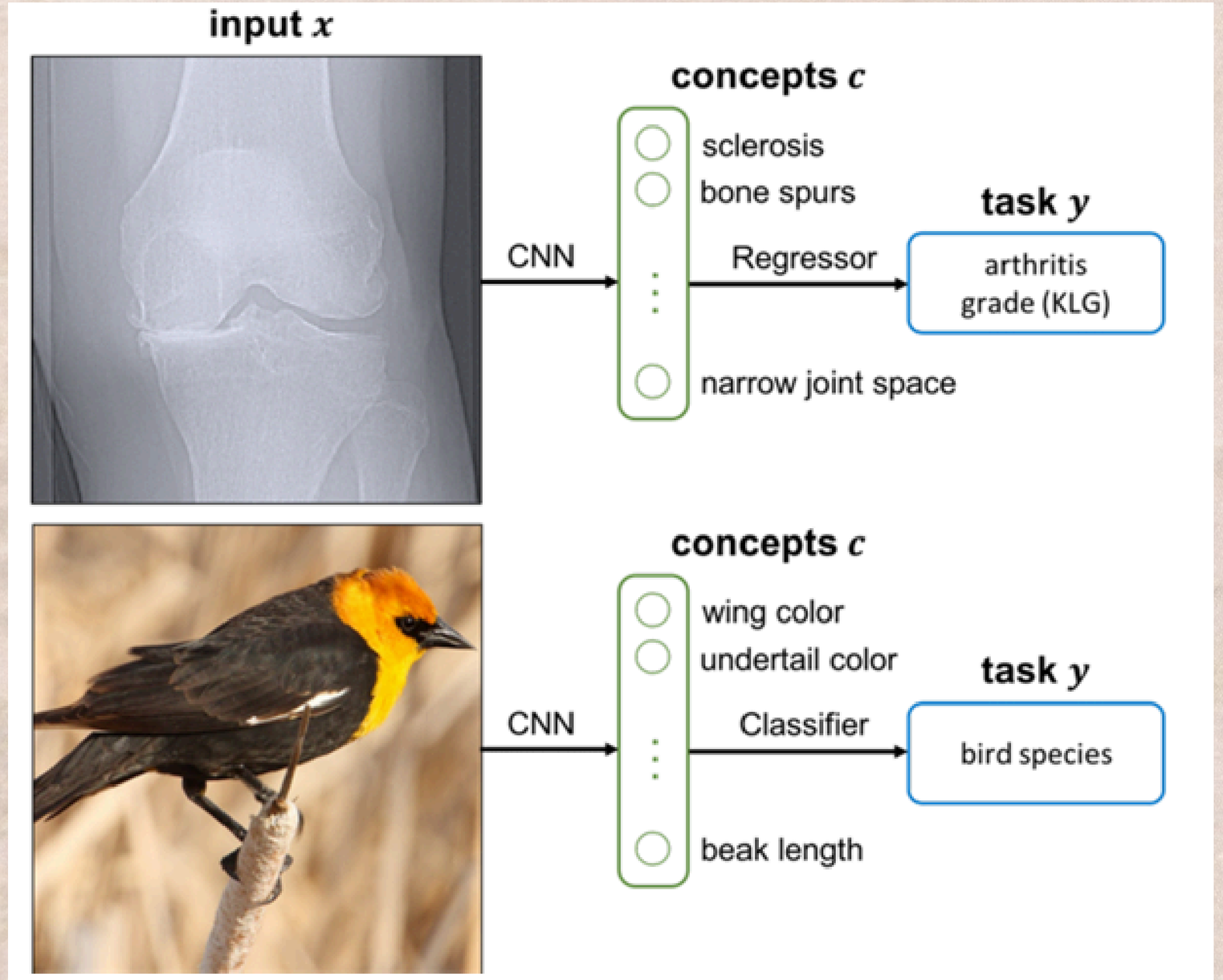
Explainable Generative AI

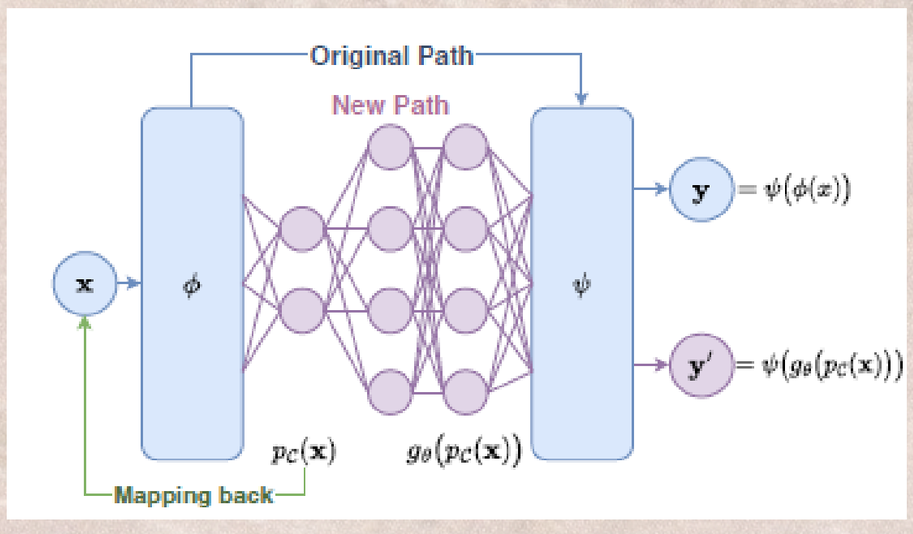
CBM ON RAG



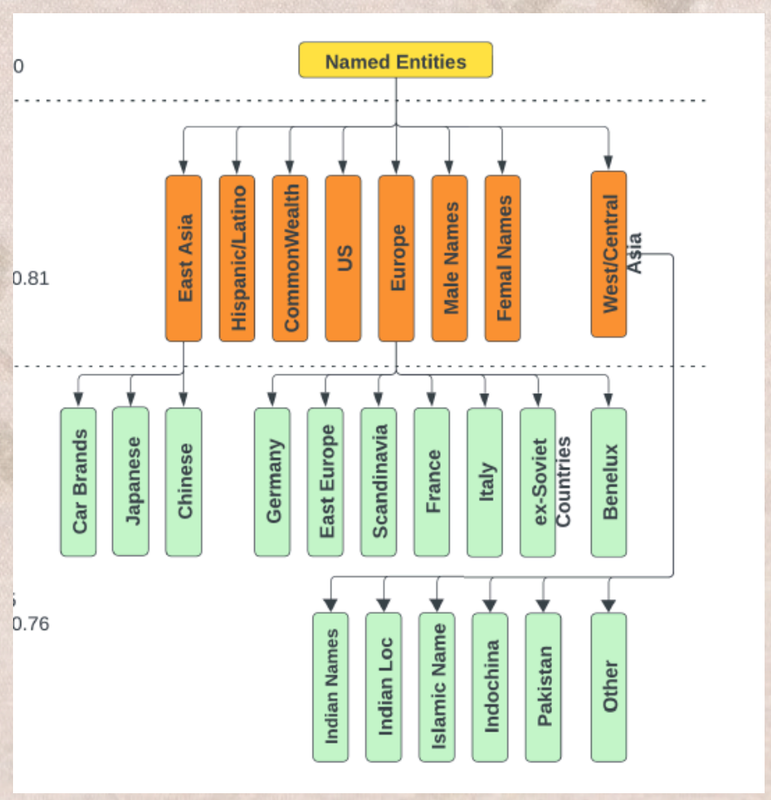
CONCEPT BOTTLENECK MODELS

Clustering the representations of embedded inputs from a specific dataset using a model with residual connections can be a powerful tool for interpretable AI, even when the model's decision process is like a black box. If the resulting clusters are human-understandable, they can make the prediction task interpretable. This can be achieved by using a set of concepts. These concepts can be generated by GPT models and then embedded, chosen by a human expert, or extracted through various clustering methods applied to a specific layer's representation. By focusing on a layer with high impact and ensuring the concepts are formed without background bias, we can approximate the model's prediction as a linear combination of these concepts. This approach will help industrialists debug their models and may even provide them with new insights relevant to their work.

CONCEPT EXTRACTION

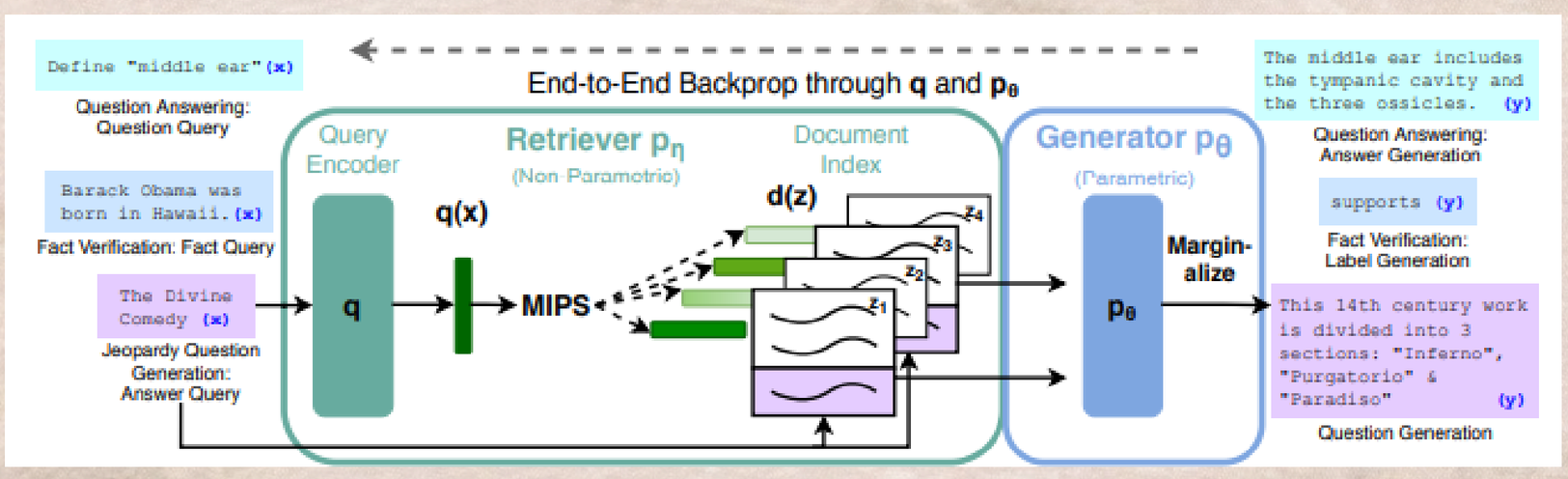


In practice, various methods are used to cluster data or separate manifolds in an embedding space. Popular techniques include K-Nearest Neighbors (KNN) and Non-Negative Matrix Factorization (NMF).



CLUSTERING ALGORITHMS FOR TEXT TOKENS, SUCH AS UNIFORM MANIFOLD APPROXIMATION AND PROJECTION (UMAP), ARE WIDELY USED. BY CONSTRUCTING FUZZY TOPOLOGICAL REPRESENTATIONS THROUGH A NEAREST NEIGHBOR DESCENT METHOD, WE CAN CREATE AN EFFECTIVE METHOD FOR FEATURE EXTRACTION.

INTERPRETABLE RAG



MAIN IDEA

The retrieved document is embedded into a latent variable Z and concatenated with the remaining token variables. This forms a latent space, which is technically considered a black box model. To debug these models or make their decision-making processes more understandable, we can use this latent space to extract meaningful concepts. Following this, a Concept Bottleneck Model (CBM) can be built after the RAG model has been made discriminative for a specific prediction task and a appropriate clustering algorithm is for text token embedding is chosen.

RETRIEVAL AUGMENTED GENERATION

RAG models utilize an input sequence xx to retrieve text documents zz, which are then used as additional context for generating the target sequence yy. The architecture consists of two main components:
(i) a retriever pη(z|x)pη(z|x), parameterized by ηη, which returns a top-K truncated distribution over text passages given the query xx; and
(ii) a generator pθ(yi|x,z,y1:i−1)pθ(yi|x,z,y1:i−1), parameterized by θθ, which generates the current token based on the original input xx, the retrieved passage zz, and the context of the previously generated tokens y1:i−1y1:i−1.
To enable end-to-end training of both the retriever and the generator, the retrieved document zz is treated as a latent variable. The retrieval component pη(z|x)pη(z|x) is based on DPR (Dense Passage Retrieval), which uses a bi-encoder architecture built on BERT.