Def:
A Reranker is an **NLP (Natural Language Processing) model or algorithm** used in various applications such as Information Retrieval, Question Answering, and Recommendation Systems to **refine and optimize the ranking of text-based search results or candidate lists.**
After an initial stage quickly retrieves a large set of potential text documents or passages relevant to a user's query, the Reranker takes this list of candidate texts. It then **leverages advanced NLP techniques to deeply analyze the semantic relevance between the query and each text, reordering them into a more precise and appropriate sequence.**

Process

Candidate Text Generation/Retrieval:

This stage quickly identifies potential text snippets (documents, paragraphs, passages, etc.) from a large text corpus that are broadly related to the user's query.

It primarily employs **keyword-based sparse models** (e.g., TF-IDF, BM25) or **text embedding-based dense models** (e.g., DPR, Sentence-BERT for vector search).NLP models at this stage prioritize speed and **high recall**, aiming to retrieve as many relevant texts as possible without missing them. However, the ranking accuracy at this stage might be relatively low.

Reranking:

This stage involves meticulously re-ordering the list of texts obtained from the candidate generation stage, using a more sophisticated **NLP model.**

The NLP models at this stage are more complex and computationally expensive, but they aim to achieve **high precision** by deeply analyzing the semantic relevance between each text candidate and the query.

The goal is to present the most relevant texts at the top of the results, maximizing the user's search experience.

Why is this important in NLP?

- **Improved Semantic Accuracy:** Initial retrieval models might only capture superficial keyword matches or simple embedding similarities. Rerankers, by employing advanced NLP techniques, understand **context, synonyms, polysemy, and complex sentence structures** within texts. This deeper understanding allows them to identify subtle semantic relationships between queries and documents, significantly boosting search accuracy.
- **Leveraging Rich Text Information:** Rerankers can utilize more comprehensive text information, such as the full content of a document, paragraph structure, headings, and metadata, to assign ranks. This helps in discerning nuanced relevance that might be missed by simple vector similarity.
- **Scalability and Efficiency:** Applying complex NLP models directly to every document in a massive text corpus is inefficient. Rerankers are applied only to a smaller subset of highly relevant text candidates filtered by the initial retrieval stage. This allows the system to maintain overall efficiency while still benefiting from the accuracy improvements offered by in-depth NLP analysis.

Types of Rerankers

- **Traditional NLP Feature-based Rerankers**
- **Deep Learning-based NLP Rerankers**

- **Traditional NLP Feature-based Rerankers**

Learning to Rank (LTR): This approach extracts **pre-defined NLP features** (e.g., text length, query term frequency, TF-IDF scores, position of query terms within the document) and then trains machine learning models (e.g., linear models, tree-based models) to predict relevance scores. This was widely used before the advent of deep learning.

- **Deep Learning-based NLP Rerankers**

With the rise of Large Language Models (LLMs), deep learning-based Rerankers have become dominant in NLP. They are much more effective at capturing complex semantic relationships in text.

Core Idea: The query and document are concatenated into a single input sequence and fed into a **single NLP model (typically a Transformer encoder)**. Within the model, **attention mechanisms** deeply model **word-level interactions** between the query and the document.

↓

Examples: Rerankers leveraging **Bidirectional Transformer-based models like BERT, RoBERTa, ELECTRA** (e.g., MonoBERT, the late interaction approach in ColBERT). The query and document are combined in a format like `[CLS] query [SEP] document [SEP]`, and the final `[CLS]` token's output from the Transformer encoder is used to predict the relevance score.

- **Advantages:** They capture highly nuanced semantic relationships and contextual matches between queries and documents, leading to **state-of-the-art search accuracy.**
- **Disadvantages:** Each query-document pair requires passing through a large NLP model, making them **computationally very expensive and slow.** They are typically applied only to a small number of top candidates (e.g., 100-200) from the initial retrieval stage.

Latest Trends in NLP Rerankers

- **Leveraging Larger LLMs:** Research is actively focused on using **larger Transformer-based Large Language Models (LLMs)** beyond BERT, such as T5 and GPT-3/4, as Rerankers to maximize their text understanding and relevance judgment capabilities.
- **Efficiency Improvements:** Given the high computational cost of LLM-based Rerankers, research into **NLP model optimization techniques** like **model distillation, quantization, and pruning** is crucial.
- **Few-shot/Zero-shot Reranking:** Exploring LLM-based methodologies like **prompt engineering** and **in-context learning** to achieve Reranking performance with limited or no labeled data.
- **Chain-of-Thought Reranking:** Moving beyond simple score prediction, LLMs are being used to provide a **"chain-of-thought"** or reasoning process explaining why a particular text is relevant to the query. This improves both explainability and potentially performance.

NLP Reranker Training Data and Performance Metrics

- **Training Data:** Rerankers are typically trained using **query-document pairs with relevance labels.** These labels can be explicit (human-annotated, e.g., MS MARCO, TREC datasets) or implicit (inferred from user behavior logs like clicks or dwell time). NLP models learn to predict relevance based on this data.

- **Performance Metrics:** The performance of NLP-based search systems is primarily evaluated using metrics that assess the ranking of text search results:
- **MRR (Mean Reciprocal Rank):** The average of the reciprocal ranks of the first relevant text found.
- **NDCG (Normalized Discounted Cumulative Gain):** A measure of ranking quality that considers the relevance of items and their position in the result list.
- **P@K (Precision@K):** The proportion of relevant texts among the top K results.