No-GPU Full Context Secure Multi-LLM
with Real-Time Fine-Tuning & Explainable AI

# From 10 Terabytes to Zero Parameter: The LLM 2.0 Revolution

Vincent Granville, PhD
Chief AI Architect
BondingAI.io
vincent@BondingAI.io

May 15, 2025, 2024

# Agenda

1. Why xLLM? What is it?
2. xLLM Innovative Features
3. xLLM Architecture and Evaluation
4. xLLM for Clustering, Data Synthetization, Predictive Analytics
5. What is coming UP next?
6. References

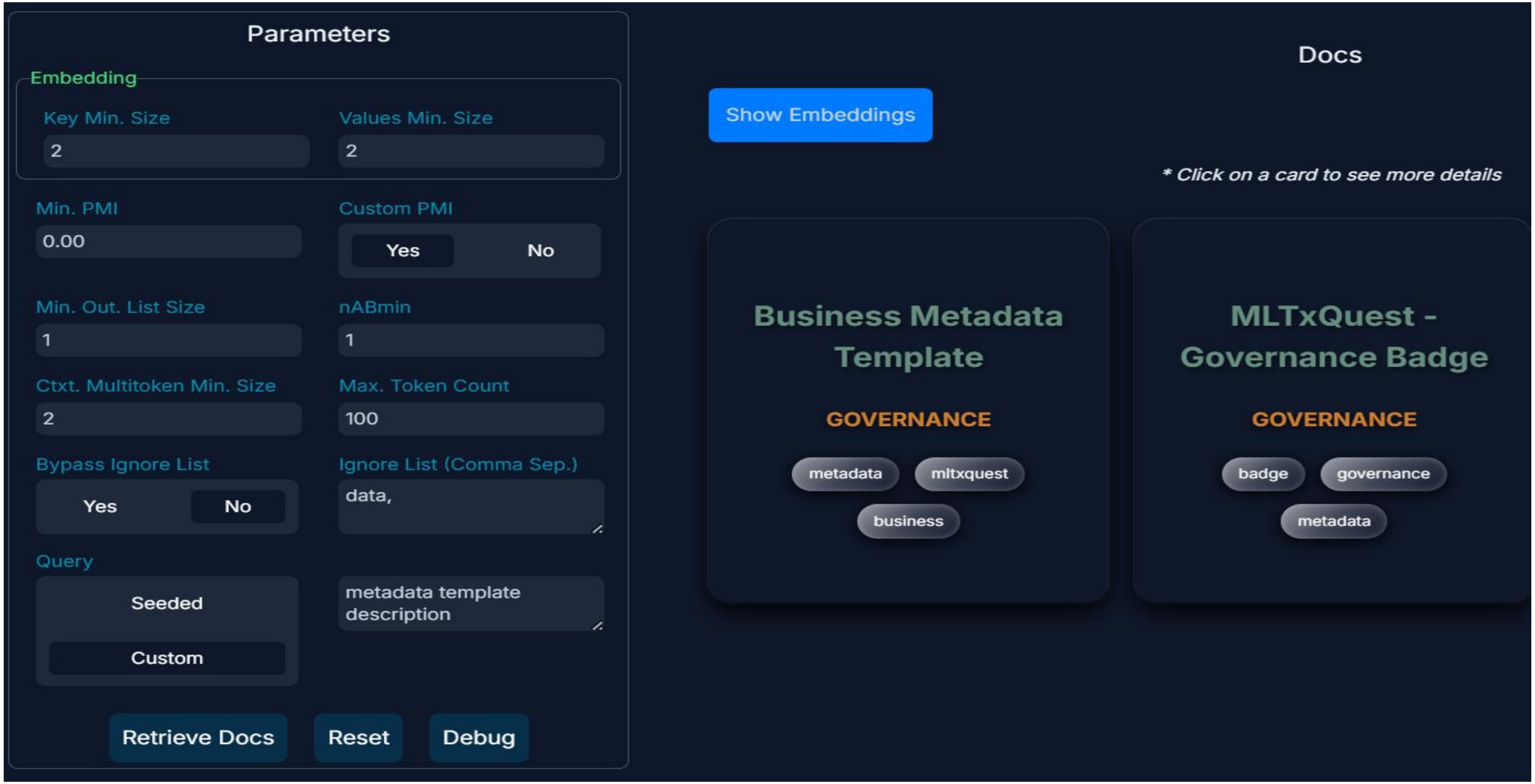
# Part 1
# Why xLLM? What is it?

# Extreme LLM (xLLM) in a Nutshell

**xLLM mission is to enable Enterprises** to build their on LLMs that fits their purpose with precision, faster, cheaper with security as well open to integrate with any other LLMs…

- **Mixture of experts**
  - Specialized sub-LLM and/or sub-LLMs for authorized users
  - LLM router to manage the sub-LLMs
  - User selects sun-LLM, agent, and hyperparameters
  - Each sub-LLM built with its own taxonomy and knowledge graph
- **No neural network, no training**
  - Thus, low cost, easy to fine-tune in real-time
  - Self-tuned based on favorite hyperparameters, and customizable
  - No GPU, no latency, exhaustive concise results, local implementation
- **Concise results**
  - Multiple sections displayed to user: links, related content, x-embeddings
  - Output with relevancy score attached to each item in each section; User offered choices for deeper or alternate queries
  - Great for search, for professional users and experts. Not just a “prompt box”; many options in the UI, like a mini browser
- **Case studies**
  - Corporate corpus with augmented sources (content + taxonomies)
  - Wolfram corpus: 15 sub-LLMs, 500 sub-categories per sub-LLM
  - Publisher, 4000 titles: clustering, predicting article performance

# Prompt Results – Card Format (web API)



Parameters

Embedding

Key Min. Size: 2

Values Min. Size: 2

Min. PMI: 0.00

Custom PMI: [✓ Yes] No

Min. Out. List Size: 1

nABmin: 1

Ctxt. Multitoken Min. Size: 2

Max. Token Count: 100

Bypass Ignore List: Yes [✓ No]

Ignore List (Comma Sep.): data,

Query: Seeded [✓ Custom]

metadata template description

Docs

*Click on a card to see more details*

Business Metadata Template

GOVERNANCE

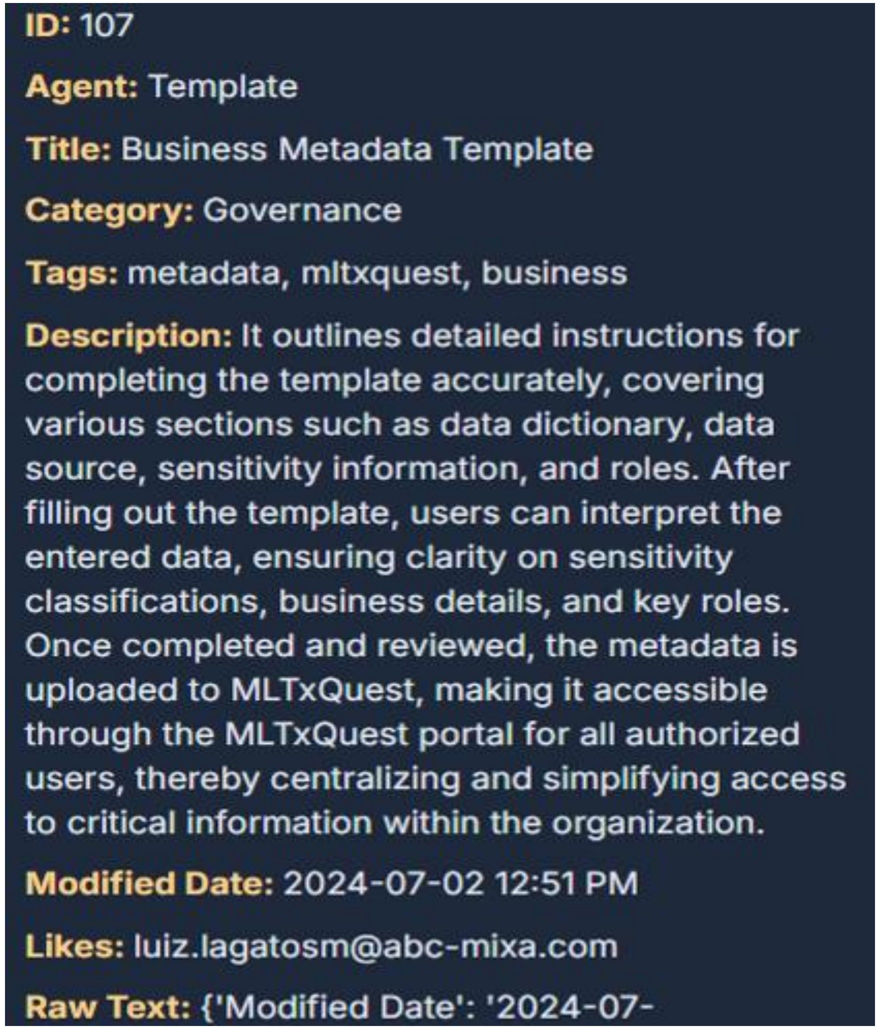
metadata mltxquest business

MLTxQuest - Governance Badge

GOVERNANCE

badge governance metadata

# Prompt Results – Card Format (web API)



**ID:** 107

**Agent:** Template

**Title:** Business Metadata Template

**Category:** Governance

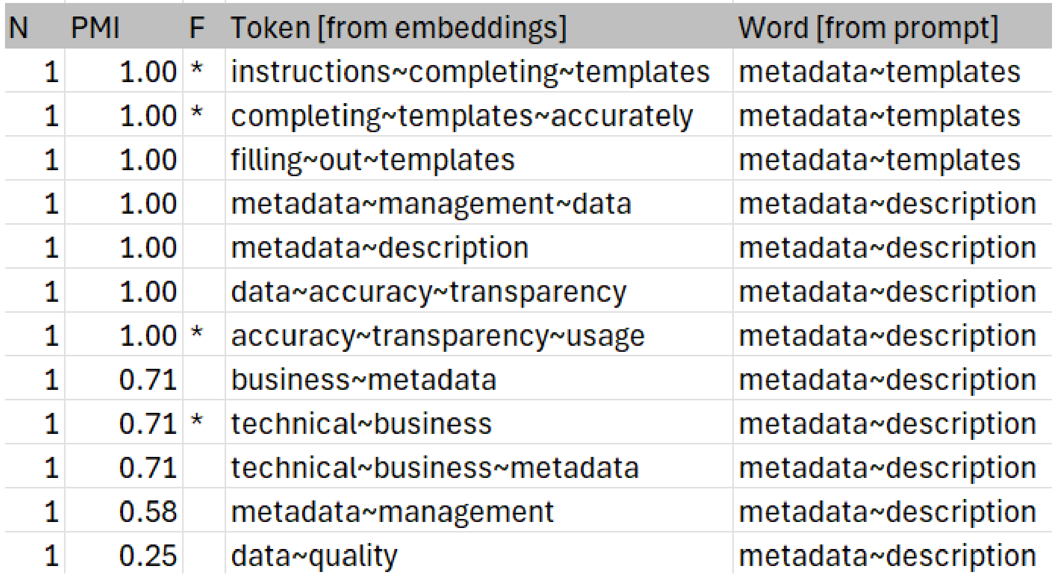
**Tags:** metadata, mltxquest, business

**Description:** It outlines detailed instructions for completing the template accurately, covering various sections such as data dictionary, data source, sensitivity information, and roles. After filling out the template, users can interpret the entered data, ensuring clarity on sensitivity classifications, business details, and key roles. Once completed and reviewed, the metadata is uploaded to MLTxQuest, making it accessible through the MLTxQuest portal for all authorized users, thereby centralizing and simplifying access to critical information within the organization.

**Modified Date:** 2024-07-02 12:51 PM

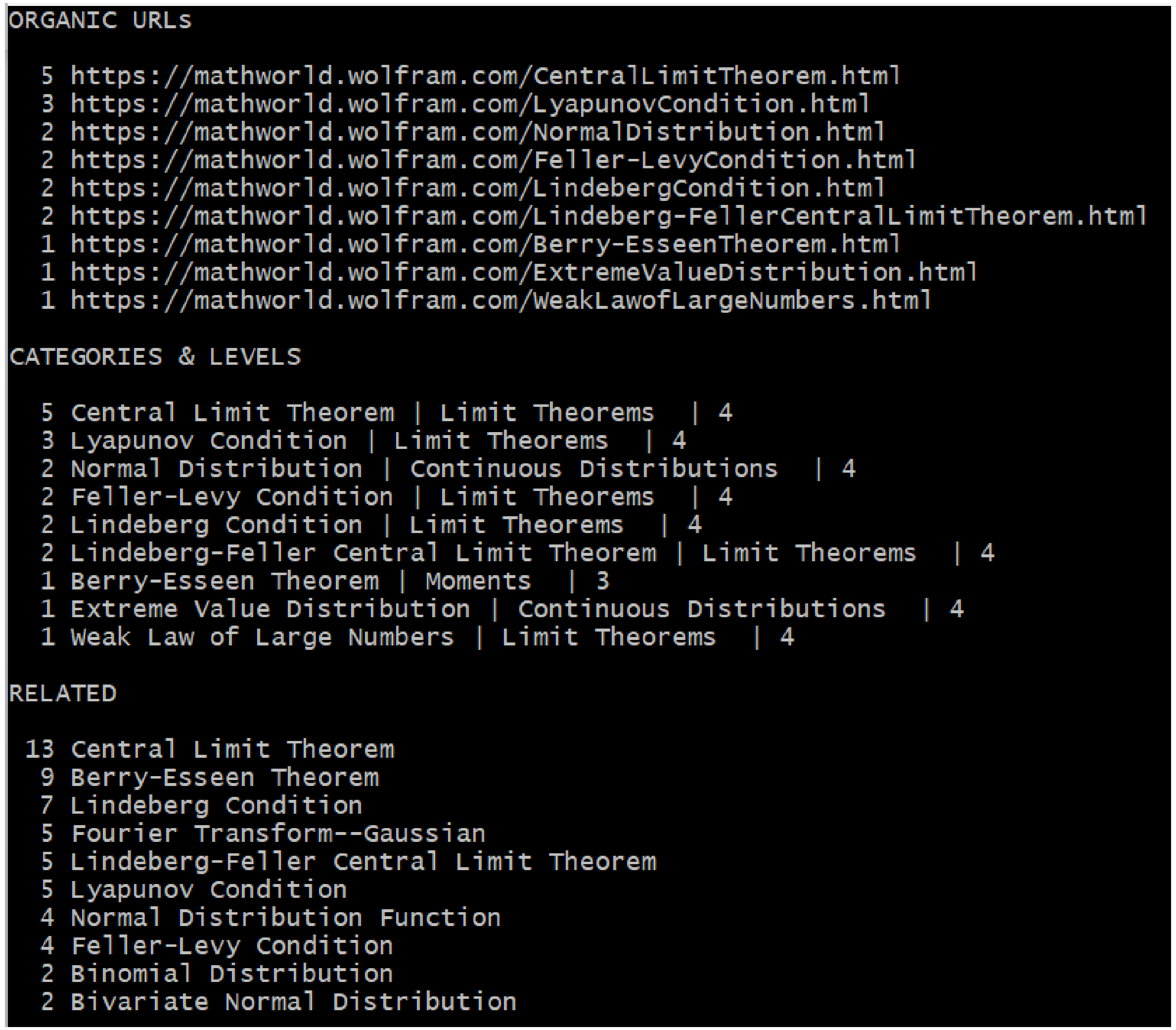
**Likes:** luiz.lagatosm@abc-mixa.com

**Raw Text:** {'Modified Date': '2024-07-

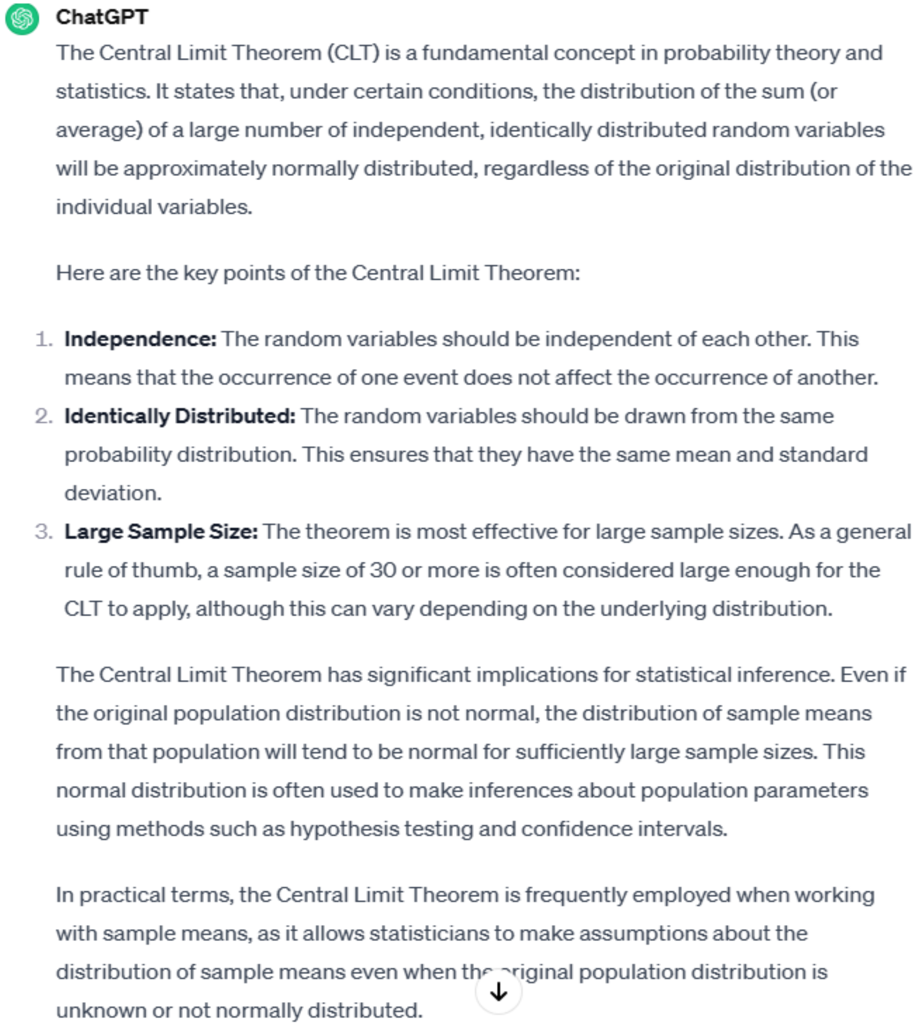


| N | PMI | F | Token [from embeddings] | Word [from prompt] |
|---|---|---|---|---|
| 1 | 1.00 | * | instructions~completing~templates | metadata~templates |
| 1 | 1.00 | * | completing~templates~accurately | metadata~templates |
| 1 | 1.00 | | filling~out~templates | metadata~templates |
| 1 | 1.00 | | metadata~management~data | metadata~description |
| 1 | 1.00 | | metadata~description | metadata~description |
| 1 | 1.00 | | data~accuracy~transparency | metadata~description |
| 1 | 1.00 | * | accuracy~transparency~usage | metadata~description |
| 1 | 0.71 | | business~metadata | metadata~description |
| 1 | 0.71 | * | technical~business | metadata~description |
| 1 | 0.71 | | technical~business~metadata | metadata~description |
| 1 | 0.58 | | metadata~management | metadata~description |
| 1 | 0.25 | | data~quality | metadata~description |

# Prompt Results – Listing Format (1)



```
ORGANIC URLs

  5 https://mathworld.wolfram.com/CentralLimitTheorem.html
  3 https://mathworld.wolfram.com/LyapunovCondition.html
  2 https://mathworld.wolfram.com/NormalDistribution.html
  2 https://mathworld.wolfram.com/Feller-LevyCondition.html
  2 https://mathworld.wolfram.com/LindebergCondition.html
  2 https://mathworld.wolfram.com/Lindeberg-FellerCentralLimitTheorem.html
  1 https://mathworld.wolfram.com/Berry-EsseenTheorem.html
  1 https://mathworld.wolfram.com/ExtremeValueDistribution.html
  1 https://mathworld.wolfram.com/WeakLawofLargeNumbers.html

CATEGORIES & LEVELS

  5 Central Limit Theorem | Limit Theorems   | 4
  3 Lyapunov Condition | Limit Theorems   | 4
  2 Normal Distribution | Continuous Distributions   | 4
  2 Feller-Levy Condition | Limit Theorems   | 4
  2 Lindeberg Condition | Limit Theorems   | 4
  2 Lindeberg-Feller Central Limit Theorem | Limit Theorems   | 4
  1 Berry-Esseen Theorem | Moments   | 3
  1 Extreme Value Distribution | Continuous Distributions   | 4
  1 Weak Law of Large Numbers | Limit Theorems   | 4

RELATED

 13 Central Limit Theorem
  9 Berry-Esseen Theorem
  7 Lindeberg Condition
  5 Fourier Transform--Gaussian
  5 Lindeberg-Feller Central Limit Theorem
  5 Lyapunov Condition
  4 Normal Distribution Function
  4 Feller-Levy Condition
  2 Binomial Distribution
  2 Bivariate Normal Distribution
```



**ChatGPT**

The Central Limit Theorem (CLT) is a fundamental concept in probability theory and statistics. It states that, under certain conditions, the distribution of the sum (or average) of a large number of independent, identically distributed random variables will be approximately normally distributed, regardless of the original distribution of the individual variables.

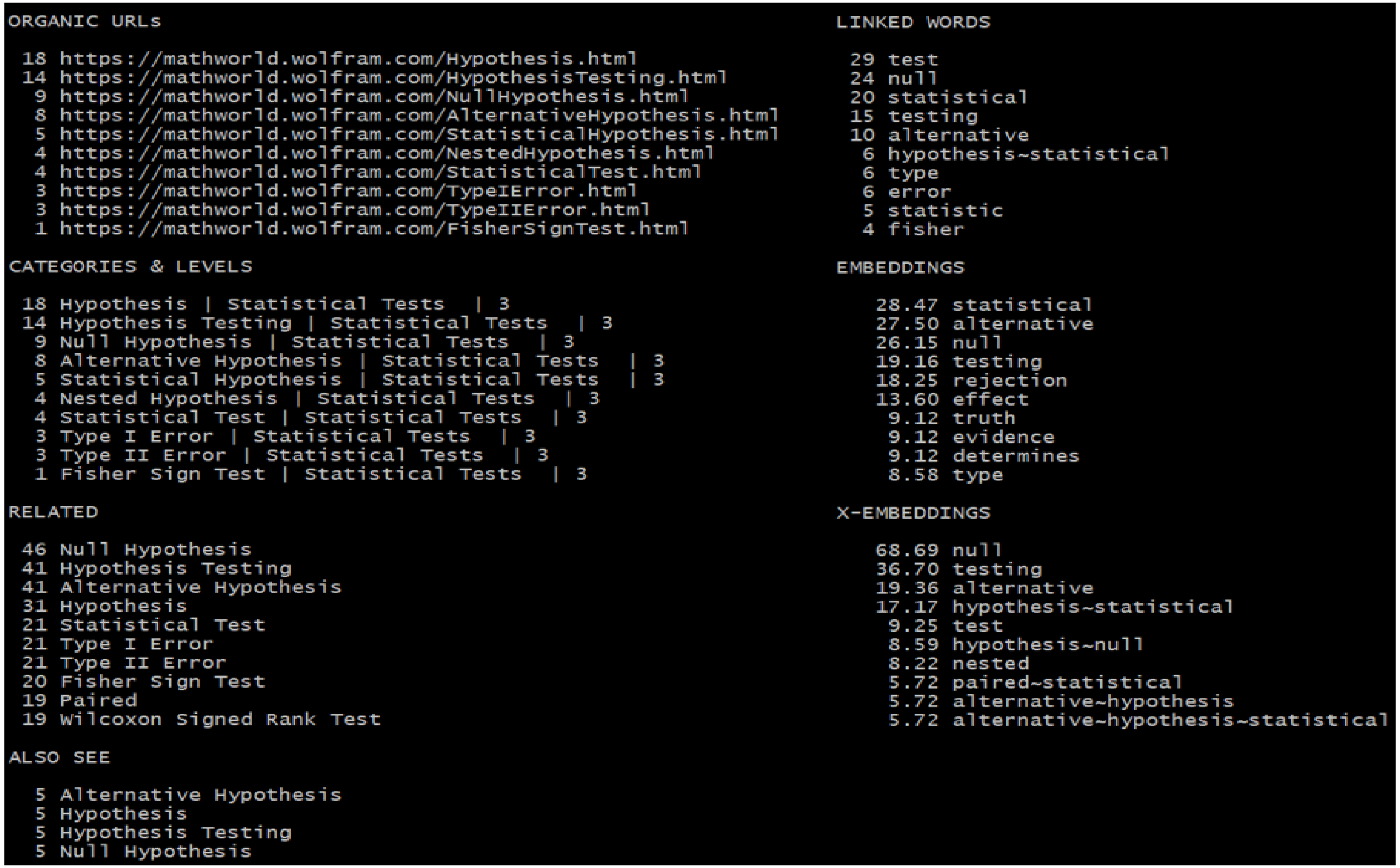
Here are the key points of the Central Limit Theorem:

1. **Independence:** The random variables should be independent of each other. This means that the occurrence of one event does not affect the occurrence of another.
2. **Identically Distributed:** The random variables should be drawn from the same probability distribution. This ensures that they have the same mean and standard deviation.
3. **Large Sample Size:** The theorem is most effective for large sample sizes. As a general rule of thumb, a sample size of 30 or more is often considered large enough for the CLT to apply, although this can vary depending on the underlying distribution.

The Central Limit Theorem has significant implications for statistical inference. Even if the original population distribution is not normal, the distribution of sample means from that population will tend to be normal for sufficiently large sample sizes. This normal distribution is often used to make inferences about population parameters using methods such as hypothesis testing and confidence intervals.

In practical terms, the Central Limit Theorem is frequently employed when working with sample means, as it allows statisticians to make assumptions about the distribution of sample means even when the original population distribution is unknown or not normally distributed.

# Prompt Results – Listing Format (2)



```
ORGANIC URLs

 18 https://mathworld.wolfram.com/Hypothesis.html
 14 https://mathworld.wolfram.com/HypothesisTesting.html
  9 https://mathworld.wolfram.com/NullHypothesis.html
  8 https://mathworld.wolfram.com/AlternativeHypothesis.html
  5 https://mathworld.wolfram.com/StatisticalHypothesis.html
  4 https://mathworld.wolfram.com/NestedHypothesis.html
  4 https://mathworld.wolfram.com/StatisticalTest.html
  3 https://mathworld.wolfram.com/TypeIError.html
  3 https://mathworld.wolfram.com/TypeIIError.html
  1 https://mathworld.wolfram.com/FisherSignTest.html

CATEGORIES & LEVELS

 18 Hypothesis | Statistical Tests  | 3
 14 Hypothesis Testing | Statistical Tests  | 3
  9 Null Hypothesis | Statistical Tests  | 3
  8 Alternative Hypothesis | Statistical Tests  | 3
  5 Statistical Hypothesis | Statistical Tests  | 3
  4 Nested Hypothesis | Statistical Tests  | 3
  4 Statistical Test | Statistical Tests  | 3
  3 Type I Error | Statistical Tests  | 3
  3 Type II Error | Statistical Tests  | 3
  1 Fisher Sign Test | Statistical Tests  | 3

RELATED

 46 Null Hypothesis
 41 Hypothesis Testing
 41 Alternative Hypothesis
 31 Hypothesis
 21 Statistical Test
 21 Type I Error
 21 Type II Error
 20 Fisher Sign Test
 19 Paired
 19 Wilcoxon Signed Rank Test

ALSO SEE

  5 Alternative Hypothesis
  5 Hypothesis
  5 Hypothesis Testing
  5 Null Hypothesis
```

```
LINKED WORDS

 29 test
 24 null
 20 statistical
 15 testing
 10 alternative
  6 hypothesis~statistical
  6 type
  6 error
  5 statistic
  4 fisher

EMBEDDINGS

   28.47 statistical
   27.50 alternative
   26.15 null
   19.16 testing
   18.25 rejection
   13.60 effect
    9.12 truth
    9.12 evidence
    9.12 determines
    8.58 type

X-EMBEDDINGS

   68.69 null
   36.70 testing
   19.36 alternative
   17.17 hypothesis~statistical
    9.25 test
    8.59 hypothesis~null
    8.22 nested
    5.72 paired~statistical
    5.72 alternative~hypothesis
    5.72 alternative~hypothesis~statistical
```

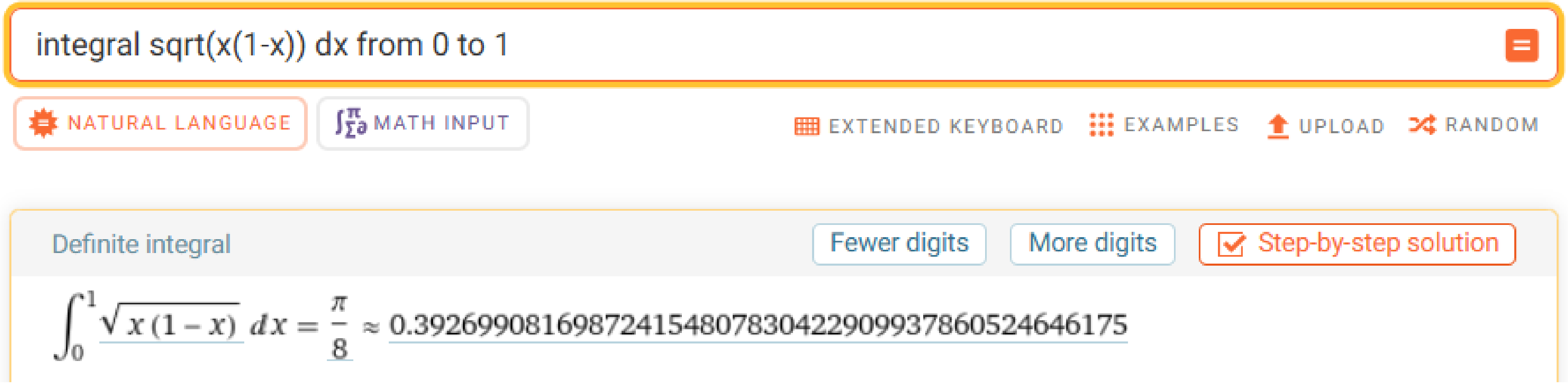
| | |
|---|---|
| Query | Hypotheses |
| Sub-LLM | Stats & Proba |
| Corpus | Wolfram Math |

# Prompt Results – Text Format

- Text entities retrieved from corpus via context chunking / indexation
  - Blended with images, datasets, URLs and so on (multimodal)
  - Knowledge graph elements included: categories, tags, related content, agents
- Generating English output (prose) with GenAI
  - Coming soon, different from simple text retrieval
  - Turn output into English summary
  - Pre-made customizable synthetic answers (template answers)
  - Blending AI with classic ML: integrating external tools  with large list of pre-made, customizable template sentences to display in prompt results

# xLLM Integration with other APIs and LLMs

- Leverage and blend capabilities of multiple LLMs (GPT, Perplexity, Mistral, etc..)
- Use external GenAI tools or libraries to turn output into nice, fluid English text
- CodeValet API for code generation
- Wolfram/Mathematica API to solve math problems



integral sqrt(x(1-x)) dx from 0 to 1

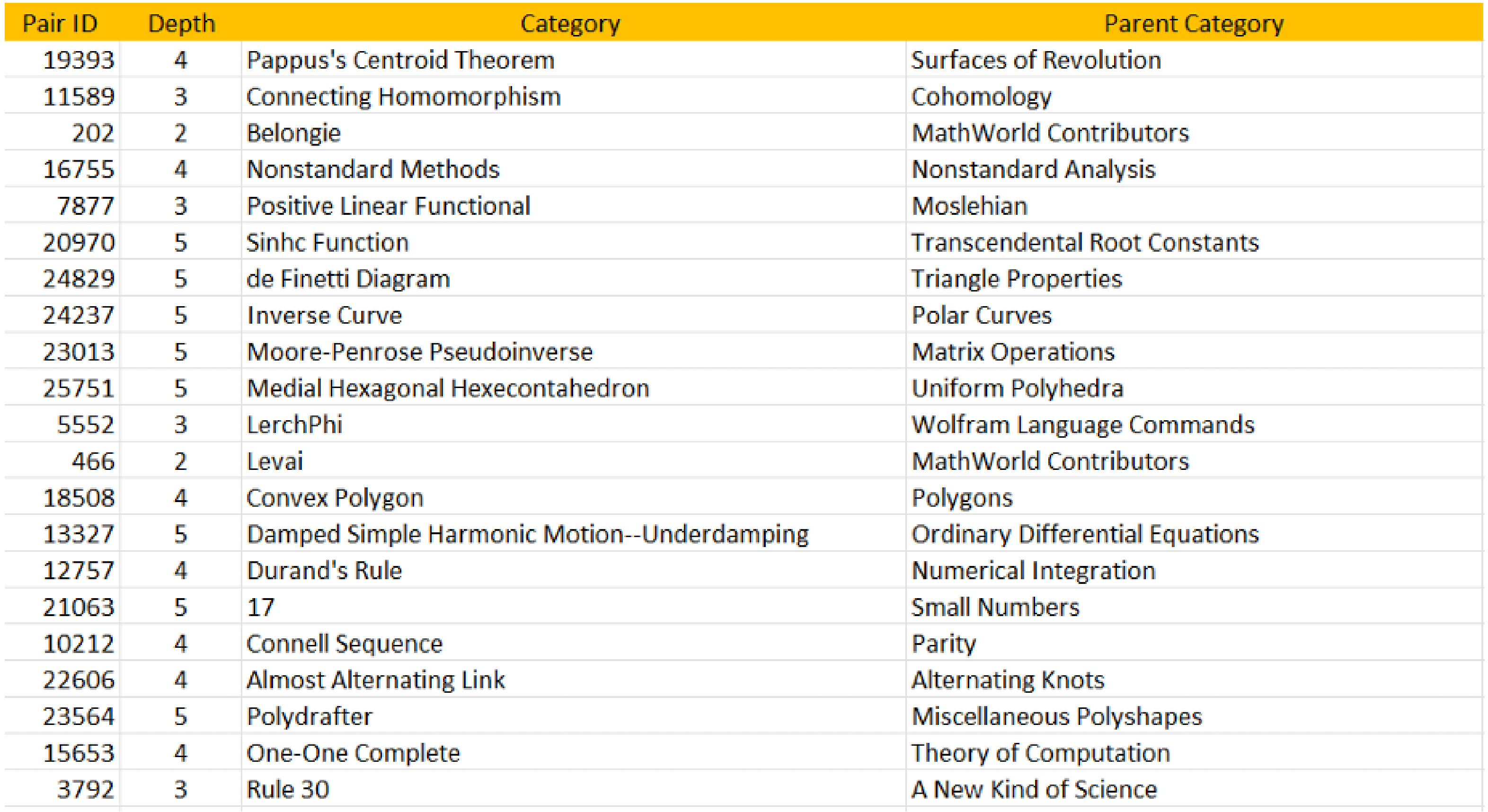
$$\int_0^1 \sqrt{x\,(1-x)}\,dx = \frac{\pi}{8} \approx 0.392699081698724154807830422909937860524646175$$

# Part 2
# xLLM Innovative Features

# Backend Features

- **Smart crawling** to retrieve embedded structure
  - Breadcrumbs (enterprise corpus), concept associations (related links)
  - Metadata, tags, taxonomy (category graph)
  - Augmented with user prompts
  - Augmented with PDFs (TOC, index, glossaries, synonyms, titles)
- **X-embeddings**
  - Variable-length embeddings stored as sparse nested hashes
  - Multi-token: “data~science” on top of single tokens “data” and “science”
  - Contextual token: “data^science”, both words in same paragraph but not adjacent
  - PMI (pointwise mutual information) instead of dot product / cosine distance
  - Parametric weights attached to tokens (no loss function to optimize)

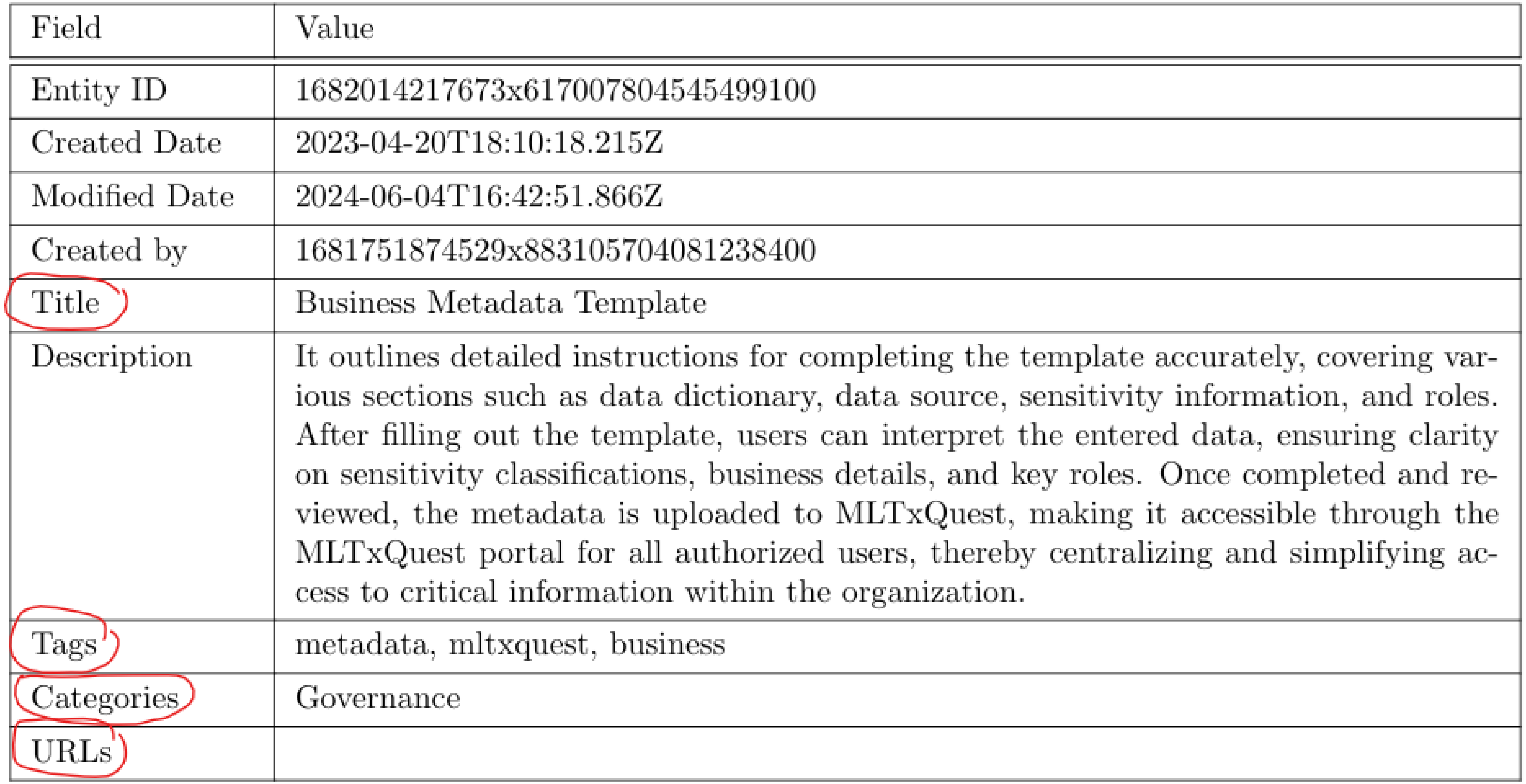
# Retrieved Taxonomy: Wolfram Example



| Pair ID | Depth | Category | Parent Category |
|---|---|---|---|
| 19393 | 4 | Pappus's Centroid Theorem | Surfaces of Revolution |
| 11589 | 3 | Connecting Homomorphism | Cohomology |
| 202 | 2 | Belongie | MathWorld Contributors |
| 16755 | 4 | Nonstandard Methods | Nonstandard Analysis |
| 7877 | 3 | Positive Linear Functional | Moslehian |
| 20970 | 5 | Sinhc Function | Transcendental Root Constants |
| 24829 | 5 | de Finetti Diagram | Triangle Properties |
| 24237 | 5 | Inverse Curve | Polar Curves |
| 23013 | 5 | Moore-Penrose Pseudoinverse | Matrix Operations |
| 25751 | 5 | Medial Hexagonal Hexecontahedron | Uniform Polyhedra |
| 5552 | 3 | LerchPhi | Wolfram Language Commands |
| 466 | 2 | Levai | MathWorld Contributors |
| 18508 | 4 | Convex Polygon | Polygons |
| 13327 | 5 | Damped Simple Harmonic Motion--Underdamping | Ordinary Differential Equations |
| 12757 | 4 | Durand's Rule | Numerical Integration |
| 21063 | 5 | 17 | Small Numbers |
| 10212 | 4 | Connell Sequence | Parity |
| 22606 | 4 | Almost Alternating Link | Alternating Knots |
| 23564 | 5 | Polydrafter | Miscellaneous Polyshapes |
| 15653 | 4 | One-One Complete | Theory of Computation |
| 3792 | 3 | Rule 30 | A New Kind of Science |

Figure 7.3: Extract from reconstructed taxonomy structure, Wolfram website

# Retrieved Context: Enterprise Example



| Field | Value |
|---|---|
| Entity ID | 1682014217673x617007804545499100 |
| Created Date | 2023-04-20T18:10:18.215Z |
| Modified Date | 2024-06-04T16:42:51.866Z |
| Created by | 1681751874529x883105704081238400 |
| Title | Business Metadata Template |
| Description | It outlines detailed instructions for completing the template accurately, covering various sections such as data dictionary, data source, sensitivity information, and roles. After filling out the template, users can interpret the entered data, ensuring clarity on sensitivity classifications, business details, and key roles. Once completed and reviewed, the metadata is uploaded to MLTxQuest, making it accessible through the MLTxQuest portal for all authorized users, thereby centralizing and simplifying access to critical information within the organization. |
| Tags | metadata, mltxquest, business |
| Categories | Governance |
| URLs | |

# Backend Features (Cont.)

- **Home-made libraries**
  - Issues with Python libraries (singularize, autocorrect, “Feller” changed to “seller”)
  - Minimize stemming and text transforms; keep plural if found in corpus
  - Important: accented characters, separators (punctuation), capital letters
  - Ad-hoc lists: home-made stopwords, do-not-singularize, do-not-autocorrect
- **Backend tables** (specific to each sub-LLM)
  - X-embeddings not the most important table; taxonomy more important
  - Compression mechanism: sorted *n*-grams
  - Backend parameters

# Backend Features (Cont.)

- **Chunking & Indexing**
  - Chunks called text entities: webpage, subsection (PDF), or JSON entity
  - Indexed for fast retrieval of full content, and for easy content linking
  - Chunks of variable length
- **NLP**
  - Python with workarounds + homemade
  - Weighted graph tokens: multi-tokens found in the context/taxonomy elements
  - Customized pointwise mutual information (PMI), instead of cosine similarity

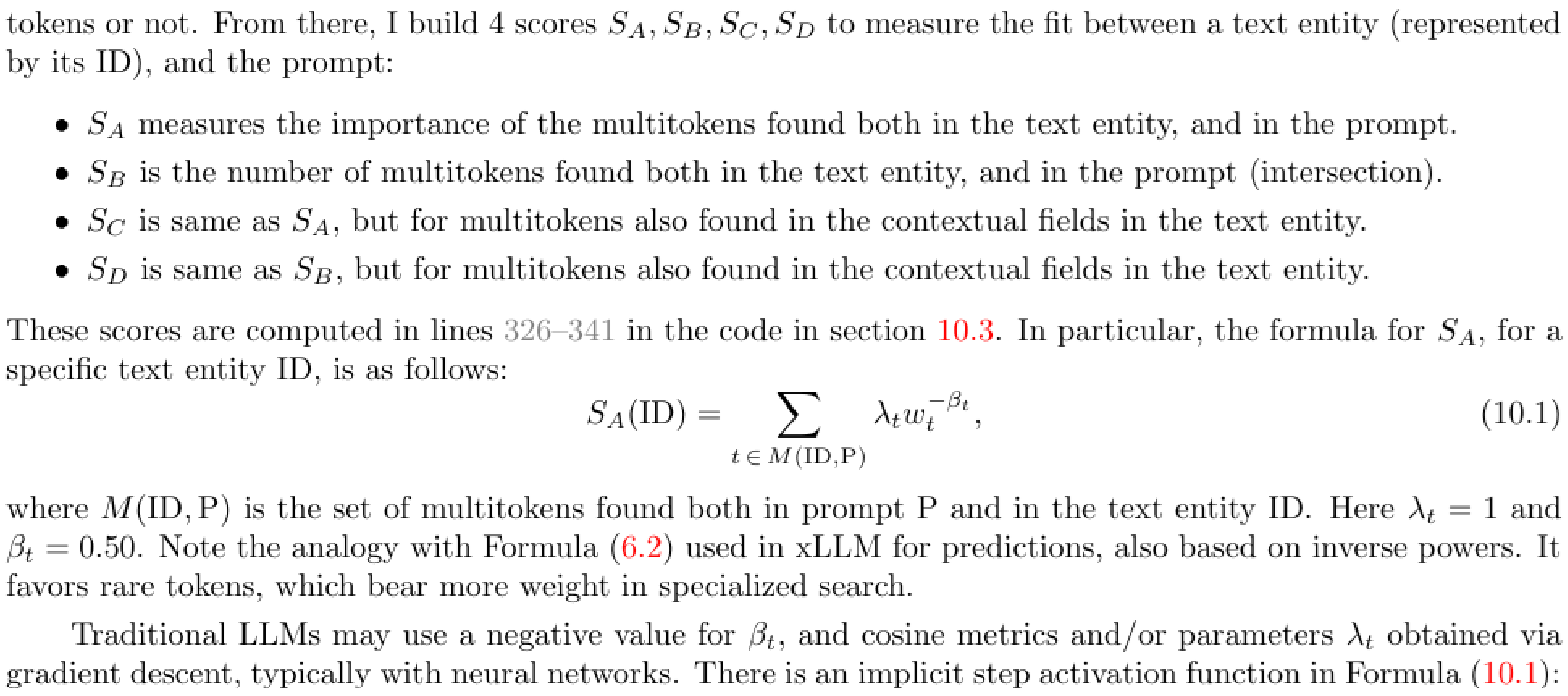
# Backend Features (Cont.)

- **Augmentation**
  - Easy integration of external sources, tested on corporate corpus
  - External content flagged via tags or other context elements
  - User told if piece of output is internal or external
  - Taxonomy augmentation
- **Agents**
  - Assigned post-crawling to text entities via clustering, for easy matching with prompt
  - Different from standard implementations (bottom up rather than top down)
- **Content Deduping**

# Frontend Features

- **User Interface**
  - Many options, not just a search box (see previous slide)
  - User can choose agents, sub-LLM, or fine-tuning in real time
  - End-user debugging with catch-all parameter set
- **Relevancy scores**
  - Goal: too many results to show to user prompt, which ones to display?
  - Graph tokens and multi-tokens with 2+ words boost score
  - Text entity with 2+ multi-token intersection with prompt, get higher score
  - Rare multi-tokens get extra boost
  - Longer text entities get extra boost

# Relevancy scores



tokens or not. From there, I build 4 scores $S_A, S_B, S_C, S_D$ to measure the fit between a text entity (represented by its ID), and the prompt:

- $S_A$ measures the importance of the multitokens found both in the text entity, and in the prompt.
- $S_B$ is the number of multitokens found both in the text entity, and in the prompt (intersection).
- $S_C$ is same as $S_A$, but for multitokens also found in the contextual fields in the text entity.
- $S_D$ is same as $S_B$, but for multitokens also found in the contextual fields in the text entity.

These scores are computed in lines 326–341 in the code in section 10.3. In particular, the formula for $S_A$, for a specific text entity ID, is as follows:

$$S_A(\text{ID}) = \sum_{t\,\in\, M(\text{ID},\text{P})} \lambda_t w_t^{-\beta_t}, \tag{10.1}$$

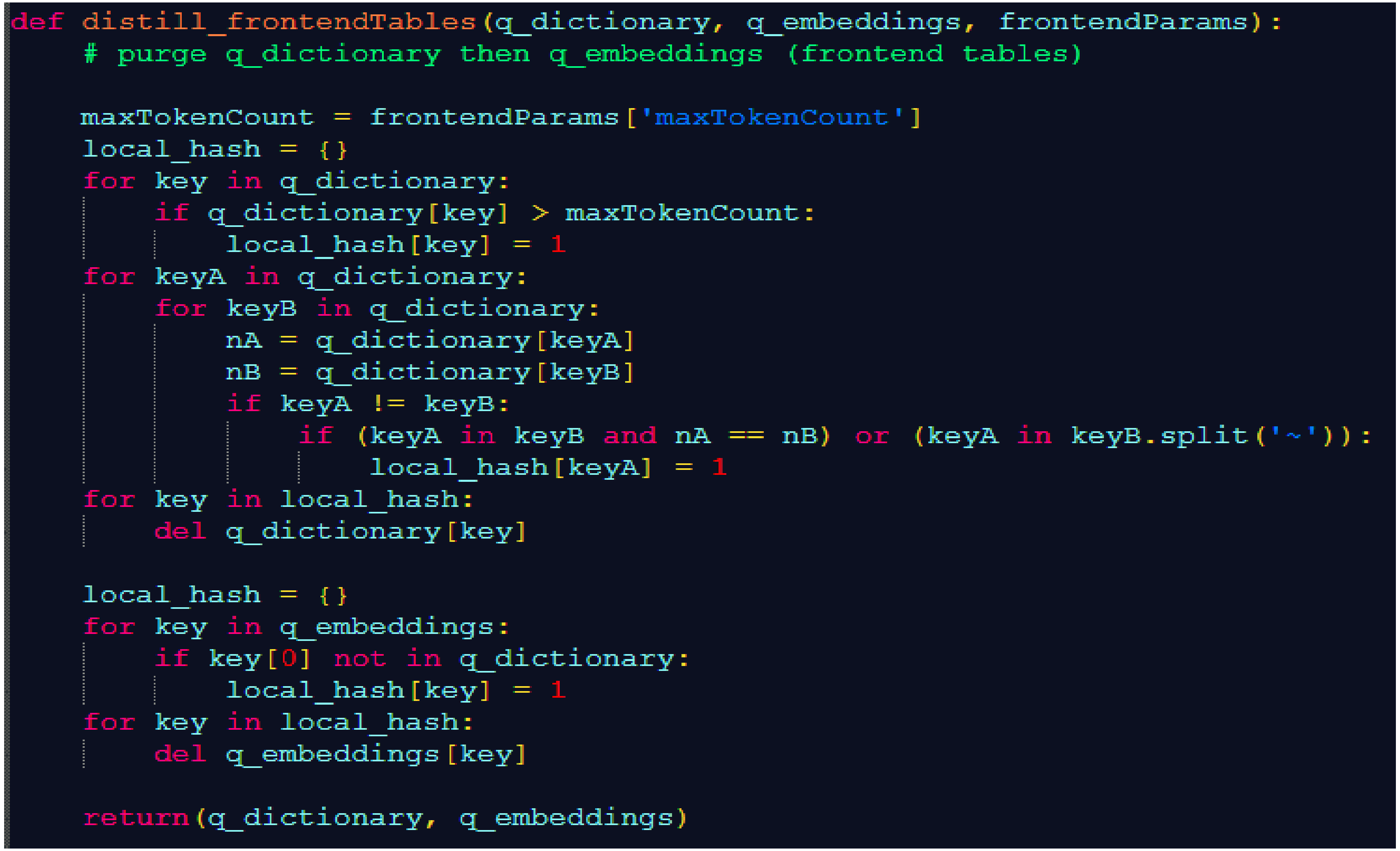
where $M(\text{ID}, \text{P})$ is the set of multitokens found both in prompt P and in the text entity ID. Here $\lambda_t = 1$ and $\beta_t = 0.50$. Note the analogy with Formula (6.2) used in xLLM for predictions, also based on inverse powers. It favors rare tokens, which bear more weight in specialized search.

Traditional LLMs may use a negative value for $\beta_t$, and cosine metrics and/or parameters $\lambda_t$ obtained via gradient descent, typically with neural networks. There is an implicit step activation function in Formula (10.1):

# Frontend Features (Cont.)

- **Distillation**
  - If multi-tokens A~B~C and A~B have same count, show results from A~B~C, not A~B
- **Acronyms and synonyms**
  - If A and B are synonyms, A in prompt but not in corpus, and B in corpus, map A to B in the prompt to retrieve B in the corpus (Goal: trying to be exhaustive)
- **Self-tuning** – Most popular front-end parameters used to build default parameters
- **Prompt cleanup** with stopwords list different from backend list
- **Disambiguation** (coming soon)

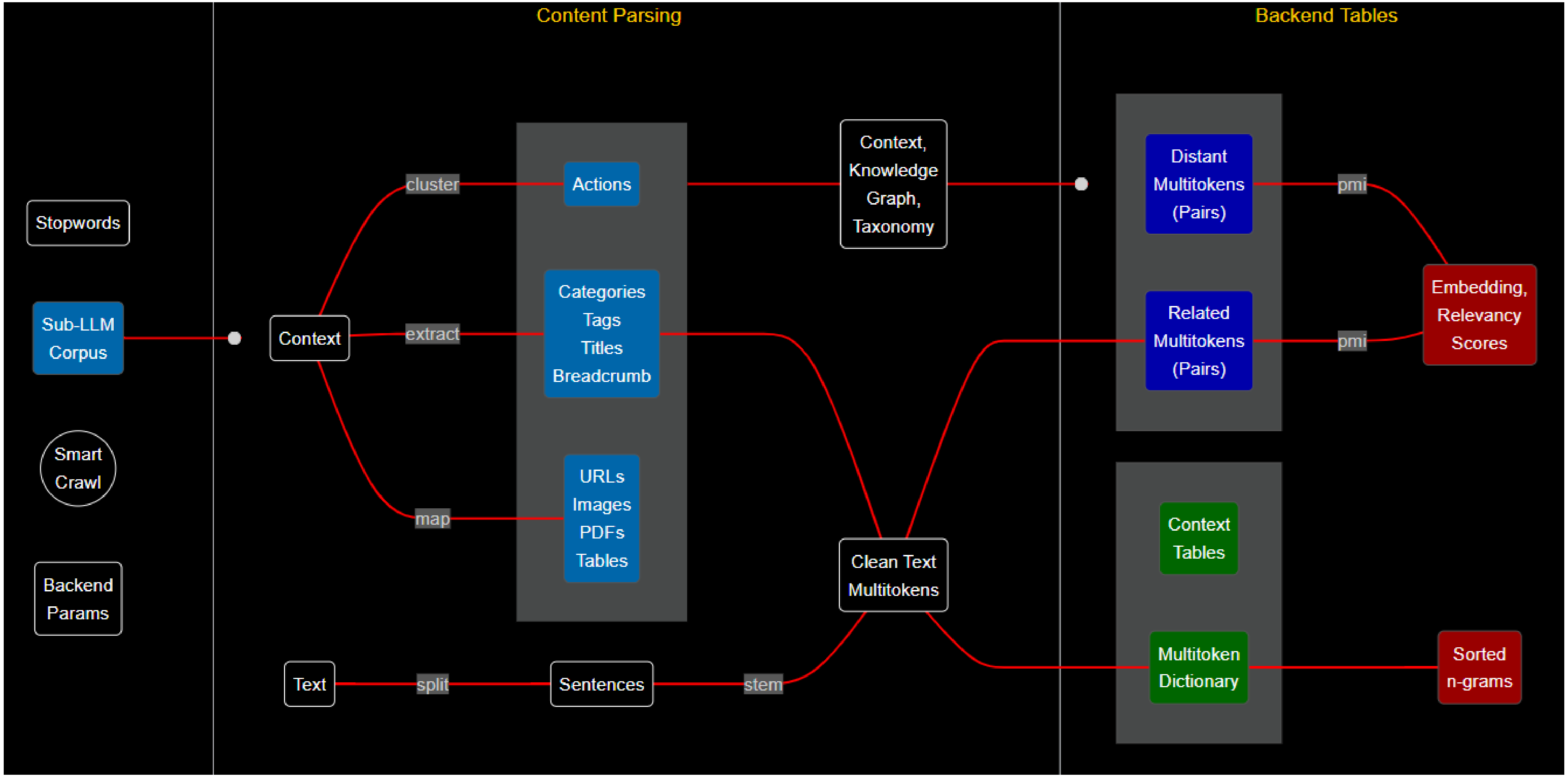
# Distillation



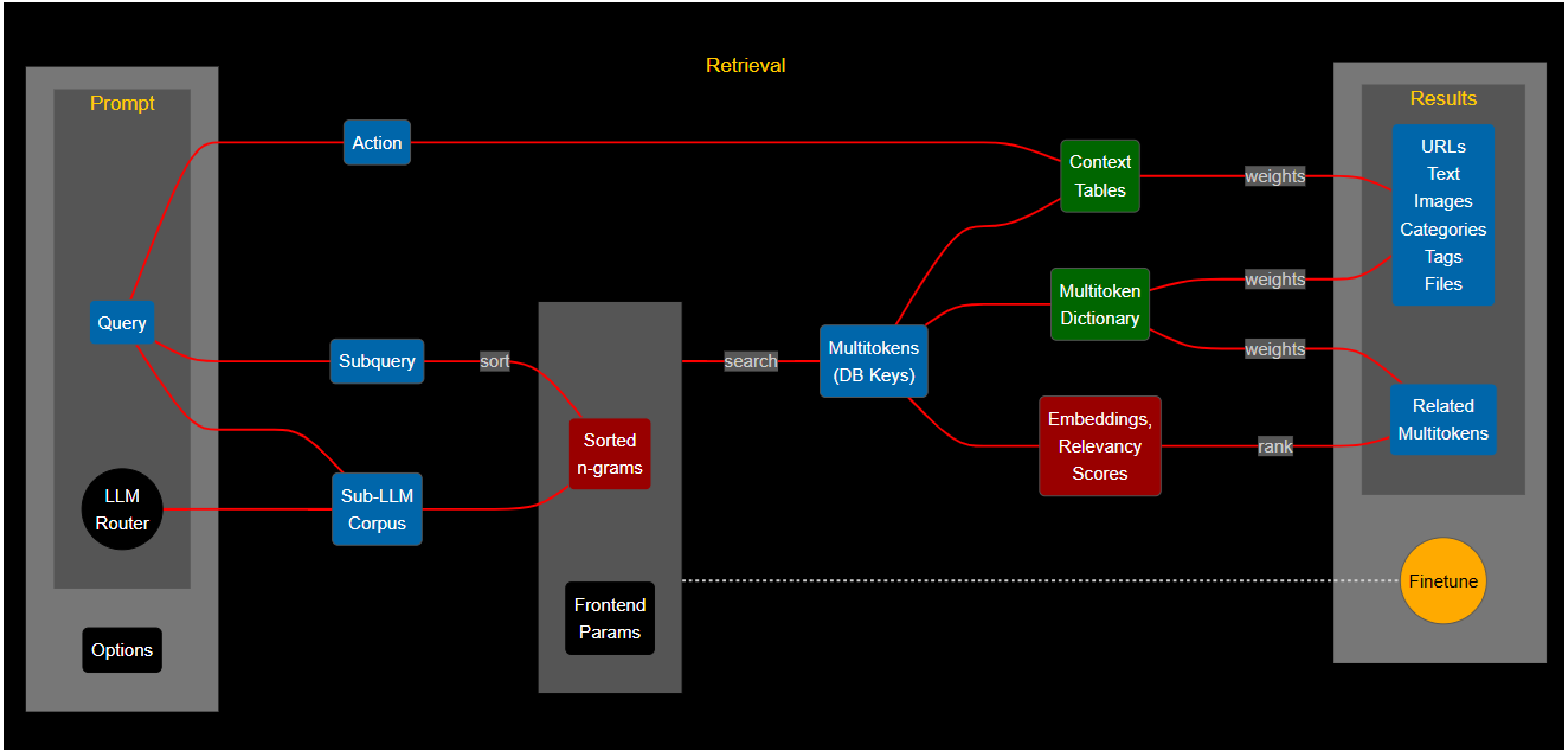
```python
def distill_frontendTables(q_dictionary, q_embeddings, frontendParams):
    # purge q_dictionary then q_embeddings (frontend tables)

    maxTokenCount = frontendParams['maxTokenCount']
    local_hash = {}
    for key in q_dictionary:
        if q_dictionary[key] > maxTokenCount:
            local_hash[key] = 1
    for keyA in q_dictionary:
        for keyB in q_dictionary:
            nA = q_dictionary[keyA]
            nB = q_dictionary[keyB]
            if keyA != keyB:
                if (keyA in keyB and nA == nB) or (keyA in keyB.split('~')):
                    local_hash[keyA] = 1
    for key in local_hash:
        del q_dictionary[key]

    local_hash = {}
    for key in q_embeddings:
        if key[0] not in q_dictionary:
            local_hash[key] = 1
    for key in local_hash:
        del q_embeddings[key]

    return(q_dictionary, q_embeddings)
```

# Part 3 xLLM Architecture and Evaluation

# Backend: Overview

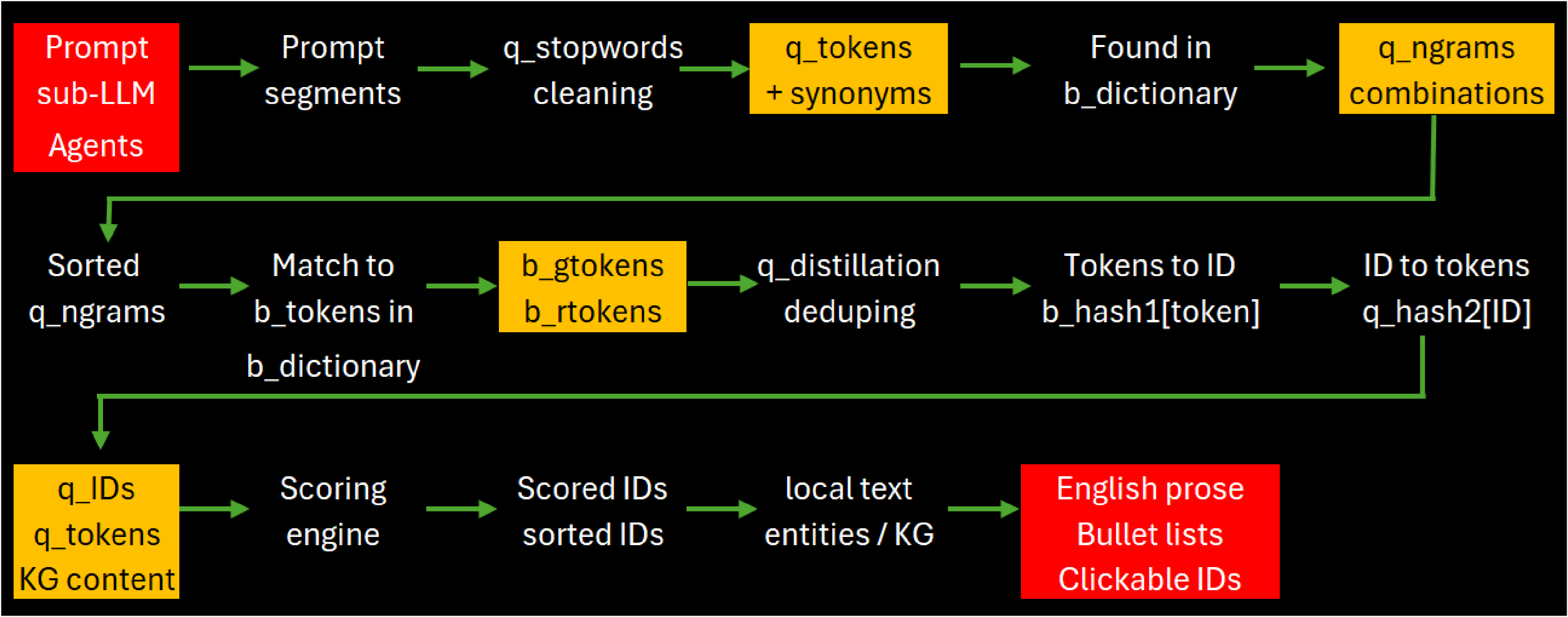


# Frontend: Overview



# Path from Prompt to Results

All tokens are multi-tokens; b_ and q_ prefix respectively for backend/global and frontend/local (q for query)
ID are attached to corpus text entities or chunks, via indexing mechanism; all tables are nested hashes
gtokens, rtokens respectively for graph and regular tokens (the former found in the KG sections of text entities)
KG stands for knowledge graph elements (tags, agents, titles, categories, URLs, related items)



Prompt sub-LLM Agents

Prompt segments

q_stopwords cleaning

q_tokens + synonyms

Found in b_dictionary

q_ngrams combinations

Sorted q_ngrams

Match to b_tokens in b_dictionary

b_gtokens b_rtokens

q_distillation deduping

Tokens to ID b_hash1[token]

ID to tokens q_hash2[ID]

q_IDs q_tokens KG content

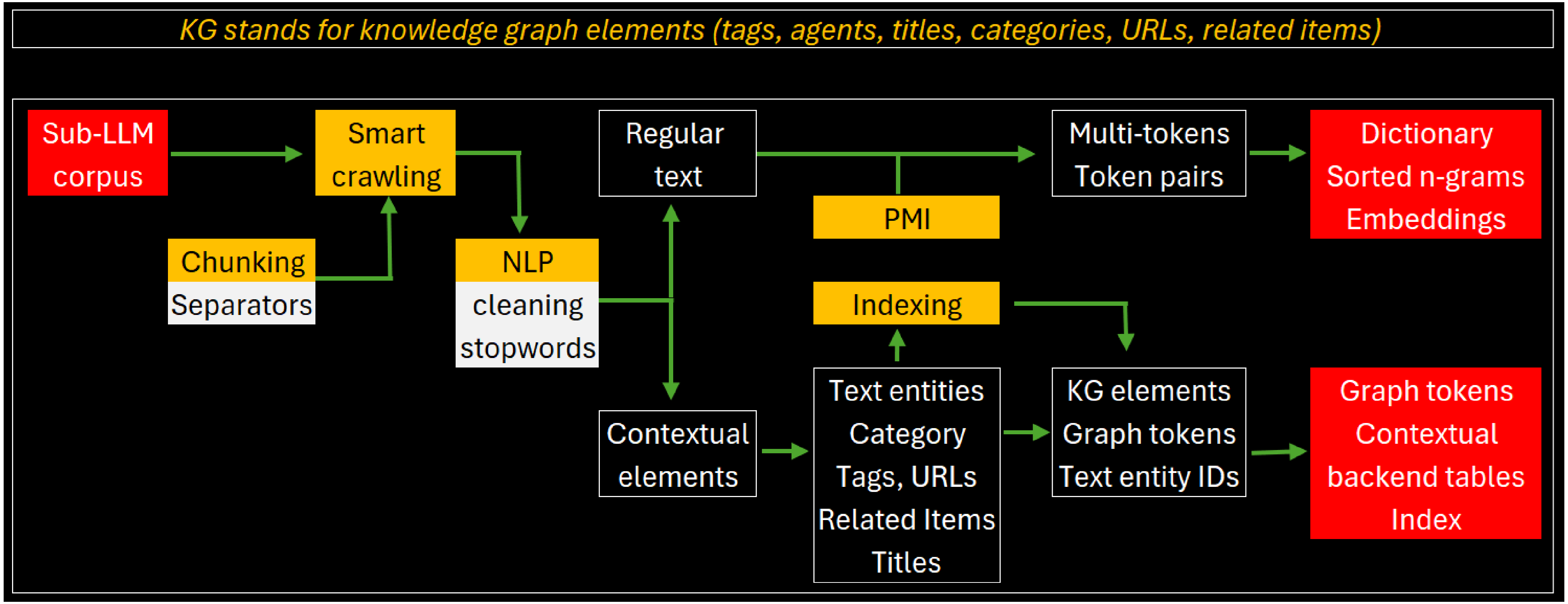
Scoring engine

Scored IDs sorted IDs

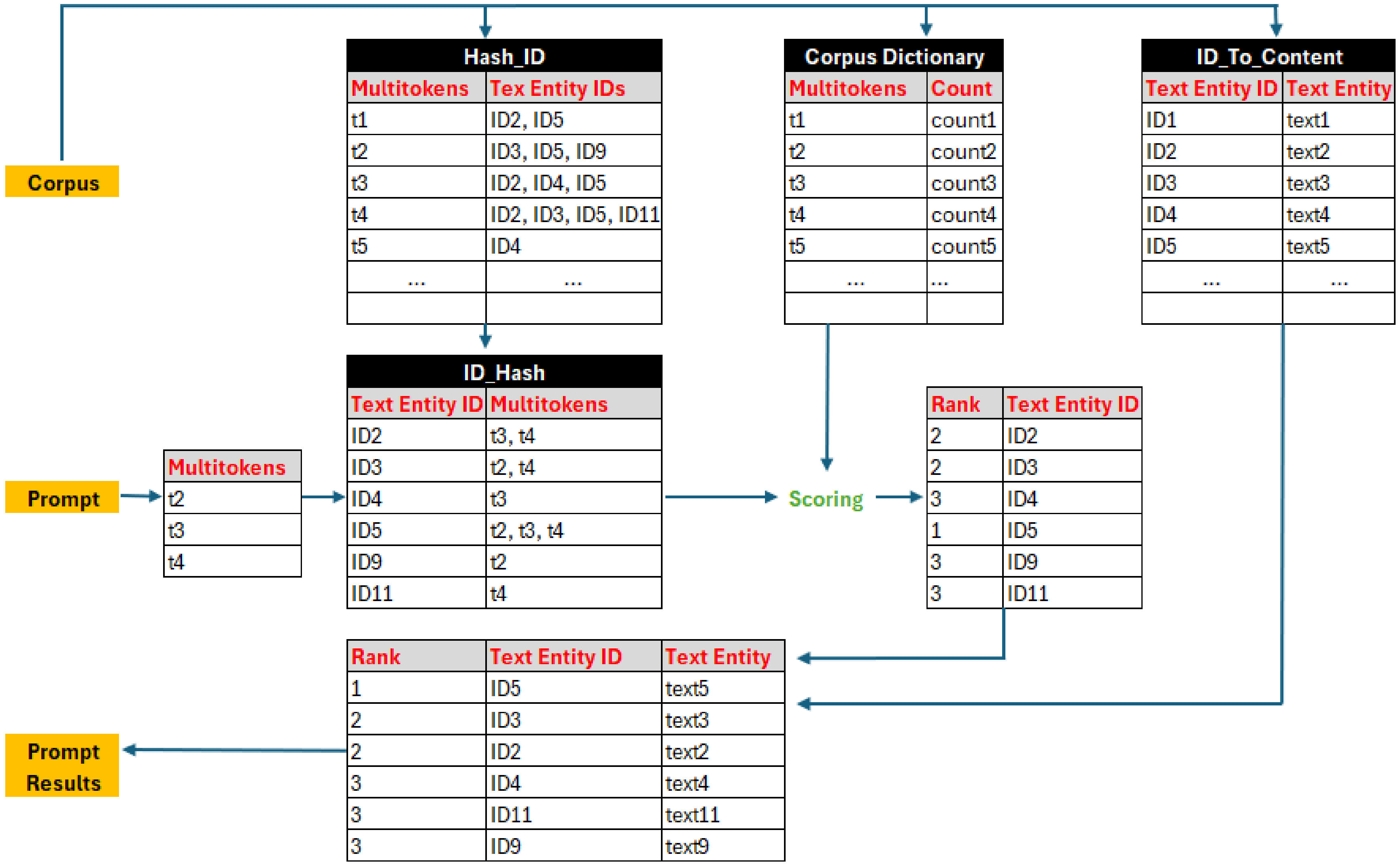
local text entities / KG

English prose Bullet lists Clickable IDs

# Path from Crawl to Backend Tables


KG stands for knowledge graph elements (tags, agents, titles, categories, URLs, related items)
Sub-LLM corpus
Smart crawling
Chunking
Separators
NLP
cleaning stopwords
Regular text
PMI
Multi-tokens Token pairs
Dictionary Sorted n-grams Embeddings
Indexing
Contextual elements
Text entities Category Tags, URLs Related Items Titles
KG elements Graph tokens Text entity IDs
Graph tokens Contextual backend tables Index

# Details: Indexation



Corpus

**Hash_ID**

| Multitokens | Tex Entity IDs |
|---|---|
| t1 | ID2, ID5 |
| t2 | ID3, ID5, ID9 |
| t3 | ID2, ID4, ID5 |
| t4 | ID2, ID3, ID5, ID11 |
| t5 | ID4 |
| ... | ... |
| | |

**Corpus Dictionary**

| Multitokens | Count |
|---|---|
| t1 | count1 |
| t2 | count2 |
| t3 | count3 |
| t4 | count4 |
| t5 | count5 |
| ... | ... |
| | |

**ID_To_Content**

| Text Entity ID | Text Entity |
|---|---|
| ID1 | text1 |
| ID2 | text2 |
| ID3 | text3 |
| ID4 | text4 |
| ID5 | text5 |
| ... | ... |
| | |

Prompt

| Multitokens |
|---|
| t2 |
| t3 |
| t4 |

**ID_Hash**

| Text Entity ID | Multitokens |
|---|---|
| ID2 | t3, t4 |
| ID3 | t2, t4 |
| ID4 | t3 |
| ID5 | t2, t3, t4 |
| ID9 | t2 |
| ID11 | t4 |

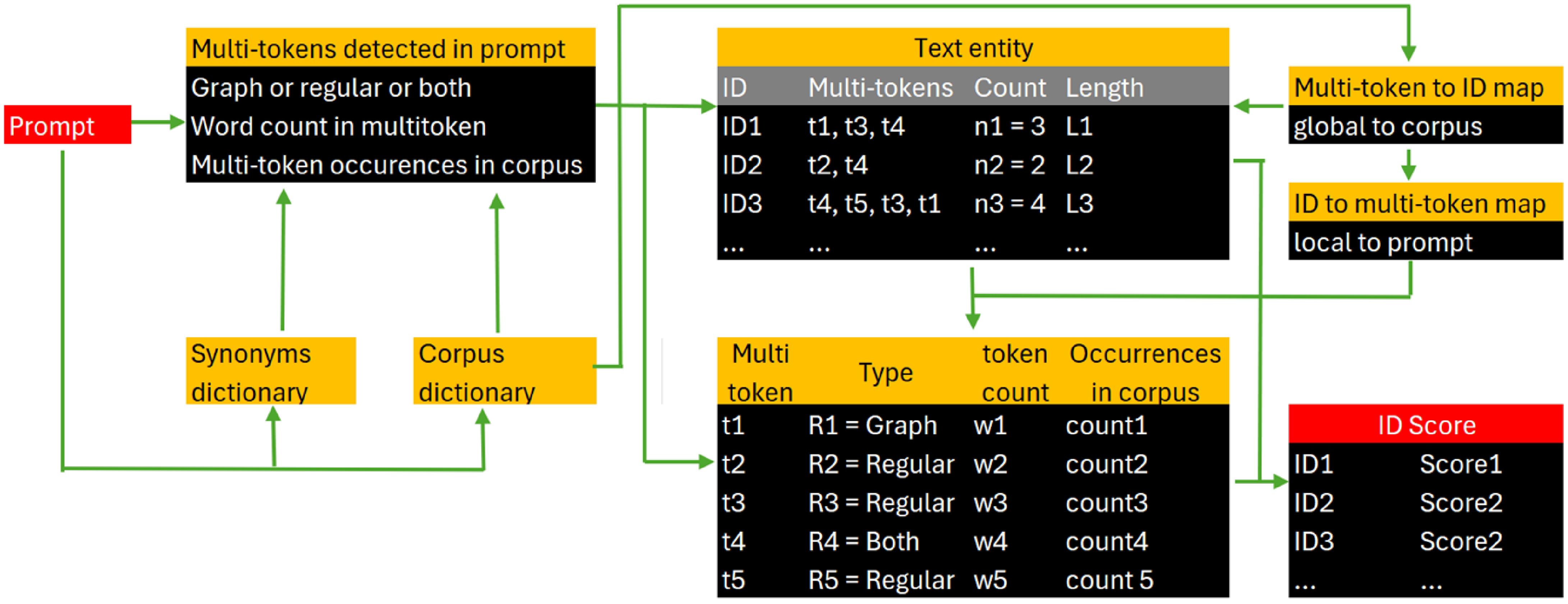
Scoring

| Rank | Text Entity ID |
|---|---|
| 2 | ID2 |
| 2 | ID3 |
| 3 | ID4 |
| 1 | ID5 |
| 3 | ID9 |
| 3 | ID11 |

Prompt Results

| Rank | Text Entity ID | Text Entity |
|---|---|---|
| 1 | ID5 | text5 |
| 2 | ID3 | text3 |
| 2 | ID2 | text2 |
| 3 | ID4 | text4 |
| 3 | ID11 | text11 |
| 3 | ID9 | text9 |

# Detail: Relevancy Algorithm



Prompt

Multi-tokens detected in prompt
- Graph or regular or both
- Word count in multitoken
- Multi-token occurences in corpus

Synonyms dictionary

Corpus dictionary

Text entity

| ID | Multi-tokens | Count | Length |
|---|---|---|---|
| ID1 | t1, t3, t4 | n1 = 3 | L1 |
| ID2 | t2, t4 | n2 = 2 | L2 |
| ID3 | t4, t5, t3, t1 | n3 = 4 | L3 |
| ... | ... | ... | ... |

Multi-token to ID map
global to corpus

ID to multi-token map
local to prompt

| Multi token | Type | token count | Occurrences in corpus |
|---|---|---|---|
| t1 | R1 = Graph | w1 | count1 |
| t2 | R2 = Regular | w2 | count2 |
| t3 | R3 = Regular | w3 | count3 |
| t4 | R4 = Both | w4 | count4 |
| t5 | R5 = Regular | w5 | count 5 |

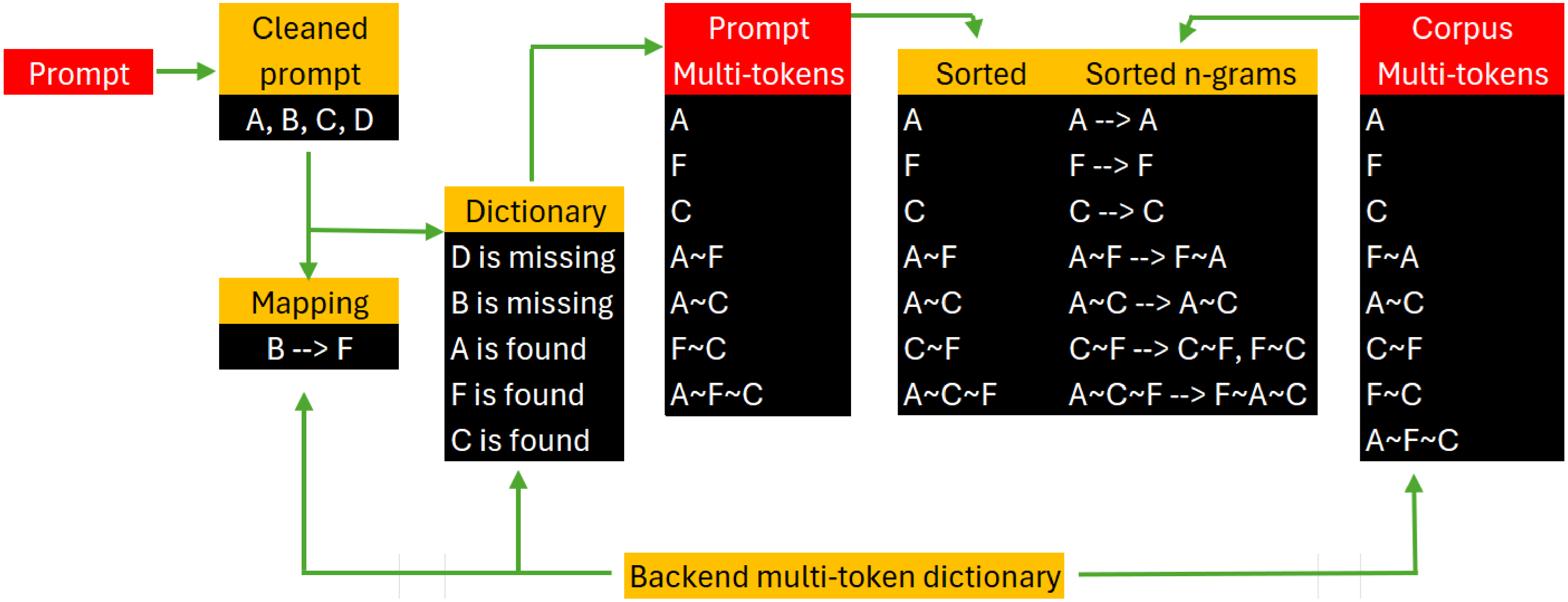
ID Score

| ID | Score |
|---|---|
| ID1 | Score1 |
| ID2 | Score2 |
| ID3 | Score2 |
| ... | ... |

Score[ID1] = F(L1, n1; count1, count3, count4; R1, R3, R4; w1, w3, w4)

Score[ID2] = F(L2, n2; count2, count4; R2, R4; w2, w4)

# Detail: Sorted *N*-Grams



Prompt

Cleaned prompt

A, B, C, D

Mapping

B --> F

Dictionary

D is missing
B is missing
A is found
F is found
C is found

Prompt Multi-tokens

A
F
C
A~F
A~C
F~C
A~F~C

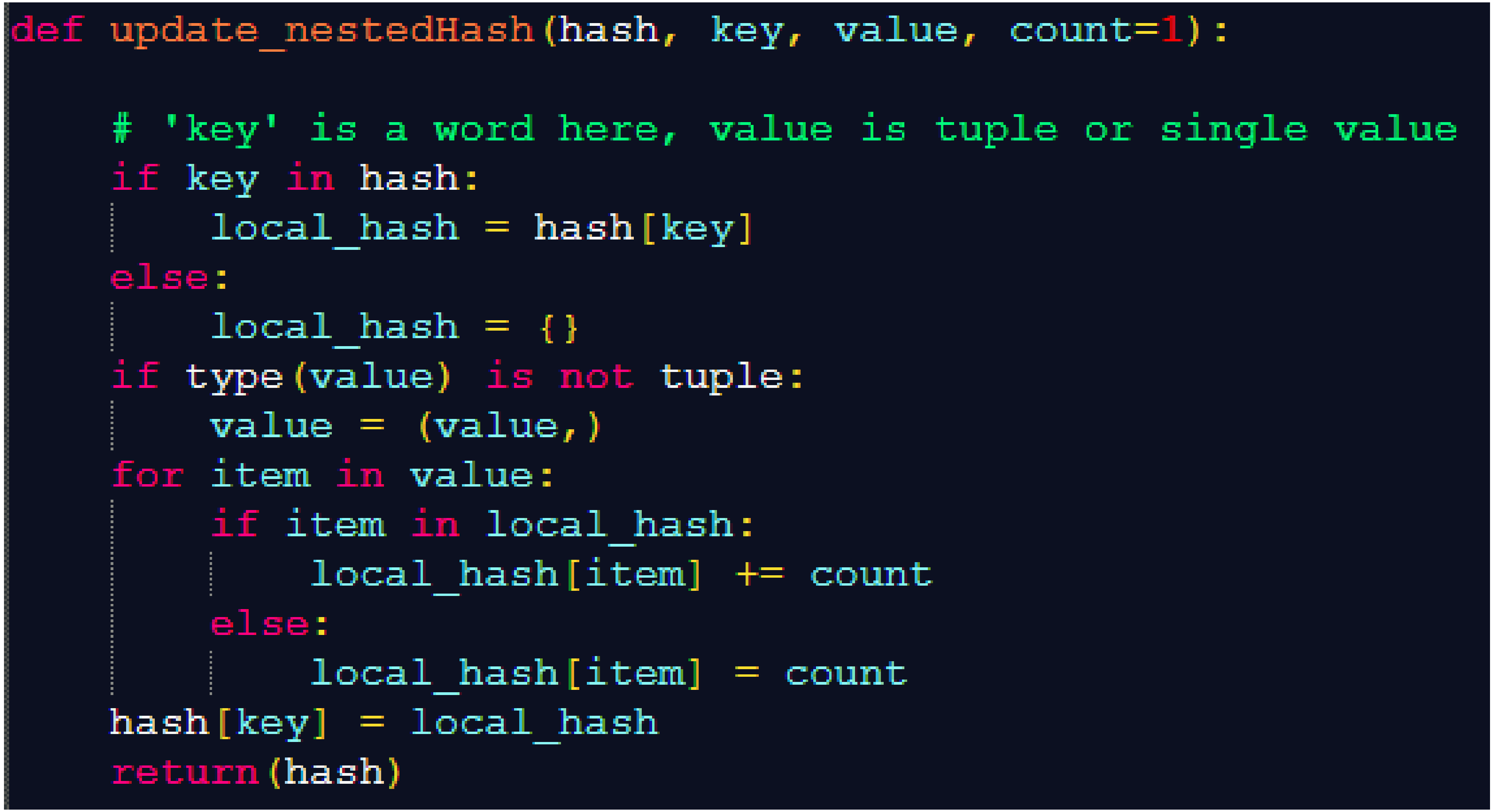
| Sorted | Sorted n-grams |
|---|---|
| A | A --> A |
| F | F --> F |
| C | C --> C |
| A~F | A~F --> F~A |
| A~C | A~C --> A~C |
| C~F | C~F --> C~F, F~C |
| A~C~F | A~C~F --> F~A~C |

Corpus Multi-tokens

A
F
C
F~A
A~C
C~F
F~C
A~F~C

Backend multi-token dictionary

# Database: Nested Hashes (like JSON)



```
def update_nestedHash(hash, key, value, count=1):

    # 'key' is a word here, value is tuple or single value
    if key in hash:
        local_hash = hash[key]
    else:
        local_hash = {}
    if type(value) is not tuple:
        value = (value,)
    for item in value:
        if item in local_hash:
            local_hash[item] += count
        else:
            local_hash[item] = count
    hash[key] = local_hash
    return(hash)
```

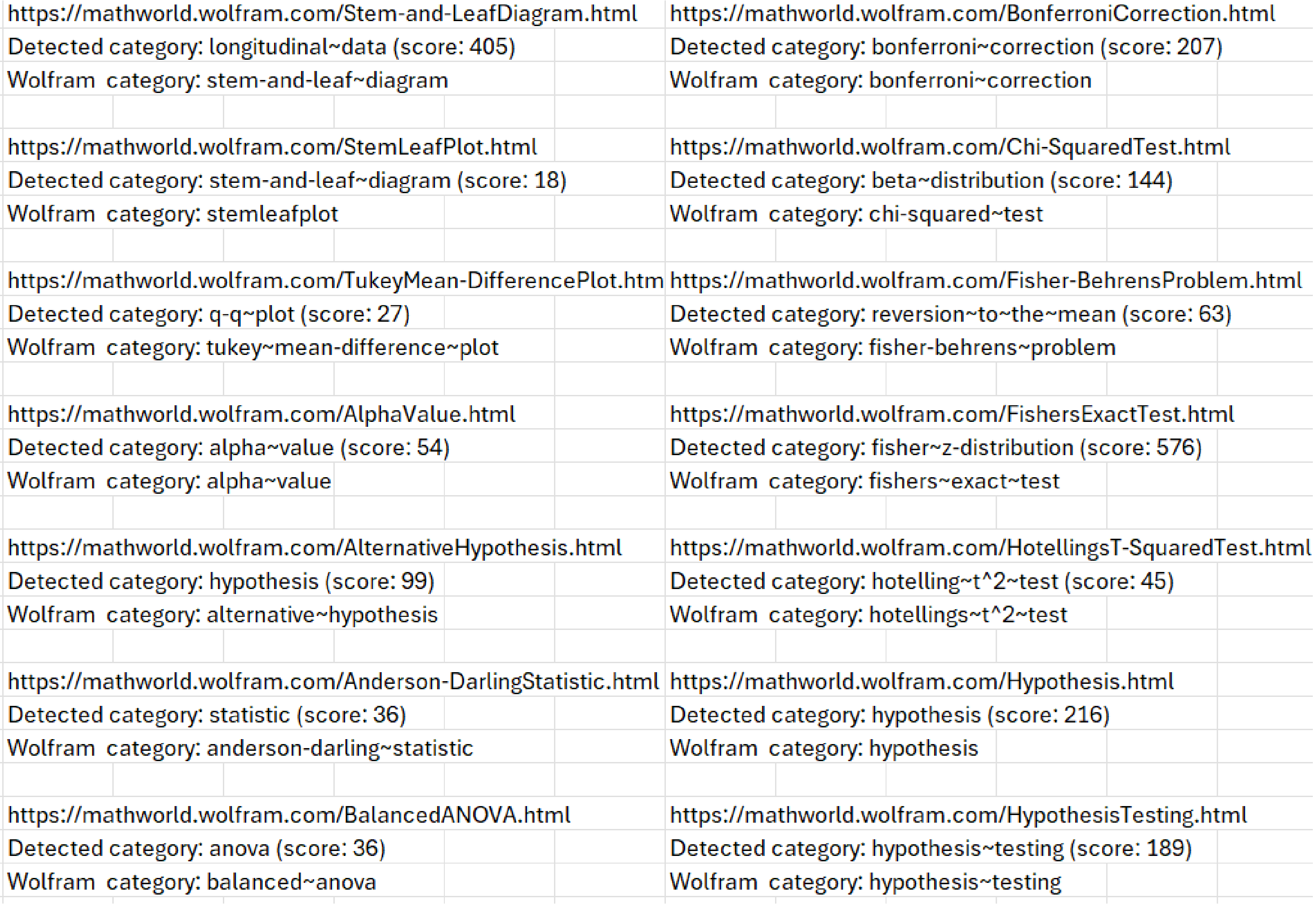
# Evaluation

- **User-based (automated)**
  - Collect favorite hyperparameters chosen by users
  - Use smart grid search to set default hyperparameters based on user favorites
  - Fine-tune on one or few sub-LLMs (like LoRA) before full optimization on (say) 200 sub-LLMs. You may fine-tune all sub-LLMs in parallel.
- **Taxonomy-based (automated)**
  - Pretend that the taxonomy backend table comes from external sources
  - Assign categories to webpages based on this “external” taxonomy
  - For each webpage, compare externally assigned to native category

# Evaluation (Cont.)

- **Evaluation challenges**
  - We are dealing with unsupervised learning: there is no perfect output except for trivial cases
  - Quality depends on user (professional users and laymen have different criteria)
  - How do you measure exhaustivity, depth, and recency?
  - Output value versus grammatical capabilities
  - How do you integrate xLLM relevancy scores attached to each item, to evaluate output quality? No other LLM return these scores

# Taxonomy-Based Evaluation

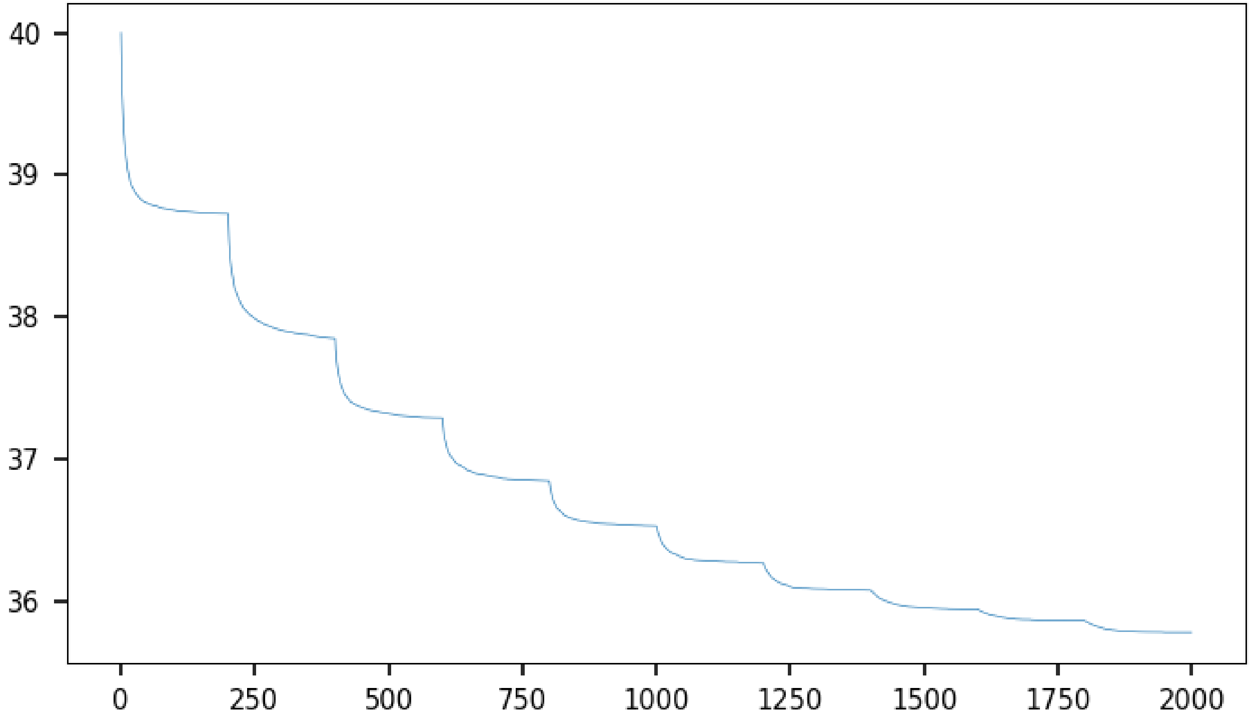


| | |
|---|---|
| https://mathworld.wolfram.com/Stem-and-LeafDiagram.html | https://mathworld.wolfram.com/BonferroniCorrection.html |
| Detected category: longitudinal~data (score: 405) | Detected category: bonferroni~correction (score: 207) |
| Wolfram category: stem-and-leaf~diagram | Wolfram category: bonferroni~correction |
| | |
| https://mathworld.wolfram.com/StemLeafPlot.html | https://mathworld.wolfram.com/Chi-SquaredTest.html |
| Detected category: stem-and-leaf~diagram (score: 18) | Detected category: beta~distribution (score: 144) |
| Wolfram category: stemleafplot | Wolfram category: chi-squared~test |
| | |
| https://mathworld.wolfram.com/TukeyMean-DifferencePlot.htm | https://mathworld.wolfram.com/Fisher-BehrensProblem.html |
| Detected category: q-q~plot (score: 27) | Detected category: reversion~to~the~mean (score: 63) |
| Wolfram category: tukey~mean-difference~plot | Wolfram category: fisher-behrens~problem |
| | |
| https://mathworld.wolfram.com/AlphaValue.html | https://mathworld.wolfram.com/FishersExactTest.html |
| Detected category: alpha~value (score: 54) | Detected category: fisher~z-distribution (score: 576) |
| Wolfram category: alpha~value | Wolfram category: fishers~exact~test |
| | |
| https://mathworld.wolfram.com/AlternativeHypothesis.html | https://mathworld.wolfram.com/HotellingsT-SquaredTest.html |
| Detected category: hypothesis (score: 99) | Detected category: hotelling~t^2~test (score: 45) |
| Wolfram category: alternative~hypothesis | Wolfram category: hotellings~t^2~test |
| | |
| https://mathworld.wolfram.com/Anderson-DarlingStatistic.html | https://mathworld.wolfram.com/Hypothesis.html |
| Detected category: statistic (score: 36) | Detected category: hypothesis (score: 216) |
| Wolfram category: anderson-darling~statistic | Wolfram category: hypothesis |
| | |
| https://mathworld.wolfram.com/BalancedANOVA.html | https://mathworld.wolfram.com/HypothesisTesting.html |
| Detected category: anova (score: 36) | Detected category: hypothesis~testing (score: 189) |
| Wolfram category: balanced~anova | Wolfram category: hypothesis~testing |

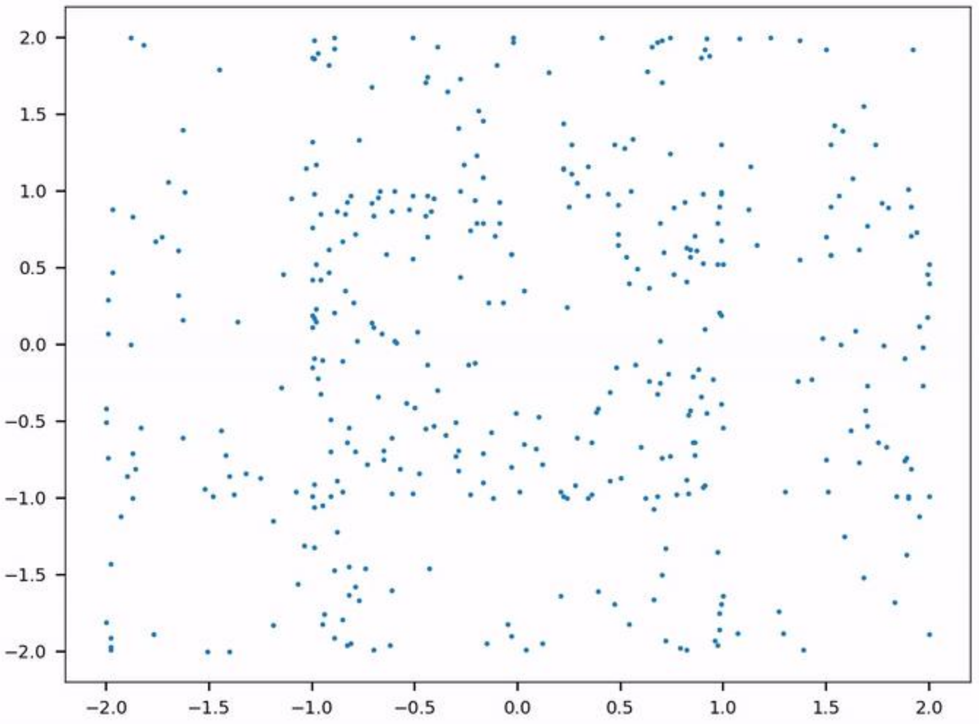
# Part 4
# xLLM for Clustering, Data Synthetization, Predictive Analytics

# Interlude – Adaptive Loss Function (ALF)

- **Adaptive loss function converging to model evaluation metric**
  - Boosts quality measured using model evaluation, reduces gradient descent failures



# xLLM for Data Synthetization (with ALF)

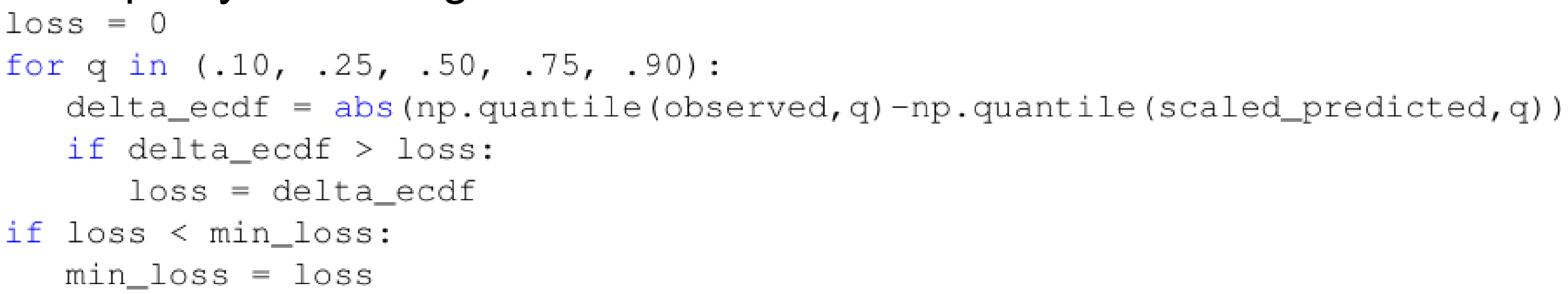


## NoGAN Tabular Data Synthetization

- Real data: 2 concentric circles
- Synthesized, NoGAN synthesizer: blue dots. Constrained synthetization to keep loss above some threshold
- As the loss function gets more granular, the synthesized data gets more similar to the real data (the training set)

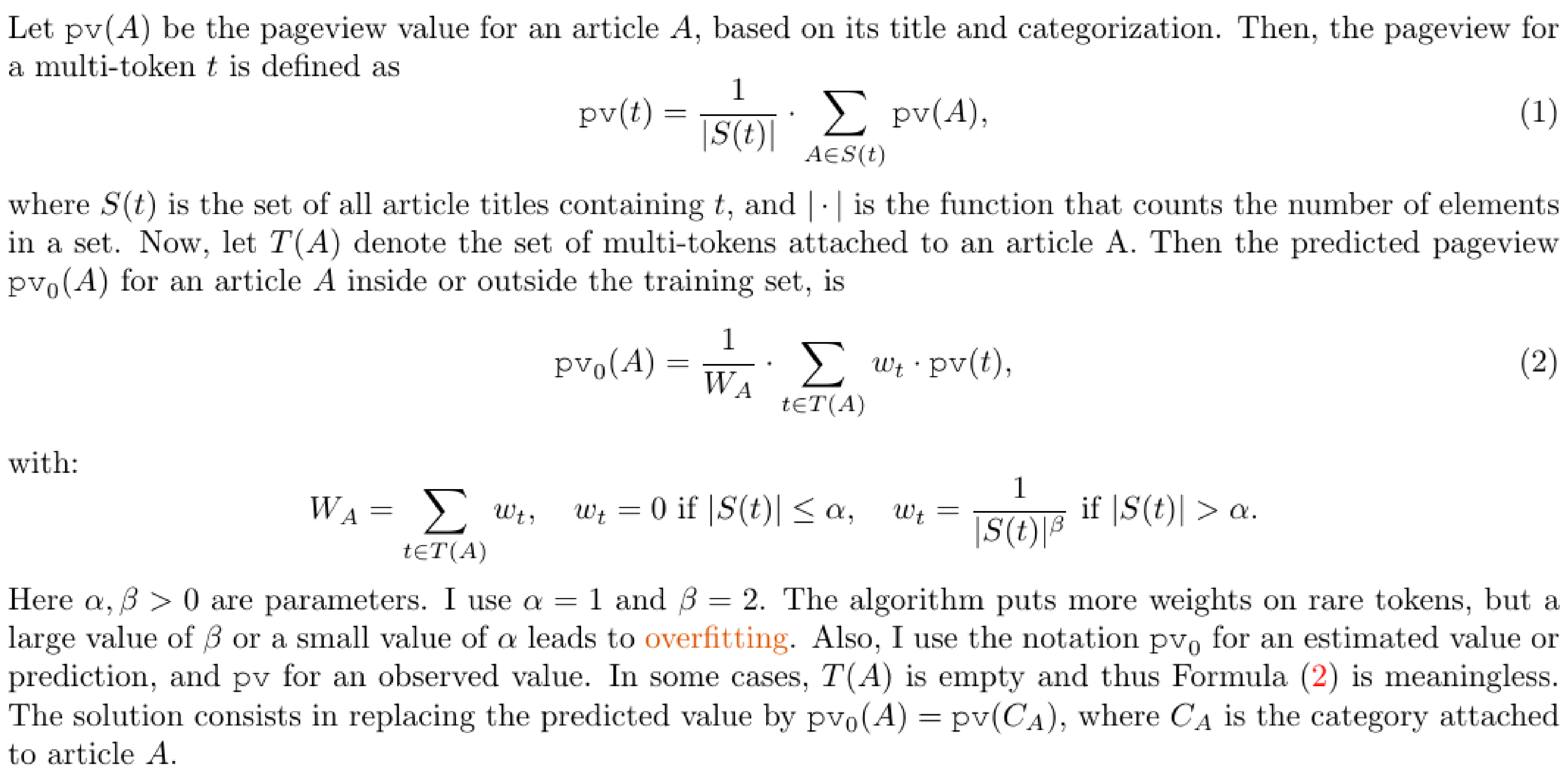
# xLLM for Predictions

- **Case study – media industry**
  - Predicting article performance (pageviews) based on title keywords and category
  - 4000 articles; pageview is normalized and time-adjusted
- **Evaluation and Loss function** (identical)
  - Based on comparing predicted with observed quantiles, using 5 quantiles (see code)
  - Good proxy to Kolmogorov-Smirnov distance



```
loss = 0
for q in (.10, .25, .50, .75, .90):
   delta_ecdf = abs(np.quantile(observed,q)-np.quantile(scaled_predicted,q))
   if delta_ecdf > loss:
      loss = delta_ecdf
if loss < min_loss:
   min_loss = loss
```

# xLLM for Predictions – Model



Let $\text{pv}(A)$ be the pageview value for an article $A$, based on its title and categorization. Then, the pageview for a multi-token $t$ is defined as

$$\text{pv}(t) = \frac{1}{|S(t)|} \cdot \sum_{A \in S(t)} \text{pv}(A), \tag{1}$$

where $S(t)$ is the set of all article titles containing $t$, and $|\cdot|$ is the function that counts the number of elements in a set. Now, let $T(A)$ denote the set of multi-tokens attached to an article A. Then the predicted pageview $\text{pv}_0(A)$ for an article $A$ inside or outside the training set, is

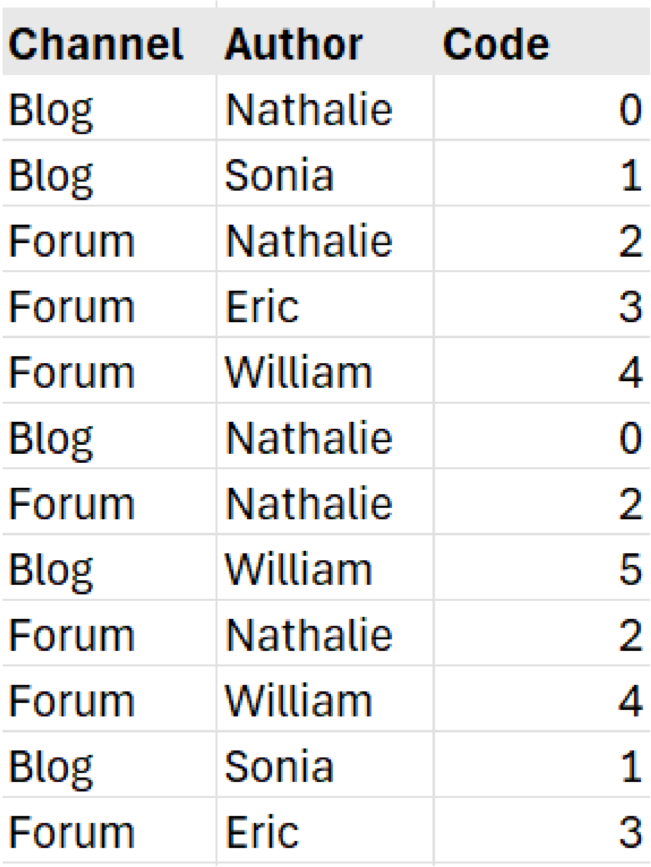
$$\text{pv}_0(A) = \frac{1}{W_A} \cdot \sum_{t \in T(A)} w_t \cdot \text{pv}(t), \tag{2}$$

with:

$$W_A = \sum_{t \in T(A)} w_t, \quad w_t = 0 \text{ if } |S(t)| \leq \alpha, \quad w_t = \frac{1}{|S(t)|^\beta} \text{ if } |S(t)| > \alpha.$$

Here $\alpha, \beta > 0$ are parameters. I use $\alpha = 1$ and $\beta = 2$. The algorithm puts more weights on rare tokens, but a large value of $\beta$ or a small value of $\alpha$ leads to overfitting. Also, I use the notation $\text{pv}_0$ for an estimated value or prediction, and $\text{pv}$ for an observed value. In some cases, $T(A)$ is empty and thus Formula (2) is meaningless. The solution consists in replacing the predicted value by $\text{pv}_0(A) = \text{pv}(C_A)$, where $C_A$ is the category attached to article $A$.

# xLLM for Predictions – Category Encoding

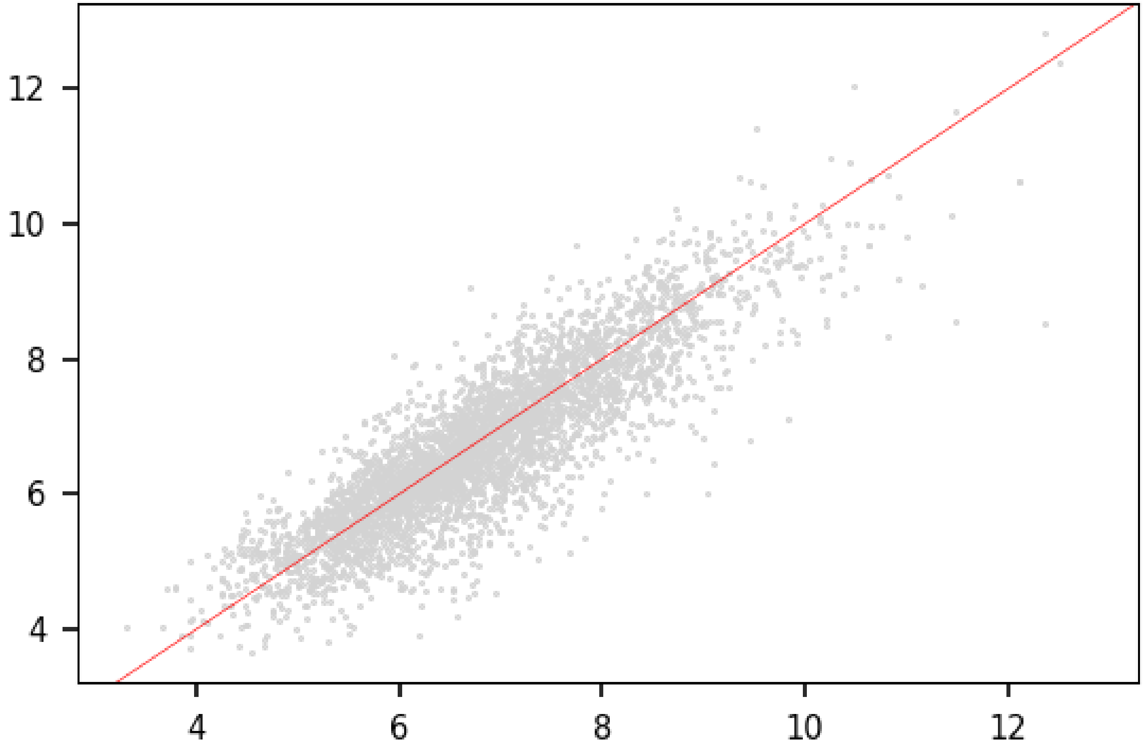


| Channel | Author | Code |
|---|---|---|
| Blog | Nathalie | 0 |
| Blog | Sonia | 1 |
| Forum | Nathalie | 2 |
| Forum | Eric | 3 |
| Forum | William | 4 |
| Blog | Nathalie | 0 |
| Forum | Nathalie | 2 |
| Blog | William | 5 |
| Forum | Nathalie | 2 |
| Forum | William | 4 |
| Blog | Sonia | 1 |
| Forum | Eric | 3 |

- Create new codes sequentially as you browse the training set.
- Aggregate codes with few observations into bundles.
- Create two **key-value mappings**. Ex:
  - Category_to_Code[‘Blog’, ‘William’] = 5
  - Code_to_category[5] = [‘Blog’, ‘William’]
- Replace the categorical features by the newly created feature, “Code”.
- Number of codes ≤ number of obs.

# xLLM for Predictions – Results

- **Observed vs predicted normalized pageview count**



# xLLM for Clustering

- **Case study – media industry**
  - Identifying patterns / clusters in popular articles based on title keywords
  - 4000 articles; pageview is normalized and time-adjusted
- **Methodology**
  - Group multi-tokens into clusters based on a similarity metric, with hierarchical clustering and *k*-medoids
  - Let *S*(*t*) be the set of articles containing the multi-token *t* in the title
  - For each multi-token group *G*, the list *L*(*G*) of articles belonging to *G* is

$$L(G) = \bigcup_{t \in G} S(t).$$

# xLLM for Clustering (Cont.)

- **Similarity between two multi-tokens *$t_1$*, *$t_2$***

$$s(t_1, t_2) = \frac{|S(t_1) \cap S(t_2)|}{|S(t_1) \cup S(t_2)|} \in [0, 1].$$

- **Remarks**
  - Multi-token clusters are non-overlapping, but article clusters may overlap
  - Sklearn clustering methods require a distance matrix as input; the matrix (derived from the similarity metric) is huge but extremely sparse.
  - In my implementation, $s(t_1, t_2)$ is computed and stored only if it is strictly positive. Using connected components for clustering, it is far more efficient than Sklearn.

# xLLM for Clustering – Sample Structure

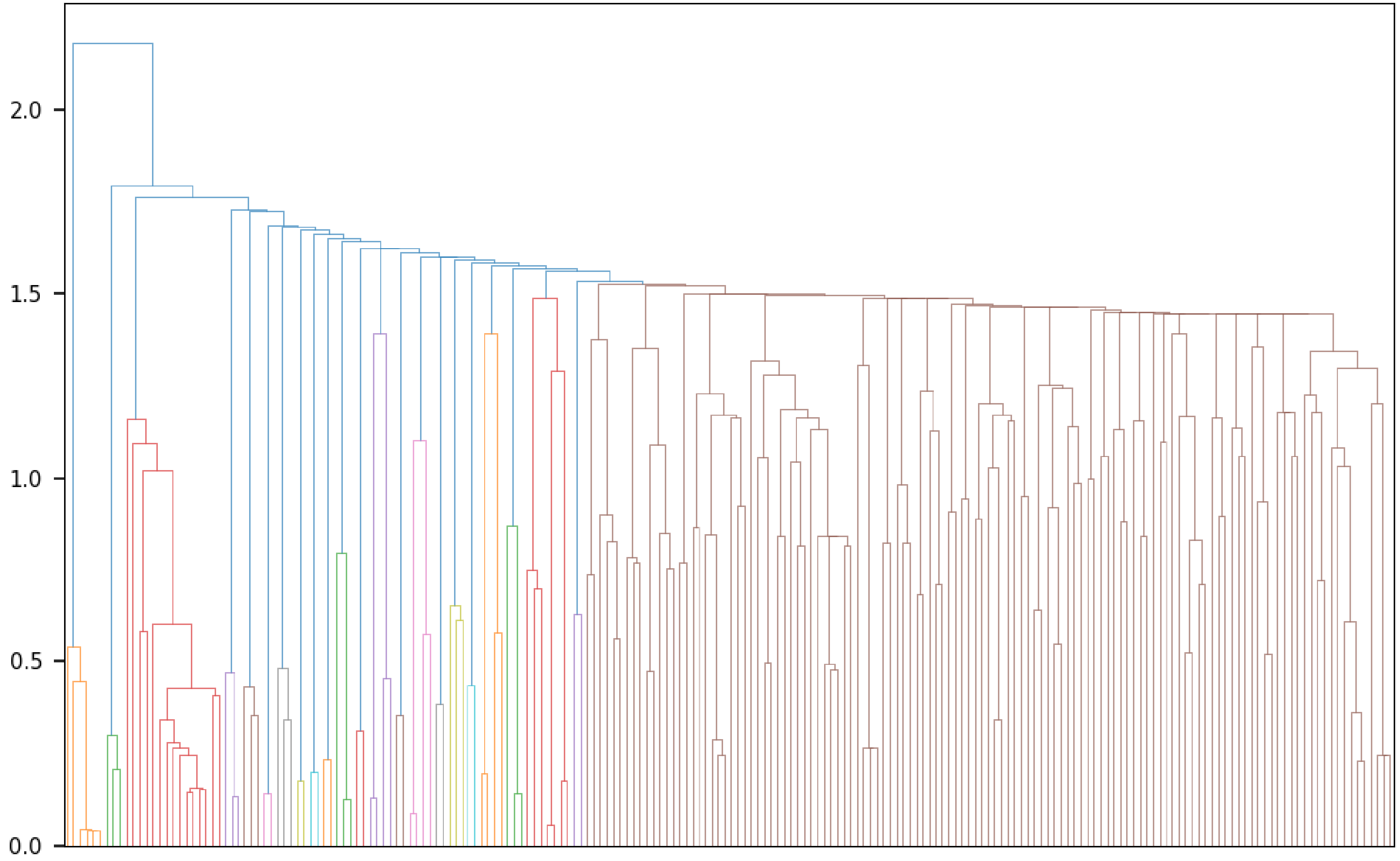
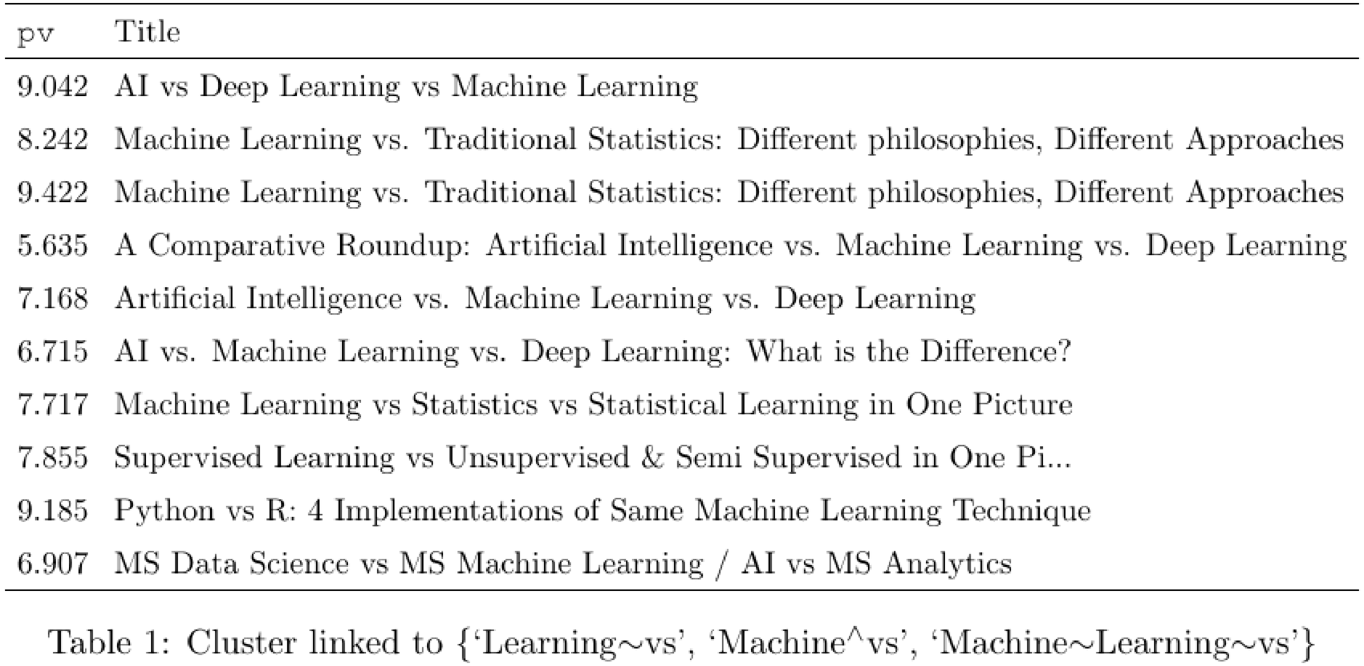


Figure 1: Multi-tokens hierarchical clustering: dendrogram (20 groups, 104 multi-tokens)

# xLLM for Clustering – Sample Cluster

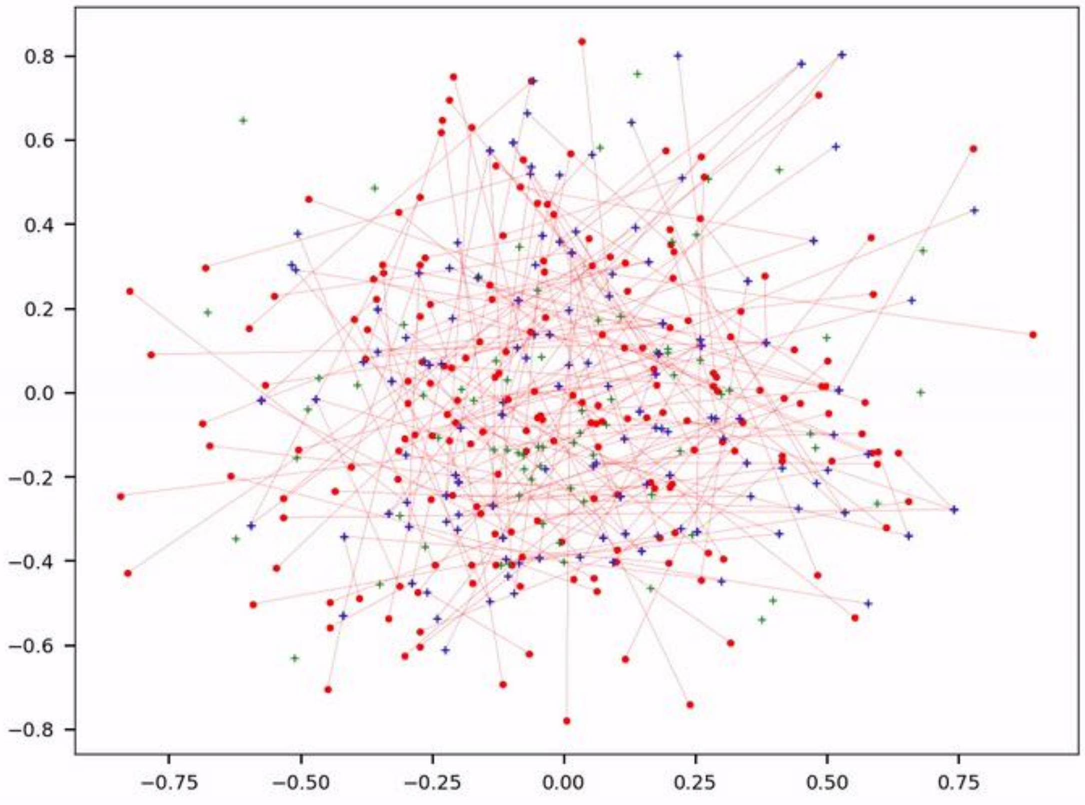
- **Cluster of popular articles** linked to multi-token cluster with 3 elements, including one contextual multi-token: “Machine^vs” (pv stands for normalized pageview)



| pv | Title |
|---|---|
| 9.042 | AI vs Deep Learning vs Machine Learning |
| 8.242 | Machine Learning vs. Traditional Statistics: Different philosophies, Different Approaches |
| 9.422 | Machine Learning vs. Traditional Statistics: Different philosophies, Different Approaches |
| 5.635 | A Comparative Roundup: Artificial Intelligence vs. Machine Learning vs. Deep Learning |
| 7.168 | Artificial Intelligence vs. Machine Learning vs. Deep Learning |
| 6.715 | AI vs. Machine Learning vs. Deep Learning: What is the Difference? |
| 7.717 | Machine Learning vs Statistics vs Statistical Learning in One Picture |
| 7.855 | Supervised Learning vs Unsupervised & Semi Supervised in One Pi... |
| 9.185 | Python vs R: 4 Implementations of Same Machine Learning Technique |
| 6.907 | MS Data Science vs MS Machine Learning / AI vs MS Analytics |

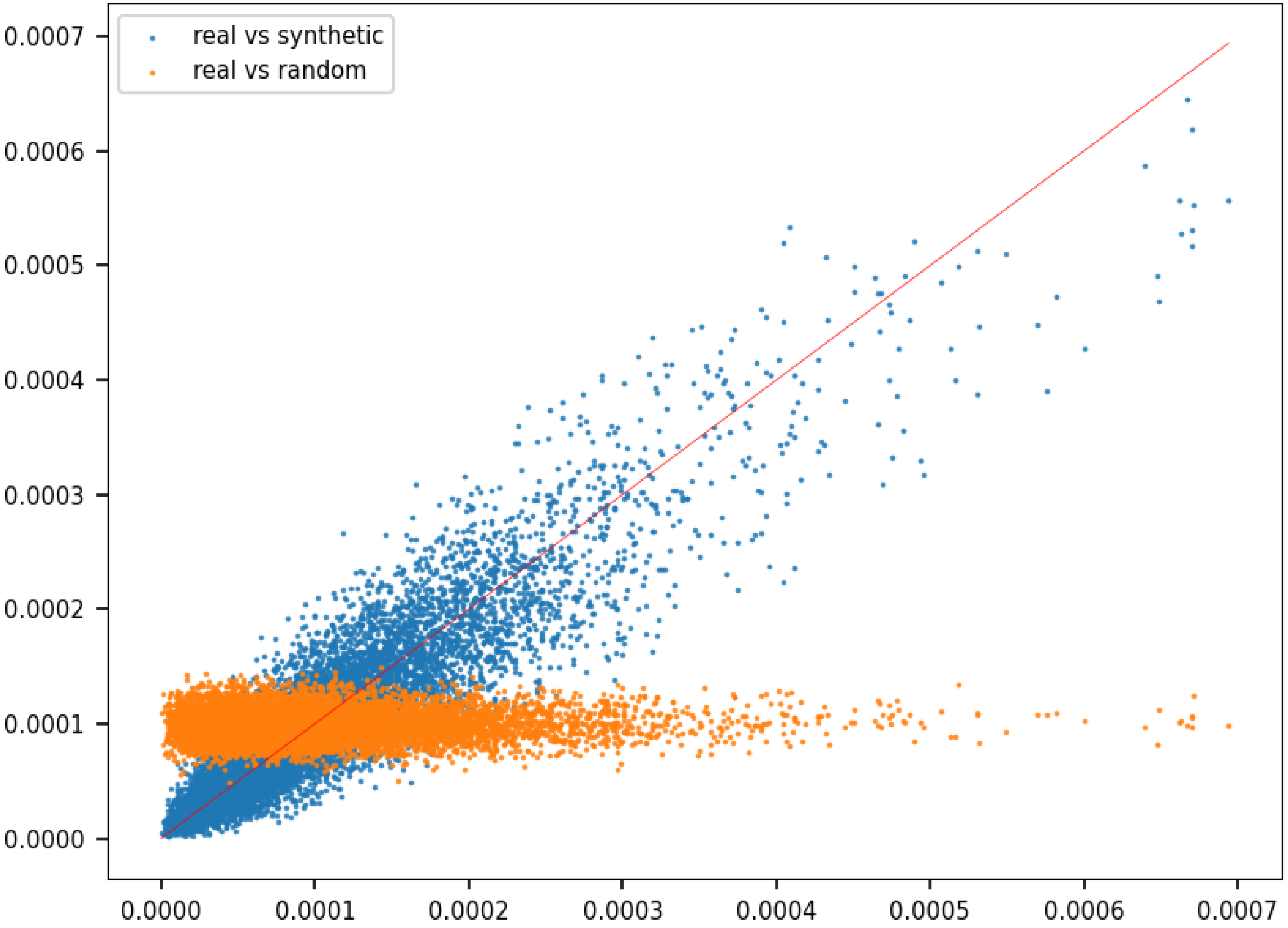
Table 1: Cluster linked to {‘Learning~vs’, ‘Machine^vs’, ‘Machine~Learning~vs’}

# Interlude – Fast Nearest Neighbor Search



- Red dot: prompt-derived embeddings
- Blue dot: backend table embedding
- Over time, arrows link red dots to their nearest blue dots
- Alternative to vector search

# xLLM for Next Token Prediction



- Next token prediction: the mother of all LLMs
- Here: predict next DNA sub-sequence to generate synthetic genomic data
- Alphabet has 4 letters
- Left: Scatterplot comparing observed vs synthetic ECDFs

# Part 5
# References

# References

- New book: “Building Disruptive AI & LLM Technology from Scratch”
- First book: “State of the Art in GenAI & LLMs – Creative Projects, with Solutions”
  - Project 2.4 – Adaptive loss function
  - Project 7.2 – Main part, includes smart crawling and x-embeddings
  - Project 8.1 – Fast approximate nearest neighbor search
  - Project 8.2 – Evaluation using taxonomy
  - Project 8.3 – xLLM for clustering and predictions
- GitHub: code, data: https://github.com/VincentGranville/Large-Language-Models
- AI Research and book access: https://mltechniques.com/resources/