) initiatives while alleviating a taxing burden for human analysts in the U.S. Census Bureau. This paper aims to provide a framework for testing and deploying LLM applications for production in the Federal government.

2. Background and Related Work

Retrievers and reranking are the primary objects of study for this project. Per standard parlance in IR, *document* generally refers to a unit of text being retrieved and can vary from a single keyword to multiple passages.

2.1 Retrievers

Given a corpus C = {D_1, D_2, …, D_n} containing a collection of documents and a query *q*, a retriever should efficiently return a list of *k* documents from C that are most relevant to the query *q* according to some metric relevant to the use-case. While researchers typically use nDCG or mean average precision to assess the performance of these systems, custom-use cases require different signals for relevance.

"Bag of word" models including TF-IDF and BM25 represent the classical approaches to retrieving relevant documents and have received interest in academia (Matveeva, et al. 2006, Wang, Lin and Metzler 2011, Asadi and Lin 2013, Chen, et al. 2017, Mackenzie, et al. 2018) as well as industry. Bing's search engine (Pedersen 2010) and Alibaba's ecommerce platform (Liu, et al. 2017) reportedly used these approaches in the early 2000s. These lexical approaches can perform modestly in first-stage candidate generation and serve as strong baselines for zero-shot retrieval (Thakur, et al. 2021), but can struggle to overcome lexical gaps (Berger, et al. 2000). Specifically, they are prone to returning false positives or irrelevant results when search terms have multiple interpretations.

Semantic search updates these methods through a few key differences: using dense representations typically learned from questions and passages by an encoder-encoder framework (Karpukhin, et al. 2020) and replacing the discrete inverted index with standard approximate nearest neighbor (ANN) to rely on distances between learned embeddings (Gillick, Presta and Singh Tomar 2018). Dualencoder neural architectures using pre-trained transformers have shown strong performance on various open-domain question answering tasks (Guo, et al. 2020, Karpukhin, et al. 2020, Liang, et al. 2020, Ma, et al. 2020) and have become a promising framework for advanced IR systems. More recently, the dense approach was extended by hybrid lexical-dense approaches aiming to combine the strengths of both approaches and overcome any weaknesses in distance-based retrieval (Luan, et al. 2021, Raffel, et al. 2020, Gao, et al. 2021).

Relatedly, research on RAG architectures (Lewis, et al. 2020) have advanced IR research as well by exhaustively studying chunking approaches (Liu, et al. 2021, Izacard and Grave 2020, Gong, et al. 2020, Majumder, et al. 2021), lengths of context windows (Zaheer, et al. 2020, Borgeaud, et al. 2022, Izacard and Grave 2020), model selection (Kandpal, et al. 2023, Zhao, et al. 2024, Neelakantan, et al. 2022), and more. Results from these studies often depend on details specific to the use-case, necessitating a kitchen-sink strategy for experimentation in this project to understand where each of these variables should be tuned.

2.2 Reranking

Reranking a collection of *k* documents after retrieval can often further improve the quality of documents retrieved using the same metric or an alternative (particularly useful when the first-candidate generation is through dense retrieval which automatically performs ranking). Retrievers and reranking together form multi-stage ranking pipelines for text ranking and have been studied in both IR and NLP contexts in recent years (Lin, Nogueira and Yates 2021).

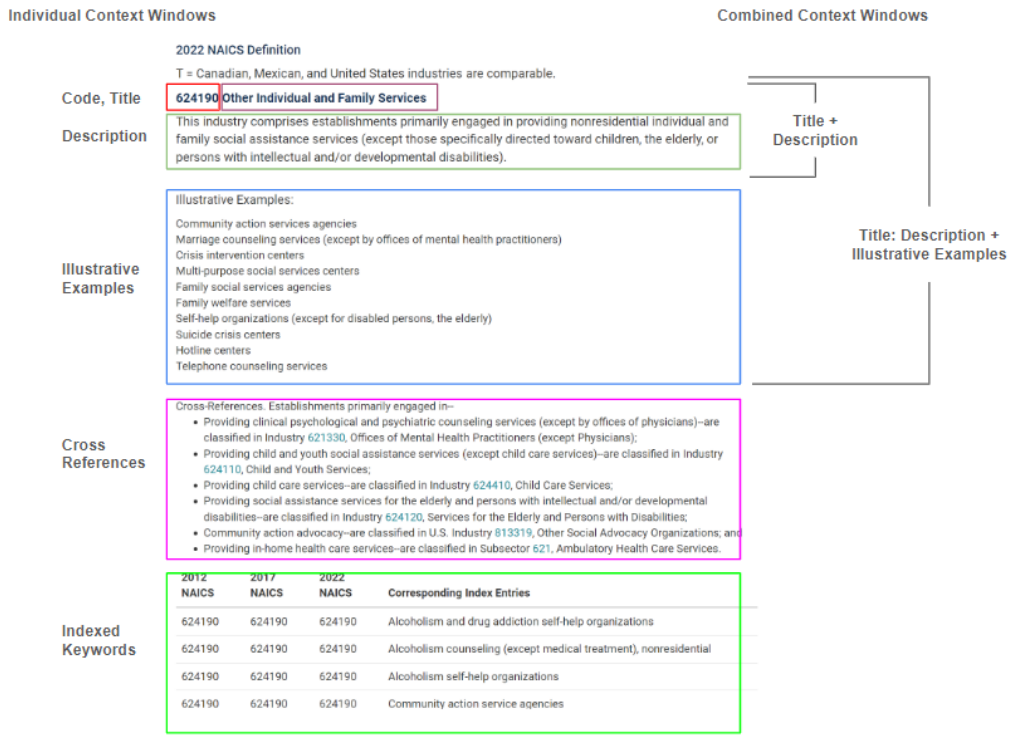
Classical approaches to reranking, including Condorcet Fuse and Reciprocal Rank Fusion (RRF), have provided effective reranking for hybrid lexical-dense retrieval systems by combining results and recomputing based on predetermined weights (Cormack, Clarke and Buttcher 2009). Yet, these methods are LLM-agnostic and require hyperparameter tuning for optimal performance—a challenge when a project lacks reliable data for development.

Contextualized Late Interaction over BERT (ColBERT) is a novel reranking strategy that computes multiple contextualized embeddings at the token level for both queries and documents and uses a maximum-similarity function for retrieval or reranking (Thakur, et al. 2021). The priority on keywords, combined with applying the attention mechanism after encoding prevents the need for storing documents in a vector space until called upon (Khattab and Zaharia 2020). ColBERT can leverage the expressiveness of a power language model while performing orders-of-magnitude faster, suggesting a powerful complementary algorithm for dense retrieval.

LLMs have also been researched in the context of reranking, with RankGPT (Sun, et al. 2023), RankVicuna (Pradeep, Sharifymoghaddam and Lin, RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models 2023), and RankZephyr (Pradeep, Sharifymoghaddam and Lin, RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! 2023) demonstrating strong reasoning capabilities with applied prompt engineering. However, this project aims to provide a self-service platform where analysts can interact with the search engine directly, necessitating fast inference. Given these constraints, we elected to experiment without seq2seq models as rerankers.

3. Proposed Methods

We explore each component of a multi-stage retrieval pipeline: assembling a corpus, model selection, reranking design, and generating synthetic testing data.



2022 NAICS Definition

T = Canadian, Mexican, and United States industries are comparable.

624190 Other Individual and Family Services

This industry comprises establishments primarily engaged in providing nonresidential individual and family social assistance services (except those specifically directed toward children, the elderly, or persons with intellectual and/or developmental disabilities).

Illustrative Examples:

Community action services agencies
Marriage counseling services (except by offices of mental health practitioners)
Crisis intervention centers
Multi-purpose social services centers
Family social services agencies
Family welfare services
Self-help organizations (except for disabled persons, the elderly)
Suicide crisis centers
Hotline centers
Telephone counseling services

Cross-References. Establishments primarily engaged in--

- Providing clinical psychological and psychiatric counseling services (except by offices of physicians)--are classified in Industry 621330, Offices of Mental Health Practitioners (except Physicians);
- Providing child and youth social assistance services (except child care services)--are classified in Industry 624110, Child and Youth Services;
- Providing child care services--are classified in Industry 624410, Child Care Services;
- Providing social assistance services for the elderly and persons with intellectual and/or developmental disabilities--are classified in Industry 624120, Services for the Elderly and Persons with Disabilities;
- Community action advocacy--are classified in U.S. Industry 813319, Other Social Advocacy Organizations; and
- Providing in-home health care services--are classified in Subsector 621, Ambulatory Health Care Services.

2012 NAICS	2017 NAICS	2022 NAICS	Corresponding Index Entries
624190	624190	624190	Alcoholism and drug addiction self-help organizations
624190	624190	624190	Alcoholism counseling (except medical treatment), nonresidential
624190	624190	624190	Alcoholism self-help organizations
624190	624190	624190	Community action service agencies

Figure 1: Example document with annotations for proposed context windows

3.1 Corpus

All *documents* containing information associated with the 1,012 unique NAICS codes are publicly available on the U.S. Census Bureau's website. Files of particular interest were the 2022 NAICS Descriptions containing *Codes*, *Titles*, *Descriptions*, and *Illustrative Examples*, the 2022 NAICS Cross-References, and the 2022 NAICS Index File which contains indexed keywords. From these files, they are chunked into their context windows and stored in a vector database. Because each *document* was reasonably small (including combined context windows), chunking size experimentation was unnecessary.

Seven *documents* (i.e. context windows) are used for experimentation. Individual context windows such as *Title* (T), *Description* (D), *Illustrative Examples* (IE), *Cross-References* (CR), and *Indexed Keywords* (IK) when used in combination represent a complementary ensemble approach. Additional context windows are combinations of these sections: *Title* + *Description* (T-D) and *Title + Description* + *Illustrative Examples* (T-D-IE). By using these sections in addition to the individual context windows, we can examine whether overlapping information helps home in on the best answers or crowds them out. Figure 2 provides an example of an existing document as seen on the legacy NAICS search engine with annotations of each context used in the experiments.

Additionally, we hypothesize that where semantic relationships are weak between queries and existing documents, a caching mechanism where historical queries are stored can help address edge cases. The cache is a sample of 679 annotated emails representing "real" queries supplied by Census analysts. The original sample contained 1,910 emails and had been stripped of any personally identifiable information. Pruning unrelated emails (general inquiries, emails lacking business descriptions, non-English, 2017 NAICS codes, etc.) left 1,291 emails, which were then denoised with Zephyr-7b-Beta (Tunstall, et al. 2023), a seq2seq model available on Hugging Face using a prompt that extracts the sentence or two where a business description is mentioned. Removing unnecessary language was crucial for improving signal in prediction and altering queries towards those expected by a search engine. From there, we performed our own annotations and removed any descriptions that evaded previous filters for relevant queries or consisted of typos and incomplete thoughts. The resulting cache consisted of 679 unique queries representing 25.59% of all unique 2022 NAICS codes. Table 2 displays the most frequently seen NAICS codes.

Table 2: Most Frequently seen NAICS codes in Cache

Count	Code	Title	Sector
37	458110	Clothing and Clothing Accessories Retailers	Retail Trade
32	459999	All Other Miscellaneous Retailers	Retail Trade
29	812990	All Other Personal Services	Other Services (except Public Administration)
20	711510	Independent Artists, Writers, and Performers	Arts, Entertainment, and Recreation
16	561720	Janitorial Services	Administrative and Support and Waste Management and Remediation Services
13	339999	All Other Miscellaneous Manufacturing	Manufacturing
12	236118	Residential Remodelers	Construction
11	541990	All Other Professional, Scientific, and Technical Services	Professional, Scientific, and Technical Services
9	445298	All Other Specialty Food Retailers	Retail Trade
8	523910	Miscellaneous Intermediation	Finance and Insurance

3.2 Embeddings Model Selection

Embeddings models were sourced from Hugging Face where open-source models are made freely available. Using the MTEB English leaderboard[2], approximately fifteen models were selected to see which model would best approximate our corpus' vocabulary. Criteria for models were relaxed on underlying training data in effort to sample diverse vocabularies, but were constrained to English and a model size of less than 1 GB.

Separate vector databases were created for the corpus and each embeddings model before performing semantic search on the synthetic data—the dataset that was believed to best approximate a query seen in a search engine. The document for this stage of experimentation was T-D-IE with results recorded for accuracy at Hit@1 and Hit@5. The highest performing model, *thenelper/gte-base* (Li, et al. 2023) with a dimension of 768 and size of 0.22 GB at an accuracy of 83.4% at Hit@5 was selected for experimentation. The lowest performing model achieved an accuracy of 62.1% at Hit@5, illuminating the importance of selecting an appropriate embeddings model for downstream effectiveness.

3.3 Reranking

ColBERT is the only reranker selected for this project. Figure 2 depicts the general architecture for ColBERT where, given a query *q* and document *d*, a query encoder f_Q encodes *q* into a bag of fix-sized embeddings E_q, while a document encoder f_D encodes *d* into another bag E_d. The relevance score for E_q and E_d is computed by summing the maximum similarity (MaxSim) operators.

[2] https://huggingface.co/spaces/mteb/leaderboard

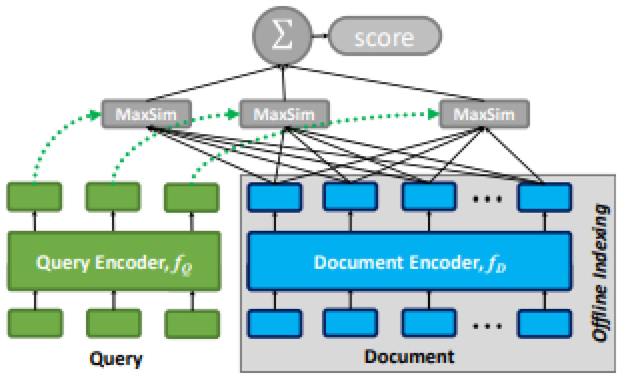


Figure 2: ColBERT architecture (Khattab and Zaharia 2020)

Dense retrieval methods often perform well on academic benchmarks because they are specifically trained for these tasks. ColBERT, unlike RRF which requires weight tuning for performance, acts as an independent algorithm with an affinity for keywords that can complement first-stage candidate generation. Hardcoded thresholds based on normalized distributions of correct answers were considered for relevance scores when combining and reranking, but decided against given these values could change if the underlying embeddings model is replaced. Additionally, varying context sizes can produce unpredictable spreads in relevance values making comparison difficult.

3.4 Testing Data

We assess retrieval performance with a synthetic dataset generated by GPT-4 of queries resembling queries describing businesses. Queries were created by providing GPT-4 with a 2022 NAICS code and a topic from which it could generate a plausible business description through parametric memory. This was performed for a total of 2,323 unique inquiries representing 48.32% of all NAICS codes. Table 3 displays the most frequently seen codes in the synthetic dataset. During data generation, we noticed many descriptions that could also apply to other codes when described ambiguously. While our prompt remained generally relaxed to source diverse language and descriptions, it is possible some query-code pairs are less than optimal. However, this was an acceptable risk for the purposes of bootstrapping search-engine compatible data for development.

Table 3: Most Frequently seen NAICS codes in Synthetic Dataset

Count	**Code**	**Title**	**Sector**
63	722511	Full-Service Restaurants	Accommodation and Food Services
43	813110	Religious Organizations	Other Services (except Public Administration)
42	541110	Offices of Lawyers	Professional, Scientific, and Technical Services
39	531210	Offices of Real Estate Agents and Brokers	Real Estate and Rental and Leasing
36	524210	Insurance Agencies and Brokerages	Finance and Insurance
36	621210	Offices of Dentists	Health Care and Social Assistance
33	236118	Residential Remodelers	Construction
31	561730	Landscaping Services	Administrative and Support and Waste Management and Remediation Services.
30	722513	Limited-Service Restaurants	Accommodation and Food Services

29	621111	Offices of Physicians (except Mental Health Specialists)	Health Care and Social Assistance

4. Experiments

Each of the five research questions are addressed in sequential order where the results of one experiment inform the next to reduce the scope of analysis and establish key mechanisms in an effective multi-stage retrieval pipeline:

1. First-Candidate Generation (Dense vs. Sparse)
2. Combinations of Context Windows in Ensemble Approaches
3. Preparing for Reranking: Identifying Improvements Beyond *k*=5 Documents
4. Reranking with ColBERT
5. Caching for Edge Cases

Test 1 addresses RQ1 and establishes how much better dense retrieval performs over sparse on each context window. Test 2 explores RQ2 and reports accuracy at *Hit@*1 and *Hit@*5 for all combinations of context windows. From these tests, we generate hypotheses about which combinations are useful, narrowing down to two for future experiments. Test 3 supports RQ3 by analyzing which approach generates reasonable increases in accuracy when given additional *k* documents, providing the best field of first-generation candidates for reranking. Within this analysis we use the two ensemble approaches selected from the previous test and apply two pruning approaches: Lazy and Greedy. Lazy lets each context window in the ensemble produce the top five documents regardless of whether a previous context window also generated a document with an identical code. The final *k* documents are sorted by document relevance score. Greedy forces each context window to produce only unique codes, replacing documents already collected only when the relevancy score for a document is higher. Test 4 addresses RQ4 by applying ColBERT as a reranker when given initial *k* documents ranging from 6-10. We use three unique reranking context windows to identify the best approach. We also keep a close eye on inference speed. Test 5 answers RQ5 by taking the wrong predictions from the best performer across previous experiments and using the cache alone to predict NAICS codes.

4.1 First-Candidate Generation (Dense vs. Sparse)

Table 4: Dense vs. Sparse Retrieval

Retrieval	Context Window	Collection Size	Hit@1	Hit@5	Speed (ms)
Dense (Semantic)	CR	3046	55.92	73.4	13.28
	D	1609	45.11	73.01	12.47
	IE	2019	24.84	35.34	12.51
	IK	20373	55.83	74.6	16.1
	T	1012	**59.66**	80.93	12.79
	T-D	1012	57.12	83.08	12.56
	T-D-IE	1012	57.73	**83.47**	13.04
Sparse (BM25)	CR	3046	16.75	30.91	5.05
	D	1609	15.5	33.28	2.99
	IE	2019	6.67	13.56	3.43
	IK	20373	15.63	33.02	26.91
	T	1012	0.39	0.99	2
	T-D	1012	17.91	36.2	2.28
	T-D-IE	1012	20.71	40.03	2.19

Dense retrieval significantly outperforms sparse across every context window. The largest context window *T-D-IE* outperforms others at 83.47%. Both *T* and *T-D* also achieve strong results suggesting the common feature across all three—a document's title—provides dense retrievers with strong signal. Notably, this pattern does not apply to the BM25 results where *T* scores just 0.99%, nearly 80% lower than semantic search. Shorter documents constrain word overlap-based retrieval strategies, particularly when language is attribute-centric.

Poor performance for *IE* in semantic search is likely a result of the many codes without associated illustrative examples. Additionally, we note the remarkably low inference speed for each context window in dense retrieval. Even *IK* for which BM25 took 0.2 seconds per query was faster in dense retrieval due to pre-indexing with a vector database. This is a crucial element of semantic search with LLMs: reliable retrieval speed even as the size of a corpus scales.

4.2 Combinations of Context Windows in Ensemble Approaches

Table 5: Top 10 Performers in Lazy Retrieval

Contexts	Overlapping Contexts	Complementary Contexts	Hit@1	Hit@5	Speed (sec)
6	[T, T-D, T-D-IE, IE]	[IK, CR]	63.32	86.87	0.07
5	[T, T-D-IE, IE]	[IK, CR]	63.19	86.78	0.06
5	[T, T-D, IE]	[IK, CR]	63.07	86.78	0.06
4	[T, T-D-IE, IE]	[CR]	61.73	86.78	0.06
5	[T, T-D, T-D-IE]	[IK, CR]	63.02	86.74	0.07
4	[T, T-D, T-D-IE]	[IK]	62.25	86.74	0.05
3	[T, T-D-IE]	[IK]	61.90	86.74	0.04
7	[T, T-D, T-D-IE, D, IE]	[IK, CR]	63.02	86.70	0.09
6	[T, T-D, D, IE]	[IK, CR]	62.81	86.66	0.07
5	[T, T-D, T-D-IE, IE]	[CR]	61.64	86.66	0.07

Table 6: Top 10 Performers in Greedy Retrieval

Contexts	Overlapping Contexts	Complementary Contexts	Hit@1	Hit@5	Speed (sec)
4	[T-D-IE, D, IE]	[CR]	58.98	85.97	0.02
2	[T-D-IE]	[CR]	58.98	85.97	0.02
5	[T-D-IE, D, IE]	[IK, CR]	58.98	85.97	0.02
3	[T-D-IE, D]	[CR]	58.98	85.97	0.02
3	[T-D-IE, IE]	[CR]	58.98	85.97	0.02
3	[T-D-IE, D, IE]	[]	58.98	85.97	0.02
3	[T-D-IE]	[IK, CR]	58.98	85.97	0.02
4	[T-D-IE, D, IE]	[IK]	58.98	85.97	0.02
4	[T-D-IE, IE]	[IK, CR]	58.98	85.97	0.02
2	[T-D-IE, IE]	[]	58.98	85.97	0.02

Table 5 and 6 display best performing ensembles sorted by *Hit@5*. Lazy retrieval appears to generate higher scores than Greedy. Moreover, Greedy's top 10 performers all posted the same score, suggesting a dominant context window in *T-D-IE*—the only context window present in each ensemble. We observe here that by generating more diverse options in a Greedy retrieval, accuracy scores underperform.

Regarding context windows, *T-D-IE* is used in 8 of the Lazy ensembles and all 10 of the Greedy, making it the most frequently seen context window and the strongest predictor of performance. *IK* and *CR* are also frequently used appearing 13 times and 14 times respectively across both retrieval strategies. However, it is difficult to discern whether they are adding any value given there are two ensembles in the Greedy Retrieval approach where neither are used and yet accuracy remains the same. This contradicts what one would assume about varying context window lengths. With *IK* and *CR* consisting of a couple of tokens and at most a sentence, they would yield higher relevance scores than *T-D-IE*. It's possible that they produce correct predictions when used, but when absent, *T-D-IE* can reliably close the gap.

Lastly, inference speed remains remarkably low despite additional context windows. With every additional context window, inference appears to slow by approximately 0.01 seconds. From Test 2, we identify two ensemble approaches that encompass the best performers in Table 9. Because no single ensemble approach generates strong performances across both retrieval strategies, we select two strong performers and look for additional clues in Test 3.

Table 9: Ensembles Selected from Test 2 Results

Context Windows	**New Name**
[T, T-D, T-D-IE, IE, IK, CR]	Comprehensive Overlapping, Lazy
[T-D-IE, IK, CR]	Comprehensive Complementary, Lazy

4.3 Preparing for Reranking: Identifying Improvements Beyond *k*=5 Documents

Comprehensive Overlapping, Lazy achieves the highest accuracy scores on the synthetic dataset at every position *k* from 5 to 10. The resulting average is the highest of any approach and is selected to continue for the following experiment.

On Lazy vs. Greedy approaches, there are a few key takeaways. Because lazy generates many documents with the same code across context windows, we observe higher accuracy scores at smaller *k* positions (explaining Test 2's results), but performance plateaus at higher positions without additional unique documents. Conversely, greedy is forced to generate unique codes which provides marginal increases in performance for each additional document.

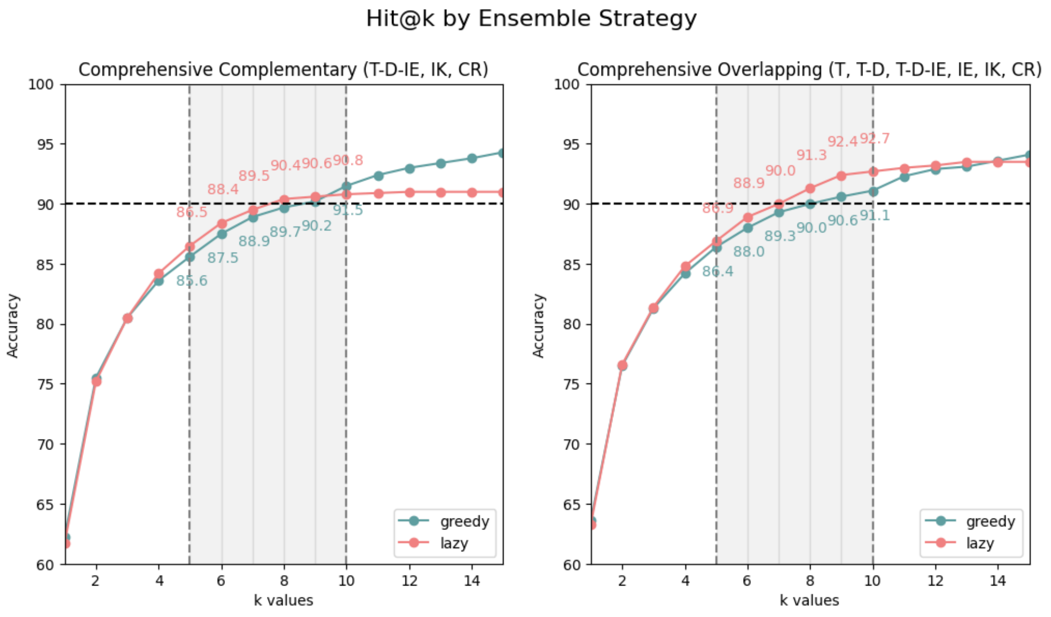


Figure 6: Accuracy as *k* Increases for Each Retrieval Strategy

4.4 Reranking

ColBERT is largely unsuccessful in improving accuracy when given *initial k* documents of any number, likely generating more false positives. The one exception is using *T-D-IE* as a reranking context improving overall performance by just 0.13% when given *k*=6 initial documents, an improvement that likely does not justify the use of a reranker given a sizable increase in inference speed. The worst performer among reranking contexts is the *S1 – Doc* where ColBERT is comparing documents across context windows without any understanding of relevance scores, implying small context windows like *IK* are being compared with large ones such as *T-D-IE*.

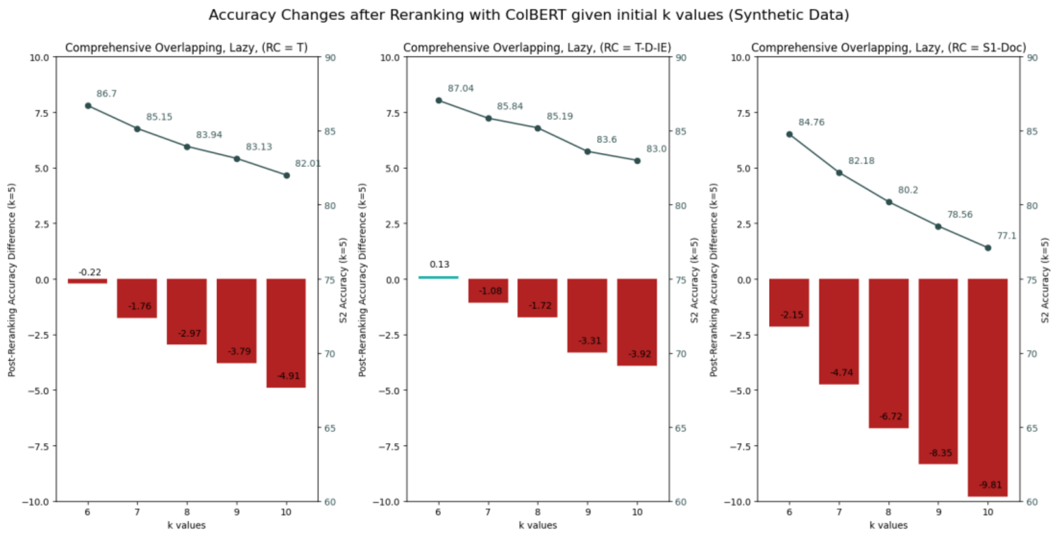
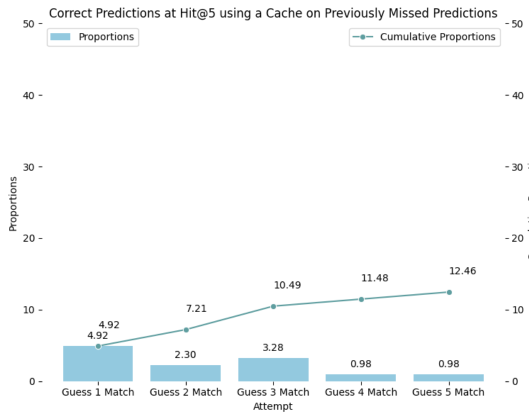
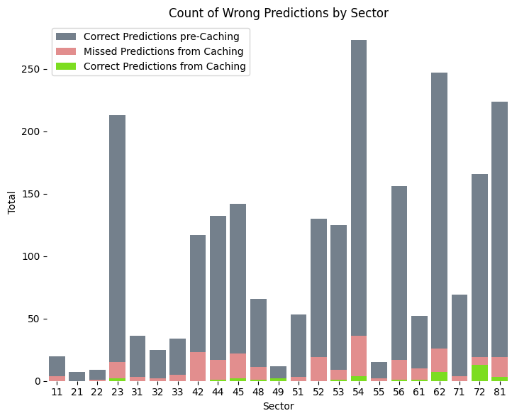


Figure 7: Accuracy Changes after Reranking with ColBERT given initial *k* values on both datasets

4.5 Caching for Edge Cases





Figures 8-9: (Left) Correct Predictions at *Hit@5* using a Cache on Previously Missed Predictions and (Right) Count of Correct and Wrong Predictions by Sector

Figure 8 displays the proportion (and accumulative proportion) of correct predictions using the cache (N=679) on those previously missed by our best performing semantic retriever. Of the remaining 305 queries missed by semantic retrieval, the cache correctly predicts 12.46% at *Hit@5*, justifying the use of a cache in addressing edge cases. Figure 9 displays the sectors where caching improve performance the most with strong performances in sectors 62 (Health Care and Social Assistance) and 72 (Accommodation and Food Services).

Providing a way of augmenting data over time will help address sectors and business classifications where the existing corpus cannot infer semantic connections. Additionally, caching provides a systematic way of identifying sectors that may warrant review for more granular subcategories as areas of the economy evolve and grow more sophisticated.

5. Conclusion

This paper introduces an alternative design for a NAICS search engine with semantic retrieval models achieving a final accuracy of 86.87% on synthetic test data. Our research delivers critical findings on building an effective multi-stage retrieval pipeline. Dense retrieval models provide a stronger selection of first-stage candidates over sparse retrieval models by relying on contextual similarity to overcome vocabulary mismatches. Additionally, combining multiple context windows expanded non-parametric memory with limited increases in inference speed, and multiple context windows associated with each code generated higher accuracy scores in early positions *k*. We observed a fast complementary reranker such as ColBERT was insufficient in boosting performance after dense retrieval, suggesting additional strategies on reranking are needed for state-of-the-art retrieval pipelines. Lastly, we proved seen-before queries can address edge cases and provide a path forward for improvement when the corpus fails. These findings offer an improved NAICS search experience and open new research directions for similar information retrieval tasks.

The findings from this study should also encourage new research opportunities on the use of LLMs for other critical services across the Bureau. The phases of experimentation in building the search engine—comparing context retrieval windows independently and within ensemble approaches, exploring embeddings models and their influence on downstream performance, reranking strategies, latency limitations, using generative models to clean or alter messy data—bear relevance on other applications beyond search. Summarization, classification, named entity recognition, and translation are among the tasks involved in the building of this project and are frequently used in services across the Bureau. Moreover, deploying LLMs and updating critical government services is a key mandate within the Federal government, motivated in part by a recent Executive Order[3] encouraging the exploration of AI technologies. By adopting new technologies such as a semantic retrieval-based NAICS search engine, the U.S. Census Bureau can continue standard of excellence and technical prowess within the Federal government.

References

Asadi, Nima, and Jimmy Lin. 2013. "Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures." *InProceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* 997-1000.

Berger, Adam, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. "Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding." *InProceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* 192-199.

Borgeaud, Sebastian, Arthur Mensch, Jordan Hoffman, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, et al. 2022. "Improving Language Models by Retrieving from Trillions of Tokens." *InInternational conference on machine learning* 2206-2240.

Chen, Ruey-Cheng, Luke Gallagher, Roi Blanco, and J. Shane Culpepper. 2017. "Efficient cost-aware cascade ranking in multi-stage retrieval." *InProceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* 445-454.

Cormack, G.V., C.L.A. Clarke, and Stefan Buttcher. 2009. "Reciprocal Rank Fusion outperforms Condorcet and individual Rank Learning Methods." *InProceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* 758-759.

[3] https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/

Gao, Luyu, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. "Complementing Lexical Retrieval with Semantic Residual Embedding." *InAdvances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43* 146-160.

Gillick, Daniel, Allessandro Presta, and Gaurav Singh Tomar. 2018. "End-to-End Retrieval in Continuous Space." *arXiv preprint arXiv:1811.08008.*

Gong, Hongyu, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. "Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension." *arXiv preprint arXiv:2005.08056.*

Guo, Mandy, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2020. "MultiReQA: A Cross-Domain Evaluation for Retrieval Question Answering Models." *InProceedings of the Second Workshop on Domain Adaptation for NLP 2021.*

Izacard, Gautier, and Edouard Grave. 2020. "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering." *arXiv preprint arXiv:2007.01282.*

Kandpal, Nikhil, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. "Large Language Models Struggle to Learn Long-Tail Knowledge." *Proceedings of the 40th International Conference on Machine Learning* 15696-15707.

Karpukhin, Vladimir, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. "Dense Passage Retrieval for Open-Domain Question Answering." *arXiv preprint arXiv:2004.04906.*

Khattab, Omar, and Matei Zaharia. 2020. "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT." *InProceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval* 39-48.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, et al. 2020. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." *Advances in Neural Information Processing Systems.* 33: 9458-74.

Li, Zehan, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. "Towards General Text Embeddings with Multi-stage Contrastive Learning." *arXiv preprint arXiv:2308.03281.*

Liang, Davis, Peng Xu, Siamak Shakeri, Cicero Noguiera dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. "Embedding-based Zero-shot Retrieval through Query Generation." *arXiv preprint arXiv:2009.10270.*

Lin, Jimmy, Rodrigo Nogueira, and Andrew Yates. 2021. "Pretrained Transformers for Text Ranking: BERT and Beyond." *InProceedings of the 14th ACM International Conference on web search and data mining* 1154-1156.

Liu, Jiachang, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. "What Makes Good In-Context Examples for GPT-3?" *arXiv preprint arXiv:2101.06804.*

Liu, Shichen, Fei Xiao, Wenwu Ou, and Luo Si. 2017. "Cascade ranking for operational e-commerce search." *In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* Halifax, Nova Scotia, Canada. 1557-1565.

Luan, Yi, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. "Sparse, Dense, and Attentional Representations for Text Retrieval." *Transactions of the Association for Computational Linguistics.*

Ma, Ji, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. "Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation." *arXiv preprint arXiv:2004.14503.*

Mackenzie, Joel, Shane Culpepper, Roi Blanco, Matt Crane, Charles Clarke, and Jimmy Lin. 2018. " Query driven algorithm selection in early stage retrieval." *InProceedings of the Eleventh ACM International Conference on Web Search and Data Mining* 396-404.

Majumder, Goutam, Partha Pakray, Ranjita Das, and David Pinto. 2021. "Interpretable semantic textual similarity of sentences using alignment of chunks with classification and regression." *Applied Intelligence* 51: 7322-7349.

Matveeva, Irina, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. "High accuracy retrieval with multiple nested ranker." *InProceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* 437-444.

Neelakantan, Arvind, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, et al. 2022. "Text and Code Embeddings by Contrastive Pre-Training." *arXiv preprint arXiv:2201.10005.*

Pedersen, Jan. 2010. "Query understanding at Bing." *In Industry Track Keynote at the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* Geneva, Switzerland.

Pradeep, Ronak, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. "RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models." *arXiv preprint arXiv:2309.15088.*

Pradeep, Ronak, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. "RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze!" *arXiv preprint arXiv:2312.02724.*

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, and Peter Liu. 2020. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." *The Journal of Machine Learning Research* 21 (1): 5485-5551.

Sun, Weiwei, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents." *arXiv preprint arXiv:2304.09542.*

Thakur, Nandan, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models." *arXiv preprint arXiv:2104.08663.*

Tunstall, Lewis, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, et al. 2023. "Zephyr: Direct Distillation of LM Alignment." *arXiv preprint arXiv:2310.16944.*

United States Census Bureau. 2019. *North American Industry Classification System.* Accessed March 2, 2024. https://web.archive.org/web/20190305072906/https:/www.census.gov/eos/www/naics/faqs/faqs.html.

Wang, Lidan, Jimmy Lin, and Donald Metzler. 2011. "A cascade ranking model for efficient ranked retrieval." *InProceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* 105-114.

Zaheer, Manzil, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, et al. 2020. "Big Bird: Transformers for Longer Sequences." *Advances in neural information processing systems* 33: 17283-97.

Zhao, Wayne Xin, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. "Dense Text Retrieval Based on Pretrained Language Models: A Survey." *ACM Transactions on Information Systems* 42: 1-60.