DRUG RECOMMENDATION SYSTEM BASED ON PATIENT-DESCRIBED SYMPTOMS

ANDRE ATTARD* and FRANCESCA MARIE PSAILA*, University of Malta, Malta

1 INTRODUCTION

News articles describing the shortage of health workers both locally and abroad are increasingly common. Indeed, the World Health Organization (WHO) estimates that by 2030 there will be a shortage of 10 million health workers worldwide [1]. Such severe understaffing impacts the mental health of medical professionals [2] which in-turn affects the quality of patient care [3]. An increase in patient misdiagnosis rates and increase in prescription errors are two detrimental outcomes of an overburdened health system [4], [5].

Individuals are now more likely to search self-diagnosis and health information online [6]. According to [7], around 35% of individuals attempt to obtain a diagnosis for their health condition online. Keeping this in mind, a shortage in medical personnel hindering individuals from accessing legitimate medical resources could increase the likelihood of self-diagnosis which can be of more harm than good [7], [8]. The use of approved online recommendation systems to both diagnose diseases and recommend medication has become increasingly popular [9]. Such systems have been found to improve clinician efficiency and accuracy whilst diagnosing a condition [10]. Meanwhile at a patient level, reliable information on possible medications is also provided by such systems [9].

In the following article, we propose a system which initially utilizes a summary of patient symptoms to provide a disease diagnosis. Based on this diagnosis, the system will then make use of a drug database to suggest a suitable medical treatment. In addition, any treatment recommended will also consider patient specific circumstances such as alcohol use or pregnancy.

2 RELATED WORK

The advent of BERT[11] (Bidirectional Encoder Representations from Transformers) and its variants like DistilBert [12] has significantly advanced the field of natural language processing (NLP) within healthcare [13]. The complexity of medical language, with its unique vocabulary and syntax, presents a unique challenge. These models have shown exceptional capability in handling the intricacies of medical language, enabling more efficient processing and interpretation of medical text.

Kanwal & Rizzo [13] present an application of BERT models in the healthcare domain. In this study a BERT base model was fine-tuned on around 50,000 discharge notes from the Medical Information Mart for Intensive Care (MIMIC-III) dataset. The model was trained to identify the ICD-9 disease labels based on the symptoms described and the diagnostic information available in the detailed clinical notes. Their method focuses on extracting meaningful phrases from the notes by analysing the correlation between tokens, segments, and positional embeddings in the notes.

The clinical documents were pre-processed and tokenized. The representation of each token was created by merging token embeddings, position embeddings, and segment embeddings. The [CLS] token, serves as the initial token for sentence classification and the [SEP] token marks the conclusion of the sequence. The base-variant of the BERT model was fine-tuned using maximum sequence length, a batch size of 8, an Adam optimizer with epsilon 1×10^{-8} and a learning rate of 3×10^{-5}. All other hyper-parameters of the pre-trained BERT model were left unchanged. Fine-tuning was found to have a significant effect on performance of base-BERT for the supervised task of classifying the [CLS] token for maximum likelihood of ICD-9 label.

Although focused on summarizing clinical notes rather than classification, the methodology underscores the potential of attention mechanisms in distilling relevant information from extensive medical documents. Moreover, the study illustrates the versatility of transformer models in processing clinical narratives, providing a foundation for exploring similar techniques in symptom-based condition prediction.

Maniar et al. [4] identified high misdiagnosis rates as a core issue within the healthcare system. Indeed, they reference multiple studies which point out the prevalence of misdiagnosis within the United States (U.S) [10], [14]. Their solution was to utilize unstructured data, available in the form of medical transcription reports, to build a model which is able to categorise any unseen records into one of four categories: 1) heart, 2) brain, 3) reproductive and 4) digestive. Mainer et al. [4] made use of the following text preprocessing techniques to handle unstructured data found in medical transcripts:

(1) Data Cleaning
   - Removal of numbers, symbols, punctuation and special characters
   - Input lowercase
   - Sentence tokenization
   - Stop-word removal
   - Word lemmatization
(2) Conversion to numerical vectors
   - Term Frequency Inverse Document Frequency (TFIDF) vectorization preprocessing algorithm

Glove and Word2Vec were also mentioned as alternatives to the TFIDF algorithm. All text preprocessing techniques mentioned can be utilised in our solution. Following the application of the above-mentioned preprocessing techniques, the following models were utilised for classification purposes: 1. Machine Learning Models a. Logistic Regression b. Random Forest 2. Deep Learning Algorithms a. Long Short-Term Memory Network (LSTM) b. Convolutional neural network - Long Short-Term Memory Network (CNN - LSTM) Both deep learning algorithms, were found to perform the best based on the results of four performance metrics (Accuracy, Precision,

*Both authors contributed equally to this research.

Authors' address: Andre Attard, andre.attard.17@um.edu.mt; Francesca Marie Psaila, francesca.psaila.17@um.edu.mt, University of Malta, Malta.

2024. ACM XXXX-XXXX/2024/2-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Recall and F1-Score) when each model was applied on a test set. Alternative models such as BERT and GPT were also highlighted as alternatives which might be of interest for future research. Based on these results, our solution would benefit from using either one of the Deep learning algorithms applied in [4] or possibly attempting to utilize the BERT/GPT model. A web application entitled 'MedicAI' was created to make the above process accessible to clinicians. Therefore, clinicians are able to use this application to classify any medical transcript record in an efficient manner and with a relatively high degree of accuracy.

2.1 Datasets Description

To proceed with our solution, we will be making use of both a structured and unstructured dataset. The unstructured dataset will be utilized to provide a disease diagnosis. Meanwhile the structured dataset will be utilised to recommend the appropriate treatment. Further detail, regarding both datasets, will be provided in Sections 2.1.1 and 2.1.2.

2.1.1 Structured Dataset. Structured datasets consist of organized collections of data presented in a predetermined format, enabling systematic storage, retrieval, and analysis [15]. The structured dataset used in this study was obtained by making use of the Selenium python package to manually web scrape the Drugs.com website [16]. The python code utilised for this task is available in the Jupyter notebook entitled `'WebScraping_DrugsDotCom.ipynb'`. An additional variable entitled `Drug_Category` can be obtained by utilising the code available in `'WebScraper_DrugClasses.ipynb'`. Selenium is used to control a web browser programmatically and the following is a high-level overview of the process to scrape the required data:

(1) The homepage of the Drugs.com [16] website is accessed.
(2) Any user interactions (such as accepting cookies) are handled
(3) The medication pages for a total of 15 conditions[1] are accessed one at a time.
(4) In each medication page, the list of drugs, drug user score, number of ratings, pregnancy category, controlled substance act category and alcohol interaction category are stored in a list.
(5) Data is then saved in a CSV file entitled `'WebScrapingOutput.csv'` or `'drug_classes_final.csv'` depending on jupyter notebook utilised.

Table 2 provides a brief description of each variable in the dataset.

The Pregnancy_Category variable can take any of the following values:

(1) B - Animal reproduction studies have failed to demonstrate a risk to the fetus
(2) C - Animal reproduction studies have shown an adverse effect on the fetus, but potential benefits may warrant use of the drug in pregnant women despite potential risks.
(3) D - There is positive evidence of human fetal risk based on adverse reaction data, but potential benefits may warrant use of the drug in pregnant women despite potential risks.
(4) x - Studies in animals or humans have demonstrated fetal abnormalities, and the risks involved in use of the drug in pregnant women clearly outweigh potential benefits.
(5) N - FDA has not classified drug

The Drug_Abuse_Category variable can take any of the following values:

(1) 2 – High potential for abuse
(2) 3 - Some potential for abuse
(3) 4 - Low potential for abuse.
(4) 5 - Lower potential for abuse (relative to 4)
(5) N - The drug is not subject to the Controlled Substances Act
(6) U - CSA Schedule is unknown
(7) M - The drug has multiple schedules. The schedule may depend on the exact dosage form or strength of the medication.

The Alcohol_Category variable can take any of the following values:

(1) X – Interacts with Alcohol

2.1.2 Unstructured Dataset. Unstructured datasets consist of loosely organized information lacking a predefined format, making them less easily searchable and analyzed compared to structured datasets [15]. The unstructured dataset used in this study is freely available on Kaggle [20] and has been compiled by Kaggle users Jessica Li and Rachel. The data was published on the 13th of November 2018 [21]. The dataset was originally published on the UCI Machine Learning repository [22] website, however as of January 2024 the data is no longer available from this source.

Table 1. Description of Variables in structured dataset

Variable Name	Description
Drugs	List of drugs
Conditions	Corresponding list of conditions
Scores	Average rating given to each drug
Number_of_Ratings	Number of individuals who rated drug [17]
Pregnancy_Category	Drug safety category during pregnancy [17]
Drug_Abuse_Category	Abuse potential Category of drug [18]
Alcohol_Category	Drug and alcohol interaction category [19]
Drug_Category	Classification of drug structure[18]

Table 2. Description of Variables in unstructured dataset

Variable Name	Description
uniqueID	Unique Identification Number
drugName	Name of drug
condition	Name of condition
review	Patient reviews
rating	Ten-Star patient ratings
Date	Date of review
usefulCount	Number of users who found review useful

As we will be utilizing the unstructured dataset to build a model which will provide a disease diagnosis, only the condition and review variables are relevant. Therefore, remaining variables will be

[1]Conditions of interest are ADHD, HIV Infection, Bipolar Disorder, Anxiety, Panic Disorder, Rheumatoid Arthritis, Depression, Diabetes, Type 2, Erectile Dysfunction, GERD, Allergic Rhinitis, Irritable Bowel Syndrome, Osteroarthritis, Overactive Bladder and Pain

dropped. In addition, to reduce heterogeneity our solution will be reduced to the 15 conditions described in Section 2.1.1

While our solution could also utilize the drugName, rating and usefulCount variables to suggest a suitable medical treatment, by making use of the structured dataset defined in Section 2.1.1 we will be including more recent treatments for each corresponding condition. On the other hand, this unstructured dataset can be utilised for disease diagnosis as it is unlikely that symptoms for relatively common diseases have changed since 2018.

3 DESIGN AND IMPLEMENTATION

3.1 Initial Data Pre-Processing

Initial data pre-processing steps were applied on both structured and unstructured datasets to ensure that both datasets can be used to obtain a final solution.

The initial data pre-processing steps applied on the unstructured dataset were as follows:

(1) Check for null values - No null values were found.
(2) Check for duplicate reviews - Reviews by the same individuals for multiple drugs treating the same condition were removed. The initial dataset contained 53, 400 entries. However, upon ensuring that only unique review per condition is available the total number of entries became 31, 695.
(3) Removal of UniqueID, drugName, rating, Date and usefulCount columns - condition and review columns were relabeled to label and text respectively.

Following these initial pre-processing steps, figure 1 represents the count plot of conditions in unstructured dataset.

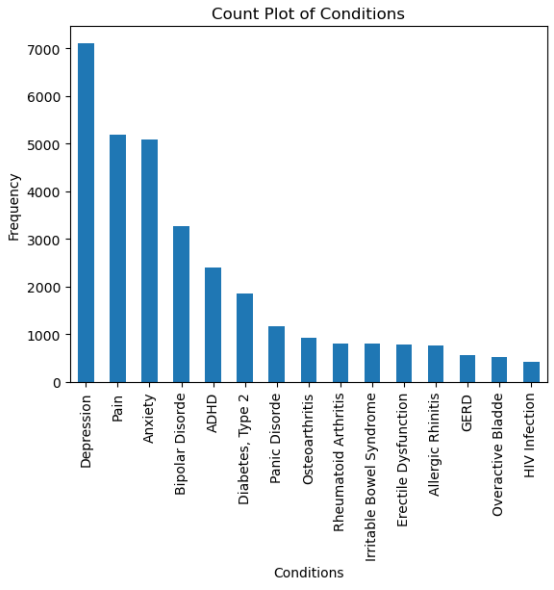

Fig. 1. Count Plot of Conditions

The structured dataset did not require as much initial data processing since we have manually web scraped the data ourselves. As a result, the only step taken was to ensure that any missing values under the Pregnancy_Category, Drug_Abuse_Category and Alcohol_Category were not null due to incomplete fields within the Drugs.com [16] website. Any missing entries in all three variables was replaced with value **U** indicating that value is unknown.

3.2 Text Pre-processing

Raw data needs to be transformed into a machine-understandable format for analysis. Data pre-processing techniques standardize and refine input data, eliminating less significant characteristics, thereby enhancing the models' predictive accuracy regarding medical conditions. To prepare the textual data for analysis with BERT and LSTM models, we employed a common pre-processing approach utilizing the Natural Language Toolkit (NLTK) in Python. The implemented techniques included:

- Lowercasing to ensure uniformity of input words
- Conversion of HTML entities to their corresponding characters to clean the data of web artifacts
- Removal of special characters to exclude irrelevant punctuation and symbols
- Sentence Tokenization
- Sentence Part-of-Speech tagging where each token was annotated with its grammatical role. Tokens were then filtered to keep only nouns, verbs, and important adjectives, with the aim of concentrating on the words most likely to carry significant meaning
- Stop word removal
- Word lemmatization to reduce text complexity

The code used for the pre-processing of the unstructured input data can be found in the python file `'pre_process.py'`.

While the pre-processing steps are designed to optimize the textual data for predictive modelling, we acknowledge some limitations inherent to the approach. The selective retention of specific parts of speech may inadvertently lead to the loss of context. Furthermore, the processes of lemmatization and stop word removal, while beneficial in reducing data complexity, risk oversimplifying the text. This could result in potentially overlooking the subtleties of patient descriptions.

3.3 Word Clouds

Following text pre-processing and prior to building DistilBERT and LSTM models, we decided to analyse the word cloud figures of a number of conditions. Word clouds are useful in interpreting the content of a large several textual documents in an instant [23]. The word cloud for the most frequent conditions available in the unstructed dataset according to figure 1 will be presented below.

The word cloud for depression is shown in figure 2:

The word cloud for pain is shown in figure 3:

The word cloud for anxiety is shown in figure 4:

An interesting observation is that in all three word clouds, the corresponding condition name is a significant term. In addition, the terms side and effect are significant terms in all three word clouds. This is expected, since the collection of texts are drug reviews left by patients. In a similar manner, terms like year, month, week, day and error are significant in all three word clouds possibly indicating the length of treatment.

Fig. 2. Word Cloud for Depression

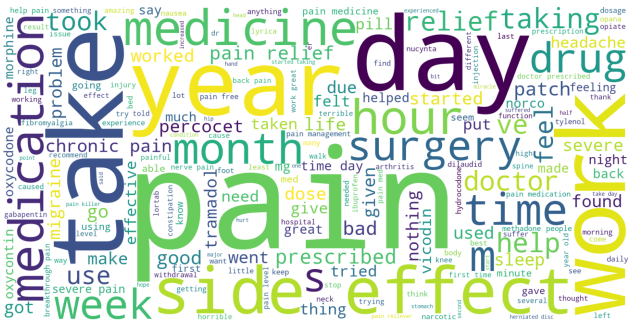

Fig. 3. Word Cloud for Pain

Fig. 4. Word Cloud for Anxiety

3.4 Machine Learning/ Deep Learning

3.4.1 DistilBERT. For the task of condition prediction, we employ DistilBERT, which is a lighter and faster variant of BERT which retains a lot of its performance capabilities [12]. Before implementing the model, the textual data is pre-processed using the techniques described in Section 3.2. Each review within our dataset underwent tokenization, using `DistilBertTokenizer`, where [CLS] tokens were added at the beginning to signify the start of a sentence and [SEP] where added at the end as delimiters, to indicate the end of a sequence. These tokens are crucial for BERT to understand the start and end of sentences, respectively. To accommodate the model's maximum input length, reviews were truncated or padded to 512 tokens, and attention masks were generated to distinguish real tokens from padding, facilitating the model's focus on relevant information. We initialized a `DistilBertForSequenceClassification` model configured to classify among the 15 distinct condition labels. The model was adapted to run on a GPU for accelerated computation.

The base variant with 12 layers was used and fine-tuned for our classification task. The batch size used was 16 and the number of epochs utilized for training was 3. Training involved the `AdamW` optimizer with a learning rate of 2×10^{-5} and an epsilon value of 1×10^{-8}, selected to minimize the loss function effectively while preventing overfitting. All other hyperparameters of the base model were left unchanged. Each training epoch comprehensively processed the dataset, with gradients computed and adjusted through backpropagation. To mitigate the exploding gradients issue, we clipped gradients to a maximum norm of 1.0. Additionally, a linear scheduler adjusted the learning rate across training epochs to optimize convergence.

The data was split into a training and validation set using an 80-20 split. Post-training, the model's performance was evaluated on a validation set, assessing accuracy to ensure the model generalizes well beyond the training data. The code for Fine-tuning the model, adapted from [24], can be found in the Jupyter notebook `'Fine_tuning_BERT.ipynb'`.

The fine-tuned model path is saved, and the state dictionary which contains all the weights and biases of the model is saved using `PyTorch`. The model was then evaluated on a test set, and the macro-averaged metrics of Average, Recall, Precision and F1-score were obtained. The code for the model evaluation is found in the Jupyter notebook `'Model_eval.ipynb'`.

3.4.2 Neural Network Model.

3.4.3 Word2Vec. Prior to utilising a deep learning model to obtain classification results, words must be represented as vectors of real numbers [25] referred to as word embeddings. As described in [4], the Word2Vec algorithm can be used to convert string tokens obtained following the text pre-processing steps in Section 3.2 into numerical vectors. In our solution, the dataset itself has been used to train the word embeddings. However, it is possible to use pre-trained word embeddings such as Google's Word2Vec [26].

The Word2Vec model was developed using the `'models.word2vec'` module within the Gensim python library [27]. This module enabled us to utilise our text dataset and the continuous bag of words (CBOW) algorithm [28] to develop word embeddings. The CBOW model considers a particular word and utilizes the surrounding words to predict this word. In our case, we utilized the four preceding words and four following words to predict the word in the middle. In addition, the number of dimensions of the embeddings was set to 100 and the minimum count of words to be considered when training model was set to 1 (all words considered). The code defining the above process is available in the `'Word2Vec_LSTM_FUNCTIONS.ipynb'` jupyter notebook.

3.4.4 Long Short-Term Memory Neural Network. Once the word embedding is obtained using the Word2Vec algorithm discussed in Section 3.4.3 we are now able to utilise any machine learning or deep learning algorithm [25]. As described in Section 2, Maniar et al.

[4] found that the Long Short-Term Memory (LSTM) algorithm performed relatively well and hence we will be using this deep learning algorithm to build a classification model for our unstructured text dataset.

The Long-Short-Term Memory (LSTM) model is a variant of Recurrent Neural Networks (RNNs). RNNs utilise hidden states to hold on to information about a previously inputted sequence of data [29]. RNNs are trained using the Backpropagation are trained using the Backpropagation through time (BPTT) algorithm. In the BPTT algorithm, the gradient of the cost function is calculated at each time step and is used to update the model parameters [30]. RNNs suffer from what is known as the vanishing gradient problem. This is when the gradient during BPTT decreases to such a degree that it becomes difficult to update the parameters of the model [29].

To counter the vanishing gradient problem, the LSTM model introduces a special memory unit which is able to store information for a long period of time [31]. This unit is controlled by three gates referred to as (1) Then input gate, (2) The output gate and (3) the forgetting gate which dictate when information should be written, read and forgotten in the special memory unit [31].

Our neural network model consists of three layers. An embedding layer is used as an input layer to convert words into vectors of real numbers as described in Section 3.3.1. This is followed by an LSTM layer applied using the `'tf.keras.layers.LSTM'` module within the TensorFlow python library [32]. The number of units representing the dimensionality of the output space was set as 64. The final layer is a dense layer used as an output layer with a softmax activation function. A softmax activation function is utilised since we have multiple outputs for classification [33]. This neural network model was then trained using a categorical cross-entropy loss and 5 epochs. The code defining this process is available in the `'Word2Vec_LSTM_FUNCTIONS.ipynb'` jupyter notebook with model training taking place in the `'Word2Vec_LSTM_Model.ipynb'` jupyter notebook. Likewise model evaluation methods are defined in the former notebook and called in the latter notebook.

4 RESULTS

Prior to training model, dataset was split into a training (70% of data) and testing (30% of data) set. The following metrics were utilised to evaluate model based on how well it is able to predict the conditions of the testing set:

- Accuracy
- (Macro-Average) Precision
- (Macro-Average) Recall
- (Macro-Average) F1-Score

Table 3 provides a brief description of each variable in the dataset.

Table 3. Disease Diagnosis Results

Classifier	Accuracy	Precision	Recall	F-Measure
DistilBERT	0.826	0.856	0.801	0.826
LSTM	0.771	0.741	0.783	0.756

The DistilBERT model obtains better performance metric values when compared to the LSTM neural network model. However, based on these results, both models seem to perform well at classifying the text data provided into one of the fifteen conditions. In the next Section, a knowledge graph built on the structured database will be used to answer a number of competency questions with significance given to predicting the highest recommended drug for a given condition.

4.1 Knowledge Graph

Using the structured database for drugs obtained from Drugs.com, a knowledge graph (KG) is developed which incorporates the data available. Studies [34],[35] have highlighted the usefulness of KGs in developing automated medicine recommendation systems for an AI-driven diagnosis and prescription. With the constructed KG we aim to enable the discovery of helpful drugs based on the predicted condition from the described symptoms.

The KG consists of 3 node types: *Condition*, *Drug*, and a third entity is introduced, *Drug_Class* (e.g., CNS Stimulants, Antihistamines, etc.). The *Drug* is a subclass of *Drug_Class*. The drug classes were obtained through a web-scraping procedure from Drugs.com. *Drug* nodes carry attributes; `Pregnancy_Category`, `Controlled_Substance_Abuse_Category`, and `Alcohol_Category`, to provide insight into the drug's safety profile.

An edge exists for each *Drug*, linking it to a *Condition*, with the relationship defined as 'TREATS'. This relationship has properties of `scores` and `number_of_ratings`, which offer a quantitative measure of the drug's overall user-determined effectiveness against the condition. Furthermore, drugs belonging to a drug class are linked by a 'BELONGS TO' edge relation.

By analysing the contents of the resulting heterogeneous graph, we aim to provide statistics on the nodes and relationships through a series of graph embedding-based queries. Namely, we aim to extract the relevant information from the KG to answer the following competency questions:

(1) Which is the highest recommended drug for a given condition?
(2)(a) Which drugs are unsafe for pregnant patients, for a given condition?
   (b) Which drugs have a high abuse potential, for a given condition?
   (c) Which drugs are known to interact with Alcohol, for a given condition?
(3) What are the common drug classes for drugs treating a specific condition?
(4) What is the highest rated drug in each Drug Class for a specific condition?
(5)(a) Which drugs are used to treat multiple conditions?
   (b) Which condition pairs have the highest number of shared drugs?

The KG is created using Neo4j, and using the Python library `py2neo` to interact with the Graph DBMS. Figure 5 displays the Knowledge Graph created where the red nodes represent the conditions, the purple nodes represent the individual drugs and the yellow nodes represent the drug classes. The graph displays distinct clusters, representing groups of drugs related to specific conditions.

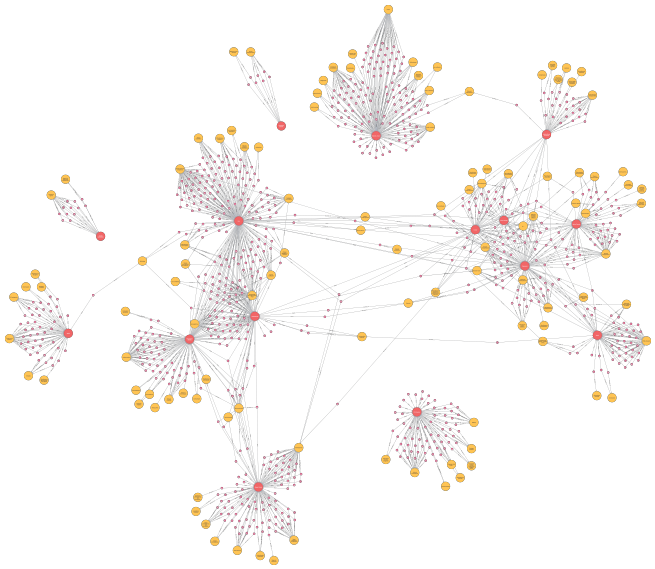

Fig. 5. Drug-Condition Knowledge Graph

However, some drug nodes serve as bridges between these clusters, suggesting their potential usage for treating multiple conditions.

To answer the questions posed, we employ various cipher queries against our knowledge graph. Below are examples of how we approach these queries.

```
query = (
f"""MATCH (d:Drug)-[r:TREATS]->(c:Condition {{name: "{condition_name}"}})
WHERE r.number_of_ratings >= 30
RETURN d.name AS Drug, r.score AS Score, d.Pregnancy_Category, d.Alcohol_Category, d.Controlled_Substance_Abuse_Category
ORDER BY r.score DESC
LIMIT 1""")
```

Fig. 6. Cipher query to answer 'Which is the highest recommended drug for a given condition?'

The query displayed above is used to retrieve the drug with the highest user rating which treats a specified condition. The query is filtered to include only those drugs that have at least 30 ratings, ensuring that the recommendations are based on a substantial amount of user feedback. In addition to the drug's name and score, the query also returns critical safety information about each drug, including its `Pregnancy_Category`, `Alcohol_Category`, and `Controlled_Substance_Abuse_Category`.

```
query = """MATCH (c1:Condition)<-[:TREATS]-(d:Drug)-[:TREATS]->(c2:Condition)
WHERE id(c1) < id(c2)
WITH c1.name AS Condition1, c2.name AS Condition2, COUNT(d) AS SharedDrugs
WHERE SharedDrugs > 1
RETURN Condition1, Condition2, SharedDrugs
ORDER BY SharedDrugs DESC"""
```

Fig. 7. Cipher query to answer 'Which condition pairs have the highest number of shared drugs?'

The query illustrated above is used to identify pairs of conditions that are treated by overlapping sets of drugs. The results are then ordered by the number of shared drugs in descending order, highlighting those pairs with the highest degree of overlap in their treatment profiles. This query can help identify diseases which respond to similar treatments, lending itself to potential drug re-purposing opportunities.

The python code utilised for creating the Knowledge graph in Figure 5 and the queries defined for answering the competency questions are available in the Jupyter notebook entitled `KnowledgeGraph.ipynb`. The full code that processes user-described symptoms in text form to predict medical conditions, and subsequently recommends drugs based on the number of ratings and reviews, can be found in the Jupyter notebook `'DrugRecommendationDEMO.ipynb'`. A similar implementation, applicable to the LSTM, was carried out in the `'Word2Vec_LSTM_FUNCTIONS.ipynb'` jupyter notebook with recommended medication shown `'Word2Vec_LSTM_Model.` jupyter notebook.

5 CONCLUSION

The performance metric results obtain by both models, as shown in Section 4, indicate that both models perform well at classifying the text data provided into one of the fifteen conditions. However, this relatively high degree of performance could be a result of the unstructured dataset utilised. Reviews of drugs are generally made by individuals who have already been diagnosed and hence are already aware of what symptoms define their condition. This phenomena was apparent in the word clouds discussed in Section 3.3, where the condition itself was a significant term. Individuals who have not been diagnosed might not be able to describe their symptoms with such accuracy.

On the other hand, the main focus of drug reviews is not to provide a description of symptoms but rather to chronicle the effectiveness of the medication prescribed. Hence, our classification models utilised somewhat unrelated text data which might have hindered performance. To address both concerns, future research can focus on obtaining medical transcript data written prior to patient diagnosis. In our case, this proved to be relatively difficult for us to do since the number of medical transcript sources with a meaningful number of entries are not readily available online.

Related to data availability, we were unable to obtain a publicly available dataset consisting of information on drugs which could also be paired with unstructured dataset to obtain a working solution. Hence manual scraping was utilised to obtain a structured dataset consisting of not only condition and drug pairs but additional information such as how the drug interacts with alcohol. Future research could expand further on this dataset by including information such as recommended drug dose and side effects. Through ontology alignment, data could be incorporated from various comprehensive drug databases or repositories, for instance, DrugBank or SIDER. This information could allow one to increase detail of knowledge graph shown in Section 4.1.

The count plot, shown in Section 3.1, implies that the conditions represented in the unstructured dataset are imbalanced. While Precision, Recall and F-Measure values for both models are still relatively high future research could utilise some form of over-sampling/under-sampling technique to combat imbalance in unstructured dataset.

Apart from changes impacting dataset, future research can also work on improving both BERT and Long Short-Term Memory (LSTM) Neural Network models. Using a more domain-specfic BERT model pre-trained on medical text [36, 37], has been shown to provide improvement in performance [38]. In addition, the LSTM neural network model could be improved by utilizing Grid-Search/Random-Search hyperparameter tuning to select the optimal parameter values.

REFERENCES

[1] World Health Organization, "Health Workforce," World Health Organization. [Online]. Available: https://www.who.int/health-topics. [Accessed: 25th January 2024].
[2] A. Rimmer, “Staff shortages are affecting doctors\textquoteright mental health, survey finds,” vol. 381, 2023, doi: 10.1136/bmj.p1121.
[3] R. Taylor-East, A. Grech, and C. Gatt, “The mental health of newly graduated doctors in Malta,” pp. S250-5, Sep. 2013.
[4] K. Maniar, S. Haque, and K. Ramzan, “Improving Clinical Efficiency and Reducing Medical Errors through NLP-enabled diagnosis of Health Conditions from Transcription Reports,” 2022.
[5] R. Despott, J. Sultana, L. Camilleri, J. Vella Szij, and A. Serracino Inglott, “Risk management of medication errors: a novel conceptual framework,” vol. 24, no. 4, pp. 523–534, Mar. 2023, doi: 10.1080/14656566.2023.2178899.
[6] A. Farnood, B. Johnston, and F. S. Mair, “A mixed methods systematic review of the effects of patient online self-diagnosing in the ‘smart-phone society’ on the healthcare professional-patient relationship and medical authority,” vol. 20, no. 1, pp. 253-020-01243–6, Oct. 2020, doi: 10.1186/s12911-020-01243-6.
[7] S. Fox and M. Duggan, "Health Online 2013," Pew Research Center: Internet, Science & Tech, United States of America, 2013. [Accessed: 04-Feb-2024]. [CID: 20.500.12592/kptb86]
[8] C. Silpa, B. Sravani, D. Vinay, C. Mounika, and K. Poorvitha, “Drug Recommendation System in Medical Emergencies using Machine Learning,” 2023, pp. 107–112.
[9] A. Sae-Ang et al., "Drug Recommendation from Diagnosis Codes: Classification vs. Collaborative Filtering Approaches," International Journal of Environmental Research and Public Health, vol. 20, no. 1, p. 309, 2022. [Online]. Available: https://doi.org/10.3390/ijerph20010309
[10] H. Singh, A. N. Meyer, and E. J. Thomas, "The frequency of diagnostic errors in outpatient care: estimations from three large observational studies involving US adult populations," BMJ Quality & Safety, vol. 23, no. 9, pp. 727-731, Sep. 2014. doi: 10.1136/bmjqs-2013-002627.
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, 'Attention Is All You Need,' arXiv preprint arXiv:1706.03762, 2017."
[12] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.”
[13] N. Kanwal and G. Rizzo, “Attention-based Clinical Note Summarization.”
[14] C. Koch, K. Roberts, C. Petruccelli, and D. J. Morgan, "The Frequency of Unnecessary Testing in Hospitalized Patients," American Journal of Medicine, vol. 131, no. 5, pp. 500-503, May 2018. doi: 10.1016/j.amjmed.2017.11.025.
[15] K. Batko and A. Ślęzak, "The use of Big Data Analytics in healthcare," Journal of Big Data, vol. 9, no. 1, p. 3, 2022. doi:10.1186/s40537-021-00553-4 [Accessed:29th December 2023]
[16] "Drugs.com," Drugs.com, Available: https://www.drugs.com/ [Accessed: 23rd December 2023].
[17] P. Sachdeva, B. G. Patel, and B. K. Patel, "Drug use in pregnancy; a point to ponder!," Indian Journal of Pharmaceutical Sciences, vol. 71, no. 1, pp. 1–7, 2009. [Online]. Available: https://doi.org/10.4103/0250-474X.51941
[18] DEA, "Drug Scheduling," DEA, 10 July 2018. [Online]. Available: https://www.dea.gov/drug-information/drug-scheduling. [Accessed: 2nd February 2024].
[19] "Medications and Alcohol," Drugs.com, [Online]. Available: https://www.drugs.com/article/medications-and-alcohol.html [Accessed: 2nd February 2024]
[20] Kaggle, "KUC Hackathon: Winter 2018," Kaggle, [Online]. [Accessed: 22nd December 2023].
[21] "KUC Hackathon: Winter 2018 Discussion," Kaggle, [Online]. [Accessed: 26th December 2023].
[22] "UCI Machine Learning Repository," [Online].[Accessed: 22nd December 2023]
[23] Y. Kalmukov, "Using word clouds for fast identification of papers' subject domain and reviewers' competences," arXiv:2112.14861 [cs.IR], 2021.
[24] N. Mohan, "Fine-tuning BERT for text classification," Kaggle, [Online]. Available: URL. [Accessed: 13th December 2023].
[25] Dr. J. Azzoparti, "– Mining Large Data Feature Extraction from Unstructured Data," ICS5111, University of Malta, Msida, Malta, [Accessed: 16th December 2023].
[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781 [cs.CL], 2013.
[27] "gensim: Topic Modeling for Humans," Radim Řehůřek, [Online]. Available: https://radimrehurek.com/gensim/. [Accessed: 13th January 2024].
[28] "Word2Vec - gensim: models," Ted Boy, [Online]. [Accessed: 13th January 2024].
[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv:1412.3555 [cs.NE], Cornell University Library, Ithaca, 2014.
[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986. doi: 10.1038/323533a0.
[31] Q. He, C. Wu, and Y. Si, "LSTM with Particle Swarm Optimization for Sales Forecasting," Electronic Commerce Research and Applications, vol. 51, 2022, Art. no. 101118. doi: 10.1016/j.elerap.2022.101118.
[32] "tf.keras.layers.LSTM," TensorFlow, [Online]. [Accessed: 14th January 2024].
[33] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," CoRR, vol. abs/1811.03378, 2018. [Online]. Available: http://arxiv.org/abs/1811.03378.
[34] M. Mann, F. Ilievski, M. Rostami, and B. Shbita, “Open Drug Knowledge Graph.”

[35] L. Li et al., “Real-world data medical knowledge graph: construction and applications,” vol. 103, p. 101817, Mar. 2020.
[36] Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So CKang, J. Biobert: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2019, 36, 1234–1240. [CrossRef] [PubMed]
[37] Huang, K.; Altosaar, J.; Ranganath, R. Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv 2019, arXiv:1904.05342.
[38] M. Khadhraoui et al, "Survey of BERT-Base Models for Scientific Text Classification: COVID-19 Case Study," Applied Sciences, vol. 12, (6), 2022. . DOI: 10.3390/app12062891.