

QOSI

QOSI - Quadrillion Open Source Indexer

- [QOSI - Quadrillion Open Source Indexer](#)
- [Glossary](#)

QOSI - Quadrillion Open Source Indexer

Glossary

Basic Descriptors

Object

- **Definition:** Any unit that can be processed or analyzed by the QOSI system.
- **Types:**
 - **File, File ID,**
 - **Offset, Offset Index**
 - **Tokenized File, Tokenized File ID**
 - **Candidate Object List, Candidate Object Bitmap**
 - **MTS, MTSU**
 - **Bitmap**
 - TODO:
 - etc.

Target

- **Definition:** User-provided objects intended for analysis.
- **Descriptions:**
 - Consists of objects submitted by the user.
 - Always remains separate and is never merged into the Source.
- **Characteristics:**
 - Relatively smaller than the Source.
 - Never incorporated into the Source.
- **Workflow:**
 - Is subject to immediate tokenization and/or indexing once received.

Source

- **Definition:** Objects to match against the Target.
- **Description:**
 - Objects maintained by the service provider.
 - The main data collection (objects) for queries and comparisons.
- **Characteristics:**

- Acts as the reference pool for matching against the Target.
- Mainly OSS(Open Source Software) or other public datasets.
- ~~It is readily available. Pre-indexed.~~
- It is updated infrequently
 - e.g., weekly or monthly
- **TODO:** Move to Indexing Candidate Indexing duration has no significant impact on data structure design.

Candidate

- **Definition:** Potential objects with common subsequence with the Target.
- **Description:**
 - Identified during the Candidate Query phase.
 - Subset of the Source.
- **Characteristics:**
 - May include false positives but never false negatives.
 - Used to quickly identify potential matches in the Source.
 - In current, every candidate includes **MTS**, but may not include **MSs**.

Identified

- **Definition:** The subset of Source/Target objects that contain a common subsequence with Target/Source.
- **Description:**
 - Identified during the Comparing phase.
 - **Identified Source:** Source objects that contain a common subsequence with the Target.
 - **Identified Target:** Target objects that contain a common subsequence with the Source.

Index

- **Definition:** A metadata to identify the order of the data. Similar to ID.
- **Description:**
 - Many objects (ex: File, Tokenized File, Token, etc.) have their own Index.

Immutable

- **Definition:** Would not be changed during the process.
- **Description:**
 - **The Index** is immutable after the indexing phase(when the index is created).

Dataset

Dataset

- **Definition:** An aggregated collection of objects.
- **Characteristics:**
 - Each Dataset may contain numerous objects
 - A dataset may contain only the same type or different types of objects.
 - source code, text, etc.
 - **File, Tokenized File, Candidate Object List**

Source Dataset

- **Definition:** The comprehensive set of objects maintained as the Source.
- **Usage:**
 - Interchangeably referred to simply as "Source."

Target Dataset

- **Definition:** The complete set of objects the user provides.
- **Usage:**
 - Interchangeably referred to simply as "Target."

Token and Sequence

Token

- **Definition:** The smallest unit of data derived from the content of a file.
- **Description:**
 - Generated through tokenization—splitting code or text into discrete elements.
 - Varies by language; e.g., whitespace-based for Western languages, morphological analysis for Japanese.
- **Characteristics:**
 - Enables fine-grained comparison between Source and Target files.
 - Significantly reduces duplication compared to entire lines or blocks of text.
- **Details:**
 - **Code Data:** Tokenized according to code syntax rules.

- **Text Data (Natural Language):**
 - **Western Languages:** Split by spaces (plus punctuation considerations).
 - **Chinese:** Every character is treated as a token.
 - **Korean:** Split by spaces; postpositional particles and suffixes are separately tokenized.
 - **Japanese:** Uses morphological analysis to segment words based on grammar and context.
 - **Special Cases:** Korean/Japanese require specialized tokenizers due to morphological complexity.

Syntax Token

- **Definition:** A token that does not reflect its original string but its syntax role.

String Token

- **Definition:** A token that reflects its original string.

Sequence and Subsequence

- **Definition:**
 - **Sequence:** An ordered list of tokens derived from a file or segment.
 - **Subsequence:** A contiguous subset of tokens within a sequence.
- **Description:**
 - Used for matching and comparison processes (e.g., detecting shared code fragments).

Minimum Token Sequence (MTS) and Minimum Token Sequence Unit (MTSU)

- **Definition:**
 - **MTS:** A fixed-size subsequence of tokens from the original sequence.
 - **MTSU:** The hashed version of an **MTS**, serving as a fundamental unit for indexing, candidating, and comparing.
- **Characteristics:**
 - **MTSU** is generated by hashing an **MTS**.
 - Because an **MTS** is multiple tokens combined, it reduces the chance of random duplication compared to single-token matching.

Minimum Subsequence (MSs) (or Minimum Token Subsequence (MTSs))

- **Definition:** The subsequence that is requested by minimum size(length) by the user.
- **Details:**
 - **MSs** should be equal or greater(longer) than **MTS**.
 - When comparing on Comparator, connect **MTS(MTSU)** to construct **MSs**.
- **Example:**
 - **MTS:** 3 tokens
 - **MSs:** 5 tokens
 - When finding a **MSs** `A B C D E` in a file, the Comparator will find **MTS** `A B C`, `B C D` and `C D E` by **MTSU** and match it to **MSs** `A B C D E`.

Hash Function

- **Definition:** A function that converts an **MTS** into a fixed-size value (an **MTSU**).
 - **Description:**
 - Ensures consistent, quick comparisons.
 - Takes not only a **Token**, but **Token Subsequence** to get more identification power.
-

File and Tokenized File

File

- **Definition:** A code and/or text container.
- **Description:**
 - Consists of a sequence of original tokens (raw source code, natural language text).
 - Only **Reporter**(viewer) access the **file** directly
 - In **Indexing**, and **Comparing** phases, the system uses **Tokenized Files**.
- **Characteristics:**
 - Typically stored in a filesystem or code repository.
 - **Source Files** are stored in **Archiver**.
 - Can be part of either **Source Dataset** or a **Target Dataset**.

Tokenized File

- **Definition:** A file after it has been converted into a sequence of tokens.

- **Description:**

- Produced by a tokenizer from the raw file.
- Consists of pairs: (**Token Index**, **MTSU**, separated token #1, separated token #2, ...).

Token Index

- **Definition:** The token's position in the token sequence of the file.
-

Partitioning

- **Definition:** Dividing something by **MTSU** to optimize searching.
 - **Candidate Index**, **File Archive**, and etc.
-

Query

Query, Query Request, and Query Result

- **Definition:**
 - **Query:** The act of searching the Source for common subsequences found in the Target.
 - **Query Request:** Specific search parameters used to locate relevant subsequences in Source files.
 - **Query Result:** The outcome of a Query, typically pairing Target files with matching Source files.
- **Description:**
 - Involves scanning across the entire Target dataset to find potential matches in the Source.
 - Candidate Queries help identify which Source files might contain matching subsequences.
- **Types:**
 - **Exact Query:** Finds subsequences that match exactly between Source and Target.
 - **Similar Query:** Finds similar subsequences, given a user-defined similarity threshold.
- **Result:**
 - Pairs each Source file with the relevant Target file.
 - Includes index ranges (start/end) of matching subsequences for both files.
 - By Token Index
 - By Line Number

Candidate Query

Definition:** A preliminary, broad search returns only a list of candidate files from the Source.

- **Description:**
 - Operates over the entire Source dataset.
 - Allows false positives but no false negatives (i.e., it might over-include but never miss a true match).
 - It does not directly support similar queries (though smaller **MTS** sizes can approximate similarity).

Batch Query

- **Definition:** A method for processing multiple queries to improve efficiency.
- **Characteristics:**
 - Reduces repeated work by handling similar queries in a single pass.
 - Constructed by **MTS** list
 - Link between **Source ID** and **MTS** is not sent to **Nominator**, but stored in **Querier**.
- **Example:**
 - **Target Object A's Token Sequence:** A B C D E F G
 - A's **MTS:**
 - A B C
 - B C D
 - C D E
 - D E F
 - E F G
 - **Target Object B's Token Sequence:** A B C E F G
 - B's **MTS:**
 - A B C
 - B C E
 - C E F
 - E F G
 - Batch Query: A B C, B C D, C D E, B C E, C E F, E F G
 - Common **MSs** among **Target Object** (ex: A B C, E F G) are not duplicated in **Batch Query**

Preprocess Phase

Preprocessor

- **Definition:** A component that converts a **File** to a **Tokenized File**, and extract metadata.
- **Description:**
 - Extracts metadata from the file.
 - Include Tokenizing(**Tokenizer**)

Tokenizer

- **Definition:** A component that converts a **File** to a sequence of tokens and make a **Tokenized File**.

File Archiver

- **Definition:** Retrieves a **Source File** and/or **Source Tokenized File**'s content by the **file ID**.
 - **Description:**
 - Archive **Source Files** and **Source Tokenized Files**, and its metadata.
-

Index Phase

Candidate Index (The Index)

- **Definition:** A data structure built from candidate Source files to facilitate queries.
- **Description:**
 - Used to respond to **Candidate Queries** quickly.
 - It acts as a high-level map of **Source Files**.
 - Is **immutable** after the indexing phase.
 - **Nominator** does not modify **The Index**.
 - Is organized (sorted) by **MTSU** to speed up lookups.
 - **Partitioning** is applied to **The Index** for efficient searching.
- **Types:**
 - **Raw Candidate Index**
 - Is not optimized or compressed.
 - May use **RocksDB** or **LevelDB**.
 - **Candidate Index, The Index**
 - Is optimized and compressed.
 - **Nominator** uses this index.

Indexer (Source File Indexer)

- **Definition:** Builds **the Index** from the Source datasets.
- **Description:**
 - Converts Source Files into Tokenized Files.
 - Generates **MTSU** entries and relevant metadata for the Index.
- **Workflow**
 - From **Tokenized File**, record the **file ID** by **MTSU** to **Candidate Index**.

Index Compressor

- **Definition:** Extract Key-Value and Compress to reduce the size of the Index to improve performance.
 - **Description:**
 - **Workflow:**
 - Extract Key-Value pairs from **Candidate Index**.
 - Compress Key-Value pairs by **Elias-Fano Encoding**.
-

Query Phase

Querier

- **Definition:** Generates and executes queries based on **Target Files**.
- **Description:**
 - Before **Querier**, it should pass **Preprocessing** for **Target**.
 - Produces **Query Requests** that lists **MTSU** and metadata from the **Target Object**.
 - Generate **Batch Query**
 - May group or deduplicate **MTSU** to optimize further searching.
 - Stores **MTSU** and **Source ID** map.

Nominator

- **Definition:** Identifies candidate files during query operations.
- **Description:**
 - Takes a Query Request and looks up possible file matches from **the Index**.
 - Returns a list of **Source file IDs** that contain matching **MTSU**.
 - Key: **MTSU**

- Value: **Source file ID** list (by **Bitmap**)
-

Comparison Phase

Merger

- **Definition:** Consolidates and refines the final list of candidate files after a **candidate query** from **Nominator**.
- **Description:**
 - Produces the final candidate file set for the Query.
 - Combines results across multiple **MTSU** lookups for each **Target File**.
 - Use **Bitmap** to merge(OR operation) the candidate file list.

File Extractor

Comparator

Evaluator

Reporter

Partitioning

Language Family

- **Definition:** A grouping of languages based on similar syntax or representation.
- **Description:**
 - Languages in the same family may share identical tokenization patterns.
 - A single language can belong to multiple families if it exhibits shared features with different groups.
 - Helps partition Source data for more efficient indexing and searching.

Parameter

Parameter

- **Definition:** A user-defined setting that customizes query behavior.
 - **Types:**
 - **Similarity Rate:** Sets the degree of overlap needed for matches in a Similar Query.
 - **Search Unit (by token):**
 - **MTSU Size:** The smallest hashable subsequence size.
 - **Minimum Search Unit Size:** The smallest token sequence considered for matching.
 - **Description:**
 - Affects how queries are performed and how results are filtered.
 - Allows for tuning accuracy versus performance.
-