Early Results on Treatment Recommendation System

Shuai Xu

Abstract—

Medication error happened frequently which may lead to a serious result. 42% of the medication error is cause by the lack of medicine knowledge of new drugs and lack of experience about some diseases. In recent years, medical recommendation plays an important role in the related fields. In this project, I will build a Treatment Recommendation System (TRS) based on NLP techniques. The dataset I used in this project is a web-scraped dataset from PatientsLikeMe.com. The first step is data preprocessing and Exploratory Data Analysis for some visualization. For the recommendation system main part, I use the lexical database (NLTK WordNet) and corpus statistics in NLP to measure the semantic similarity between words and words, sentences and sentences. Then, implement the Collaborative Filtering on the Recommendation System to find similar patients to recommend treatments. During the recommendation process, using blurry search technique to use word association to find symptom synonyms to auxiliary improve the recommendation results and cover rate. Finally, use the precision, recall, and f1-score to evaluate the performance.

***Key Words* — Treatment Recommendation System, Collaborating Filtering, Natural Language Processing, Semantic Analysis, Semantic Similarity, lexical database, Word Association**

I. Introduction

Recommender systems are widely used in ecommerce, social media, and web services [1]. However, in the recent years, Recommender System plays an important role in the medical area. Medication error happened frequently which may lead to a serious result. 42% of the medication error is cause by the lack of medicine knowledge of new drugs and lack of experience about some diseases [2]. Besides, the fast developed expanding database of new drugs will lead to the problem: doctors can’t learn all those new drugs in the short time and can’t remember all the effects accurately.

Thus, the recommendation system in medical area comes to our eyes. Treatment Recommendation System (TRS) may help doctor precisely prescribe drugs or provide treatments for the patients. Treatment Recommendation System, on the professionals’ side, could help them find the similar patients with similar symptoms and specific condition they have or good treatments they received, and to help doctor to come up ideas of treatments for new patients and identify drug prescription more accurately [3]. It’s a good way to improve the healthcare decision making. Also, on the patients’ side, Recommendation System could analyze the user health status and monitor their lifestyle to dynamically provide some suggestions for the user in time, motivated them, like do more exercise, stop smoking, etc [4]. Personalization will be another important aspect of medical recommendation system, which could provide some effective and reasonable advice for the right doctor with good patients’ feedback [5] or the right hospital to save money save time. Furthermore, helping patients find healthcare personalized precisely trustable therapy [6].

For the previous works, it covered some different NLP text mining similarity measures for sentence, collaborative filtering, knowledge-based (content-based filtering), and hybrid filtering recommendation based on trusting doctor-patient relationship [8], other ML algorithms, like decision tree, clustering, etc, or CF based on DL frameworks [3].

In this paper, The TRS will include two stages: first part will predict the k top possible conditions, then based on the conditions, provide the treatments. TRS finds relative symptom, and conditions (Word Association by synonyms) to help recommend, make the system more flexible. Also, I will deal with multiple text treatments outputs combine with different text similarity measures, not one single target (one treatment, hospital department, or numeric doctor id, etc.). Several similarity measures will be weighted and combine together as the final similarity measure.

Expected contributions of this work will be successfully build the two-stage TRS based on Collaborative Filtering and Content-based Filtering. This two-stage TRS could support to input the blurry symptoms (Word Association), and then find the possible conditions. With the patient or doctor's help to choose the right condition, then recommend treatments based on the condition.

II. BACKGROUND/RELATED WORK

The recommendation system in healthcare area is like the information query and filtering system in other area. It performs an important role in future medical area. Three are three most common recommendation algorithms, which are Collaborating Filtering, Content-based Filtering, and Hybrid Filtering which combine the pros of two other methods. Cosine similarity is a very good method to measure the similarity of word vectors, and this method was used in most of the CF, CB recommendation papers. Only focus on one measurement may lead to lose important potential information in other aspects.

Besides, most of the paper is going to predict the single label, such as the doctor, hospital department. Also not support the blurry search to improve the recommendation result.

Thus, I decide to cascade multiple similarity measures together to measure the overall similarity to see the difference. Then, from the blurry input symptom, I will use the symptom association, age, and gender information to predict the possible conditions from the exist dataset. Then, based on the top possible condition, we will recommend treatments. The blurry search will improve the performance of TRS.

Word association is introduced after user input symptom information. This could allow the blurry search, like if we type some symptoms that are not exist in the dataset, it will not give us the good recommendation result, but if we use the word association to find similar word in the dataset that will improve the probability to get a good recommendation and improve the accuracy of diagnosis.

1) Symptom association during user input
Based on semantic similarity, symptoms occurrences, or find synonyms based on NLTK WordNet to get similar words for the user choice. Then, we could provide top k synonyms with high probability to the user. Patients or doctors can choose the symptoms associated with their own feeling or determinate the possible conditions.

2) Association of conditions
Calculate the weight of each symptom in each condition based on the frequency, and then select the top k conditions as associative conditions. Provide those associative conditions to users as the evidence of the final treatment recommendation.

III. METHODOLOGY

3.1. Data Source

The dataset from the website PatientLikesMe.com, which is web-scraped by GitHub user [9]. The original dataset has 3350 patient information observations with 8 column features, includes, UserID, age, gender, city, state, condition, symptom, and treatment. Name, City, State, and Condition are single column text feature, Gender is categorical feature, Age is numeric feature, Symptom is multi-columns text feature, and final recommendation output is Treatment, which is multi-columns text feature. For the basic idea, in this project, I will only use five of the columns: Age, Gender, Condition, Symptom, and Treatment.

Table 1: dataset info

Feature	**Feature type**	**Description**
UserID	Text	Patient's ID
Age	Numeric	Patient's age
Gender	Categorical	Patient's gender
City	Text	Patient's location
State	Text	Patient's location
Condition	Text	Final condition
Symptom	Text	Patient's symptom
Treatment	Text	Final Treatments

3.2. Data Preprocessing

3.2.1 Deal with missing values

I firstly check the missing value in the dataset. I found there are some missing values in the column State and Treatment. Because the treatment is the ground truth column, I droped those rows with Treatment=NaN; for those rows with State=NaN, I keep them, because I decide not to use the State column in this project. After drop the empty rows, there are 3328 datapoints in the dataset.

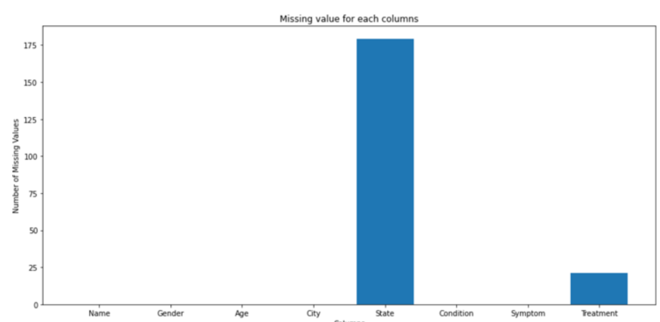

Figure 1: missing values

3.2.2 Data Cleaning

When I check the dataset, I found there are some unexpected marks in the symptom and treatment columns, such as 'redness)' and '"Itching'. I need to remove those unexpected marks) and ". Also, I find there are some treatments filled by symptoms, like 'Pain",Fatigue' which are obviously not the treatment but the symptoms. After I removed rows with those keywords and fix the unexpected marks occur in the features, there are total 3286 datapoints left in the dataset.

3.2.3 Mapping

Because the symptom and treatment column has multiple words split by comma, I created a mapping to find the inner relation between those features. I set a one-to-N mapping of N different treatmens for one condition and also a one-to-N mapping of N different symptoms for one condition. Then, I created a N-to-one mapping for one symptom may occur in N conditions and also a N-to-on mapping for one treatment may use for N conditions. After obtaining the mapping relation, we could dig more about the dataset.

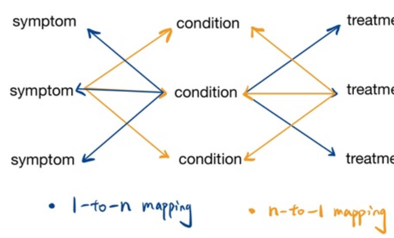

Figure 2: mapping

IV. RESULTS

4.1 Exploratory Data Analysis

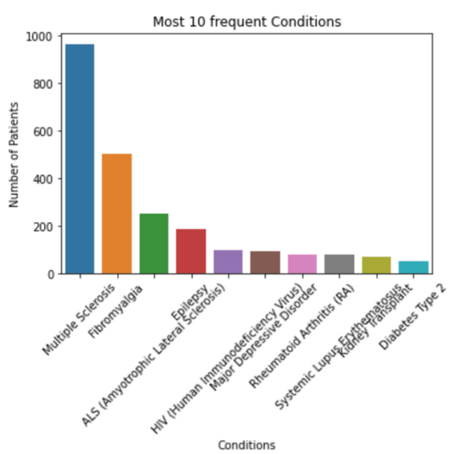

Figure 3: Top 10 frequent conditions

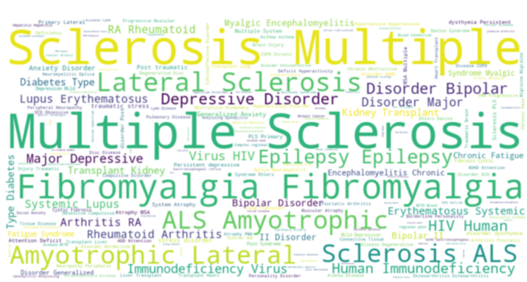

Figure 4: Word Cloud of condition frequency

From the above plot 3, we could see the 10 most common condition people usually suffered for.

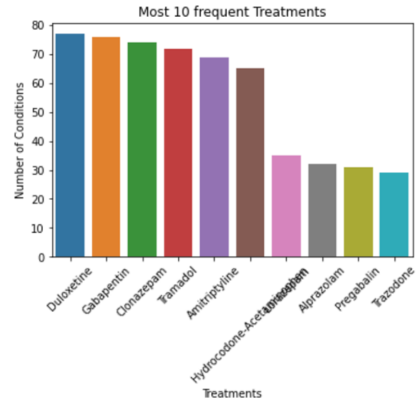

Figure 5: Top 10 frequent Treatments

Figure 6: Word Cloud of treatment frequency

From the above figure 5, we could find the 10 most common use treatments in all the conditions.

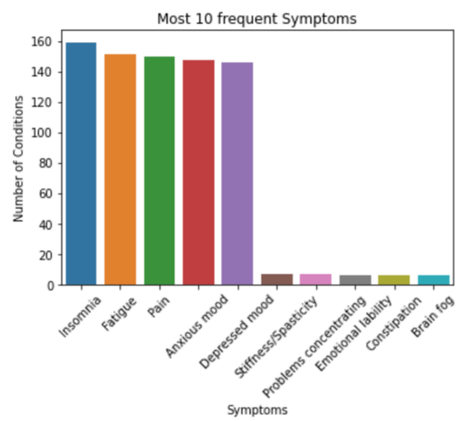

Figure 7: Top 10 frequent symptoms

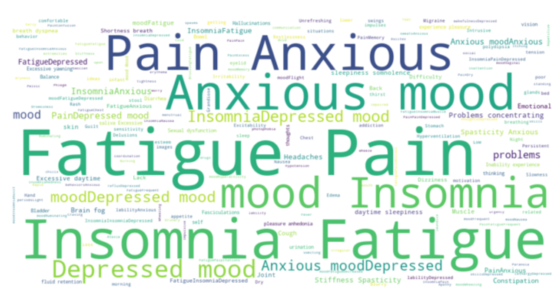

Figure 8: Word Cloud of symptom frequency

From the above figure 7, we could find there are five very common symptoms, Insomnia, Fatigue, Pain, Anxious mood, and Depressed mood.

V. FUTURE WORK

1. Train Test set split. I will use the stratify train test split to obtain the train and test set with ratio of 4 to 1. However, some of the conditions only have 1 datapoint, can't use the stratify resampling for this situation. We could not simply consider them as the outlier and discard. The validation set will be the 1/5 of the training set.

2. Treatment Recommendation System Framework: there are three basic recommendation algorithms: Collaborating Filtering (CF), Content-based Filtering (CB), and Hybrid Filtering [10]. I will implement Collaborating Filtering in this project. Because there is no Rating feature in this dataset, we could not simply treat this Treatment Recommender System like a normal RS, such as Movie RS. We can't use Matrix Factorization or Rating prediction to help recommend which based on user ratings. I will focus on use different measurements to match the similarity between words and consider the frequency information. For CB, it measures the similarity between items, and the RS will recommend similar items based on users' preference or previous similar items. For CF, it measures the user-to-user similarity, which is recommend items to the similar new user based on the previous users' experience and preference. In TRS, CF will measure the similarity between each patient based on their age, gender, symptoms, and conditions. When recommending treatments, patients will get a list of recommended treatments that may match their conditions. All those matches are based on similarities. To find similar patients, I will use the size of the union of previous patients and new patients divide by (N1 + N2) to get the ratio as the similarity. Then, use the gender and range of age to filter the patient information. To cover more possibility, I will consider the top k similar users' treatment and mix together to get the final recommendation result. We could try different k to see the TRS evaluation difference.

3. Similarity: NLP is applied to find the synonyms, match semantic similarity, word order similarity, word similarity, etc. [11]. From [11], authors introduced a workflow of calculating sentence similarity using a lexical database (WordNet), which has connections between words, and corpus statistics in NLP. Basically, follow and implement the workflow to measure the sentence similarity. In this project, there is no sentence, thus, we consider a group of words as a sentence, such as a group of symptoms. The algorithm considered the sentence as word sequence, thus, that's the reason why could use sentence similarity measure in this project.

There are three different similarity measures in the workflow. The first one is word semantic similarity. Assume the similarity between two words is based on both length and depth. We find the shortest path distance between the synsets of the two words for comparison based on path similarity. Then, find the scaling depth between two words based on the hierarchical structure of knowledge. This is the hierarchical distance.

The second one is sentence semantic similarity. Sentence semantic similarity is measured by cosine similarity between semantic vectors. Semantic vector is gotten from the information-content weighted words, where information-content is negative correlation with the word appear frequency: high frequent words have less information, low frequent words have more information.

The last one, word order similarity. In some extent, like sentence "abc" and "cba", although they have the same Bag of Words, but are totally different sentence. Here comes the word order similarity. We firstly vectorize the sentence (in this project, it should be the group of words, like symptoms) to show the basic structure of sentence, which is called word order vector in the paper. Then, measure the difference between two word order vectors to calculate the word order similarity. Finally, we combine the sentence semantic similarity and word order similarity together, because the meaning of sentences should be represented by both the sentence content and word order [12].

For the corpus, I prefer to find one contains biomedical or medicine ontology for the use of implement word association and semantic similarity more accurately. Use specific corpus will improve the performance and lower the time complexity. Use Spacy model 'en_core_web_lg' is expensive for space and time.

4. Use word association to help find possible matching symptoms or conditions. Find ways to find medical terms synonyms from certain corpus or from knowledge base like NLTK WordNet. After we get the synonym sets for specific words, we push back to check if there are any of them hit the symptoms we already had in the dataset. If there exist new hit, we could add them into the check list and compute the similarity again to find the potential choice for patients. This is the idea of blurry searching. Also, I could try to blurry the condition to some similar conditions if I use the symptoms to predict the condition.

5. After finish using the symptoms and condition to recommend treatments, I will try to separate the recommendation into two stages. The first stage is to use the age, gender, and symptoms information to predict the possible conditions based on patient similarity. This might be a little bit challenge, because the symptoms may occur for several conditions and lead to the wrong prediction, especially will not use ML/DL models but just use similarity measures. Then, we could use the top k condition we predicted, or the patients or professionals recognized to final recommend based on those k conditions.

6. Evaluation. Because this is a recommendation system, the baseline I will use the random recommendation to do the comparison to see if the TRS is learned from the data or not. This project is offline project and will use the offline evaluation techniques for the recommendation system. The metrics I chose will be Precision, Recall, and f1 score [13]. The comparison of prediction and label in the classification problem is change to the correct and cover rate in the Recommendation System problem. Precision shows the percent of correct recommendation by the number of recommendation results. Precision measures the system's ability to reject any nonrelevant data in the dataset. Recall

measures the ability to find all relevant results. Precison@k for top k recommendation is

$$\text{Precison@k} = \frac{|y \cap \hat{y}|}{|\hat{y}|}$$

Recall@k for top k recommendation is:

$$\text{Recall@k} = \frac{|y \cap \hat{y}|}{|y|}$$

F1 score for top k recommendation is:

$$\text{F1} = \frac{2 * Precision@k * Recall@k}{Precision@k + Recall@k}$$

References

[1] Ko, H., Lee, S., Park, Y., & Choi, A. (2022, January 3). *A survey of recommendation systems: Recommendation models, techniques, and Application Fields*. MDPI. Retrieved March 23, 2022, from https://www.mdpi.com/2079-9292/11/1/141

[2] Y. Bao and X. Jiang, *An intelligent medicine recommender system framework,* 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), 2016, pp. 1383-1388, doi: 10.1109/ICIEA.2016.7603801.

[3] Sahoo, Abhaya Kumar, et al. "DeepReco: Deep Learning Based Health Recommender System Using Collaborative Filtering." *Computation*, vol. 7, no. 2, 22 May 2019, p. 25, 10.3390/computation7020025. Accessed 8 Nov. 2019.

[4] Ali, Syed Imran, et al. "A Hybrid Framework for a Comprehensive Physical Activity and Diet Recommendation System." *Khu.elsevierpure.com*, Springer Verlag, 2018, khu.elsevierpure.com/en/publications/a-hybrid-framework-for-a-comprehensive-physical-activity-and-diet-2. Accessed 24 Mar. 2022.

[5] Khoie, Mohammad, et al. "A Hospital Recommendation System Based on Patient Satisfaction Survey." *Applied Sciences*, vol. 7, no. 10, 21 Sept. 2017, p. 966, 10.3390/app7100966.

[6] Gräßer, Felix & Beckert, Stefanie & Küster, Denise & Abraham, Susanne & Malberg, Hagen & Schmitt, Jochen & Zaunseder, Sebastian. (2017). Neighborhood-based Collaborative Filtering for Therapy Decision Support.

[7] Ju, Chunhua, and Shuangzhu Zhang. "Doctor Recommendation Model Based on Ontology Characteristics and Disease Text Mining Perspective." *BioMed Research International*, vol. 2021, 8 Aug. 2021, p. 7431199, www.ncbi.nlm.nih.gov/pmc/articles/PMC8379386/, 10.1155/2021/7431199. Accessed 24 Mar. 2022.

[8] Han, Qiwei & Martínez de Rituerto de Troya, Íñigo & Ji, Mengxin & Gaur, Manas & Zejnilovic, Leid. (2018). A Collaborative Filtering Recommender System in Primary Care: Towards a Trusting Patient-Doctor Relationship. 10.1109/ICHI.2018.00062.

[9] "Treatment-Recommender/Crawler at Master · SooryaR/Treatment-Recommender." *GitHub*, github.com/sooryaR/Treatment-Recommender/tree/master/Crawler. Accessed 24 Mar. 2022.

[10] Isinkaye, F.O., et al. "Recommendation Systems: Principles, Methods and Evaluation." *Egyptian Informatics Journal*, vol. 16, no. 3, Nov. 2015, pp. 261–273, www.sciencedirect.com/science/article/pii/S1110866515000341, 10.1016/j.eij.2015.06.005.

[11] Pawar, Atish, and Vijay Mago. *IEEE TRANSACTIONS on KNOWLEDGE and DATA ENGINEERING 1 Calculating the Similarity between Words and Sentences Using a Lexical Database and Corpus Statistics*.

[12] Li, Yuhua & McLean, David & Bandar, Zuhair & O'Shea, James & Crockett, Keeley. (2006). Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Transactions on Knowledge and Data Engineering. 18. 1138-1150. 10.1109/TKDE.2006.130.

[13] Jia, Zheng, et al. "A Patient-Similarity-Based Model for Diagnostic Prediction." *International Journal of Medical Informatics*, vol. 135, 1 Mar. 2020, p. 104073, reader.elsevier.com/reader/sd/pii/S1386505619310925?token=34AD41A20F588E512AA0C530C08D555695C3F9216FA2410F1784165C62E4919FB17D447F2231FDCA20E9504FEE823508&originRegion=us-east-1&originCreation=20220324073354, 10.1016/j.ijmedinf.2019.104073. Accessed 24 Mar. 2022.