# Enhanced Doctor Recommendation with Explore and Exploit Algorithms based on Patient Symptom Descriptions and Fine-tuned Language Models

Keshav Chandak
*IMT2021003*

Sunny Kaushik
*IMT2021007*

Tanmay Jain
*IMT2021015*

Suyash Chavan
*IMT2021048*

***Abstract*—A new method for automatic doctor selection, based on the description of patients' symptoms is presented in this paper. Through fine-tuning, the DistilGPT-2 model was calibrated using 112,000 conversations between doctors and their patients. The intention behind this undertaking was to enable patients to explain their symptoms so that they would be able to communicate more effectively with healthcare providers. After that, we applied the SpaCy library in order to determine keywords from the language model's output. To achieve mapping of these keywords onto appropriate doctor specialties, a weighted combination of Rouge-L and Rouge-1 scores were used. In our methodology, we have three selected top specialist fields as the closest match to patient symptoms. It should also be noted that what we are doing is pioneering in this area by employing Explore and Exploit algorithms like MultiArm Bandits, Upper confidence bound (UCB), Thompson Sampling and Contextual Bandits for recommending physicians thereby implying a direction towards better medical practice and care provision at large. The code are available here:GitHub**

***Index Terms*—Multi-Arm Bandits, Upper Confidence Bound, SpaCy, Thompson Sampling, BERTScore**

## I. Introduction

Effective communication between patients and healthcare providers is fundamental to ensuring accurate diagnosis and appropriate treatment. However, articulating symptoms accurately can be challenging for patients, potentially leading to miscommunication and suboptimal healthcare outcomes. In this context, applying the advances in artificial intelligence (AI) and natural language processing (NLP) presents a promising opportunity to streamline the process of symptom description and doctor recommendation, ultimately enhancing the quality of healthcare delivery.

In this paper, we propose a novel approach to automate the task of doctor recommendation based on patient-provided symptom descriptions. Traditional doctor recommendation systems often rely on structured data such as medical records or user ratings, which may not capture the nuances of patient symptoms expressed in natural language. But our methodology overcomes these limitations because it adopts sophisticated NLP techniques and machine learning.

Central to our approach is the fine-tuning of the DistilGPT-2 language model on a comprehensive dataset of patient-doctor conversations. This fine-tuning process enables the language model to understand and generate responses tailored to healthcare-related queries. By training on a large corpus of real-world interactions, the model learns to recognize patterns and understand the context in which symptoms are described, thus facilitating more accurate and meaningful communication between patients and healthcare providers.

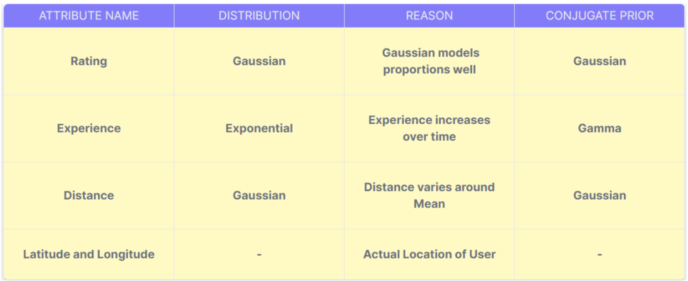
Then, we employ the SpaCy library for keyword extraction from the output generated by the language model. These extracted keywords serve as the foundation for identifying the most relevant doctor specialties corresponding to the patient's symptoms. Leveraging a combination of Rouge-1 and Rouge-L scores, we map these keywords to specialist descriptions, allowing us to recommend the top three specialist fields that are most closely aligned with the patient's condition.

Furthermore, our approach extends beyond traditional recommendation systems by incorporating explore and exploit-based algorithms such as Multi-Armed Bandit (MAB), Upper Confidence Bound (UCB), and Thompson Sampling. These algorithms are renowned for their ability to balance exploration of new options with exploitation of known information, enabling more efficient and effective doctor recommendations. By dynamically adapting to patient preferences and feedback, these algorithms ensure that patients are matched with the most suitable healthcare providers based on their unique needs and preferences.

## II. Dataset

To ensure accuracy, we opted to scrape genuine data instead of relying on pre-existing datasets. Utilizing **BeautifulSoup** library, we extracted descriptions of doctor specialties and subspecialties from [1] and [2] . This method provided total 38 specialists with their descriptions covering various domains of conditions and diseases.

The doctors dataset consisted of in total of 3 parameters related to doctor which are measured relative and comparable with the other doctors. The three attributes in the doctor's feature were average ratings over the time given to that doctor, doctor's experience and the doctor's location. In the construction of our synthetic dataset for the doctor recommendation system, different statistical distributions were chosen based on the nature of the data they represent. This section explains the reasoning behind the choice of distributions for each attribute: ratings, experience, and distance.



| ATTRIBUTE NAME | DISTRIBUTION | REASON | CONJUGATE PRIOR |
|---|---|---|---|
| Rating | Gaussian | Gaussian models proportions well | Gaussian |
| Experience | Exponential | Experience increases over time | Gamma |
| Distance | Gaussian | Distance varies around Mean | Gaussian |
| Latitude and Longitude | - | Actual Location of User | - |

Fig. 1: Doctors' Dataset

### A. Ratings

Initially, ratings were generated using a Beta distribution, which is typically used for variables that are bounded within an interval, such as ratings that might range from 0 to 10. However, we transitioned to using a Gaussian (normal) distribution. The rationale for this change was driven by the desire to reflect more realistically how ratings are typically distributed around a mean value in real-world scenarios, where most ratings cluster around a central value with fewer at the extremes (highly positive or negative). This is formally expressed as:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where $\mu$ is the mean and $\sigma^2$ is the variance of the ratings.

### B. Experience

For the experience of doctors, we employed an Exponential distribution. This decision is based on the understanding that the likelihood of encountering doctors with very high experience is lower than that of encountering those with lower experience. The exponential distribution is suitable for modeling the time between events in a process where events happen continuously and independently at a constant average rate. It captures the decay in the probability of finding doctors with increasing years of experience, which aligns well with the observation that not all experience contributes equally to expertise:

$$f(x \mid \lambda) = \lambda e^{-\lambda x}$$

where $\lambda$ denotes the rate parameter.

### C. Distance

The Gaussian distribution was also chosen for modeling the distance attribute, considering that most patients or medical facilities are likely to be concentrated around certain hubs or regions. This distribution allows us to simulate the scenario where most distances are average, with fewer occurrences of very short or very long distances. This assumption is practical in urban planning and healthcare accessibility studies, where distances usually follow a normal distribution around a central location:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

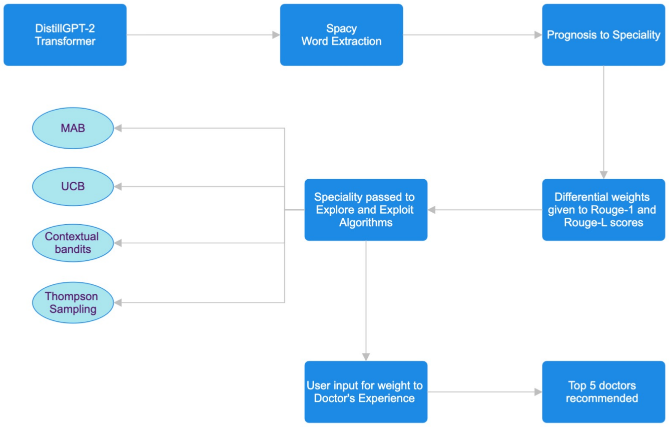

DistillGPT-2 Transformer
Spacy Word Extraction
Prognosis to Speciality
MAB
UCB
Contextual bandits
Thompson Sampling
Speciality passed to Explore and Exploit Algorithms
Differential weights given to Rouge-1 and Rouge-L scores
User input for weight to Doctor's Experience
Top 5 doctors recommended

Fig. 2: Overall Methodology

## III. Methodology

Our methodology comprises several key components aimed at automating the process of doctor recommendation based on patient-provided symptom descriptions. We detail each step below and in figures 2 and 3:

### A. LLM Fine Tuning

We began by gathering a large dataset of patient-doctor conversations from the HealthCareMagic-100k dataset available on Hugging Face. This dataset consists of 112,000 anonymized interactions, comprising a diverse range of medical queries and responses. Each conversation contains a patient's description of their symptoms and the corresponding doctor's expert advice or prognosis. To enable the language model to understand and generate relevant responses to patient symptom descriptions, we employ transfer learning techniques. Specifically, we fine-tune the DistilGPT-2 language model on 80% of the collected dataset of patient-doctor conversations. This fine-tuning process adapts the pre-trained language model to the domain-specific language and context of healthcare conversations. Patients provide descriptions of their symptoms in natural language sentences, which are entered into the fine-tuned language model for processing. The language model generates responses based on the input symptoms, providing insights and potential diagnosis.

The rest of 20% data is used as test data for calculating BERTScore which utilises n-grams for keyword matching between the prognosis result from the LLM and the ground truth given in the huggingface data.

Table I shows the average precision, f1 and recall from the BERTScore. Table II shows the average similarity score between the ground truth and our LLM prognosis output.

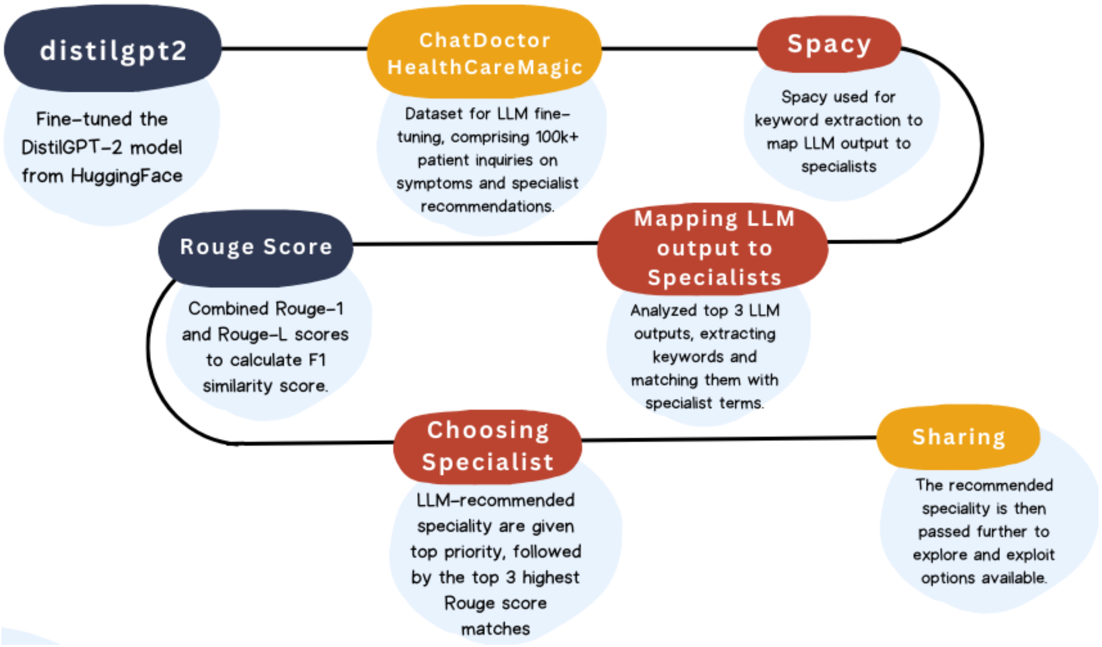

distilgpt2
Fine-tuned the DistilGPT-2 model from HuggingFace
ChatDoctor HealthCareMagic
Dataset for LLM fine-tuning, comprising 100k+ patient inquiries on symptoms and specialist recommendations.
Spacy
Spacy used for keyword extraction to map LLM output to specialists
Rouge Score
Combined Rouge-1 and Rouge-L scores to calculate F1 similarity score.
Mapping LLM output to Specialists
Analyzed top 3 LLM outputs, extracting keywords and matching them with specialist terms.
Choosing Specialist
LLM-recommended speciality are given top priority, followed by the top 3 highest Rouge score matches
Sharing
The recommended speciality is then passed further to explore and exploit options available.

Fig. 3: Summary of choosing most suitable specialists

| Metric | Value |
|---|---|
| Precision | 0.5541 |
| Recall | 0.5389 |
| F1 Score | 0.5449 |

TABLE I: Performance Metrics

| Metric | Value |
|---|---|
| Cosine Similarity | 0.4172 |
| Pearson Correlation Coefficient | 0.2867 |
| KL Divergence | 22.0280 |

TABLE II: Other Metrics

### *B. Keyword extraction using SpaCy*

To map the prognosis text to the most relevant specialist description text, simple similarity metrics like Cosine Similarity, Pearson's Correlation Coefficient, etc didn't work well because of the bias towards longer specialty descriptions, as similarity metrics take into account the dot product with each word, thus adding to the score.

Thus, we had utilised SpaCy to extract relevant keywords such as nouns, adjectives and adverbs from the prognosis result and specialty descriptions of all 38 specialties, for ROUGE score calculations as given in the next part.

### *C. Specialist Mapping using ROUGE Scores*

In our methodology we have used the ROUGE Score as a metric for evaluation. (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used in natural language processing tasks to assess the similarity between text documents.

*1) ROUGE-1 Score:* Rouge-1 evaluates the overlap of 1-grams (individual words) between the extracted keywords and the extracted kwywords from the descriptions of specialist fields. It calculates the precision, recall, and F1-score based on the number of overlapping unigrams between the two sets of text.

*2) ROUGE-L Score:* Rouge-L, on the other hand, considers the longest common subsequence (LCS) between the extracted keywords and the specialist descriptions. It evaluates the recall of the LCS, which represents the longest sequence of words that appears in both sets of text.

In our methodology, after extracting keywords the prognosis given by fine tuned LLM, we employ a **weighted** combination of Rouge-1 and Rouge-L scores to determine the relevance of these keywords to specialist descriptions as follows:

$$\text{Weighted Rouge Score} = 0.3 \times \text{Rouge-L} + 0.7 \times \text{Rouge-1}$$

This approach ensures that both the overlap and linguistic similarity between the extracted keywords and specialist descriptions are taken into account. Further, the weighted ROUGE Score was calculated for every Specialist's description keywords and the prognosis text's keywords.

### *D. Recommending the Specialists*

Top 3 highest score Specialities were chosen for recommendation. Further, if a doctor speciality was recommended in the prognosis text, it was given the highest weightage. To prevent irrelevant specialities from being recommended, we chose a threshold such that if the absolute difference between the scores of those specialities was greater than the threshold then they weren't recommended.

### *E. Mapping of Specialities to the doctor*

After the prediction of the top 3 specialities then we are running the explore and exploit algorithms to recommend the best doctor for the speciality or the list of specialities returned based on the calculated Rouge Score as described in the section-C. We have applied a total of 4 algorithms to predict the best doctor. While predicting the doctor the user needs to enter their age and their preference to the experience of the doctor, as the experience of the doctor as a matter of importance varies from user to user. The four algorithms will be discussed in the detail further along with the result analysis and their comparison.

## IV. Explore and Exploit Algorithms

### *A. Multi-Armed Bandits*

The Multi-Armed Bandit (MAB) model implemented for the doctor recommendation system employs a balance between exploration and exploitation strategies. This balance is controlled by the parameter $\epsilon$, which influences the decision-making process of the bandit algorithm.

*1) Algorithm Description:* The algorithm initializes by normalizing the ratings and experience values of doctors to ensure a fair comparison between these parameters. Composite scores are computed as a weighted sum of these normalized values:

$$\begin{aligned}\text{Composite Score} = {} & \text{Normalized Ratings} \times w_r \\ & + \text{Normalized Experience} \times w_e\end{aligned} \tag{1}$$

where $w_r$ and $w_e$ are the weights for ratings and experience, respectively. Each doctor's score is then used to rank them, and decisions are made based on the epsilon-greedy policy:

- With probability $\epsilon$, the algorithm explores by randomly selecting a doctor.
- With probability $1-\epsilon$, the algorithm exploits by selecting the doctor with the highest composite score not yet chosen in the current recommendation round.

*2) Effect of Varying $\epsilon$:* The value of $\epsilon$ crucially affects the performance of the MAB algorithm. A higher $\epsilon$ increases the frequency of exploration, leading to more diverse recommendations but potentially lower immediate rewards (ratings). Conversely, a lower $\epsilon$ focuses on exploiting known high-performing doctors based on past interactions, which can reinforce biases or result in a less diverse set of recommendations.

*3) Analysis of Ratings and Experience Parameters:* In this implementation, the impact of doctors' ratings and experience on the recommendation process is modulated by their respective weights ($w_r$ and $w_e$). As the weight on experience ($w_e$) increases, the system tends to favor doctors with more experience over those with potentially higher ratings but less experience. This approach can be particularly effective in scenarios where the reliability and depth of knowledge resulting from more extensive experience are critical. However, this might also overlook younger doctors with high competence but less experience. The choice of $\epsilon$ and the weights assigned to ratings versus experience significantly influence the behavior of the MAB system. Optimizing these parameters based on the specific needs and values of the healthcare provider can lead to an effective balance in doctor recommendations, enhancing patient satisfaction and trust in the healthcare system.

*4) Recommendation Results:* When $\epsilon = 0.5$, the system balances between exploring new options and exploiting known data. The results include recommendations for doctors that might not always have the highest immediate ratings but offer a potential for discovering under-appreciated talent. As $\epsilon$ approaches 1, exploration dominates, increasing the variability in doctor selection and potentially leading to new insights into the available medical expertise.

### *B. Upper Confidence Bound (UCB) Algorithm*

The Upper Confidence Bound (UCB) algorithm is utilized within our doctor recommendation system to effectively balance between exploiting known data and exploring lesser-known options, thus enhancing the robustness of the system in making recommendations. This section details the UCB algorithm's implementation and its operational dynamics within the context of healthcare recommendations.

*1) Algorithm Initialization:* At the outset, the UCB algorithm normalizes both the ratings and experience data of doctors to ensure a level playing field:

$$\text{Normalized Ratings} = \frac{\text{Ratings} - \min(\text{Ratings})}{\max(\text{Ratings}) - \min(\text{Ratings})}$$

$$\text{Normalized Experience} = \frac{\text{Experience} - \min(\text{Experience})}{\max(\text{Experience}) - \min(\text{Experience})}$$

Composite scores are then computed using a weighted sum of these normalized metrics: where $w_r$ and $w_e$ represent the weights for ratings and experience, respectively.

*2) Recommendation Process:* For each specialist requested, the algorithm assesses all associated doctors. The UCB for each doctor is calculated based on their historical performance and exploration factor, determined as follows:

$$\text{UCB} = \text{Average Composite Score} + \sqrt{\frac{2\ln(\text{Total Counts}+1)}{\text{Count of Doctor}}}$$

This formula ensures that doctors who have been less explored (i.e., chosen fewer times) have a higher potential UCB, prompting the system to explore them.

Here in the formula for UCB in the exploration part where the logarithm is there. We have done (t+1) to ensure that in case of t=0, it doesn't reach -infinity and this is called confidence interval adjustment.

*3) Selection and Updates:* Doctors are ranked according to their UCB scores, and the top five are selected for recommendation. This method ensures a mix of both highly rated and potentially underrated doctors being recommended. Following each cycle of recommendations, the model updates the selection count and the cumulative composite score for each selected doctor, thereby refining future recommendations.

*4) Dynamic Adaptation:* As more data accumulates through user interactions, the UCB algorithm continuously updates its parameters, thereby improving its accuracy and reliability. The exploration component naturally decreases as the confidence in the doctors' scores increases, allowing the system to shift gradually from exploration to exploitation.The UCB algorithm, by its nature, adapts over time, becoming more efficient as it learns from each interaction. It is particularly suited for environments like medical recommendations, where the stakes are high, and the diversity of options is vast. This adaptive approach helps ensure that patients receive the most reliable and tested recommendations tailored to their specific medical needs.

## C. Thompson Sampling

A modified form of Thompson Sampling is employed to discover multiple distribution structures, such as rating, experience, and distance. Thompson Sampling effectively explores data, providing a variety of optimal results. This section will detail the modified Thompson Sampling algorithm and its functionality.

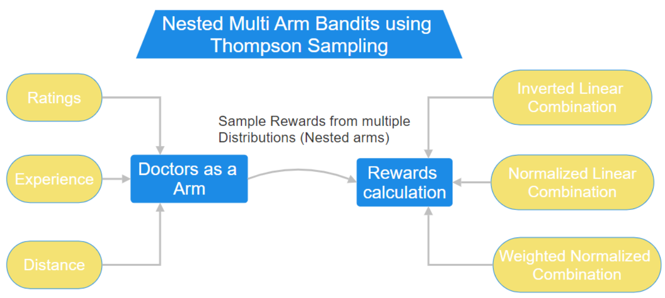

Nested Multi Arm Bandits using Thompson Sampling
Ratings
Experience
Distance
Doctors as a Arm
Sample Rewards from multiple Distributions (Nested arms)
Rewards calculation
Inverted Linear Combination
Normalized Linear Combination
Weighted Normalized Combination

Fig. 4: Thompson Sampling

*1) Description of Arms:* Here, every doctor represents an arm. Each arm has multiple parameters corresponding to each distribution of rating, experience, and distance (distance between doctor and patient).

Rating is initialized with a Normal distribution (mean = 1, variance = 1), experience with an Exponential distribution (rate = 1, mean = 1), and distance with a Normal distribution (mean = 0, variance = 1). Thus, each arm now has six parameters, two for each distribution.

*2) Reward Calculation and Selection of Arm:* We *sample* values from three distributions corresponding to rating, experience, and distance for every arm. To calculate the reward for selecting an arm, we consider the following:

1) Distance is inversely proportional to reward, while experience and rating are directly proportional.
2) We normalize these three values to convert them to the range [0, 1], since all three distributions have different ranges of values.
3) Finally, the score is calculated as a weighted linear combination of these three values, with weights assigned according to user requirements.

After calculating the reward for every arm, we select the arm with the maximum possible reward.

*3) Updating Arm Parameters:* We will now update the parameters of the selected arm. We retrieve the actual values of the selected arm from the doctors dataset.

Next, we consider the conjugate priors and posteriors for the distributions and apply the corresponding update rules for distribution parameters as follows in table 5:

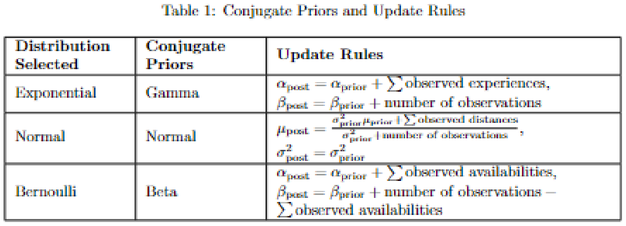


Table 1: Conjugate Priors and Update Rules

| Distribution Selected | Conjugate Priors | Update Rules |
|---|---|---|
| Exponential | Gamma | $\alpha_{\text{post}} = \alpha_{\text{prior}} + \sum \text{observed experiences},$<br>$\beta_{\text{post}} = \beta_{\text{prior}} + \text{number of observations}$ |
| Normal | Normal | $\mu_{\text{post}} = \frac{\sigma^2_{\text{prior}}\mu_{\text{prior}} + \sum \text{observed distances}}{\sigma^2_{\text{prior}} + \text{number of observations}},$<br>$\sigma^2_{\text{post}} = \sigma^2_{\text{prior}}$ |
| Bernoulli | Beta | $\alpha_{\text{post}} = \alpha_{\text{prior}} + \sum \text{observed availabilities},$<br>$\beta_{\text{post}} = \beta_{\text{prior}} + \text{number of observations} - \sum \text{observed availabilities}$ |

Fig. 5: Update Rules for Distribution parameters

*4) Algorithm analysis:*

- Probabilistic approach: Samples from the posterior distribution of parameters.
- Suitable for exploring complex reward distributions.
- Requires less training and provides faster real-time inference.

## D. Contextual Bandits

The contexts of users and doctors are utilized to enhance the predicted reward. Rather than maintaining separate multi-armed bandits for each user, we can employ a single generalized bandit through the use of contextual bandits.

In this section, we will examine a specific variant of contextual bandits: *LinearUCB*. A flowchart of contextual bandits can be referred to in Figure 6.

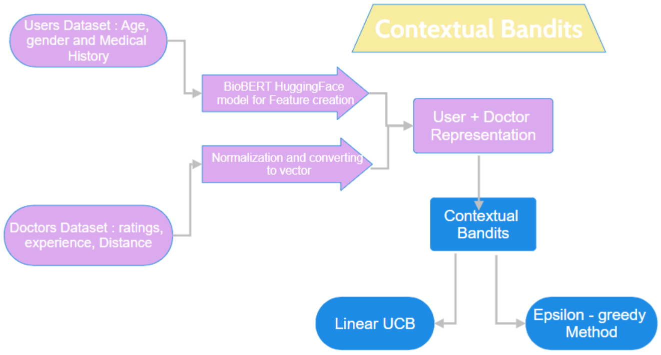

Contextual Bandits
Users Dataset : Age, gender and Medical History
BioBERT HuggingFace model for Feature creation
Normalization and converting to vector
Doctors Dataset : ratings, experience, Distance
User + Doctor Representation
Contextual Bandits
Linear UCB
Epsilon - greedy Method

Fig. 6: Contextual Bandits Flowchart

*1) Context Creation:* The users' dataset includes age, gender, and medical history, represented as a list of disease-symptom pairs. To obtain a numerical representation of a user, we transform the medical history into numerical vectors using embeddings generated from the *BioBERT* transformers model. Age and gender are then directly incorporated as numerical values.

For the doctors' representation, we directly convert each attribute value into a numerical vector, as all attributes are already numerical. The final context vector is created by concatenating the user and doctor representations.

*2) Calculating LinUCB scores for arms:* at time t, we have context representation matrix (X) and rewards we got (r) for previours t-1 arm selections, we find weights w using linear regression as

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{r}$$

Now UCBScore(t, a) denotes the UCB score of selecting arm a at time t, then

$$\text{UCBScore(t, a)} = \mathbf{x}_{t,a}^T\mathbf{w} + \alpha\sqrt{\mathbf{x}_{t,a}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_{t,a}}$$

Here, $\mathbf{x}_{t,a}$ represents the context representation of the user at time $t$ and arm $a$. The parameter $\alpha$ provides control over the degree of exploration.

*3) Selection of arm and updation of w and X:* We will select an arm $a_t$ with highest UCBScore(t, a) at time t as follows -

$$a_t = \arg\max_{a\in\mathcal{A}}(\text{UCBScore(t, a)})$$

Then update Context representation matrix by appending $\mathbf{x}_{t,a}$ to it and append reward we get for selecting $a_t$ to r.

*4) Algorithm analysis:*

- Deterministic approach: Uses linear upper confidence bounds.
- Suitable when the reward function is well understood.
- Requires more training and results in slower inference.

## V. Evaluation Metrics

To effectively measure the performance of the recommendation algorithms employed within our system, we utilize several key metrics. These metrics help us to understand the effectiveness, breadth, and user satisfaction provided by our recommendations. Below, we describe each of these metrics in detail.

### A. Regret

Regret is a fundamental metric in the context of recommendation systems, particularly when using reinforcement learning approaches such as Multi-Armed Bandits. It measures the difference between the reward obtained by the chosen action and the reward that would have been obtained by the best possible action. Lower regret values indicate that the system is making near-optimal choices more frequently.

$$\text{Cumulative Regret} = \sum_{t=1}^{T}(\text{Max Reward}_t - \text{Observed Reward}_t)$$

Where $T$ is the number of trials, Max Reward$_t$ is the highest possible reward at trial $t$, and Observed Reward$_t$ is the reward received by following the chosen action. Here, considering the maximum reward is 1 then the results for the regret are:

- **Multi-Armed Bandits**: For the Multi-Armed Bandits if the number of trials taken into consideration is 5000 then the regret is ranging between 600-1000.
- **UCB**: For UCB, if the number of trials is 5000 the regret is ranging between 700-1200, highlighting that UCB prefers to choose the new arm which are unexplored under the concept of optimism under uncertainity.

### B. Coverage

Coverage measures the proportion of the possible recommendation space that is actually utilized by the recommendation algorithm. High coverage indicates a diverse set of recommendations, which is particularly important in healthcare to cater to a wide variety of patient needs.

$$\text{Coverage} = \frac{\text{Number of Unique Recommendations}}{\text{Total Number of Possible Recommendations}}$$

Comparing the coverage between UCB and MAB, the coverage is better in case of UCB as the algorithm of UCB always prefers the arm or the doctor which hasn't been explored before and hence the coverage for the UCB is greater is than the MAB.

### C. Hit Ratio

The Hit Ratio is used to assess the accuracy of the recommendations provided. It calculates the proportion of times the recommended item was indeed what the user was looking for. In the context of doctor recommendations, a "hit" might be defined as a case where the recommended doctor meets the user's specific requirements or conditions.

$$\text{Hit Ratio} = \frac{\text{Number of Hits}}{\text{Total Number of Recommendations}}$$

### D. K-Reciprocal Hit Rate

K-Reciprocal Hit Rate extends the concept of the hit ratio by considering the position of the correct recommendation within the top-K list. It emphasizes not only the correctness of the recommendation but also its ranking.

$$\text{K-Reciprocal Hit Rate} = \frac{1}{\text{Rank of Top-K correct recommendation}}$$

This metric is particularly useful when the order of recommendations matters, such as prioritizing more suitable doctors higher in the list.

## VI. Future Works

While this paper offers valuable insights into processing medical queries, we can further enhance by integrating real data for fair-use purposes by government agencies. This will help us to assess and give region-wise accurate results like recommending specialists in the city to which the patient belongs. We can further improve the results by using location of patient while mapping it to the most relevant and nearest doctor. In this way, we can create a all-in-one application which will tailor responses for every patient around the world. Another notable work can be comparing all the results from bandits as given above, and display the most relevant specialists, thus giving tailored result. Notably, novel methods can be experimented which may have more accurate result or be computationally efficient than our proposed methods.

## References

[1] Hatice Özbolat ”Text Summarization: How to Calculate BertScore” in Medium released on September 28, 2023.

[2] Clément Christophe et. al. , ”Evaluating Fine-Tuning Strategies for Medical LLMs: Full-Parameter vs. Parameter-Efficient Approaches,” *AAAI 2024 Spring Symposium - Clinical Foundation Models*.