Medicine Recommendation System

Raghunadh
IMT2021042

Kushal Dasari
IMT2021035

Prachoday
IMT2021034

***Abstract*—In the current medical scenario new discoveries are made everyday and new medicines, conditions and symptoms are added to the list of things doctors and medical professionals are expected to know about. Also patients might want to get a second opinion of things sometimes. In some cases it helps to have an instant medicine recommender that could potentially help save lives.**

Through this paper we wish to tackle the above problems with a personalized medcine recommendation system.

Keywords

BioBert, Multi Arm Bandits, NER, UCB.

I. Introduction

Our project first extracts symptoms from a patient description which then we use to find the most similar Use from the uses column by comparing the vectors, after finding the use we recommend medicine's using Multi Arm Bandits (UCB approach).

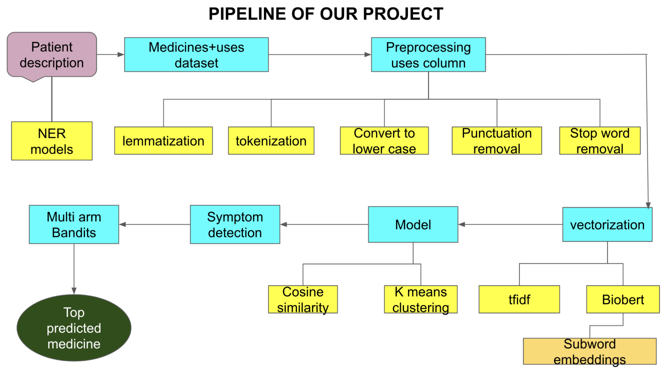


Fig. 1. Pipeline

Essential parts of our pipeline include -

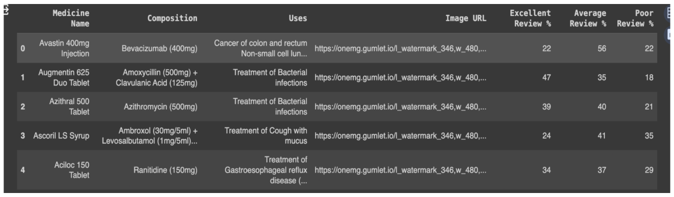
- Extracting symptoms from patient description: We use NER models to extract the relevant symptoms from a text description of the patients symptoms.
- Vectors of the uses column: We produce the vectors for the uses column using biobert which we use to compare with the extracted symptoms that we got using NER models, by doing this we find the most similar use for the symptom.
- Recommending medicines for the symptom: There are multiple medicines for a given symptom in the dataset. So, We use MAB(UCB approach) to find the best medicine and recommend it. Our novelty lies in the formula used in UCB.

II. Dataset's used and preprocessing

A. Dataset's

- Medicine_Details.csv.

This dataset contains the uses column, which is the uses related to the medicine that a particular company is manufacturing. This column will later be vectorized and used to find the most similar use for a symptom.

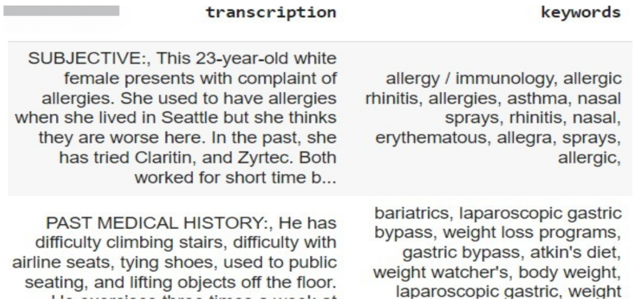


	Medicine Name	Composition	Uses	Image URL	Excellent Review %	Average Review %	Poor Review %
0	Avastin 400mg Injection	Bevacizumab (400mg)	Cancer of colon and rectum Non-small cell lun...	https://onemg.gumlet.io/l_watermark_346,w_480,...	22	56	22
1	Augmentin 625 Duo Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections	https://onemg.gumlet.io/l_watermark_346,w_480,...	47	35	18
2	Azithral 500 Tablet	Azithromycin (500mg)	Treatment of Bacterial infections	https://onemg.gumlet.io/l_watermark_346,w_480,...	39	40	21
3	Ascoril LS Syrup	Ambroxol (30mg/5ml) + Levosalbutamol (1mg/5ml)...	Treatment of Cough with mucus	https://onemg.gumlet.io/l_watermark_346,w_480,...	24	41	35
4	Aciloc 150 Tablet	Ranitidine (150mg)	Treatment of Gastroesophageal reflux disease (...	https://onemg.gumlet.io/l_watermark_346,w_480,...	34	37	29

Fig. 2. Medicine_details.csv

- mtsamples.csv

This dataset has the patients description as a paragraph and the important medical syptoms that are present in this description. This will be used to test our NER model, which extracts symptoms from the patient description.



transcription	keywords
SUBJECTIVE:, This 23-year-old white female presents with complaint of allergies. She used to have allergies when she lived in Seattle but she thinks they are worse here. In the past, she has tried Claritin, and Zyrtec. Both worked for short time b...	allergy / immunology, allergic rhinitis, allergies, asthma, nasal sprays, rhinitis, nasal, erythematous, allegra, sprays, allergic,
PAST MEDICAL HISTORY:, He has difficulty climbing stairs, difficulty with airline seats, tying shoes, used to public seating, and lifting objects off the floor.	bariatrics, laparoscopic gastric bypass, weight loss programs, gastric bypass, atkin's diet, weight watcher's, body weight, laparoscopic gastric, weight

Fig. 3. mtsamples.csv

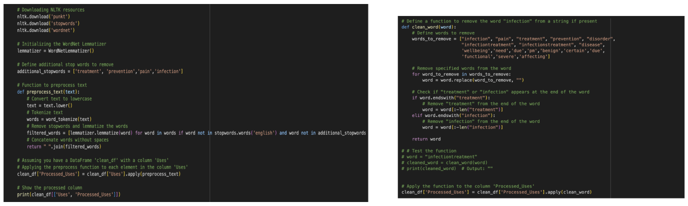
B. Pre-processing

We found "84" duplicate rows in the data set, Since 84 is very small comparative to total rows(11800) ,we dropped the rows.

We identified the unique uses in the uses column then we converted them to lower case and and also removed

Identify applicable funding agency here. If none, delete this.

punctuation after that we identified the stop words in each uses, We removed them and also lemmatized them.
Then we formed a new column called Processed_uses which was added to the Dataframe.



```
# Downloading NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Initializing the WordNet Lemmatizer
lemmatizer = WordNetLemmatizer()

# Define additional stop words to remove
additional_stopwords = ['treatment', 'prevention','pain','infection']

# Function to preprocess text
def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()
    # Tokenize text
    words = word_tokenize(text)
    # Remove stopwords and lemmatize the words
    filtered_words = [lemmatizer.lemmatize(word) for word in words if word not in stopwords.words('english') and word not in additional_stopwords
    # Concatenate words without spaces
    return " ".join(filtered_words)

# Assuming you have a DataFrame 'clean_df' with a column 'Uses'
# Applying the preprocess function to each element in the column 'Uses'
clean_df['Processed_Uses'] = clean_df['Uses'].apply(preprocess_text)

# Show the processed column
print(clean_df[['Uses', 'Processed_Uses']])
```

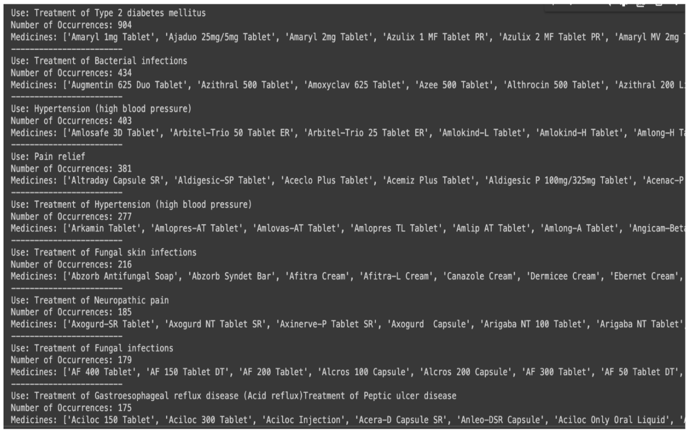
```
# Define a function to remove the word "infection" from a string if present
def clean_word(word):
    # Define words to remove
    words_to_remove = ["infection", "pain", "treatment", "prevention", "disorder",
                       "infectiontreatment", "infectionstreatment", "disease",
                       'wellbeing','need','due','pm','benign','certain','due',
                       'functional','severe','affecting']

    # Remove specified words from the word
    for word_to_remove in words_to_remove:
        word = word.replace(word_to_remove, "")

    # Check if "treatment" or "infection" appears at the end of the word
    if word.endswith("treatment"):
        # Remove "treatment" from the end of the word
        word = word[:-len("treatment")]
    elif word.endswith("infection"):
        # Remove "infection" from the end of the word
        word = word[:-len("infection")]

    return word

# # Test the function
# word = "infectiontreatment"
# cleaned_word = clean_word(word)
# print(cleaned_word)  # Output: ""


# Apply the function to the column 'Processed_Uses'
clean_df['Processed_Uses'] = clean_df['Processed_Uses'].apply(clean_word)
```

Fig. 4. Processed_uses

C. Exploratory Data Analysis

We are first finding out the unique uses and then printing the top 10 occurring uses and the medicine associated with it.

Through this we can see that there are multiple medicines associated with a single use.



```
Use: Treatment of Type 2 diabetes mellitus
Number of Occurrences: 904
Medicines: ['Amaryl 1mg Tablet', 'Ajaduo 25mg/5mg Tablet', 'Amaryl 2mg Tablet', 'Azulix 1 MF Tablet PR', 'Azulix 2 MF Tablet PR', 'Amaryl MV 2mg 
------------------------
Use: Treatment of Bacterial infections
Number of Occurrences: 434
Medicines: ['Augmentin 625 Duo Tablet', 'Azithral 500 Tablet', 'Amoxyclav 625 Tablet', 'Azee 500 Tablet', 'Althrocin 500 Tablet', 'Azithral 200 L
------------------------
Use: Hypertension (high blood pressure)
Number of Occurrences: 403
Medicines: ['Amlosafe 3D Tablet', 'Arbitel-Trio 50 Tablet ER', 'Arbitel-Trio 25 Tablet ER', 'Amlokind-L Tablet', 'Amlokind-H Tablet', 'Amlong-H Ta
------------------------
Use: Pain relief
Number of Occurrences: 381
Medicines: ['Altraday Capsule SR', 'Aldigesic-SP Tablet', 'Aceclo Plus Tablet', 'Acemiz Plus Tablet', 'Aldigesic P 100mg/325mg Tablet', 'Acenac-P
------------------------
Use: Treatment of Hypertension (high blood pressure)
Number of Occurrences: 277
Medicines: ['Arkamin Tablet', 'Amlopres-AT Tablet', 'Amlovas-AT Tablet', 'Amlopres TL Tablet', 'Amlip AT Tablet', 'Amlong-A Tablet', 'Angicam-Beta
------------------------
Use: Treatment of Fungal skin infections
Number of Occurrences: 216
Medicines: ['Abzorb Antifungal Soap', 'Abzorb Syndet Bar', 'Afitra Cream', 'Afitra-L Cream', 'Canazole Cream', 'Dermicee Cream', 'Ebernet Cream',
------------------------
Use: Treatment of Neuropathic pain
Number of Occurrences: 185
Medicines: ['Axogurd-SR Tablet', 'Axogurd NT Tablet SR', 'Axinerve-P Tablet SR', 'Axogurd  Capsule', 'Arigaba NT 100 Tablet', 'Arigaba NT Tablet',
------------------------
Use: Treatment of Fungal infections
Number of Occurrences: 179
Medicines: ['AF 400 Tablet', 'AF 150 Tablet DT', 'AF 200 Tablet', 'Alcros 100 Capsule', 'Alcros 200 Capsule', 'AF 300 Tablet', 'AF 50 Tablet DT',
------------------------
Use: Treatment of Gastroesophageal reflux disease (Acid reflux)Treatment of Peptic ulcer disease
Number of Occurrences: 175
Medicines: ['Aciloc 150 Tablet', 'Aciloc 300 Tablet', 'Aciloc Injection', 'Acera-D Capsule SR', 'Anleo-DSR Capsule', 'Aciloc Only Oral Liquid', '
```

Fig. 5. top10uses

III. Extraction of symptoms

A. Model used

In our study, we utilized the `en_ner_bc5cdr_md` model from the spaCy library to efficiently extract medical symptoms from textual data. This pre-trained model, specifically designed for biomedical corpora, enabled high-precision identification of symptom-related entities within paragraphs. The choice of `en_ner_bc5cdr_md` significantly enhanced our text mining capabilities, contributing to the depth and reliability of our findings.

B. Testing of our model

We conducted our experiments using the `mtsamples.csv` dataset, which contains both descriptions and keywords, the latter being specific medical terms. We applied our NER model to all transcription entries in this dataset to evaluate its efficacy. Specifically, we checked for the presence of each keyword listed in the dataset within the predictions made by our model.

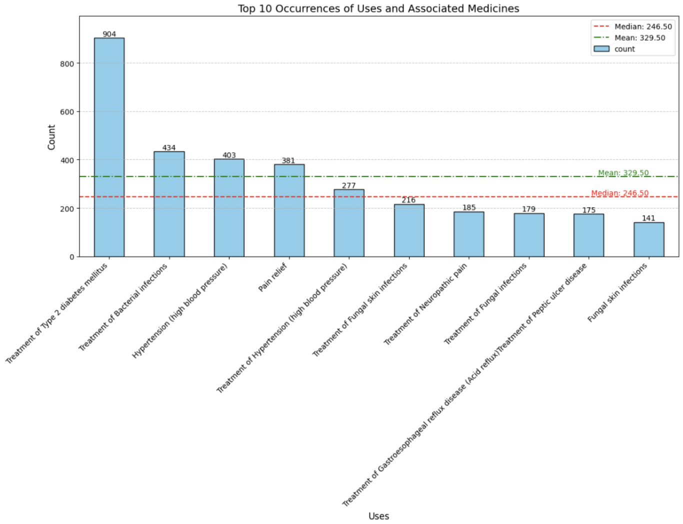
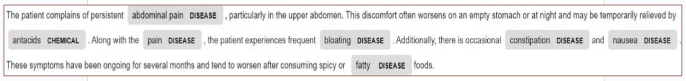


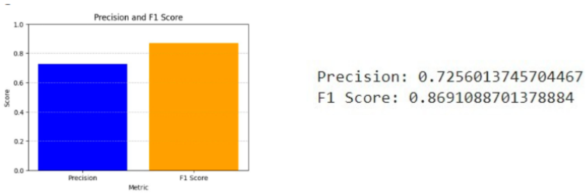
Fig. 6. bar_graph



The patient complains of persistent abdominal pain DISEASE , particularly in the upper abdomen. This discomfort often worsens on an empty stomach or at night and may be temporarily relieved by antacids CHEMICAL . Along with the pain DISEASE , the patient experiences frequent bloating DISEASE . Additionally, there is occasional constipation DISEASE and nausea DISEASE .
These symptoms have been ongoing for several months and tend to worsen after consuming spicy or fatty DISEASE foods.

Fig. 7. Named Entity Recognition

The evaluation was quantified by calculating the percentage of keywords accurately identified by our model and deriving the F1 scores to measure the precision, recall, and overall accuracy of the model. We observed we got pretty good f1 score upon evaluating our model.



```
Precision: 0.7256013745704467
F1 Score: 0.8691088701378884
```

Fig. 8. F1 score

IV. Vectorization

We have to do vectorization to ’uses’ column of medicines dataset to recommend medicines based on user’s symptoms. We cannot deal with text in machine learning, so we decided to vectorize it using the below algorithms.

A. Tfidf

We employed the TF-IDF (Term Frequency-Inverse Document Frequency) technique for vectorization. Initially, we used this basic model to do vectorize uses of medicines.

B. Bio bert and subword embedding

We utilized the BioBERT model to calculate vectors for the ”Processed uses” column of medicine dataset. BioBERT is trained on extensive biomedical literature and datasets,

leveraging a transformer-based architecture with self-attention mechanisms. We also did subword embedding to deal with out of vocabulary words. As we are working on medical domain, we thought that this is the best model that can be used. We can also do fine tuning on large dataset for better vector representation
This picture shows how Bio Bert works.

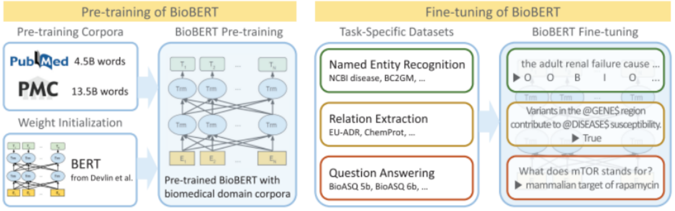
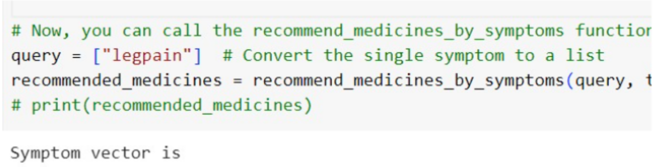

Pre-training of BioBERT
Pre-training Corpora
PubMed 4.5B words
PMC 13.5B words
Weight Initialization
BERT from Devlin et al.
BioBERT Pre-training
Pre-trained BioBERT with biomedical domain corpora
Fine-tuning of BioBERT
Task-Specific Datasets
Named Entity Recognition NCBI disease, BC2GM, ...
Relation Extraction EU-ADR, ChemProt, ...
Question Answering BioASQ 5b, BioASQ 6b, ...
BioBERT Fine-tuning
the adult renal failure cause ... ▶ O O B I O ...
Variants in the @GENE$ region contribute to @DISEASE$ susceptibility. ▶ True
What does mTOR stands for? ▶ mammalian target of rapamycin

Fig. 9. Working the BioBert

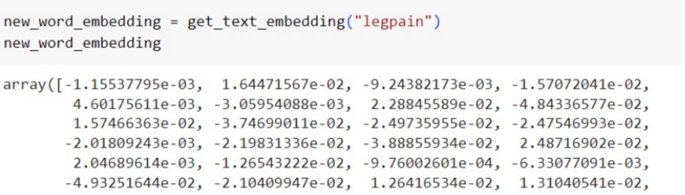
C. TF-IDF Vs BioBert

TF-IDF, while effective for representing word importance within a dataset, faces challenges with out-of-vocabulary words, limiting its applicability in diverse text corpora. In contrast, BioBert's utilization of subword embeddings allows it to capture the semantic meaning of both in-vocabulary and out-of-vocabulary terms, offering a more comprehensive representation of textual data across various domains and enhancing downstream tasks such as information retrieval and text classification.
Below is the comparison between TF-IDF and Bio-Bert



```
# Now, you can call the recommend_medicines_by_symptoms functior
query = ["legpain"]  # Convert the single symptom to a list
recommended_medicines = recommend_medicines_by_symptoms(query, t
# print(recommended_medicines)

Symptom vector is
```

Fig. 10. vector for leg pain in tf-idf(null vector)



```
new_word_embedding = get_text_embedding("legpain")
new_word_embedding

array([-1.15537795e-03,  1.64471567e-02, -9.24382173e-03, -1.57072041e-02,
        4.60175611e-03, -3.05954088e-03,  2.28845589e-02, -4.84336577e-02,
        1.57466363e-02, -3.74699011e-02, -2.49735955e-02, -2.47546993e-02,
       -2.01809243e-03, -2.19831336e-02, -3.88855934e-02,  2.48716902e-02,
        2.04689614e-03, -1.26543222e-02, -9.76002601e-04, -6.33077091e-03,
       -4.93251644e-02, -2.10409947e-02,  1.26416534e-02,  1.31040541e-02,
```

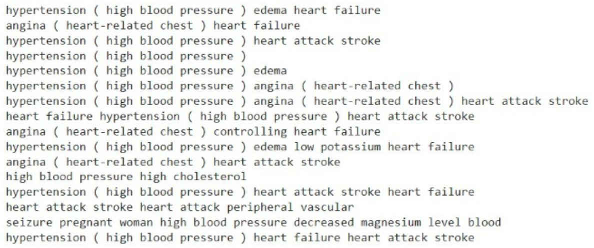
Fig. 11. vector for leg pain in bio-bert(some vector)

V. Similarity and Clustering

We have now obtained better vectors for uses of medicines. Uses column in our dataset after doing pre-processing gives us the symptoms which the medicine cures. We convert all these symptoms in our dataset into vectors using the above biobert method. When we see a new symptom, we first convert it into vector using the same model. We then compare it with the vectors of existing symptoms. The medicine corresponding to the most similar symptom will be recommended.

A. K Means

We applied K Means to all the uses of medicines. We obtained clusters of similar uses. We kept cosine similarity as distance between uses when we applied this algorithm as it is more suitable for this task than eucledian distance. We used elbow plot to optimally find the number of clusters. From the graph and manual observation, we inferred that we had to make 27 clusters.
This is the sample cluster that we have obtained.



```
hypertension ( high blood pressure ) edema heart failure
angina ( heart-related chest ) heart failure
hypertension ( high blood pressure ) heart attack stroke
hypertension ( high blood pressure )
hypertension ( high blood pressure ) edema
hypertension ( high blood pressure ) angina ( heart-related chest )
hypertension ( high blood pressure ) angina ( heart-related chest ) heart attack stroke
heart failure hypertension ( high blood pressure ) heart attack stroke
angina ( heart-related chest ) controlling heart failure
hypertension ( high blood pressure ) edema low potassium heart failure
angina ( heart-related chest ) heart attack stroke
high blood pressure high cholesterol
hypertension ( high blood pressure ) heart attack stroke heart failure
heart attack stroke heart attack peripheral vascular
seizure pregnant woman high blood pressure decreased magnesium level blood
hypertension ( high blood pressure ) heart failure heart attack stroke
```

Fig. 12. sample cluster ralated to heart

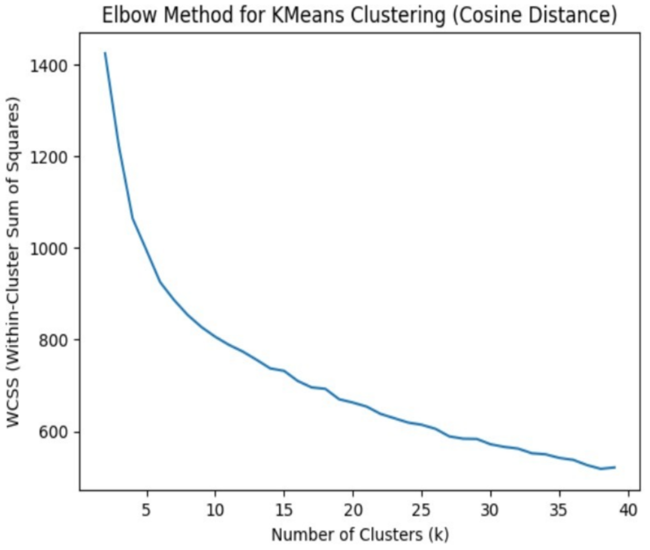
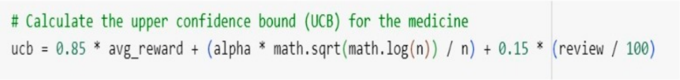


Fig. 13. Pipeline

VI. Multi Arm Bandits

We utilized data analysis to identify multiple medications targeting the same symptom. To streamline recommendations, we implemented a multi-armed bandit approach, specifically the upper confidence bound (UCB) algorithm. Feedback from users, denoted as either ”0” (dislike) or ”1” (like), served as the reward signal. Upon receiving negative feedback (0), the algorithm ceased recommending the respective medication to that user, optimizing subsequent suggestions.

To enhance initial recommendations, we integrated medicine reviews into the UCB algorithm. This modification prioritizes medications with positive reviews over random selection, refining the recommendation process from the outset.And this average reward, denoted as avg_reward, is computed based on the rewards provided by users up to the current point in time, which are stored in the dataset's 'Rewards' column.Basically in the MAB view each arm having a rewards list.



```
# Calculate the upper confidence bound (UCB) for the medicine
ucb = 0.85 * avg_reward + (alpha * math.sqrt(math.log(n)) / n) + 0.15 * (review / 100)
```

Fig. 14. UCB

VII. Working of UCB(Algorithm)

Initialize Variables: Initialize variables best_medicine and max_ucb to keep track of the medicine with the highest Upper Confidence Bound (UCB) and its value, respectively.

Iterate Through Medicines:

For each medicine in the dataset:

1.Calculate the average reward for the medicine based on the rewards received so far.

2.Retrieve the review percentage for the medicine from external data.

3.Calculate the number of times the medicine has been recommended (n).

4.Compute the Upper Confidence Bound (UCB) for the medicine using the UCB formula incorporating exploration and exploitation terms.

5.Update best_medicine if the calculated UCB is higher than the current maximum.

Return Recommended Medicine:

Return the medicine with the highest UCB as the recommended choice.

After this, we ask for feedback : 0 or 1.

Update Reward Function:

We defined a function update_reward to update the reward array of a medicine based on received feedback.

VIII. Further Works

Expanding our project from single-user functionality to accommodate multiple users entails transitioning from upper confidence bound (UCB) bandits to contextual bandits. Additionally, to enhance our model's performance, we aim to leverage GPUs for pre-training on larger datasets, facilitating the creation of more refined vector representations. We can also include side-effects of medicines when we predict medicines. We can also try to incldue medical history of the patient and the diseases frm which he has suffered till now.

IX. Conclusion

In conclusion, our Medicine Recommendation System addresses the pressing need for accurate and efficient medication suggestions in the medical field. By leveraging advanced techniques such as Named Entity Recognition (NER), vectorization using BioBERT, and multi-armed bandit algorithms like the Upper Confidence Bound (UCB) approach, we have developed a system capable of analyzing patient symptoms, identifying relevant medications, and recommending the most suitable treatment options.

Through extensive preprocessing, vectorization, and clustering of medical data, our system ensures that recommendations are tailored to individual patient needs, taking into account the similarity between symptoms and the effectiveness of different medications. Additionally, by incorporating user feedback and medication reviews, we continuously refine our recommendations, providing users with personalized and reliable suggestions.

While our current system demonstrates promising results, there is ample room for further enhancement and expansion. Future work could involve incorporating contextual bandit algorithms to support multiple users and refining vector representations through pre-training on larger datasets. Furthermore, the inclusion of medication side-effects and additional medical data sources could further enrich our recommendation system, ultimately improving patient outcomes and healthcare delivery.

In essence, our Medicine Recommendation System represents a significant step forward in leveraging artificial intelligence and data-driven approaches to enhance medical decision-making, ultimately benefiting both patients and healthcare providers alike.

References

[1] https://github.com/kushaldasari/medicine_recsys.

[2] https://allenai.github.io/scispacy/

[3] https://www.kaggle.com/code/nihilus888/medicine-recommendation-system/input

[4] https://www.kaggle.com/code/debbiechu/medical-transcriptions-nlp/input

[5] https://huggingface.co/dmis-lab/biobert-large-cased-v1.1-squad