

Neural Networks for Natural Language Processing

Jun.-Prof. Sophie Fellenz

13.01.2025

Week 10 – Fine-tuning, RAG

Invitation to a lecture at the RPTU Kaiserslautern:

- Title: **AI/ML/Generative AI at Amazon: What we learned**
- **AWS pioneer Constantin Gonzalez Schmitz** on how Amazon applies (Gen) AI/ML to innovate, optimize, and scale globally!
- The event is **free of charge**. Please register at: <https://www.eventbrite.com/e/aimlgenerative-ai-at-amazon-what-we-learned-tickets-1139713707379?aff=oddtcreator>
- Date and Time: **Tuesday, January 14 2025, 11:30 – 13:15**
- Where: **Audimax 42-115, RPTU Kaiserslautern, Erwin-Schrödinger-Straße**

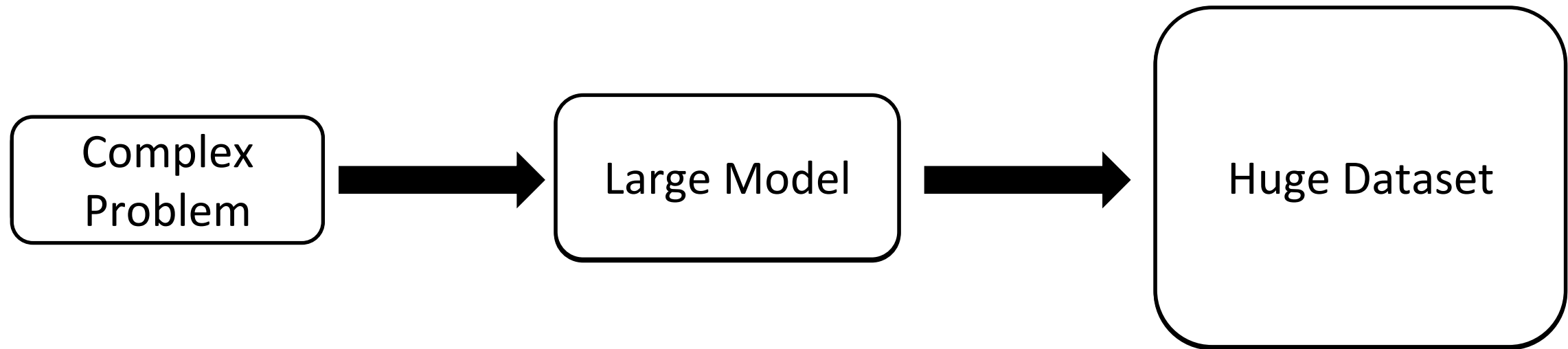
Agenda

- Advanced Fine-Tuning
- RAG

Motivation Finetuning

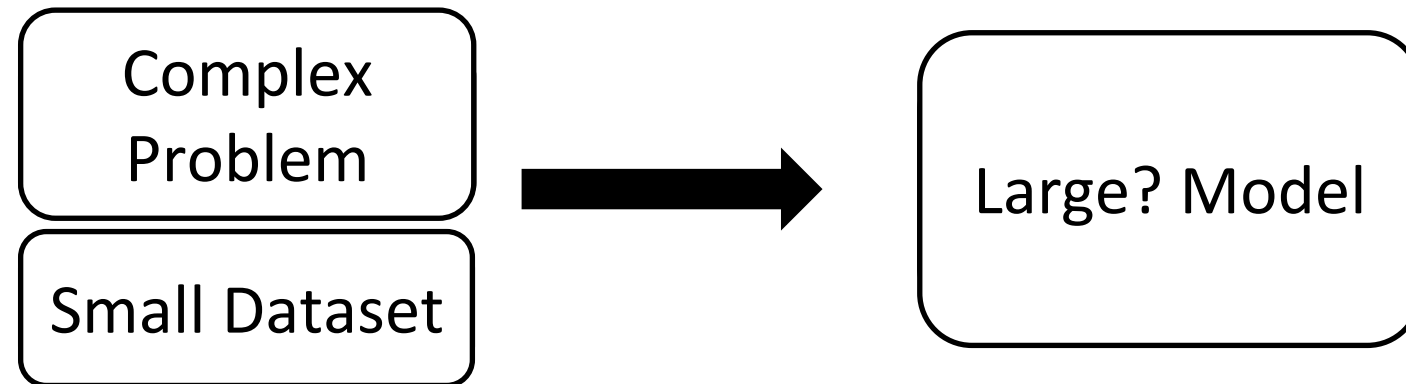
Motivation

Ideally:



Motivation

Realistically:

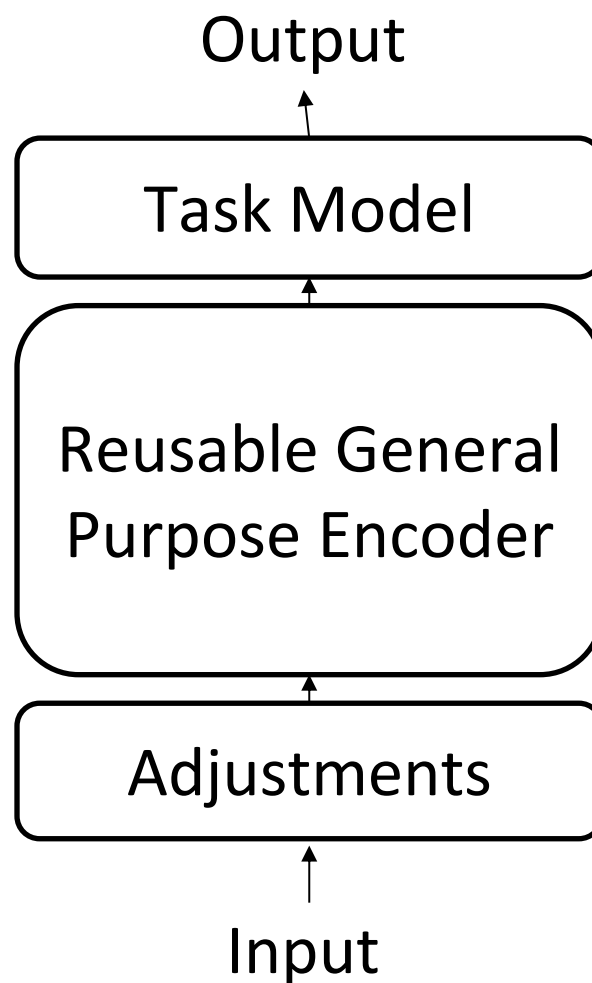


Foundation Model

- Develop general purpose neural network encoder for text
- Usable on any new language understanding task
- Minimal learning for application task

⇒ Make model that already understands English language

General Purpose Encoder



Zero-shot vs. Few-shot

Zero-Shot and Few-Shot Learning

- GPT-3 almost impossible to fine-tune so we sometimes don't
- Zero-shot learning: Ask model to follow instructions
- Few-shot learning: Ask model to learn from few examples

Zero-Shot and Few-Shot Learning

Zero-Shot

Translate English to French

cheese → ?

Few-Shot

Translate English to French

hello → bonjour

wine → vin

bread → pain

cheese → ?

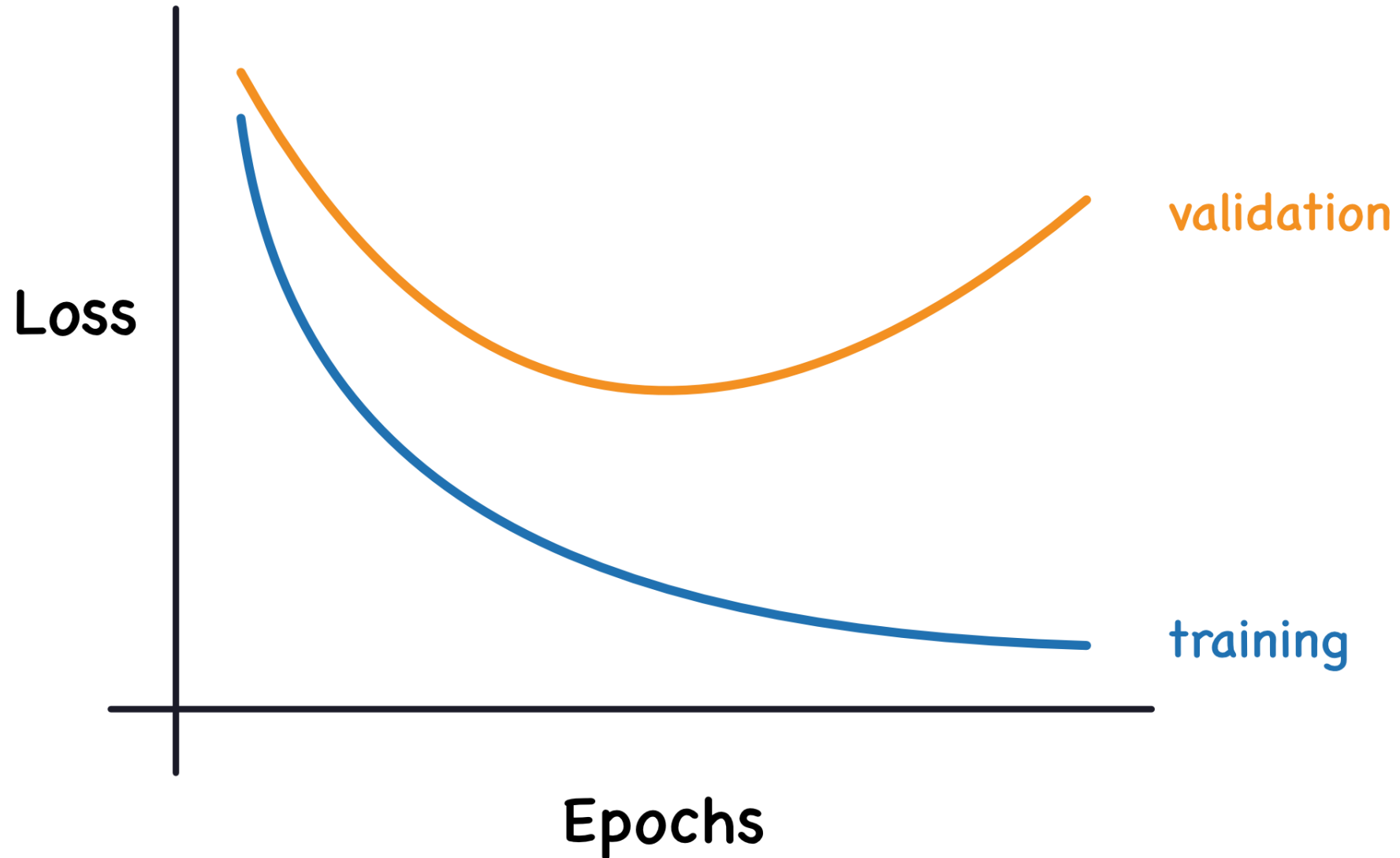
Challenges of Finetuning

Catastrophic forgetting

- Neural network trained on a new task loses its ability to perform well on a previously learned task
- Example: autonomous vehicle model is fine-tuned for learning a new road sign and forgets to recognize pedestrians
- Techniques to address this:
 - Experience replay
 - Back-regularization to previous model states
 - ...

Overfitting

The Learning Curves



Transfer Learning Optimization

Freezing vs. Full Finetuning

- Freezing: Keep certain layers of the pre-trained model fixed (frozen) and only update others during fine-tuning.
 - Reduces computational cost and memory usage
 - Prevents Overfitting
 - Preserves General Features
- Trade-off Flexibility vs. Stability
 - Freezing focuses on leveraging pre-trained knowledge.
 - Full finetuning enables greater flexibility at the cost of potentially losing generalizable features.

Gradual unfreezing of layers

- A fine-tuning strategy where layers of a pre-trained model are progressively unfrozen, starting with the topmost layers (closest to the output) and moving downwards.
- Balance the preservation of pre-trained knowledge with adaptation to the target task.

How It Works

1. Start with Frozen Layers:
 - Initially, all layers except the final few are frozen.
 - Focus on fine-tuning task-specific features in the final layers.
2. Progressively Unfreeze Lower Layers:
 - Gradually unfreeze earlier layers, one at a time or in groups.
3. Adjust Learning Rates:
 - Use a smaller learning rate for previously frozen layers to prevent overwriting pre-trained knowledge.

Layer-wise learning rate adjustments

A fine-tuning strategy where different layers of a neural network are assigned different learning rates.

Lower Layers:

- Typically learn general features (e.g., edges in vision, syntax in language models).
- Use smaller learning rates to preserve pre-trained knowledge.

Higher Layers:

- Capture task-specific features and require more adjustment.
- Use larger learning rates to adapt to the target task.

Regularization Techniques for Finetuning

Dropout and DropConnect

Dropout

Definition: A regularization technique where a random subset of neurons is "dropped out" (set to zero) during training, forcing the model to learn robust, distributed representations.

- Works with both fully-connected and convolutional layers
- Easy to implement, coarse regularization

DropConnect

Definition: A variant of Dropout where instead of dropping neurons, individual weights are randomly set to zero during training.

- Works well with fully connected layers
- Harder to implement, more granular regularization

Weight Regularization

- Techniques to constrain the magnitude of model weights during training to prevent overfitting and improv
- Prevents Overfitting: Encourages simpler models that generalize better to unseen data.
- Stabilizes Optimization: Reduces large weight updates, leading to smoother convergence.
- Improves Generalization: Helps adapt pre-trained weights to new tasks without over-specialization. e generalization.

How It Works

- Add a regularization term to the loss function:
 - $\text{Loss} = \text{Task Loss} + \lambda \cdot R(W)$
 - where $R(W)$ is the regularization term and λ controls its strength.
- Regularization types:
 - L1: $R(W) = \sum |w|$
 - L2: $R(W) = \sum w^2$

Mixout

- **Definition:** Mixout is a dropout-inspired regularization method that stochastically replaces a portion of a model's weights with their pre-trained values during training.
- **Purpose:** Helps retain knowledge from pre-trained weights while allowing the model to adapt to the fine-tuning task.
- Prevents Catastrophic Forgetting
- Improves Generalization
- Encourages smoother optimization compared to standard dropout by mixing pre-trained and updated weights

Data Augmentation for Finetuning

Data Augmentation

- **Enhance Diversity:** Apply transformations (e.g., cropping, flipping, or masking) to increase data variability and improve model generalization.
- **Task-Specific Augmentations:** Use augmentations relevant to the domain, such as synonym replacement in text, noise addition in audio, or geometric transformations in images.
- **Synthetic Data Generation:** Generate new samples using techniques like GANs, data synthesis, or back-translation (for text) to enrich training datasets.
- **Augmentations for Low-Resource Scenarios:** Leverage techniques like data interpolation or mixup to create additional training examples for underrepresented classes.
- **Cross-Domain Augmentations:** Borrow augmentations from related domains to transfer knowledge and improve robustness in the target domain.
- **Consistency Constraints:** Use augmentations to enforce consistency in model predictions under varied input conditions, improving robustness.

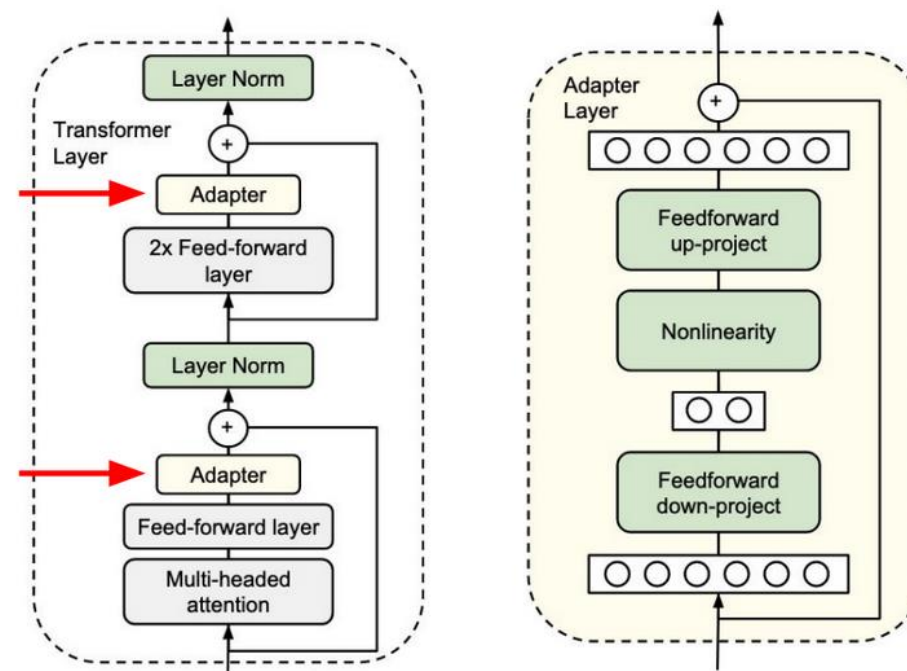
Data Augmentation Examples

- Synonym Replacement: Replace words with synonyms (e.g., "great" → "excellent").
- Back-Translation: Translate the text to another language and back (e.g., English → French → English).
- Fine-tuning a model for legal document classification (e.g., identifying contract clauses).
 - Cross-Domain Augmentation: Use data from a related but different domain, such as news articles or financial reports, to generate training data for legal text classification.

Advanced Model Customization

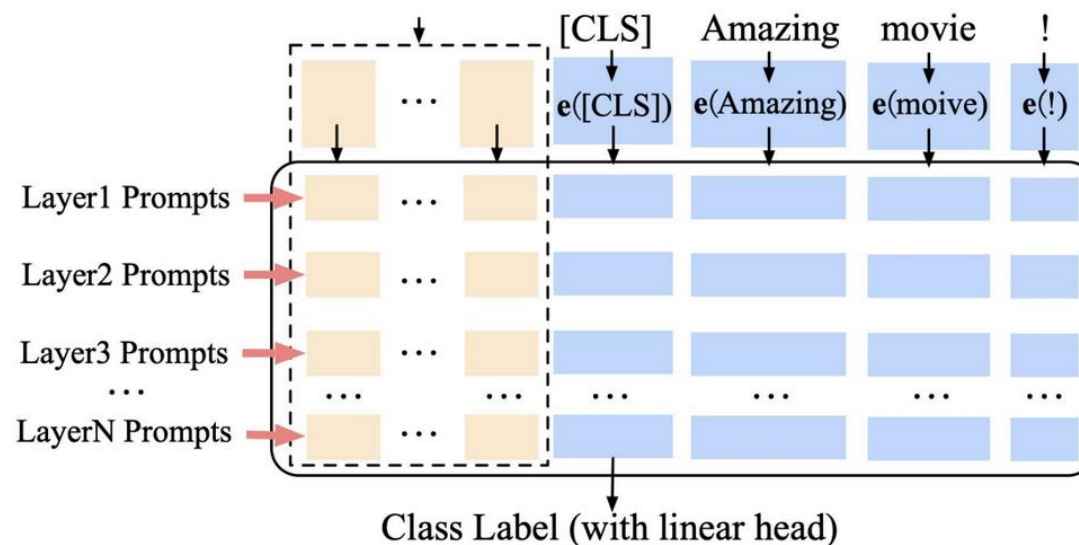
Adapters for Parameter-Efficient Tuning

- Adapter layer within model layers
 - Inject a small tunable component (adapter) in the model
 - Freeze rest of the model parameters
- ~3.6% of the model parameters
- Comparable performance to full fine-tuning



Prompt Tuning

- Like having tunable layer(s) of input embeddings in the model
- For a given task, we can freeze the model and inject a new set of tunable input representations
- Maybe only the first layer, or every layer
- Works best for large models, and has many variants



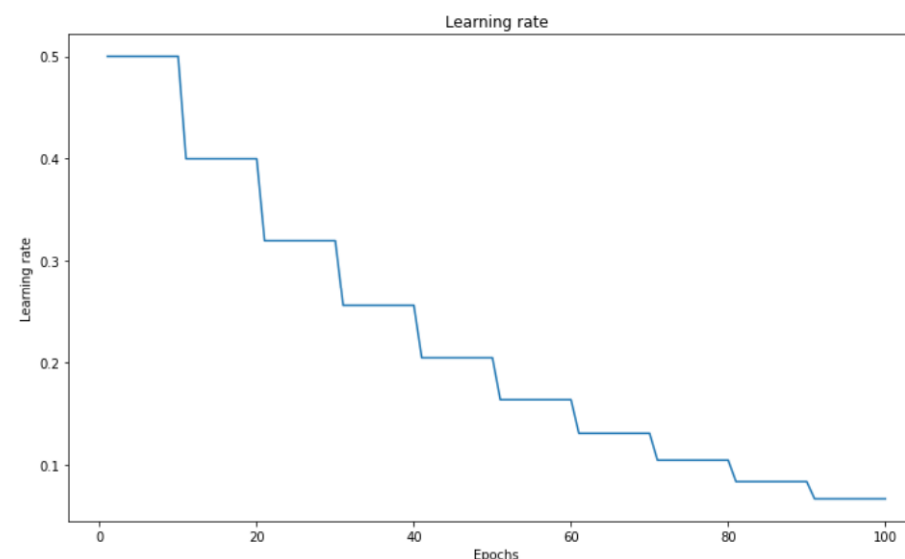
Optimization and Training Dynamics

Learning Rate Schedules

- **Definition:** Learning rate schedules dynamically adjust the learning rate during training to improve convergence and model performance.
- **Purpose:** Helps balance stability and exploration during optimization.

Common Learning Rate Schedules

- Step Decay
- Exponential Decay
- Cosine Annealing
- Warm-Up with Decay
- Cyclic Learning Rates (CLR)
- Linear Decay with Warm Restarts



Loss Function Engineering

- Task-Specific Customization
- Handling Class Imbalance
- Multi-Objective Optimization
- Auxiliary Losses: Introduce auxiliary losses to guide fine-tuning, especially in scenarios where additional information can be leveraged (e.g., intermediate feature matching or consistency constraints).
- ...

Curriculum Learning

Key Principles:

- **Gradual Complexity:** Start with simpler examples and progress to harder ones for smoother learning.
- **Dynamic Adjustment:** Adapt data difficulty based on model performance during training.
- **Task-Specific Sequencing:** Prioritize data based on relevance or difficulty to align with task goals.
- **Scaffold Training:** Use auxiliary tasks or sub-objectives as stepping stones to the main task.
- **Data Sampling:** Begin with clean, smaller subsets before introducing the full dataset.

Practical Strategies:

- Multi-stage fine-tuning for generalization and detail focus.
- Gradual introduction of complex augmentations.
- Balance exposure to diverse examples at each stage.
- Leverage human-graded difficulty for meaningful progression.

Contrastive Finetuning

Key Concepts:

- **Contrastive Loss:** Aligns similar pairs, separates dissimilar ones (e.g., NT-Xent, SupCon).
- **Pair Selection:** Use positive pairs (similar) and negative pairs (dissimilar) to refine embeddings.

Strategies:

- **Data Augmentation:** Create diverse views for stronger positive pairs.
- **Hard Negatives:** Focus on challenging dissimilar examples to sharpen boundaries.
- **Pretrained Models:** Leverage strong initial embeddings for better results.

Applications:

Few-shot learning, search/retrieval, multimodal alignment (e.g., text & images).

Cross-lingual Fine-tuning

Beyond English

Key Concepts:

- Multilingual Transfer: Fine-tuning a model on one language to generalize across multiple languages.
- Shared Representations: Leverage shared linguistic features to align embeddings across languages.

Strategies:

- Pretrained Multilingual Models: Start with models like mBERT, XLM-R for a strong multilingual foundation.
- Translation-Based Training: Use translated or parallel data to align representations.
- Language-Specific Adaptation: Fine-tune separately for low-resource languages to handle unique nuances.
- Data Augmentation: Generate synthetic multilingual data to boost diversity and performance.

Applications:

- Cross-Lingual Retrieval: Enable searches across languages.
- Multilingual NLP Tasks: Enhance tasks like classification, NER, and QA for multiple languages.
- Low-Resource Support: Improve performance for languages with limited training data.

Post-hoc Methods vs. In-training adjustments

Post-Hoc Methods

b) Advantages

- Model Agnostic: Can be applied to any trained model, regardless of architecture.
- Less Computationally Intensive: No need to retrain or fine-tune the model from scratch.
- Ease of Implementation: Often simpler than modifying the training pipeline.

c) Limitations

- Reactive Approach: Addresses bias after it has already been encoded in the model.
- Limited Effectiveness: May not fully eliminate bias, especially if it is deeply embedded in the model.
- May Sacrifice Accuracy: Adjusting outputs can reduce the model's predictive performance.

Post-Hoc Methods

- Bias Filtering: Identify and filter out biased outputs in real-time using pre-defined rules or heuristics.
- Reweighting Outputs
 - Adjust the model's output probabilities to reduce disparities across sensitive groups.
 - Example: Normalize sentiment scores across gender groups to reduce biased sentiment prediction.
- Counterfactual Data Augmentation
 - Scenario: A hiring model predicts job suitability based on resumes.
 - **Original Input:** "John graduated from MIT and has 5 years of experience."
 - **Counterfactual Input:** "Jane graduated from MIT and has 5 years of experience."
 - **Objective:** Ensure the model's predictions for "John" and "Jane" are consistent.

In-training Adjustments

- Adjust Loss Function
 - Example: upweight underrepresented samples
- Balanced Data Sampling

In-training Adjustments

b) Advantages

- Proactive Bias Mitigation: Addresses biases before they are encoded in the model.
- Tailored Fairness: Allows for customized fairness metrics (e.g., equal opportunity, equalized odds).
- Better Integration: Often results in models that achieve both fairness and accuracy more effectively.

c) Limitations

- Increased Complexity: Requires changes to the training pipeline or model architecture.
- Higher Computational Cost: May involve adversarial training or additional constraints that slow down training.
- Trade-Offs with Accuracy: Striking the right balance between fairness and predictive performance can be challenging.

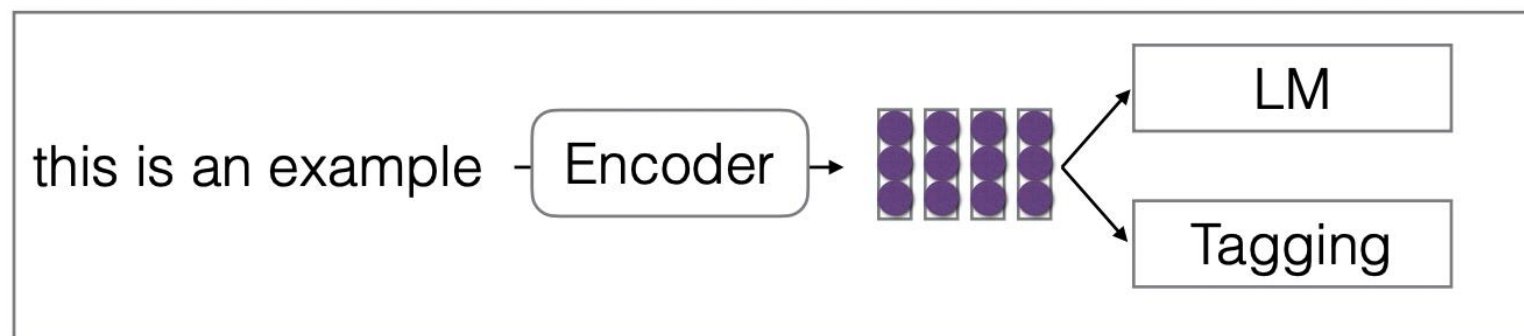
Safety vs. Helpfulness

- A question-answering model that is maximally safe will always say „Sorry, I cannot help you.“
- A maximally helpful model will always try to answer the question, even if it asks something unethical.
- Trade-off needs to be taken into account during fine-tuning

Multi-task learning

Standard Multi-task Learning

- Train representations to do well on multiple tasks at once



Source: <https://phontron.com/class/anlp2022/assets/slides/anlp-07-pretraining.pdf>

- Often as simple as randomly choosing minibatch from one of multiple tasks

Retrieval Augmented Generation

USING LOCAL LLMS WITH LOCAL DATA

BASE MODEL

What was the name of that movie Chris emailed me about last year?



Sorry, I don't have access to any specific information about emails or ...

BASE MODEL + USER DATASET

What was the name of that movie Chris emailed me about last year?

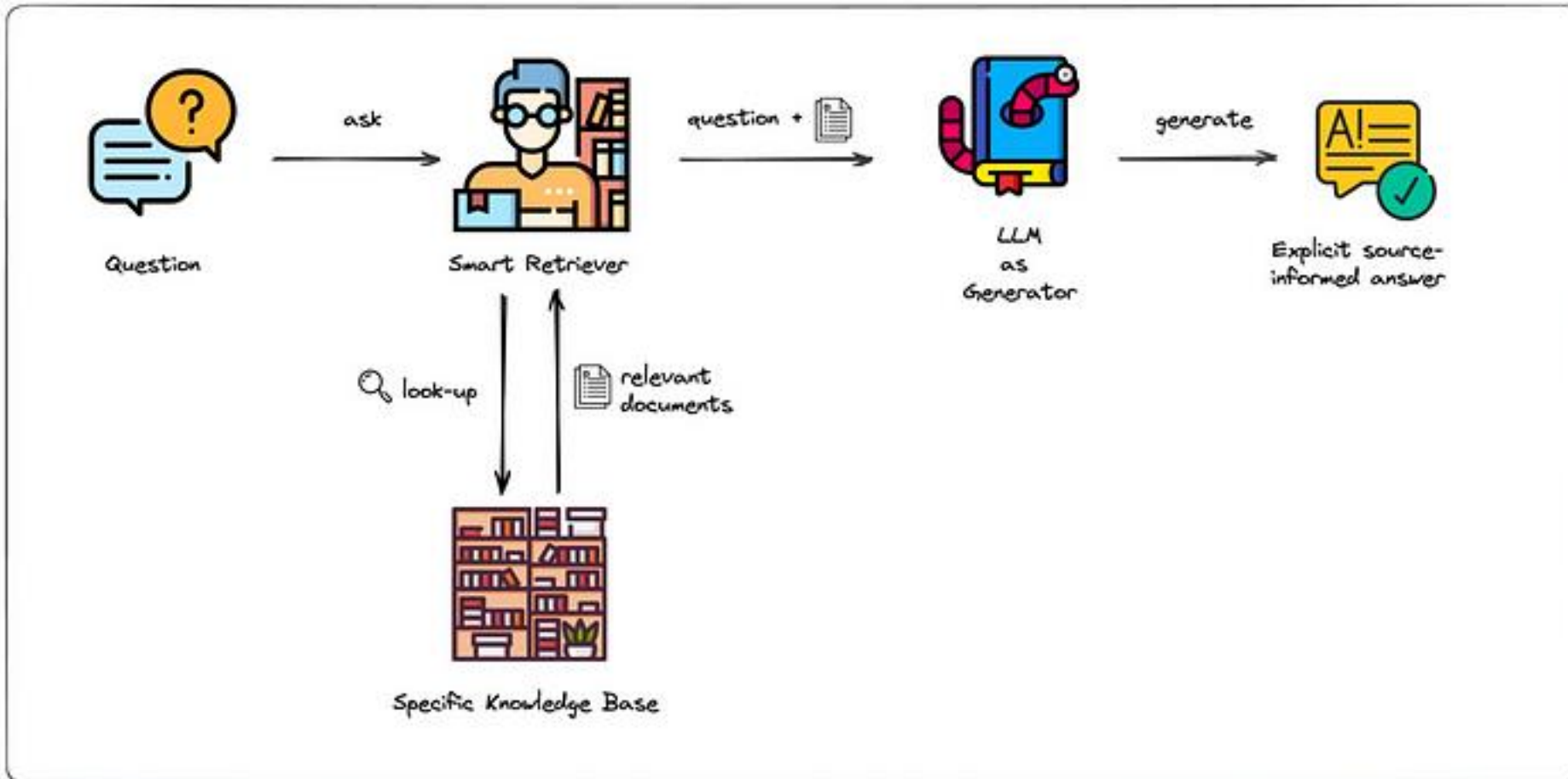


Chris Thomson emailed you about "The Fall" on October 23rd last year. The Fall is an adventure fantasy film released in 2006, starring...



Source: <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/>

Simplified RAG Architecture



Source: <https://blog.ml6.eu/leveraging-llms-on-your-domain-specific-knowledge-base-4441c8837b47>

Why RAG

Limitations of purely generative models:

- Incomplete knowledge in pretrained models.
- Knowledge staleness (e.g., GPT may not know recent events).
- Hallucinations (confidently generating incorrect information).

Advantages of combining retrieval:

- Access to up-to-date, external information.
- Improved factual accuracy and grounding.
- Better performance in specialized domains (e.g., legal, medical).

Key Components

1. Retriever:

- Finds relevant documents or passages based on the input query.
- Common retrievers: BM25, dense retrieval models like DPR (Dense Passage Retriever).

2. Generator:

- A generative model (e.g., GPT, BART, T5) that uses the retrieved context to produce responses.

3. Fusion Mechanism:

- How retrieved documents are integrated with the input query to generate an output.

Retrieval Techniques

- Sparse Retrieval
 - TF-IDF, BM25: Lexical matching-based retrieval.
- Dense Retrieval
 - Embedding-based retrieval using neural networks (e.g., Sentence-BERT, DPR).
 - Advantages of dense retrieval in capturing semantic similarity.

Fusion Strategies

1. Early Fusion:

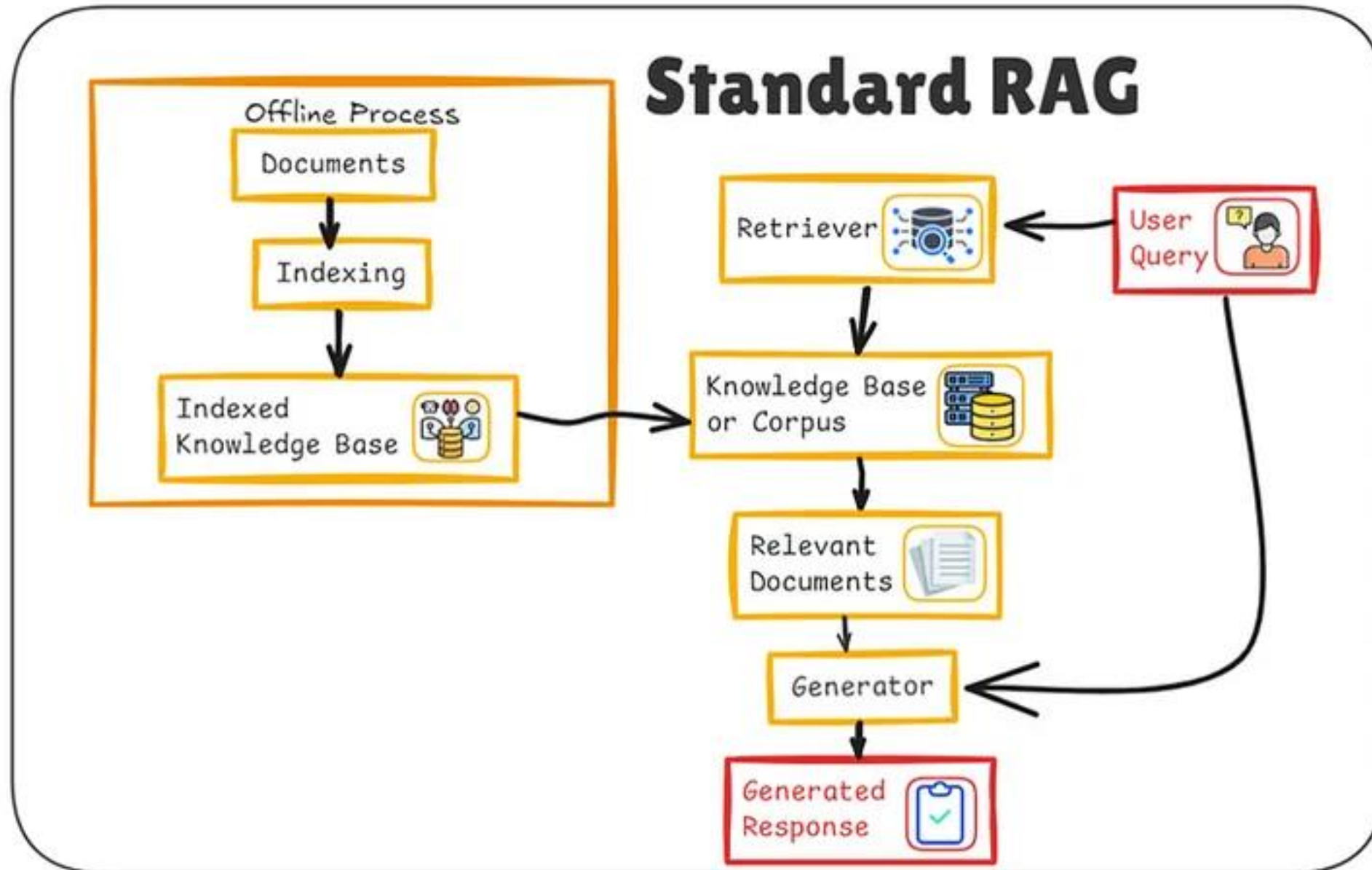
- Concatenate retrieved documents with the query as a single input to the generator.

2. Late Fusion:

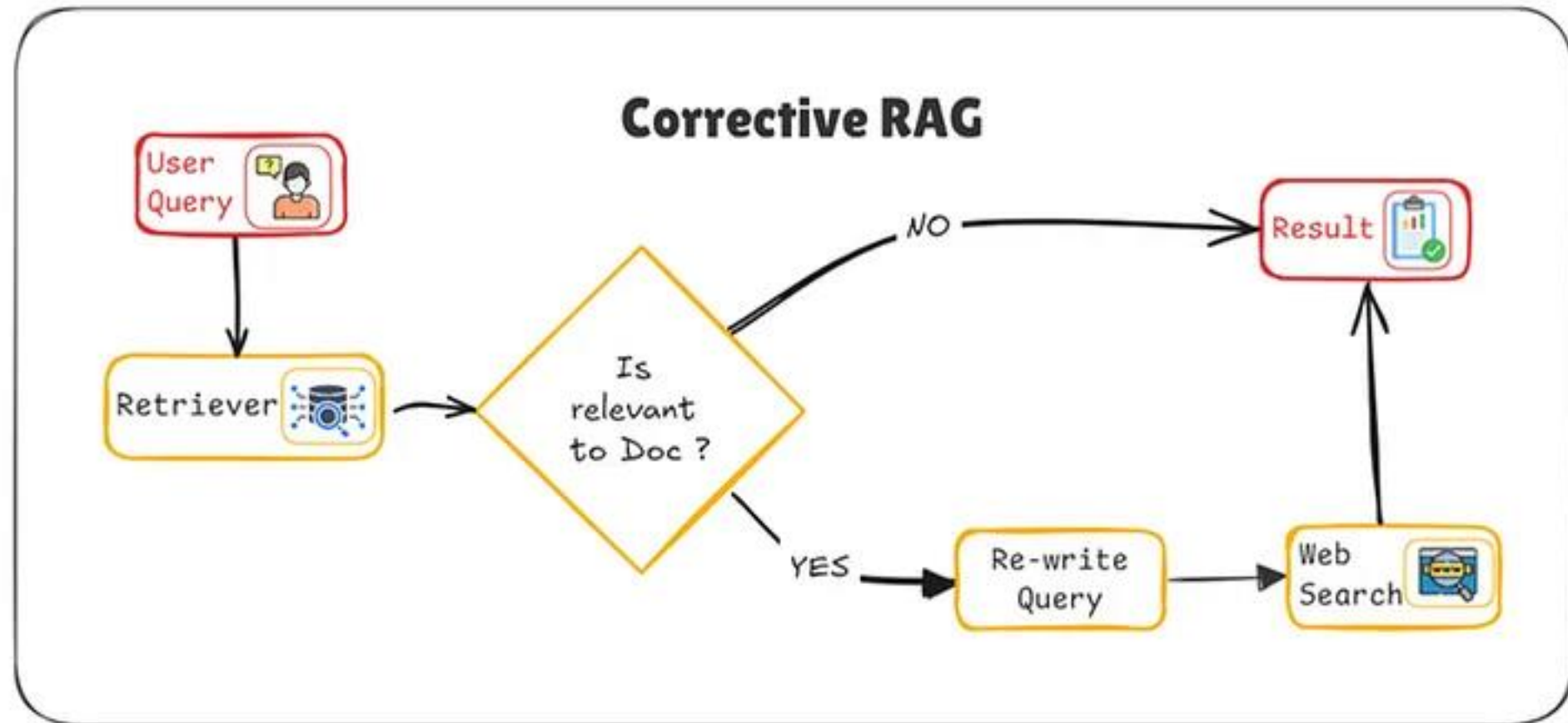
- Generate separate responses for each retrieved document, then rank or ensemble them.

Knowledge Sources

1. Static Knowledge Bases: Wikipedia, academic databases, etc.
2. Dynamic or Custom Sources: Internal documents, APIs, or search engines.

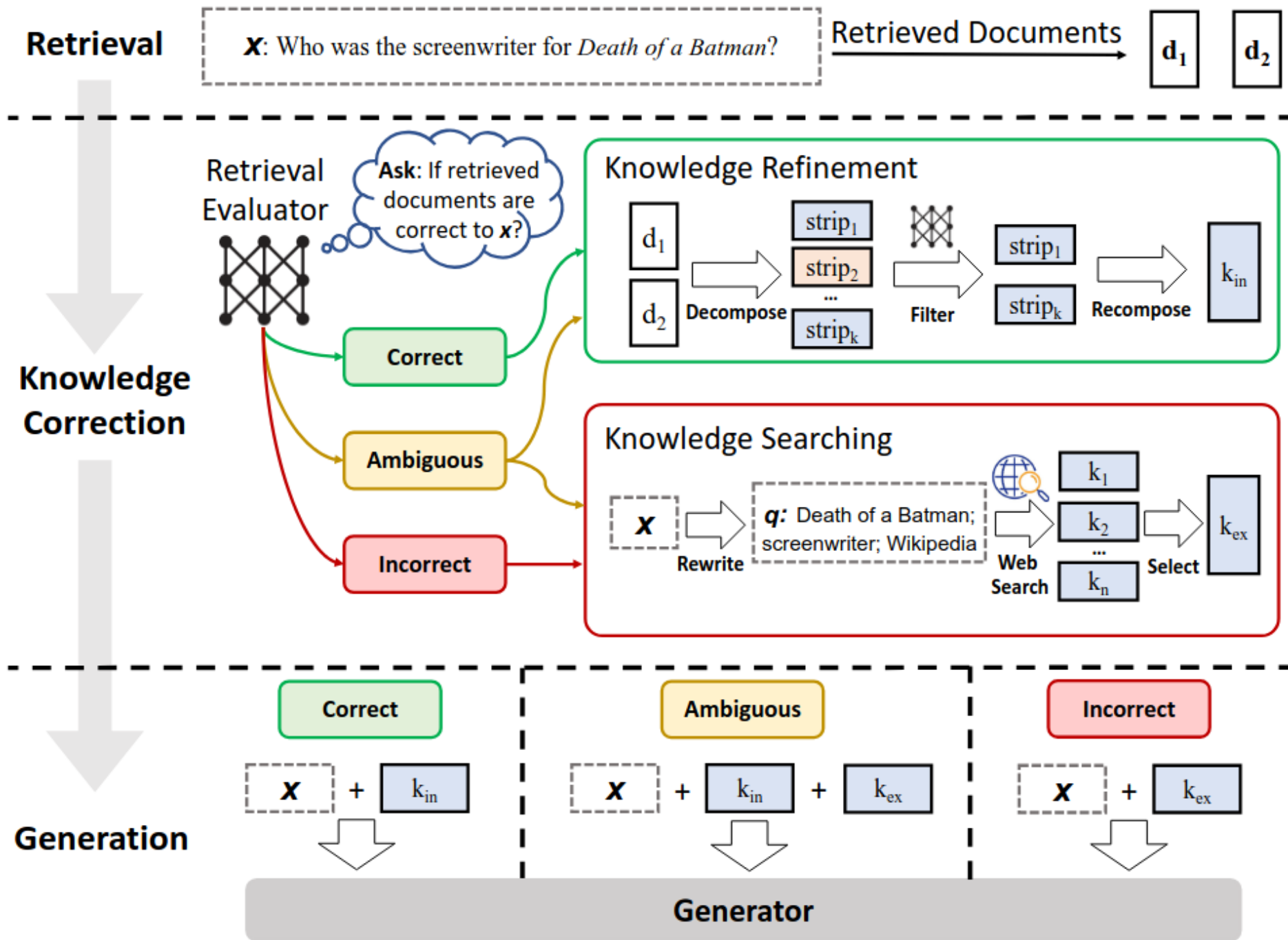


Source: <https://medium.com/the-hack-weekly-ai-tech-community/understanding-different-rag-techniques-0186ea5b9a13>

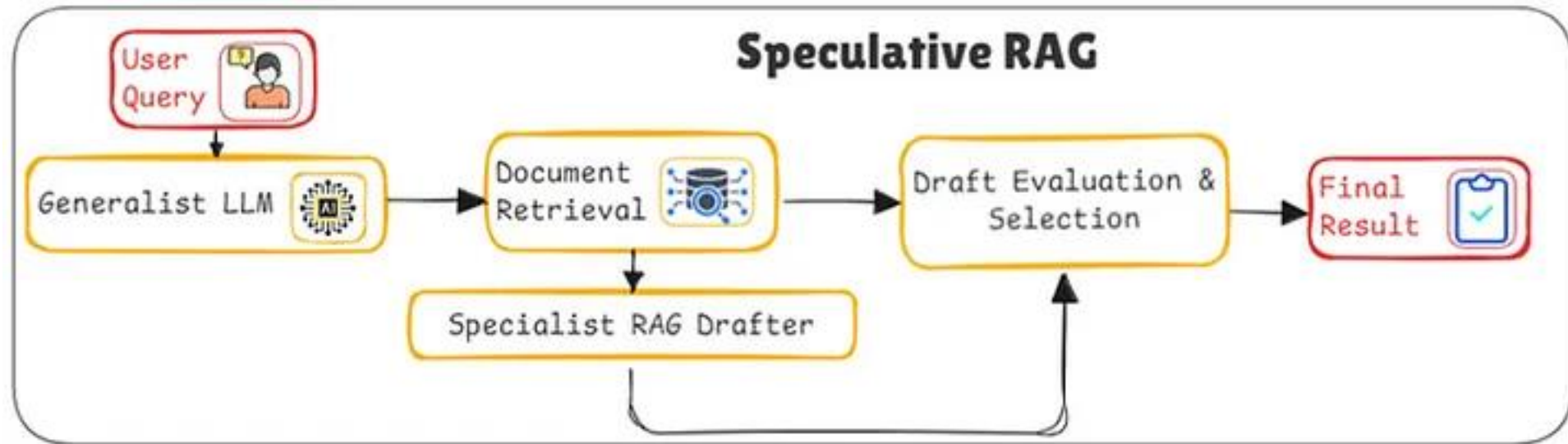


Source: Yan et al. 2024 „Corrective Retrieval Augmented Generation“

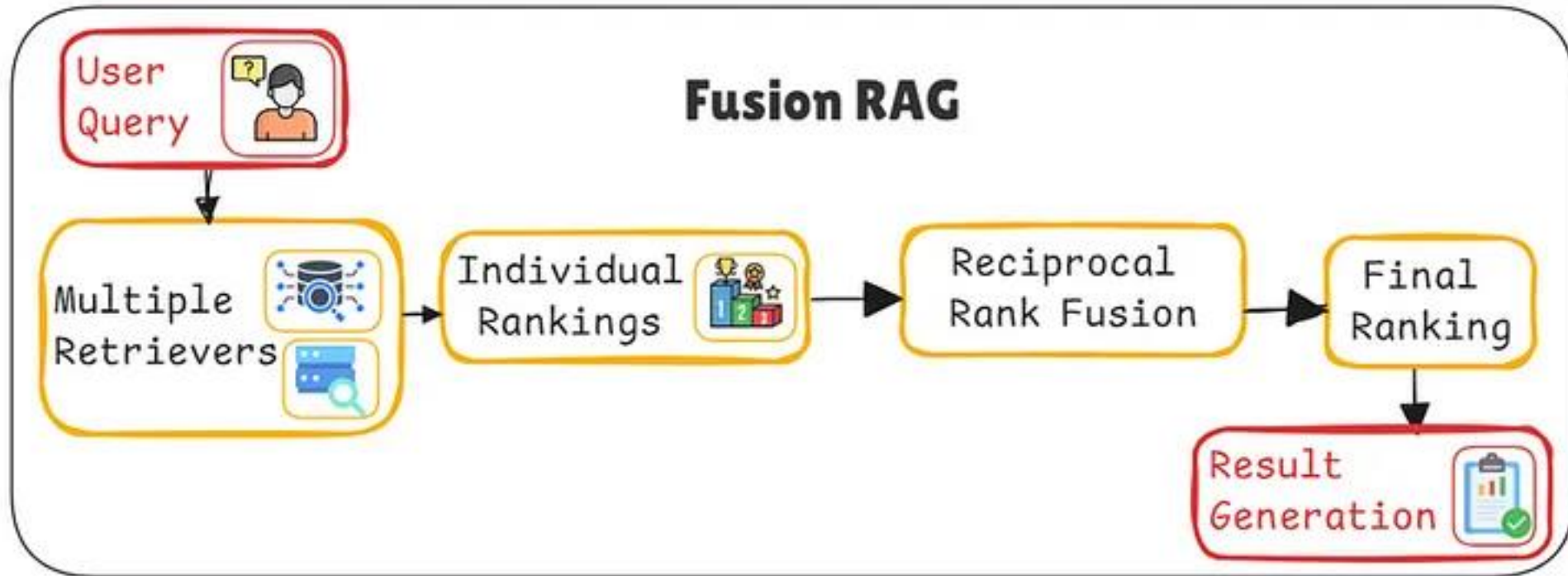
<https://medium.com/the-hack-weekly-ai-tech-community/understanding-different-rag-techniques-0186ea5b9a13>



Source: Yan et al. 2024 „Corrective Retrieval Augmented Generation“



Source: Wang et al 2024 „ Speculative RAG: Enhancing Retrieval Augmented Generation through Drafting“
<https://arxiv.org/abs/2407.08223>
<https://medium.com/the-hack-weekly-ai-tech-community/understanding-different-rag-techniques-0186ea5b9a13>



Source: Rackauckas 2024 „ RAG-Fusion: a New Take on Retrieval-Augmented Generation“

<https://doi.org/10.5121/ijnlc.2024.13103>

<https://medium.com/the-hack-weekly-ai-tech-community/understanding-different-rag-techniques-0186ea5b9a13>

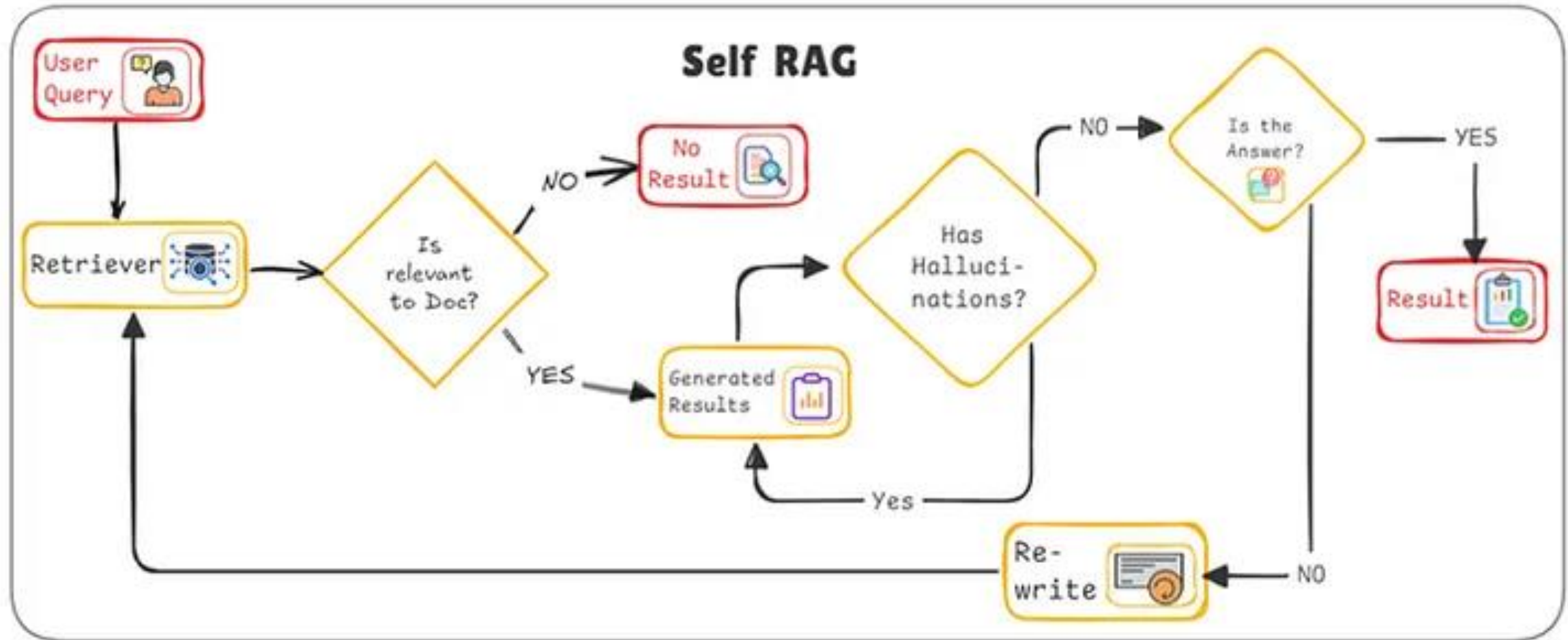
Query: IP rating of mounted IM72D128

Generated queries: ['1. What is the IP rating of the mounted IM72D128?', '2. IP rating explained for mounted IM72D128.', '3. Waterproofing capabilities of the IM72D128 with its IP rating.', '4. How does the IP rating of the IM72D128 affect its durability when mounted?']

Bot answer: The mounted IM72D128 has an IP57 rating, indicating a high level of protection against both dust and water. The first digit, 5, signifies its resistance to solid objects like dust or dirt. The second digit, 7, represents its resistance to water. This rating means that the IM72D128 has a sealed design, preventing water or dust from entering between the membrane and backplate. This robust design ensures no mechanical blockages or electric leakage issues typically seen in MEMS microphones. As a result, the IM72D128 is highly durable and can withstand exposure to water and dust without significant damage. Its sealed MEMS design allows it to function properly even in challenging environmental conditions, providing long-term durability and reliable performance.

Forum solution: Yes, you are right. IM72D128 has IP57 rating. The recommended Reflow soldering and board assembly of the microphone is given in section 7 (page: 14-15) of the datasheet. Please also refer to the application note: MEMS microphone mechanical & acoustical implementation.

Source: Rackauckas 2024 „ RAG-Fusion: a New Take on Retrieval-Augmented Generation“
<https://doi.org/10.5121/ijnlc.2024.13103>



Source: Asai et al 2024 „ Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection“

<https://openreview.net/forum?id=hSyW5go0v8>

<https://medium.com/the-hack-weekly-ai-tech-community/understanding-different-rag-techniques-0186ea5b9a13>

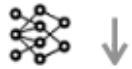
Prompt How did US states get their names?

Step 1: Retrieve on demand



Step 2: Generate segment in parallel

Prompt + 1



Relevant 11 of 50 state names
come from persons.

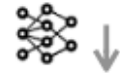
Supported

Prompt + 2



Irrelevant Texas is named
after a Native American tribe.

Prompt + 3



Relevant California's name has its
origins in a 16th-century novel
Las Sergas de Esplandián.

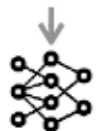
Partially

Step 3: Critique outputs and select best segment



US states got their names from a variety of sources. 11 of 50
states names are come from persons. 1 26 states are named
after Native Americans, including Utah. 4

Prompt: Write an essay of your best summer vacation



No Retrieval

My best summer vacation is when my family and I embarked on a road trip along ...

Source: Asai et al 2024 „ Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection“
<https://openreview.net/forum?id=hSyW5go0v8>

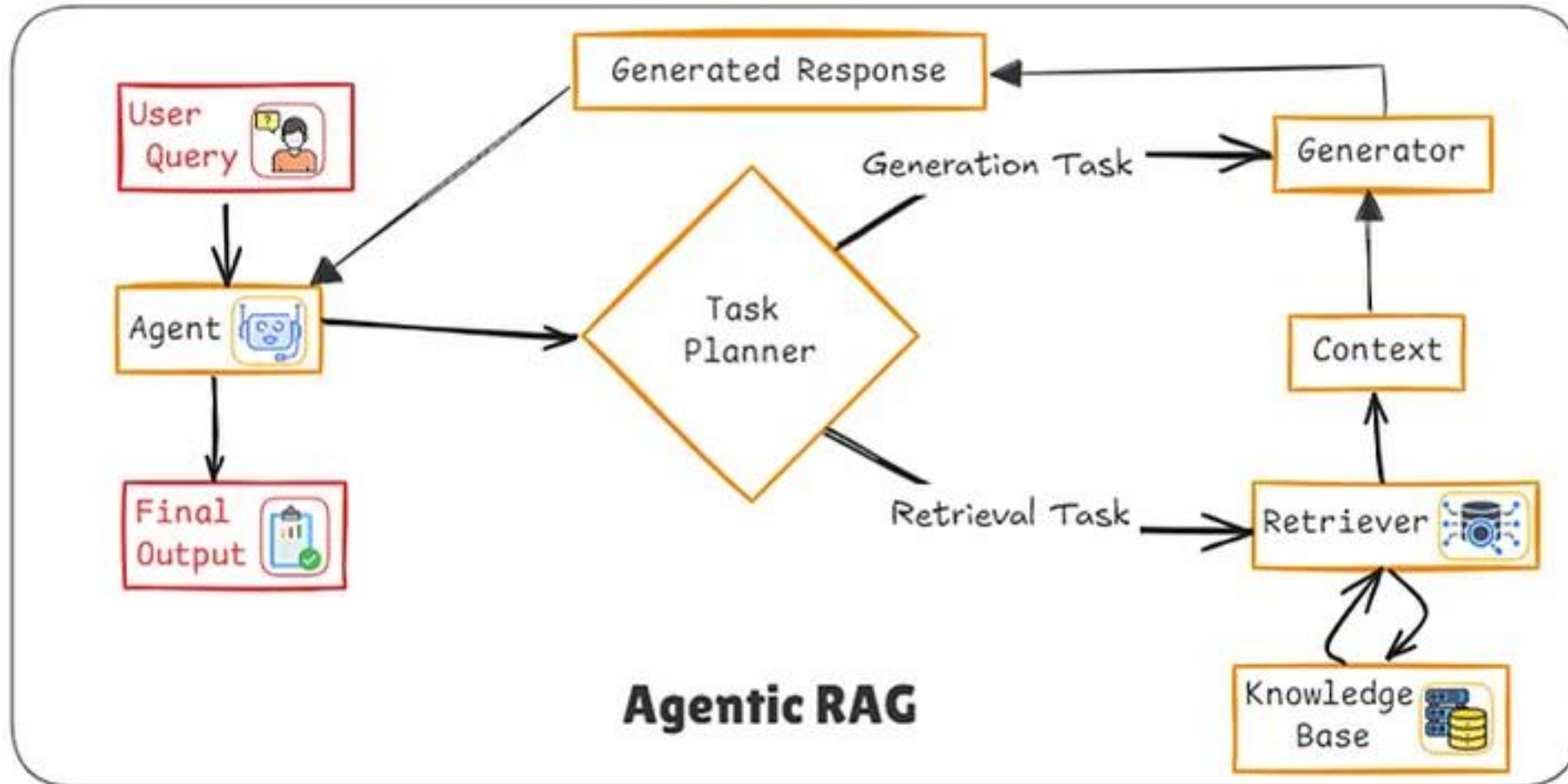
Type	Input	Output	Definitions
Retrieve	$x / x, y$	{yes, no, continue}	Decides when to retrieve with \mathcal{R}
ISREL	x, d	{ relevant , irrelevant}	d provides useful information to solve x .
ISSUP	x, d, y	{ fully supported , partially supported, no support}	All of the verification-worthy statement in y is supported by d .
ISUSE	x, y	{ 5 , 4, 3, 2, 1}	y is a useful response to x .

x : input

y : output

d : relevant passage

Source: Asai et al. 2024 „ Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection“
<https://openreview.net/forum?id=hSyW5go0v8>



Choosing the right RAG technique

- Complex tasks with changing requirements: Agentic RAG
- Maintaining long-context consistency and fluid interactions: Self-RAG
- Complex question answering, knowledge bases: Fusion RAG
- Real-time decision-making, balance of speed and accuracy: Speculative RAG
- High focus on accuracy (e.g. legal cases): Corrective RAG

Handling long retrieval contexts

Rank-Then-Generate

- Chunking: Break long documents or passages into smaller chunks
- Prioritized or Top-K Selection: Rank chunks or documents based on relevance, select only top-K
- Some relevant information may be discarded if it isn't in the top-K

Generate-Then-Rank

- Late Fusion: generate output for each chunk to generate intermediate outputs, then fuse the outputs to create the final response
- Increased computational overhead due to multiple generations

Handling long retrieval contexts

Summarize

- Summarize long retrieval contexts into shorter representations before feeding to generator

Hierarchical Processing

- Level 1: Retrieve and process chunks to extract key concepts or facts.
- Level 2: Use these extracted insights to form the final generative response.

Memory-Augmented Models

- Use memory networks or attention mechanisms to dynamically focus on the most relevant segments of the retrieved context.

Hybrid Methods

Conclusion

- Different solutions depending on the problem:
 - Fine-tuning (full, partial)
 - Prompt engineering
- RAG

Invitation to a lecture at the RPTU Kaiserslautern:

- Title: **AI/ML/Generative AI at Amazon: What we learned**
- **AWS pioneer Constantin Gonzalez Schmitz** on how Amazon applies (Gen) AI/ML to innovate, optimize, and scale globally!
- The event is **free of charge**. Please register at: <https://www.eventbrite.com/e/aimlgenerative-ai-at-amazon-what-we-learned-tickets-1139713707379?aff=oddtcreator>
- Date and Time: **Tuesday, January 14 2025, 11:30 – 13:15**
- Where: **Audimax 42-115, RPTU Kaiserslautern, Erwin-Schrödinger-Straße**

- Next Lecture: Neural Topic Models

References

- Bowman, Sam. “Pre-Training and Transfer Learning” (lecture, Machine Learning for Language Understanding, NYU, 14.02.2021)
- [\[GPT\]](#) Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).
- [\[GLUE\]](#) Wang, Alex, et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding." (2018).
- [\[SuperGLUE\]](#) Wang, Alex, et al. "Superglue: A stickier benchmark for general-purpose language understanding systems." (2019).

References

- [\[ADAPTER\]](#) Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M. and Gelly, S., 2019, May. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.