**REPORT**

**On**

**A Multi-Module AI System for Intelligent Health Insurance Support Using Retrieval-Augmented Generation**

**MASTEROF TECHNOLOGY**

**IN**

**Artificial Intelligence and Machine Learning**

SUBMITTED BY

**Sejal Hanmante**

**(PRN - 24070149024)**

**Shivani Patil**

**(PRN - 24070149025)**

A red logo with a globe and text

Description automatically generated

**SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE – 412115**

**(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY))**

**2024-25**

**Abstract**

This project presents a smart NLP-based system for assisting users through the complexities of health insurance. The system integrates conversational assistance, health insurance policy suggestion, and health insurance document retrieval in one, easy-to-use platform. It allows users to pose questions naturally, receive personalized suggestions for suitable health insurance policies, and retrieve specific clauses or information from policy documents. It seeks to streamline the user interaction by pulling out unstructured insurance content into digestible insights readily available. With natural language understanding and semantic retrieval, the system allows for wiser decision-making and improved access to critical insurance information. Performance of each module is evaluated using [insert chatbot eval metric] for conversation quality, [insert recommendation system metric] for policy suitability, and [insert retrieval eval metric] for document-level information accuracy. This introduction demonstrates the potential of NLP in enabling digital interaction in the healthcare insurance space.

**Keywords:** Natural Language Processing (NLP), Health Insurance, Policy Recommendation, Information Retrieval, Conversational AI, Semantic Search, Retrieval-Augmented Generation (RAG), Document Understanding, Question Answering, Insurance Chatbot, User-Centric Systems

1. **Introduction**

Health insurance serves as a critical safeguard against the escalating costs of medical care, offering individuals and families financial protection during health crises. In India, despite the implementation of significant public health initiatives like Ayushman Bharat, a substantial portion of the population remains uninsured or underinsured [1]. As of 2024, approximately 700 million Indians lack health insurance coverage, highlighting a significant gap in financial protection against health-related expenses [2].

The complexity of health insurance policies contributes to this gap. Policy documents are often laden with technical jargon and intricate terms, making them difficult for the average consumer to comprehend. This lack of clarity can lead to misunderstandings about coverage details, resulting in unexpected out-of-pocket expenses during medical emergencies. Furthermore, the overwhelming number of available insurance plans, each with varying terms and conditions, adds to the confusion, making it challenging for consumers to make informed decisions.​

The integration of Artificial Intelligence (AI) and Natural Language Processing (NLP) into the health insurance sector presents a promising solution to these challenges. AI technologies can analyze vast amounts of data to predict health risks more accurately, leading to fairer premium calculations and more personalized coverage options . NLP, a subset of AI, enables machines to understand and interpret human language, facilitating the automation of complex tasks such as summarizing policy documents and answering customer queries in real-time. These technologies can demystify insurance policies, making them more accessible and understandable to the general public.​

In India, the adoption of AI and NLP in health insurance is gaining momentum. Insurers are leveraging these technologies to enhance customer service, streamline claims processing, and develop more accurate risk assessment models. For instance, AI-driven chatbots are being used to provide instant responses to customer inquiries, while NLP algorithms are employed to extract and summarize key information from lengthy policy documents, aiding consumers in understanding their coverage details[3]​

Despite these advancements, challenges remain in fully realizing the potential of AI and NLP in the health insurance sector. Issues such as data privacy concerns, the need for high-quality data, and the integration of these technologies into existing systems must be addressed. Moreover, ensuring that AI and NLP tools are user-friendly and accessible to individuals with varying levels of digital literacy is crucial for widespread adoption.​

To bridge the gap between complex insurance policies and consumer understanding, this project proposes the development of an NLP-based system comprising three core components:​

1. **Conversational Assistant**: An AI-powered chatbot designed to answer user queries regarding insurance policies in a straightforward and comprehensible manner, reducing the need for manual document review.​
2. **Policy Recommendation Engine**: A system that analyzes individual needs and preferences to suggest the most suitable insurance plans, considering factors such as specific medical conditions, budget constraints, and desired benefits.​
3. **Document Retrieval System**: A tool that allows users to upload insurance policy documents and search for specific clauses or information, providing easy access to relevant details without navigating through complex texts.​

By integrating these components, the proposed system aims to empower consumers with the knowledge and tools necessary to make informed decisions about their health insurance coverage. This approach not only enhances transparency and understanding but also contributes to increased insurance penetration and financial protection for individuals across India.

1. **Literature Survey**

**Table 1: Literature Review**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Year** | **Methods Used** | **Dataset** | **Major Findings** | **Limitations** |
| [4] | 2024 | Fine-tuning, RLHF, Prompt Engineering, RAG | Various datasets for personalized LLM evaluation | Explores various personalization techniques for LLMs to improve user-specific responses | Lack of high-quality user-labeled datasets, cold-start problem, privacy concerns |
| [5] | 2024 | RAG, Dense Vector Retrieval (FAISS), Hybrid Search (BM25 + Neural Reranking) | Proprietary unstructured documents | RAG significantly enhances response relevance by incorporating external knowledge | Scaling RAG, reliability issues, hallucination in LLMs |
| [6] | 2023 | NER, Transformer-based NLP models (BERT, RoBERTa), OCR | 9 Swedish condo insurance policy documents | Automates information extraction, reducing manual effort in insurance policy analysis | Imbalanced dataset, limited positive samples, need for human interpretation |
| [7] | 2020 | Rule-based Chatbot, Transformer-based Conversational AI (DialoGPT), Intent Recognition (BERT) | Cornell movie dialogue dataset, custom insurance dataset | Enhances customer support by automating policy-related queries | Limited by dataset scope, lack of real-time learning |
| [8] | 2023 | Decision Trees, Random Forest, NLP-based Claim Analysis (BERT, LSTM) | 88 real-life road traffic accident cases | AI-driven models improve dispute resolution efficiency and fairness | Small dataset, legal language complexity |

A general deep learning framework was introduced by [9] for non-factoid question answering without the support of any linguistic resources, making it flexible with respect to language and domain. Various architectures were tried and evaluated. A novel QA corpus and task in the insurance domain was presented. Experimental results indicated robust improvements over baselines, registering a top-1 accuracy of 65.3% on the test set, indicating good potential for practical applications.

Large Language Models (LLMs) are transforming the insurance sector by automating operations, enhancing customer interaction, and facilitating risk assessment. Although bringing significant operational efficiency, their deployment necessitates close attention to data privacy, security, and ethics. With robust risk management and compliance protocols, LLMs can assist insurers in streamlining efficiency, trust, and innovation [10].

The study by [11] suggests an AI-driven system for automatic information retrieval and extraction from PDF files using PyPDF2 for text extraction, FAISS for vector search, and embedding models such as OpenAI and HuggingFace for semantic meaning. In contrast to conventional keyword searches, the system retrieves information based on conceptual meaning and context by embedding text chunks into high-dimensional vector spaces and using cosine similarity for matching. Performance comparisons with other embedding models indicate dramatic boosts in retrieval effectiveness and efficiency. The paper additionally mentions applications involving legal research, academic repositories, and enterprise knowledge management, while future work entails improving multimedia management and semantic penetration. A recent systematic review by [12] highlighted the increased contribution of Large Language Models (LLMs) in healthcare and underscored the importance of having clearer evaluation frameworks. The research classified 519 articles published from January 2022 to February 2024, exploring evaluation data types, healthcare tasks, NLP/NLU tasks, evaluation dimensions, and medical specialties. Interestingly, only 5% of the studies employed actual patient care data, and the majority examined evaluating medical knowledge (44.5%), diagnosis (19.5%), and patient education (17.7%). Administrative uses such as billing code assignment, prescription writing, and clinical notetaking were seldom tested. Question answering was the most prevalent NLP/NLU task (84.2%), whereas fairness, bias, robustness, and deployment considerations were not thoroughly explored. Internal medicine was the best-represented specialty (42%), while areas such as nuclear medicine, physical medicine, and medical genetics were underrepresented. These results highlight critical shortcomings in LLM assessment, especially in terms of deployment to real-world conditions and extended healthcare contexts.

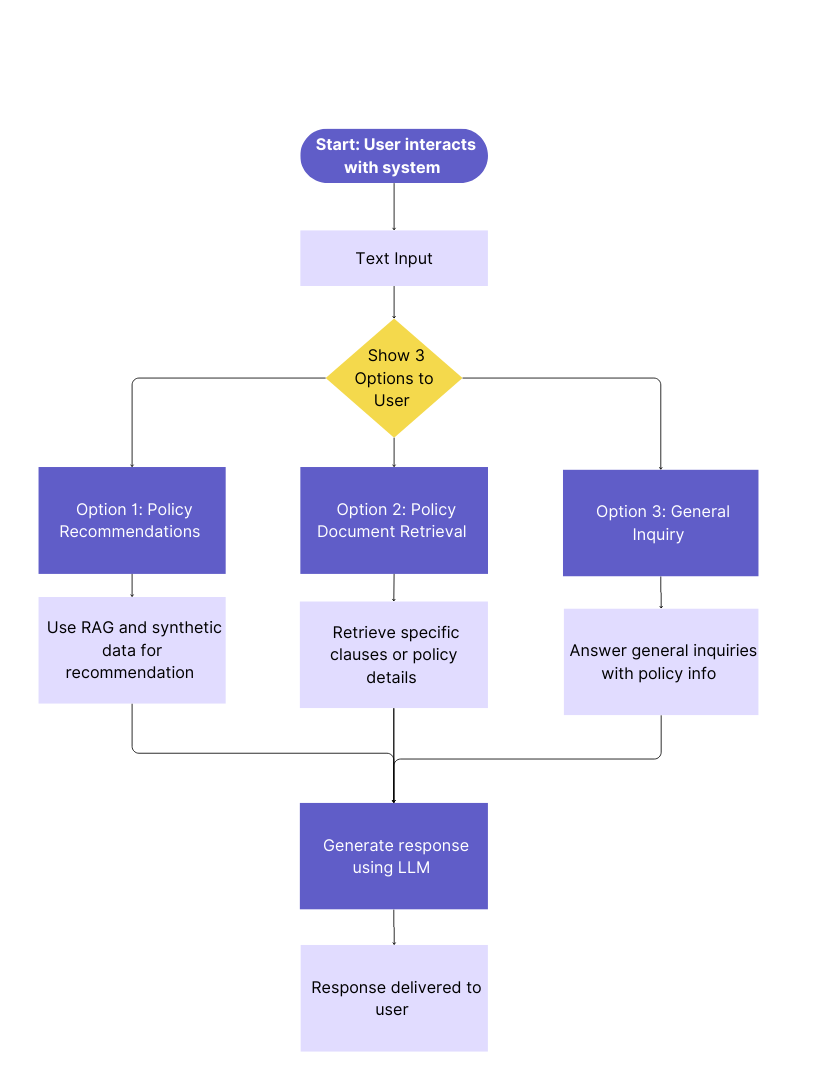
The health insurance sector is facing digitalization via AI developments. InsureGenie, a platform based on AI proposed by [13], makes enrollment easier via multilingual voice support and NLP, supporting smooth user engagement across languages. It responds to intricate questions, offers customized advice, and becomes better over time via ongoing learning. This research points out how AI-based solutions such as InsureGenie can make health insurance more user-friendly and accessible.

GPT-based systems show potential in improving the efficiency and accuracy of medical assistance but face challenges in handling sensitive healthcare information. Traditional chatbot architectures have evolved from rule-based systems to neural encoder-decoder models, significantly enhancing conversational capabilities. RAG models further improve response quality by integrating external knowledge sources during generation, proving valuable for open-domain tasks, including healthcare. Despite their success, challenges such as retrieval quality, scalability, and ethical concerns like bias and misinformation remain critical areas for future research [14], [15], [16].

The shows that most of the earlier work in Natural Language Processing (NLP) and Artificial Intelligence (AI) is generic in nature for applications like customer care automation, decision-making systems, and information retrieval. Most of the work is focused on general domains, for instance, e-commerce, health care, or legal cases and is focused on enhancing response relevance using techniques such as fine-tuning, reinforcement learning, and retrieval-augmented generation (RAG). Although they show widespread progress in automation of tasks such as document retrieval and query answering, none of the earlier work focuses specifically on the issue of handling insurance policy documents in general and health insurance in particular.

One of the key differences of the suggested project is that it addresses the insurance policy segment, which has not received proper attention in previous research. The project aims to develop an NLP-based system for interpreting, suggesting, and retrieving relevant health insurance policy clauses, an industry-specific requirement that has not been fulfilled by previous solutions. Unlike general-purpose AI implementations, the project aims to derive structured, actionable information from unstructured health insurance documents, which is unique in terms of industry specificity. This is the area of previous research where concentrated attention on policy documents—i.e., health insurance—has been absent.

1. **Methodology**

****

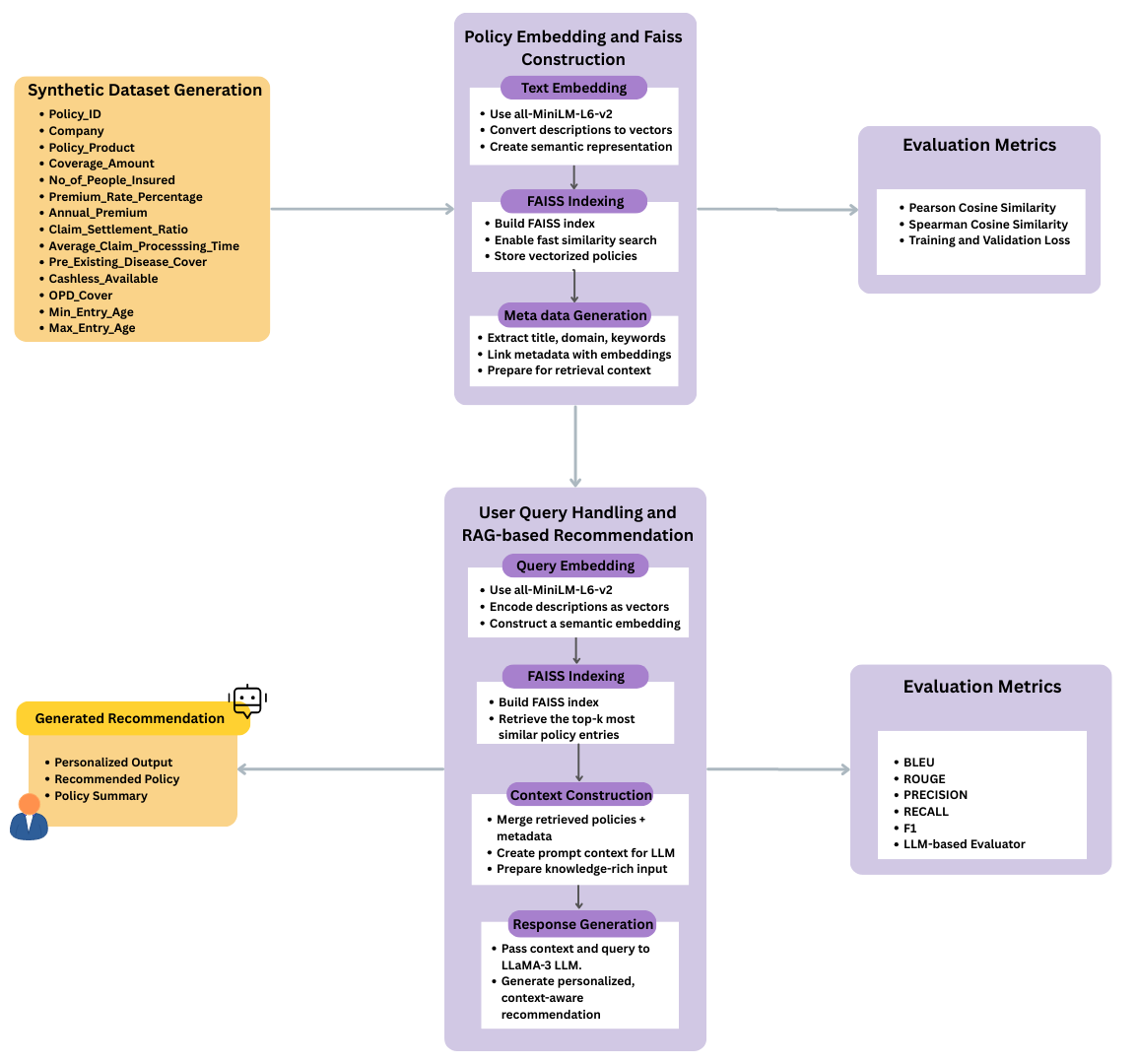
**Figure. 1 Overall System Flow**

The Figure. 1 represents the overall architecture of the proposed research study. It starts with user's interaction with the system through entering a text input. The system presents three alternatives to the user, guiding the procedure in accordance with their respective requirements:

1. Policy Recommendations – Employs Retrieval-Augmented Generation (RAG) and synthetic data for customized policy recommendations.
2. Policy Document Retrieval – Recovers specific policy information or clauses from documents.
3. General Inquiry – Answers general questions related to any policy.

Irrespective of the alternative selected, the system receives the input and produces a response based on a Large Language Model (LLM), which is then displayed to the user. All these three modules will be explained in depth in the following sub-sections.

* 1. **Policy Recommendations**

****

**Figure. 2 Policy Recommendation System Architecture**

**3.1.1 Synthetic Dataset Generation**

As demonstrated in Figure. 2 to enable system development and testing of policy recommendation system, synthetic dataset was programmatically created to mimic actual insurance policy records. Since no publicly available datasets for this space with semantic and numeric features were present, synthetic data was used to mimic end-to-end workflow and test the system under different scenarios.

The data set is comprised of structured fields typical in insurance policy records, i.e., numerical, categorical, and binary features. 1,000 synthetic policy records were generated using Python libraries such as NumPy and Pandas. Logical field interdependencies were maintained to simulate real-world situations. For example, the Annual Premium was computed as a percentage of the Coverage Amount, and entry ages were limited within typical insurance age ranges.

The following traits were included:

**Table. 2 Synthetic Dataset Description**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Feature Name** | **Description** |
| 1 | Policy\_ID | Unique identifier for each policy |
| 2 | Company | Name of the insurance provider |
| 3 | Policy\_Product | Type of insurance (e.g., Term Life, Health Cover) |
| 4 | Coverage\_Amount | Total amount insured (in INR) |
| 5 | No\_of\_People\_Insured | Number of individuals covered under the policy |
| 6 | Premium\_Rate\_Percentage | Rate used to calculate premium as a percentage of coverage amount |
| 7 | Annual\_Premium | Derived as (Coverage\_Amount × Premium\_Rate\_Percentage) / 100 |
| 8 | Claim\_Settlement\_Ratio | Percentage of claims settled successfully by the insurer |
| 9 | Average\_Claim\_Processing\_Time | Time (in days) taken to process a claim |
| 10 | Pre\_Existing\_Disease\_Cover | Binary indicator (1/0) for pre-existing disease coverage |
| 11 | Cashless\_Available | Binary indicator for availability of cashless hospitalization |
| 12 | OPD\_Cover | Binary indicator for Outpatient Department (OPD) coverage |
| 13 | Min\_Entry\_Age | Minimum age eligible to enter the policy |
| 14 | Max\_Entry\_Age | Maximum eligible age |

All numeric values were generated using uniform or normal distributions, while categorical fields were sampled from a predefined list of realistic values. This ensured balanced variability and distribution across features.

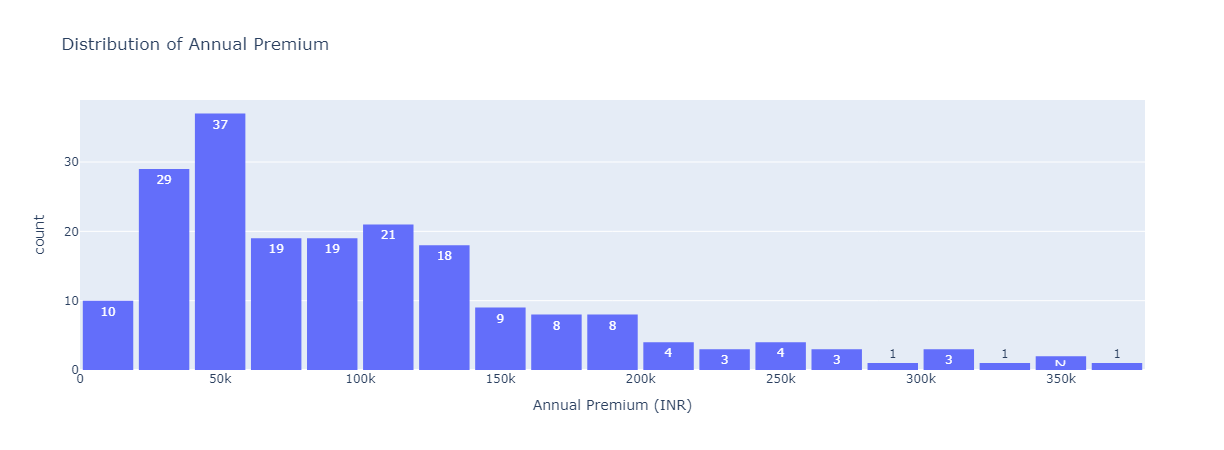


Figure. 3 Annual Premium Distribution

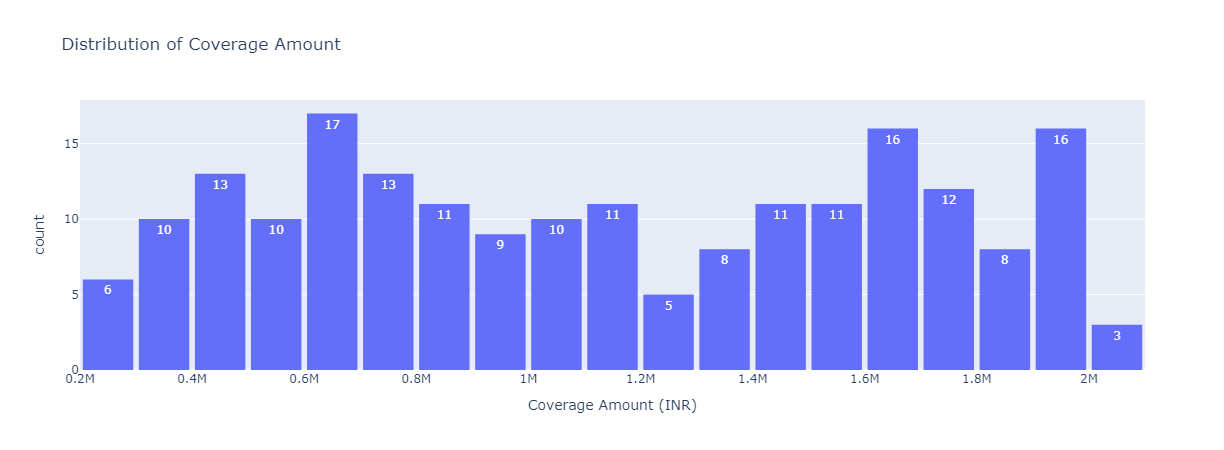
Figure. 3 is the distribution of the annual premium in INR terms across all policyholders in our data set. Premiums in real-world insurance markets will cluster around the middle, by affordability and usual levels of policies. This is a pattern that we observe in our data set as well, where the majority of policies cluster between ₹40,000 and ₹60,000 with an extremely dominant peak around ₹50,000. High-premium policies drop off increasingly, to a right-skewed distribution which is typical consumer behavior — only a smaller sub-group of consumers buy or are in a position to buy higher-level policies. Such a distribution makes machine learning models trained on this data set plausible in terms of price levels and consumer segments.

Figure. 4 Coverage amount of Policies

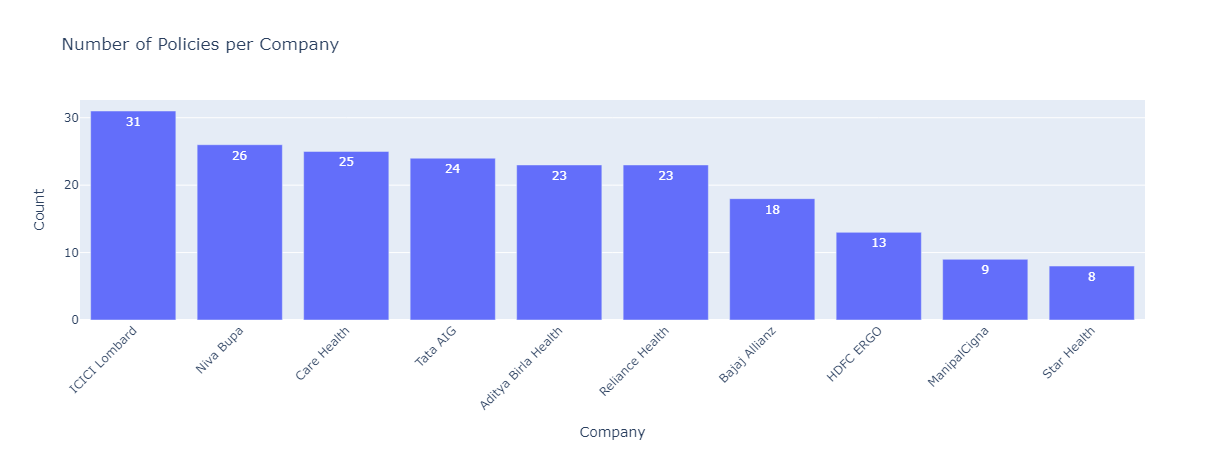
The second of these graphs i.e. Figure. 4, illustrating the distribution of the value of insurance coverage, again illustrates the consistency of the data with actual data. The values of coverage range from ₹200,000 to ₹2,000,000, illustrating a wide range of policy choices in the Indian insurance market. The distribution is not strongly skewed towards one range of values; rather, it distributes the values of coverage quite evenly, with small humps around ₹600,000 and again around ₹1,600,000 to ₹2,000,000. The spread illustrates the range of variation in customer need and financial capacity, from plain vanilla coverage plans to more complete health security. The occurrence of higher values indicates inclusion of policies for customers with higher risk profiles or group insurance plans offered by companies.

Figure. 5 Number of Policies per company

With regard to provider representation, Figure. 5 shows the number of policies by insurer. Market leaders ICICI Lombard, Niva Bupa, and Care Health have the maximum number of policies, as one would expect of their extensive market presence in India. Small players like Star Health and ManipalCigna are underrepresented again as would be expected of real market share arrangements. This distribution was kept in place on purpose while creating the datasets so that the inherent statistical bias in natural insurance data is preserved, thus making the downstream analysis or predictive modeling more realistic and representative.

**3.1.2 Policy embedding and FAISS construction**

After the synthetic insurance dataset is formed, the subsequent key step is to convert the semi-structured and structured policy data into semantic matching-supported form. It is achieved by mapping text policy descriptions to dense vectors through advanced sentence embedding models. Policy description of every policy, i.e., product name, benefits, and terms, is mapped to a numerical embedding through a transformer-based language model, i.e., all-MiniLM-L6-v2. The model, which is trained on a humongous English sentence corpus, can capture semantic relations between texts in a compact representation. After textual descriptions are embedded into embeddings, the vectorized representations are indexed into a fast search data structure with FAISS (Facebook AI Similarity Search). FAISS is a library for fast similarity search and clustering of dense vectors. In practice, the model forms a vector space where semantically similar policies are close to each other. This allows for efficient recall of the top-k nearest policy embeddings for a user query that is also embedded to the same vector space.

Metadata is retrieved and incorporated into every policy entry to further enhance the retrieval system. Metadata includes policy name, appropriate insurance domain keywords (health or maternity, for instance), and keyword extraction. Imbedded vector representation and metadata help the system provide rich descriptive context while recalling the most related policies so more coherent and informative responses are generated in later phases.

This process replicates the retrieval-based systems in real life in today's recommendation systems and semantic search tools, where the ability to interpret and compare user intent with semantic product descriptions is crucial. The goal is not just keyword matching, but to understand query intents and policy content in depth, thus making the retrieval intelligent and personalized.

**3.1.3 Evaluation Metrics**

To evaluate the effectiveness of the policy recommendation system, we used embedding-based retrieval combined with ranking metrics. Specifically, we leveraged the Sentence Transformer model (all-MiniLM-L6-v2) to encode both the health insurance policy descriptions and a set of carefully crafted evaluation queries into dense vector representations.

A FAISS index (IndexFlatL2) was constructed using the policy embeddings to enable efficient nearest-neighbor search. During evaluation, each sample query was embedded and compared against the indexed policies to retrieve the top-k (k=5) most relevant policies based on cosine similarity (approximated via L2 distance).

To enable evaluation, each query was manually associated with a set of relevant document indices based on expert knowledge. Several retrieval evaluation metrics were computed. Precision@K measured the proportion of retrieved documents that were relevant within the top-k results, while Recall@K captured how many of the total relevant documents were successfully retrieved. The Hit@K metric indicated whether at least one relevant document appeared in the retrieved set. Additionally, the Normalized Discounted Cumulative Gain (NDCG@K) was calculated to assess not just the presence but the ranking quality of the retrieved documents, giving higher weight to relevant documents appearing earlier.

The formula for DCG (Discounted Cumulative Gain) is:

where reli is the relevance score at position *i*.

The NDCG is computed by normalizing DCG by the Ideal DCG (IDCG):

For the evaluation of the LLM-generated responses, a set of 10 diverse queries related to health insurance requirements was prepared. For each query, both the generated answer from the LLM-RAG system and a corresponding high-quality reference answer were created. The evaluation was conducted using multiple metrics: BLEU score, ROUGE-L, and BERTScore. BLEU measured the n-gram overlap between the generated and reference answers, ROUGE-L evaluated the longest common subsequence similarity, and BERTScore assessed the semantic similarity at a deeper contextual level. The evaluation script computed these metrics across all 10 query-response pairs, providing a comprehensive view of the model’s fluency, relevance, and semantic fidelity. This multi-metric approach ensured a balanced assessment of the LLM's ability to generate clear, accurate, and contextually relevant policy recommendations.

**Brevity Penalty (BP):**

The **Brevity Penalty** is primarily used in machine translation evaluation, particularly as a component of the BLEU score. It penalizes hypotheses that are shorter than the reference translations to discourage the generation of excessively brief outputs. It is formally defined in Equation ():

where:

* *c*= total length of generated sentences
* *r* = total length of reference sentences

### **BERTScore**

**BERTScore** evaluates text generation quality by computing token-level similarity using contextual embeddings from pretrained language models such as BERT. Unlike traditional n-gram overlap metrics, BERTScore captures semantic similarity. The three components of BERTScore—Precision, Recall, and F1—are defined in Equations (2), (3), and (4), respectively.

**Precision@K**

It is a commonly used metric in information retrieval to evaluate the relevance of the top K retrieved documents. It is defined as the proportion of relevant documents within the top K results, as shown in Equation (5):

This metric assesses the exactness of the retrieval system.

### **Recall@K**

**It** measures the proportion of all relevant documents that are retrieved in the top K results. It focuses on the **completeness** of the retrieval system. It is defined in Equation (6):

**3.1.4 User Query Handling and RAG-based Recommendation**

RAG architecture is used to execute user queries and generate recommendations. The architecture uses a two-stage model with a dense retriever for obtaining context snippets and an LLM for response in a natural manner. In this process, the LLM used is LLaMA 3, but it is accessed via the OpenRouter API.

The retrieval module identifies the top-k most semantically relevant documents or units of knowledge most similar to the user's input query. The retrieved units are incorporated into the initial query and given as input to the generative model. Contextual retrieval guarantees the LLM to output responses as a function of pertinent knowledge, hence promoting informativeness and factuality.

This hybrid architecture prevents generation-only model defects by capitalizing on external, verifiable sources of information. It is particularly well-suited for contexts in which dynamic or domain knowledge is required and hallucination or generic generation is poised to erode user trust or usability.

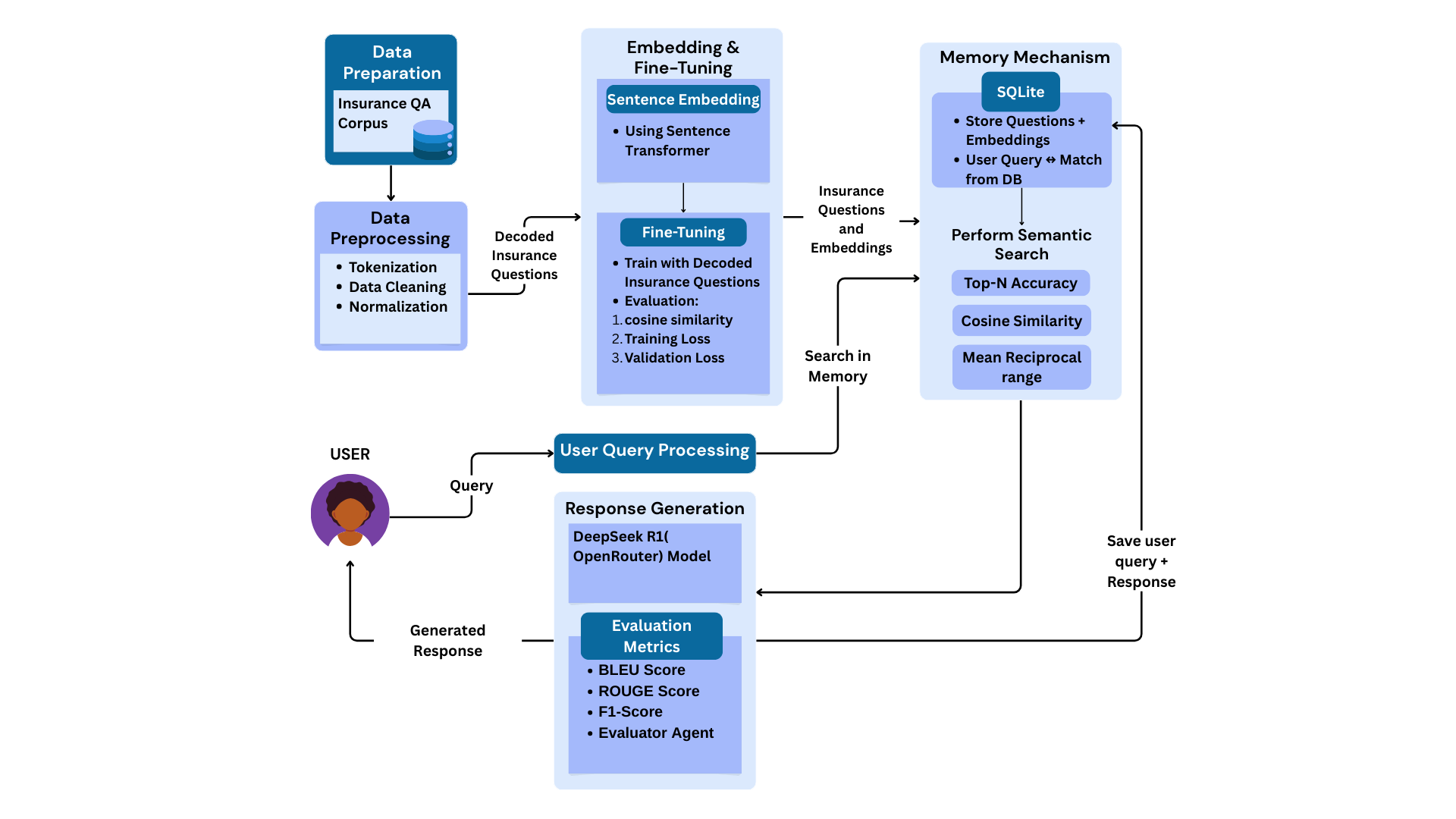
**3.1.5 Generated Recommendations**

After the retrieval and understanding phase of the query, the system goes on to create context-sensitive recommendations through the generative capability of the LLaMA 3 language model. The recommendations are created through the combination of the semantic meaning of the user query and factually related retrieved segments, thereby creating responses that are not only coherent but also user-specific for informational need.

The generative model processes input sequences as a type of enriched prompts, with the top-k retrieved document snippets augmented by the original query. The model can generate grounded responses according to real-world world data, with no hallucination and ensuring the accuracy of facts. The output recommendations can also differ in form based on the domain and intention of the user.

The generative model is trained for language fluency and optimizing query and context semantic alignment. This is achieved through the use of the transformer architecture in LLaMA 3 with attention mechanisms capable of capturing long-range query term and relevant factual content dependencies. The entire process yields semantically accurate, contextually suitable, and user-focused responses. Such recommendations not only respond to the user's question but also to further questioning or decision-making, thus strengthening the interactive feature of the system.

1. **Insurance Chatbot**



***Figure. 6 Insurance Chatbot System Architecture***

The system proposed here in Figure 6 aims to adequately respond to insurance-related questions by using a multi-stage natural language processing pipeline. The dataset collection is the initial stage, where we utilize the Insurance QA Corpus consisting solely of insurance-domain questions. Since this stage is a preparatory one, there is no official evaluation metric used.

During the data preprocessing process, input questions are subjected to text normalization, tokenization, and noise removal. Quality of preprocessing is assessed with:

Token Statistics like average token length and standard deviation assist in tracking the text processing consistency.

Then, the filtered questions are input to a sentence embedding model employing a pre-trained Sentence Transformer and converted into dense vector representations from textual questions. Embedding quality is ensured via cosine similarity verification and qualitative checking by hand for semantic preservation. For domain-specificity improvement, the sentence transformer is fine-tuned on decoded insurance questions. The training process is monitored by measurements like training and validation loss, whereas post-training similarity measurements—Pearson and Spearman cosine similarity—and classification accuracy offer an indication of embedding alignment with semantic intentions.

Each Qi is represented as a vector encoding q i ∈Rn. To measure embedding quality, we calculate cosine similarity between vectors as given below in equation:

A higher similarity score indicates semantic closeness.

After embedding, questions and their vector representations are retained in an SQLite database to enable efficient retrieval. The quality of the database is evaluated through latency metrics, coverage analysis to verify complete data entry storage, and redundancy checks for removal of duplicates. A memory component, again utilizing SQLite, enables quick query-response matching from past interactions, measured through hit rate and response time. The system supports both voice and text-based user queries, where voice input is converted into text using a speech-to-text engine. The performance of this module is measured in terms of Word Error Rate (WER) and processing speed. Testing involves Query Latency, Time taken to retrieve results and Redundancy Check as depicted in the equation below which ensure uniqueness using.

The memory mechanism allows retrieval of previous interactions. For a given user query, a match is attempted with historical queries using embedding similarity. Performance is evaluated using

Response Time: Time from input query to returned response.

User query processing accepts text or voice inputs. For voice input, speech-to-text conversion accuracy is measured using:

Where:

* *S*: Substitutions,
* *D*: Deletions,
* *I*: Insertions,
* *N*: Total words in the reference.

For each incoming query, the system first attempts a memory search to retrieve pre-existing responses. This is evaluated using Recall@K and memory recall metrics to assess how accurately the system fetches relevant past responses. If no exact memory match is found, the system performs semantic search using cosine similarity between the user query vector *￼* and all stored embeddings. For each incoming query, the system first attempts a memory search to retrieve pre-existing responses. This is evaluated using memory recall metrics to assess how accurately the system fetches relevant past responses. If no exact match is found, the system proceeds to semantic search, computing cosine similarity between the user query and stored question embeddings to identify the most semantically aligned candidates. Search performance is measured via precision, recall, Top-N accuracy, and Mean Reciprocal Rank (MRR).

Upon identifying the most relevant question(s), the system generates an appropriate response using the OpenRouter (DeepSeek R1) language model. This generation process is rigorously evaluated using NLP metrics such as BLEU, ROUGE, and F1-score, ensuring grammatical fluency, content relevance, and lexical overlap with expected answers. The generated responses, along with corresponding user queries, are then stored in the memory database for future recall, promoting conversational continuity.

These generated responses, with accompanying user questions, are thereafter archived in the memory database to be recalled in the future to foster conversational continuity. Last, the response is presented to the user, performance measures comprising response time and user satisfaction, the latter being measured through direct feedback or survey processes. Such an all-embracing approach ensures an efficient, intelligent, and contextually relevant insurance query solution system.

The Evaluator Agent in the given system is aimed at conducting qualitative evaluations of responses from chatbots based on a domain-adapted large language model (LLM). This agent is triggered with real-life user queries and their respective chatbot-provided answers to judge the responses in terms of coherence, domain adequacy, informativeness, and linguistic simplicity. By tapping into the reasoning ability of a cutting-edge LLM, the agent mimics expert human judgment to ascertain whether the chatbot response aligns with user expectations in an insurance advisory setting.

To enable this assessment, a collection of domain-specific questions was assembled to cover a wide range of actual-world information requirements. The questions were formulated to challenge the system in different types of insurance—like health, life, term, and motor insurance—and comprised procedural as well as conceptual questions. These questions were fed into the Evaluator Agent along with the responses generated, and the agent evaluated each pair on the basis of semantic consistency and adequacy.

Prompt Examples Used for Evaluation:

Prompt 1: *How do I file a car insurance claim online?*  
Prompt 2: *What documents are needed for a health insurance claim?*  
Prompt 3: *How can I pay my life insurance premium via UPI?*  
Prompt 4: *How do I add a family member to my health insurance policy?*  
Prompt 5: *What riders can I add to my term insurance plan?*  
Prompt 6: *How do I update my nominee details?*  
Prompt 7: *Can you explain the waiting period in health insurance in simple terms?*  
Prompt 8: *Is life insurance mandatory in India?*  
Prompt 9: *What are the tax benefits of health insurance?*  
Prompt 10: *Can I transfer my bike insurance to a new owner?*

These queries were specifically selected to challenge the system across multiple dimensions such as policy understanding, procedural guidance, document-related requirements, and regulatory knowledge. The Evaluator Agent plays a crucial role in measuring how effectively the system generalizes across these different query types and maintains accuracy in a highly sensitive domain like insurance.

1. **Policy Document Retrieval**

The system being proposed in Figure 7 starts off with the data preparation stage, where policy files in PDF form are gathered. These files are preprocessed to obtain their text content using the PyMuPDF (fitz) library. The obtained text is then divided into sentences with the help of the SpaCy NLP library, allowing more meaningful and tractable chunks for downstream processing. These sentence-level chunks are clumped into contextually consistent chunks, making sure that enough information is retained for semantic comprehension upon retrieval.

After chunking the textual data, the subsequent step is embedding generation. In this step, every chunk is fed into a domain-specific transformer model, llmware/industry-bert-insurance-v0.1, which is specifically fine-tuned for the insurance domain. The model generates contextual embeddings for every chunk, which are then mean pooled and normalized using L2 normalization methods available in the FAISS library. These normalized chunk vectors represent the semantic content of the policy text and are stored for similarity-based retrieval.

During vector indexing, these chunk embeddings are loaded into a FAISS index (IndexFlatIP) that enables fast similarity search via inner product calculations, essentially computing cosine similarity given the pre-normalization step. In addition to the FAISS index, a policy\_chunk\_map is kept in place that maps every embedding back to its associated text segment so that original content retrieval is accurate. This framework enables rapid lookup of the most suitable content for a given user query.

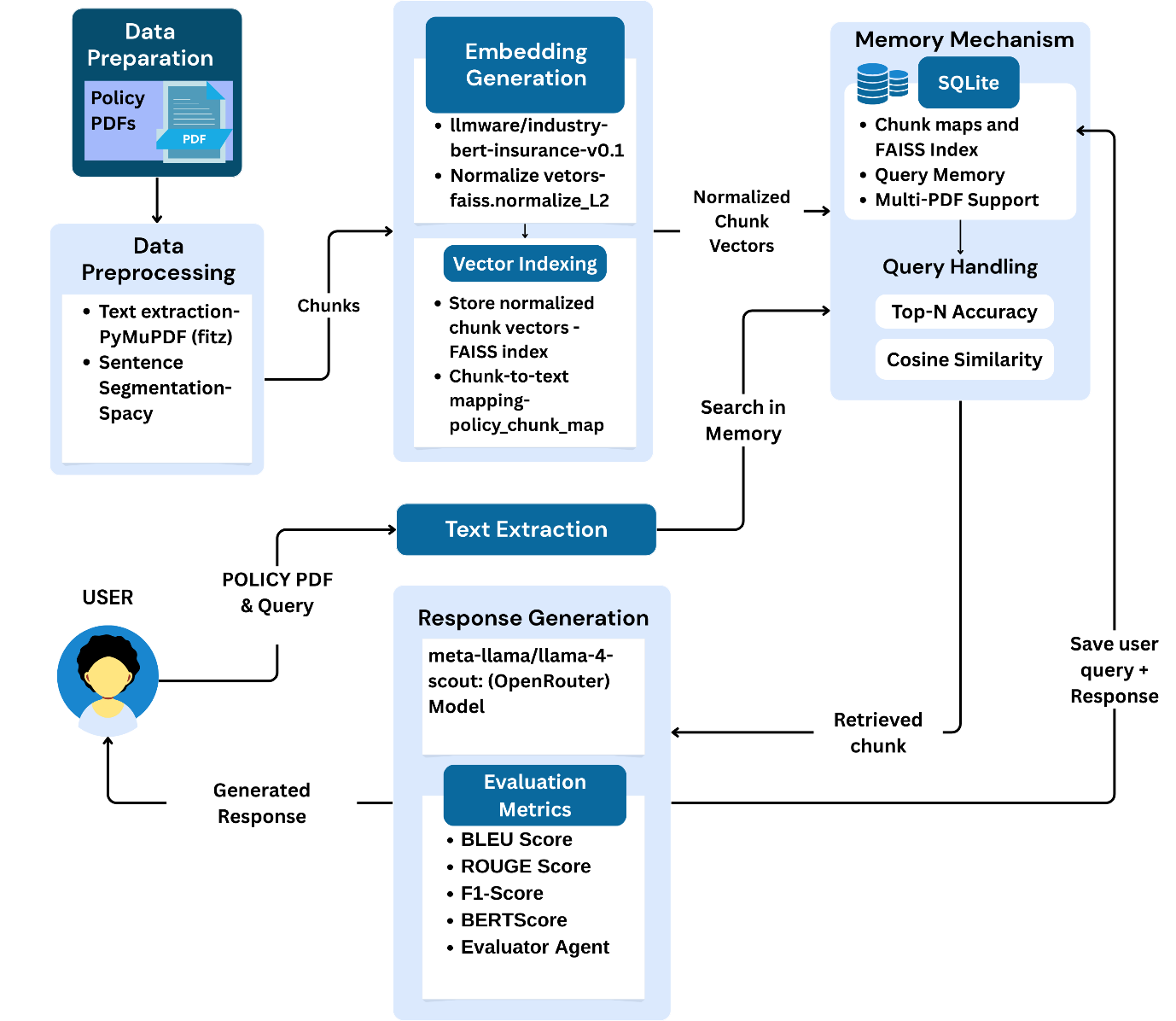
****

Figure 7 Policy Document Retrieval System Architecture

The memory mechanism is backed by a lightweight SQLite database that stores the FAISS index, policy-to-chunk mappings, and query-response pairs. This module also facilitates memory features like support for multiple policy documents, query tracking, and historical search accuracy metrics such as top-N accuracy and cosine similarity scores. It guarantees persistent and contextual awareness over user sessions.

If the user provides a policy PDF and a natural language query, the system first looks for the matching FAISS index and chunk mapping of the policy in the memory. If it does not find one, the document goes through the entire preprocessing and indexing pipeline. The user query is encoded by the same transformer model and normalized. The query vector that results is compared with the FAISS index to find the top-k most similar chunks.

The chunk thus retrieved is fed to the response generation module, which makes use of a high-performing LLM, namely the meta-llama/llama-4-scout model hosted on OpenRouter. This model is asked to create a plain language summary and explanation of the retrieved clause, put in context to clearly and informatively answer the user's question. The response thus generated is returned to the user as the output.

For system evaluation, the generated responses are quantitatively evaluated in terms of quality using common NLP metrics like bleu score, rouge score, f1-score, and bertscore. Apart from that, an evaluator agent is also integrated in order to give a qualitative measure of the answers. These measures guarantee the reliability, relevance, and interpretability of the system's outputs, thus proving the efficacy of the retrieval-augmented generation framework in addressing complex insurance-related questions. Overall, this methodology combines semantic vector search, domain-specific transformer models, and advanced large language models in a Retrieval-Augmented Generation (RAG) pipeline, enabling accurate, explainable, and user-friendly responses to policy-related queries.

In the policy document retrieval system, the evaluator agent is employed for judging the quality of the produced responses based on how well the retrieved chunk in the insurance PDF covers the user's question. This helps ensure that not only is the response semantically close to the question, but it is also accurate and helpful to the user. After a chunk is fetched through FAISS-based search on the vectors and the matching LLM-produced summary is generated, the output is assessed by the evaluator agent through a guided prompt aimed at capturing qualitative feedback along significant dimensions.

The instruction asks the evaluator agent to examine the summary produced by the LLM and score it on four dimensions—relevance, accuracy, clarity, and helpfulness—using a 0-to-5 scale. The agent also supplies a brief remark justifying its decision, besides these scores. This approach allows for automated scoring that simulates human-like evaluation, which improves the reliability and efficiency of the document retrieval system.

The exact prompt used in this evaluation is:

Prompt:  
“LLM-generated Summary:  
{summary}

Rate the following from 0 to 5:  
Relevance (does the chunk answer the query?)  
Accuracy (is the summary faithful?)  
Clarity (is it understandable?)  
Helpfulness (would a user find it useful?)

Also write a short evaluator comment.

Respond like:  
Relevance: 4  
Accuracy: 3  
Clarity: 5  
Helpfulness: 4”

The queries used in the document retrieval evaluator agent to test the system on real-world insurance policy concerns include:

Prompt 1: *Is newborn baby covered under this plan?*  
Prompt 2: *What is the waiting period for pre-existing diseases?*  
Prompt 3*: Does this policy cover maternity expenses?*  
Prompt 4: *Are day care procedures included?*  
Prompt 5: *What is the cashless hospital network?*  
Prompt 6: *Is OPD treatment reimbursable?*  
Prompt 7: *What is the coverage amount for critical illness?*  
Prompt 8: *Are ambulance charges covered?*  
Prompt 9: *Is there a claim settlement ratio mentioned?*  
Prompt 10: *What documents are needed for claim filing?*

These queries help evaluate how well the document retrieval system can locate the appropriate sections of a policy document and generate coherent, accurate summaries that are useful for end-users seeking specific policy information.

1. **Evaluation Metrics**

The effectiveness of the system is quantitatively evaluated using well-established Natural Language Generation (NLG) and Information Retrieval (IR) metrics. These include BLEU, ROUGE, and Precision, Recall, and F1-score derived from the ROUGE evaluation suite. The same evaluation metrics are used for Policy Recommendation, Document Retrieval and Insurance Chatbot.

**BLEU (Bilingual Evaluation Understudy)**

BLEU measures the degree of n-gram overlap between the system-generated output and a reference response. It rewards precision in generated segments while penalizing overly short responses using a brevity penalty. The BLEU score is calculated as seen in equation():

Where:

* ​ r epresents the precision of n-grams
* is the weight assigned to each n-gram level (commonly uniform)
* denotes the brevity penalty.

**ROUGE-L (Longest Common Subsequence)**

ROUGE-L is used to evaluate the quality of generated text based on the longest common subsequence (LCS) shared with the reference. This metric considers both precision and recall, and the F1-score is derived accordingly as seen in equation():

Here, LCS(X,Y) is the length of the longest common subsequence between the generated output X and reference text Y.

1. **Results**

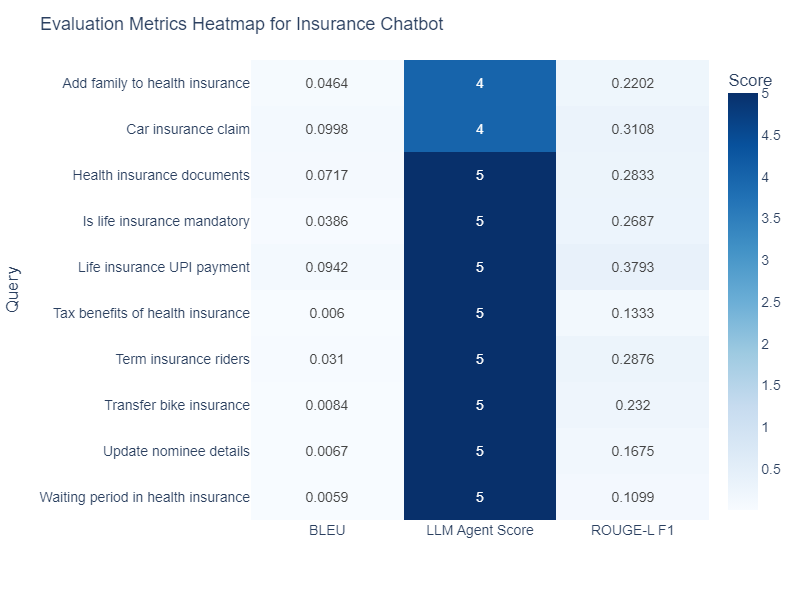


Figure 8 Evaluation Metrics of Insurance Chatbot

Figure 8 shows a heatmap of the performance of our chatbot on ten representative insurance‐related questions, with BLEU score, ROUGE‑L F1, and LLM Agent Score as the assessment axes. The largely cool blue colors in the BLEU column (values <0.10) highlight regularly low word‐overlap with reference responses, which indicates the model's paraphrasing over emulating exact phrasing. Conversely, the ROUGE-L F1 column shows mid-range blues (0.11–0.38), which means moderate sequence-level overlap and partial capture of important phrases. Lastly, the LLM Agent Score column is distinct in deep navy (scores of 4–5 out of 5), reflecting high human-rated relevance and adequacy of responses. These trends together serve to indicate that although lexical measures such as BLEU underestimate the performance of the chatbot, both ROUGE-L and particularly human evaluation establish that the system consistently delivers correct information in a natural, conversational manner.

To evaluate the retrieval capability of the policy recommendation system, we conducted experiments using 10 diverse insurance-related queries. The FAISS index and SentenceTransformer embeddings were used to retrieve the top-5 most relevant policy documents for each query based on vector similarity (L2 distance). Below are some sample retrieval results:

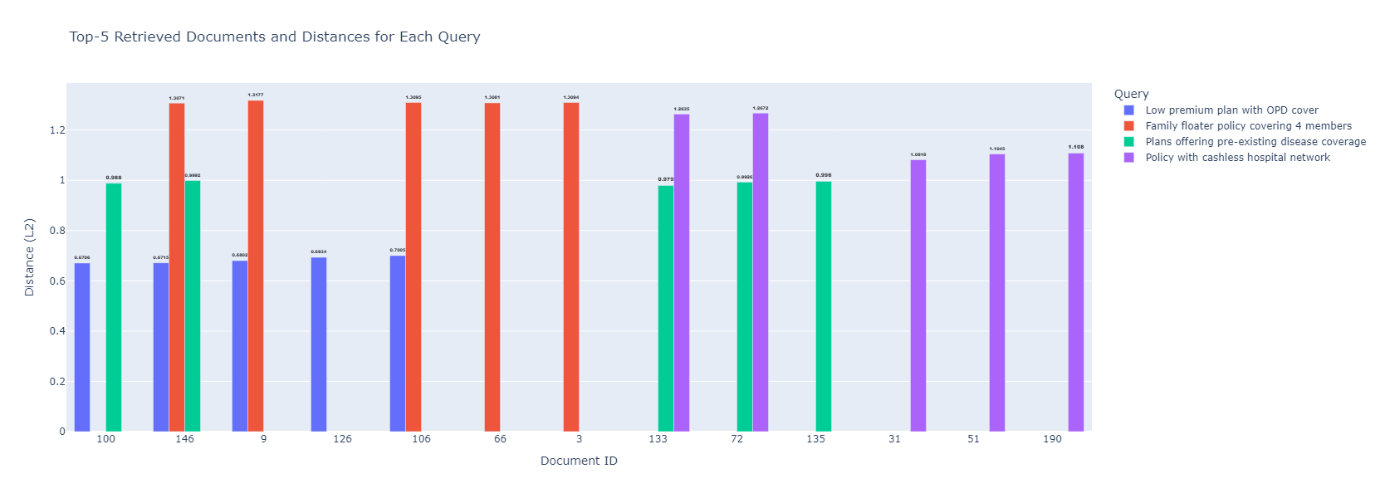


Figure 9 Sample Policy Retrieval Results

The retrieval process is based on semantic similarity between user queries and policy descriptions, where a lower distance value indicated a higher semantic relevance to the query. For instance, policies related to "Low premium plan with OPD cover" achieved distance scores as low as 0.6706, reflecting strong retrieval performance. Similar patterns were observed for other queries such as "Plans offering pre-existing disease coverage" and "Family floater policy covering 4 members," demonstrating the system's ability to retrieve relevant policies effectively across diverse user needs.

The FAISS-based retrieval evaluation demonstrated the effectiveness of the policy recommendation system in retrieving relevant insurance policies, the system achieved a Precision@5 score of 0.333, a Recall@5 of 0.833, an NDCG@5 of 0.69, and a perfect Hit@5 score of 1.0, indicating robust retrieval performance with all queries successfully retrieving at least one relevant policy.

#### **Table 3: Evaluation Metrics for Policy recommendation RAG**

|  |  |
| --- | --- |
| Evaluation Metric | Value |
| BLEU Score | 0.0 |
| ROUGE-L | 8.74 |
| BERT Precision | 0.819 |
| BERT Recall | 0.861 |
| BERT F1 Score | 0.84 |

It can be observed from Table 3 that the LLM-RAG evaluation on 10 queries revealed insightful findings across multiple metrics. Despite a BLEU score of 0.0, suggesting low exact n-gram overlap between the generated and reference answers, the ROUGE-L score of 8.74 indicated a reasonable level of structural similarity. More importantly, the semantic evaluation using BERTScore demonstrated strong performance, with a BERT Precision of 0.819, BERT Recall of 0.861, and a BERT F1 score of 0.84. These high BERTScore values confirm that the generated responses were contextually and semantically close to the reference answers, even if the exact wording differed, showcasing the LLM's ability to produce meaningful and relevant recommendations.

Table 4. Evaluation Metrics for Document Retrieval

|  |  |
| --- | --- |
| Evaluation Metric | Average Value |
| BLEU Score | 0.0704 |
| ROUGE-L F1 | 0.3167 |
| Token F1 Score | 0.3777 |
| BERTScore F1 | 0.8443 |

The evaluator agent based on knowledge retrieval from a document is evaluated on user-uploaded documents with related queries. Results showed in Table 4 conveys a low BLEU score (0.0704) and moderate ROUGE-L F1 (0.3167) and Token F1 (0.3777), indicating variation in phrasing. However, a high BERTScore F1 (0.8443) confirmed strong semantic alignment between generated and reference answers. Thus, the system effectively captures meaning even when wording differs.

1. **Conclusion**

The overall finding of this project emphasizes the effective integration of domain-specific language models, semantic retrieval methods, and large language model-based generation to develop an intelligent insurance aid system. The system successfully solves three key tasks: natural language query handling through a chatbot, personalized policy recommendation through retrieval-augmented generation, and accurate document clause retrieval with explanatory summarization. By using a highly fine-tuned insurance-specific Sentence Transformer model to embed queries as well as documents, together with FAISS to perform similarity search efficiently, the system guarantees quick and accurate return of relevant policy details. Additionally, applications of OpenRouter-hosted big language models such as LLaMA to summarise and generate natural language responses greatly improve the readability and usability of responses for end-users.

The evaluator agent module provides an added layer of validation, replicating human judgment to measure the system-generated answers on relevance, accuracy, clarity, and helpfulness. This feedback process not only guarantees consistent behavior but also facilitates iterative refinement of the system. The outcomes of this project illustrate the feasibility of employing LLMs and RAG-based methods in highly controlled and paper-intensive sectors such as insurance, opening the way for more personal, more transparent, and more user-friendly online insurance services. By the synthesis of NLP, IR, and LLM methods, the project provides an efficient and scalable system for domain-specific question answering and document understanding, with applications widely applicable in various customer-oriented industries.

**References:**

[1] Express Healthcare, “The future of health insurance: Integrating AI for fair pricing and better coverag,” Express Healthcare.

[2] A. Poddar, “The State of Insurance and Healthcare in India,” PlumHQ.

[3] S. Kolambe and P. Kaur, “Insurance Policy Summarization Using Natural Language Processing (NLP),” *Frontiers in Health Informatics*, vol. 13, pp. 4774–4782, Apr. 2024.

[4] Z. Zhang *et al.*, “Personalization of large language models: A survey,” *arXiv preprint arXiv:2411.00027*, 2024.

[5] S. Prabhune and D. J. Berndt, “Deploying Large Language Models With Retrieval Augmented Generation,” *arXiv preprint arXiv:2411.11895*, 2024.

[6] H. : Frank, K. Ämnesgranskare, D. Vega D’aurelio, E. Andrésdóttir, E. Furberg, and J. Hedberg, “Uppsala/Visby Automated extraction of insurance policy information.”

[7] M. Nuruzzaman and O. K. Hussain, “IntelliBot: A Dialogue-based chatbot for the insurance industry,” *Knowl Based Syst*, vol. 196, p. 105810, 2020.

[8] W. Zhang, J. Shi, X. Wang, and H. Wynn, “AI-powered decision-making in facilitating insurance claim dispute resolution,” *Ann Oper Res*, pp. 1–30, 2023.

[9] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, “Applying deep learning to answer selection: A study and an open task,” in *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, 2015, pp. 813–820.

[10] C. Cao *et al.*, “LLMs for Insurance: Opportunities, Challenges and Concerns,” 2024.

[11] B. Chandra, P. Preethika, S. Challagundla, and Y. Gogireddy, “End-to-End Neural Embedding Pipeline for Large-Scale PDF Document Retrieval Using Distributed FAISS and Sentence Transformer Models,” *Journal ID*, vol. 1004, p. 1429.

[12] S. Bedi *et al.*, “A systematic review of testing and evaluation of healthcare applications of large language models (LLMs),” *medRxiv*, p. 2024, 2024.

[13] R. P. Pramila, S. Varsha, and others, “Artificial Intelligence and Natural Language Processing (NLP) Integrated Multilingual Health Insurance Application,” in *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2024, pp. 1141–1145.

[14] N. P. KS, S. Sudhanva, T. N. Tarun, Y. Yuvraaj, and D. A. Vishal, “Conversational chatbot builder–smarter virtual assistance with domain specific AI,” in *2023 4th International Conference for Emerging Technology (INCET)*, 2023, pp. 1–4.

[15] S. Ayanouz, B. A. Abdelhakim, and M. Benhmed, “A smart chatbot architecture based NLP and machine learning for health care assistance,” in *Proceedings of the 3rd international conference on networking, information systems & security*, 2020, pp. 1–6.

[16] J. Genesis and F. Keane, “Integrating Knowledge Retrieval with Generation: A Comprehensive Survey of RAG Models in NLP,” 2025.