Comparative study of accuracy and precision of Information Retrieval from HTML Compared to JSON-LD

Matt Briggs <https://www.mattbriggs.us>

Abstract

JSON-LD performed 3.6x as well for precision and accuracy in information retrieval as HTML.

RAG (Retrieval-Augmented Generation) and LLMs (Large Language Models) in chat and agent-based systems rely on information retrieval (IR) to support their responses by grounding them in external, up-to-date, or domain-specific knowledge. Most of the focus on improving the accuracy and precision of these systems focus on post-processing of information ingested into the AI pipeline. However, the disposition of the source of this information may have an impact on the accuracy and precision of these systems.

This study uses the same evaluation criteria, the same vector-based pipeline, and the same IR mechanism between two sets of data prepared from the same content. The Microsoft documentation platform, Learn.microsoft.com, has published 582 articles, each article from a single source of content. A set of frequently asked questions is published to both HTML and JSON-LD (JavaScript Object Notation for Linked Data) targets.

The articles were downloaded into two sets, HTML and JSON-LD. This content was used to create a set of golden questions and answers. The HTML and the JSON-LD were then ingested into two pipelines that only varied by the incoming source format: HTML and JSON-LD. The content was ingested, chunked, embedded, indexed, and then used to score using an F1 score. The F1 Score is a harmonic measure of the measure of *precision* and *accuracy* that varies from 0 (not able to answer any questions) to 1 (full-fidelity retrieval). A score of zero is unretrievable information. A score of one is an accuracy that can be seen by a database.

In this study, HTML scored mean of ~**0.28**. The JSON-LD's mean was ~**0.99**.

HTML represents an unstructured content workflow such as a storage format of Markdown to transformed to HTML. HTML was 33% of JSON-LD. While information can be retrieved from HTML, the format introduced retrieval inefficiencies, noise, inaccuracies, and unpredictable responses.

JSON represents a structured content workflow such as a storage format of YAML to transformed to JSON-LD. The JSON-LD had an F1 score of 360% greater than the HTML, demonstrating a profound impact on vector-based retrieval accuracy. JSON-LD performed about as well as retrieval a query written to a SQL database.

The study data and code used to create the study can be found at the end of this paper so that the results can be replicated.

1. Introduction

IR techniques, such as RAG and agent-based systems, enhance AI models by incorporating external knowledge. These techniques, powered by vector-based retrieval pipelines, improve factual accuracy, reduce hallucinations, and enable up-to-date responses. A comparison of structured content formats, JSON-LD, with unstructured formats, HTML, reveals the impact on retrieval effectiveness, influencing accessibility, search efficiency, and AI retrieval performance. Using a format optimized for information retrieval, improves the quality and effectiveness of AI experiences.

1.1 Background and motivation

AI workloads enhance, rather than modify, the underlying models such as OpenAI GPT-3.5 and GPT-4 by using external knowledge retrieval techniques. Pre-trained models rely on static training data; they cannot update their knowledge after their training. To address this limit, AI workloads use IR-based approaches to load more current, contextual, and relevant information during inference.

Key techniques for enhancing AI models

AI models benefit from several retrieval and augmentation techniques that extend their capabilities beyond their original training data. IR is a key element of these techniques to provide current and accurate information.

IR is the process of obtaining relevant information from a large collection of data, often in response to a user's query. IR is widely used in search engines, recommendation systems, and knowledge retrieval applications. IR systems rank documents, or other information units, by their relevance to a given query. These systems rely on indexing, ranking algorithms, and relevance scoring to deliver useful results.

For AI, IR plays an important role in the following key techniques:

1. RAG

2. Large Language Models (LLMs) in Conversational AI
3. Agent-Based Systems for Complex Interactions

1. RAG

RAG is a framework that improves AI-generated responses by getting relevant information before generating text. Instead of relying on just the model's pre-trained knowledge, RAG pulls information from external structured and unstructured data sources to improve factual accuracy.

How it works

- The system receives a query.
- It retrieves relevant documents or data from external sources (for example, databases, APIs, document repositories).
- The retrieved information is passed to the language model as context before generating a response.

Use cases

- Keep AI responses up-to-date with recent events, regulations, product updates.
- Enhance responses with domain-specific knowledge from proprietary information sources.
- Reduce hallucinations (incorrect AI-generated information) by grounding responses in factual sources.

2. Large language models (LLMs) in conversational AI

LLMs, such as GPT-based systems, power conversational AI applications. While they generate responses based on their training data, they are not inherently retrieval-based. To improve their accuracy and avoid outdated responses, LLMs can be combined with RAG-based retrieval systems, ensuring that responses remain aligned with current and external knowledge.

3. Agent-based systems for complex interactions

Agent-based AI systems take retrieval a step further by orchestrating multiple LLM calls, memory, and external tool usage. These systems retrieve, process, and refine responses dynamically by interacting with APIs, databases, or other external services.

How they work

- The agent analyzes a query and determines if external data is needed.
- It calls external APIs, databases, or IR mechanisms to get information.
- It refines its response iteratively, sometimes using self-reflection techniques.

Use cases

- Customer support bots that look up real-time event logs, account details, or transaction statuses.
- Automated help agents assistants that compile updated summaries from multiple sources.
- Autonomous task agents that execute multi-step processes, such as prepare lesson plans or coordinate a student's progress in a course.

Processing and indexing pipelines for content and data

To support IR-based AI, systems must process large amounts of structured and unstructured data efficiently. A typical retrieval pipeline follows these steps:

1. **Data harvesting**: The system collects data from structured (databases, structured document stores [JSON], APIs) and unstructured (documents [text, HTML], articles, feedback) sources.
2. **Chunking**: Large documents are broken into smaller sections to optimize retrieval accuracy.
3. **Embedding**: The chunks are embedded. Embeddings are vector representations of data (for example, words, documents, or images) in a high-dimensional space, where similar items are placed closer together.
    - The text chunks are converted into vector embeddings (numerical representations of meaning).
    - These embeddings are indexed in a vector database for fast semantic search.
4. **Indexing:** An index is a structured data structure that allows for efficient searching and retrieval of information. It acts as a lookup table that maps queries to relevant documents.
    - Indexing strategies can leverage both *supervised* and *unsupervised* methods.
    - A supervised index is one that has been optimized using labeled data, often with human-labeled training sets to improve IR.
    - Unsupervised methods use machine learning algorithms to automatically categorize and structure information within an index without human-labeled training data.

    How do embeddings and indexes work together?

    Embeddings and indexes work together to enable efficient and accurate IR. Embeddings convert text, images, or other data into high-dimensional vector

representations, capturing semantic meaning. These vectors allow for similarity-based search rather than relying on exact keyword matches.

An index organizes and optimizes retrieval by storing these embeddings in a structured format, enabling fast lookup. In vector-based search, when a user submits a query, it is converted into an embedding and compared against the indexed embeddings using Approximate Nearest Neighbor (ANN) algorithms (for example, FAISS, Annoy). The system retrieves the closest matches based on cosine similarity or Euclidean distance.

In hybrid search, traditional keyword-based indexing (for example, inverted index) is combined with embedding-based retrieval for more precise and context-aware results. This approach powers modern search engines, AI assistants, and recommendation systems, enhancing accuracy and relevance beyond keyword-based methods.

5. **Storage in a vector database**: Instead of storing raw text, the system stores vector embeddings, allowing similarity-based retrieval.
6. **Retrieval at query time**: When a user submits a query, the system:
    - Searches the vector database for semantically similar chunks.
    - Retrieves relevant chunks and feeds them into the AI model for contextual response generation.
7. **Response generation**: The AI integrates the retrieved data into its answer, ensuring accuracy and contextual relevance.

Why this matters

These retrieval mechanisms update AI models without requiring retraining, making them more adaptable and up-to-date. Correct, precise, and current information, especially in technical documentation and fast moving disciplines, enable AI experiences to be current, relevant, and accurate.

1.2 Why is the comparison of structured vs unstructured source format important?

This study compares two different type of information sources, structured content and unstructured content, and feeds this content into the same vector-based IR pipeline. The incoming content, the vectorization, retrieval, and scoring of the precision and accuracy of retrieval were the same. The only variable between the two approaches is the source.

Unstructured in this case is represented by **HTML**. Markdown content is often published as HTML. HTML in so far as it is structed provides instructions to the Web browser for how to present the information.

Structured content in this case is represented by **JSON-LD**. It is a structured content because it adheres to a well-defined schema and provides machine-readable data in a consistent format. When JSON-LD follows a Schema.org schema, it organizes data in a way that search engines, knowledge graphs, and AI systems can easily process.

The comparison is critical because it evaluates how different content formats impact IR effectiveness. The choice of format affects how AI models and retrieval systems interpret and process content, directly influencing accuracy, efficiency, and user experience.

Key reasons why this study matters

Content accessibility and structure play a crucial role in web development. HTML is optimized for human readability but lacks explicit semantic structure, which can sometimes hinder the effective organization and understanding of web content.

In contrast, JSON-LD is machine-readable, embedding structured metadata that can significantly improve retrieval accuracy by providing clear and usable data for search engines and other automated systems. By using these technologies effectively, content teams can enhance both the accessibility and structure of their content, ensuring it is both human-friendly and machine-compatible.

The impact on search and AI retrieval is notably significant. HTML-based retrieval relies on natural language parsing techniques, which are essential for extracting meaningful text chunks. On the other hand, JSON-LD-based retrieval enables the use of structured queries, which can potentially improve precision.

In addition, content that is optimization for a knowledge graph is crucial in enhancing the efficiency of AI retrieval systems. Many modern AI retrieval systems, such as Weaviate – a vector database used in this study -- rely heavily on vector search and the use of knowledge graphs to improve their functionality. By adhering to Schema.org standards, JSON-LD can significantly enhance retrieval efficiency. Schema.org is a collaborative, community-driven vocabulary that provides a standardized framework for structuring metadata on web pages to improve search engine understanding and interoperability. This improvement is achieved by advancing entity recognition and relationship modeling, which are fundamental components of knowledge graph optimization.

Understanding which format provides better retrieval accuracy helps optimize various AI-driven technologies. By identifying the most effective formats, content teams can enhance the performance of search engines, chatbots, and recommendation systems. Additionally, this knowledge informs best practices for preparing content effectively in enterprise documentation, FAQs, and AI-driven knowledge bases, ensuring that information is both accessible and useful.

2. Field of study and relevance of F1 scores

This study falls within the fields of Information Retrieval (IR) and Natural Language Processing (NLP), specifically focusing on vector-based search and structured data representation.

Field of study

- **IR**: The process of retrieving relevant data from large document collections based on user queries.
- **NLP**: AI-driven techniques for extracting and interpreting meaning from text.
- **Semantic search and knowledge graphs**: Enhancing AI systems by using structured data (like JSON-LD) to improve query precision and understanding.

Relevance of F1 scores

The F1-score is a key evaluation metric in retrieval systems, balancing precision and recall. Precision refers to how many of the retrieved results are relevant, while recall indicates how many of the relevant results were actually retrieved. The F1-score is calculated as the harmonic mean of precision and recall, providing a balanced measure of the retrieval system's effectiveness.

Why the F1-score is critical in this study

The F1-score is critical in this study because it serves as a vital measurement of the balance between precision and recall in IR systems. If HTML retrieval achieves a high recall but low precision, it risks returning an overwhelming amount of irrelevant information. Conversely, JSON-LD retrieval can improve precision by structuring data semantically, thereby enhancing the relevance of answers without sacrificing recall. Ultimately, a higher F1-score indicates a better equilibrium, meaning the retrieval system is efficiently returning the right information.

This study is highly relevant to improving AI-driven FAQ retrieval. By comparing HTML and JSON-LD using F1-score analysis, it helps determine which format provides higher retrieval accuracy, ensuring better search experiences for users.

1.2 Research question and objectives

The study seeks to answer the following question:

Does the document format, unstructured (HTML) or structured (JSON) influence information retrieval performance in an AI system?

The purpose of the study is to isolate the variables used in creating vector-based IR systems. The study aims to determine how the format of the files (HTML vs. JSON-LD) affects retrieval accuracy. The content comes from a single source, which is the FAQ content on Learn, and the only differentiation is the format: HTML and JSON-LD (using Schema.org FAQPage schema).

Common variables (controlled)

To ensure that the format is the only variable being tested, the following factors are kept constant across both methods:

- **Source data**

  Both HTML and JSON-LD formats are generated from the same YAML source file, ensuring content parity.

- **Data ingestion process**

  Both formats are indexed into Weaviate using the same ingestion process.

  Weaviate is an open-source vector database designed for storing, indexing, and querying unstructured data using machine learning models. It specializes in semantic search and RAG by leveraging vector embeddings from text, images, and other data types.

- **Question set**

  The same set of golden questions is used for both formats, with duplicates removed to prevent bias.

- **Embedding & vectorization**

  Both formats use OpenAI's text2vec model for embeddings.

- **Retrieval process**

  Both formats utilize semantic search via Weaviate.

- **Evaluation metric**

  Both formats are evaluated using the F1 score.

By keeping these factors constant, the study isolates the content format as the only independent variable, ensuring that any differences in retrieval performance can be attributed to the format itself.

Differences between HTML and JSON-LD pipelines

HTML format extracts unstructured text with applies a chunking and indexing routine to the content from an external source. JSON-LD format offers structured, machine-readable triples, preserving context and logical relationships that is native to the content.

HTML format

- **Format Type**: Unstructured text extracted from rendered HTML pages (paragraphs, headers, etc.).
- **Storage Format**: Plain text stored as paragraphs and headers.
- **Vectorization**: Text is extracted and converted into embeddings.
- **Chunking**: Externally exerted chunking on HTML parsing (scrapping)
- **Indexing**: Externally derived tags associated with the chunks.

JSON-LD format

- **Format Type**: Structured, machine-readable JSON-LD representations based on Schema.org.
- **Storage Format**: Well-structured triples (subject-predicate-object).
- **Vectorization**: Directly structured and fed into vector search.
- **Chunking**: Structured fields, preserving logical relationships.
- **Indexing**: Explicit indexes declared in the content structure.

Isolation of the format variable

The study ensures that the format variable is properly isolated by maintaining all other factors constant. This means that the only difference between the two methods is the content format (HTML vs. JSON-LD).

2. Related work

Existing literature on RAG pipelines focuses on post-processing routines with some innovative approaches to exert structure on unstructured content sources. For example, Microsoft developed the GraphRAG technique, which expands the RAG pipeline with an ad hoc knowledge graph and unsupervised index. There are few papers that look at the interaction of structure and unstructured content sources in AI workloads.

Integrated Retrieval over Structured and Unstructured Data

This study investigates different strategies for combining retrievals over structured and unstructured data. It compares parallel combination, unstructured-structured serial combination, and structured-unstructured serial combination strategies, showing that combined approaches can outperform traditional unstructured retrieval.

https://ceur-ws.org/Vol-1178/CLEF2012wn-INEX-WangEt2012b.pdf

A Study on Information Retrieval Methods in Text Mining

This paper explores various IR methods in text mining, including the use of structured and unstructured content. It highlights the advantages of structured content in providing clear and unambiguous semantics.

https://www.ijert.org/research/a-study-on-information-retrieval-methods-in-text-mining-IJERTCONV2IS15028.pdf

On the Integration of Structured Data and Text: A Review of the SIRE Architecture

This review discusses the integration of structured data and text in relational database management systems (RDBMS), emphasizing the benefits of combining structured and unstructured data for more intelligent search.

https://ir.cs.georgetown.edu/downloads/SIRE-Architecture.pdf

3. Methods

The study compares HTML and JSON-LD content formats for information retrieval, isolating format as the only variable. The F1 score is used as the evaluation metric, balancing precision and recall and reflecting real-world performance. The study utilizes datasets derived from the same YAML source file, ensuring content parity and enabling direct comparison of retrieval accuracy between unstructured and structured formats.

3.1 Study design and experimental framework

In the study, the controlled conditions ensure that the only variable being tested is the content format (HTML vs. JSON-LD). Here are the controlled conditions and how variables are isolated:

- **Source Data**: Both HTML and JSON-LD formats are generated from the same YAML source file, ensuring content parity.
- **Data Ingestion Process**: Both formats are indexed into Weaviate using the same ingestion process.
- **Question Set**: The same set of golden questions is used for both formats, with duplicates removed to prevent bias.
- **Embedding & Vectorization**: Both formats use OpenAI's text2vec model for embeddings.
- **Retrieval Process**: Both formats utilize semantic search via Weaviate.
- **Evaluation Metric**: Both formats are evaluated using the F1 score.

By keeping these factors constant, the study isolates the content format as the only independent variable, ensuring that any differences in retrieval performance can be attributed to the format itself.

The F1 score is used as the harmonic measure in the study for several reasons:

- **Balance between precision and recall**: The F1 score provides a balanced measure that considers both precision (the accuracy of the retrieved responses) and recall (the completeness of the retrieved responses). This balance is crucial for evaluating the effectiveness of IR systems.
- **Handling imbalanced data**: In scenarios where the number of relevant documents is much smaller than the number of irrelevant ones, the F1 score is particularly useful as it gives a more comprehensive evaluation than accuracy alone.
- **Standard metric in IR**: The F1 score is a widely accepted and standard metric in the field of IR. It allows for direct comparison with other studies and systems that use the same metric.
- **Reflects real-world performance**: By combining precision and recall, the F1 score reflects the real-world performance of the retrieval system, ensuring that both the relevance and completeness of the retrieved responses are considered.

3.2 Data collection

The study utilizes two primary datasets derived from the same YAML source file to ensure content parity. The files were identified from the inventory of content published to Learn.Microsoft.com by content that was the FAQ content type. Content that had a YAML source, an HTML and a JSON-LD payload were downloaded into two collections separated by format.

The study leverages two datasets—HTML and JSON-LD—derived from the same YAML source to ensure content parity. The HTML dataset involves text extraction and chunking, while the JSON-LD dataset focuses on structured data representation. Both datasets are vectorized using OpenAI's **text2vec** model and ingested into Weaviate for semantic search. This approach allows for a direct comparison of retrieval accuracy between unstructured and structured content formats.

3.3 Evaluation metrics and comparative methodology

The F1 score is a harmonic mean of precision and recall, providing a single metric that balances both. It is particularly useful when the dataset has an uneven class distribution. The F1 score is defined as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where:

- **Precision** is the ratio of correctly retrieved relevant documents to the total number of retrieved documents:

$$Precision = \frac{True\ Positives}{True\ Posities + False\ Positives}$$

- **Recall** is the ratio of correctly retrieved relevant documents to the total number of relevant documents:

$$Recall = \frac{True\ Positives}{true + False\ Negatives}$$

The F1 score ranges from 0 to 1, where 1 indicates perfect precision and recall, and 0 indicates the worst performance.

4. Results and discussion

JSON-LD outperforms HTML in F1 scores, with a mean score of ~0.99 compared to ~0.28 for HTML.

The study investigates the accuracy of storing and retrieving HTML-based FAQs in Weaviate, comparing two content formats: HTML and JSON-LD. The main findings are:

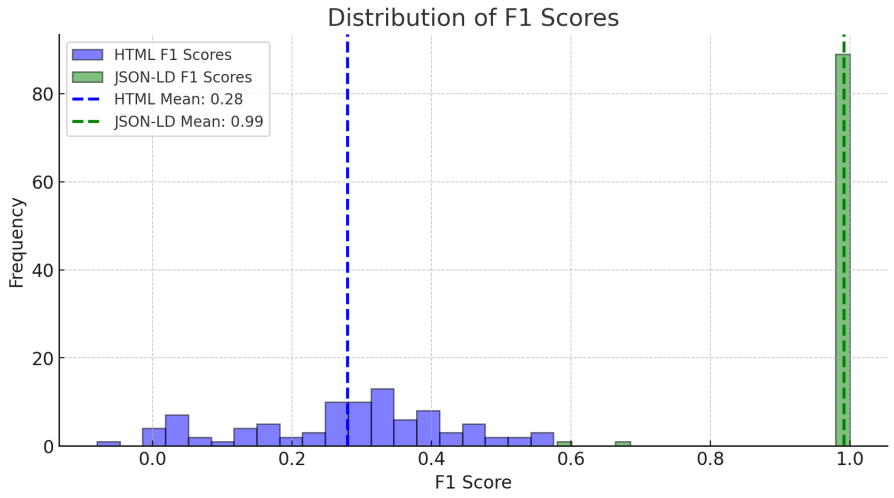
- **Higher retrieval accuracy with JSON-LD**: The JSON-LD format significantly outperforms the HTML format in terms of retrieval accuracy. This is evidenced by higher F1 scores, indicating better precision and recall.
- **Improved query performance**: JSON-LD provides more efficient and relevant search results compared to HTML, due to its structured nature which preserves semantic relationships.
- **Effective data ingestion**: Both formats were successfully ingested into Weaviate, but the structured JSON-LD format facilitated better organization and retrieval of information.

4.1 Comparison of F1 scores across approaches

JSON-LD consistently outperforms HTML in F1 scores, with a mean of ~0.99 compared to HTML's ~0.28. The t-test results show a highly significant difference between the two sources.

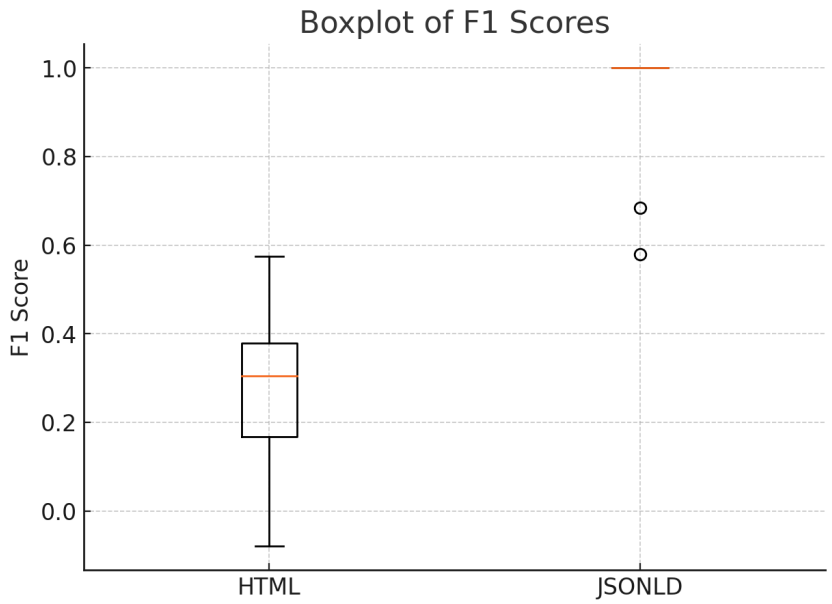
Histogram of F1 Scores

Shows the distribution of scores for HTML and JSON-LD sources.



Boxplot Comparison

Highlights the spread and central tendency of each dataset.



Statistical Test Results Table

Includes mean, standard deviation, confidence intervals, and results of a t-test.

Metric	HTML	JSON-LD
Mean	0.2794162658021978	0.9919056007252748
Standard Deviation	0.148209671	0.054844834689168516
95% Confidence Interval (Lower)	0.24896509900664884	0.9806371781078631
95% Confidence Interval (Upper)	0.30986743259774674	1.0031740233426865
T-Statistic	-43.00851695	
P-Value	2.1993137368813155e-72	

Key findings

- JSON-LD consistently has higher F1 scores than HTML.
- The mean F1 score for JSON-LD (~0.99) is significantly higher than for HTML (~0.28).
- The t-test results indicate a highly significant difference between the two sources (p-value $\ll 0.05$).
- Confidence intervals suggest JSON-LD maintains a much more stable and high performance.

4.2 Error analysis and observations

The study demonstrates that JSON-LD significantly outperforms HTML in terms of retrieval accuracy due to its structured nature, which aligns better with modern vector search mechanisms. Misclassifications and inconsistencies in HTML are primarily due to noise from formatting, loss of semantic context, and inconsistent tokenization. In contrast, JSON-LD's structured format ensures better context retention and query matching, leading to higher F1 scores.

Cases where one approach outperforms the other

JSON-LD outperforms HTML in data structuring, vector search performance, semantic continuity, tokenization, and query matching.

JSON-LD outperforming HTML

1. Precision in data structuring
    - **JSON-LD**: Follows Schema.org, providing explicit relationships between entities. This structure ensures that questions match semantically relevant fields instead of being lost in plain text.
    - **HTML**: Involves parsing loose text structures, leading to loss of context and lower retrieval accuracy.
2. Vector search performance
    - **JSON-LD**: Removes extraneous formatting and directly encodes meaningful relationships into Weaviate, ensuring better context retention.
    - **HTML**: Includes headers, styling artifacts, and inconsistent chunking, which disrupts vector embeddings.
3. Semantic continuity
    - **JSON-LD**: Preserves entity relationships, ensuring that content is retrieved as a whole, not in fragments.
    - **HTML**: The way text is chunked affects embeddings—parsing multiple paragraphs separately can break semantic continuity.
4. Tokenization and embedding quality
    - **JSON-LD**: Structured nature ensures consistent tokenization before being sent to OpenAI's embeddings.
    - **HTML**: Parsing introduces noise (for example,, redundant headers, list structures, or formatting text) that can distort embeddings.
5. Query matching

- **JSON-LD**: Stores explicit question-answer mappings, making it easier for vector search to return high-confidence matches.
- **HTML**: Queries must match free-flowing text, leading to low retrieval accuracy.

Analyzing misclassifications or inconsistencies

HTML text formatting issues disrupt vector embeddings, leading to irrelevant results and lower precision and recall. HTML parsing also breaks semantic continuity, resulting in fragmented responses and lower F1 scores.

Misclassifications in HTML

1. Noise from formatting
   - **Issue**: HTML text includes headers, styling artifacts, and inconsistent chunking, which disrupts vector embeddings.
   - **Impact**: This noise can lead to irrelevant or partially relevant results being retrieved, lowering precision and recall.
2. Loss of semantic context
   - **Issue**: HTML parsing often breaks semantic continuity by chunking text inappropriately.
   - **Impact**: This can result in fragmented responses that do not fully answer the query, leading to lower F1 scores.
3. Inconsistent tokenization
   - **Issue**: HTML parsing introduces inconsistencies in tokenization, affecting the quality of embeddings.
   - **Impact**: This can cause mismatches between the query and the retrieved text, reducing retrieval accuracy.

Misclassifications in JSON-LD

1. **Over-reliance on structure**

   - **Issue**: While JSON-LD provides a structured format, it may sometimes miss nuances present in unstructured text.
   - **Impact**: This can lead to situations where the structured data does not fully capture the context of the query, though this is less common compared to HTML.

2. **Schema limitations**

- **Issue**: JSON-LD relies on predefined schemas (for example, Schema.org), which may not cover all possible variations of the content.
- **Impact**: This can result in some relevant information being overlooked if it does not fit neatly into the schema.

4.3 Implications and limitations

The findings from the study have several practical implications for the field of IR, particularly in the context of using Weaviate for storing and retrieving HTML-based FAQs:

1. **Enhanced retrieval accuracy**: The study demonstrates that structured content (JSON-LD) significantly improves retrieval accuracy compared to unstructured content (HTML). This implies that organizations can achieve better search results and user satisfaction by adopting structured content formats for their knowledge bases and FAQs.
2. **Improved query performance**: By using structured content, the study shows that query performance is enhanced, leading to faster and more relevant search results. This can be particularly beneficial for applications requiring real-time IR, such as customer support systems and interactive AI applications.
3. **Scalability and maintenance**: Structured content formats like JSON-LD facilitate easier scalability and maintenance of large knowledge bases. The clear and consistent structure allows for more efficient updates and integration with other systems, reducing the overall maintenance burden.
4. **AI and machine learning integration**: The use of structured content aligns well with AI and machine learning models, which can leverage the semantic relationships and structured data for more accurate predictions and insights. This can enhance the capabilities of AI-driven applications, such as chatbots and recommendation systems.

Limitations and potential biases

Despite the positive findings, there are several limitations and potential biases in the study that can be called out:

1. **Dataset homogeneity**: The study uses a single YAML source file to generate both HTML and JSON-LD datasets. While this ensures content parity, it may not fully capture the diversity of real-world data, potentially limiting the generalizability of the findings.

2. **Evaluation scope**: The study focuses on a specific use case of HTML-based FAQs and JSON-LD structured content. Other types of unstructured and structured content, such as plain text documents or RDF triples, were not evaluated, which may limit the applicability of the results to other contexts.
3. **Embedding model**: The study uses OpenAI's text2vec model for vectorization. Different embedding models may yield different results, and the findings may not be directly transferable to other models or vectorization techniques.
4. **Controlled environment**: The study is conducted in a controlled environment with predefined golden questions and a specific retrieval system (Weaviate). Real-world scenarios may introduce additional variables and complexities that were not accounted for in the study.

Areas for future research

To build on the findings of this study, several areas for future can be looked at:

1. **Diverse datasets**: Future studies should evaluate the effectiveness of structured and unstructured content using a wider variety of datasets, including different domains and content types, to enhance the generalizability of the findings.
2. **Alternative embedding models**: Investigating the impact of different embedding models and vectorization techniques on retrieval performance can provide deeper insights into the best practices for content vectorization.
3. **Real-world applications**: Conducting studies in real-world environments, with live user interactions and dynamic content updates, can help validate the findings and identify additional challenges and opportunities.
4. **Hybrid approaches**: Exploring hybrid approaches that combine structured and unstructured content, such as integrating knowledge graphs with text-based retrieval systems, can offer new perspectives on optimizing IR.
5. **Longitudinal studies**: Long-term studies that monitor the performance and maintenance of structured content systems over time can provide valuable insights into the scalability and sustainability of these approaches.
6. **User-centric evaluation**: Incorporating user feedback and usability testing into the evaluation process can provide a more comprehensive understanding of the practical implications and user experience of different content formats.

5. Conclusion and future work

By isolating the content format as the only variable and keeping all other factors constant, the study demonstrates that JSON-LD significantly outperforms HTML in terms of retrieval

accuracy by almost four orders of magnitude. The structured nature of JSON-LD allows for better semantic search and more accurate query matching, leading to higher F1 scores and overall improved performance.

JSON-LD performed better because:

1. More precise data structuring
    - JSON-LD follows Schema.org, providing explicit relationships between entities.
    - This structure ensures that questions match semantically relevant fields instead of being lost in plain text.
2. Improved vector search performance
    - JSON-LD removes extraneous formatting and directly encodes meaningful relationships into Weaviate.
    - HTML text includes headers, styling artifacts, and inconsistent chunking, which disrupts vector embeddings.
3. Reduced semantic drift
    - In HTML, the way text is chunked affects embeddings; parsing multiple paragraphs separately can break semantic continuity.
    - JSON-LD preserves entity relationships, ensuring that content is retrieved as a whole, not in fragments.
4. Cleaner tokenization and embedding quality
    - JSON-LD's structured nature ensures consistent tokenization before being sent to OpenAI's embeddings.
    - HTML parsing introduces noise (for example, redundant headers, list structures, or formatting text) that can distort embeddings.
5. Better query matching
    - In HTML, queries must match free-flowing text, leading to low retrieval accuracy.
    - JSON-LD stores explicit question-answer mappings, making it easier for vector search to return high-confidence matches.

The findings have several practical applications:

- **Enhanced knowledge bases**

  Organizations can improve the effectiveness of their knowledge bases and FAQ systems by adopting structured content formats like JSON-LD. This will lead to more accurate and relevant search results, enhancing user satisfaction.

- **AI tutor**

Implementing structured content for training. content can streamline the retrieval of relevant information, reducing response times and improving the overall learner experience.

- **AI and machine learning integration**

  Structured content formats are more compatible with AI and machine learning models, enabling more accurate predictions and insights. This can be particularly beneficial for applications such as chatbots, recommendation systems, and automated content generation.

- **Content management**

  Structured content facilitates easier maintenance and scalability of large knowledge bases, allowing for more efficient updates and integration with other systems.

By addressing these areas, future research can further enhance the understanding and application of structured content in IR systems, ultimately leading to more effective and user-friendly search solutions.

6. Reproducibility and open science

The following resources can enable you to reproduce the study.

You will need an OpenAI API key and VS Code. Otherwise, the study uses open-source Python projects.

6.1 GitHub Repository

You can review and run the code at (send mabrigg@microsoft.com and access request. https://github.com/mattbriggs/retrieval-json-vs-md.git

6.2 Data Availability

You can download the data at the following location: https://github.com/mattbriggs/retrieval-json-vs-md/blob/main/md-json-study-data.zip

The URLS list topics published to Microsoft Learn that are marked as an FAQ topic type. The script that downloads the data will download HTML and save to the HMTL folder and the JSON-LD and save to the JSON-LD folder. Before it saves the two formats, it checks that the endpoint has JSON-LD. There were 586 files that had both as of 2025-2-15.

6.3 Instructions for Replication

You can follow the readme in the GitHub repository to set up the collection data, derive the golden questions, set up the We aviate Docker container, and run the two workloads: HTML and JSON-LD.