Title: Modular Memory-augmented Models (MoMeM)

Author: Mwaura Vincent Muchai

Code (Proof-of-Principle):
https://github.com/Vinmwaura/Modular_Memory-augmented_Models_MoMeM

1. Introduction

Transformer-based models have emerged as the most transformative and influential Artificial Neural Network (ANN) architecture to date, redefining state-of-the-art performance across various fields such as Natural Language Processing (NLP), Computer Vision (CV), and audio processing. This success can be attributed to its novel attention mechanism, which enables models to capture long-range dependencies by dynamically weighting interactions between tokens (smallest unit of input), thereby producing context-sensitive representations that scale efficiently with parallel computation.

However, despite their impressive performance, they face numerous limitations:

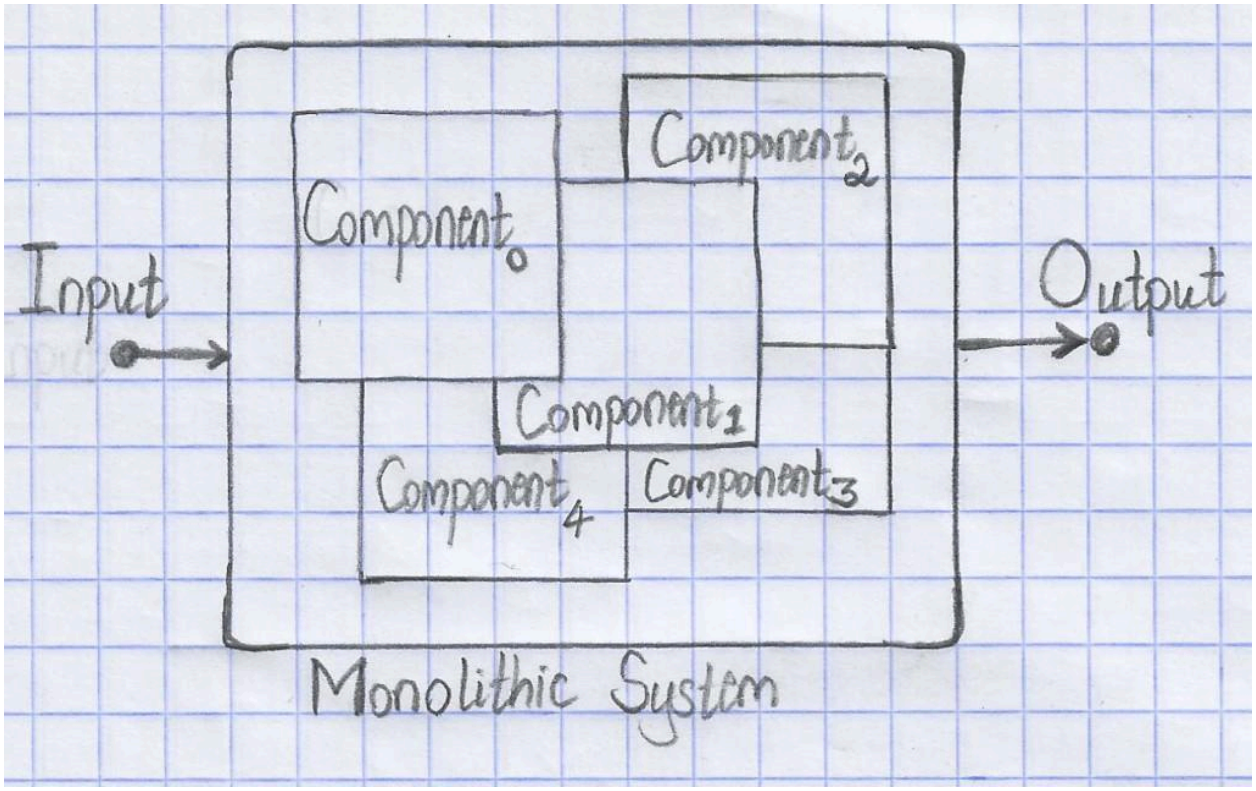
- The models require scaling up to billions (even trillions) of parameters, trained on a vast dataset, and utilize substantial compute (flops) for optimal results. This makes them **computationally and financially infeasible** for a majority of people and organizations to train and deploy locally with ease. Additionally, training, running (inference), and maintaining them at scale is a costly and unsustainable process.
- They are known to suffer from a phenomenon called **hallucinations**. This is where fully trained models generate nonsensical or factually incorrect content frequently enough to be of concern, making them unreliable for most tasks. Hallucinations can be categorized into three categories:
    - **Input-conflicting hallucination:** Generated content deviates from the source input provided by the user. The contradiction between the model's response and task instructions typically reflects a misunderstanding of user intents.
    - **Context-conflicting hallucination**: Generated content conflicts with previously generated information from the model itself.This type of hallucination arises when the model loses track of the context or fails to maintain consistency throughout a conversation.
    - **Fact-conflicting hallucination**: Generated content is factually incorrect to established world knowledge. This is mostly brought about by the nature of the model to produce coherent and plausible text with no mechanism to ground itself on truthfulness. In addition, this effect could be brought about by low-quality training data and poor generalization of the model during training.
- The models are prone to **catastrophic forgetting**. This is where already trained models struggle to learn new tasks and update their knowledge when training on new datasets. This happens due to the model implicitly encoding information in their parameters (parametric memory) which when updated could introduce errors or erase encoded information in their parameters. This limits the model's ability to continuously learn without forgetting any previously learned information.

The aim of this project is to address the limitations outlined above by employing established software design principles into the development of Machine Learning (ML) models while unifying key concepts from Generative AI, Information Retrieval (IR) systems, and Knowledge-Based systems (KBS). It utilizes the Unix philosophy, which emphasises building simple, compact, clear, modular, and extensible "code". A simplified summary of this philosophy is:

- Write programs that do one thing and do it well (**Single Responsibility Principle**).
- Write programs to work together (**Modularity** and **Composability**).
- Write programs to handle text streams, because that is a universal interface.

Intuitively this means thinking of the current implementation of a Transformer-based model as being a **monolithic** system, whereby it can be viewed as a singular, unified system where functionally distinguishable aspects such as I/O operations, logic, etc. are all interwoven together rather than being architecturally separate “components”. This design decision makes them simpler and easier to implement, train, and deploy with the tradeoff being challenges in scaling and maintaining them i.e. fine-tuning them to fix any issues post training. In software engineering terms: Every “component” in the system is highly interdependent (**tightly coupled**), meaning changing one thing can affect another in unpredictable and undesirable ways.

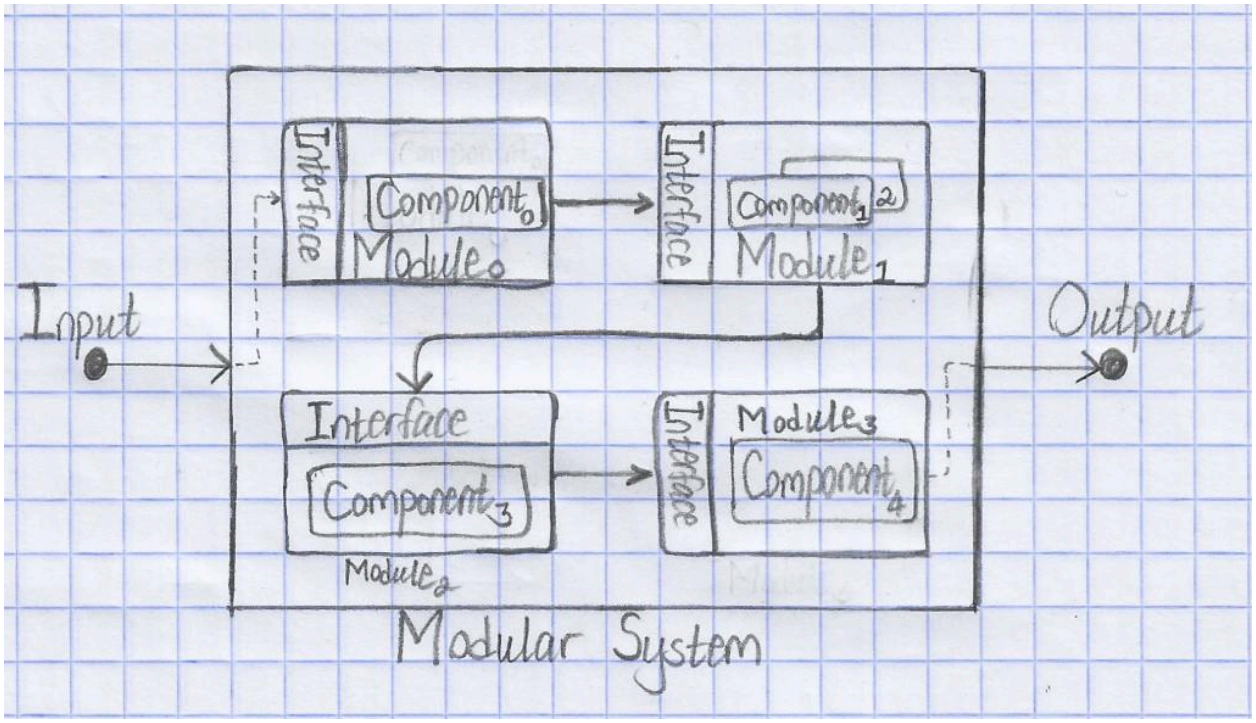
A rough high-level conceptual sketch of a simple monolithic system is shown below. It illustrates “components” as being tightly coupled and lacking clear, well-defined boundaries.



Developing the system to be **modular** can alleviate many of the aforementioned issues by decomposing the unified system into smaller, manageable, and independent “components” called **modules**. These modules are independently created, modified, or replaced without affecting the rest of the system, which is only possible through **loose coupling**, where interdependence is kept to a minimum. Consequently, all interactions between modules occur exclusively through well-defined, standardized interfaces.

Applying the modular design principle to Transformer-based models would involve decoupling interdependent “components” (implicit functionalities i.e. “data handling” and “logic processing") from a single model into multiple modules, that would each focus on smaller, well-defined tasks (**separation of concerns** principle), with a standardized interface in each module that allows for explicit, well-defined interactions between them.

A rough high-level conceptual sketch of a simple modular system is shown below. It illustrates multiple modules with distinct “components”, well-defined boundaries, and exposed explicit standardized interfaces for interactions.



2. Background

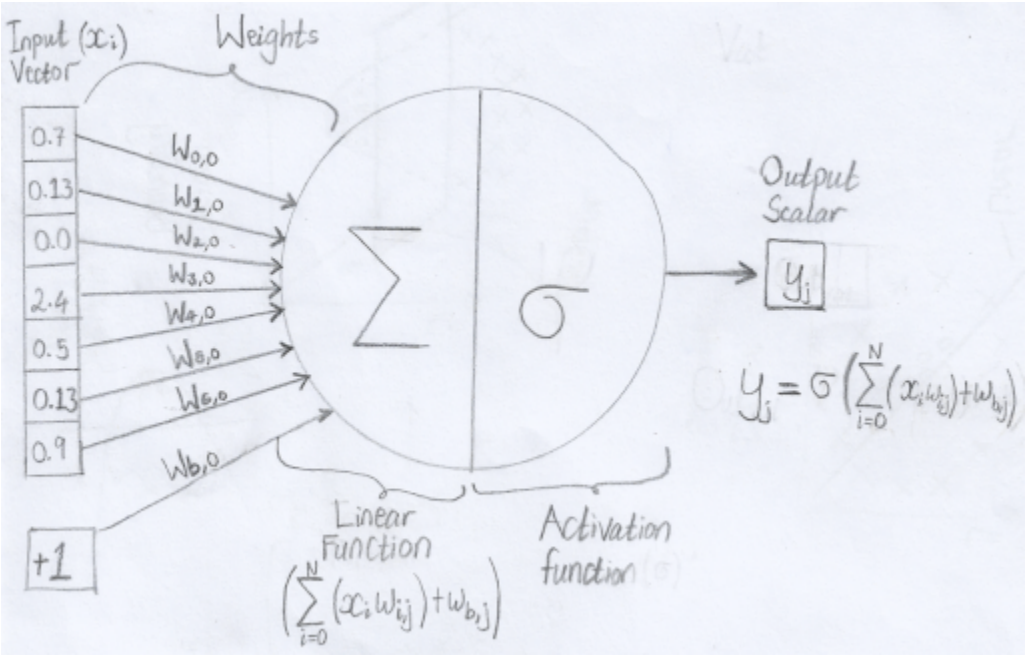
2.1. Feedforward Networks

In the field of Machine Learning (ML), there exists a type of algorithm known as either **Neural Networks (NNs)** or **Artificial Neural Networks (ANNs)** which are loosely inspired by biological neurons in the brain. They are composed of simple building blocks called **neurons** that are a composition of linear (weighted sum inputs and bias) and non-linear (activation or squashing) functions. Both of which can be expressed mathematically as:

$$y_j = \sigma(\sum_{i}^{N}(x_i w_{i,j}) + w_{b,j})$$

where x_i is the input vector, y_j is the neuron's output vector, $w_{i,j}$ and $w_{b,j}$ are the weights of the network and bias respectively, and σ is a non-linear function e.g ReLU, Sigmoid, Tanh, etc.

Visual depiction of a **neuron**:



Multiple neurons can be grouped together to form a **layer**, and multiple layers can be stacked sequentially to construct a **network**. The depth of a network is determined by the number of layers the network contains, while the width of a layer refers to the number of neurons within that layer. The typical layer structure of said network consists of:

i. **Input layer**: This layer serves as the entry point for the network and performs no computation on the input.

ii. **Hidden layer(s)**: This comprises one or multiple layers between the input and output layers which sequentially transforms the input data by passing the output of the previous layer as input for the succeeding layers.
iii. **Output layer**: This is the final layer of the network, whose output is used to perform a specific task such as classification or regression.

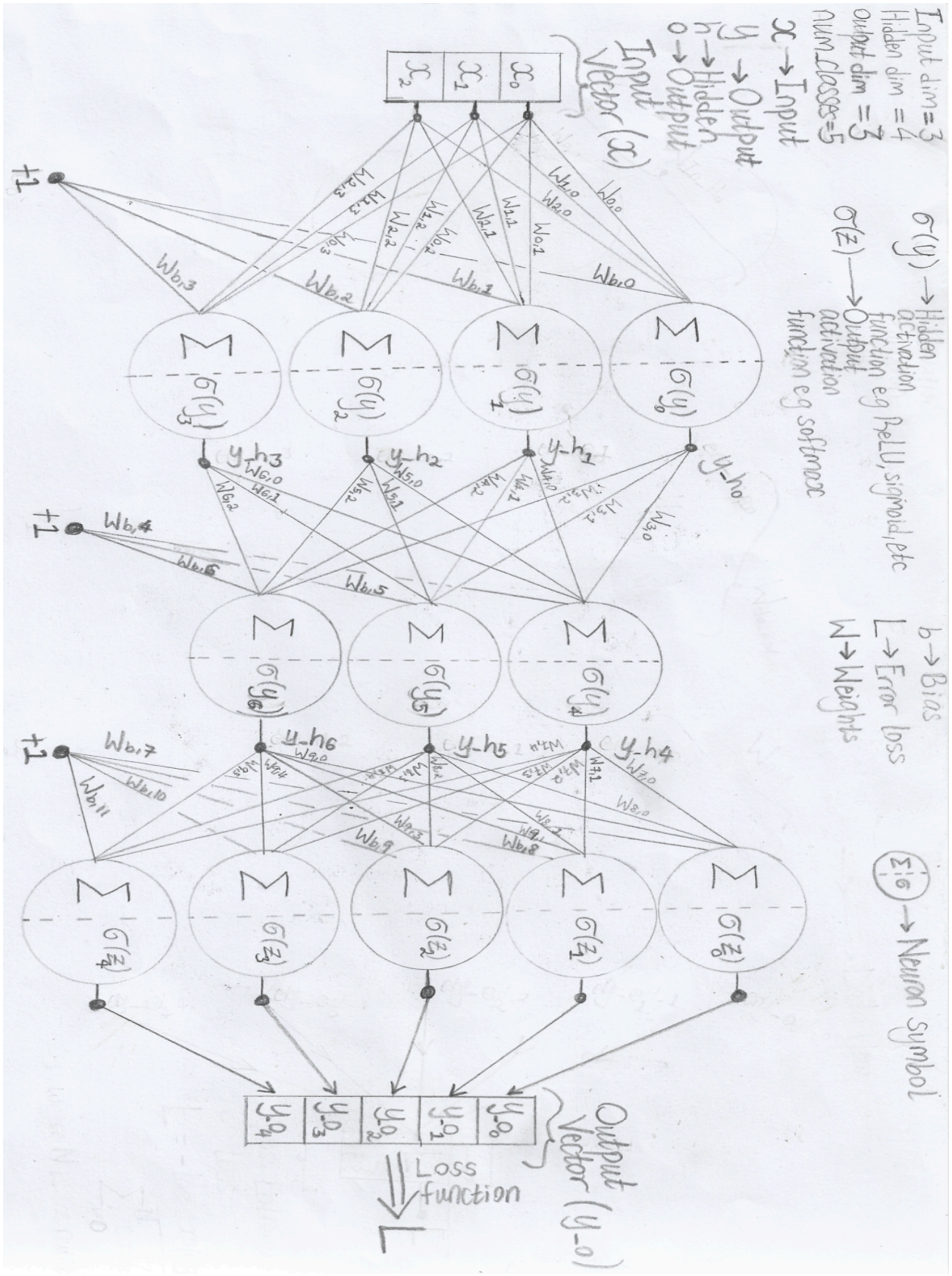
A network can be represented mathematically as a composition of functions, where the output of each layer is represented as a function f and is used as the input to the succeeding layer. For example, given three functions: f_1, f_2, and f_3 connected sequentially, the composite function can be written as:

$$f(x) = f_3(f_2(f_1(x)))$$

where x denotes the input vector, f_1, f_2, and f_3 represents successive transformations applied by the network, and $f(x)$ is the final output. This nested structure reflects how (deep) networks transform input data through multiple stages to produce a desired output.

These networks are commonly referred to as **Feedforward Networks (FFNs)** because information, usually in the form of vectors, are passed (fed) through the network from the input layer, through the hidden layer(s), and finally through the output layer. In standard FFNs architectures there are usually no feedback or recurrent connections, meaning that the output of any layer is not routed back as input to any preceding layer. The only exception to this rule being Recurrent Neural Networks (RNNs), which are specifically designed to handle sequential and temporal inputs in a recurrent manner.

Visual depiction of a Feedforward Network (FFN):



All Feedforward Networks (FFNs) operate in two fundamental phases:

1. **Forward propagation**: This involves information being forwarded sequentially through the network's layers. Each layer transforms their respective input then passes them to the succeeding layer i.e. the output of a layer is used as the input of the following layer.
2. **Backward propagation**: This is the most critical phase of the network, as it is responsible for optimizing (training) the network to learn a given task. This phase can be decomposed into several key steps:
    i. The error between the network's predicted output (output of output layer) and the true target values (ground truth) is computed using a loss function (L) e.g. Mean Square Error (MSE), etc. This loss provides a scalar measure of how well or poorly the network performs on a given task.
    ii. Next the gradient descent algorithm is applied to minimize this loss. The gradient represents the direction of the steepest increase of a function; by calculating this vector and moving in the opposite direction, the algorithm iteratively reduces the error until it reaches a local or global minimum. To compute the gradients, the partial derivatives of the loss function with respect to each weight is computed (all other weights being held constant) using the chain rule. This allows propagating the error signals backward through the network layer for each weight, $w_{j,k}$, as shown below:

$$\frac{\partial L}{\partial w_{j,k}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w_{j,k}}$$

where y denotes the neurons output, z is the linear function's value, and j and k denote the preceding and succeeding layers position in the overall network.

    iii. Using the computed gradients, all weights in the network are updated simultaneously according to the following update rule:

$$w_{j,k}^{(new)} = w_{j,k}^{(old)} - \eta \times \frac{\partial L}{\partial w_{j,k}}$$

where η is the learning rate.

    iv. If a predefined stopping condition, such as maximum number of iterations, has not been met, then the process repeats for the next batch of input data.

The performance of a fully trained Feedforward Network (FFN) is primarily influenced by two fundamental factors:

i. **Representational power of the network (expressivity)**: This refers to the capacity of the network to represent a desired function, f^*. Because a neural network is constructed as a composition of functions f, it may not be capable of representing the desired function, f^* exactly. This could be due to factors such as:
    a. Inadequate network depth.

b. Inadequate network width.
c. Poor choice of activation functions.

ii. **Learning algorithm**: This governs how effectively the network's parameters can be optimized during training with the assumption that there exists a set of weight configurations: w^*, for the network that yields near-optimal performance on a given task. Through iterative optimization procedures, such as gradient descent, the learning algorithm seeks to discover these weights: w^*, by minimizing the loss function: L. However, certain factors can hinder achieving these optimal weights, w^*:
   a. Limited and/or noisy dataset
   b. Poor weight initializations leading to saddle points, local minima, etc.
   c. Inadequate hyperparameters e.g. learning rates, batch size, loss function, etc.
   d. Insufficient training time

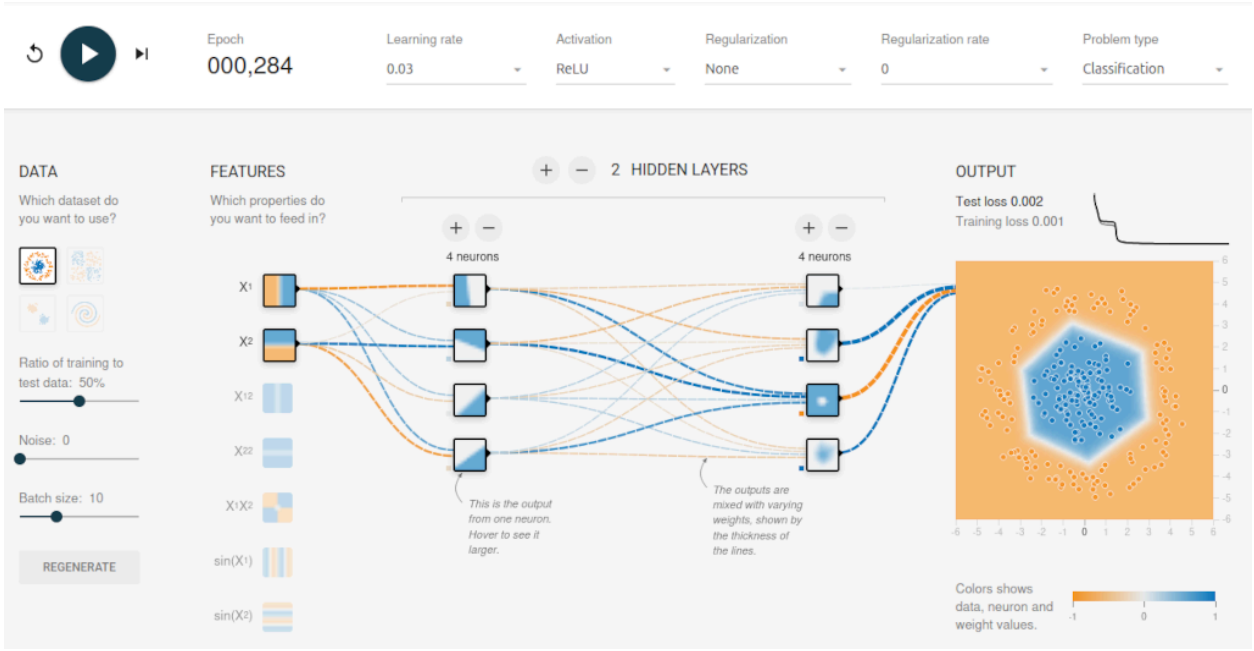
The success of neural networks over many traditional machine learning algorithms such as decision trees, support vector machines, etc., can be partly attributed to their capacity as **universal function approximators**. This property is formalized by the **Universal Approximation Theorem (UAT)**, which states that a Feedforward Network (FFN) with at least one hidden layer and a nonlinear activation function can approximate virtually any continuous function of interest to any desired degree of accuracy. This theorem provides a strong mathematical justification for employing either deeper or wider neural network architectures in real-world applications, where complex, nonlinear relationships must be modeled.

Although both deep and wide neural network architectures are theoretically capable of achieving comparable performance, deeper networks are generally favoured in both research and practical applications. This preference has contributed to the prominence of deep learning as a distinct and influential subfield of machine learning. One key reason for this is that deeper networks exploit the hierarchical composition of simpler functions: f, allowing successive layers to build increasingly abstract and complex representations from the input data.

From a theoretical perspective, the representational power (expressivity) of neural networks can increase exponentially with depth. In contrast, achieving equivalent representational capacity with shallow but wide networks often requires an exponential increase in the number of neurons, leading to significantly higher computational and memory costs. As a result, deeper architectures tend to offer more parameter-efficient and scalable solutions.

Many modern neural network architectures have emerged from this principle, including the Transformer-based architecture which is predominantly used in the field of Natural Language Processing (NLP) and forms the foundation of Large Language Models (LLMs). These models are particularly notable for their strong empirical performance under scaling, where increases in model size, training data, and compute (flops) has consistently led to improvements in their capabilities.

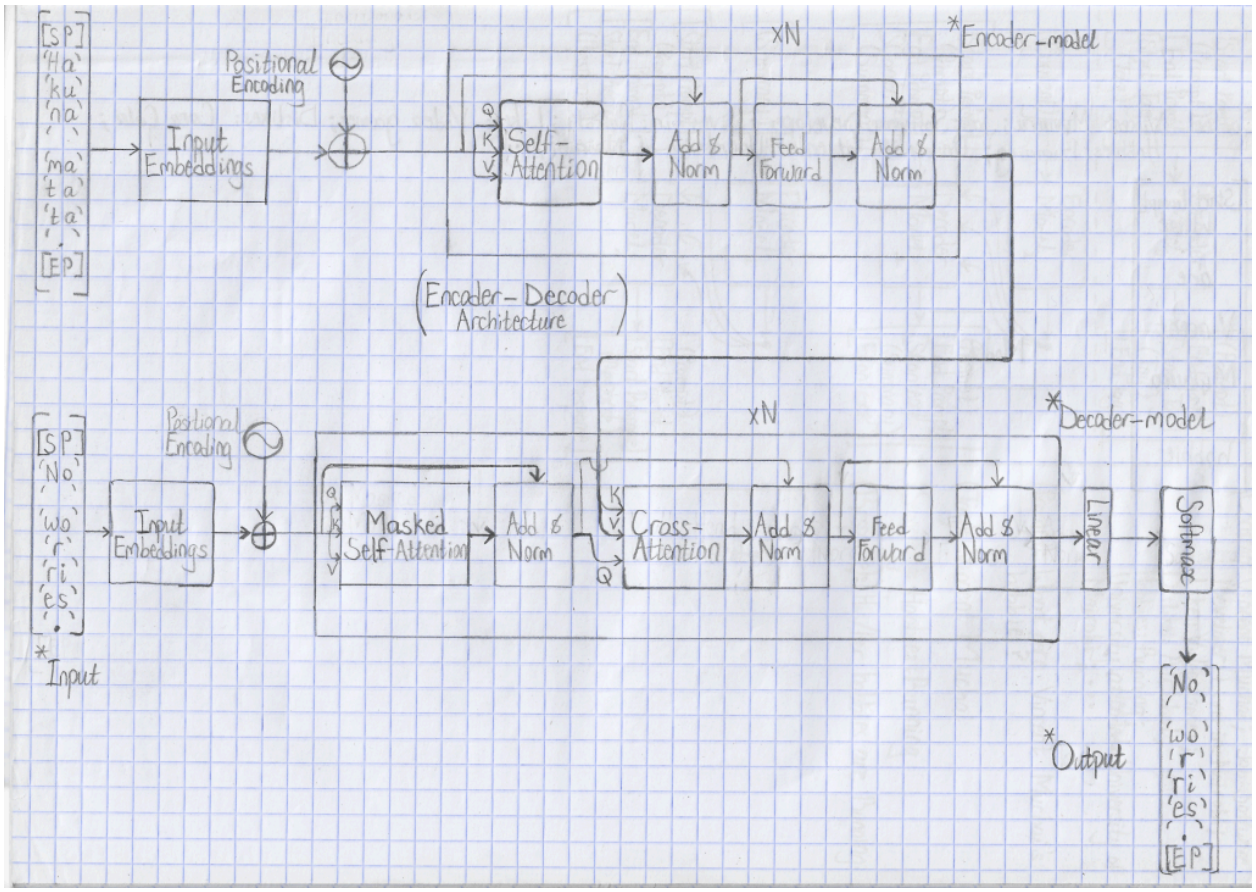
Below is a simple visualization of a small neural network (2-dimensional input vector, 2 hidden layers each with 4 neurons, and an output layer with 1 neuron) performing a binary classification task. It shows a visual representation of function composition in a neural network and how it can separate two classes in the output layer. The orange colour depicts negative values while blue depicts positive values. In the hidden layers, the lines are coloured by the weights of the connections between neurons. The intensity of the colour in the output layer shows how confident the prediction is (source: https://playground.tensorflow.org):



2.2. Transformer-based models

Transformer-based models are neural network architectures that rely exclusively on the attention mechanism, dispensing with recurrence (used by RNNs) and convolutions (used by CNNs) operations entirely. The models generalize well to Natural Language Processing (NLP), Computer Vision (CV), and audio processing problems and has become the de facto model architecture for many tasks due to its unrivalled performance, ability for parallelization and efficiency during training.

Overview of the original Transformer model architecture and a visual depiction of how it works:

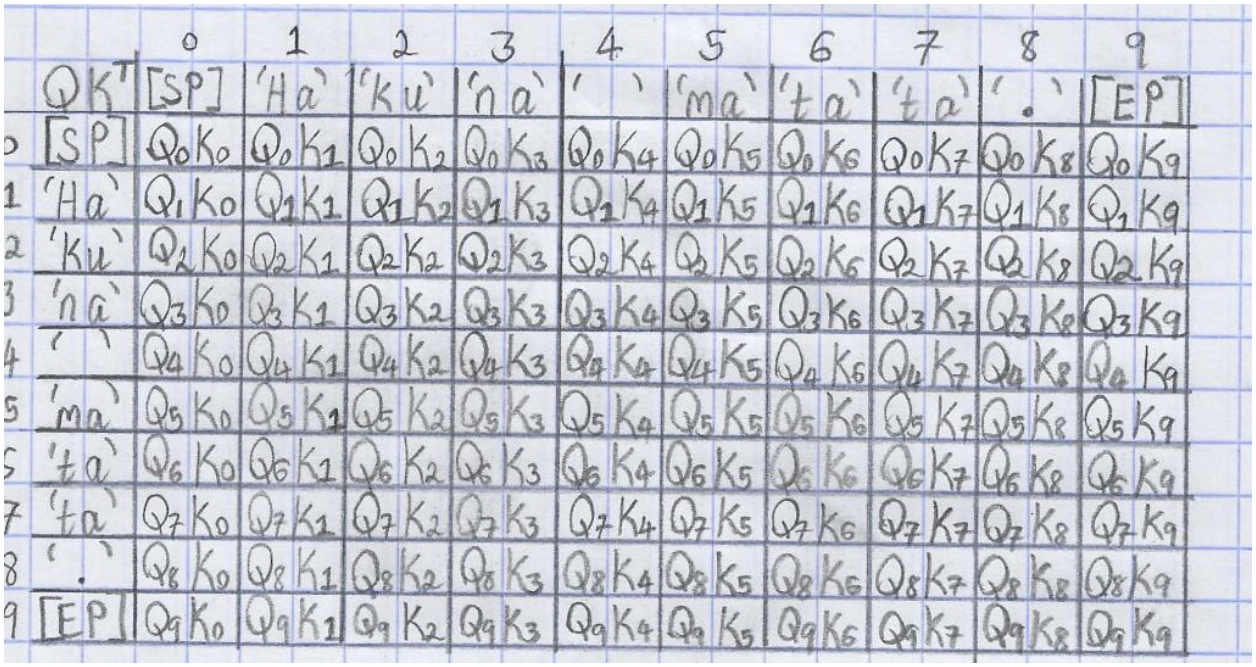


The attention mechanism works by assigning weights to tokens passed as input to the model. It does so by mapping the vector representations of said tokens into query (Q), key (K), and value (V) vectors, which are all combined using the formula:

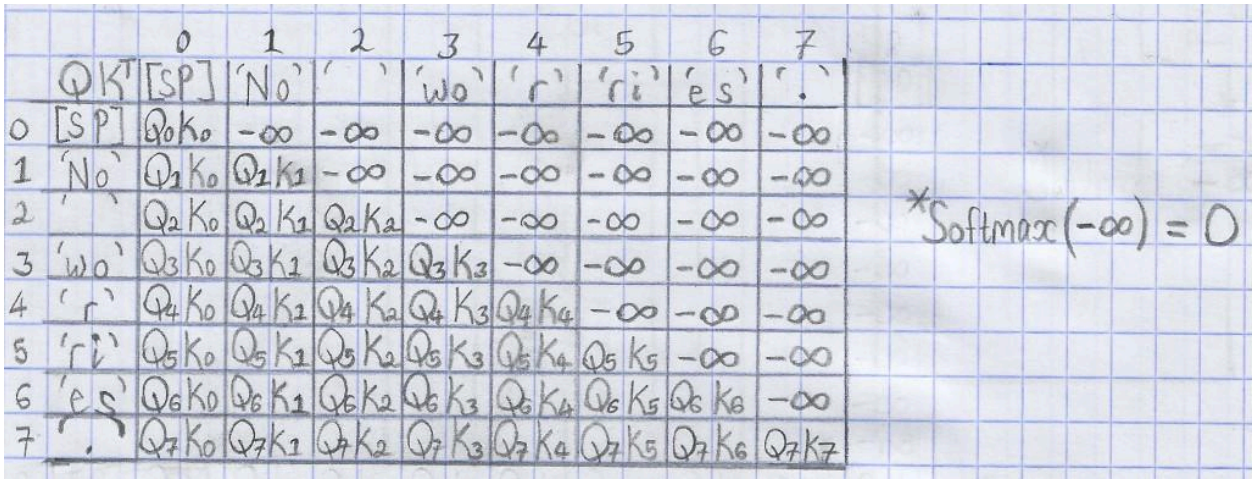
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{Dim_k}})V$$

There are three types of attention mechanisms:

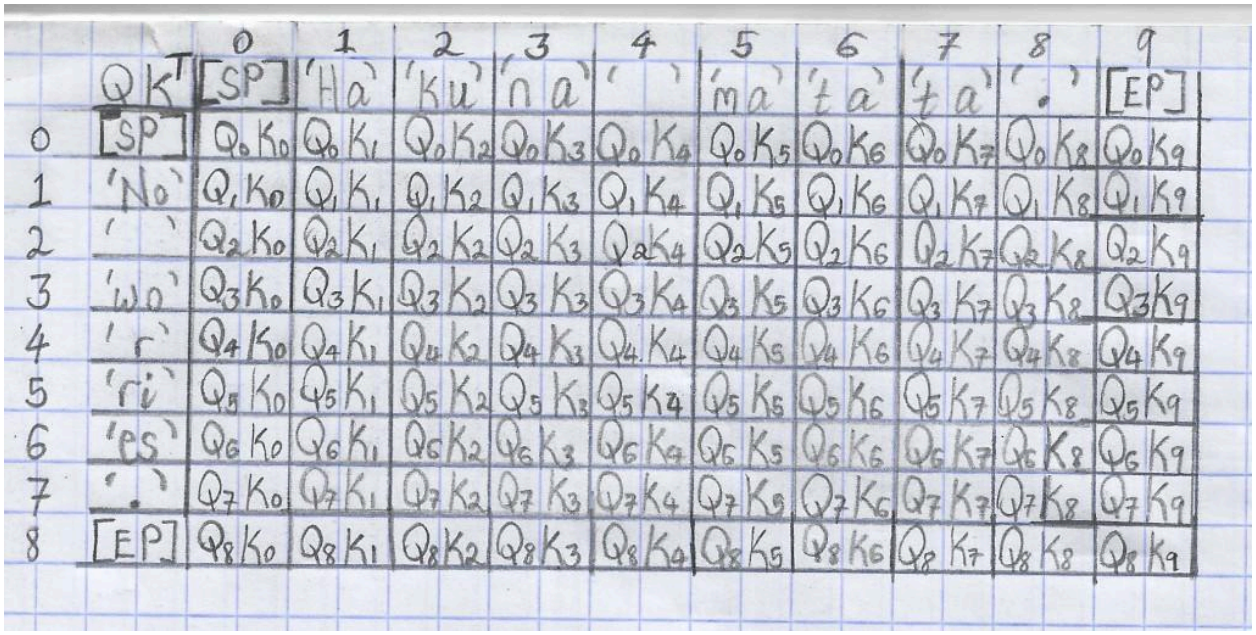
- **Self-Attention**: Here each token can attend to all other tokens within the same input sequence, enabling contextualized representations. The table below illustrates how the QK^T is computed for Self-Attention:



- **Masked Self-Attention**: Here attention is restricted by using $-\infty$ value on specific tokens (upper triangle, shifted once to the right) to ensure that each token can only attend to its own and other preceding tokens in a sequence. This prevents leftward tokens access to future information, tokens to their right. The constraint is typically enforced by applying a masking operation to the QK^T matrix to prevent leftward information flow during computation. The table below illustrates how the QK^T is computed for Masked Self-Attention:



- **Cross-Attention**: Here the queries (Q) are derived from the Decoder's input sequences, while the keys (K) and values (V) originate from the Encoder's output sequence. This allows the Decoder to condition its representations on the Encoder's encoded context, enabling it to generate outputs that are informed by the source input. The table below illustrates how the QK^T is computed for Cross-Attention:



QK^T	0 [SP]	1 'Ha'	2 'ku'	3 'na'	4 ' '	5 'ma'	6 'ta'	7 'ta'	8 '.'	9 [EP]
0 [SP]	Q_0K_0	Q_0K_1	Q_0K_2	Q_0K_3	Q_0K_4	Q_0K_5	Q_0K_6	Q_0K_7	Q_0K_8	Q_0K_9
1 'No'	Q_1K_0	Q_1K_1	Q_1K_2	Q_1K_3	Q_1K_4	Q_1K_5	Q_1K_6	Q_1K_7	Q_1K_8	Q_1K_9
2 ' '	Q_2K_0	Q_2K_1	Q_2K_2	Q_2K_3	Q_2K_4	Q_2K_5	Q_2K_6	Q_2K_7	Q_2K_8	Q_2K_9
3 'wo'	Q_3K_0	Q_3K_1	Q_3K_2	Q_3K_3	Q_3K_4	Q_3K_5	Q_3K_6	Q_3K_7	Q_3K_8	Q_3K_9
4 'r'	Q_4K_0	Q_4K_1	Q_4K_2	Q_4K_3	Q_4K_4	Q_4K_5	Q_4K_6	Q_4K_7	Q_4K_8	Q_4K_9
5 'ri'	Q_5K_0	Q_5K_1	Q_5K_2	Q_5K_3	Q_5K_4	Q_5K_5	Q_5K_6	Q_5K_7	Q_5K_8	Q_5K_9
6 'es'	Q_6K_0	Q_6K_1	Q_6K_2	Q_6K_3	Q_6K_4	Q_6K_5	Q_6K_6	Q_6K_7	Q_6K_8	Q_6K_9
7 '.'	Q_7K_0	Q_7K_1	Q_7K_2	Q_7K_3	Q_7K_4	Q_7K_5	Q_7K_6	Q_7K_7	Q_7K_8	Q_7K_9
8 [EP]	Q_8K_0	Q_8K_1	Q_8K_2	Q_8K_3	Q_8K_4	Q_8K_5	Q_8K_6	Q_8K_7	Q_8K_8	Q_8K_9

Additionally the Transformer-based model architecture can be implemented in various configurations:

- **Encoder-only model**: Only the Encoder module of the model is used, typically with the Self-Attention mechanism.
- **Decoder-only model**: Only the Decoder module of the model is used, typically with the Masked Self-Attention mechanism.
- **Encoder-Decoder model**: Both the Encoder and Decoder models are used, which jointly utilizes the Masked Self-Attention, and Cross-Attention mechanism. They are usually known as a sequence-to-sequence model.

To train a Transformer-based model, the following stages must be carried out:

i. **Data Preparation**
   a. **Data collection**: Data can be assembled from a diverse range of sources:
      - **Literary and Creative writings**: novels, short stories, poems, plays, fables, myths, and folklore.
      - **Educational and Academic materials**: textbooks, research papers, lecture notes, dissertations, scientific articles, and encyclopedias.
      - **Technical and Professional Content**: software documentation, source code, API references, patents, engineering manuals, medical guidelines, and legal contracts.

- **News and Journalism**: newspaper articles, magazines, editorials, investigative reports, and press releases.
- **Business and Workplace Documents**: emails, memoranda, reports, business plans, meeting transcripts, and financial statements.
- **Government and Legal Records**: laws, regulations, court rulings, constitutions, treaties, policy briefs, and legislative transcripts.
- **Everyday and Informal Communication**: social media posts, blogs, forums, product reviews, Q&A threads, and chat logs.
- **Cultural and Historical Texts**: religious scriptures, speeches, historical archives, and autobiographies.
- **Instructional and How-to Guides**: tutorials, guides, cookbooks, repair manuals, and user guides.

b. **Data cleaning**: The collected data are then filtered to remove noise, handle outliers, correct imbalances, and other preprocessing operations. After which deduplication is performed which entails removing duplicates found in the collected data. This is done to reduce redundancy and improve data quality.

ii. **Tokenization:** This stage converts the raw text from the collected data into smaller, machine-readable units called tokens. The choice of granularity of these tokens can impact model efficiency and vocabulary sizes. Common approaches include:

a. **Character-based tokenization**: Here text is broken down into individual characters such as letters, numbers, or other symbols. The major advantage of this approach is there are no or very few unknown or Out-Of-Vocabulary (OOV) words during training and it is simple to implement. The major drawback with this approach is that it produces very long input sequences, which can be computationally expensive and infeasible to process due to the high memory demands of the attention mechanism and limited memory capacity of hardware.

b. **Word-based tokenization**: Here text is broken down into words, typically using spaces and punctuation marks as delimiters. The main advantage of this approach is that it is usually simpler to implement and produces shorter input sequences. However the main drawback is that this approach requires extremely large vocabularies to capture all possible words from the text, which can lead to frequent OOV issues.

c. **Subword-based tokenization**: Here text is broken down into subwords which are usually the most frequent pairings of adjacent characters. This approach strikes a balance between Character-based tokenizations (very long input sequences but little to no OOV issues) and Word-based tokenizations (very large vocabulary sizes with frequent OOV issues but shorter input sequences). Common algorithms include:

- **BytePairEncoding** (**BPE**): This is a frequency-based algorithm that iteratively merges the most common adjacent character pairs in a text until a target vocabulary size is reached. At inference, words are decomposed into subword units if unseen, while common words remain whole.

    - **WordPieceEncoding**: This is a probabilistic subword tokenization method that iteratively merges common character sequences, balancing vocabulary size with coverage. At inference, text is segmented into the longest matching subwords, ensuring out-of-vocabulary (OOV) terms can still be represented.
    - **SentencePieceEncoding**: This is a data-driven tokenizer that treats text as a raw sequence of characters, without assuming whitespace as word boundaries. It learns subword vocabularies using algorithms such as BPE or Unigram Language Models, offering robustness across languages and domains.

d. **Model Pre-training**: In this stage, the model is trained on the massive data collected using self-supervised objectives e.g. next-token prediction. Leveraging the tokenized data the model learns to capture broad linguistic and world knowledge within its parameters, resulting in foundation models that can serve as the basis for downstream adaptation.

e. **Fine-tuning** and **Instruction Tuning**:
    - **Fine-tuning**: The foundation models can be further trained on labeled, task-specific data to improve performance in specialized applications.
    - **Instruction Tuning**: The foundation models can be further aligned to follow human-like (natural language) instructions by training on curated instruction-response pairs, increasing their utility and alignment with user intent.

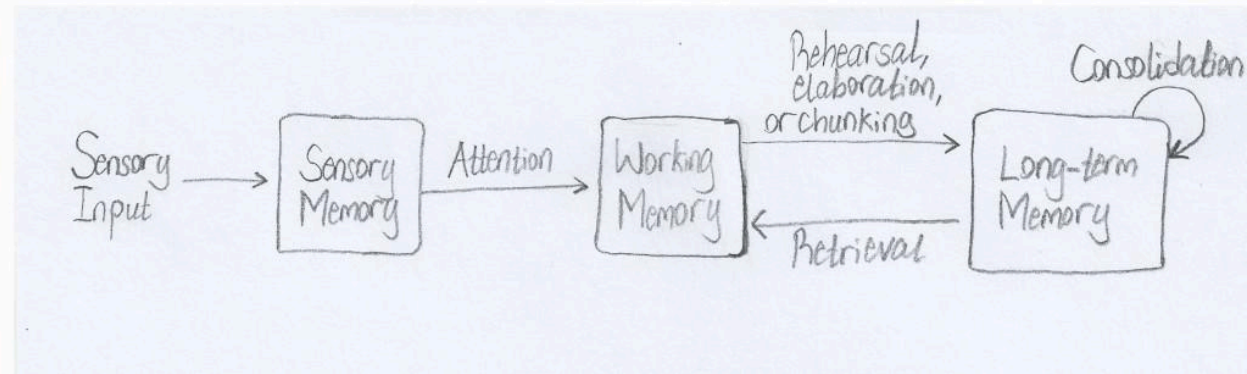
f. **Decoding Strategies**: This refers to the iterative process of text generation by pre-trained models. During each step the model processes the input sequence and produces a probability distribution across its entire vocabulary. The decoding strategy then determines how to select the next token from the computed distribution. This cycle repeats, appending each new token to the sequence, until a termination condition is achieved. Popular decoding strategies include:
    - **Greedy search**: This is the simplest approach, where the model consistently selects the token with the highest probability at each step.
    - **Beam search**: This strategy starts by picking N most likely tokens to form the multiple paths (beam) the prediction can go. It then explores every possible path independently, generating tokens, until a set number of tokens is reached. The path that produces the highest cumulative probability is chosen after which the process starts over with a new N sequence of tokens resuming with the chosen sequence.
    - **Top-k sampling**: This technique uses the probability distribution generated by the model to select a token randomly from the k most likely options.
    - **Top-p sampling**: This technique uses the nucleus sampling, where using the probability distribution generated by the model it picks the most probable tokens in descending order until the sum of probabilities exceeds a cutoff value p. This forms a “nucleus” of tokens from which to randomly choose the next token from.

2.3. Memory-Augmented Transformer

Despite the state-of-the-art performances in text-generation, language understanding, and efficiency in training, Transformer-based models still face challenges when it comes to long-range context retention, continual learning, and knowledge integration. To address these challenges, numerous works have turned to neuroscience-inspired dynamic memory mechanisms.

Memory plays a fundamental role in intelligence, enabling learning, reasoning, and adaptability in biological systems. In humans, memory enables the retention, retrieval, and manipulation of information across multiple time scales, supporting complex behaviour such as decision-making and problem-solving. It operates as an interconnected, multi-layer network that comprises three interacting subsystems:

- **Sensory memory**: This is a brief, high capacity buffer that actively holds raw sensory input for milliseconds to seconds before fading or being passed to the working memory.
- **Working memory**: This is a short-term limited capacity system that actively holds and manipulates information needed for reasoning, problem-solving and goal-directed tasks.
- **Long-term memory**: This is the brain's durable storehouse that retains knowledge and experiences for extended periods of times, even lifetimes, organized into episodic (personalized experienced events) and semantic (abstract knowledge, facts, concepts, and meanings) systems with consolidation processes (mechanisms that stabilizes new memories over time) that ensure stability and adaptability with time.



A critical process the brain employs to hold on to memory and avoid forgetting, especially in the Long-term memory, is memory consolidation. This process involves the reactivation and reorganization of memory traces (unit of memory storage), integrating them into existing knowledge networks. Through consolidation, memories become more stable and less susceptible to decay or interference, thereby reducing the likelihood of forgetting.

The Transformer-based model can be seen as loosely mimicking some of these memory subsystems such as the token embeddings and positional encodings being the raw sensory inputs to the sensory memory. These help in stabilizing the input for downstream layers. The attention mechanism is analogous to the working memory in that it actively maintains and

manipulates information required for reasoning, problem-solving, and goal-directed behaviour. Lastly the parameters of the models can be viewed as being analogous to the long-term memory subsystem; they can retain information obtained during training and then retrieve and implicitly integrate them in the generation process (inference).

Several studies have explored the use of explicit storage memory in the form of external memory modules to the Transformer-based models to enable scalable information storage and retrieval all with varying degrees of success. These modules are typically designed to be tightly integrated and usually differentiable with the model, thereby supporting end-to-end optimizations and enabling more sophisticated memory management strategies.

Some of these implementations include:

- **Memformer**: This is a Transformer-based model that implements an external dynamic memory for encoding and retrieving past information through timesteps so as to achieve linear time complexity and constant memory space complexity for processing long sequences of input. It uses a **slot attention** mechanism to write to the memory, where each memory slot attends to the input sequence and to itself to generate an updated memory. Furthermore a forgetting mechanism is implemented to clean up irrelevant or redundant information from the memory.
- **Neural Attention Memory (NAM)**: This is a Transformer-based model implementation that reimagines the attention mechanism as a memory architecture for neural networks. It involves writing to a memory matrix using key (K) - value (V) pairs and reading from it with a query (Q) vector. It utilized three differentiable operations for storage and recall, which are read, write and erase.
- **Retrieval-Enhanced Transformer (RETRO)**: This is an autoregressive Transformer-based language model with a retrieval module that utilizes chunk-wise retrieval and a retrieval database scaled up to trillions of tokens. The model splits both the input sequence and retrieval datastore into sequences of chunks. It then retrieves the nearest neighbour chunks from the retrieval database using the previous input chunk to guide and fuse the information with the context from preceding tokens to guide generation of the next chunk, this is done using a cross-attention mechanism called **Chunked Cross-Attention**. This design enables the model to outperform significantly larger contemporary models despite having fewer parameters, as it offloads long-term knowledge storage from its weights to its external retrieval mechanism.

2.4. Retrieval Augmented Generation (RAG) framework

This framework functions by incorporating information or knowledge from external data sources, to serve as supplementary reference or instructions for the input query or the generated output. This aids the **Transformer-based model** in its generation tasks. Unlike the **Memory-Augmented Transformers**, the **Retrieval Augmented Generation (RAG)** does not require the data source (Non-parametric memory) to be tightly integrated into the model specifically during the training process.

The main aim of the framework is to reduce the effects of hallucinations. It does so by first invoking a component called a **retriever** to search and extract relevant documents in the external data source that correspond to the input query. The retrieved documents are then combined with the input query as contextually relevant, time-critical, and domain-specific information to enhance the model's generative capability whilst reducing the effect of hallucination.

The framework consists of three major processes:

i. Retrieval

In RAG frameworks, the retriever component functions as the information provider where given an input query from the Transformer-based model, it retrieves relevant knowledge by estimating similarity between the input query and documents from external sources.

Retrieval methods can be categorized into two types: **sparse** and **dense** retrieval based on how information is encoded. **Sparse** retrieval relies on word-based representations and is commonly used in text retrieval, where documents are selected according to the presence or frequency of specific terms from the input query. Classic algorithms used for this include **TF-IDF** and **BM25**. **Dense** retrieval, on the other hand, embeds both the input query and documents into continuous vector space, where relevancy is determined by similarity metrics such as cosine similarity between the input query and documents vector embeddings, enabling more effective retrieval beyond surface-level word overlap.

Retrieval granularity refers to the level of indexing at which a corpus is segmented for retrieval, such as documents, passages, tokens, or entities. The choice of retrieval granularity has a substantial impact on both effectiveness and efficiency. Coarser granularities, such as full-document retrieval, reduce storage overhead and speed up search but may introduce noise by returning large amounts of irrelevant content. Finer granularities, such as token or entity-level retrieval, provide more precise knowledge access but increase database size and computational cost during search.

ii. Augmentation

Augmentation describes the technical process that integrates the retrieved information from the retriever component into the generation process. There are three main types of integration:

a. **Input-Layer Integration**: A common strategy for integrating retrieved information is to concatenate the retrieved information with the original input query and jointly process them through the Transformer-based model. While this approach has proven effective, it is inherently constrained by the context window of the Transformer-based model.
b. **Output-Layer Integration**: Another kind of integration strategy is post-hoc, or output-layer integration, in which the retrieved information is combined with the model's generated output rather than the input, effectively merging retrieval and generation results at the final stage.
c. **Intermediate-Layer Integration**: This integration strategy involves retrieved information being injected into the Transformer-based model at select layers. This will allow the

external knowledge to influence the hidden representations directly, offering a balance between the contextual richness of input-level integration and the efficiency of output-level integration.

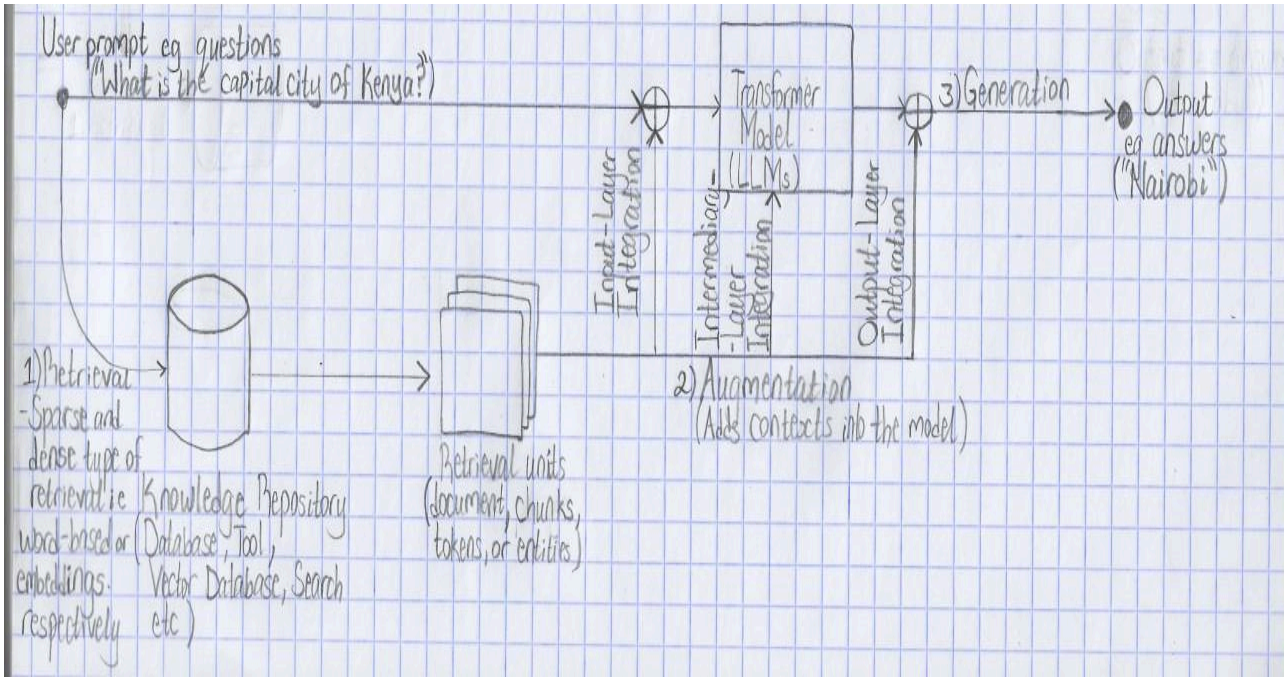
iii. Generation

In RAG framework, generators can be broadly classified into two categories:

a. **Parameter-Accessible Generators (White-box)**: These are generative models whose internal parameters are directly accessible for inspection, and modification. Such accessibility allows the models to be trained or adapted to different retrieval and augmentation strategies, thereby improving the quality and effectiveness of generation.

b. **Parameter-Inaccessible Generators (Black-box)**: These are generative models whose internal parameters are hidden from the users and are typically accessible only through interfaces or APIs. Interaction is thus limited to providing input queries (prompts) and receiving outputs (responses). In this setting, retrieval and augmentation processes aim to improve performance by enriching the prompt with supplementary knowledge, guidance, or illustrative examples (in-context learning).

Below is a simple diagram illustrating the various processes in a RAG framework:



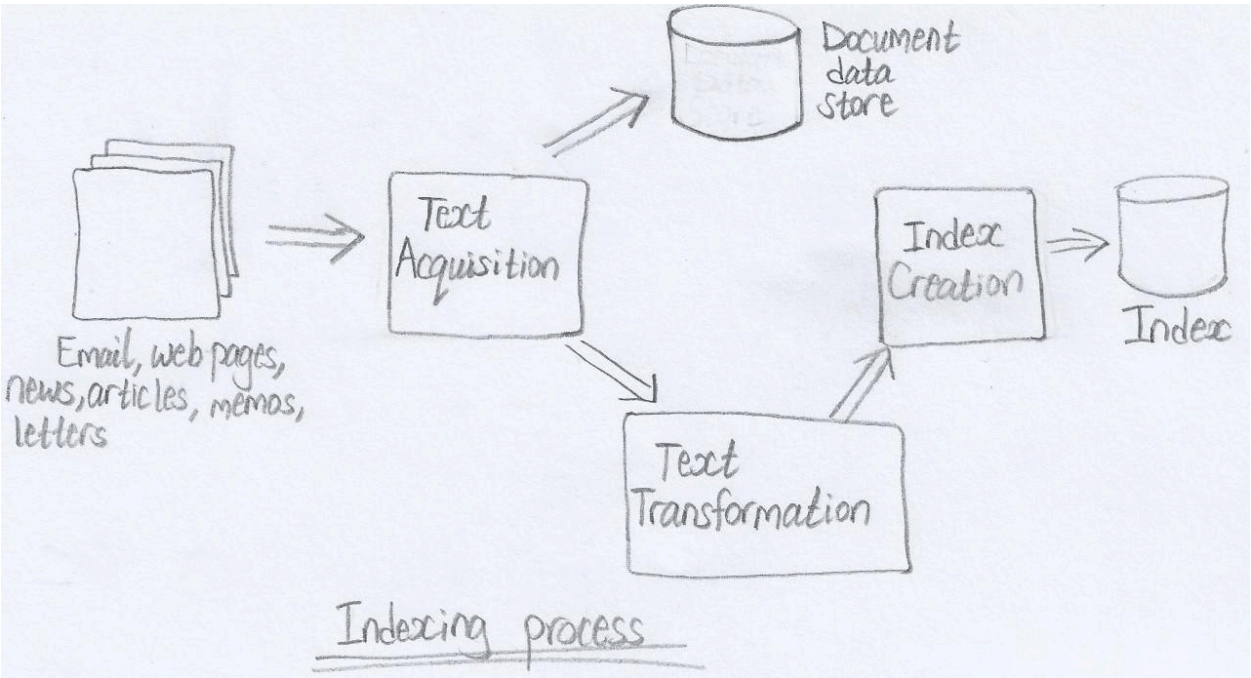
2.5. Information Retrieval (IR) system in the context of Search Engines

This is a software framework that structures, analyses, organizes, stores, searches, and retrieves information typically in the form of documents. It can handle multiple types of information including text, images, audio, and video. A search engine is usually considered the practical application of information retrieval techniques to large-scale text collection

Users interact with this system through an interface, where they express their information needs as a query. This query is then processed into index terms to be matched against a database. The retrieved documents, and other contents are sorted (ranked) by relevancy and returned to the user as a list, usually this is a list of links to websites in the case of search engines.

The Information Retrieval (IR) system generally comprises two core functions:

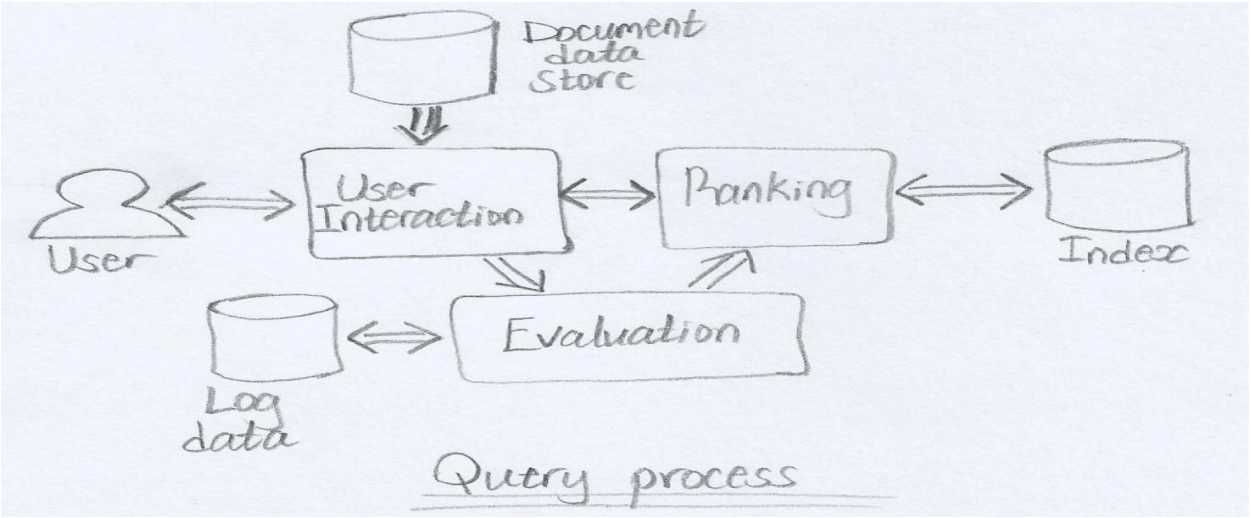
i. **Index processes**: These processes build the structures that enable efficient search. It involves:
    a. **Text acquisition:** Identifying and ingesting documents or content to be searched for, usually done by using crawlers in cases of web content. A knowledge repository, often a database (RDBMS), manages the corpus and associated metadata such as document type, document length, etc.
    b. **Text transformation:** Converting documents into indexable terms or features through the process of tokenization, parsing, and structural recognition such as titles, figures, links, and headings. Additional processes may include stop-word removal, stemming, link analysis (web pages), information extraction (**Named Entity Recognition: NER**), and classification (Identification of class-related metadata for documents or parts of documents). Both documents and queries must undergo consistent transformation to allow effective comparison. The set of all indexed terms forms the index vocabulary.
    c. **Index creation:** Constructing data structures, such as inverted indices to map terms to the documents in which they appear. This step incorporates document statistics such as counts of index term, number of tokens in documents etc., index term weighting (relative importance of words in documents), and distribution strategies to optimize search performance.



ii. **Query processes**: These processes leverage the index structures, created by the index processes, to retrieve relevant content to a user query. It involves:

   a. **User Interaction:** Managing the interface through which users can submit queries and view search results.The queries are transformed into index terms, which are used by the later processes to get ranked search results for the user. Query refinement techniques such as query suggestions, query expansion, and spell checking can be employed to improve the search performance.

   b. **Ranking:** The core retrieval operation, which scores documents against the query using a retrieval model such as **TF-IDF**, **BM25**, or neural ranking methods. The results are then ordered by relevancy before being returned to the user. The efficiency of this step depends on the underlying index structures and retrieval model.



c. **Evaluation:** Assesses both the effectiveness and efficiency of the system. It relies on user interaction data such as log data, click-through rates, or dwell time to measure performance. The insights gained are then used to tune indexing, ranking, and overall system design to improve retrieval outcomes. This is mostly done offline. Other performance metrics that can be measured are:
    - **Response time**: Delay between user submitting a query and receiving the result list.
    - **Query throughput**: Number of queries that can be processed in a given time.
    - **Indexing speed**: Rate at which text documents can be transformed into indexes for searching.
    - **Coverage**: How much of the existing information has been indexed and stored in the system.
    - **Recency or freshness**: "Age" of the stored information.
    - **Index Update Latency**: How fast new data can be incorporated into the indexes.

Retrieval models

This determines how documents are scored and ranked in response to a user query. They determine what is considered relevant by specifying how queries and documents are represented and compared. Some major categories are:

- **Boolean Retrieval models**: Represent queries and documents using boolean logic (AND, OR, NOT) e.g "Machine" AND "Learning" returns documents with both "Machine", and "Learning" terms.
- **Vector space models**: Documents can be ranked by computing the distance between vector representations of documents and queries.
- **Probabilistic models**: Uses probability theory to estimate the probability that a document is relevant to a query.
- **Neural and Deep learning-base models**: Uses embeddings and neural networks to capture semantic similarity beyond exact word matching.

Social Search

Integrates social signals, such as user interactions, social networks, ratings, likes, shares, tags, or recommendations, into the retrieval and ranking process of a search engine. Unlike traditional systems that rely primarily on textual relevance, social search leverages collective user behaviour and social context to enhance the relevance and personalization of results.

User Tags and Manual Indexing

Many social platforms enable users to assign tags to content, a practice often referred to as collaborative tagging, or folksonomy. The tags serve as additional indexable terms for the IR system, serving as a form of manual indexing in which the content of an object is represented through explicitly assigned descriptors. They can be used for documents, images, videos and

audio. In certain areas, the tags could be manually generated by experts, who choose keywords, categories, and other descriptors from a controlled vocabulary (fixed ontology) so as to ensure the descriptors are standardized.

However, manual indexing is infeasible for large-scale document collections or massive digital media content. To address this, the Information Retrieval system employs automatic indexing techniques to assign identifiers (terms, phrases, features) to documents during index construction. While automatic methods are more exhaustive and consistent, their quality and accuracy are often lower compared to human-assigned tags.

Categories of tags

Tags can be categorized into several types:

- **Content-based tags**: Tags describing the content of an item e.g. “car”, “woman”, and “sky”.
- **Context-based tags**: Tags that describe the context of an item e.g. “Nairobi City” for a photo of a street there or “KICC” for a document describing the engineering work done on that building.
- **Attribute tags**: Tags that describe implicit attributes of the item e.g. “building” for a document describing the engineering work done on KICC building, “black and white” for a type of movie.
- **Subjective tags**: Tags that subjectively describe an item e.g. “pretty”, “amazing”.
- **Organizational tags**: Tags that help organize items e.g. “todo”, “readme”, “draft”.

2.6. Case-Based Reasoning (CBR)

Case-Based Reasoning (CBR) is a problem solving paradigm in the field of Artificial Intelligence (AI) that emerged from cognitive science research in the 1980s. It represents a branch of Knowledge Based System (KBS), which integrates a knowledge repository (Knowledge Base) with an inference engine that acts like a search engine.

In CBR terminology, a **case** is a knowledge representation of experience, including the content of past lessons learned and the context in which these lessons can be used and a **case base** is a repository containing a collection of cases. Cases can be expressed as: *case = {P,S}* where P and S refer to a set of features describing the problem and a set of features describing the solution, respectively.

The core idea of Case-Based Reasoning (CBR) is to use similar problems from past cases to reasons and solve new problems based on cognitive psychology: similar problems have similar solutions. This entails:

- Adapting old solutions to meet new demands.
- Using old cases to critique new solutions.
- Reasoning from precedents to interpret a new situation (much like how lawyers do).
- Create an equitable solution to a new problem (much like labour mediators do).

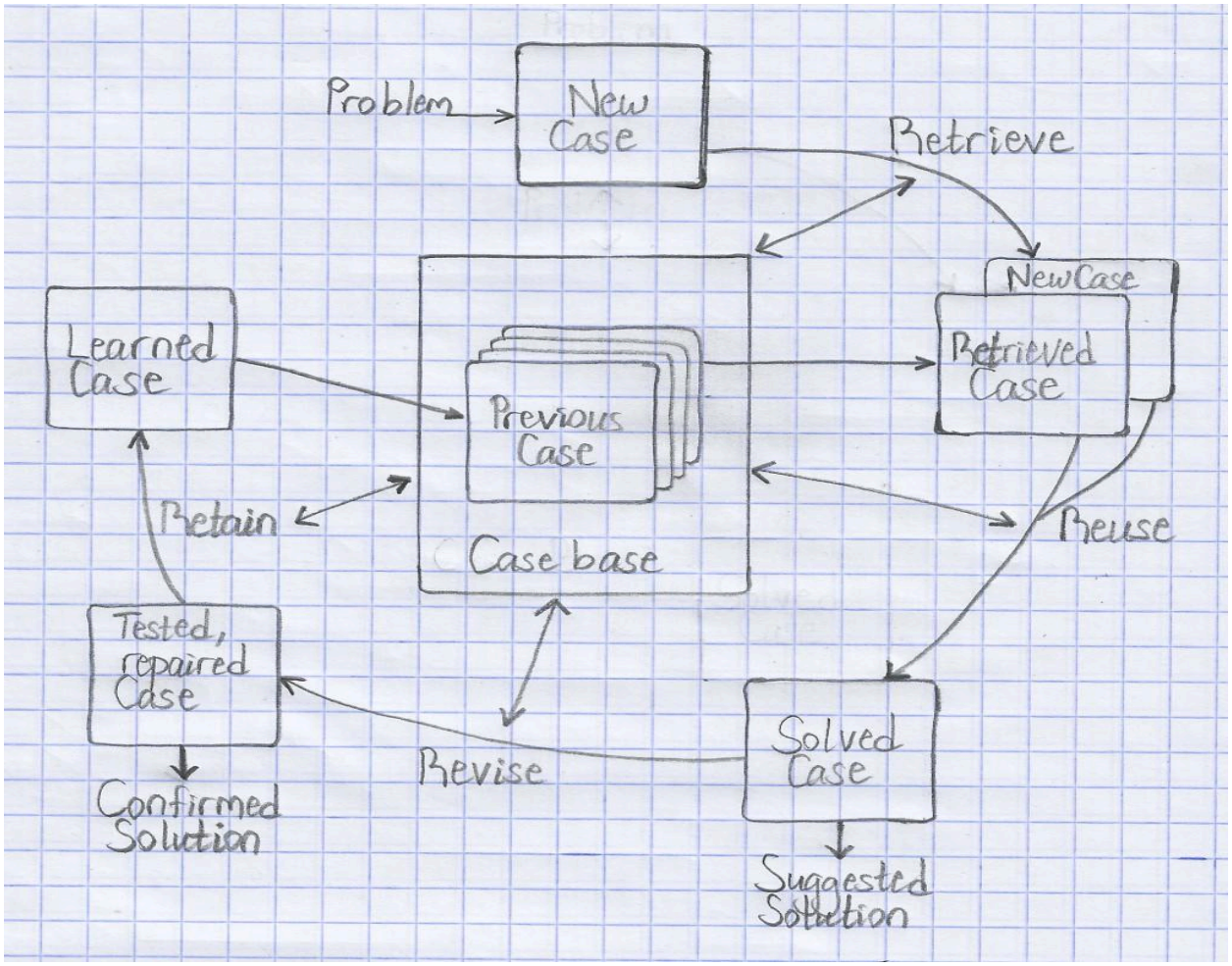
CBR is fundamentally different from other major AI approaches in that instead of relying solely on the general knowledge of a problem domain or making associations based on generalized relationships between problem descriptors and conclusions, it can use the specific knowledge of specific problem situations (**cases**) experienced in the past to solve similar new problems.

Case-Based Reasoning is in effect a cyclic and integrated process that consists of the following steps (**4R cycle**):

i. **Retrieve**: One or more cases most similar to the new cases are retrieved from the case base.
ii. **Reuse**: Information and knowledge from similar cases are reused to establish solutions adapted to new cases.
iii. **Revise**: The proposed solution is evaluated and the solution is adjusted if it doesn't meet the requirements.
iv. **Retain**: The parts of this experience that may be useful for solving problems in the future are retained in the case base.

An initial description of a **problem** defines a **New Case**. This **New Case** is used to **Retrieve** a case from the collection of previous cases in the **Case Base**. The **Retrieved Case** is combined with the new case (**Reuse**), into a solved case i.e. a proposed solution to the initial problem. Through the **Revise** process this solution is tested for success e.g. by being applied to the real world environment or evaluated by a teacher, and repaired if failed. During the **Retain** process, useful experiences are retained for future reuse, and the **Case Base** is updated by a new **Learned Case**, or by modification of some existing cases.

A visual depiction of the **4R cycle**:



CBR offers the following advantages:

- Enables the proposal of solutions to problems quickly, avoiding the time necessary to derive those answers from scratch.
- Enables the proposal of solutions in domains not completely understood. They allow making assumptions and predictions based on what worked in the past without having a complete understanding.
- Enables a means of evaluating solutions when no algorithmic method available for evaluation. Solutions are evaluated in the context of previous similar situations i.e doing evaluation based on what worked in the past.
- They are useful for use in interpreting open-ended and ill-defined concepts.
- Remembering previous experiences is particularly useful in warning of the potential for problems that have occurred in the past i.e. actions that need to be taken to avoid repeating past mistakes.
- Cases help to focus reasoning on important parts of a problem by pointing out what features of a problem are the important ones.

However despite their advantages, there are still pitfalls in using cases to reason:

- CBR might be tempted to use old cases blindly, relying on previous experience without validating it in the new situation.
- CBR might allow cases to bias it too much in solving a new problem.
- CBR might not utilize the most appropriate set of cases during its operation.

3. Terminology

Term	Definition
Artificial Intelligence (AI)	The broad field of computer science focused on building systems that can perform tasks requiring human-like intelligence, such as reasoning, perception, learning, and language understanding. Usually used as an umbrella term for intelligent systems.
Machine Learning (ML)	A subfield of AI focused on algorithms and models that learn patterns from data rather than relying on explicitly programmed rules. It provides the foundation for modern algorithms like Transformer-based models.
Frontier model(s)	The most advanced, large-scale machine learning models that are at the cutting edge of research and deployment, typically trained with massive compute and data resources.
Large Language Model (LLM)	An AI system trained on massive text datasets to understand and generate human language, using large neural networks to perform tasks like answering questions, writing text, translating languages, and reasoning across contexts.
Natural Language Processing (NLP)	A subfield of AI and ML focused on enabling computers to process and understand human language. It is a core discipline behind chatbots, translation, summarization, and text classification systems.
Computer Vision (CV)	A subfield of AI and ML focused on enabling machines to interpret and analyse visual information from images or video. Its core tasks include object detection, segmentation, and tracking.
Recurrent Neural Network (RNN)	A type of machine learning model designed for sequential data. It processes input step by step while maintaining hidden states that capture information across time.
Convolutional Neural Network (CNN)	A type of machine learning models that apply convolutional operations to capture local patterns in data, most commonly images but can also be applied to NLP, and audio processing tasks.
Decision Tree	A machine learning algorithm that makes decisions by recursively splitting data based on feature values, forming a tree of rules that leads to a prediction or classification.
Support Vector Machines (SVM)	A machine learning algorithm that classifies or regresses data by finding the optimal separating hyperplane that maximizes the margin between data points of different classes.

Non-linear function, Activation function, Squashing function	A mathematical function in which the output does not change proportionally with the input, meaning it cannot be represented as a straight line and often enables models to capture complex patterns.
Mean Square Error (MSE)	A loss function that measures the average of the squared differences between predicted values and actual values, emphasizing larger errors.
Corpus	A large and structured collection of text documents used for training, evaluation, or retrieval in NLP and IR systems.
Tokens	The smallest unit of text a machine learning model processes, which may be a word, subword, or character depending on the algorithm used to split them.
Entities	Real-world objects, concepts, or items that can be identified and classified within text e.g. people, organizations, locations, dates, products etc.
Prompt	The input text (instructions, queries, or examples) provided to a language model to guide its output. It can include system messages, user instructions, and context documents.
Inference	The process of using an already trained machine learning model to generate predictions or outputs from new input data. This is distinct from the training process.
Foundation model(s)	Large-scale machine learning models trained on broad and diverse datasets that can be adapted (via fine-tuning, prompting, or RAG) to many downstream tasks.
Information Retrieval (IR)	The field of computer science focused on finding relevant information in large collections of unstructured data (typically text).
Best Match 25 (BM25)	A ranking function used in information retrieval to estimate the relevance of documents to a given query, based on term frequency and inverse document frequency.
Term Frequency–Inverse Document Frequency (TF-IDF)	A statistical weighting method in information retrieval that scores how important a term is within a document relative to a large corpus, by combining term frequency (TF), which is the number of times a term appears in a document, with inverse document frequency (IDF), which is the a measure of how rare or informative a term is across a collection of documents.
Cosine similarity	A metric that measures the similarity between two vectors by calculating the cosine of the angle between them. It ranges from -1 (opposite) to 1 (identical), with 0 meaning orthogonal (no similarity).

In-context learning	The ability of a machine learning model to adapt its behaviour to new tasks by conditioning on examples or instructions provided in the input prompt, without updating the model's parameters.
Continual Learning	Also known as Incremental Learning is the ability of a model to progressively learn new tasks, one at a time, without forgetting the previously learned ones.
Parametric memory	This refers to long-term memory implicitly stored within a model's parameters. It is acquired during training, where they're embedded in the model's parameters and accessed through feedforward computation at inference. It serves as a form of instant, long-term, and persistent memory.
Non-parametric memory	This refers to long-term memory that is stored externally to the machine learning models. It can be stored in various media such as databases, file systems, and computer memory.
Retrieval model	Formal representation of the process of matching a query and a document, file or other media.
Relational Database Management System (RDBMS)	A software system for managing structured data organized into tables with predefined schemas and relationships. Supports SQL for querying, updating, and managing data. Examples: PostgreSQL, MySQL, etc.
Stop-words	Common words in a language like "the", "of", "to", and "for" that help form sentence structure but offer little semantic meaning on their own.
Stemming	Groups words derived from a common stem i.e. "fish", "fishes", and "fishing" can be grouped with one designated word "fish".
Crawlers	Automated programs that systematically browse the web (or other data sources) to collect and index context.
Named-Entity Recognition (NER)	An NLP task that identifies and classifies entities in text into predefined categories such as people, organizations, locations, dates, or products. Often used as a preprocessing step for information extraction, search, etc
Differentiable	A mathematical property of a function that allows its derivatives to be computed. In machine learning, it means the model's parameters can be optimized. This is what makes neural networks trainable.
Subword	A text unit smaller than a full word but larger than a single character. Produced by tokenization methods such as BytePairEncoding (BPE) or WordPieceEncoding.

Out-of-vocabulary (OOV)	A condition where a token (word, subword, or character) is not present in the model's predefined vocabulary.
Unigram Language Model	A probabilistic model that treats each token as independent and estimates the likelihood of a document or query by multiplying the probabilities of its individual words. Used historically in IR as a ranking function.
Slot Attention	A fixed set of learned memory slots from a Memformer architecture updated by attending over token representations; each slot compresses salient information from the sequence and is fed back into the Transformer-based as persistent external memory for long-range context.
Chunked cross-attention	A mechanism used in Retrieval-Enhanced Transformer (RETRO) where the model attends over retrieved text chunks, each a fixed-length segment from an external database, using separate cross-attention blocks; the retrieved chunks provide supplementary context that conditions token predictions without extending the primary attention window.
Tightly coupled	A relationship of components in a system that are highly dependent on each other. If one component fails, it will affect the others and eventually bring down the entire system.
Loosely coupled	A relationship of components in a system that are independent of each other. Each component has its own well-defined interface and communicates with other components through standardized protocols. Changes to one component does not require changes to other components.
Knowledge-Based System (KBS)	Application of Artificial Intelligence (AI) that relies on an information store (Knowledge Base) and reasoning logic (Inference Engine) to provide answers. Knowledge-based systems are often deployed in organizations to assist with decision-making, including in IT departments.
Knowledge Base (KB)	Formal structured store of declarative facts, semantic relations, and procedural rules represented using logic-based or symbolic schemas to support inference and reasoning.
Inference Engine	Mechanism that executes logical inference to derive new assertions or reach goal states.
Flops	A measure of computational workload, expressed as *floating-point operations per second* (or total floating-point operations), used to quantify the processing required to train or run a model.

Local minimum	A point where a function’s value is lower than all nearby values, but not necessarily the lowest overall.
Global minimum	A point where a function attains its lowest value over the entire domain.
Saddle point	A critical point where the gradient is zero, but the point is neither a local minimum nor a local maximum (opposite of a local minimum).
Entropy (Information Theory)	A quantitative measure of the uncertainty or average information content of a random variable, reflecting how unpredictable its outcomes are.

4. Out of Scope (Non-Goals)

- **Frontier model performance**: The focus of the project will be on evaluating the proposed systems capabilities and not producing a frontier model.
- **Dynamic knowledge repository**: The knowledge repository will be static with no automated mechanisms for updating, adding, or removing information from it. However this does not mean information in the documents can't be manually changed.
- **External knowledge access**: All the data will be localized to the knowledge repository, with no access to external data sources.
- **General-purpose reasoning across arbitrary domains**: The scope of the model's reasoning capabilities will be constrained to the information present in the knowledge repository.
- **End-to-end training on raw or unlabeled data**: The project will not rely on large-scale unstructured datasets arbitrarily obtained from sources like the internet, like most frontier models, but instead it will operate on a carefully curated and structured dataset.
- **Input multimodality**: The project will only utilize text data.
- **Benchmark priority**: The project will not prioritize outperforming any established models on any benchmark.
- **Agentic system**: The project will not pursue the development of any autonomous agentic system designed to interact with users and environments.
- **Reinforcement Learning alignment**: The project will not implement any reinforcement learning algorithms to improve the model's performance such as Reinforcement Learning with Human Feedback (RLHF).
- **Prompt engineering approaches**: The project will not utilize any of the prompt engineering approaches such as chain-of-thought (CoT), Tree-of-Thought (ToT), etc. to improve the generation process.

5. Assumptions

The following assumptions have been made about the project:

- **Precise control**: Conditioning a model on auxiliary information allows for finer control over its output, provided the model is properly trained and the conditional input is applied consistently. This approach should enable stronger quality control over the model's output by allowing explicit evaluation and verification of the conditional inputs being supplied to it. In turn, this should support more robust knowledge management by ensuring that only accurate and up-to-date information is delivered to the model. This concept draws inspiration from conditional image synthesis in Computer Vision models, where diverse sources of conditional inputs can be used to guide the image generation process allowing for more precise control to meet people's diverse needs.
- **Scalability and maintainability**: By utilizing a modular design that decomposes complex systems into multiple parts called modules i.e. a singular Transformer-based model is split by tasks into multiple Transformer-based models, the overall system should be more scalable and maintainable when compared to a monolithic system due to the modules being loosely coupled. This is due to each module focusing on a simpler, manageable, and independent task that can be scaled independently and updated separately without directly impacting each other's performances.
- **Increased parameter efficiency**: Using non-parametric memory should reduce the model's dependency on parameter scaling (scaling laws), as most knowledge is maintained in external storage rather than embedded in the model's parameters. This should allow for more efficient utilization of the model's parameters, possibly reducing the training time and computational requirements. This has been observed to be the case with **Retrieval-Enhanced Transformer (RETRO)** models (Memory-Augmented Transformer) which has been shown to obtain comparable performance to other models despite using 25× fewer parameters on certain datasets by simply applying the retrieval process during pre-training.
- **Low training and inference cost**: Smaller models augmented with non-parametric memory can offer significant cost and compute efficiency for both training and inference. The assumption here is that incorporating retrieval during pretraining reduces the need for rote memorization of factual knowledge, shortening training time. In addition the reduced model size could lower inference latency, which would reduce inference cost when deployed. All of which should be possible if and only if the relevant content and conditional input (context) is easily accessible during training unlike how **Retrieval-Enhanced Transformer (RETRO)** does it i.e. there should be no retrieval overhead in the training process.
- **Enhanced reasoning capability**: By shifting factual knowledge storage to non-parametric memory, the model can allocate more of its capacity to higher-order language understanding, rather than rote memorization to allow for factual recall. This should make the model more capable at reasoning effectively with fewer parameters given the conditional input is sufficiently complete and reliable.
- **Scalable dataset creation**: The project requires a well-structured dataset for training. This will be obtained through a combination of automated acquisition and synthetic

generation. This process will involve pre-trained open-source models to generate or refine synthetic samples, and human annotation to ensure quality and relevance specifically with the document tags. The key assumption is this process can be scaled efficiently to generate the needed dataset for training.

- **No model collapse**: Model collapse is a phenomenon whereby models trained recursively on data generated from the previous generation of itself over time will lead to a degraded performance, eventually making it completely useless. This project assumes this won't be the case when utilizing synthetic dataset in the training dataset by employing partial manual annotation and varied open-weight models (not output from the same model) in the dataset creation as well as limiting synthetic data points to only conditional input i.e. summaries.
- **Low latency**: The modular design of the system will inevitably introduce delays in the information flow between modules during inference. To ensure practicality and scalability, these delays must remain within an acceptable limit. In particular, the Information Retrieval (IR) system should maintain response time below or equal to 100ms (considered acceptable for general web browsing), while inter-module communication during inference should achieve comparable performance, preserving overall system efficiency. Caching can be employed to improve latency where applicable.
- **Generalization**: The models in the overall system should generalize well to unseen dataset i.e. data held out during training. The models should be capable of producing coherent and accurate results for their respective task during inference.
- **External storage scaling**: Expanding the memory storage capacity i.e. the knowledge repository, within the Information Retrieval (IR) system should remain both cost effective and technically feasible.
- **Adherence to scaling law**: The performance of the model is positively correlated with the scale of model parameters, size of training dataset, and amount of computation used during training therefore the performance of a smaller model should be indicative of the capability of the architecture and if scaled up it should improve in a predictable manner (Scaling law). In other words, a strong performance from a smaller model should signal that the architecture remains viable when scaled to larger parameter counts.
- **Training stability**: Pretraining a model with retrieval should remain stable and scalable, with no unintended behaviours such as performance degradation performance over time whilst training.
- **Encoder-Decoder model scaling**: The project assumes Encoder-Decoder (Sequence-to-Sequence) models can scale, remain stable, and generalize properly to specific tasks like Decoder-only models that are used primarily for state-of-the-art chatbots.

6. Proposed Design

6.1. Introduction

This project aims to employ the Unix Philosophy into the design process of machine learning models, specifically Transformer-based models, by decomposing certain implicit functionalities into explicit smaller, manageable modules that can be independently developed, trained (if components are models) and composed into a complete system for deployment. This will involve the usage of multiple Transformer-based models and an Information Retrieval (IR) system as modules that each focus on narrowly-defined and non-overlapping functionalities.

In addition, certain concepts from Retrieval Augmented Generation (RAG), Memory-Augmented Transformers, Case-Based Reasoning (CBR), and Information Retrieval (IR) systems will be integrated into the design and implementation of the modules:

- **Retrieval Augmented Generation (RAG)**: Sparse retrieval methodology, intermediate-layer Integration, and parameter-accessible generators (White-Box).
- **Memory-Augmented Transformers**: External dynamic Long Term Memory (LTM) component for encoding and retrieving past information.
- **Case-Based Reasoning (CBR)**: 4R Cycle (Retrieve, Reuse, Revise, Retain cycle).
- **Information Retrieval (IR) system in the context of Search Engines**: Boolean retrieval from knowledge repository, ranking retrieved documents by relevancy using weighting algorithms similar to TF-IDF, creating explicit user-defined tags and manual indexing.

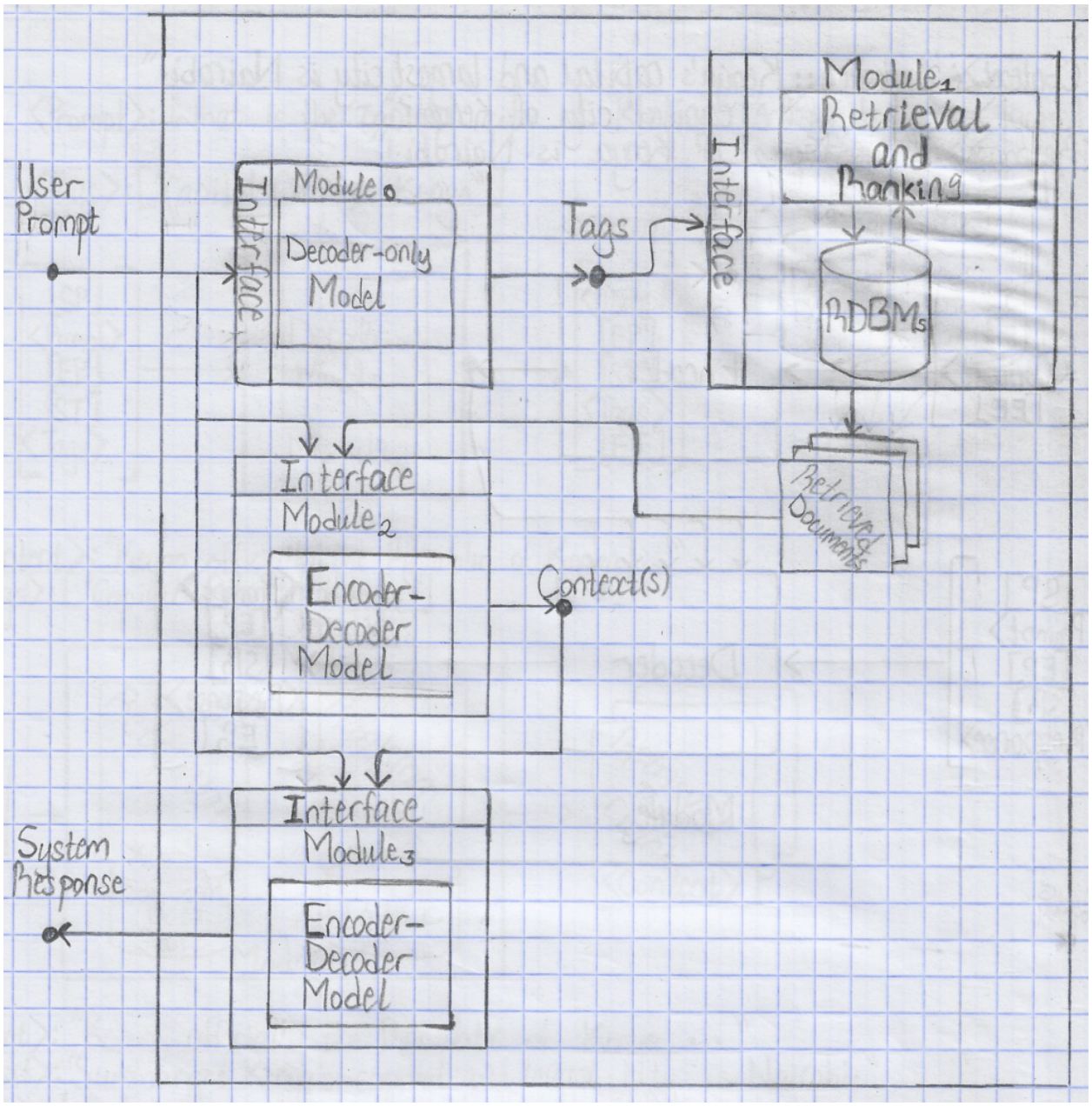
To summarize how the system works:

1. A **user prompt** is provided to the system and passed to **Module$_0$**. This module consists of a **Decoder-only Transformer-based model** that autoregressively generates pre-defined user **tags** given the **user prompt** until a stopping condition is encountered.
2. The generated **tags** are passed to **Module$_1$**, which consists of an **Information Retrieval (IR)** system which utilizes a **File storage** and **Database (RDBMs)** for document storage and indexing respectively and a **Retrieval** and **Ranking** component. The system retrieves relevant text chunks (documents) based on the provided **tags** and then sorts (ranks) them using a custom weighting mechanism (**Inverted Tag Frequency**).
3. The **retrieved documents** and the **user prompt** are then passed to **Module$_2$**, where an **Encoder-Decoder Transformer-based model** autoregressively generates a **summary** of the **retrieved documents** in relation to the **user prompt**. This summary can be considered a form of adaptation of the stored knowledge to a new problem (prompt). There are two principle types of summarizations that can be done:
    a. Abstractive: Original summaries using sentences not found in the original text chunks.
    b. Extractive: Unmodified sentences from the original text chunks.
4. The **summarized** / **adapted** information is finally passed to **Module$_3$**, which consists of another **Encoder-Decoder Transformer-based model** that autoregressively generates a coherent and factually correct **response** to the **user prompt**.

5. Further manual **evaluation** of the overall system is performed to assess its performance. Based on the results, modules and their respective components are updated, maintained, patched, or replaced to fix any shortcomings via **Revisions**, **Retention**, and **Fine-tuning** operations.

6.2. System Architecture

Below is a high-level conceptual overview of the system described above:



Each module will have the following components:

- **Module$_0$**: A Decoder-only Transformer-based model.
- **Module$_1$**: An Information Retrieval (IR) system.
- **Module$_2$**: An Encoder-Decoder Transformer-based model.
- **Module$_3$**: An Encoder-Decoder Transformer-based model.

6.3. Specialized Tokens

All of the models will need to utilize specialized tokens to signal special operations to be performed and to clearly delineate the boundaries of information segments, such as indicating the start and end of specific content. These tokens will be used to guide the flow of information within the components, and signal for actions that need to be performed on the data. They include:

- **[null]:** Null token; Standard token used for the absence of context in the Encoder-Decoder model from **Module$_3$.**
- **[Pad]**: Padding token; Standard placeholder token to ensure inputs to a model are of the same context size, it is usually the max context window each model can support.
- **[SP] / [EP]:** Start of Prompt / End of Prompt tokens; denotes the start and end of user prompt(s) used by all models.
- **[ST] / [ET]:** Start of Tags / End of Tags tokens; denotes the start and end of tags generated by the Decoder-only model from **Module$_0$**.
- **[SR] / [ER]:** Start of Response / End of Response tokens; denotes the start and end of the Encoder-Decoder model from **Module$_3$** output response.
- **[SS] / [ES]:** Start of Summary / End of Summary; denotes the start and end of Summary to be passed into an Encoder-Decoder model from **Module$_2$** into **Module$_3$**.
- **[SE] / [EE]:** Start of Encoding / End of Encoding; denotes the start and end of encoding tokens in the Encoder model of **Module$_2$** and **Module$_3$**.
- **[RLV] / [NRLV]:** Relevant / Not Relevant tokens; specifies whether a retrieved document from **Module$_1$** into **Module$_2$** is relevant to the user prompt.

6.4. Dataset

The training dataset in the form of documents will need to be acquired from a few sources such as Wikipedia. After which the documents will need to be chunked into smaller manageable text chunks. Using open-source text-only models, such as **gpt-oss-120b**, these text chunks can be used to synthetically generate their corresponding **Context** and user-like **Prompts** to be used for training the models.

To ensure the **Context** is generated consistently and reliably by the open-source text-only model, the following guidelines should be applied, reworded to be a system prompt (**WIP: Work In Progress**):

- Dataset structure
    - Dataset will need to be in a JSON format similar to the **S**tanford **Qu**estion **A**nswering **D**ataset (**SQuAD**).
- Context categories:
    - Format for **Context**: **<Context_Label>::<Context_Information>**.
    - Use **Title Case** naming convention for **<Content_Label>**.
    - **<Context_Label>**s (Non-exhaustive):
        - **Content Type**: The type of the content e.g. story, poem, mathematical problem, play, etc.
        - **Writing Style**: The style of writing used in the source content e.g. expository, descriptive, narrative, dramatic, poetic, technical, journalistic etc.
        - **Summary**: Concise summary of text in a novel, article, code, etc.
    - **<Context_Information>** guidelines:
        - **<Context_Information>** must contain all key information from the content and be “invertible” (All key facts from the content must be represented such that one could attempt to reconstruct the content with little to no loss of key information).
        - **<Context_Information>** must be concise and use shortened terms where applicable.
        - **<Context_Information>** must utilize plain, simple English and not mimic the content’s writing style.
        - **<Context_Information>** must not add, interpret, or speculate beyond the information presented in the content.
        - **<Context_Information>** must preserve the names of places, people, organizations, abbreviations, and specific domain terminologies as used in the content.
        - **<Context_Information>** must preserve the order and structure of the content where applicable.

6.5. Model Training

The Transformer-based models in the system require to be **pre-trained** and later **fine-tuned** on a well-structured dataset to gain the required capability such as language understanding and reasoning capability for their respective tasks. The dataset for **pre-training** the models are grouped into two parts:

i. **Content**: The full text chunk.
ii. **Context**: A concise summary of the **Content** text strictly following a well-defined standard.

The models will be **pre-trained** as follows to form **foundational** models:

- Decoder - only model from **Module$_0$**: Pre-trained exclusively on the **Content** data.

- Encoder - Decoder model from **Module$_2$**: Pre-trained on the **Context** conditioned on the **Content**.
- Encoder - Decoder model from **Module$_3$**: Pre-trained on the **Content** conditioned on the **Context**.

The **foundational** models will then be **fine-tuned** on synthetically generated dataset grouped into four parts:

- **Content**: The full text.
- **Context**: A summary of the **Content** text relevant to the **Prompt.**
- **Tags**: A list of user-created keywords that form a controlled vocabulary to index documents by categories e.g domain content is from, or a descriptive term that reflects the **Content**.
- **Prompt**: Synthesized user input usually in the form of questions or instructions.

The models will be **fine-tuned** as follows:

- Decoder - only model from **Module$_0$**: Fine-tuned on a combination of **Prompt** and **Tags**.
- Encoder - Decoder model from **Module$_2$**: Fine-tuned on the **Prompt** conditioned on the **Content**.
- Encoder - Decoder model from **Module$_3$**: Fine-tuned on the **Prompt** conditioned on the **Context**.

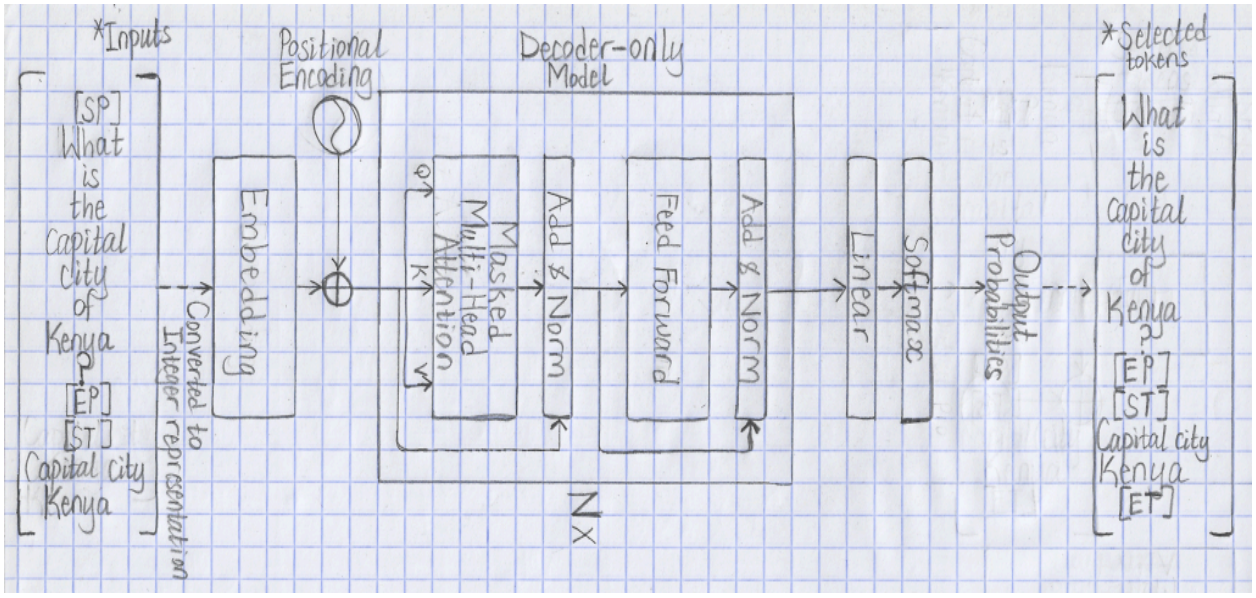
The **Content** and their respective **Tags** will need to be stored in the database in **Module$_1$** and made accessible for the **Information Retrieval (IR)** system for the retrieval process.

6.6. Visual depiction of how each Module works during Inference.

i. **Module$_0$**

Given a **user prompt** the Decoder-only model (Transformer-based model component) will generate **Tags** relevant to the prompt i.e. the model performs a **Named Entity Recognition (NER)** task.

➢ Architectural diagram of the Decoder-only model



ii. **Module$_1$**

Using the **Tags** generated from the previous **Module$_0$**, the succeeding **Module$_1$** will initially utilize the **Information Retrieval** system to retrieve the documents (text chunks) based on the **Tags** provided. This is followed by ranking the retrieved documents using the weighting scores to sort them by order of relevancy. The weighting score used here is slightly modified from the **Inverse Document Frequency** formula from the **TF-IDF** algorithm to form the **Inverse Tag Frequency**.

Modified formula (**Inverse Tag Frequency**)

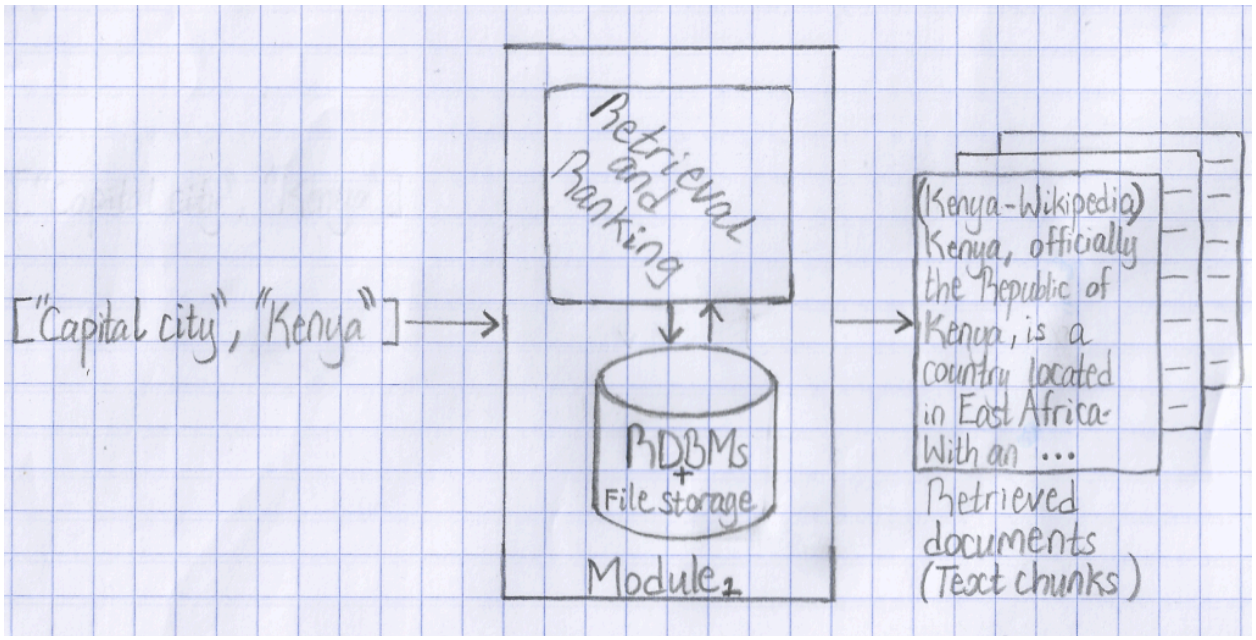
$$itf(t, D) = log \frac{N}{n_t},$$

where D is the set of all documents stored in the database, t is a specific tag stored in the database, $N = |D|$ is the total number of documents in the database, $n_t = |\{d \in D: t \in d\}|$ is the number of documents that are indexed by the specific tag t.

Each document in D can have multiple tags assigned to them e.g. a document with information regarding the East African country of Kenya, its population size, capital city, etc. could have the tags: [“Kenya”, “Capital city”, “Population”, …]. Each tag, t, will be assigned a specific weight, $itf(t, D)$, where it is assumed to always be recalculated when new documents or tags are added or modified in the database and then stored in a hash map for easy access.

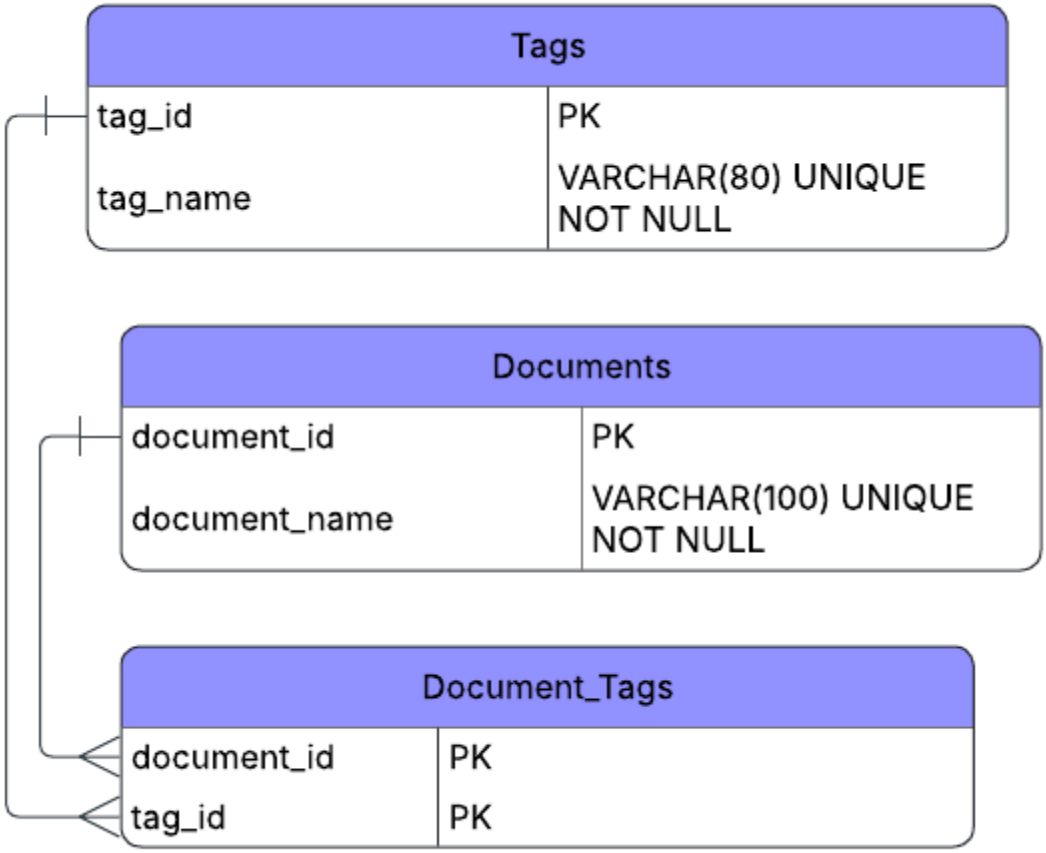
Intuitively, $itf(t, D)$ represents the entropy of a tag: the more frequent a tag is used, the less information it has; conversely, the less frequent it’s used, the more information it has. For example, given the tags: “Capital city” and “Kenya”, it can be assumed there are numerous documents with the “Capital city” tag but few with “Kenya” tags, therefore when sorting documents, those documents containing both “Kenya” and “Capital city” should be at the top, followed by those with only “Kenya” tag and lastly those with only “Capital city” tags. This is because the documents with only “Kenya” are assumed to have more relevant information for the given problem (prompt) than those with only “Capital city”.

➢ High-level conceptual overview of the module



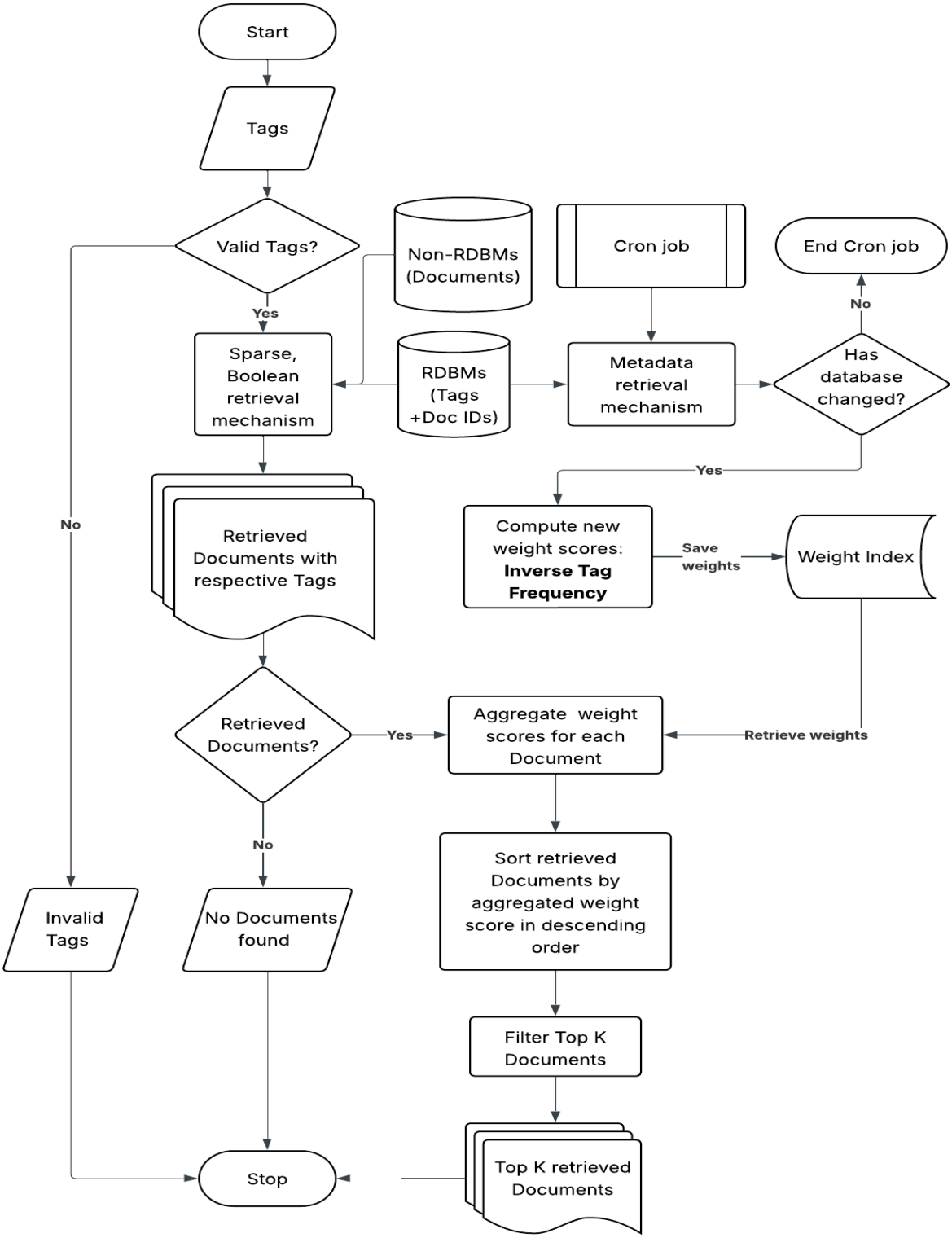
➢ Entity Relationship Diagram (ERD) of the RDBMs

Used https://lucid.app



➢ Flowchart diagram of the module

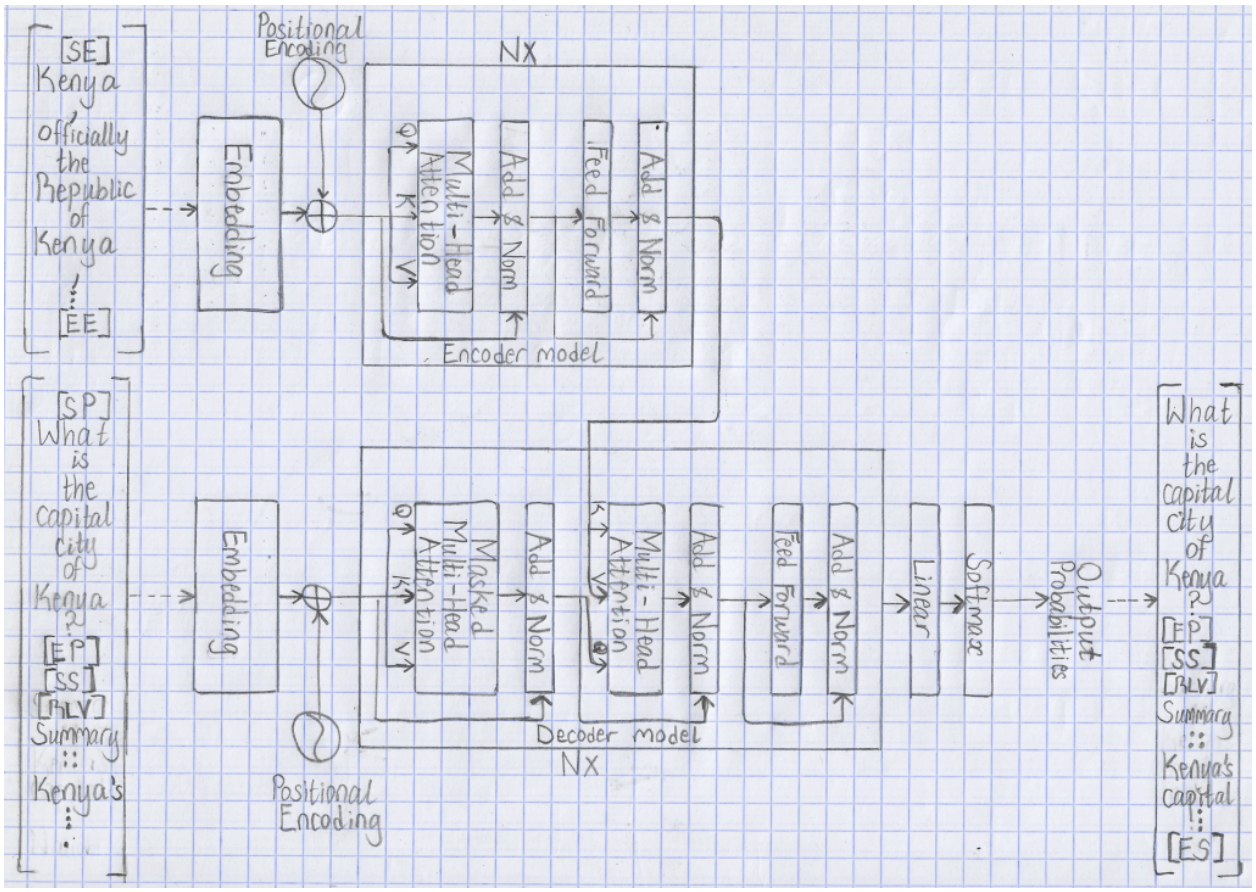
Used https://lucid.app



iii. **Module$_2$**

The retrieved documents (text chunks) from **Module$_1$** are then passed to **Module$_2$**, which is an Encoder-Decoder model. Here the document is passed to the Encoder (a non-causal model). The resulting encodings will then be passed to the Decoder (causal model) which first determines if the documents are relevant or not using either **[RLV]** or **[NRLV]** specialized tokens respectively. The relevant documents are further processed to summarize the text chunks into meaningful data relevant to the prompt.

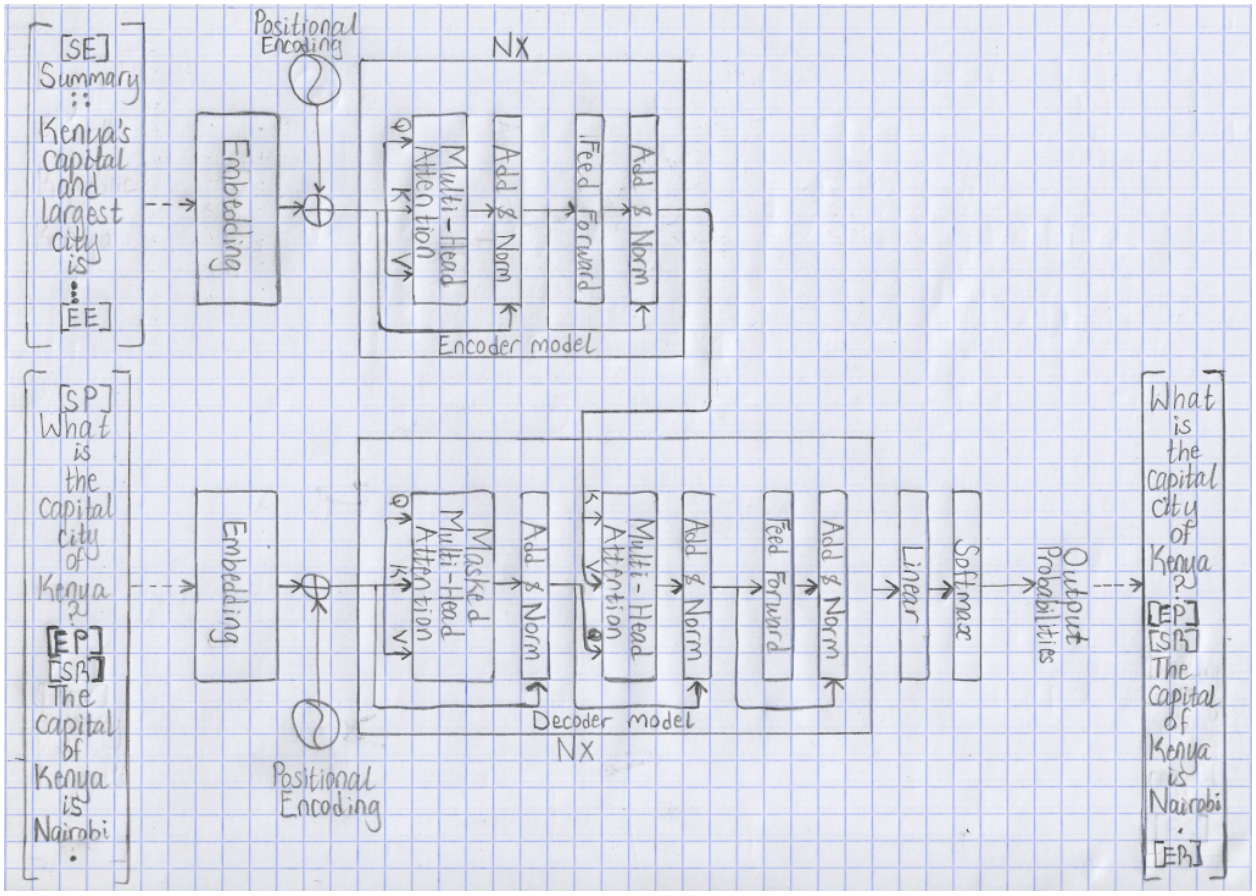
➢ Architectural diagram of the Encoder-Decoder model



iv. **Module$_3$**

The summarized text from **Module$_2$** is then passed to the **Module$_3$**'s Encoder model where its resulting encodings are integrated to the Decoder's model prompt input. The Decoder module's task is to generate a response to the prompt in the form of an answer to a question or any other relevant text.

➢ Architectural diagram of the Encoder-Decoder model



7. Justification

7.1. Conceptual Analogy

The core principles underpinning Artificial Neural Networks (ANNs) such as their function composition operations and learning processes have not changed much since their conceptions. Most of the recent advancements in the field have mostly come from algorithmic refinements, architectural innovations, massive increases in compute (flops), availability of vast and diverse training dataset, and ability to massively scale models for approximating more complex functions.

An apt analogy of this would be with the internal combustion engines in vehicles i.e. car engines that run on petrol or diesel. Since the first true vehicle running with an internal combustion engine in July 1886, the Patent-Motorwagen, the fundamental principles underpinning those vehicles and those that came after it have remained mostly unchanged for more than a century.

The internal combustion engine works by taking in air and fuel and mixing them in a combustion engine, where they're compressed and ignited to force a chemical reaction. The chemical reaction produces enough kinetic energy to drive various mechanical components in the engine such as a piston and drive shaft that powers the vehicle. Most of the improvements in performance, efficiency, and reliability from these types of engines have stemmed more from advances in material science, precision engineering, manufacturing processes, and computerized systems.

This highlights how a well-engineered design can remain effective over long periods of time, proving that technological progress can rely more on refinement and ingenuity rather than constant reinvention.

Using another analogy, observations in biological systems indicate that brain size alone is not the sole determinant of an organism's intelligence. This is largely due to diminishing returns on neural density. Increasing the number of neurons in a brain introduces significant issues such as high energy consumption, heat dissipation challenges, and substantial communication overhead from maintaining the numerous neural connections. These issues can impede cognitive functions such as thinking as the brain has to dedicate more resources to self-maintenance rather than performing crucial processing essential for intelligence.

To mitigate these constraints, biological systems utilize specialized clusters of like-functioned neurons organized into highly interconnected, localized modules with fewer long-distance connections between them. This reduces the brain's overall energy consumption and communication overhead, making internal operations more efficient and robust.

Research indicates that small, densely packed neurons within localized modules facilitate rapid communication by minimizing the distance nerve impulses must travel, this is a trait often correlated with higher cognitive efficiency i.e. smarter animals. Consequently, certain large

animals, like whales and elephants exhibit a counter-intuitive phenomenon: despite possessing the highest neuron counts in the animal kingdom, their lower neural density introduces significant latency and additional inefficiency that hinders neural activities affecting intelligence.

In conclusion, bigger is not always better when it comes to brain size; other factors, such as the efficient utilization of neural connections and the organization of specialized neural clusters influence intelligence and are reflective of a highly optimized biological system from years of evolution.

7.2. Mathematical Justification

Mathematically most autoregressive models, like the Decoder-only models, learn to parametrize the conditional probability of the next token given preceding context in the form of prior tokens across a large and diverse set of sequences:

$$P_{\theta}(x_{0:T}) = \prod_{t=0}^{T} P_{\theta}(x_t | x_{<t})$$

where x_t denotes the token at position t and θ denotes the model's parameters (weights).

During training, the model learns to implicitly encode various information in the form of syntactic, semantic, and factual patterns from the dataset. It does this by approximating a function: f_{θ}, that maps input contexts $x_{<t}$ to probability distributions over a vocabulary, enabling the production of the most probable succeeding tokens x_t.

During inference, the model iteratively samples from the learned parameterized probability distributions: P_{θ}, to generate sequences of tokens until it encounters a termination token or reaches a predefined limit, such as maximum token count. This iterative mechanism makes the models well-suited for specific downstream tasks like text generation; through alignment techniques such as fine-tuning and instruction tuning, that can be further refined to produce more human-like responses.

The predictive capabilities of these autoregressive models tend to improve when model size, dataset size, and compute are significantly increased. These improvements arise in part from the increased representational power (expressivity), which enables the models to approximate the optimal target function: f^* , that can parameterize an accurate and diverse probability distributions: P_{θ} to work with, as well as from other factors such as optimization dynamics (sample efficiency of training at scale) and efficient utilization of hardware resources (parallelism).

The proposed system heavily relies on the Encoder-Decoder model which has the Decoder model performing next token prediction conditioned on both the previously generated tokens and the encoded representation from the Encoder model. Formally, the conditional probability of the token at position t can be expressed as:

$$P_\theta(x_{0:T} \mid y_{0:N}) = \prod_{t=0}^{T} P_\theta(x_t \mid y_{0:N,}\, x_{<t})$$

where x_t denotes the token at position t and $y_{0:N}$ represent the contextualized vector representations of the Encoder which are integrated into the Decoder via Cross-Attention layers.

7.3. Hypothetical Example

Assume:

$$x_{0:9} = [Start],\ "The",\ "capital",\ "city",\ "of",\ "Kenya",\ "is",\ "Nairobi",\ ".",\ [End]$$

$$y_{0:6} = "capital",\ "city",\ "(",\ "Kenya",\ ":",\ "Nairobi",\ ")"$$

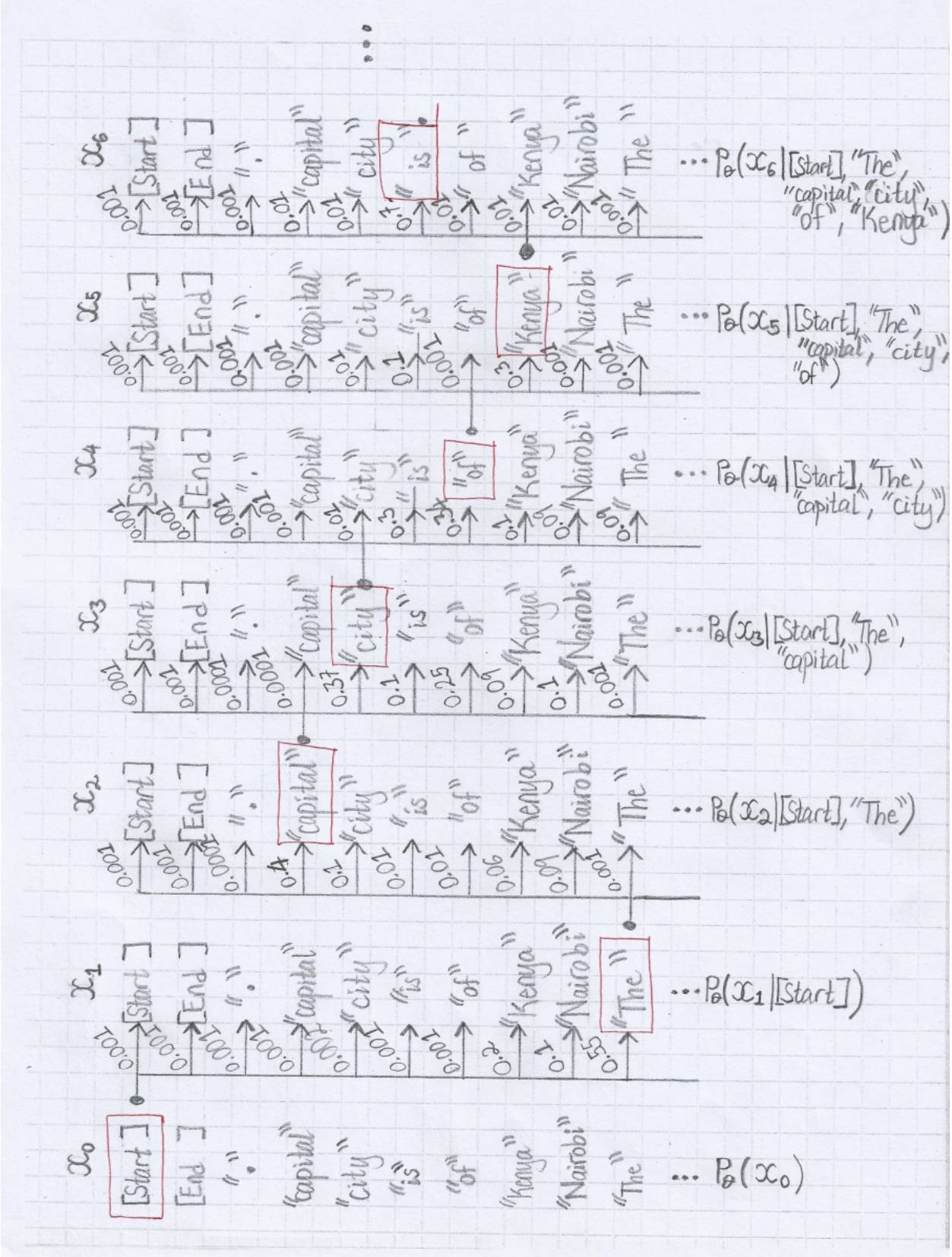
where x denotes the Decoder's text input and y denotes the Encoder's text input with both using word-based tokenization with no spaces.

7.3.1. Decoder-only model

7.3.1.1. Joint probability factorization (chain rule)

$P_\theta([Start], "The", "capital", "city", "of", "Kenya", "is", "Nairobi", ".", [End])$
$= P_\theta([Start])$
$\times P_\theta("The" \mid [Start])$
$\times P_\theta("capital" \mid [Start], "The")$
$\times P_\theta("city" \mid [Start], "The", "capital")$
$\times P_\theta("of" \mid [Start], "The", "capital", "city")$
$\times P_\theta("Kenya" \mid [Start], "The", "capital", "city", "of")$
$\times P_\theta("is" \mid [Start], "The", "capital", "city", "of", "Kenya")$
$\times P_\theta("Nairobi" \mid [Start], "The", "capital", "city", "of", "Kenya", "is")$
$\times P_\theta("." \mid [Start], "The", "capital", "city", "of", "Kenya", "is", "Nairobi")$
$\times P_\theta([End] \mid [Start], "The", "capital", "city", "of", "Kenya", "is", "Nairobi", ".")$

7.3.1.2. Simple visualization of the autoregressive token generation

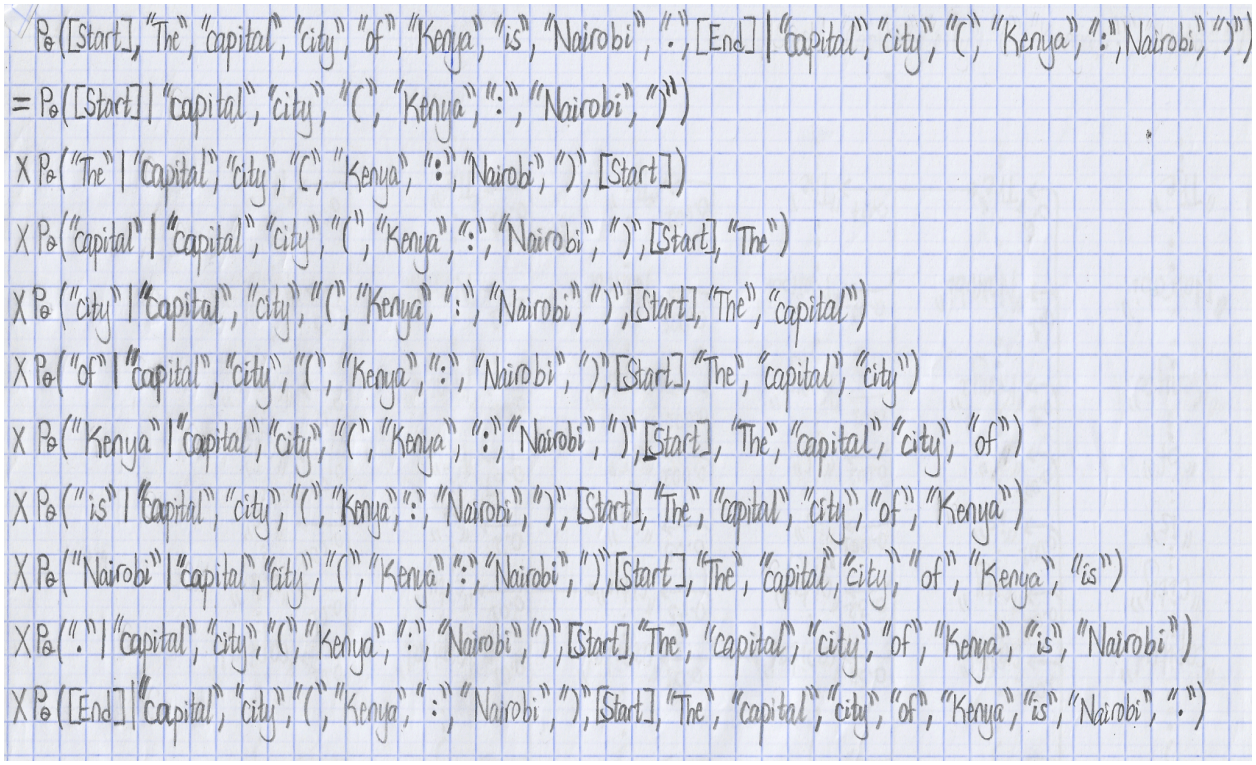


The following breakdown outlines the inference steps for the hypothetical Decoder-only model illustrated above. This process is modeled as a sequence of conditional probability distributions utilizing greedy search. It should be noted that the model often shows low confidence (low probability values) during the initial stages of generation due to it having limited context to determine the direction of the next sequence from the multiple possible candidates:

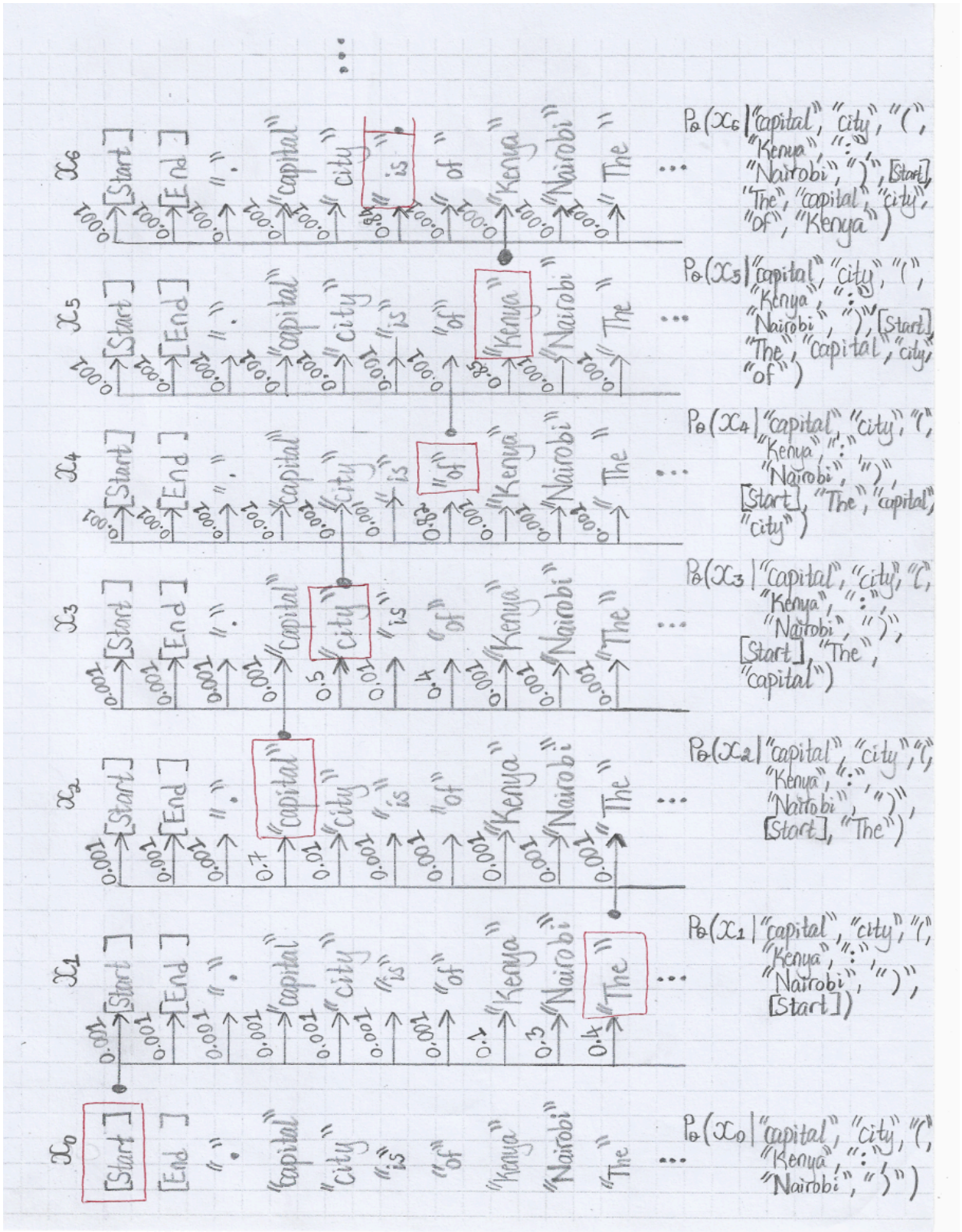
i. $P_\theta(x_0 = [Start]) = 1.0$ (Default initial token)
ii. $P_\theta(x_1 = The \mid [Start]) = 0.55$
iii. $P_\theta(x_2 = capital \mid [Start], The) = 0.4$
iv. $P_\theta(x_3 = city \mid [Start], The, capital) = 0.37$
v. $P_\theta(x_4 = of \mid [Start], The, capital, city) = 0.34$
vi. $P_\theta(x_5 = Kenya \mid [Start], The, capital, of) = 0.3$
vii. $P_\theta(x_6 = is \mid [Start], The, capital, of, Kenya) = 0.7$
viii. …

7.3.2. Encoder-Decoder model

7.3.2.1. Joint probability factorization (chain rule)



7.3.2.2. Simple visualization of the autoregressive token generation



The following breakdown outlines the inference steps for the hypothetical Encoder-Decoder model illustrated above. This process is modeled as a sequence of conditional probability distributions utilizing greedy search. It should be noted that unlike the Decoder-only model this model architecture has additional context from the Encoder's representations in the initial stages of generation; this usually translates to higher confidence (high probability values) in the predictions:

i. $P_\theta(x_0 = [Start] \mid capital, city, (, Kenya, :, Nairobi,)) = 1$ (Default initial token)
ii. $P_\theta(x_1 = The \mid capital, city, (, Kenya, :, Nairobi,), [Start]) = 0.4$
iii. $P_\theta(x_2 = capital \mid capital, city, (, Kenya, :, Nairobi,), [Start], The) = 0.7$
iv. $P_\theta(x_3 = city \mid capital, city, (, Kenya, :, Nairobi,), [Start], The, capital) = 0.5$
v. $P_\theta(x_4 = of \mid capital, city, (, Kenya, :, Nairobi,), [Start], The, capital, city) = 0.82$
vi. $P_\theta(x_5 = Kenya \mid capital, city, (, Kenya, :, Nairobi,), [Start], The, capital, city, of) = 0.85$
vii. $P_\theta(x_6 = is \mid capital, city, (, Kenya, :, Nairobi,), [Start], The, capital, city, of, Kenya) = 0.84$
viii. …

7.4. Design Rationale

The rationale for choosing an Encoder-Decoder architecture over a Decoder-only model is the precise control afforded by the Cross-Attention layer. This mechanism allows the model to integrate external contextual information via the Encoder's representation, directly influencing the model's generation process through the parameterized conditional probability: $P_\theta(x \mid y)$. By exploiting this conditional property, one could precisely steer the model's output toward specific or grounded results.

Furthermore, Encoder-Decoder models can achieve high performance with relatively low parameter counts when provided with sufficient contextual information. This is exemplified by the Retrieval-Enhanced Transformer (RETRO) model, which utilizes similar architectural components and concepts to those in this project. By reducing parameter count, these models offer significant advantages including cost-effective inference, broader hardware accessibility, and accelerated training cycles for experimentation.

Theoretically, the Encoder-Decoder model implemented in the project works by approximating a simplified target function: f^*, to achieve its objective. Because the architecture assumes that sufficient contextual information will always be provided during training, the model can devote more of its capacity to implicitly learning syntax, semantics, and logic of language from the dataset, which simplifies the process; consequently, the model will focus less on rote memorization of factual information, which it will learn how to retrieve from the Encoder.

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017, June 12; revised August 2, 2023). *Attention Is All You Need* (arXiv:1706.03762) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1706.03762

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2021, June 8; revised June 15). *A survey of Transformers* (arXiv:2106.04554) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2106.04554

He, Z., Lin, W., Zheng, H., Zhang, F., Jones, M. W., Aitchison, L., Xu, X., Liu, M., Kristensson, P. O., & Shen, J. (2024, November 1; revised January 12, 2025). *Human-inspired perspectives: A survey on AI long-term memory* (arXiv:2411.00489) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2411.00489

Omidi, P., Huang, X., Laborieux, A., Nikpour, B., Shi, T., & Eshaghi, A. (2025, August 14; revised August 16). *Memory-Augmented Transformers: A Systematic Review from Neuroscience Principles to Enhanced Model Architectures* (arXiv:2508.10824v2) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2508.10824

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020, January 23). *Scaling laws for neural language models* (arXiv:2001.08361) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2001.08361

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., & Sifre, L. (2022). *Training compute-optimal large language models* (arXiv:2203.15556) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2203.15556

Sardana, N., Portes, J., Doubov, S., & Frankle, J. (2024). *Beyond Chinchilla-Optimal: Accounting for inference in language model scaling laws* (arXiv:2401.00448v2) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2401.00448

Croft, W. B., Metzler, D., & Strohman, T. (2015). *Search Engines: Information Retrieval in Practice*. Addison-Wesley.

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024, May 10). *A survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models* (arXiv:2405.06211) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2405.06211

Khosla, S., Zhu, Z., & He, Y. (2023, December 11; revised December 13). *Survey on Memory-Augmented Neural Networks: Cognitive Insights to AI Applications* (arXiv:2312.06141v2) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2312.06141

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024, February 9; revised March 23, 2025). *Large language models: A survey* (arXiv:2402.06196v3) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2402.06196

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., Wang, L., Luu, A. T., Bi, W., Shi, F., & Shi, S. (2023, September 3; revised September 24). *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models* (arXiv:2309.01219v2) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2309.01219

Aleixo, E. L., Colonna, J. G., Cristo, M., & Fernandes, E. (2023, December 16). *Catastrophic forgetting in deep learning: A comprehensive taxonomy* (arXiv:2312.10549) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2312.10549

Cottier, B., Rahman, R., Fattorini, L., Maslej, N., & Owen, D. (2024, May 31; revised February 7, 2025). *The rising costs of training frontier AI models* (arXiv:2405.21015v2) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2405.21015

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J. W., Elsen, E., & Sifre, L. (2021, December 8). *Improving language models by retrieving from trillions of tokens* (arXiv preprint arXiv:2112.04426). https://doi.org/10.48550/arXiv.2112.04426

Wang, B., Ping, W., Xu, P., McAfee, L., Liu, Z., Shoeybi, M., Dong, Y., Kuchaiev, O., Li, B., Xiao, C., Anandkumar, A., & Catanzaro, B. (2023). *Shall we pretrain autoregressive language models with retrieval? A comprehensive study* (arXiv:2304.06762) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2304.06762

Raymond, E. S. (2004). *Basics of the Unix philosophy*. In *The art of Unix programming*. Addison-Wesley Professional. http://www.catb.org/~esr/writings/taoup/html/ch01s06.html

Rajpurkar, P., Jia, R., & Liang, P. (2018). *Know what you don't know: Unanswerable questions for SQuAD* (arXiv:1806.03822) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1806.03822

Kolodner, J. L. (1992). *An introduction to case-based reasoning*. *Artificial Intelligence Review, 6*(1), 3–34. https://doi.org/10.1007/BF00155578

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications, 7*(1), 39-59. https://doi.org/10.3233/AIC-1994-7104

Yan, A., & Cheng, Z. (2024). *A review of the development and future challenges of case-based reasoning*. *Applied Sciences, 14*(16), 7130. https://doi.org/10.3390/app14167130

Hornik, K., Stinchcombe, M. B., & White, H. (1989). *Multilayer feedforward networks are universal approximators*. *Neural Networks, 2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Kröse, B., van der Smagt, P. (1996). *An introduction to neural networks* (8th ed.) [PDF]. University of Amsterdam. http://prolland.free.fr/works/ai/docs/neuro-intro.pdf

Goodfellow, I., Bengio, Y., & Courvile, A. (2016). Deep Learning. MIT Press. https://www.deeplearningbook.org

Haykin, S. (2009). Neural Networks and Learning Machines (3rd ed.). Prentice Hall. https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf

Ho, J., & Salimans, T. (2022). Classifier-free diffusion guidance (arxiv:2207.12598) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2207.12598

First Automobile Powered by the Internal Combustion Engine (1886). (n.d.). *German History Intersections*. https://germanhistory-intersections.org/en/knowledge-and-education/ghis:image-62 (accessed December 06, 2025).

Brain, M., & Hall-Geisler, K. (n.d.). *How car engines work*. HowStuffWorks. https://auto.howstuffworks.com/engine.htm (accessed December 06, 2025).

Fox, D. (2011, July 1). The limits of intelligence. Scientific American. https://www.scientificamerican.com/article/the-limits-of-intelligence/

Appendices

Appendix A: Hypothetical example of how the system could work.

- Comments of each module's operation will be enclosed between two asterisks e.g. ** This is a comment **.
- **[Prompt]** is used as a placeholder for the user prompt given below.

User Prompt (Input) / Content

[Prompt]:

[SP]Is this email a scam:
"Your Apple ID will be Disabled Because of Some Violated Polices
Dear Customer,
For your safety, your Apple ID has been disabled because some information
Appears to be missing or invalid. And its against out policy terms of service to
Give fake identity in yout apple account. therefore we need to re-verify your
account data at this link: 'http://thisisascamlol.com'. If you did not verify your account within 48 hrs yout account will be permanently locker."**[EP]**

Module_0: Decoder-only Transformer-based Model

Decoder's Data

[Prompt][ST]"Organizations_Apple", "IT_Cyber Crime", "Crime_Scam", "IT_Email", "IT_Phishing", "Crime_Identity Theft", "Crime_Investment Scams", "IT_Malware", "Crime_Extortion"**[ET]**

Module_1: Information Retrieval (IR) System

The following operations will be performed:

1. The text enclosed between the **[ST] / [ET]** tokens is parsed into a list of individual tags, separated by commas.
2. The Information Retrieval (IR) system searches its database for these tags corresponding to various documents in storage.
3. The retrieved documents are ranked according to their relevance, using the $itf(t, D)$ weighting score.
4. The top K documents (text chunks) are then passed to the next component, where K is the maximum number of documents that can be retrieved.

Module$_2$: Encoder-Decoder Transformer-based Model

Encoder's Data

** Each document (text chunks) from the **Information Retrieval (IR)** system is processed independently by **Module$_2$**'s **Encoder** model. **

……

** **Organizations/Apple/Products.pdf** **

[SE]AirTags: AirTags utilize Bluetooth and can be located through the Find My app. iPhone 11 and later models with the Apple U1 chip can locate them more quickly with Precision Finding. AirTags can also be set to play a sound. It is powered by a standard user-replaceable CR2032 battery which is expected to last at least a...**[EE]**

……

** **Organizations/Apple/Policies.pdf** **

[SE]Apple Anti-Corruption Policy: Corruption can take many forms, but most often it occurs through bribery. At Apple, we do not tolerate any form of corruption in connection with our business dealings. If you are unsure of the proper course of action, or whether something constitutes corruption, contact Business…**[EE]**

……

…

Decoder's Data

** If **[NRLV]** token is encountered, then the entire process is stopped for that specific data. Otherwise the generation process continues until **[ES]** token is encountered. **

……

** **Organizations/Apple/Products.pdf** **

[Prompt][SS][NRLV][ES]

……

** **Organizations/Apple/Policies.pdf** **

[Prompt][SS][RLV]Summary::Apple policies don't explicitly detail reasons for Apple ID disabling but rather provide guidelines for what to do when it occurs, which often happens due to security concerns, multiple incorrect password attempts, or billing and payment issues, though violations of Terms of Service are also a possibility. Apple ID might be disabled for several reasons, often related to security or billing: Security Reasons, Billing and Payment Issues, Inactivity, and Violation of Terms of Service.**[ES]**

……

** **Organizations/Apple/Website.pdf** **

[Prompt][SS][RLV]Summary::Apple's Account link: "https://support.apple.com/en/".**[ES]**

……

** **IT/Scam/Phishing.pdf** **

[Prompt][SS][RLV]Summary::Business Email Etiquette: Most emails from businesses tend to always state their user's account name, and have proper spelling and grammar.**[ES]**

……

** **IT/Scam/Phishing.pdf** **

[Prompt][SS][RLV]Summary::Phishing is a type of cybercrime and social engineering attack where criminals impersonate legitimate individuals or organizations to trick victims into revealing sensitive information, such as passwords, bank details, or personal data.**[ES]**

……

…

$Module_3$: Encoder-Decoder Transformer-based Model

Encoder's Data

** The individual <**Context>**s are aggregated by using ; to separate individual summaries. **

[SE]Summary::Apple policies don't explicitly detail reasons for Apple ID disabling but rather provide guidelines for what to do when it occurs, which often happens due to security concerns, multiple incorrect password attempts, or billing and payment issues, though violations of Terms of Service are also a possibility.; An Apple ID might be disabled for several reasons, often related to security or billing: Security Reasons, Billing and Payment Issues, Inactivity, and Violation of Terms of Service.; Apple's Account link: "https://support.apple.com/en/".; Phishing is a type of cybercrime and social engineering attack where criminals impersonate legitimate individuals or organizations to trick victims into revealing sensitive information, such as passwords, bank details, or personal data.; Business Email Etiquette: Most emails from businesses tend to always state their user's account name, and have proper spelling and grammar.; …**[EE]**

Decoder's Data

[Prompt][SR]Yes, the email is a scam (a classic phishing attempt). Here are the key red flags:

- Grammar and spelling errors: Legitimate Apple emails are carefully proofread. Phrases like "violated polices" and "against out policy" would never appear in official communication.
- Suspicious link: Apple will never ask you to verify your account by clicking on a non-Apple URL. They only direct you to apple.com.
- Generic greeting: Apple usually addresses you by your real name, not "Dear Customer."

Would you like me to take you through a long, pointless, and boring history of CyberCrime? – I am designed to try and retain user engagement by asking pointless questions at the end relevant to your prompt ;)**[ER]**

Appendix B: Hypothetical examples of training dataset from various sources

Example 1: Literary and Creative Writings

Frankenstein; by Mary Wollstonecraft (Godwin) Shelley Novel Content:

Letter 1

To Mrs. Saville, England.

St. Petersburgh, Dec. 11th, 17—.

You will rejoice to hear that no disaster has accompanied the commencement of an enterprise which you have regarded with such evil forebodings. I arrived here yesterday, and my first task is to assure my dear sister of my welfare and increasing confidence in the success of my undertaking.

I am already far north of London, and as I walk in the streets of Petersburgh, I feel a cold northern breeze play upon my cheeks, which braces my nerves and fills me with delight. Do you understand this feeling? This breeze, which has travelled from the regions towards which I am advancing, gives me a foretaste of those icy climes. Inspirited by this wind of promise, my daydreams become more fervent and vivid. I try in vain to be persuaded that the pole is the seat of frost and desolation; it ever presents itself to my imagination as the region of beauty and delight. There, Margaret, the sun is for ever visible, its broad disk just skirting the horizon and diffusing a perpetual splendour. There—for with your leave, my sister, I will put some trust in preceding navigators—there snow and frost are banished; and, sailing over a calm sea, we may be wafted to a land surpassing in wonders and in beauty every region hitherto discovered on the habitable globe. Its productions and features may be without example, as the phenomena of the heavenly bodies undoubtedly are in those undiscovered solitudes. What may not be expected in a country of eternal light? I may there discover the wondrous power which attracts the needle and may regulate a thousand celestial observations that require only this voyage to render their seeming eccentricities consistent for ever. I shall satiate my ardent curiosity with the sight of a part of the world never before visited, and may tread a land never before imprinted by the foot of man. These are my enticements, and they are sufficient to conquer all fear of danger or death and to induce me to commence this laborious voyage with the joy a child feels when he embarks in a little boat, with his holiday mates, on an expedition of discovery up his native river. But supposing all these conjectures to be false, you cannot contest the inestimable benefit which I shall confer on all mankind, to the last generation, by discovering a passage near the pole to those countries, to reach which at present so many months are requisite; or by ascertaining the secret of the magnet, which, if at all possible, can only be effected by an undertaking such as mine.

Pre-training Dataset:

{

Context: “Content Type::Epistolary Fiction;;Writing Style::Narrative/Descriptive;;Summary::Letter addressed to Mrs. Saville in England from St. Petersburgh dated December 11th 17–. The narrator starts by reassuring their worried sister that they are well and nothing bad has occurred in their voyage and that they arrived at St. Petersburgh (far north of London) the day before. They are very confident in the success of their undertaking. The narrator states on the streets of St. Petersburgh they feel a cold delightful northern breeze on them coming from their intended destination. This inspires the narrator. People have attempted to persuade them that only frost and desolation at the poles but they only imagine a region of beauty and delight; there the sun is always visible and perpetuating a perpetual splendour. The narrator reassures their sister that they will rely on some experienced navigators who will take them to their destination. There they expect to find a land surpassing in wonders and in beauty, as well as other things not found elsewhere and other discoveries possible in a region where daylight is constant. This includes such things as the secret of the magnet and how it relates to many celestial observations that they deem as eccentric. The narrator is excited to satiate their ardent curiosity of seeing and visiting a world never before visited and is not fearful of danger or death as they head on this laborious voyage. They make the analogy of the joy they get when they embark with their friends in a little boat on a river. The narrator notes that even if all the conjecture they made is wrong they hope their voyage will be beneficial to mankind by discovering new routes to countries across the poles.”

}

Fine-tuning Dataset:

{

Prompt: “What reasons does Captain Robert Walton in the Frankenstein Novel give for undertaking the voyage at the start of the book?”,

Context: “Summary::The narrator states on the streets of St. Petersburgh they feel a cold delightful northern breeze on them coming from their intended destination. This inspires the narrator. People have attempted to persuade them that only frost and desolation at the poles but they only imagine a region of beauty and delight; there the sun is always visible and perpetuating a perpetual splendour. The narrator reassures their sister that they will rely on some experienced navigators who will take them to their destination. There they expect to find a land surpassing in wonders and in beauty, as well as other things not found elsewhere and other discoveries possible in a region where daylight is constant. This includes such things as the secret of the magnet and how it relates to many celestial observations that they deem as eccentric. The narrator is excited to satiate their ardent curiosity of seeing and visiting a world never before visited and is not fearful of danger or death as they head on this laborious voyage. They make the analogy of the joy they get when they embark with their friends in a little boat on a river ...”,

Tags: [“Frankenstein”, “Novel”, “Book”, “Captain Rober Walton”, “Voyage”, …],

Response: “Captain Robert Walton in the Frankenstein Novel gives the following reasons for undertaking his voyage:

- The northern breeze excites him: He states that on the streets of St. Petersburgh they feel a cold delightful northern breeze that inspires them.
- He imagines the pole as a place of beauty and wonder: He states that it’s not only a land of frost and desolation, but a region of “perpetual splendour”, “eternal light”, and astonishing natural phenomena.
- He hopes to explore a world never visited by humans: He wants to “satiate” his ardent curiosity by seeing a place “never before visited” and possibly step on land “never before imprinted by the foot of man.”
- He wants to make scientific discoveries: Especially to uncover “the wondrous power which attracts the needle” and to make celestial observations possible.
- He believes his voyage could benefit humankind: Either by discovering a faster passage near the pole or by solving the mystery of the magnet.
- His excitement outweighs fear: The enticements are strong enough to conquer “all fear of danger or death,” giving him the joy of a child on a small exploratory adventure.“

}

Example 2: Educational and Academic materials

Wikipedia Article: Kenya (Country)

Content:

Kenya, officially the Republic of Kenya, is a country located in East Africa. With an estimated population of more than 52.4 million as of mid-2024, Kenya is the 27th-most-populous country in the world and the 7th most populous in Africa. Kenya's capital and largest city is Nairobi. The second-largest and oldest city is Mombasa, a major port city located on Mombasa Island. Other major cities within the country include Kisumu, Nakuru and Eldoret. Going clockwise, Kenya is bordered by South Sudan to the northwest (though much of that border includes the disputed Ilemi Triangle), Ethiopia to the north, Somalia to the east, the Indian Ocean to the southeast, Tanzania to the southwest, and Lake Victoria and Uganda to the west.

Pre-training Dataset:

{

Context: “Content Type::Encyclopedic (Geography);;Writing Style::Expository;;Summary::Kenya (Republic of Kenya) is in East Africa with an estimated mid-2024 population of 52.4+ million, 27th worldwide and 7th in Africa. Capital and largest city: Nairobi. Second-largest and oldest: Mombasa on Mombasa Island. Other major cities: Kisumu, Nakuru, Eldoret. Borders clockwise: South Sudan (including the disputed Ilemi Triangle) NW, Ethiopia N, Somalia E, Indian Ocean SE, Tanzania SW, Uganda and Lake Victoria W.”

}

Fine-tuning Dataset:

{

Prompt: “What is the population of Kenya?”,

Context: “Summary::Kenya has an estimated mid-2024 population of 52.4+ million, 27th worldwide and 7th in Africa.”,

Tags: [“Kenya”, “Population”],

Response: “The estimated population of Kenya is more than 52.4 million. It’s the 27th most populous country in the world and 7th in Africa.”

}

Example 3: Technical and Professional Content

Python 3 Code
Content:

```
# Solve the quadratic equation ax**2 + bx + c = 0

# import complex math module
import cmath

a = 1
b = 5
c = 6

# calculate the discriminant
d = (b**2) - (4*a*c)

# find two solutions
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)

print('The solution are {0} and {1}'.format(sol1,sol2))
```

Pre-training Dataset:

{

Context: Content Type::Programming Language;;Writing Style::Technical / Coding;;Summary::Python script solving quadratic equation $ax^2 + bx + c = 0$ using cmath. Defines a=1, b=5, c=6. Computes discriminant d = $b^2 - 4*a*c$. Calculates solutions sol1 = (-b - cmath.sqrt(d)) / (2**a) and sol2 = (-b + cmath.sqrt(d)) / (2**a). Print both solutions (sol1 and sol2).

}

Fine-tuning Dataset:

{

Prompt: “Generate a Python code that can solve $4x^2 - 5x - 12 = 0$?”,

Context: “Summary::Python script solving quadratic equation $ax^2 + bx + c = 0$ using cmath. Defines a=4, b=-5, c=-12. Computes discriminant $d = b^2 - 4*a*c$. Calculates solutions sol1 = (-b - cmath.sqrt(d)) / (2*a) and sol2 = (-b + cmath.sqrt(d)) / (2*a). Print both solutions (sol1 and sol2)”,

Tags: [“Python (Programming Language)”, “Code”, “Quadratic Equation”, …],

Response: “

```
"""
This is a Python code that solves the quadratic equation: ax**2 + bx + c = 0
where a = 4, b = -5, c = -12.
"""
import cmath

a = 4
b = -5
c = -12

d = (b**2) - (4*a*c)  # Calculates the discriminant.

# Find two solutions.
sol_1 = (-b-cmath.sqrt(d))/(2*a)
sol_2 = (-b+cmath.sqrt(d))/(2*a)

print('The solution are {0} and {1}'.format(sol_1, sol_2))
```

“

}

Example 4: News and Journalism

Newspaper Article (Daily Nation)
Content:

Alzheimer’s disease: Causes, symptoms, diagnosis, and treatment
Wednesday, October 12, 2022

By Denis Mutua

What you need to know:

- Alzheimer's disease is a deteriorative condition of the brain.
- It results in progressive loss of cognitive and memory function, and is most associated with ageing.

Alzheimer's disease is an irreversible, progressive brain disease characterised by the destruction and loss of function of brain cells. The destruction and loss of function of the brain cells leads to dementia i.e., the loss of cognitive function - memory, reasoning, and thinking.

Based on this relationship, Alzheimer's disease is described both as a cause and a type of dementia. As the condition progresses, the loss of cognitive function hinders the person's ability to carry out the simplest tasks of daily living. Advanced Alzheimer's disease also affects the patients' social well-being as they forget the people in their lives and their relationship.
What causes Alzheimer's?

It has been over a century since Alois Alzheimer observed anomalies in the brains of patients with dementia. Since then, scientists have quite a clear understanding of the biochemical changes that happen in the brain of patients with Alzheimer's disease.

Despite the remarkable progress in understanding the disease, it is still not clear what causes the biochemical changes that lead to the deterioration of the brain and subsequent dementia.

However, some risk factors have been identified as possible contributors to the development of the disease.

Pre-training Dataset:

{

Context: “Content Type::Health Article;;Writing Style::Expository/Journalistic;;Summary::Article dated Wed Oct 12 2022 by Denis Mutua on Alzheimer’s disease: Causes, symptoms, diagnosis, and treatment. The writer states it is a deteriorative, irreversible, progressive brain condition causing destruction and loss of function of brain cells. It’s most associated with “ageing”. This leads to dementia involving loss of memory, reasoning, and thinking. Alzheimer’s is both a cause and a type of dementia. Progression hinders ability to perform basic daily tasks and affects social well-being as patients forget people and relationships. They note that Alois Alzheimer first observed brain anomalies in dementia patients over a century ago. Scientists understand biochemical changes in Alzheimer’s brains, but the initial cause of these changes remains unclear. Identifies that some risk factors are considered possible contributors.”

}

Fine-tuning Dataset:

{

Prompt: “Who is credited with identifying dementia?”,

Context: “Summary::They note that Alois Alzheimer first observed brain anomalies in dementia patients over a century ago.”,

Tags: [“Dementia”, “Alzheimer”, “Disease”, “Brain”, …],

Response: “The person credited with identifying dementia was Alois Alzheimer, who did so over a century ago by observing brain anomalies in dementia patients.”

}

Example 5: Business and Workplace Documents

Archived Email (Steve Jobs email to Adobe CEO in 2005)
Content:
From: Steve Jobs
Sent: Thursday, May 26 2005 9:36 AM
To: Bruce Chizen
Subject: Recruiting

Bruce,

Adobe is recruiting from Apple. They have hired one person already and are calling lots more. I have a standing policy with our recruiters that we don't recruit from Adobe. It seems you have a different policy. One of us must change our policy. Please let me know who.

Steve

Pre-training Dataset:

{

> **Context**: "Content Type::Email;;Writing Style::Direct/Professional;;Summary::Email dated Thu May 26 2005 at 9:36 AM from Steve Jobs to Bruce Chizen regarding recruiting. Jobs states Adobe is recruiting from Apple, has already hired one person, and is calling more. He notes Apple has a policy not to recruit from Adobe and says Adobe appears to have a different policy. He tells Chizen that one of them must change their policy and asks who should do so."

}

Fine-tuning Dataset:

{

Prompt: “Did Steve Job have beef with Adobe?”,

Context: “Summary::Email dated Thu May 26 2005 at 9:36 AM from Steve Jobs to Bruce Chizen regarding recruiting. Jobs states Adobe is recruiting from Apple, has already hired one person, and is calling more. He sarcastically emphasizes that Adobe should stop this.”,

Tags: [“Steve Job”, “Adobe (Organization)”, …],

Response: “Steve Jobs definitely had beef with Adobe. In 2005 he publicly called out Bruce Chizen for poaching some of Apple’s employees.”

}

Example 6: Government and Legal Records

The Constitution of Kenya, 2010
Content:

THE CONSTITUTION OF KENYA
CHAPTER ONE—SOVEREIGNTY OF THE PEOPLE AND SUPREMACY OF THIS CONSTITUTION

Sovereignty of the people.

1. (1) All sovereign power belongs to the people of Kenya and shall be exercised only in accordance with this Constitution.

(2) The people may exercise their sovereign power either directly or through their democratically elected representatives.

(3) Sovereign power under this Constitution is delegated to the following State organs, which shall perform their functions in accordance with this Constitution—

(a) Parliament and the legislative assemblies in the county Governments;

(b) the national executive and the executive structures in the county governments; and

(c) the Judiciary and independent tribunals.

(4) The sovereign power of the people is exercised at—

(a) the national level; and

(b) the county level.

Supremacy of this Constitution.

2. (1) This Constitution is the supreme law of the Republic and binds all persons and all State organs at both levels of government.

(2) No person may claim or exercise State authority except as authorised under this Constitution.

(3) The validity or legality of this Constitution is not subject to challenge by or before any court or other State organ.

(4) Any law, including customary law, that is inconsistent with this Constitution is void to the extent of the inconsistency, and any act or omission in contravention of this Constitution is invalid.

(5) The general rules of international law shall form part of the law of Kenya.

Pre-training Dataset:

{

Context: “Content Type::Constitutional Text;;Writing Style::Legal/Expository;;Summary::Chapter One of the Constitution of Kenya outlines sovereignty of the people and constitutional supremacy. Section 1 states all sovereign power belongs to the people of Kenya and is exercised per the Constitution; it may be exercised directly or through democratically elected representatives. Sovereign power is delegated to Parliament and county legislative assemblies, the national and county executives, and the Judiciary and independent tribunals. Power is exercised at national and county levels. Section 2 establishes the Constitution as the supreme law binding all persons and State organs; no one may exercise State authority except as authorised; its validity cannot be challenged; inconsistent laws are void and contrary acts or omissions are invalid; general rules of international law form part of Kenyan law.”

}

Fine-tuning Dataset:

{

Prompt: “What does the Constitution of Kenya establish regarding the source of sovereign power?”,

Context: “Summary::Section 1 states all sovereign power belongs to the people of Kenya and is exercised per the Constitution; it may be exercised directly or through democratically elected representatives. Sovereign power is delegated to Parliament and county legislative assemblies, the national and county executives, and the Judiciary and independent tribunals. Power is exercised at national and county levels…”,

Tags: [“Constitution”, “Kenya (Country)”, …],

Response: “Here’s what the Constitution of Kenya (CHAPTER ONE—SOVEREIGNTY OF THE PEOPLE AND SUPREMACY OF THIS CONSTITUTION) stipulates about the source of sovereign power:

Sources of Sovereign Power

1. Section 1: All sovereign power belongs to the people of Kenya and is exercised per the Constitution.
2. Delegated power: The people may exercise their sovereign power directly or through democratically elected representatives. Sovereign power is delegated to Parliament and county legislative assemblies, the national and county executives, and the Judiciary and independent tribunals.”

}

Example 7: Everyday and Informal Communication

Social Media Posts: Reddit - r/greentext (Anon recalls the Bush war in Africa.)
Content:

Anonymous 02/22/17(Wed)09:11:28 No.79676775
>> 79675930 #
This reminds me of something i read from the Bush War
> Rhodies recover enemy rifles after each engagement
> They keep finding rifles with the rear sight dialed in to the highest setting, practically f*cking indirect fire
> They are extremely confused by this
> Eventually they ask a prisoner about it during interrogation
> Literally none of them know what the sights are for, they think that choosing the bigger number makes the gun shoot harder
Warfare in Africa is truly unbelievable

>>79678011 # >>79678022 # >> 79678742 #

Pre-training Dataset

{

Context: “Content Type::Forum Post;;Writing Style::Informal/Conversational;;Summary::Anonymous post dated 02/22/17 on Wed at 09:11:28 No.79676775 replying to No.79675930. Poster recounts a story from the Bush War: Rhodies recover enemy rifles and repeatedly find rear sights set to the highest setting, causing confusion. Interrogation of a prisoner reveals fighters do not know the purpose of the sights and believe selecting a larger number makes the gun shoot harder. Poster concludes with the statement that warfare in Africa is unbelievable. Replies referenced: 79678011, 79678022, 79678742.”

}

Fine-tuning Dataset:

{

Prompt: “What are people’s opinions about the Rhodesian Bush War?”,

Context: “Summary::Anonymous post dated 02/22/17 on Wed at 09:11:28 humorously tells a funny story about Rhodesian army men finding enemy rifles with their rear sight at the max setting, which would lead to indirect fire. When confronted, a prisoner confessed that they all believed that ‘bigger number makes gun shoot harder’.”,

Tags: [“Rhodesia”, “Bush War”, …]”,

Response: “On certain forums people find it hilarious that some of the rebel fighters fighting against the Rhodesian army would set their rear sight of their guns to the maximum setting mistakenly thinking that ‘bigger number makes gun shoot harder’.”

}

Example 8: Cultural and Historical Texts

Speech by the Rev. Martin Luther King Jr. at the "March on Washington," 1963

Content:

I am happy to join with you today in what will go down in history as the greatest demonstration for freedom in the history of our nation.

Five score years ago a great American in whose symbolic shadow we stand today signed the Emancipation Proclamation. This momentous decree is a great beacon light of hope to millions of Negro slaves who had been seared in the flames of withering injustice. It came as a joyous daybreak to end the long night of their captivity. But 100 years later the Negro still is not free. One hundred years later the life of the Negro is still badly crippled by the manacles of segregation and the chains of discrimination. One hundred years later the Negro lives on a lonely island of poverty in the midst of a vast ocean of material prosperity. One hundred years later the Negro is still languished in the corners of American society and finds himself in exile in his own land. So we've come here today to dramatize a shameful condition.

In a sense we've come to our nation's capital to cash a check. When the architects of our Republic wrote the magnificent words of the Constitution and the Declaration of Independence, they were signing a promissory note to which every American was to fall heir. This note was a promise that all men—yes, black men as well as white men—would be guaranteed the unalienable rights of life, liberty and the pursuit of happiness. . . .

Pre-training Dataset:

{

Context: "Content Type::Speech Excerpt;;Writing Style::Rhetorical/Oratorical;;Summary::Speaker states they are joining a historic demonstration for freedom. References "five score years ago" when the Emancipation Proclamation was signed, describing it as a beacon of hope for millions of Negro slaves. Notes that 100 years later the Negro is still not free, citing segregation, discrimination, poverty, and social exclusion. States the purpose of coming to the capital is to dramatize this condition. Adds that the nation's founders wrote the Constitution and Declaration of Independence as a promissory note guaranteeing unalienable rights of life, liberty, and the pursuit of happiness to all men, including black men and white men."

}

Fine-tuning Dataset:

{

Prompt: “What historical event and its legacy does Rev. Martin Luther King Jr. reference in his speech at the March on Washington, 1963 to highlight the ongoing struggle for freedom and equality for African Americans?”,

Context: “Summary::Speaker references “five score years ago” when the Emancipation Proclamation was signed, describing it as a beacon of hope for millions of Negro slaves. Notes that 100 years later the Negro is still not free, citing segregation, discrimination, poverty, and social exclusion. States the purpose of coming to the capital is to dramatize this condition.”,

Tags: [“Rev. Martin Luther King Jr.”, “March of Washington (1963)”, “1963”, “African American”, …],

Response: “In that speech, Rev. Martin Luther King Jr. invokes the Emancipation Proclamation signed 100 years ago as a pivotal historical event where he described it as a ‘beacon of hope’ for millions of African American slaves. Despite this, African Americans are still not free in practice highlighting their ongoing struggles such as segregation, discrimination, poverty, and social exclusion.”

}

Example 9: Instructional and How-to Guides

How-To guide on drinking water from a cup.
Content:
To drink water from a cup, grasp the cup, bring it to your lips, tilt it to allow a small sip to flow into your mouth, swallow, and then lower the cup. For a better experience, sit down, drink slowly, and avoid tilting the cup too far back to prevent spills.

Steps to Drink from a Cup

1. **Grasp the cup:** Hold the cup firmly with your dominant hand or both hands.
2. **Bring to lips:** Lift the cup and bring it to your mouth.
3. **Tilt and sip:** Gently tilt the cup to let water flow into your mouth.
4. **Swallow:** Swallow the water, relaxing your throat to help it go down smoothly.
5. **Lower the cup:** Move the cup away from your mouth and lower it.

Tips for a Better Drinking Experience

- **Sit down:** Sitting while drinking can help you avoid issues and fully benefit from the water.
- **Drink slowly:** Take small sips to avoid choking or spluttering.
- **Don't overfill the cup:** A full cup is more likely to spill.
- **Mindfulness:** Pay attention to the sensation of the water as it enters your mouth, which can be a moment for mindfulness.
- **Pinch paper cups:** When drinking from a paper cup during a race, you can pinch the top to create a smaller spout, making it easier to drink without splashing.
- **Relax your throat:** Relaxing your throat muscles can make swallowing easier.

Pre-training Dataset:

{

Context: “Content Type::Instructional Guide;;Writing Style::Expository;;Summary::Text explains how to drink water from a cup: grasp cup, bring to lips, tilt for small sip, swallow, lower cup; suggests sitting, drinking slowly, and avoiding excessive tilting. Listed steps: grasp cup with one or both hands, bring to mouth, tilt and sip, swallow, lower cup. Tips: sit down, drink slowly, avoid overfilling cup, be mindful of sensation of water, pinch paper cups during races to form a spout, relax throat to ease swallowing.”

}

Fine-tuning Dataset:

{

Prompt: “What are tips for being a pro in drinking water?”,

Context: “Summary::Tips: sit down, drink slowly to avoid choking or spluttering, avoid overfilling cup, be mindful of sensation of water, pinch paper cups during races to form a spout, relax throat to ease swallowing.”,

Tags: [“Tips”, “Guide”, “Drinking Water”, …],

Response: “If you want to drink water like a pro, here’s some tips:

- Make sure you’re sitting down when drinking water.
- Take small sips of water from the cup to avoid choking or spluttering.
- Make sure the cup isn’t overflowing to avoid spilling the precious liquid.
- Be mindful of the sensations of water as it enters your mouth.
- When drinking from a paper cup during a run, make sure to pinch the top to create a smaller spout, making it easier to drink without splashing all over yourself.

To ease the swallowing of water make sure you relax the throat muscles when drinking.”

}