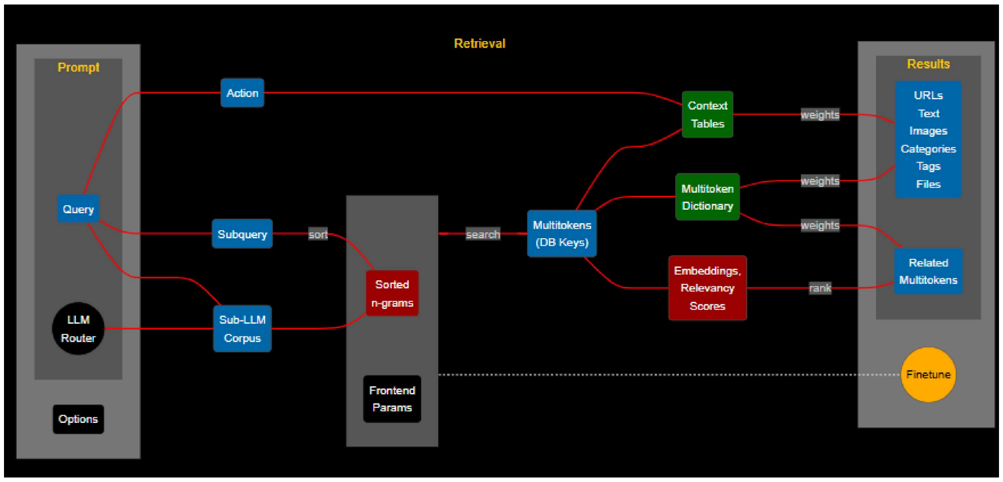
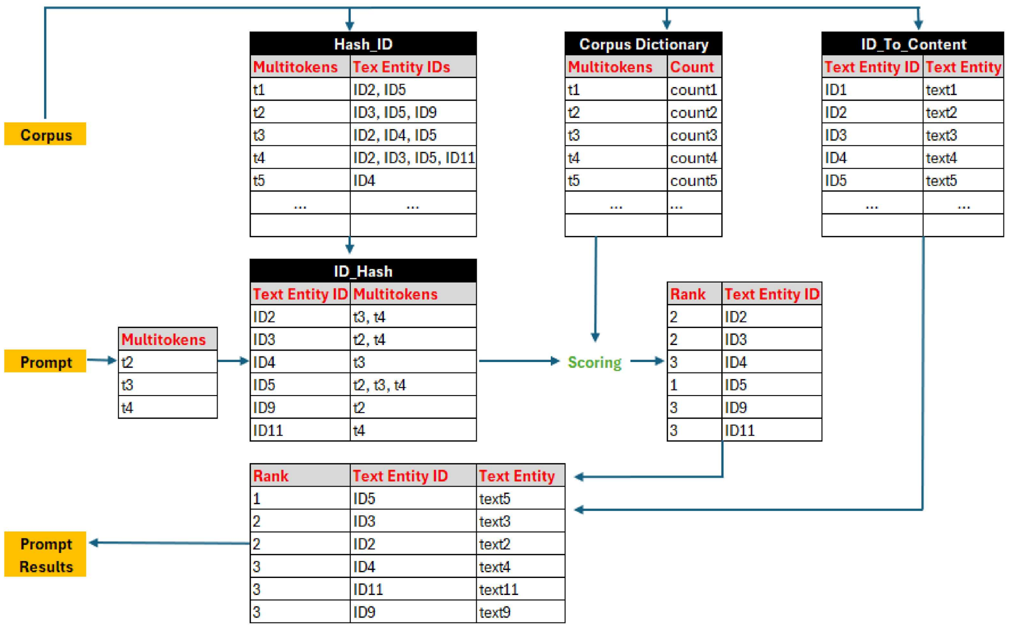
Building Disruptive AI & LLM Technology From Scratch





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
...	...

ID_Hash

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

Multitokens
t2
t3
t4

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

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

Vincent Granville, Ph.D. | www.MLTechniques.com | Version 2.0, October 2024

Contents

I Hallucination-Free LLM with Real-Time Fine-Tuning **6**

1 Enterprise xLLM for Search & Retrieval **7**
- 1.1 Efficient database architecture: nested hashes . . . 7
- 1.2 From frontend prompts to backend tables . . . 8
- 1.3 What is not covered here . . . 9

2 Parameters, features, and fine-tuning **11**
- 2.1 Backend parameters . . . 11
- 2.2 Frontend parameters . . . 12
- 2.3 Agents . . . 12
- 2.4 Reproducibility . . . 13
- 2.5 Singularization, stemming, auto-correct . . . 14
- 2.6 Augmentation, distillation, and frontend tables . . . 14
- 2.7 In-memory database, latency, and scalability . . . 15

3 Case study **17**
- 3.1 Real-time fine-tuning, prompts, and command menu . . . 18
- 3.2 Sample session . . . 19
- 3.3 Web API for enterprise xLLM . . . 29
  - 3.3.1 Left panel: command menu and prompt box . . . 29
  - 3.3.2 Right panel: prompt results . . . 29
  - 3.3.3 Next steps . . . 31
- 3.4 Conclusions and references . . . 31

4 Appendix **32**
- 4.1 Python code . . . 32
- 4.2 Thirty features to boost LLM performance . . . 44
  - 4.2.1 Fast search and caching . . . 44
  - 4.2.2 Leveraging sparse databases . . . 45
  - 4.2.3 Contextual tokens . . . 45
  - 4.2.4 Adaptive loss function . . . 45
  - 4.2.5 Contextual tables . . . 45
  - 4.2.6 Smart crawling . . . 45
  - 4.2.7 LLM router, sub-LLMs, and distributed architecture . . . 45
  - 4.2.8 From one trillion parameters down to two . . . 46
  - 4.2.9 Agentic LLMs . . . 46
  - 4.2.10 Data augmentation via dictionaries . . . 46
  - 4.2.11 Distillation done smartly . . . 46
  - 4.2.12 Reproducibility . . . 47
  - 4.2.13 Explainable AI . . . 47
  - 4.2.14 No training, in-memory LLM . . . 47
  - 4.2.15 No neural network . . . 47
  - 4.2.16 Show URLs and references . . . 48
  - 4.2.17 Taxonomy-based evaluation . . . 48
  - 4.2.18 Augmentation via prompt data . . . 48
  - 4.2.19 Variable-length embeddings, indexing, and database optimization . . . 48
  - 4.2.20 Favor backend over frontend engineering . . . 48
  - 4.2.21 Use NLP and Python libraries with caution . . . 49
  - 4.2.22 Self-tuning and customization . . . 49

4.2.23 Local, global parameters, and debugging . . . 49
4.2.24 Displaying relevancy scores, and customizing scores . . . 50
4.2.25 Intuitive hyperparameters . . . 50
4.2.26 Sorted n-grams and token order preservation . . . 50
4.2.27 Blending standard tokens with tokens from the knowledge graph . . . 50
4.2.28 Boosted weights for knowledge-graph tokens . . . 51
4.2.29 Versatile command prompt . . . 51
4.2.30 Boost long multitokens and rare single tokens . . . 51
4.2.31 Disambiguation . . . 51
4.3 LLM glossary . . . 51

II Outperforming Neural Nets and Classic AI 55

5 Building and evaluating a taxonomy-enriched LLM 57
5.1 Project and solution . . . 59
5.2 Python code . . . 60

6 LLM for Clustering and Predictions 66
6.1 Project and solution . . . 67
6.2 Visualizations and discussion . . . 69
6.3 Python code . . . 70

7 Fast, High-Quality NoGAN for Data Synthetization 78
7.1 Project description . . . 78
7.2 Solution . . . 80
7.3 Python implementation . . . 81

8 Hierarchical Bayesian NoGAN for Data Synthetization 87
8.1 Methodology . . . 87
8.1.1 Base algorithm . . . 88
8.1.2 Loss function . . . 88
8.1.3 Hyperparemeters and convergence . . . 89
8.1.4 Acknowledgments . . . 90
8.2 Case studies . . . 90
8.2.1 Synthesizing the student dataset . . . 91
8.2.2 Synthesizing the Telecom dataset . . . 93
8.2.3 Other case studies . . . 94
8.2.4 Auto-tuning the hyperparameters . . . 95
8.2.5 Evaluation with multivariate ECDF and KS distance . . . 96
8.3 Conclusion . . . 97
8.4 Python implementation . . . 98

9 Boosting Model Evaluation with Smart Adaptive Loss Functions 107
9.1 Project and solution . . . 108
9.2 Python code . . . 112

III Innovations in Statistical AI 117

10 Building a Ranking System to Enhance Prompt Results 119
10.1 Relevancy scores and rankings . . . 120
10.2 Case study . . . 121
10.2.1 xLLM for auto-indexing, cataloging and glossary generation . . . 121
10.2.2 xLLM for scientific research . . . 122
10.3 Python code . . . 123
10.4 Appendix: smart ranking in a nutshell . . . 129

11 Probabilistic Search: Alternative to Vector Search 131
11.1 Motivation and architecture . . . 132
11.2 Applications . . . 133
11.2.1 Embeddings and large language models . . . 133

11.2.2 Generating and evaluating synthetic data . . . 134
11.2.3 Clustering, dataset comparisons, outlier detection . . . 135
11.3 Project and solution . . . 136
11.4 Python code . . . 137

12 Strong Random Generators for Reproducible AI 141
12.1 Strong Randomness and reproducibility: two key components . . . 141
12.2 Computing the digits of special math constants . . . 142
12.2.1 P-adic valuations . . . 142
12.2.2 Digit blocks, speed of convergence . . . 143
12.2.3 A plethora of interesting pseudo-random sequences . . . 143
12.3 Testing random number generators . . . 144
12.3.1 Theoretical properties of the digits of $\sqrt{2}$ . . . 144
12.3.2 Fast recursion and congruential equidistribution . . . 145
12.3.3 Exponential system: predicting the next block . . . 147
12.4 Python code . . . 149
12.4.1 Fast recursion . . . 149
12.4.2 Main code . . . 150

13 Sampling Outside the Observation Range 155
13.1 Quantile convolution . . . 155
13.2 Truncated Gaussian mixtures and bias detection . . . 157
13.3 Case studies . . . 157
13.3.1 Conclusion . . . 158
13.4 Python code . . . 159

14 Additional Resources 162
14.1 Accelerating convergence of parameter estimates . . . 162
14.1.1 First case study . . . 162
14.1.2 Second case study . . . 162
14.2 Generic, all-in-one curve fitting, regression and clustering . . . 164
14.2.1 Solution, R-squared and backward compatibility . . . 165
14.2.2 Model upgrades . . . 166
14.2.3 Case studies . . . 166
14.2.4 Python code . . . 170
14.3 A simple geospatial interpolation method . . . 171
14.3.1 Problem in two dimensions . . . 171
14.3.2 Spatial interpolation of the temperature dataset . . . 173
14.3.3 Python code . . . 174
14.4 Math-free, parameter-free gradient descent . . . 174
14.4.1 Introduction . . . 175
14.4.2 Implementation details . . . 175
14.4.3 General comments about the methodology and parameters . . . 178
14.4.4 Mathematical version of gradient descent and orthogonal trajectories . . . 179
14.4.5 Python code . . . 180

15 Trading the S&P 500 Index 181
15.1 Non-standard strategies . . . 182
15.2 Selecting a strategy based on its features . . . 183
15.3 Python code and dataset: 40 years' worth of historical data . . . 184

Bibliography 188

Index 190

Introduction

This book features new advances in game-changing AI and LLM technologies built by GenAItechLab.com. Written in simple English, it is best suited for engineers, developers, data scientists, analysts, consultants and anyone with an analytic background interested in starting a career in AI. The emphasis is on scalable enterprise solutions, easy to implement, yet outperforming vendors both in terms of speed and quality, by several orders of magnitude.

Each topic comes with GitHub links, full Python code, datasets, illustrations, and real life case studies, including from Fortune 100 company. Some of the material is presented as enterprise projects with solution, to help you build robust applications and boost your career. You don't need expansive GPU and cloud bandwidth to implement them: a standard laptop works.

Part I focuses on high performance in-memory agentic multi-LLMs for professional users and enterprise, with real-time fine-tuning, self-tuning, no weight, no training, no latency, no hallucinations, no GPU. Made from scratch, leading to replicable results, leveraging explainable AI, adopted by Fortune 100. With a focus on delivering concise, exhaustive, relevant, and in-depth search results, references, and links. See also the section on 31 features to substantially boost RAG/LLM performance.

Part II discusses related large-scale systems also benefiting from a light-weight but more efficient architecture. It features LLMs for clustering, classification, and taxonomy creation, leveraging the knowledge graphs embedded in and retrieved from the input corpus when crawling. Then, in chapters 7 and 8, I focus on tabular data synthetization, presenting techniques such as NoGAN, that significantly outperform neural networks, along with the best evaluation metric. The methodology in chapter 9 applies to most AI problems. It offers a generic tool to improve any existing architecture relying on gradient descent, such as deep neural networks.

Part III features a collection of methods that you can integrate in any AI system to boost performance. Based on a modern approach to statistical AI, they cover probabilistic vector search, sampling outside the observation range, strong random number generators, math-free gradient descent, beating the slow statistical convergence of parameter estimates dictated by the Central Limit Theorem, exact geospatial interpolation for non-smooth systems, and more. Efficient chunking and indexing for LLMs is the topic of chapter 10. Finally, chapter 15 shows how to optimize trading strategies to consistently outperform the stock market.

About the author

Vincent Granville is a pioneering GenAI scientist and machine learning expert, co-founder of Data Science Central (acquired by a publicly traded company in 2020), Chief AI Scientist at MLTechniques.com, former VC-funded executive, author and patent owner – one related to LLM. Vincent's past corporate experience includes Visa, Wells Fargo, eBay, NBC, Microsoft, and CNET.

Vincent is also a former post-doc at Cambridge University, and the National Institute of Statistical Sciences (NISS). He published in *Journal of Number Theory*, *Journal of the Royal Statistical Society* (Series B), and *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is the author of multiple books, available here, including "Synthetic Data and Generative AI" (Elsevier, 2024). Vincent lives in Washington state, and enjoys doing research on stochastic processes, dynamical systems, experimental math and probabilistic number theory.

Index

χ^2 distribution, 147
k-NN, 52, 132
k-means clustering, 67
k-medoids clustering, 67
n-gram, 8
  sorted, 50
p-adic valuation, 142
p-value, 147, 149

abbreviation dictionary, 14
acronyms, 46, 52
action, 8, 12, 51
activation function, 97
Adam (stochastic gradient descent), 52
agent, 12, 18, 29, 46, 53, 119, 130
  multi-agent, 51
ANN (approximate nearest neighbors search), 44, 54, 134
  probabilistic ANN (pANN), 52, 134
augmentation, 15, 48, 52
  augmented knowledge graph, 15
  taxonomy augmentation, 15, 122
auto-correct, 14
auto-encoder, 52
auto-regressive model, 53
auto-tuning, 89, 95, 97

backend, 29, 52, 53, 119
base, 146
basin of attraction, 182
batch, 89, 91, 97
Bayesian hierarchical models, 90
benchmarking, 52
Beurling primes, 164
binary search
  interpolated, 135
binning, 108
bit shift, 145
bootstrapping, 69, 166

caching, 44, 54
cataloging, 122
categorical feature, 89
CDF regression, 167
central limit theorem, 162
chunking, 121, 129
confidence interval, 69, 162
  model-free, 81
confidence region, 166
congruential equidistribution
  asymptotic, 145
connected components, 67
context, 119
  contextual pairs, 45
  contextual tables, 45
  contextual tokens, 45
contour level, 175, 179
contour lines, 175
convergence
  absolute, 171
  conditional, 171
convolution product, 155
copula, 90, 98
cosine similarity, 54, 134
covariance matrix, 110
Cramér's V, 89
crawling
  smart crawling, 8, 45
critical line (number theory), 175
cross-validation, 48, 68, 87, 166
customization (LLM), 49

data distillation, 79
database
  graph database, 53
  JSON, 48
  key-value pairs, 48
    nested hash, 48
  vector database, 48
debugging, 49
deep neural network (DNN), 52, 97, 107
dendrogram, 69
diffusion, 52, 80
diffusion model, 155
digit block, 143
digit-preserving, 143
dimensionality reduction, 166
Dirichlet character, 164
Dirichlet-L function, 164
disembiguation, 51
distance matrix, 66
distillation, 14, 46, 52
distributed computing, 46
distribution
  logistic, 166
DNN (deep neural network), 45
dot product, 133, 165
dummy variable, 89
dynamical system, 182

ECDF (empirical distribution), 54, 78, 88, 107
embedding, 19, 45, 52

quantized embedding, 48
variable length embedding, 7, 48, 133
empirical density function, 96
empirical distribution, 67, 167
multivariate, 54, 78, 87, 90, 96, 107, 135
encoding
smart encoding, 68
EPDF (empirical probability density function), 107
epoch (neural networks), 132, 137
equidistribution modulo 1, 146
Euler product, 164
evaluation, 107
evaluation metrics, 48, 52, 53
exhaustivity (LLM/RAG output), 119
explainable AI, 50, 52, 78, 87, 97

feature encoding, 54
feature engineering, 53
feature selection, 166
fine-tuning, 52, 130
exhaustive results, 47, 119
in real time, 17
LLM parameters, 46
self-tuning, 49
frontend, 52

GAN (generative adversarial network), 13, 78, 107, 155
hierarchical GAN, 90
NoGAN, 14
Gaussian mixture model, 110, 155
generative adversarial network, 52, 80, 87, 134, 155
GIS, 173
GPT, 52
GPU, 47
gradient descent, 69, 88, 97, 110, 132, 179, 181
steepest, 136
stochastic, 107, 136, 137
gradient operator, 165
graph
graph database, 53
knowledge graph, 48
GRH (Generalized Riemann Hypothesis), 164
grid search, 79, 181
smart grid search, 47, 111, 181

Hadamard product, 88
hallucination, 48
hash table
inversion, 66
nested hash, 8, 45, 46, 48, 53, 66, 119, 122
Hellinger distance, 96, 107
multivariate, 45
hierarchical Bayesian model, 98
hierarchical clustering, 67
hierarchical deep resampling, 87, 90
holdout, 90
Hungarian algorithm, 107, 132
hyperparameter, 47, 50, 52, 79, 89
hyperrectangles, 78

identifiability (statistics), 131
imputation (missing values), 90
in-memory database, 16
in-memory LLM, 16, 47
indexing, 129
auto-indexing, 122
glossary generation, 122
text entity index, 48, 119
integer division, 145
integral probability metrics, 96
interpolation, 182

JSON, 8, 45
database, 48, 53

kernel, 155
kernel density estimation, 155
key-value database, 8, 53
knowledge graph, 48, 119, 129
Kolmogorov-Smirnov distance, 67, 78, 89, 90, 96, 109, 135
multivariate, 45, 54, 107
kriging, 171

label, 95
Lagrange multiplier, 165
LangChain, 53
large language model (LLM)
evaluation, 122
large language models (LLM), 57
Lasso regression, 165
latency, 15
law of the iterated logarithm, 147
learning rate, 177
link function, 167
LLaMA, 45, 53
LLM (large language model)
agentic LLM, 12
debugging, 49
for auto-indexing, 122
for cataloging, 122
for clustering, 13
for glossary generation, 122
for predictive analytics, 13
in-memory, 16
LLM router, 8, 46, 119
multi-LLM, 17
self-tuned, 49
sub-LLM, 17, 119
xLLM, 17
logistic distribution, 166
logistic regression, 166
loss function, 67, 69, 88, 89, 97, 107, 132, 136, 181
adaptive loss, 45, 107

matrix
positive semidefinite, 110
mean squared error, 165
Mersenne twister, 141, 144
mixture model, 96
mixture of experts, 46, 49, 119
model identifiability, 165
modulus (complex number), 175
multi-agent system, 51, 53

multimodal system, 53
multinomial distribution, 78
multiple root, 171
multitoken (see token), 52

nearest neighbors
  K-NN, 131
  approximate (ANN), 131
    probabilistic (pANN), 131
  search, 132
NLG (natural language generation), 53
NLP (natural language processing), 14, 53
node (interpolation), 96
NoGAN, 52, 78, 108, 134
  constrained, 108
  probabilistic, 108
normal number, 146
normalization, 53, 54
notebookML, 122

OpenAI, 110
orthogonal trajectory, 175
overfitting, 47, 67, 78, 165, 181

parameter
  in neural networks, 53, 54
  in xLLM, 53
parametric bootstrap, 170
partial derivative, 171
period, 142
PMI (pointwise mutual information), 7, 9, 19, 31, 49, 50
pointwise mutual information (PMI), 54, 133
Poisson process, 137
positive semidefinite (matrix), 110
prediction interval, 166
prefix, 142
principal component analysis, 90
PRNG (pseudo-random number generator), 14, 47, 141
probability density function, 155
prompt
  command prompt options, 48
  prompt compression, 134
  prompt engineering, 48
  prompt synthesizer, 122
Python library
  GenAI-evaluation, 109
  Sklearn, 67
  Sklearn_extra, 67

quantile, 79, 109
  extrapolated, 157
quantile function, 91, 155
quantile regression, 166
quantization, 48

R-squared, 165, 166
Rademacher function, 164
radix search, 135
RAG (retrieval augmentation generation), 15, 53, 119
random walk
  simple, 147
rank
  relevancy rank, 120
regularization, 54
reinforcement learning, 49, 54, 97
rejection sampling, 157
relevancy
  relevancy score, 47, 50, 119, 130
replicability, 47
reproducibility, 14, 47
resampling, 108
residue, 145
retrieval, 48
  retrieval augmentation generation (RAG), 53
Riemann hypothesis
  generalized, 164
Riemann zeta function, 164, 175
run test, 145

saddle point, 177
scalability, 15
score
  relevancy rank, 120
  relevancy score, 29, 47
  smart scoring, 130
search
  probabilistic search, 48
  vector search, 48
seed (random number generator), 13, 89
self-tuning, 52, 54
separator (text), 54
similarity metric, 67
simulated annealing, 97
singularization, 14
steepest descent, 179
stemming, 14
stopwords, 8, 49
swap, 88, 97
synonyms dictionary, 14
synthetic data, 54, 107, 157, 171, 174
  constrained, 108, 111
synthetic function, 164

taxicab distance, 137
taxonomy, 53
  augmentation, 122
  taxonomy creation, 57, 122
tensor, 89
TensorFlow, 89
text entity, 12, 29, 46, 48, 52, 119
  sub-entity, 18
  text entity ID, 29
time complexity, 79
token, 52, 54
  contextual, 45, 66
  graph token, 119, 121, 130
  multitoken, 7, 19, 52, 66, 119
training, 47, 52
transformer, 13, 45, 47, 52, 54, 97
truncated Gaussian, 157

unit ball, 137

validation set, 78, 90, 166
vanishing gradient, 89
variational auto-encoder, 52
vector database, 48, 54
vector search, 48, 54, 131
vectorization, 79

Wasserstein GAN (WGAN), 97, 107
Weibull distribution, 137
weighted regression, 166
Wiener process, 147

XGboost, 78
xLLM, 17, 57
    e-xLLM (for enterprise corpus), 119