MANCHESTER
1824
The University of Manchester

Provenance Of Information in Large Language Models (LLMs) using Semantic Similarity Search

A dissertation submitted to The University of Manchester for the degree of

Master of Science in Advanced Computer Science

in the Faculty of Science and Engineering

Year of submission

2025

Student ID

10719978

School of Engineering

Contents

Word Count: 7,905

Abstract

This project investigates provenance in large language models (LLMs) using an information retrieval-based approach. While prior work on membership inference and data extraction has shown that LLM's can memorise and expose training data, these methods are largely adversarial and offer limited transparency. This study develops and evaluates a retrieval pipeline designed to link LLM outputs back to their original sources, providing an alternative provenance mechanism. The system employs three representation strategies include BM25, SPLADE and all-mpnet-base-v2 as a dense embedding model. Document chunks and chapters are indexed using Facebook artificial intelligence similarity search (FAISS), and a cross-encoder re-ranker is applied to refine ranking. After evaluating this system, the results show that dense models outperform sparse models by a lot with text chunking techniques that preserve provenance and context in data.

The findings proves that information retrieval-based pipelines are effective for provenance tracking in LLMs, offering transparency beyond adversarial methods. Overall, this project illustrates the potential of tuning a retrieval pipeline to address the legal and ownership case of copyright issues.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Acknowledgments

I express my heartfelt gratitude to my family for their unwavering support and motivation throughout my academic career. There encouragement has been an instrument in pushing me to achieve my best work.

Additionally, I extend my sincere thanks to Professor Jonathan Shapiro, my project supervisor. His guidance and expertise have been invaluable. His mentorship and guidance has helped me approach this work with all my confidence.

Lastly, I am grateful for the dedicated Professors at the University of Manchester. Their knowledge sharing and thought-provoking lectures have sparked my curiosity high, broadened my horizons, and enriched my learning experience.

1. Introduction

Large Language Models (LLMs) have proliferated in recent years, demonstrating remarkable capabilities across a range of tasks including question answering, text summarisation, and code generation. As these models are trained on vast quantities of data scraped from the internet, the provenance of their output has become a significant concern. Specifically, there are growing concerns regarding potential copyright infringement and the use of unauthorised data in their training corpora. Models such as ChatGPT, Google Gemini, and Llama can produce creative content inspired by various authors and creators, this outputted content does not convey the text of the original works verbatim but rather rephrases or summarises it, contrasts it with other text and generates a result that seems to posses a degree of originality (Riofrio Martinez-Villalba, J.C., 2024).

Tracing information within LLMs is a profound technical challenge. Unlike a traditional database, knowledge in these models is not stored in a searchable format but is instead encoded implicitly across billions of parameters during training, a concept known as parametric knowledge (Yu, H., et al, 2024). This effectively renders the model a "black box," where the internal mechanisms mapping inputs to outputs are opaque. In the context of AI auditing, access to these systems is typically categorized. Black-box access, the most common for proprietary models, permits an auditor only to submit queries and analyse the resulting outputs. In contrast, white-box access provides full transparency into the model's architecture, weights, and training data, allowing for deeper, more direct forms of analysis. Casper, S., et al, (2024) stated that recent evaluations of advanced AI systems have predominantly depended on black-box access wherein auditors can solely query the system and examine the replies. However, white box access to the systems internal operations allows auditors to perform further fine tuning of the models and allows detailed interpretation of the outputs.

While existing research has explored methods like data attribution techniques such as watermarking and influence functions, these methods often focus on the training data's effect on model behaviour rather than pinpointing the direct textual origin of a specific generated output. They can demonstrate that a data point had an influence, but not necessarily what the source text was (Koh, P.W.W., et al, 2019). To address this gap, this paper proposes and evaluates a novel framework for verifying the provenance of LLM generated text using a two step, answer to source matching methodology. First, a standalone LLM is prompted to an answer based on its parametric knowledge of a controlled corpus, in this case, the novel "To Kill a Mockingbird". Second, this generated answer is used as a query in an advanced retrieval system. This system employs semantic search, powered by dense vector embeddings and a cross-encoder re-ranker, to perform a forensic analysis and locate the most probable source passages within the original text.

1.1. Background and Motivation

This section aims to provide an overview of the thesis background and related concepts to build understanding of the methodologies and technologies used. The follow up sections will introduce notions related to the foundations of LLMs architectures, GPT models and training paradigms, information retrieval and provenance tracking, legal, ethical, and copyright considerations, and finally the motivation of the project.

1.1.1. The Foundation of LLMs Architecture

Most state-of-the-art LLMs are built on the transformer architecture, which marked a shift from the earlier natural language processing (NLP) approaches such as Recurrent Neural Networks (RNNs). The transformer architecture introduces attention-based models compared to the recurrent mechanisms used by the classic models. Schmidt, Robin M, (2019) describes the RNN as a type of neural network architecture used to uncover patterns in sequential data using Multi-Layer Perceptrons to pass information back in cycle to itself. RNNs process text one word at a time in a sequence, passing information along a chain, while effective for shorter texts, this approach struggles with two major challenges which are the vanishing gradient and the lack of parallelization.

Gradient descent is used to train deep neural networks using backpropagation which initializes parameters randomly and trains repeatedly. These parameters are responsible for computing partial derivatives for the model loss, which either vanishes or explodes as the model's depth grows due to the chain rule (Abuqaddom, I., et al. 2021). The sequential nature of RNNs means that calculation of a distant word cannot begin until the previous word has been processed, this makes training on massive datasets increasingly slow and computationally inefficient.

The transformer architecture was introduced by Vaswani, A., et al. (2017) in the paper "Attention is all you need", they proposed a self attention mechanism solving the problem of RNNs. This mechanism allows a model to weigh the influence of all words in the input sequence simultaneously, regardless of their position. Self attention enables a model to capture complex relationships between words and helps it understand context behind a given sequence of words. In a Transformer the model takes in a sequence of tokens and creates three vectors, query Q, key K, and value V. Attention is computed by comparing each query with all keys to measure their similarity. The resulting similarity scores are scaled and normalized through the SoftMax function which produces a probability distribution over the keys. If Q and K are represented as row vectors, the SoftMax output is also a row vector while values V, expressed as column vector are combined according to these weights. This yields a sum of values, producing the output representation. (Vaswani, A., et al. 2017).

The other important characteristic of the transformer is the encoder and decoder structures of the architecture. Each is usually constructed using six layers of feed-forward networks, normalization, and multi-head attention. While the decoder combines self-attention, which focuses on previously generated tokens, and encoder-decoder attention, which combines encoder outputs with decoder states to help with prediction, the encoder uses multi-head attention to model relationships within the input sequence (Raganato, A. and Tiedemann, J., 2018,).

1.1.2. Generative Pre-trained Transformer (GPT) Models and Training Paradigms

GPT is an outstanding innovation in the field of NLP It is driving researcher's efforts to create models that are capable of understanding and communicating language in ways that are remarkably similar to those of humans. GPT gained popularity following the release of ChatGPT by OpenAI, a research firm dedicated to creating artificial intelligence (AI) technologies. GPT series has progressed through successive iterations each addressing limitations of earlier NLP approaches that heavily relied on task specific labelled datasets.

Yenduri, G., et al, (2024) explained that GPT-1 was released in 2018, demonstrating the effectiveness of pre-training on large volumes of unlabelled text, using a 12-layer transformer decoder to achieve zero shot performance on many tasks. GPT-2 in 2019 expanded this approach with 1.5 billion parameters, significantly improving performance in tasks such as summarisation and translation through its ability to capture long range dependencies. GPT-3 in 2020 marked a breakthrough with 175 billion parameters trained on the Common Craw corpus enabling it to generate human-like text, perform simple coding, and carry out reasoning tasks. This model was initially updated to GPT-3.5 which included both text and code from diverse web sources. The most recent version GPT-4 is a multi model trained on public and other datasets, featuring extended context windows and reinforcement learning from human feedback. Furthermore, instruction tuning is employed to enhance a model's ability to execute user instructions and perform practical tasks. This is achieved by fine-tuning the model on a diverse dataset of tasks, which have been annotated with human-generated prompts and feedback (Peng, B., et al. 2023).

GPT is trained on a diverse corpus of text data including books, websites and other publicly available sources collected through techniques such as text mining. These sources may include copyrighted works raising concerns about unauthorized use and potential infringement.

1.1.3. Information Retrieval (IR) and Provenance Tracking

Information retrieval systems are vital to our everyday online experiences. Their core objective is to identify and return documents from a corpus that are relevant to a user's information need, typically expressed as a query. Two principal paradigms govern the architecture of these systems, their application depending on the nature of the information need, these are traditional lexical models and modern semantic models. Traditional lexical retrieval also known as sparse retrieval operates on a principle of exact keyword matching. These systems describe documents and queries as high dimensional, sparse vectors, as demonstrated by fundamental algorithms like term frequency-inverse document frequency (TF-IDF) (Sparck Jones, K., 1972) and its successor best match 25 (BM25) (Robertson, S.E. and Walker, S., 1994). Relevance is determined by the similarity of keywords, weighted by their statistical importance. TF-IDF is the common approach to compute similarity between a query and a document. This intuition is that a term is important if it appears frequently in a document but not across many documents in the collection. The relevance score R between document d and query q is therefore calculated by summing the product of the TF-IDF weights of each term t that appears in both the query and the document. This way, documents that share rare but more informative terms with the query receive higher similarity scores, while common uninformative terms contribute less (Li, X., et al, 2024). While highly efficient, these models are inherently limited by the vocabulary mismatch problem whereby they often fail to retrieve relevant documents that use different terminology to describe the same topic.

In contrast semantic retrieval is a dense retrieval which aims to overcome these limitations by matching based on contextual meaning. Li, X., et al, (2024) stated that the deep semantic information of texts is captured by dense retrieval techniques based on the bidirectional encoding representations derived from the BERT model (Devlin, J., et al, 2019), which drastically improves retrieval precision with the recent development of deep learning. The dense retrieval encodes the meaning of queries and documents into dense vector embeddings.

There three fundamental steps that an information retrieval system follows, Hiemstra, D., (2009) outlined them as the representation of document content, the representation of a user's information need, and the comparison of these representations to establish relevance. In the context of large language models (LLMs), these steps are operationalized through dense retrieval pipelines. Pre-trained transformer encoders map queries and documents into semantic vector embeddings, which are then indexed in a vector database for efficient retrieval. During inference, a user's query is embedded in the same space and compared with stored document vectors, most commonly using cosine similarity. This embedding-based approach allows retrieval to capture semantic meaning, enabling LLMs to ground their outputs in external sources rather than relying solely on memorized training data. Such retrieval-augmented generation (RAG) pipelines are particularly relevant for provenance, as they not only improve factual accuracy but also provide explicit links to the origin of retrieved content, addressing issues of transparency, trust, and copyright.

The final stage of an IR system usually consists of using a cross-encoder model to re-rank the relevant documents retrieved from the first stage of retrieval. Petrov, A.V., et al, (2024) explained that cross-encoder models which encode the query and the documents concurrently as a single textual input, are usually used to provide the best results for document re-ranking. These algorithms are known for being resilient when generalizing across retrieval tasks and domains.

1.1.4. Legal, Ethical and Copyright Considerations

Copyright concerns have become a central issue in the development of LLMs, with several companies facing lawsuits for alleged infringement. Sunstein (2024) reports that The New York Times filed a complaint against OpenAI and Microsoft, claiming extensive copyright violations. The lawsuit argues that OpenAI's models, including ChatGPT, were trained on millions of the newspaper's articles without authorization. Evidence presented included more than 100 instances where, when prompted with quotes from Times articles, ChatGPT reproduced substantial portions of the original text with only minor paraphrasing. This case highlights the growing legal and ethical debate around the use of copyrighted material in AI training, raising questions about intellectual property rights, fair use and the boundaries of permissible data usage in machine learning.

1.2 Aims and Objectives

The aim of this project is to investigate whether the provenance of LLM's can be identified by developing and evaluating an information retrieval system capable of linking model responses to their potential source documents.

1.2.1. Objective

The objective of this project is to design and implement an information retrieval system capable of indexing a representative text corpus such as, a book or article and retrieve candidates relative to a LLM generated output text while having black box access. This includes experimenting with retrieval strategies and analysing the results.

1.3 Related Work

In this section the current understanding of how the provenance of information in LLMs is being uncovered will be discussed highlighting key themes, methodologies and knowledge gaps. By critically examining the existing evidence, this section will establish a foundation for developing more effective provenance tracking systems.

1.3.1. Membership Inference Attacks (MIAs)

Membership inference attacks have been widely studied as a method to test whether specific data records were used in the training of a machine learning model. The central idea is to infer whether a given input was part of the training set by exploiting differences in model behaviour between seen and unseen data points (Shokri, R., et al., 2017). In the context of LLMs, MIA can be applied by probing the model with candidate text passages and measuring the model's likelihood or confidence scores. A high confidence response may indicate that the passage was memorized during training.

Hayes, J., et al (2017) demonstrated that adversaries might determine if a certain data point was included in a model's training set by exploiting differences in generative adversaries' networks (GANs) respond to known versus unknown inputs. Their approach involved training an attacker model in both black-box and white-box settings, showing that overfitting increases the model's confidence on training samples than. The attacker was given a dataset which was assumed to have the data points used in training of the target model. If an attacker possesses access to the true labels inside the dataset, they can train a discriminative model on these examples to accurately classify the training samples. In this case if the target model overfits the training data, the confidence value for samples in the training set increases. To improve the attacks, regularization techniques such as weight normalization, dropout, and differentially private mechanisms were employed. Finally for evaluation, naïve Euclidean distance is used by computing the distance between the generated sample and every real sample in the dataset. Experiments across three datasets revealed striking vulnerabilities with the white-box attacks reaching 100%, black-box attacks whit limited knowledge reaching 81%, while black-box attacks with no prior knowledge performed above random choice at 40% These results established MIAs as a key threat to model confidentiality. While this approach provides evidence of data leakage, it is limited to its probabilistic nature restricting accurate identification of the underlying data source used by large language models.

1.3.2. Privacy and Data Extraction

Building on the MIAs foundation, Carlini, N., et al, (2021) demonstrated that large language models can be vulnerable to data extraction attacks, where memorized training data is revealed verbatim through model outputs. Their research showed that even without overfitting, large models like GPT-2 could be prompted to generate verbatim sequences from their training data. The authors designed targeted prompts to produce sequences from the training set including sensitive information such as personally identifiable data. Their methodology involved generating a large number of samples from the model and then using various metrics such as perplexity ratio between the target model and a smaller reference model to identify outputs that were unexpectedly likely or likely memorized. Unlike membership inference which only determines whether a sample was used in training, data extraction recovers the actual text itself. Their findings highlight the present the issue of unintended memorization of data in large models and the motivating research into provenance and mitigation techniques such as differentially private training and transparency mechanisms.

1.3.3. Data Watermarking

While the previous methods are effective, proving membership is a challenging task, Wei, J.T.Z., et al, (2024) Proposed an initiative-taking method using data watermarking techniques. Instead of passively searching for memorized data, their approach allows a copyright holder to intentionally embed a unique randomly generated watermark into their data before public release. This watermark can be a random sequence of characters or subtler Unicode substitutions. The detection of this watermark in a model's output is then outlined as a hypothesis test, providing a statistical guarantee on the false detection rate. This method shifts the idea of exploratory extraction to verifiable auditing, giving data owners a principled method to check for unauthorized use of their data in LLM pretraining even with black-box access to the model.

1.3.4. Conclusion

The ability of watermarking techniques to provide verifiable provenance is robust but they still have limitation as they can be dodged through paraphrasing, and they are difficult to implement on existing models. Furthermore, membership inference and data extraction attacks illustrate that LLMs can memorize and unintentionally leak training data, both approaches are largely adversarial and retrospective, aiming to expose vulnerabilities rather than to provide a systematic solution for provenance. Moreover, these methods do not create verifiable connections between model outputs and their possible source text but rather show whether or how a model has memorized data. To link generated outputs to candidate documents, this project investigates an information retrieval-based strategy that involves indexing and querying an external corpus. The system aims to offer a transparent provenance mechanism that enhances current privacy focused research by utilizing dense retrieval and semantic similarity techniques. This allows for the detection of memorization as well as uncovering the links between generated text and source material.

2. Methodologies

This section aims to outlines the methodological framework adopted to investigate provenance in large language models through information retrieval techniques. It describes the overall research design, the datasets selected and their preprocessing, and the representation of documents and queries. The chapter then details the indexing and retrieval strategies implemented, the approach used for provenance tracking, and the evaluation metrics applied to assess system performance. Finally, the implementation tools and limitations of the methodology are discussed, providing a clear foundation for the subsequent results and analysis.

2.1. Research Design

The methodology follows a retrieval augmented model in which a corpus is processed, indexed, and queried to simulate how provenance can be identified for model responses in a black box setting. Both sparse and dense retrieval strategies are considered whereby, sparse retrieval provides a traditional baseline while dense retrieval powered by embeddings capture semantic relationships that go beyond keyword overlap. This dual design allows for comparative evaluation of retrieval effectiveness across different representations.

2.2. Data Preparation

The dataset used in this project is derived from the publicly available published book “To Kill a Mockingbird” (Lee, H., 2010), this dataset was selected as it presents a unique combination of rich thematic depth and a well-defined manageable structure. Its exploration of complex social issues provides a robust basis for thematic analysis, while its consistent narrative and distinct character voices make it an excellent benchmark for evaluating a model's understanding plot progression and long-range contextual relationships.

For consistency and interpretability, the text of the book was segmented at chapter level ensuring that provenance links could be traced back to clearly defined source units. This approach allows the retrieval system not only to identify passages but also attribute them to their broader contextual origin.

Preprocessing involved several steps to prepare the text for indexing and retrieval. First, the entire text was segmented into individual chapters, this was done by using Pythons’ regular expression library. A function from the library was used to split the document whenever it finds the pattern “Chapter” followed on or more digits, effectively creating a list where each item is the text of a chapter. After segmentation, the second step further processes each chapter’s text into smaller pieces using a custom sentence chunking function. This step is important to ensure that text fits within the context window of a language model. The function consists of several parts that does further processing, first a simple sentence tokenizer is used to intelligently split blocks of text into a list of individual meaningful sentences. The tokenizer correctly manages standard and non-standard punctuation and abbreviations in the text removing the need to manually manage them. The third step in the function is to pack the sentences from each chapter into chunks using a specified maximum number of tokens, as determined by an external model tokenizer. Chunking is important for boosting the effectiveness and accuracy of the retrieval process, as it divides large datasets into manageable and coherent units that can be efficiently indexed and retrieved. Moreover, structuring chunks around meaningful document elements such as titles, rather than relying on token lengths can enhance the retriever’s ability to return contextually relevant information (Rahul, 2024). To ensure context is not lost between chunks, a key feature to resolve this was the implementation of an overlap. When a chunk is created, the last few words are used at the beginning of the next chunk to help the model understand the relationship between consecutive pieces of text. Finally, edge case handling was implemented as a fallback for rare instances where a single sentence is longer than the maximum number of tokens allowed. This functionality forcefully split that sentence by words to ensure no piece of text is too large.

For the sparse model BM25, data was prepared differently due to its lexical limitations treating different word forms as complete unrelated words, lemmatization was used to enhance the search effectiveness. Balakrishnan, V. and Lloyd-Yemoh, E., (2014) explained that lemmatization uses vocabulary and morphological analysis of words to remove inflectional endings, thereby returning words to their dictionary form. This process ensures accuracy by analysing whether query words are used as verbs or nouns, so that the correct base form is selected. Furthermore, the text data is stripped of any regular expressions and punctuation expect apostrophes to allow the model to distinguish words such as “father’s” with “fathers”. Instead of sentence tokenizer a word tokenizer is used.

2.3. Document Representation

Once the dataset has been pre-processed and segmented into coherent chunks, document representation is next crucial step in enabling effective retrieval. This step determines how both queries and documents are encoded for comparison, in this project both sparse and dense approaches were employed to allow for comparative evaluation.

2.3.1. Best Match 25 (BM25)

In the sparse retrieval setting, the BM25 algorithm was implemented as a baseline. BM25 is a probabilistic retrieval function that extends the traditional TF-IDF framework by incorporating term frequency capacity and document length normalization. It counts how often each term appears in the document, applying saturation control to prevent frequent words from dominating the scoring function. The algorithm adjusts scores-based on document length to ensure that longer documents are not unfairly penalized or rewarded (Robertson, S. and Zaragoza, H., 2009). The final encoding produces a sparse vector in which each dimension corresponds to a vocabulary term and contains non-zero values only for the terms that exist in the document.

2.3.2. Sparse Lexical and Expansion Model (SPLADE)

To improve upon the limitations of lexical methods, sparse lexical and expansion model for document retrieval (SPLADE) (Formal, T., et al, 2021) was also considered, specifically the SPLADE coCondenser EnsembleDistil. SPLADE uses a BERT-based transformer architecture with a masked language modelling head (MLM) to create a sparse vector representation that can include terms that are not present in the original document. Documents are tokenized using BERT’s wordPiece tokenizer, creating sub word tokens from the 30,522 BERT vocabulary. These tokens are processed through multiple encoder blocks, building rich contextual representations. For each token in an input sequence, the model estimates an importance score for every other token in the entire vocabulary. To build a single representation, for the entire vocabulary or query, SPLADE first applies a log-saturation function to importance scores. For sparsity it uses the rectified linear units (ReLU) (Agarap, A.F., 2018) to set any negative scores to zero, this is a principal concept of the model’s sparsity as only terms predicted to have positive impact are retained. The model then applies weight scaling to the scores by applying a logarithm function to lessen the impact of high scoring terms. Finally, a pooling step is added to aggregate the contextualized importance scores into a single sparse vector. The resulting representation is a sparse vector where non-zero dimensions represent both original terms and semantically related expansion terms learned by the model.

2.3.3. Sentence Transformer Encoder for Dense Representation

For dense retrieval representations the project employed the sentence transformer model all-mpnet-base-v2. Built on Microsoft's masked and permuted language modelling (MPNet) architecture it was designed to overcome limitations of previous pre-training methods such as BERT's masked language modelling and XLNet's permuted language modelling (Yang, Z., et al, 2019). BERT's MLM objective masks tokens and predicts them individually, resulting in the loss of dependence information between masked words. In contrast, XLNet's catches these dependencies but suffers from a positional discrepancy because it does not look at the positional information of the tokens it is attempting to predict. MPNet integrates both these disadvantaged techniques by masking tokens simultaneously permuting input sequences during training, which allows the model to better capture dependencies across long contexts while maintaining bidirectional understanding (Song, K., et al, 2020).

For the all-mpnet-base-v2 to become a sentence transformer it is fine-tuned using the sentence-BERT methodology to generate high quality sentence embeddings. This process uses a Siamese network architecture, where two identical MPNet models process a pair of sentences. The model is trained on over a billion sentence pairs with a contrastive learning objective. Given an input query or document chunk, the model processes the tokenized sequences through multiple transformer layers, producing contextual embeddings for each token. To create a single vector for the entire input, a mean pooling operation is applied to average all the tokens and produce a final 768-dimensional (Reimers, N. and Gurevych, I., 2019). These dense vectors capture both syntactic and semantic information, ensuring that text with similar meanings are mapped closely to one another in the embedding space. In this project, both document chunks specifically paragraphs and queries were encoded into such vectors.

2.4. Facebook Artificial Intelligence Similarity Search (FAISS) Indexing

To enable efficient storage and retrieval of the dense representations, the FAISS library was employed, it is an open-source toolkit specifically designed for large scale similarity search and clustering of dense vectors. FAISS provides highly optimized implementation of approximate nearest neighbour search (ANNS) algorithms that can operate on millions of high dimensional vectors, making it particularly well suited for dense retrieval pipelines (Douze, M., et al, 2024). The core of FAISS is based on an index structure that splits and compresses the vector space, the techniques consist of two main stages coarse quantization and vector quantization. Coarse quantization involves partitioning the high dimensional vector space into a set of cells, this is achieved by selecting a number of centroids using clustering algorithms like K-means. Each document vector is assigned to the nearest centroid. This results in an inverted file structure, with each centroid pointing to a list of documents stored within its cell. This structure significantly reduces the search space as a query only needs to be compared to documents in a few relevant cells rather than the full dataset. Finally, vector quantization compresses the high dimensional vectors using a product quantizer to shrink their size and reduce high memory usage and speed up calculations (Johnson, J., et al, 2019).

In this project, both document chunks and the query representation generated by the sentence transformer model were stored in a FAISS index. The index was constructed to handle inner product and cosine similarity computations, allowing for relevance ranking based on semantic similarity between query and document embeddings. By utilizing FAISS vector compression and indexing algorithms, the system was able to achieve scalable search performance while maintaining accuracy. FAISS supports a wide range of indexing techniques, including flat indexes for nearest neighbour search and hierarchy quantization base indexes approximation search. Given the moderate size of the dataset utilized in this study, a flat index with inner product (IndexFlatIP) was used and then later normalized to convert it into cosine similarity. This conversion is possible because the inner product of two L2-normalized vectors is mathematically equivalent to their cosine similarity. This step assures that the retrieval is based solely on the vectors orientation in the semantic search, rather than their magnitude which is a conventional and reliable metric for semantic importance.

Incorporating FAISS into the retrieval pipeline enabled efficient similarity computation and ranking of candidate documents. After identifying the top-k results, provenance was retained by mapping each retrieved vector to its original metadata, which included the chapter and source identifiers.

2.5. Query Model ChatGPT

By using the OpenAI API as an environmental variable, ChatGPT 4.1 mini was selected as the base model for testing provenance. The goal was to generate text that could be analysed for its origins and its relationship to the novel “To Kill a Mockingbird”. A prompt was engineered to guild the model’s behaviour establishing is as a “creative short story write capable of answering question” focused on a specific style and knowledge base. This constraint was crucial for keeping the model’s output relevant to the novel being analysed. By instructing the model to generate text “relative to the novel”, the prompt ensures that the generated content is likely to be a candidate for the retrieval-based provenance test. The LLM’s output is used to query the novels text.

The model’s temperature parameter was set to 0.7. This is an important hyperparameter that controls the level of creativity of the models generated output. Jithin James (2023) confirmed that temperature controls the outputs diversity, it adjusts the level of randomness in the generated output by modifying the SoftMax function which oversees calculating the probability of the next word in each sequence. A lower temperature closer to 0 makes the model less creative and more focused, while a higher temperature makes the model’s choices more diverse and unpredictable. The temperature value chosen enables the model to generate varied and fascinating text rather than repeating predictable phrases. This is useful for assessing the model’s capacity to recall and synthesize information from the novel in a nonverbatim style. The chosen value ensures that the LM output was sufficiently varied to test the retrieval systems capability to retrieve relevant passages even when the output does not match the source text.

2.6. Re-ranking With a Cross-Encoder Model

Initial retrieval methods whether sparse or dense, often return a broad set of candidate documents that vary in relevance to the query. To improve the ranking quality this project employed a cross-encoder model for re-ranking. Unlike bi-encoders which independently encode queries and documents into embeddings, a cross-encoder jointly processes the query and documents together through a transformer-based architecture. This allows it to explicitly represent fine-grained interactions between tokens in the query and the candidate passages, resulting in more accurate relevance estimation (Nogueira, R. and Cho, K., 2019).

While computationally expensive than bi-encoders due to the joint encoding, they consistently achieve superior performance on benchmark datasets such as Microsoft machine reading comprehension (MS MARCO) (Nguyen, T., et al, 2019). The cross-encoder used in this project is the one trained on the MS MARCO dataset, the ms-marco-MiniLM-L-6-v2 trained for passage retrieval tasks. The dataset includes over a million anonymized queries from Bing search query records as well as over a hundred thousand human written generated answers. It includes up to nine million passages from web pages.

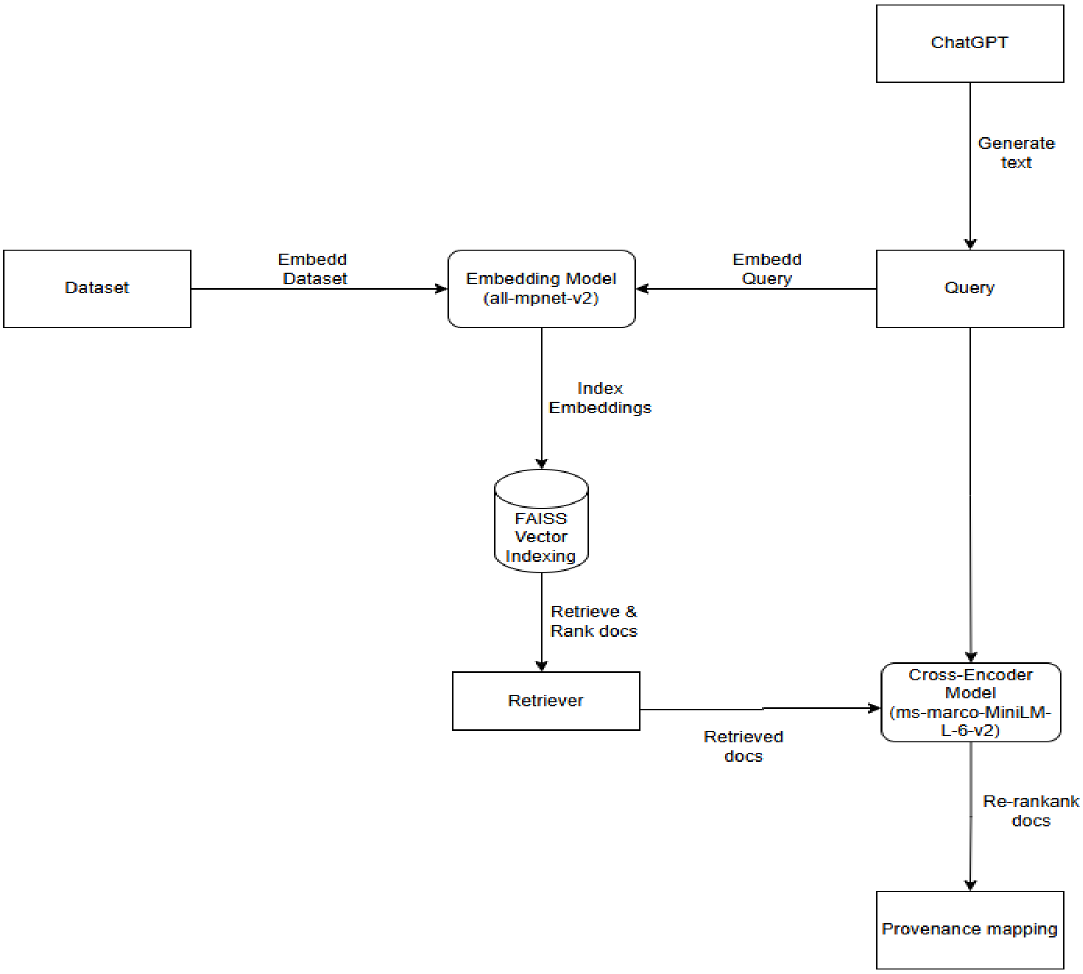

Figure 1 Dense retrieval pipeline

3. Evaluation

This chapter presents the evaluation of the proposed information retrieval pipeline for provenance tracking in large language models. The aim of this chapter is to assess the systems ability to accurately retrieve relevant passages, preserve provenance links to source documents, and compare the effectiveness of sparse and dense retrieval methods. The evaluation is structure around quantitative performance metrics, and qualitative assessment of the retrieval outputs, and critical discussion of the systems strength and limitations.

3.1. Evaluation Framework

The evaluation framework is structured around two dimensions. First being the retrieval performance of different representation strategies including BM25, SPLADE, and all-mpnet-base-v2 embeddings compared using the established information retrieval metrics such as Precision@k, Recall@k, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (nDCG). These metrics capture both accuracy of retrieved results and their ranking quality depending on contextual relevance. Second, provenance is evaluated by examining whether the retrieved results can be consistently mapped to their original chapter or document context. This dual perspective ensures that the evaluation not only measures the effectiveness of the retrieval system but also test its capabilities to return traceable and verifiable outputs.

3.2. Experimental Setup

3.2.1. Data Chunking retrieval

To evaluate the effectiveness of the proposed provenance aware information retrieval pipeline, a series of experiments were carried out using a structured text corpus segmented at chapter level. The dataset was pre-processed so that each paragraph chunk contained a maximum of 450 tokens with an overlap of 60 tokens, with each chunk linked to its corresponding chapter metadata. The metadata was in structured to have variables in the following manner, the chapter number, chunk id and the text of the current chunk. Each chapter had a corresponding chunk id which contained information about the current chapter paragraph number, coupled with the text variable to store the paragraphs text. This ensured that retrieved results could be mapped not only to specific chapters but also to their broader source context. At the end of this chunking process 369 chunks were created. The table below shows an example of the segmented data:

Chapter	Chunk_id	Text
5	5_1	My nagging got the...

Table 1 Chunked data outcome

The retrieval pipeline incorporated three document representation discussed in the methodology section, BM25, SPLADE, and the all-mpnet-base-v2 sentence transformer. In the dense retrieval implementation, the embeddings generated by processing the documents and queries with the all-mpnet-base-v2 were stored using a FAISS flat for index for efficient nearest neighbour search. While in the sparse retrieval baselines, BM25 computed similarity core directly from lexical term frequencies and inverse document frequency weighting. Furthermore, SPLADE generated sparse expanded vectors that retained interpretability while capturing contextual expansions. For all retrieval methods, the system returned the top-k ranked candidate passages which were then evaluated both independently and the where applicable reranked with a cross-encoder model to refine the relevance ordering. This design allowed for comparative analysis of how traditional lexical approaches, hybrid sparse expansion models, and dense semantic embeddings contribute to retrieval accuracy and provenance tracking. Moreover, the way the data was pre-processed allowed the evaluation of the systems at its capability to retrieve chapter or paragraph relevant to the query.

3.2.2.Paragraph level retrieval

To ensure a granular and effective evaluation, the retrieval system was designed to operate on smaller passage level units of text rather than entire chapter. This granular approach was applied to both sparse and dense retrieval models to create a consistent and robust baseline for comparison. For sparse retrievers BM25 and SPLADE, this chunking strategy was particularly critical. These models rely heavily on the exact frequency and distribution of keywords within a document. This approach maximises the likelihood of strong keyword-base match between the LLM's generated output and a specific relevant segment of the novel. A smaller chunk size mitigates the document length problem (Lv, Y. and Zhai, C., 2011) of these models where term frequency in a long chapter can dilute the importance of a specific keyword.

Once the sparse model computes similarity scores between the query and each paragraph chunk, the top paragraphs are retrieved and ranked regardless of which chapter they belong to. This approach comes with two implications; chunk level retrieval might improve only on recall of contextually relevant passages since smaller units reduce the likelihood of diluting the relevant content with unrelated material. The other implication that can occur is that fragmentation is introduced as context can be spread across multiple chapters or lost entirely if none of the chapter's chunks appear within the top-k results. This means that while the system is able to retrieve highly specific matches, provenance at the chapter level may be incomplete or completely missing, highlighting a trade off between granularity and contextual coherence.

3.2.3.Chapter level retrieval

Conversely, the dense retrieval component of the system was evaluated on both passages level and full chapter retrieval capabilities. The motivation of this dual evaluation on the dense system was to investigate whether the semantic understanding of a dense model could effectively handle the increased contextual complexity and information density of an entire chapter. Unlike sparse models with the problem of document length, dense model vectors are semantically robust allowing the model to potentially identify a relevant concept or theme even if the query does not share explicit lexical overlap with the text.

In practice, each chapter chunk was embedded into vectors using the embedding model and stored in the FAISS index. The chapter level retrieval was implemented using a grouping strategy on the indexed passages. After indexing the documents, the top 100 chunks which are semantically similar to the query are pulled from the index. These chunks are then grouped back to their original chapter. This is the core of the chapter level retrieve, instead of only returning the to chunks, the system uses them as evidence to score their parent chapters. For each chapter that contains at least one top chunks. This scoring is determined by an aggregation method with maximum and mean options. When using maximum aggregation, a chapter is given the score of the highest scoring chunk. This method emphasizes on chapter that contain at least one highly relevant passage, presuming that a single strong signal is sufficient to indicate the chapter relevance. With the mean aggregation, a chapter score is the average of all its retrieved chunks. This method rewards chapters that contain multiple relevant passages indicating broader theme or contextual match. Finally, 10 top chapters with 3 relevant chunks are returned based on these scores.

However, this design involves a different trade off. While chapter level retrieval assures entire provenance and lowers the possibility of losing contextual coherence, it can reduce precision. This is because a single vector can cover a wide range of topics, and the retrieval system may return a relevant chapter if only a small portion addresses the query. This demonstrates a contradiction between contextual completeness and targeted relevance.

3.2.4.Distribution of token length in chunks

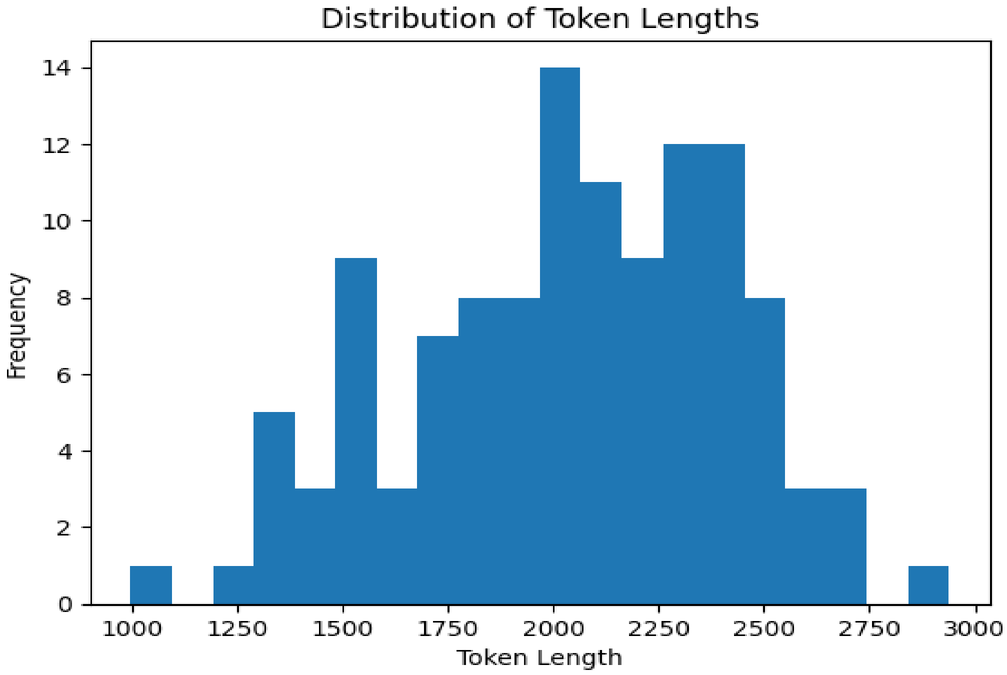

Figure 2 Toke distribution and token length

The histogram above illustrates the distribution of token length across the generated text chunks. Most chunks fall within the range of 1,500 to 2,500 tokens with relatively a few exceeding 2,700 tokens. This demonstrates that the chunking process produced reasonably balanced segments, avoiding both excessively short and very long passages. Ensuring token length is important as long chunks may dilute query importance while short chunks may lack sufficient content.

3.2.5.Distribution of sentence counts in chunks

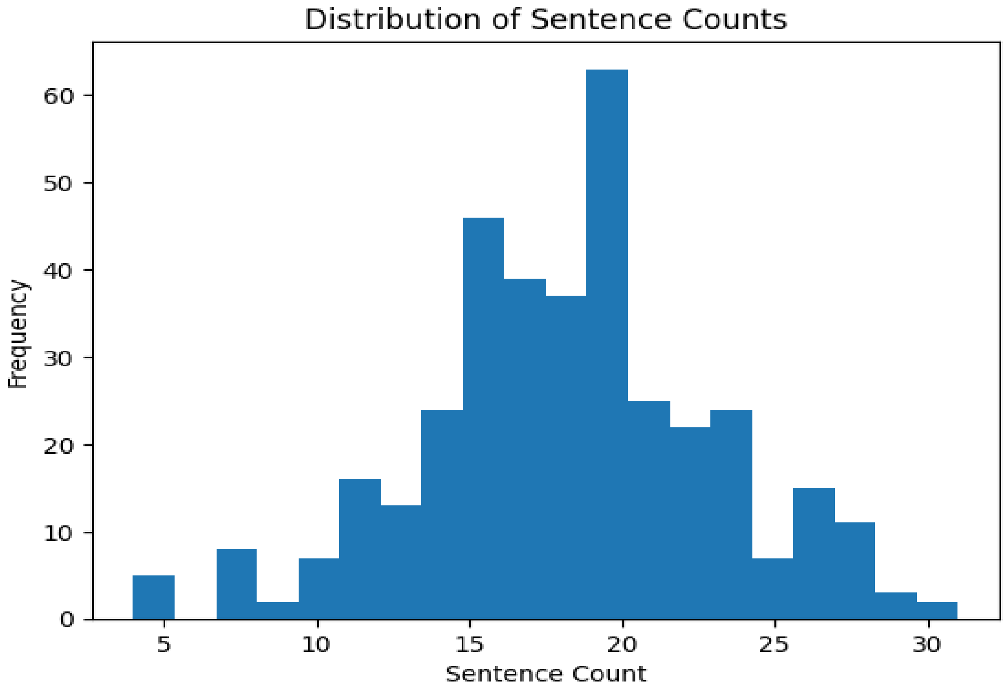

Figure 3 Sentence distribution counts

In this histogram the sentence count distribution is analysed. The analysis shows that most chunks contains 15 and 25 sentences, with a peak around 18 to 20. This indicates that chunking preserved semantic coherence by keeping passages at a natural paragraph length rather than cutting them randomly. Having balanced sentences in the ensures that the retriever has enough contextual information per chunk to capture semantic relevance without introducing fragmentation. Together, these analysis results confirm that the chose chunking strategies produced coherent and manageable retrieval units.

3.2.6.Prompt design

Since this project investigates the provenance in LLM's, the evaluation required queries that accurately reflect how LLM outputs may raise provenance issues. Prompts were created with three aims of producing both direct quotations, paraphrased text related to the dataset and completely irrelevant text not available in the dataset. Directional quotation style prompts encouraged the LLM to reproduce text more closely aligned with the source material, providing a test for retrieval precision. Contrary to this, paraphrased style prompts encouraged the model to generate content semantically similar but not lexically identical to the source. Finally, to test the system retrieval capabilities, unrelated prompts were used to verify the retrieval performance and the level of semantic relevance matching in the system. The variation in prompts varied in style and complexity, some were short and factual, while others were longer and more interpretive.

Prompt Style	Prompt	Generated output	Expected output
Short question	“What was happening to Jem in the novel To Kill a Mockingbird?”	In Harper Lee’s novel *To Kill a Mockingbird*, Jem Finch experiences a significant and poignant transformation throughout the story, both emotionally and physically...	Chapters across the novel describing Jem’s transformation.
Long and creative	“Write a short story about a summer in Manchester Uk in the style of Harper Lee.”	The summer sun hung low over Maycomb, casting long shadows through the dusty streets and swaying branches of ancient oaks...	Chapters early in the novel and at the end of the novel.
Irrelevant	“Explain how Bluetooth works.”		No chapter to be returned.

Table 2 Prompt design and expected output

3.3. Evaluation Metrics

The effectiveness of the retrieval system was evaluated using a set of standard information retrieval. These metrics were chosen to capture several aspects of the retrieval quality including accuracy, ranking performance, and graded relevance. Precision and recall are most fundamental, with precision measuring a proportion of retrieved documents that are relevant, and recall indicating the proportion of relevant documents retrieved. These measurements are frequently integrated into the F1 score which is used to balance them and calculate their harmonic mean. For ranked retrieval tasks, metrics such as Precision@k and Recall@k are used to assess performance at the top-k of the results list which is especially significant in practical applications where users rarely examine results. Mean Average Precision (MAP), captures both precision and ranking quality across all quires it is defined as the mean of the average precision scores for each query.

Normalized Discounted Cumulative Gain (nDCG) and Mean Reciprocal Rnak (MRR), further account for graded relevance and ranking order, providing a more nuanced evaluation of system performance (Manning, C.D., 2008). During evaluation, these metrics required ground truth document to compare the retrieved documents to. This is a significant challenge because ground truth is not readily available when evaluating a black box LLM like ChatGPT. It is not known which specific passages if any the model used during its training to generate a particular response.

To overcome this challenge, an alternative ground truth methodology was adopted. For each query the most relevant passages from the novel were chosen using manual annotation and preliminary search using an unbiased relevance model. These human annotated passages served as the ground truth. This approach allows us to mimic a ground truth environment, offering a measurable and objective basis for assessing the retrieval system's performance in a real-world provenance scenario.

3.4. Evaluation Scenario

Several evaluation scenarios were developed to fully analyse the effectiveness of the provenance retrieval system. These scenarios represent various retrieval configurations and levels of text granularity, allowing for systematic comparison between sparse and dense models. Each scenario aims to emphasize distinct trade-offs between retrieval accuracy, provenance preservation, and contextual coherence.

The evaluation was therefore structured around two primary dimensions. First, retrieval granularity was altered, with experiments conducted at both and complete chapter levels. This distinction determines whether fine grained retrieval provides higher precision at the expense of fragmentation, or whether chapter level retrieval maintains stronger provenance but lower specificity. Second, representation strategies were evaluated, using BM25 as a lexical baseline, SPLADE as a sparse semantic hybrid, and all-mpnet-base-v2 as a dense semantic model. Finally, cross-encoder re-ranking was assessed as a second stage refining process to determine its ability to increase ranking quality.

4. Results and Discussion

This chapter presents the results of the experiments described in the evaluation framework. The focus is on assessing how well the system retrieved contextually relevant passages and preserved provenance across different retrieval strategies and level of granularity. The results are organised into quantitative performance metrics, and qualitative provenance tracking of outputs, and comparative analyses of sparse and dense models. Together these findings provide insight into the strength and limitations of the proposed approach.

4.1. Quantitative Results

This section presents the quantitative outcomes of the retrieval outcomes of the retrieval experiments, focusing on the comparative performance of sparse and dense models across different granularity using evaluation metrics discussed in chapter 3. Results are reported separately for paragraph level and chapter level retrieval, with each configuration testing using BM25, SPALDE and all-mpnet-base-v2. In addition to the effect of cross encoder reranking on the retrieved candidates is examined to examine ranking quality. The table below shows the metrics after retrieving and ranking documents before re-ranking. Given the same prompt to the LLM for the three models, the generated outputs differ but are similar in the theme and context.

Retrieval Level	Model	Precision@10	Recall@10	F1 Score	MAP	NDCG@10	MRR
Paragraph	BM25	0.1	0.1	0.10	0.0478	0.0733	0.1428
Paragraph	SPLADE	0.2	0.2	0.0	0.045	0.158	0.25
Paragraph	All-mpnet-base-v2	0.8	0.8	0.800	0.933	0.860	1.0
Chapter	All-mpnet-base-v2	0.9	0.9	0.9	0.978	0.933	1.0

Table 3 Model results before re-ranking

4.2. Qualitative provenance Tracing and Comparative Analysis

The results in table 3 show that the sparse baseline performed rather poorly with BM25 achieving a precisio@10 and Recall@10 of only 0.1, with an F1 score of 0.10, whilst SPLADE improved slightly at 0.2 for both precision and recall but still had low performance. These results highlight the limitation of sparse models when used on lengthy narrative texts, where lexical overlap is insufficient to capture semantic relationships. In comparison, the dense model performed significantly better. At the paragraph level, it achieved Precision@10 and Recall@10 of 0.8 with an F1 score of 0.8, while at chapter level it achieved 0.9 on all three metrics. The ranking metrics supports this by review MAP chich increased to 0.933 and 0.978 at the paragraph and chapter levels, whereas Ndcg@10 approached 0.933 in the chapter level retrieval. Importantly, an MRR of 1.0 implies that a relevant passage was retrieved at the top rank position.

After re-ranking the documents, the retrieved documents in tow of the models, SPLADE and all-mpnet-base-v2. The results did not improve in SPLADE, whilst in the dense model only a few results such as MRR and MAP increased to 0.99. At the chapter level, the precision recall curve confirmed this improvement, showing precision close to 1.0 across most recall values (area under the curve (AUC) = 0.80, average precision (AP) = 0.90), indicating that re-ranking consistently placed relevant chapters at the top while preserving provenance. These findings highlight that dense retrieval combined with cross-encoder re-ranking is far more effective for provenance tracking than sparse methods, as it maintains near-perfect precision while sparse models fail to benefit from re-ranking.

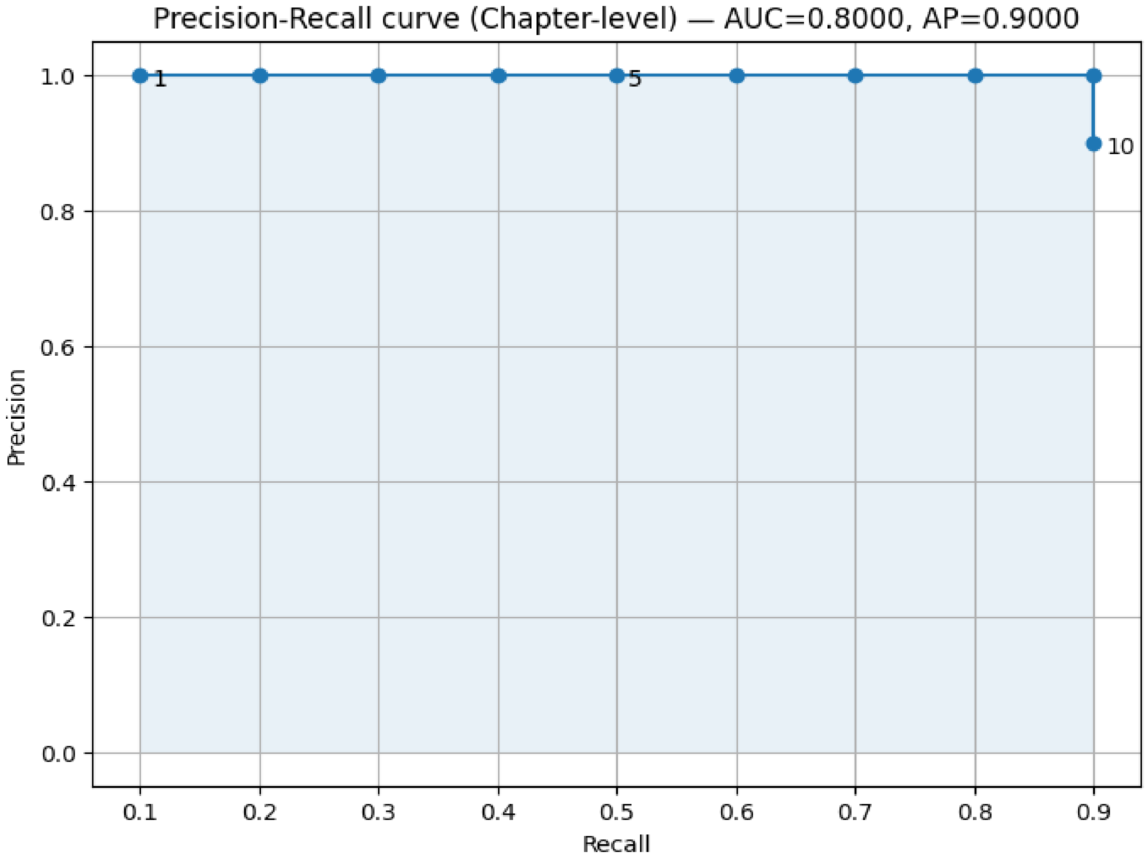

Table 4 Precision vs recall with AUC & AP after re-ranking

4.3. Discussion

This approach showed strong retrieval performance with dense embeddings and cross-encoder re-ranking, achieving near perfect precision at the chapter level and reliable provenance tracking. Paragraph level retrieval improved recall by capturing fine grained matches, while FAISS indexing enabled efficient search. Regardless of this strength this approach faces limitations when it comes to the use of sparse models as these models performed poorly and did not benefit from re-ranking. Dense retrievals, while effective they are computationally expensive especially with cross-encoder re-ranking. Chapter level retrieval may reduce precision by returning overly broad results, while paragraph level retrieval risks context fragmentation. Finally, the results are based on a single corpus limiting generalisability.

5. Conclusion and Future Work

This project sets out to investigate the provenance in large language models through the design and evaluation of an information retrieval-based pipeline. By comparing sparse retrieval methods (BM25 and SPLADE) with dense embeddings (all-mpnt-base-v2), and integrating FAISS indexing with cross encoder re-ranking, the study demonstrates that dense retrieval provides superior performance for provenance tracking. Chapter level retrieval achieved near perfect precision and preserved contextual coherence, while paragraph level retrieval improved recall but introduced fragmentation. Sparse models performed poorly overall, underscoring their limitations in semantically complex tasks. The results confirm that information retrieval pipelines can serve as a viable approach for enhancing transparency and accountability in LLM outputs.

5.1. Future Work

Future work could be focused on improving the efficiency of this pipeline by exploring hybrid sparse dense retrieval models that balance performance and computational cost. Extending the evaluation to lager and more diverse datasets would also improve the generalisability of the finings, provenance tracking could be enhanced by incorporating watermarking or attribution models and techniques, enabling not only retrieval-based provenance but also verifiable guarantees of source origin. Finally, user focused evaluation studies could assess the practical value of provenance explanations in real word applications.

References

Riofrio Martinez-Villalba, J.C., 2024. Refounding Copyright in the Right to Dialogue: Consequences for ChatGPT and other Artificial Intelligence Services.

Yu, H., Atanasova, P. and Augenstein, I., 2024. Revealing the parametric knowledge of language models: A unified framework for attribution methods. *arXiv preprint arXiv:2404.18655*.

Casper, S., Ezell, C., Siegmann, C., Kolt, N., Curtis, T.L., Bucknall, B., Haupt, A., Wei, K., Scheurer, J., Hobbhahn, M. and Sharkey, L., 2024, June. Black-box access is insufficient for rigorous ai audits. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (pp. 2254-2272).

Koh, P.W.W., Ang, K.S., Teo, H. and Liang, P.S., 2019. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, *32*.

Schmidt, Robin M. "Recurrent neural networks (rnns): A gentle introduction and overview." *arXiv preprint arXiv:1912.05911* (2019).

Abuqaddom, I., Mahafzah, B.A. and Faris, H., 2021. Oriented stochastic loss descent algorithm to train very deep multi-layer neural networks without vanishing gradients. *Knowledge-Based Systems*, *230*, p.107391.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, *30*.

Raganato, A. and Tiedemann, J., 2018, November. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: analyzing and interpreting neural networks for NLP* (pp. 287-297).

Yenduri, G., Ramalingam, M., Selvi, G.C., Supriya, Y., Srivastava, G., Maddikunta, P.K.R., Raj, G.D., Jhaveri, R.H., Prabadevi, B., Wang, W. and Vasilakos, A.V., 2024. Gpt (generative pre-trained transformer)—A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE access*, *12*, pp.54608-54649.

Peng, B., Li, C., He, P., Galley, M. and Gao, J., 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Sparck Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), pp.11-21.

Robertson, S.E. and Walker, S., 1994, July. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University* (pp. 232-241). London: Springer London.

Li, X., Jin, J., Zhou, Y., Zhang, Y., Zhang, P., Zhu, Y., & Dou, Z., 2024. From Matching to Generation: A Survey on Generative Information Retrieval. *ACM Transactions on Information Systems*, 43, pp. 1 - 62. https://doi.org/10.1145/3722552.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019, June. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

Balakrishnan, V. and Lloyd-Yemoh, E., 2014. Stemming and lemmatization: A comparison of retrieval performances.

Hiemstra, D., 2009. Information retrieval models. *Information Retrieval: searching in the 21st Century*, pp.1-19.

Petrov, A.V., MacAvaney, S. and Macdonald, C., 2024, March. Shallow cross-encoders for low-latency retrieval. In *European Conference on Information Retrieval* (pp. 151-166). Cham: Springer Nature Switzerland

Manning, C.D., 2008. *Introduction to information retrieval*. Syngress Publishing.

Sunstein., 2024, The New York Times V. openai: The biggest IP case ever, Sunstein LLP. Available at: https://www.sunsteinlaw.com/publications/the-new-york-times-v-openai-the-biggest-ip-case-ever (Accessed: 26 August 2025).

Shokri, R., Stronati, M., Song, C. and Shmatikov, V., 2017, May. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)* (pp. 3-18). IEEE.

Hayes, J., Melis, L., Danezis, G. and De Cristofaro, E., 2017. Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U. and Oprea, A., 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)* (pp. 2633-2650).

Wei, J.T.Z., Wang, R.Y. and Jia, R., 2024. Proving membership in LLM pretraining data via data watermarks. *arXiv preprint arXiv:2402.10892*.

Lee, H., 2010. *To Kill a Mockingbird*. Random House.

Rahul, 2024, A guide to chunking strategies for retrieval augmented generation (RAG), Zilliz Learn. Available at: https://zilliz.com/learn/guide-to-chunking-strategies-for-rag (Accessed: 29 August 2025).

Robertson, S. and Zaragoza, H., 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, *3*(4), pp.333-389.

Formal, T., Piwowarski, B. and Clinchant, S., 2021, July. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2288-2292).

Agarap, A.F., 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R. and Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.Y., 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, *33*, pp.16857-16867.

Reimers, N. and Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.E., Lomeli, M., Hosseini, L. and Jégou, H., 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Johnson, J., Douze, M. and Jégou, H., 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, *7*(3), pp.535-547.

Jithin James. (2023). *The Impact of Temperature in LLMs: Balancing Determinism and Creativity*.Medium. Available at: https://medium.com/@jithinpjames/the-impact-of-temperature-in-llms-balancing-determinism-and-creativity-95a066e10ce6

Nogueira, R. and Cho, K., 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R. and Deng, L., 2016. Ms marco: A human-generated machine reading comprehension dataset.

Lv, Y. and Zhai, C., 2011, July. When documents are very long, bm25 fails!. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 1103-1104).

Appendices