Generative Information Retrieval

The Web Conference 2024 tutorial – Section 3

Yubao Tang[a], Ruqing Zhang[a], **Zhaochun Ren**[b], **Weiwei Sun**[c], Jiafeng Guo[a] and **Maarten de Rijke**[d]

https://TheWebConf2024-generative-IR.github.io

May 14, 2024

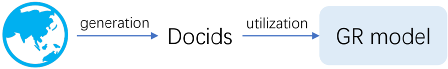
[a] Institute of Computing Technology, Chinese Academy of Sciences & UCAS
[b] Leiden University
[c] Shandong University
[d] University of Amsterdam

Section 3:
Docid design

Challenges of docid design

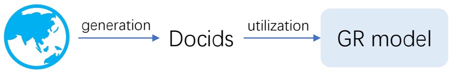
- Shall we use randomize numbers as the docids?
- If not, how to construct proper docids for the documents?
- Would the choices of different docids affect the model performance (effectiveness, capacity, etc.)?

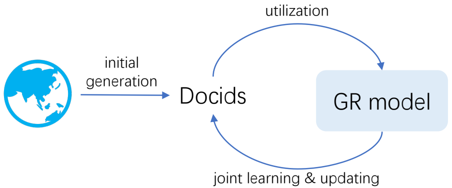
Categorization of docids



- Pre-defined static docids

Categorization of docids



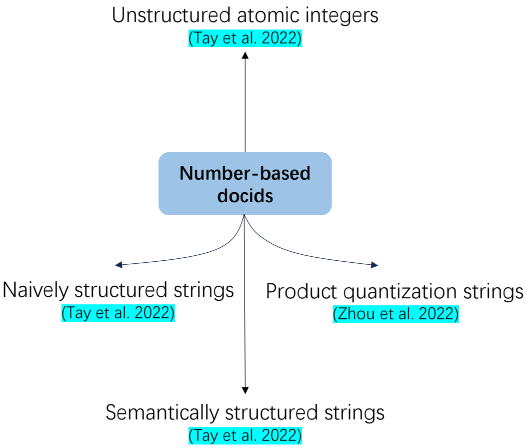


- Pre-defined static docids
- Learnable docids

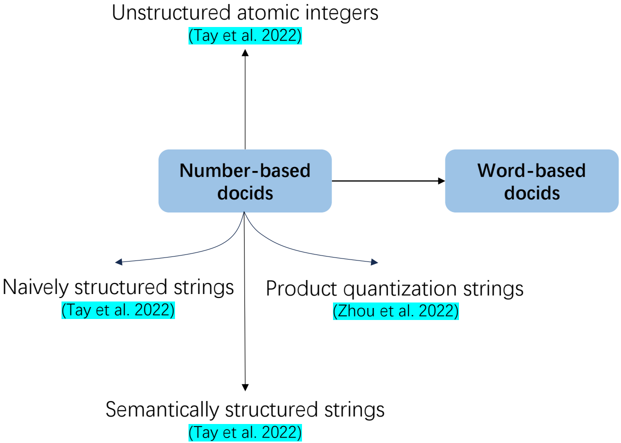
Roadmap of pre-defined static docids

Number-based docids

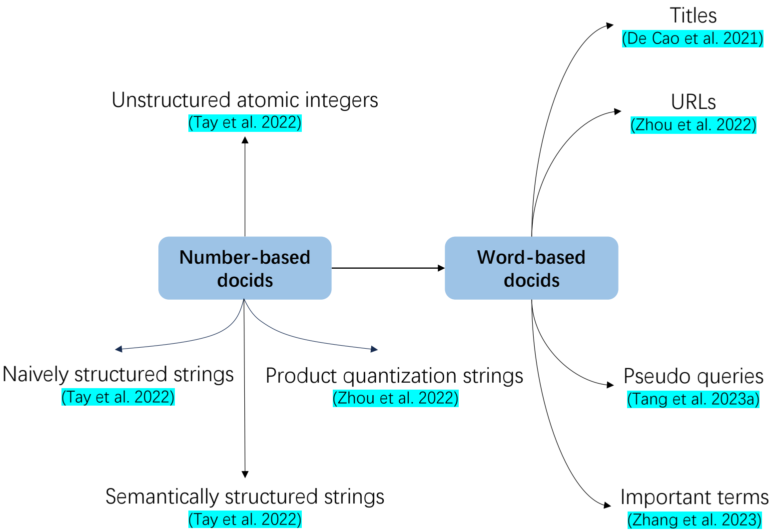
Roadmap of pre-defined static docids



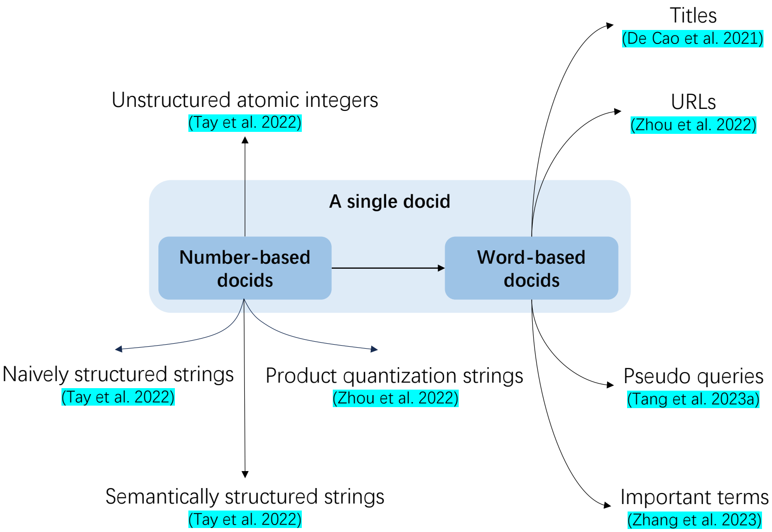
Roadmap of pre-defined static docids



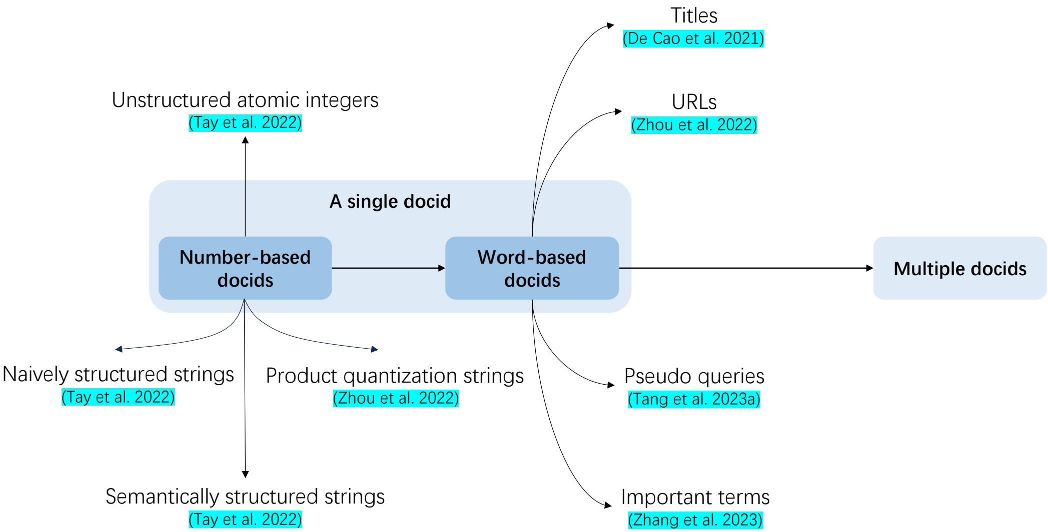
Roadmap of pre-defined static docids



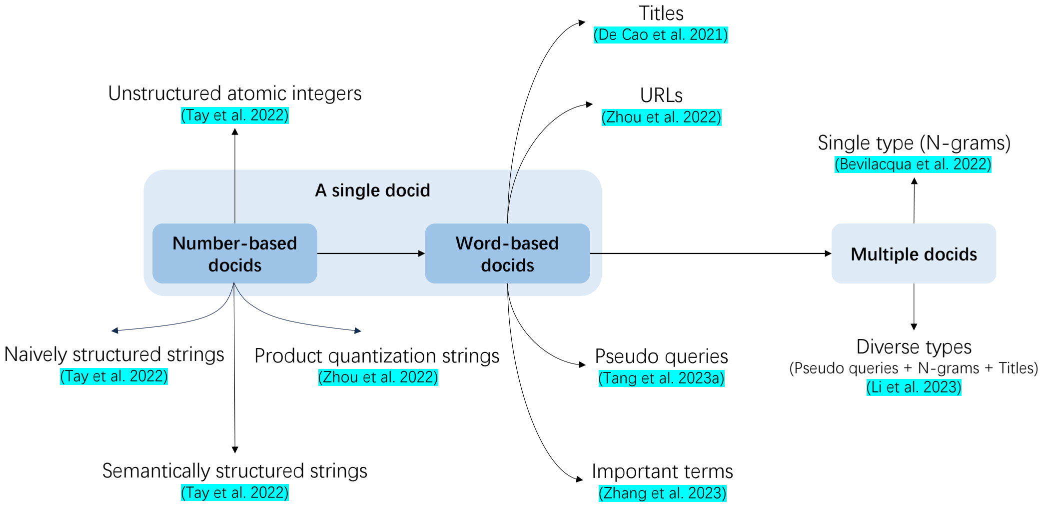
Roadmap of pre-defined static docids



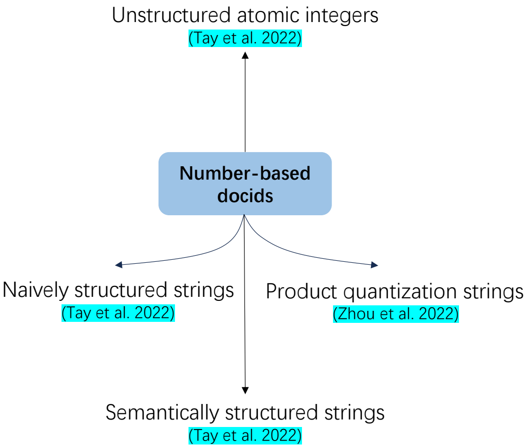
Roadmap of pre-defined static docids



Roadmap of pre-defined static docids



A single docid: Number-based



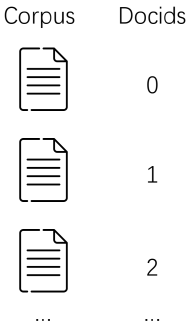
Number-based: Unstructured atomic integers

- An arbitrary (and possibly random) unique integer identifier

Source: [Tay et al., 2022]

Number-based: Unstructured atomic integers

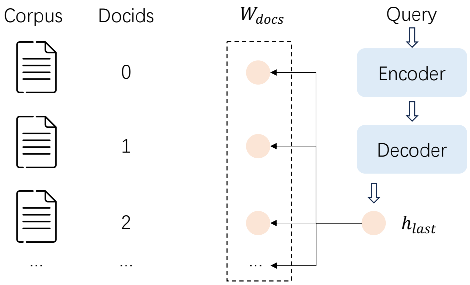
- An arbitrary (and possibly random) unique integer identifier



Source: [Tay et al., 2022]

Number-based: Unstructured atomic integers

- **Decoding formulation**: learn a probability distribution over the docid embeddings, i.e., emitting one logit for each unique docid

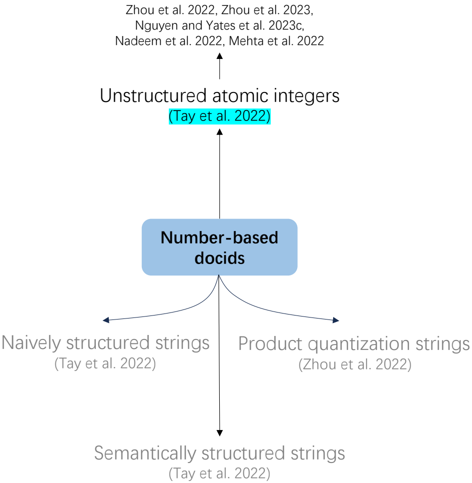


$$O = \mathrm{Softmax}([W_{docs}]^T h_{last}),$$

where $[W_{docs}]$ is the document embedding matrix, and h_{last} is the last layer's hidden state of the decoder

Source: [Tay et al., 2022]

Unstructured atomic integers and subsequent work



Unstructured atomic integers and subsequent work

Zhou et al. 2022, Zhou et al. 2023,
Nguyen and Yates et al. 2023c,
Nadeem et al. 2022, Mehta et al. 2022

↑

Unstructured atomic integers
(Tay et al. 2022)

↑

Number-based docids

Naively structured strings
(Tay et al. 2022)

Product quantization strings
(Zhou et al. 2022)

Semantically structured strings
(Tay et al. 2022)

Easy to build: analogous to the output layer in standard language model

Unstructured atomic integers: obvious constraints

👎 The need to learn embeddings for each individual docid

Unstructured atomic integers: obvious constraints

- 👎 The need to learn embeddings for each individual docid
- 👎 The need for the large softmax output space

Unstructured atomic integers: obvious constraints

- 👎 The need to learn embeddings for each individual docid
- 👎 The need for the large softmax output space

It is challenging to be used on large corpora!

Number-based: Naively structured strings

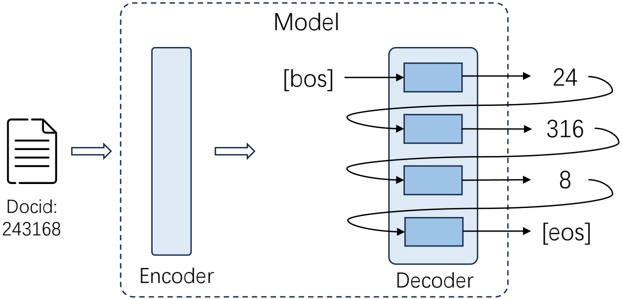
- Treat arbitrary unique integers as tokenizable strings

Corpus	Docids		Tokenized docid strings
	243168	Tokenizing based on model's vocabulary ⟹	24 316 8
...	...		...

Source: [Tay et al., 2022]

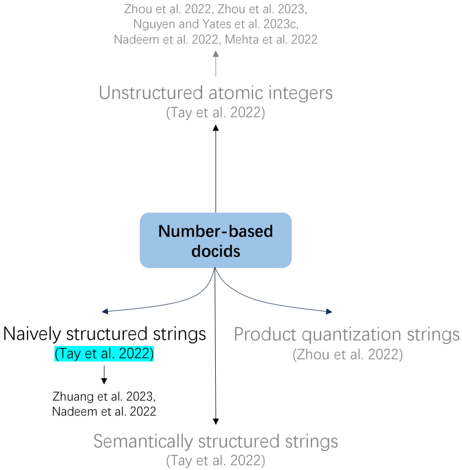
Number-based: Naively structured strings

- **Decoding formulation**: Generating a docid string in a token-by-token manner

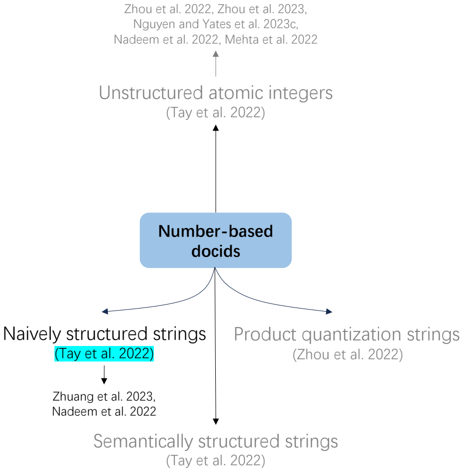


Source: [Tay et al., 2022]

Naively structured strings and subsequent work



Naively structured strings and subsequent work



Such a way frees the limitation for the **corpus size** that comes with unstructured atomic docid

Naively structured strings: obvious constraints

👎 Identifiers are assigned in an arbitrary manner

Naively structured strings: obvious constraints

- 👎 Identifiers are assigned in an arbitrary manner
- 👎 The docid space lacks semantic structure

Number-based: Semantically structured strings

Properties:

- The docid should capture some information about the semantics of its associated document

Source: [Tay et al., 2022]

Number-based: Semantically structured strings

Properties:

- The docid should capture some information about the semantics of its associated document
- The docid should be structured in a way that the search space is effectively reduced after each decoding step

Source: [Tay et al., 2022]

Number-based: Semantically structured strings

Properties:

- The docid should capture some information about the semantics of its associated document
- The docid should be structured in a way that the search space is effectively reduced after each decoding step

↓

Semantically similar documents share docid prefixes

Source: [Tay et al., 2022]

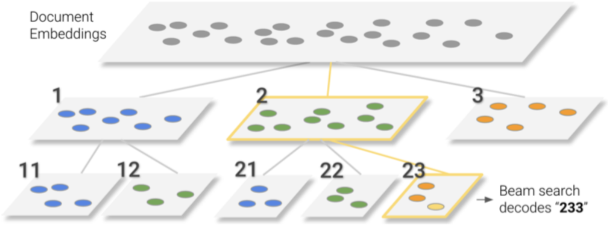
Number-based: Semantically structured strings

- A hierarchical clustering algorithm over document embeddings to induce a decimal tree

Source: [Tay et al., 2022]

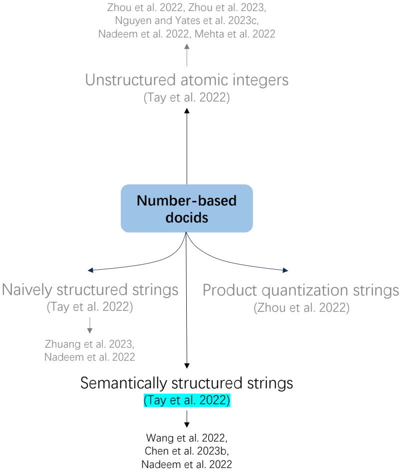
Number-based: Semantically structured strings

- A hierarchical clustering algorithm over document embeddings to induce a decimal tree

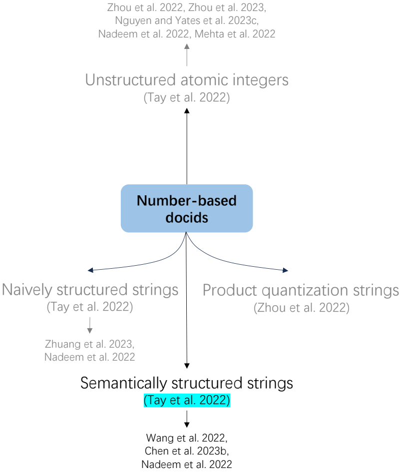


Source: [Tay et al., 2022]

Semantically structured strings and subsequent work

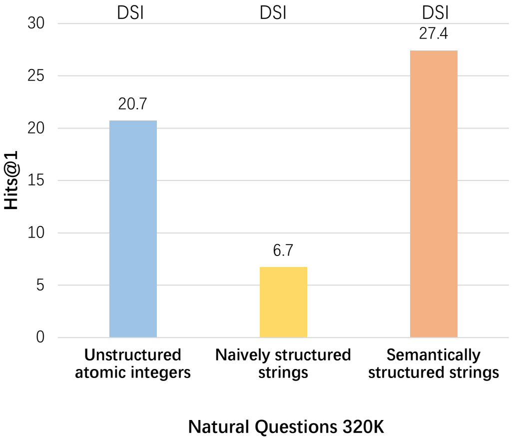


Semantically structured strings and subsequent work



- The document semantics can be incorporated in the decoding process
- It is not limited by the size of the corpus

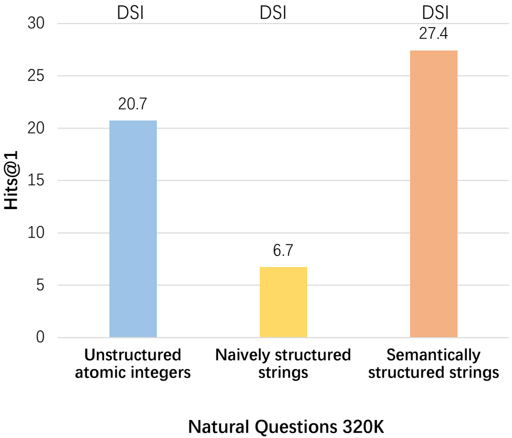
Performance comparisons [Tay et al., 2022]



- Backbone: T5-base
- Observations: imbuing the docid space with semantic structure can lead to better retrieval capabilities

Data source: Tay et al. [2022]

Performance comparisons [Tay et al., 2022]



- Backbone: T5-base
- Observations: imbuing the docid space with semantic structure can lead to better retrieval capabilities

This is only about "identifiers"

Later sections will discuss the performance compared to traditional IR models

Data source: Tay et al. [2022]

Number-based: Product quantization strings

- Product quantization (PQ) is a technique used for vector compression

Image source: Product Quantization for Similarity Search

Number-based: Product quantization strings

- Product quantization (PQ) is a technique used for vector compression
- An original vector is represented by a short code composed of its subspace quantization indices

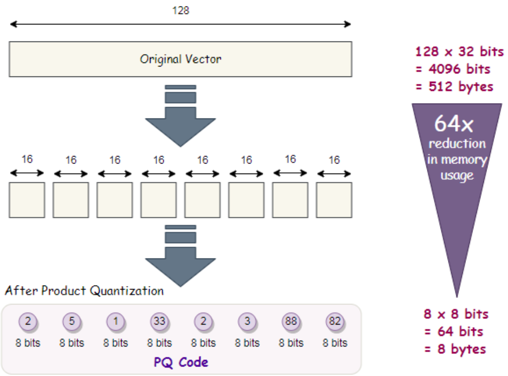


Image source: Product Quantization for Similarity Search

Number-based: Product quantization strings

Given all D-dimensional embedding vectors of documents [Zhou et al., 2022],

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Number-based: Product quantization strings

Given all D-dimensional embedding vectors of documents [Zhou et al., 2022],

- Divide the D-dimensional space into m groups

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Number-based: Product quantization strings

Given all D-dimensional embedding vectors of documents [Zhou et al., 2022],

- Divide the D-dimensional space into m groups
- Perform K-means clustering on each group to obtain k cluster centers

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Number-based: Product quantization strings

Given all D-dimensional embedding vectors of documents [Zhou et al., 2022],

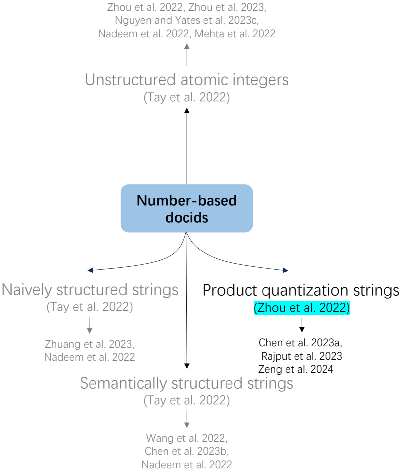
- Divide the D-dimensional space into m groups
- Perform K-means clustering on each group to obtain k cluster centers
- Each embedding vector can be represented as a set of m cluster identifiers. For each document d, its product quantization string identifier id_{PQ} can be defined,

$$id_{PQ} = PQ(Encoder(d)),$$

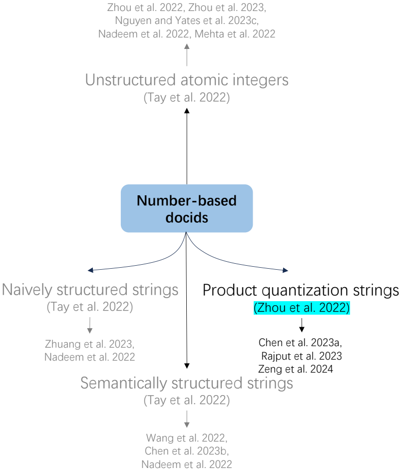
where $Encoder(\cdot)$ can be implemented by different language models

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Product quantization strings and subsequent work

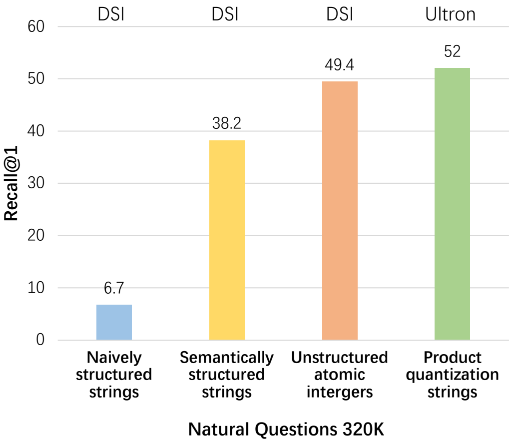


Product quantization strings and subsequent work



- Preserving dense vector semantics in a smaller space
- Capturing local semantic information

Performance comparisons



- Backbone: T5-base
- Observations: Product quantization string docids improves over structured semantic docids

Data source: Wang et al. [2023]

Number-based docids: Summary

- Docids based on integers are easy to build

Number-based docids: Summary

- Docids based on integers are easy to build
- Unstructured atomic integers and naively/semantically structured strings can maintain uniqueness

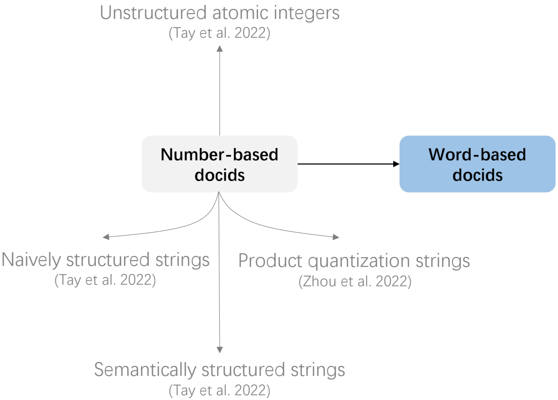
Number-based docids: Summary

- Docids based on integers are easy to build
- Unstructured atomic integers and naively/semantically structured strings can maintain uniqueness
- They are composed of unreadable numbers

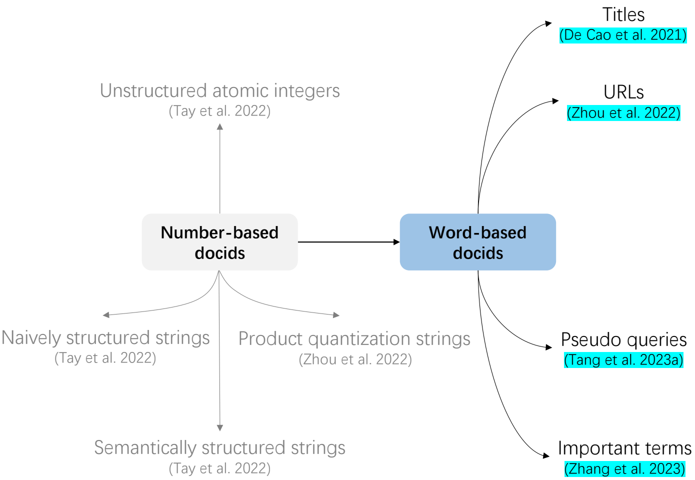
Number-based docids: Summary

- 👍 Docids based on integers are easy to build
- 👍 Unstructured atomic integers and naively/semantically structured strings can maintain uniqueness
- 👎 They are composed of unreadable numbers
- 👎 It is challenging to interpret the model's understanding of the corpus

A single docid: Word-based



A single docid: Word-based



A single docid: Word-based

The fundamental inspiration

- The query is usually keyword-based natural language, which can be challenging to map into a numeric string, while mapping it to words would be more intuitive

Word-based: Titles

- Document titles: be able to summarize the main content

Image source: Wikipedia and CNN

"Autoregressive Entity Retrieval". De Cao et al. [2021]

Word-based: Titles

- Document titles: be able to summarize the main content

Information retrieval Decoding target

Article Talk

From Wikipedia, the free encyclopedia

Information retrieval (**IR**) in computing and information science is the process of obtaining information system resources that are relevant to an information need from a collection of those resources. Searches can be based on full–text or other content–based indexing. Information retrieval is the science[1] of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds.

Automated information retrieval systems are used to reduce what has been called information overload. An IR system is a software system that provides access to books, journals and other documents; it also stores and manages those documents. Web search engines are the most visible IR applications.

Chiamaka Nnadozie's father didn't want her to play soccer. Nigerian star defied him and rewrote the record books

Decoding target

By Michael Johnston and Amanda Davies, CNN

5 minute read · Updated 10:06 AM EDT, Wed November 1, 2023

(CNN) — It wasn't always plain sailing for Paris FC and Nigerian goalkeeper, Chiamaka Nnadozie, throughout her now-flourishing career.

Growing up in a family of boys and men – who had all tried their hand at going professional – Nnadozie's ambition to follow suit wasn't greeted with unyielding enthusiasm. Quite the opposite.

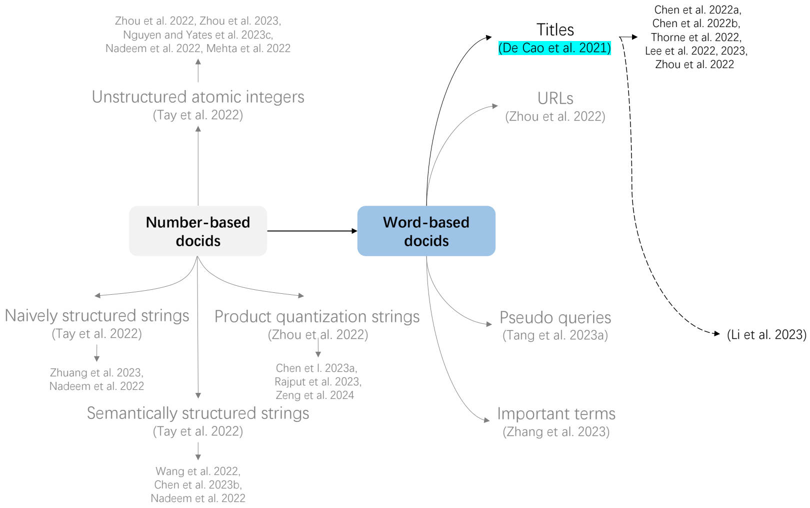
"It wasn't very good from my family. They never let me play, especially my dad," the 22-year-old told CNN's Amanda Davies.

"Whenever I went to play soccer, he would always tell me: 'Girls don't play football. Look at me. I played football, I didn't make it. Your brother, he played, he didn't make Your cousin played, he didn't make it. So why do you want to choose this? Why don't you want to go to school or maybe do some other things?'" Nnadozie recollected.

Image source: Wikipedia and CNN

"Autoregressive Entity Retrieval". De Cao et al. [2021]

Titles and subsequent work



Titles: Obvious constraints

👎 Depending on certain special document metadata

Titles: Obvious constraints

- 👎 Depending on certain special document metadata
- 👎 The titles may be duplicated (i.e., web datasets), and require further investigation

Titles: Obvious constraints

- 👎 Depending on certain special document metadata
- 👎 The titles may be duplicated (i.e., web datasets), and require further investigation
- 👎 Time-consuming step of producing titles and requiring increasingly sophisticated domain knowledge

For a while, mainly evaluated on Wikipedia-based tasks (with well-written titles)!

Wikipedia-based tasks

Fact Verification
De Cao et al. 2021, Chen et al. 2022b, Chen et al. 2022a, Thorne et al. 2022, Lee et al. 2023

Entity Linking
De Cao et al. 2021, Chen et al. 2022b, Lee et al. 2023

Slot Filling
De Cao et al. 2021, Chen et al. 2022b, Lee et al. 2023

Open Domain QA
De Cao et al. 2021, Chen et al. 2022b, Zhou et al. 2022, Lee et al. 2023

Dialogue
De Cao et al. 2021, Chen et al. 2022b, Lee et al. 2023

Multi-hop retrieval
Lee et al. 2022

Word-based: URLs

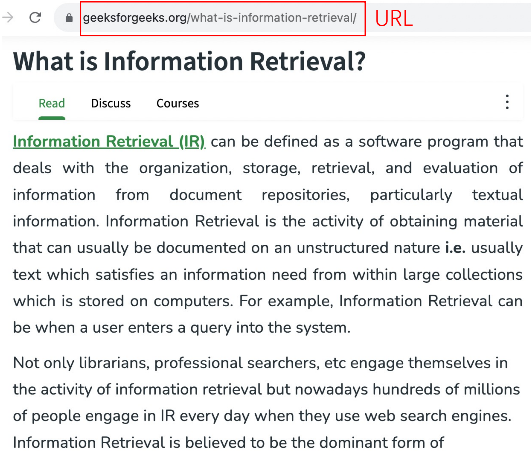
- The URL of a document contains certain semantic information and can uniquely correspond to this document

Image source: Geeksforgeeks

"TOME: A Two-stage Approach for Model-based Retrieval". Ren et al. [2023]

Word-based: URLs

- The URL of a document contains certain semantic information and can uniquely correspond to this document



What is Information Retrieval?

Read Discuss Courses

Information Retrieval (IR) can be defined as a software program that deals with the organization, storage, retrieval, and evaluation of information from document repositories, particularly textual information. Information Retrieval is the activity of obtaining material that can usually be documented on an unstructured nature **i.e.** usually text which satisfies an information need from within large collections which is stored on computers. For example, Information Retrieval can be when a user enters a query into the system.

Not only librarians, professional searchers, etc engage themselves in the activity of information retrieval but nowadays hundreds of millions of people engage in IR every day when they use web search engines. Information Retrieval is believed to be the dominant form of

Image source: Geeksforgeeks

Word-based: URLs

https://en.wikipedia.org/wiki/Nevada

$\Downarrow$ tokenize

https :// en . Wikipedia . org / wiki / N e vada

- Ren et al. [2023] solely utilized tokenized URLs as the docid

"TOME: A Two-stage Approach for Model-based Retrieval". Ren et al. [2023]

Word-based: URLs

https://en.wikipedia.org/wiki/Nevada

$\Downarrow$ tokenize

https :// en . Wikipedia . org / wiki / N e vada

- Ren et al. [2023] solely utilized tokenized URLs as the docid
- The tokenized symbols of URLs are well aligned with the vocabulary of the generative language model, thereby enhancing the generative capacity

"TOME: A Two-stage Approach for Model-based Retrieval". Ren et al. [2023]

Word-based: URLs

- However, not all URLs provide sufficient semantic information

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Word-based: URLs

- However, not all URLs provide sufficient semantic information
- Zhou et al. [2022] proposed to combine the URL and the document title as docids to guarantee both the uniqueness and semantics of docids

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Word-based: URLs

- However, not all URLs provide sufficient semantic information
- Zhou et al. [2022] proposed to combine the URL and the document title as docids to guarantee both the uniqueness and semantics of docids

For a while, mainly evaluated on Web search datasets (with available URLs)!

"Ultron: An ultimate retriever on corpus with a model-based indexer". Zhou et al. [2022]

Web search datasets

MS MARCO
Nguyen et al. 2016

Natural Questions
Kwiatkowski et al. 2019

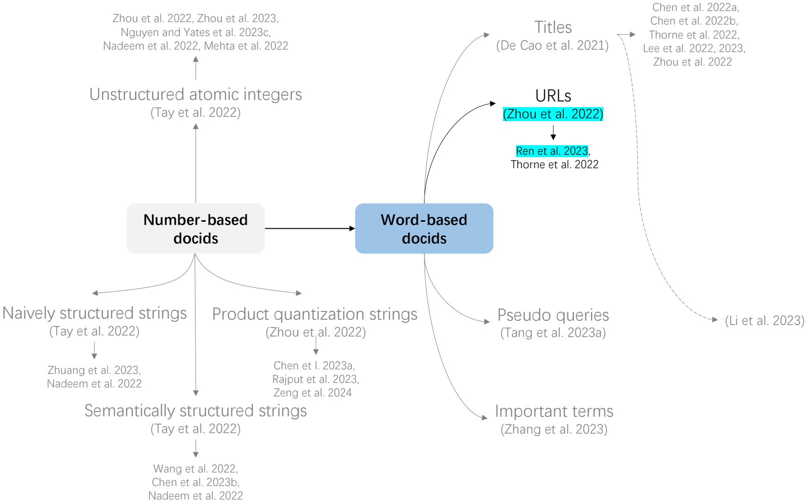
Trec-CAR
Dietz et al. 2017

Robust04
Voorhees et al. 2004

ClueWeb09-B
Clarke et al. 2010

Gov2
Clarke et al. 2004

URLs and subsequent work



If the special document metadata is not available

It is necessary to design automatic docid generation techniques

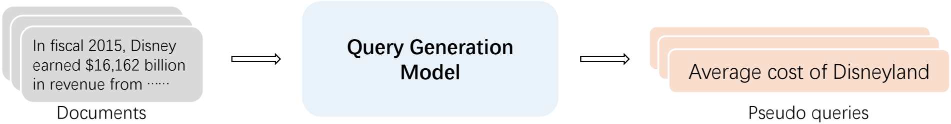
Word-based: Pseudo queries

- Doc2Query technique: pseudo queries are likely to be representative or related to the contents of documents

"Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies". Tang et al. [2023]

Word-based: Pseudo queries

- Doc2Query technique: pseudo queries are likely to be representative or related to the contents of documents



"Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies". Tang et al. [2023]

Word-based: Pseudo queries

- Docid repetition problem
  - Tang et al. [2023] use the top 1 generated query as the docid for each document
  - Based on statistics, about 5% and 3% docids of documents are not unique in MS MARCO and Natural questions datasets, respectively
  - It is reasonable that different documents may share the same docid if they share very similar essential information

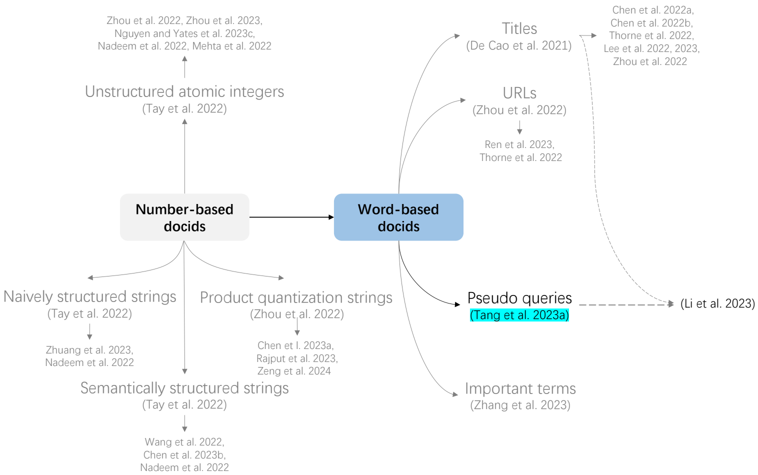
"Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies". Tang et al. [2023]

Word-based: Pseudo queries

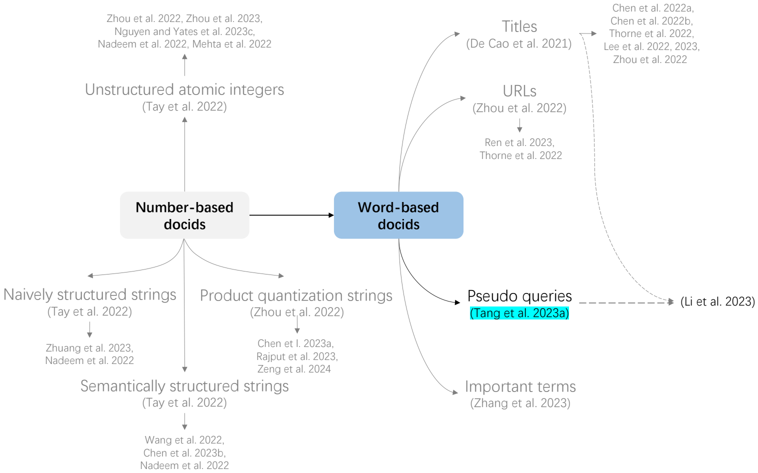
- Docid repetition problem
  - Tang et al. [2023] use the top 1 generated query as the docid for each document
  - Based on statistics, about 5% and 3% docids of documents are not unique in MS MARCO and Natural questions datasets, respectively
  - It is reasonable that different documents may share the same docid if they share very similar essential information
- Countermeasure
  - If a docid corresponds to multiple documents, return all of them in a random order, while keeping the relative order of documents corresponding to other docids

"Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies". Tang et al. [2023]

Pseudo queries and subsequent work



Pseudo queries and subsequent work



Without the requirements of certain document metadata, e.g., titles and URLs

False pruning [Zhang et al., 2023]

Titles, URLs and pseudo queries:

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

False pruning [Zhang et al., 2023]

Titles, URLs and pseudo queries:

- One pre-defined sequence

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

False pruning [Zhang et al., 2023]

Titles, URLs and pseudo queries:

- One pre-defined sequence
- The requirement for the exact generation

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

False pruning [Zhang et al., 2023]

Titles, URLs and pseudo queries:

- One pre-defined sequence
- The requirement for the exact generation
- If a false prediction about its docid is made in any step of the generation process, the targeted document will be missed from the retrieval result

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

The permutation of docids becomes critical

Word-based: Important terms [Zhang et al., 2023]

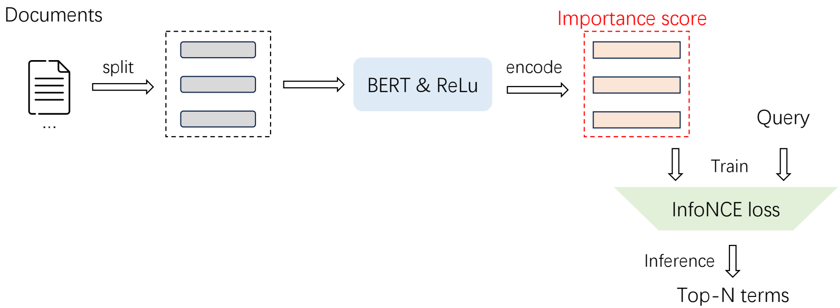
- **Any permutation** of the **term set** will be a **valid** identification for the corresponding document

Word-based: Important terms [Zhang et al., 2023]

- **Any permutation** of the **term set** will be a **valid** identification for the corresponding document
- Important terms: A set of document terms that have high importance scores

Important terms: AutoTSG [Zhang et al., 2023]

- Importance scores: The relevance scores of terms with respect to the query



"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

Important terms: AutoTSG [Zhang et al., 2023]

Docid repetition problem

- If the number of terms is sufficiently large, all documents within the corpus can be unique

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

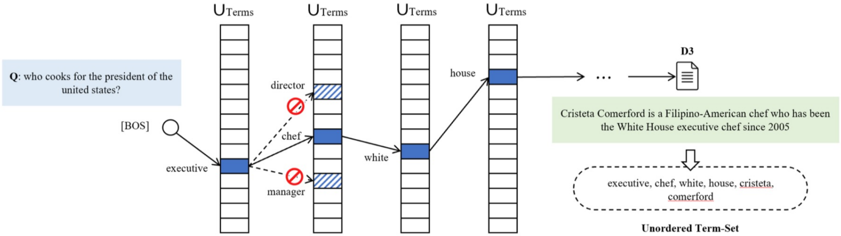
Important terms: AutoTSG [Zhang et al., 2023]

Docid repetition problem

- If the number of terms is sufficiently large, all documents within the corpus can be unique
- For a moderate-scale corpus like Natural Questions, specifying 12 terms is already sufficient to ensure uniqueness

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

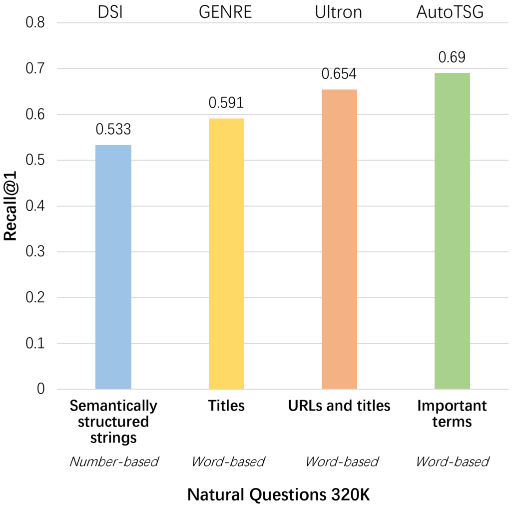
Important terms: AutoTSG [Zhang et al., 2023]



- Any permutation of the term-set docid will lead to the retrieval of the corresponding document

"Term-Sets Can Be Strong Document Identifiers For Auto-Regressive Search Engines". Zhang et al. [2023]

Performance comparisons



- Backbone: T5-base
- Using important term sets obtained through relevance matching as docids help represent the important information of the document
- This method also mitigates the issue of false pruning

Data source: Zhang et al. [2023]

Word-based docids: Summary

👍 Semantically related to the content of the document

Word-based docids: Summary

- Semantically related to the content of the document
- Good interpretability

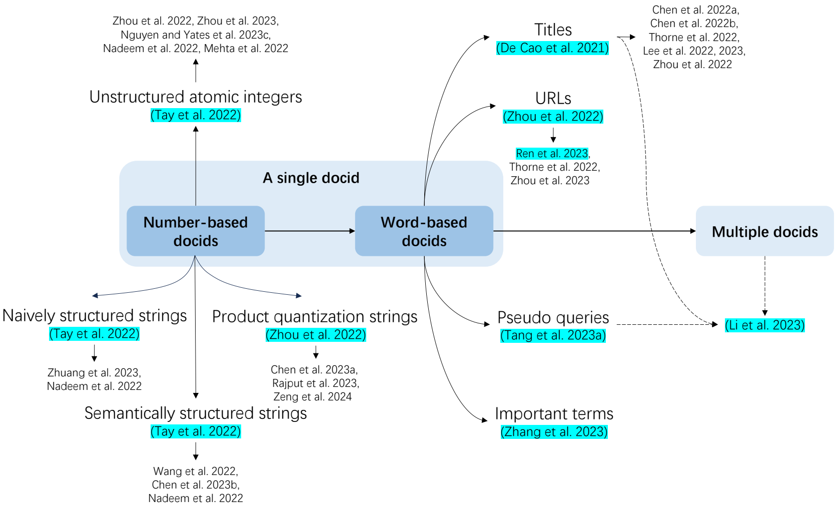
Word-based docids: Summary

- Semantically related to the content of the document
- Good interpretability
- Rely on metadata or labeled data

Word-based docids: Summary

- 👍 Semantically related to the content of the document
- 👍 Good interpretability
- 👎 Rely on metadata or labeled data
- 👎 May lead to duplication

A single docid: Summary



A single docid: Summary

👍 The design of a single docid is relatively straightforward

A single docid: Summary

- 👍 The design of a single docid is relatively straightforward
- 👍 The GR model may easily learn the one-to-one mapping relationship

A single docid: Summary

- 👍 The design of a single docid is relatively straightforward
- 👍 The GR model may easily learn the one-to-one mapping relationship
- 👎 These designs are typically short strings, providing limited information about the document

A single docid: Summary

- 👍 The design of a single docid is relatively straightforward
- 👍 The GR model may easily learn the one-to-one mapping relationship
- 👎 These designs are typically short strings, providing limited information about the document
- 👎 A single type of docid only represents a document from one view; and might be insufficient to effectively capture the entirety of the document's content

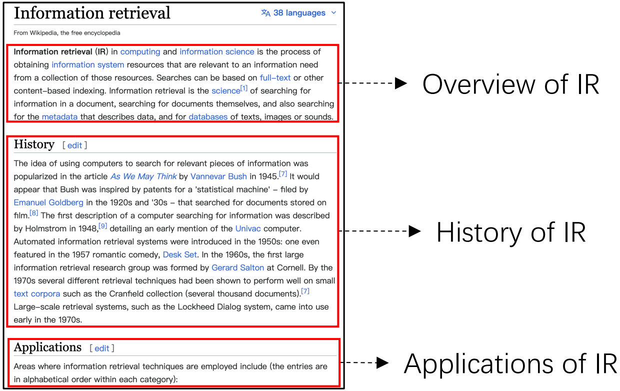
Multiple docids

- Multiple docids can provide complementary information from different views

Image source: Wikipedia

Multiple docids

- Multiple docids can provide complementary information from different views



Information retrieval

38 languages

From Wikipedia, the free encyclopedia

Information retrieval (**IR**) in computing and information science is the process of obtaining information system resources that are relevant to an information need from a collection of those resources. Searches can be based on full-text or other content-based indexing. Information retrieval is the science[1] of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds.

History [edit]

The idea of using computers to search for relevant pieces of information was popularized in the article *As We May Think* by Vannevar Bush in 1945.[7] It would appear that Bush was inspired by patents for a 'statistical machine' – filed by Emanuel Goldberg in the 1920s and '30s – that searched for documents stored on film.[8] The first description of a computer searching for information was described by Holmstrom in 1948,[9] detailing an early mention of the Univac computer. Automated information retrieval systems were introduced in the 1950s: one even featured in the 1957 romantic comedy, Desk Set. In the 1960s, the first large information retrieval research group was formed by Gerard Salton at Cornell. By the 1970s several different retrieval techniques had been shown to perform well on small text corpora such as the Cranfield collection (several thousand documents).[7] Large-scale retrieval systems, such as the Lockheed Dialog system, came into use early in the 1970s.

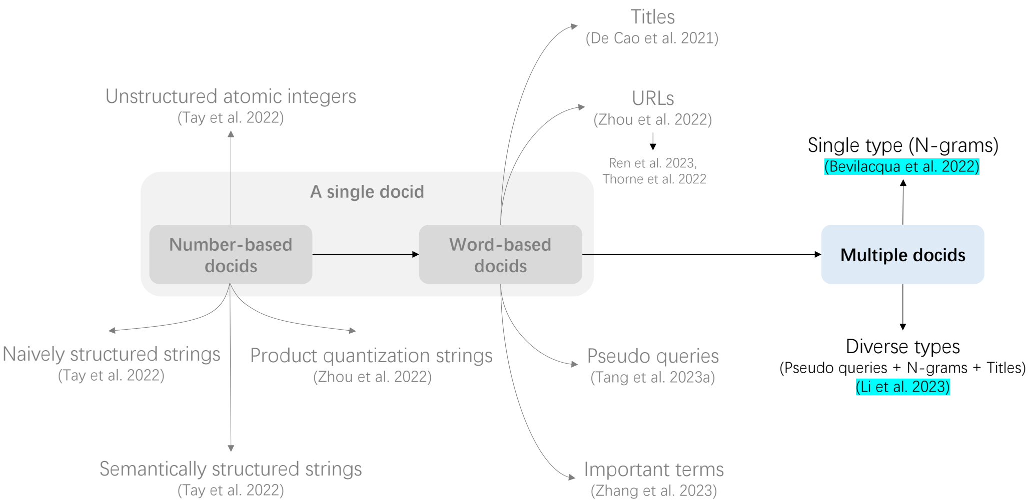
Applications [edit]

Areas where information retrieval techniques are employed include (the entries are in alphabetical order within each category):

Multi-view information

Image source: Wikipedia

Multiple docids



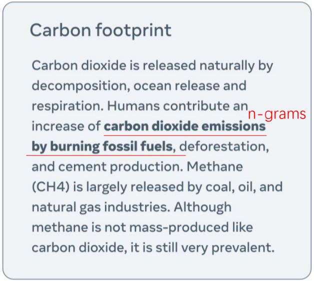
Multiple docids: Single type (N-grams) [Bevilacqua et al., 2022]

- All n-grams (i.e., substrings) in a document are treated as its possible docids

"Autoregressive Search Engines: Generating Substrings as Document Identifiers". Bevilacqua et al. [2022]

Multiple docids: Single type (N-grams) [Bevilacqua et al., 2022]

- All n-grams (i.e., substrings) in a document are treated as its possible docids
- Part of n-grams as docids during training: Only the terms from the document that have **a high overlap with the query** are chosen as the target docids



"Autoregressive Search Engines: Generating Substrings as Document Identifiers". Bevilacqua et al. [2022]

Multiple docids: Single type (N-grams) [Bevilacqua et al., 2022]

Docid repetition problem

- A heuristic scoring function is designed to address this during inference

"Autoregressive Search Engines: Generating Substrings as Document Identifiers". Bevilacqua et al. [2022]

Multiple docids: Single type (N-grams) [Bevilacqua et al., 2022]

Docid repetition problem

- A heuristic scoring function is designed to address this during inference

We will discuss this in Section 5!

"Autoregressive Search Engines: Generating Substrings as Document Identifiers". Bevilacqua et al. [2022]

Multiple docids: Single type (Important n-grams) [Chen et al., 2023]

- The **important n-grams** occurring in a document as its docids

"A Unified Generative Retriever for Knowledge-Intensive Language Tasks via Prompt Learning". Chen et al. [2023]

Multiple docids: Single type (Important n-grams) [Chen et al., 2023]

- The **important n-grams** occurring in a document as its docids
- N-gram importance is determined by the **relevance between n-grams and the query**:

"A Unified Generative Retriever for Knowledge-Intensive Language Tasks via Prompt Learning". Chen et al. [2023]

Multiple docids: Single type (Important n-grams) [Chen et al., 2023]

- The **important n-grams** occurring in a document as its docids
- N-gram importance is determined by the **relevance between n-grams and the query**:
  - Step 1: The query and its relevant document are concatenated with special delimiter tokens as a single input sequence
  - Step 2: Feed it into the original BERT model to get the [CLS] vector
  - Step 3: The token importance is computed by averaging the [CLS]-token attention weights
  - Step 4: The importance for the n-gram is the average of these tokens' importance

"A Unified Generative Retriever for Knowledge-Intensive Language Tasks via Prompt Learning". Chen et al. [2023]

Single type (Important n-grams) [Chen et al., 2023]: An example

ID for document retrieval

Important n-grams

1. was an American entrepreneur, industrial designer
2. Jobs was forced out of Apple
3. He died of respiratory arrest related

Steven Paul Jobs (February 24, 1955 – October 5, 2011) ***was an American entrepreneur, industrial designer***, business magnate, media proprietor, and investor.

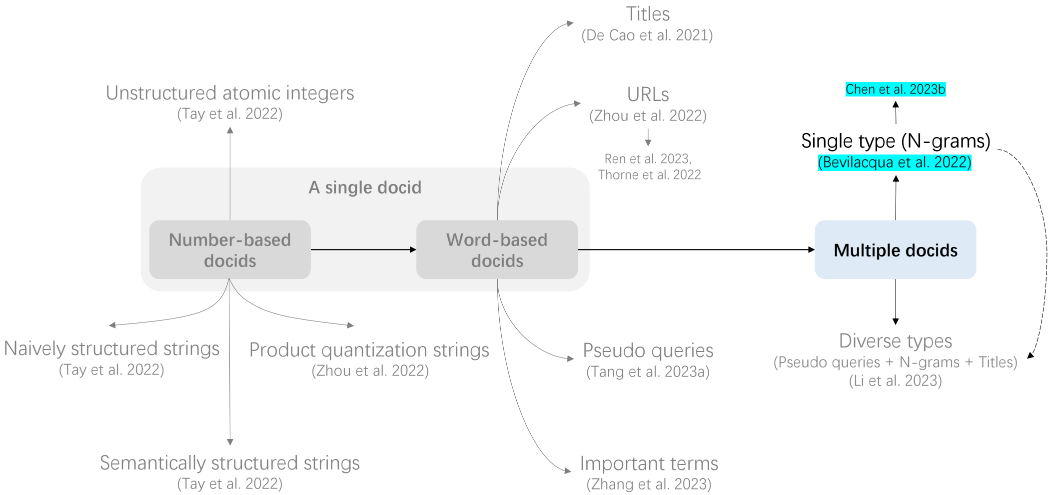
[…] In 1985, ***Jobs was forced out of Apple*** after a long power struggle with the company's board and its then-CEO John Sculley […]

In 2003, Jobs was diagnosed with a pancreatic neuroendocrine tumor. ***He died of respiratory arrest related*** to the tumor on October 5, 2011 at the age of 56.

- Countermeasure for docid repetition problem: Similar to Bevilacqua et al. [2022]

"A Unified Generative Retriever for Knowledge-Intensive Language Tasks via Prompt Learning". Chen et al. [2023]

Single type (N-grams) and subsequent work



Multiple docids: Diverse types (MINDER) [Li et al., 2023]

Query: Who is the singer of *does he love you?*

↑**Relevant**

Passage (*https://en.wikipedia.org/wiki/Does_He_Love_You*)

"Does He Love You" is a song written by Sandy Knox and Billy Stritch, and recorded as a duet by American country music artists Reba McEntire and Linda Davis. It was released in August 1993 as the first single from Reba's album "Greatest Hits Volume Two". It is one of country music's several songs about a love triangle. "Does He Love You" was written in 1982 by Billy Stritch. ......

Multiview Identifiers

Title: Does He Love You

Substrings: "Does He Love You" is a song ..., recorded as a duet by American country music artists Reba McEntire and Linda Davis, ...

Pseudo-queries:

Who wrote the song does he love you?

Who sings does he love you?

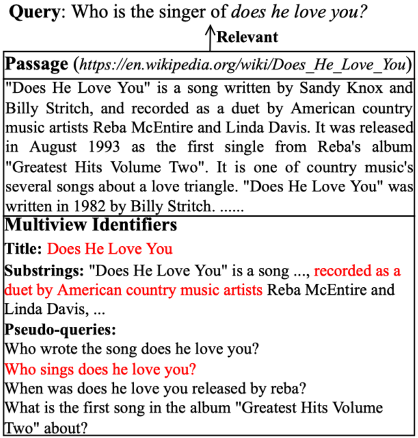
When was does he love you released by reba?

What is the first song in the album "Greatest Hits Volume Two" about?

- Three views of docids

"Multiview Identifiers Enhanced Generative Retrieval". Li et al. [2023]

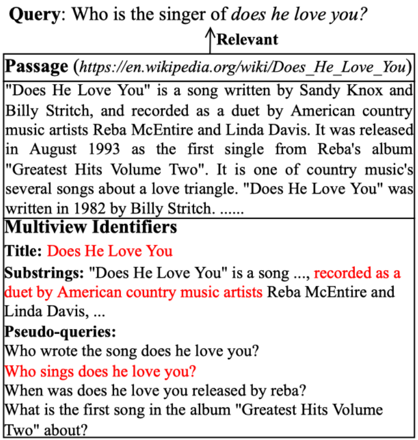
Multiple docids: Diverse types (MINDER) [Li et al., 2023]



- Three views of docids
  - Title: Indicate the subject of a document

"Multiview Identifiers Enhanced Generative Retrieval". Li et al. [2023]

Multiple docids: Diverse types (MINDER) [Li et al., 2023]



- Three views of docids
  - Title: Indicate the subject of a document
  - Substrings (N-grams): Be also semantically related

"Multiview Identifiers Enhanced Generative Retrieval". Li et al. [2023]

Multiple docids: Diverse types (MINDER) [Li et al., 2023]

Query: Who is the singer of *does he love you?*

↑**Relevant**

Passage (*https://en.wikipedia.org/wiki/Does_He_Love_You*)

"Does He Love You" is a song written by Sandy Knox and Billy Stritch, and recorded as a duet by American country music artists Reba McEntire and Linda Davis. It was released in August 1993 as the first single from Reba's album "Greatest Hits Volume Two". It is one of country music's several songs about a love triangle. "Does He Love You" was written in 1982 by Billy Stritch. ......

Multiview Identifiers

Title: Does He Love You

Substrings: "Does He Love You" is a song ..., recorded as a duet by American country music artists Reba McEntire and Linda Davis, ...

Pseudo-queries:

Who wrote the song does he love you?

Who sings does he love you?

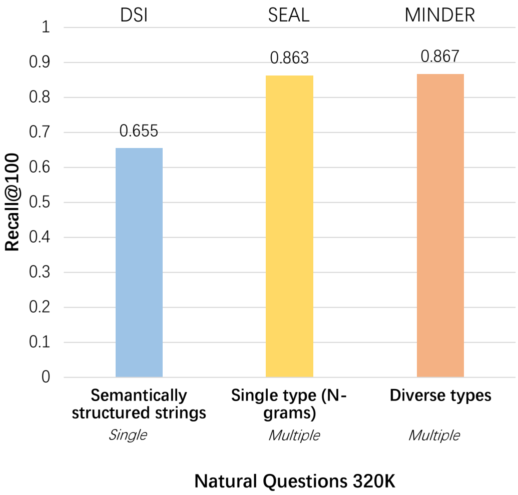
When was does he love you released by reba?

What is the first song in the album "Greatest Hits Volume Two" about?

- Three views of docids
  - Title: Indicate the subject of a document
  - Substrings (N-grams): Be also semantically related
  - Pseudo-queries: Integrate multiple segments and contextualized information

"Multiview Identifiers Enhanced Generative Retrieval". Li et al. [2023]

Performance comparisons



Data source: Li et al. [2023]

- Backbone: BART-large
- Results: Using multiple docids for a document yields better results than using a single docid

Multiple docids: Summary

👍 Multiple docids can provide a more comprehensive representation of the document, assisting the model in gaining a multifaceted understanding

Multiple docids: Summary

👍 Multiple docids can provide a more comprehensive representation of the document, assisting the model in gaining a multifaceted understanding

👍 Similar docids across different documents can reflect the similarity between the documents

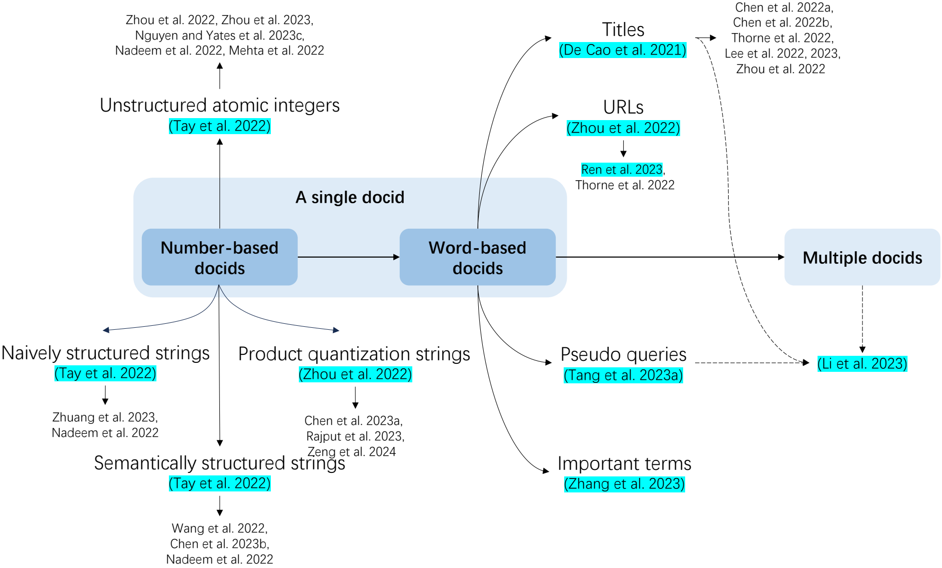
Multiple docids: Summary

- 👍 Multiple docids can provide a more comprehensive representation of the document, assisting the model in gaining a multifaceted understanding
- 👍 Similar docids across different documents can reflect the similarity between the documents
- 👎 GR models with the increased docid numbers demand more memory usage and inference time

Multiple docids: Summary

- Multiple docids can provide a more comprehensive representation of the document, assisting the model in gaining a multifaceted understanding
- Similar docids across different documents can reflect the similarity between the documents
- GR models with the increased docid numbers demand more memory usage and inference time
- It is challenging to design discriminative multiple docids for a document

Pre-defined static docids: Summary



Pre-defined static docids: Summary

Docid type		Construction	Uniqueness	The degree of semantic connection to the document	Relying on labeled data	Relying on metadata
A single docid: Number-based	Unstructured atomic integers (Tay et al. 2022)	Easy	Yes	None	No	No
	Naively structured strings (Tay et al. 2022)	Easy	Yes	None	No	No
	Semantically structured strings (Tay et al. 2022)	Moderate	Yes	Weak	No	No
	Product quantization strings (Zhou et al. 2022)	Moderate	No	Moderate	No	No
A single docid: Word-based	Titles (De Cao et al. 2021)	Easy	No	Strong	No	Yes
	URLs (Zhou et al. 2022, Ren et al. 2023)	Easy	Yes	Strong	No	Yes
	Pseudo queries (Tang et al. 2023a)	Moderate	No	Strong	Yes	No
	Important terms (Zhang et al. 2023)	Hard	Yes	Strong	Yes	No
Multiple docids	Single type: N-grams (Bevilacqua et al. 2022)	Easy	No	Moderate	No	No
	Diverse types (Li et al. 2023)	Moderate	No	Strong	Yes	Yes

Pre-defined static docids: Obvious constrains

Not specifically optimized for retrieval tasks

Pre-defined static docids: Obvious constrains

- Not specifically optimized for retrieval tasks
- Difficult to learn semantics and relationships between documents

How to design learnable docids tailored for retrieval tasks?

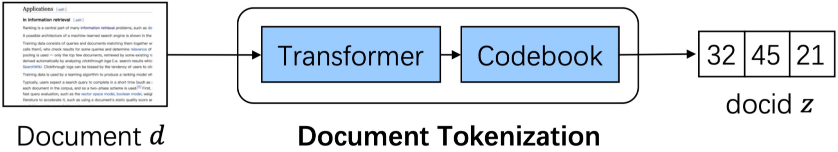
Learnable docids

- **Repeatable docids**:
  - GenRet [Sun et al., 2023] learns to tokenize documents into short discrete representations via a discrete auto-encoding, jointly training with the retrieval task
  - ASI [Yang et al., 2023] combines both the end-to-end learning of docids for existing and new documents and the end-to-end document retrieval based joint optimization

Learnable docids

- **Repeatable docids**:
  - GenRet [Sun et al., 2023] learns to tokenize documents into short discrete representations via a discrete auto-encoding, jointly training with the retrieval task
  - ASI [Yang et al., 2023] combines both the end-to-end learning of docids for existing and new documents and the end-to-end document retrieval based joint optimization
- **Unique docids**:
  - NOVO [Wang et al., 2023] uses unique n-gram sets identifying each document and can be generated in any order and can be optimized through retrieval tasks

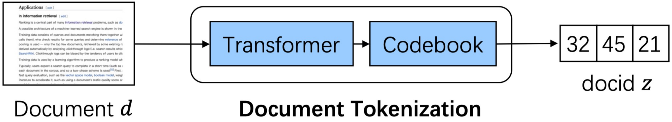
Repeatable learnable docids: GenRet [Sun et al., 2023]



- Docid: A sequence of discrete numbers is the docid for a given document converted by a document tokenization model
- Training: Jointly training with a document tokenization task, reconstruction task and retrieval task

"Learning to Tokenize for Generative Retrieval". Sun et al. [2023]

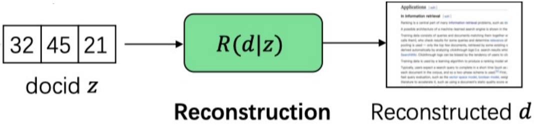
Repeatable learnable docids: GenRet [Sun et al., 2023]



- Document tokenization task: Produce docids for documents

"Learning to Tokenize for Generative Retrieval". Sun et al. [2023]

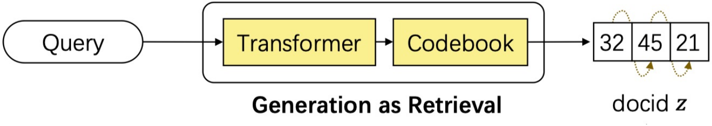
Repeatable learnable docids: GenRet [Sun et al., 2023]



- Reconstruction task: Learn to reconstruct a document based on a docid

"Learning to Tokenize for Generative Retrieval". Sun et al. [2023]

Repeatable learnable docids: GenRet [Sun et al., 2023]



- Retrieval task: Generate relevant docids directly for a query

"Learning to Tokenize for Generative Retrieval". Sun et al. [2023]

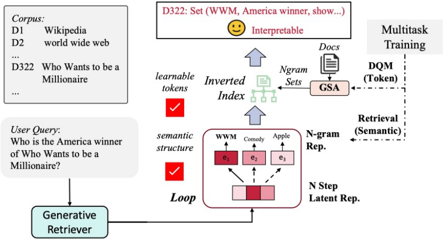
Repeatable learnable docids: GenRet [Sun et al., 2023]

Docid repetition problem

- All corresponding documents are retrieved and shuffled in an arbitrary order

"Learning to Tokenize for Generative Retrieval". Sun et al. [2023]

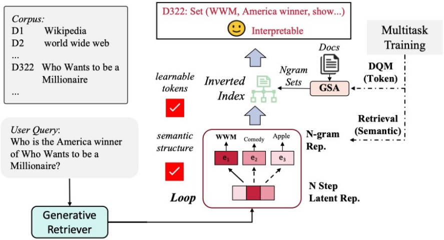
Unique learnable docids: NOVO [Wang et al., 2023]



- Docid: Unique n-grams sets of the documents obtained from global self-attention

"NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR". Wang et al. [2023]

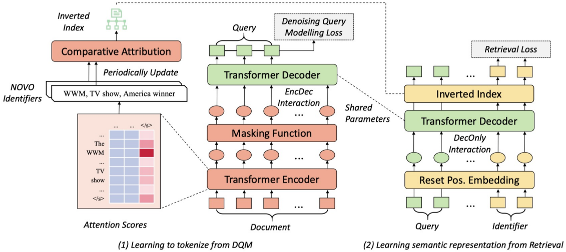
Unique learnable docids: NOVO [Wang et al., 2023]



- Docid: Unique n-grams sets of the documents obtained from global self-attention
- Decoding: A document can be retrieved by generating its n-grams in the sets in any order

"NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR". Wang et al. [2023]

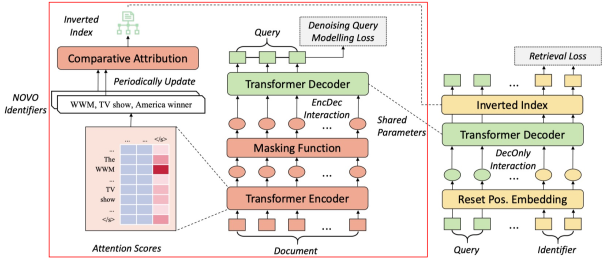
Unique learnable docids: NOVO [Wang et al., 2023]



- Docids are learned by the denoising query modeling task and retrieval task jointly

"NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR". Wang et al. [2023]

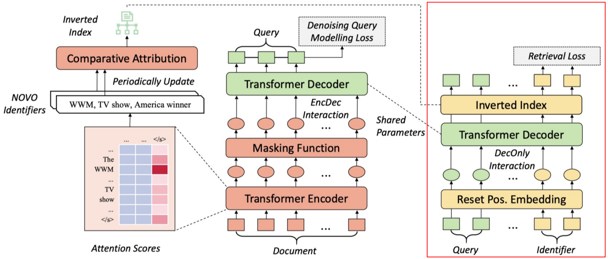
Unique learnable docids: NOVO [Wang et al., 2023]



- Denoising query modeling task: By learning to generate queries with noisy documents, n-grams that are more relevant to the query are may be filtered out

"NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR". Wang et al. [2023]

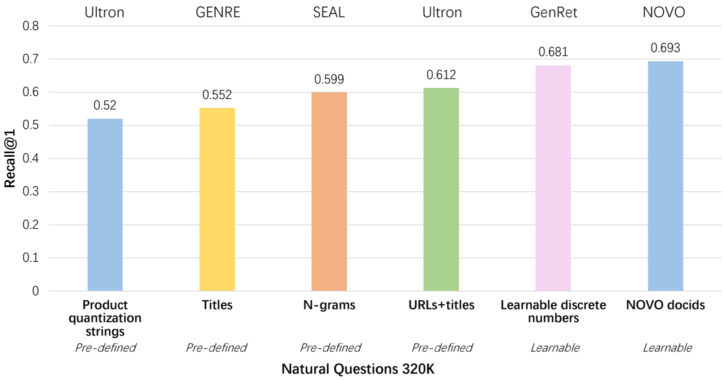
Unique learnable docids: NOVO [Wang et al., 2023]



- Retrieval task: The model learns the mapping from the query to relevant docids to update docid semantics

"NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR". Wang et al. [2023]

Performance comparisons



- Backbone: T5-base
- Results: Two learnable docids yields better results than partial pre-defined static docids

Data source: Wang et al. [2023]

Learnable docids: Summary

- It can be optimized together with the ultimate goal of GR to better adapt to retrieval

Learnable docids: Summary

- 👍 It can be optimized together with the ultimate goal of GR to better adapt to retrieval
- 👍 A learnable approach can enable number-based docids like those in GenRet [Sun et al., 2023] to perform well

Learnable docids: Summary

- It can be optimized together with the ultimate goal of GR to better adapt to retrieval
- A learnable approach can enable number-based docids like those in GenRet [Sun et al., 2023] to perform well
- It relies on complex task design for learning

Learnable docids: Summary

- 👍 It can be optimized together with the ultimate goal of GR to better adapt to retrieval
- 👍 A learnable approach can enable number-based docids like those in GenRet [Sun et al., 2023] to perform well
- 👎 It relies on complex task design for learning
- 👎 The learning process is complex, as docids change and require iterative learning

Docid design: Summary

- **Shall we use randomize numbers as the docids?**
  - Random number strings can serve as docids, but their effectiveness is limited

Docid design: Summary

- **Shall we use randomize numbers as the docids?**
  - Random number strings can serve as docids, but their effectiveness is limited
- **How to construct proper docids for the documents?**
  - Designing predefined or learnable docids based on the semantics of the documents

Docid design: Summary

- **Shall we use randomize numbers as the docids?**
  - Random number strings can serve as docids, but their effectiveness is limited
- **How to construct proper docids for the documents?**
  - Designing predefined or learnable docids based on the semantics of the documents
- **Would the choices of different docids affect the model performance(effectiveness, capacity, etc.)?**
  - The length and quantity of docids both impact the effectiveness of the model's performance
  - The influence on capacity is yet to be explored

Docid design: Summary

Docid type				
Pre-defined	Single	Number-based	- Simplified construction	- Low interpretability - Moderate performance
		Word-based	- High interpretability - Good performance	- Single-perspective representation of documents
	Multiple		- Comprehensive document representations - Better performance	- Slightly more complex construction
Learnable			- Adapting to GR objectives - Best performance	- Complex learning process

Docid design: Summary

Docid type				
Pre-defined	Single	Number-based	- Simplified construction	- Low interpretability - Moderate performance
		Word-based	- High interpretability - Good performance	- Single-perspective representation of documents
	Multiple		Comprehensive document representations - Better performance	- Slightly more complex construction
Learnable			Adapting to GR objectives - Best performance	- Complex learning process

Docid design: Summary

Docid type				
Pre-defined	Single	Number-based	- Simplified construction	- Low interpretability - Moderate performance
		Word-based	- High interpretability - Good performance	- Single-perspective representation of documents
	Multiple		Comprehensive document representations - Better performance	- Slightly more complex construction
Learnable			Adapting to GR objectives - Best performance	- Complex learning process

Based on these docids

Model training $\rightarrow$ **Section 4!**

Model inference $\rightarrow$ **Section 5!**

Coffee break

References

References i

M. Bevilacqua, G. Ottaviano, P. Lewis, W.-t. Yih, S. Riedel, and F. Petroni. Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*, pages 31668–31683, 2022.

J. Chen, R. Zhang, J. Guo, M. de Rijke, Y. Liu, Y. Fan, and X. Cheng. A unified generative retriever for knowledge-intensive language tasks via prompt learning. In *46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1448–1457, 2023.

N. De Cao, G. Izacard, S. Riedel, and F. Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021.

Y. Li, N. Yang, L. Wang, F. Wei, and W. Li. Multiview identifiers enhanced generative retrieval. In *61st Annual Meeting of the Association for Computational Linguistics*, pages 6636–6648, 2023.

R. Ren, W. X. Zhao, J. Liu, H. Wu, J.-R. Wen, and H. Wang. Tome: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 6102–6114, 2023.

W. Sun, L. Yan, Z. Chen, S. Wang, H. Zhu, P. Ren, Z. Chen, D. Yin, M. de Rijke, and Z. Ren. Learning to tokenize for generative retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Y. Tang, R. Zhang, J. Guo, J. Chen, Z. Zhu, S. Wang, D. Yin, and X. Cheng. Semantic-enhanced differentiable search index inspired by learning strategies. In *29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

Y. Tay, V. Q. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. Gupta, T. Schuster, W. W. Cohen, and D. Metzler. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*, volume 35, pages 21831–21843, 2022.

Z. Wang, Y. Zhou, Y. Tu, and Z. Dou. Novo: Learnable and interpretable document identifiers for model-based ir. In *Proceedings of the 32nd ACM Conference on Information and Knowledge Management*, 2023.

T. Yang, M. Song, Z. Zhang, H. Huang, W. Deng, F. Sun, and Q. Zhang. Auto search indexer for end-to-end document retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

References iii

P. Zhang, Z. Liu, Y. Zhou, Z. Dou, and Z. Cao. Term-sets can be strong document identifiers for auto-regressive search engines. *arXiv preprint arXiv:2305.13859*, 2023.

Y. Zhou, J. Yao, Z. Dou, L. Wu, P. Zhang, and J.-R. Wen. Ultron: An ultimate retriever on corpus with a model-based indexer. *arXiv preprint arXiv:2208.09257*, 2022.