2020

Drug Reviews: Cross-condition and Cross-source Analysis by Review Quantification Using Regional CNN-LSTM Models

Ajith Mathew Thoomkuzhy
Technological University Dublin

Recommended Citation

Thoomkuzhy, A. (2020). *Drug reviews: cross-condition and cross-source analysis by review quantification using regional CNN-LSTM models*. Masters Dissertation. Technological University Dublin. DOI:10.21427/fcve-2g86

Drug Reviews: Cross-Condition and Cross-Source Analysis By Review Quantification using Regional CNN-LSTM Models

Ajith Mathew Thoomkuzhy

A dissertation submitted in partial fulfilment of the requirements of

Technological University Dublin for the degree of

M.Sc. in Computing (Data Analytics)

January 2020

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: **Ajith Mathew Thoomkuzhy**

Date: **05 January 2020**

Abstract

Pharmaceutical drugs are usually rated by customers or patients (i.e. in a scale from 1 to 10). Often, they also give reviews or comments on the drug and its side effects. It is desirable to quantify the reviews to help analyze drug favorability in the market, in the absence of ratings. Since these reviews are in the form of text, we should use lexical methods for the analysis. The intent of this study was two-fold: First, to understand how better the efficiency will be if CNN-LSTM models are used to predict ratings or sentiment from reviews. These models are known to perform better than usual machine learning models in the case of textual data sequences. Second, how effective is it to migrate such information extraction models across different drug review data sets and across different disease conditions. Therefore three experiments were designed, first, an In-domain experiment where train and test data are from the same dataset. Two more experiments were conducted to examine the migration capability of models, namely cross-data source, where train and test are from different sources and cross-disease condition model training, where train and test data belong to different disease conditions in the same dataset. The experiments were evaluated using popular metrics such as RMSE, MAE, R^2 and Pearson's coefficient and the results showed that the proposed deep learning regression model works less successfully when compared to the machine learning sentiment extraction models in the literature, which were done on the same datasets. But, this study contributes to the existing literature in the quantity of research work done and in quality of the model and also suggests the future researchers on how to improve. This work also addressed the shortcomings in the literature by introducing dimensionality to sentiment, which represents user feelings better than sentiment polarity.

Keywords: Drug Reviews, Sentiment Extraction, Regional CNN, LSTM, Machine Learning, Model Migration, Cross-Data Source, Cross-Disease Condition, Linear Regression

Acknowledgments

I would like to express my sincere thanks to my Supervisor **Dr. Luis Miralles** for his valuable and timely guidance throughout this work. It was a great honour to work and study under his supervision.

Second, I would like to thank **Dr. Luca Longo** for helping me build my idea into a polished proposal. Without his comments, it would have been impossible to complete this dissertation.

Special thanks to all staff of the School of Computing in TUD for their sincerity and hard work in mentoring and helping me to complete this course and thesis.

I would also like to thank my friends **Dennis John** and **Tijo Thomas** and my classmates for assisting me in completing this work.

I should also mention everyone else who ignited the spark in me, supported and guided me through the entirety of this project.

Last but not least, I want to thank my beloved family for supporting me and constantly encouraging me to pull up my socks.

Contents

Declaration **I**

Abstract **II**

Acknowledgments **III**

Contents **IV**

List of Figures **VIII**

List of Tables **XI**

List of Acronyms **XII**

1 INTRODUCTION **1**

1.1 Background 1

1.2 Research Problem 2

1.3 Research Objectives 3

1.4 Research Methodologies 4

1.5 Scope and Limitations 5

1.6 Thesis Outline 6

2 LITERATURE REVIEW AND RELATED WORK **8**

2.1 Introduction 8

2.2 Literature and Existing Works on Text Quantification 9

2.2.1 Approaches to Solving Problem 10

2.2.2 Gaps in Research . . . 11
2.2.3 Existing Works on Drug Reviews . . . 12
2.2.4 Why Regional CNN-LSTM? . . . 13
2.3 TDSP and Data Mining Life-cycle . . . 14
2.4 Machine Learning: Definition and Algorithms . . . 17
2.4.1 Data Mining and Machine Learning . . . 17
2.4.2 Types of Machine Learning Algorithms . . . 18
2.5 Linear and Logistic Regression . . . 20
2.6 Neural Networks . . . 23
2.6.1 Convolutional Neural Networks (CNN) . . . 25
2.6.2 Recurring Neural networks and LSTM . . . 26
2.7 Text Mining . . . 28
2.7.1 Natural Language Processing and Named Entity Recognition . . . 30
2.7.2 Sentiment Extraction . . . 32
2.7.3 Word Vectorization . . . 34
2.8 Model Validation Techniques . . . 36
2.8.1 Fitting of the Model . . . 36
2.8.2 Cross Validation . . . 38
2.9 Model Evaluation Techniques . . . 41
2.9.1 Root Mean Square Error (RMSE) . . . 42
2.9.2 Mean Absolute Error . . . 42
2.9.3 R Squared (R^2) and Adjusted R^2 . . . 43
2.9.4 Pearson's Correlation Coefficient . . . 43
2.9.5 Cohen's Kappa Interrater . . . 44
2.10 Conclusion . . . 45

3 EXPERIMENT DESIGN AND METHODOLOGY 48

3.1 Introduction . . . 48
3.2 Design Flow . . . 49
3.3 Business Understanding . . . 49

3.4 Data Acquisition and Understanding 50
3.4.1 Data Acquisition 50
3.4.2 Data Understanding 52
3.4.3 Data Preparation 52
3.5 Modelling 53
3.5.1 Model Flow Definition 53
3.5.2 The Three Experiments 55
3.6 Conclusion 56

4 IMPLEMENTATION AND RESULTS 57
4.1 Introduction 57
4.2 Data Understanding 58
4.2.1 Data Feature: *review* 58
4.2.2 Data Feature: *rating* 59
4.2.3 Data Feature: *condition* 60
4.3 Data Preparation 61
4.3.1 Steps in Data Pre-processing 61
4.3.2 Word Embedding 62
4.4 Modelling 63
4.4.1 First Architecture: CNN-LSTM 63
4.4.2 Second Architecture: Regional CNN-LSTM 64
4.5 Results 66
4.5.1 Experiment 1: In-Domain 66
4.5.2 Experiment 2: Cross- Condition 67
4.5.3 Experiment 3: Cross- Source 68
4.6 Conclusion 69

5 EVALUATION AND DISCUSSION 71
5.1 Introduction 71
5.2 Evaluation of Results 72
5.3 Research Hypothesis 73

5.4 Discussion on Strengths and Weakness of Research 75
5.5 Conclusion 76

6 CONCLUSION **77**
6.1 Introduction 77
6.2 Research Overview 78
6.3 Problem Definition 79
6.4 Design/Experimentation, Evaluation and Results 79
6.5 Contributions and Impact 80
6.6 Future Work and Recommendations 81
6.7 Conclusion 82

References **84**

A Snippets of Code **92**
A.1 Data Pre-processing 92
A.2 Model Architecture 95

B Valence and Arousal Lexicons **96**

List of Figures

1.1 TDSP Life cycle showing all the phases 5

2.1 CRISP-DM Life cycle 14
2.2 KDD Process 16
2.3 Data Science: A Venn Diagram developed by Drew Conway showing how Machine Learning is related to Data Science 18
2.4 The Three Machine Learning Models 19
2.5 Factors to Classify Regression 21
2.6 Example of Logistic Regression Curve 22
2.7 Typical Neural Network architecture showing the different types of nodes . 23
2.8 Convolutional Neural Network Design shows Convolutional and Pooling layers 25
2.9 LSTM Architecture with the three Gates 27
2.10 Relationship between Text Mining and Data Science 29
2.11 Different Approaches and Models used for Sentiment Analysis 32
2.12 Sentiment Dimensionality: Valence and Arousal coordinates 34
2.13 Word Vectorization Types 35
(a) word2Vec 35
(b) GloVe 35
2.14 Model Capacity vs Model Fitting 37
2.15 Splitting of Datasets into Train, Validation and Test sets 38
2.16 k- fold Cross Validation Scheme 39
2.17 Leave one out Cross Validation (k=n) 40

2.18 Evaluation Metrics vs Types of Models 41
2.19 Graphs of various correlation coefficient R values 43
2.20 Interpretation of Kappa values 44

3.1 TDSP Level Design Flow prepared for this study 49
3.2 Detailed Design Flow of the Modelling phase that shows the three steps . . 54

4.1 Sample training data 58
4.2 Word Count of Reviews 59
(a) Summary 59
(b) BoxPlot 59
4.3 Rating distribution 59
4.4 Word Cloud of conditions in training data 60
4.5 Word Embedding done using GloVe on Review split into five Regions . . . 63
(a) Before Embedding 63
(b) After Embedding 63
4.6 First designed Model Architecture with single CNN and LSTM units 63
4.7 The Final Model Definition: Regional CNN-LSTM 65
(a) Final Architecture to Implement the Model 65
(b) Model Summary 65
4.8 Experiment 1- Configuration 66
4.9 Experiment 2- Configuration 67
4.10 Experiment 3- Configuration 68

5.1 Bar Graph Comparing the kappa values: In-Domain and Cross-Source . . . 74

A.1 Data Cleaning code 92
A.2 Replacing drug names and quantity with generic words 93
A.3 Spell correction and NER 93
A.4 Abbreviation Expansion 94
A.5 Stemming and Lemmatization 94
A.6 Definition of model in Python 95

B.1 Example of Valence Arousal Lexicons of English words 96

List of Tables

2.1 A Comparison of works done on Drug Reviews 12

2.2 TDSP, CRISP-DM,SEMMA and KDD: A Comparison 16

3.1 Drugs.com: Relevant fields and types 51

3.2 Drugslib.com: Relevant fields and types 51

3.3 The Three Experiments 55

4.1 Experiment 1- Results 66

4.2 Experiment 2- Results 68

4.3 Experiment 3- Results 69

5.1 Evaluation of the Experiments 72

5.2 Comparing this Study with Existing Standard 73

List of Acronyms

CNN	Convolutional Neural Network
RNN	Recurring Neural Network
LSTM	Long Short-Term Memory
TDSP	Team Data Science Process
CRISP-DM	Cross-Industry Standard Process for Data Mining
AI	Artificial Intelligence
ReLU	Rectified Linear Unit
KDD	Knowledge Discovery in Databases
SEMMA	Sample, Explore, Modify, Model and Assess
NLP	Natural Language Processing
NER	Named Entity Recognition
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
TF/IDF	Term Frequency/ Inverse Document Frequency
NLTK	Natural Language ToolKit
VA	Valence-Arousal pair
LOOCV	Leave One Out Cross Validation
ADR	Adverse Drug reaction
UCI	University of California, Irvine

Chapter 1

INTRODUCTION

1.1 Background

Public opinion on products was mostly neglected in a pre-internet era. But, now, each comment on a product has the capability to affect its sales. Hence, it is important for manufacturers to analyze their product reviews. Customers provide reviews in two forms, comments and ratings. Ratings can be easily classified into positive or negative. But comments are written by the user and depends on how they feel about the product in that particular moment. Ratings also depend on how they feel, but one person's perception of any n-point rating scale is different from another person. Reviews cannot be directly classified into positive or negative and hence, we need to do sentiment extraction. Human eyes can interpret one or two or even a few reviews but when it comes to thousands, it is difficult and expensive to employee the required workforce. Therefore, we need to implement machine learning applications along with sentiment analysis (Al-Moslmi, Omar, Abdullah, & Albared, 2017; Denecke, 2015).

An important field where review classification is necessary is in pharmaceutical drug evaluation. Patients' opinion on each drug must be carefully evaluated to decide a desirability factor and also to study on the side effects.

1.2 Research Problem

As mentioned in the background section, comment review analysis is a very hard job for the sales department of the pharmaceutical companies. Often, companies stick to the ratings alone to find out the most successful drug they sell due to lack of review quantification techniques. Patients give the rating as a rough value compared to their actual thoughts which are entered in the comments. Patient A might give rating 5 for average satisfaction, while patient B might give 9. These will cause errors in the study done by the company and can cause the inflation of drugs in the market unfavourable to the patients and can also discredit the company. Imagine an employee listening to the patients' comments and giving the rating on behalf of them, such that the ratings given will be of uniform scale. But, in real life, it cannot be achieved with one person since the volume of drug reviews dealt by a company is huge. Hence, we need an AI system that converts patients' remarks into ratings.

Typical pharmaceutical companies also conduct control groups where they give the drug to a closed group of patients, to gather consumer opinion (Kallumadi, Gräßer, Malberg, & Zaunseder, 2018). The output of such groups is also in the form of comments. People often fill out review forms which are structured, and can be interpreted easily but, usually, the last field in all forms asks for any extra input from the reviewer. This is the most important field and the entries are not structured, rather sentences written in the reviewer's style.

This proves that a machine learning system is necessary to go through the reviews to give each drug a proper rating or sentiment ranking in multiple stages of a drug promotion cycle. Text mining is the aptest for this and is discussed in detail in chapter 2.

To effectively understand the goal of any research it is necessary to answer the research question. The research question for this study can be framed as **"Can Text quantification techniques, particularly CNN-LSTM models and natural language processing (NLP), be used for drug review regression into a 10-point scale rating and can such models be migrated across different health conditions and different drug review data sources?"**

1.3 Research Objectives

The principal goal of this study is to understand the capability of CNN and LSTM models in the quantification of drug review text data. These models were proven to have higher efficiency (Wang, Yu, Lai, & Zhang, 2016) than other machine learning models.

This particular study has three primary objectives, wherein the first one deals with sentiment extraction and the other two validates transferability of models and are as follows:

1. Implement the **Regional CNN and LSTM models** to extract drug rating or valence-arousal pair by mining the textual review data. Then the results are evaluated.

2. Implement the same models as in the above case **across two or more disease conditions** that is, the models will be trained in drug reviews of one disease condition and will be tested in others.

3. Again, implement the same models **across two data sets**. They will be trained in Drugs.com data and will be tested in Drugslib.com data.

These objectives can be summarized in the form of an alternate hypothesis: The efficiency of information extracted from reviews can be improved if text quantification, using deep learning techniques such as CNN-LSTM models and Natural Language Processing (NLP) are used as compared to machine learning methods.

The study is done to find experiments that can validate the above hypothesis and reject the null hypothesis.

The above objectives will be satisfied by following the below steps:

- Extensive research into the existing literature for text quantification and sentiment extraction and discovering the best techniques.
- Extract, process and prepare the drug reviews data for analysis.
- Designing experiments to test the three cases.
- Record the shortfalls within the experiments.
- Evaluate the results of the experiments to conclude whether the alternate hypothesis is satisfied or not.
- Prepare notes on a future extension to the study and possibilities based on the listed shortfalls.

1.4 Research Methodologies

The plan is to conduct experiments on drug review dataset by developing deep neural network models to quantify (from 1 to 10 rating or VA sentiment pair) the comments on drugs given by the users. The study proposes to improve the efficiency of the state-of-the-art methods using deep learning methodologies and therefore, the type of research is secondary. The null hypothesis will be rejected if the obtained efficiency is more than the previous efficiency or else the null hypothesis will be accepted. So, this research employs inductive reasoning. The work adds on to previous such studies and hence should be considered to have a constructive form.

To conduct the experiment, two publicly available drug review data sets need to be extracted, processed, cleaned and transformed. These processes are done using python language packages in a Jupyter notebook environment. The data sets have about ten fields but only the reviews and ratings are considered which are text and numeric types respectively. Again, python language will be further used for model development, training, sentiment extraction and rating prediction.

The research process followed will be the same as in the industrial machine learning standards such as the TDSP. The Team Data Science Process (TDSP)[1] is a definitive life-cycle and is an improvement on CRISP-DM and KDD life-cycles. It has five levels of process, namely, business understanding or customer requirement understanding, data acquisition and data understanding, modelling, deployment and customer acceptance or satisfaction as seen in figure 1.1 on page 5.

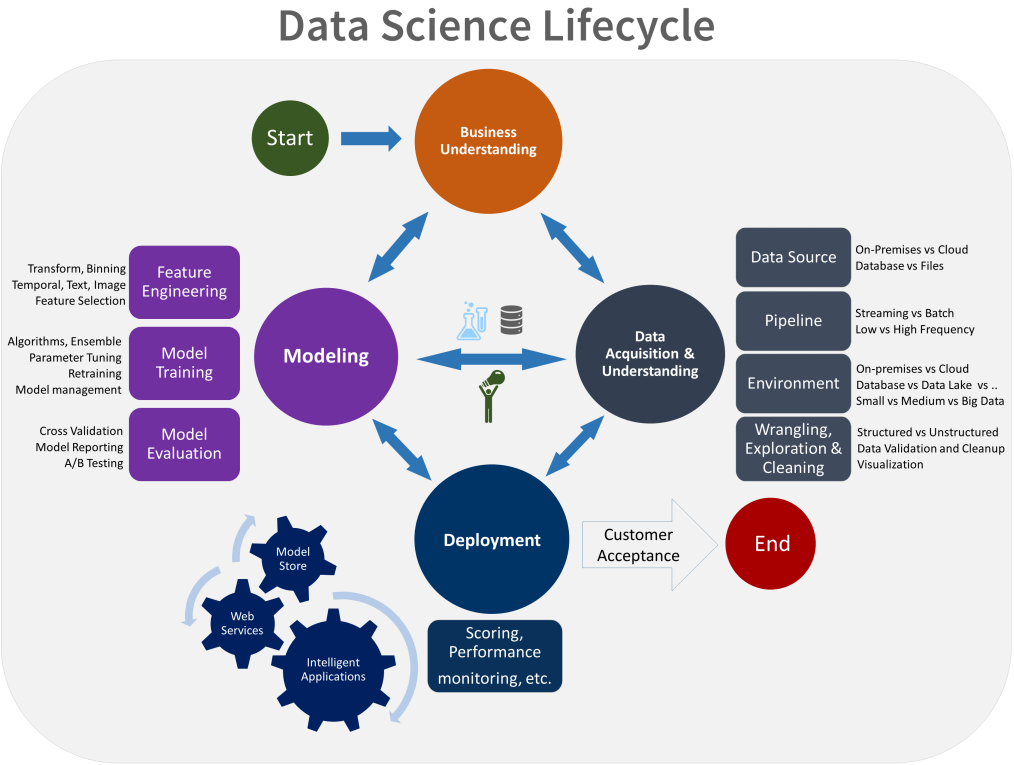


Figure 1.1: TDSP Life cycle showing all the phases

(Source: Microsoft Azure)

1.5 Scope and Limitations

Efficient analysis of patients' opinions about the drugs and trying to maintain a positive level will help pharmaceutical companies to sustain in the market. Precise identification of the pros and cons of each drug by understanding users' feelings can also help to improve the

[1] https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle

financial stance of the company.

The scope of this proposed work is the efficient text quantification into rating or sentiment, of drug reviews using CNN-LSTM models and also checking the migration capabilities of these models across different sources and condition. The accuracy of these experiments will be compared to existing works to determine if the research is successful. The data required for this work will be extracted from Drugs[2] and Drugslib[3] drug review websites.

The major limitations that will be faced during the design, implementation, validation and evaluation of this research work are:

- The data set cannot be considered as a huge one in terms of the number of reviews per drug. Therefore the drug-wise analysis is difficult.
- Pre-processing of the data should be careful and is not same as in the case of text summarizing since the removal of redundant and short words might affect the sentiment.
- The number of reviews for a particular drug is not balanced and hence cannot consider drug name as a factor in machine learning. Therefore cannot be considered to be a real-world study. In the real world, the data sets should be balanced.
- Not all the disease conditions are given balanced importance in the data set and hence only a few can be considered for cross- condition analysis. The number of reviews per disease conditions are also not balanced.
- CNN and LSTM models are used to extract valence and arousal values of sentiment, but the sentiment is not essentially a bi-directional entity. It can have more than two components.

1.6 Thesis Outline

The remaining sections of this dissertation work are organized as given below:

[2]https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29

[3]https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Druglib.com%29

- **Chapter 2- Literature Review and Related Work** discusses the existing works in sentiment extraction, text mining, natural language processing, named entity recognition, neural networks in general, CNN and LSTM in specific, regression methods and machine learning and data mining as a whole.

- **Chapter 3- Experiment Design and Methodology** deals with the design of the experiments that are to be conducted to get an answer for the research question and also the data sets used will be defined in detail in this section. The machine learning models will be designed and described along with result measurement techniques.

- **Chapter 4- Implementation and Results** has the actual implementation of the experiments and the outputs are compared with parameters to get the efficiency metrics.

- **Chapter 5- Evaluation and Discussion**Evaluation is done by comparing the outputs with the expected answer of the research question and the difference between them are discussed on.

- **Chapter 5- Conclusion** section summarizes the thesis work with brief descriptions of all the above chapters. In addition, the impact of the work on the field of science and business of pharmaceuticals are also discussed along with considerations and suggestions on possible future extensions to this work.

Chapter 2

LITERATURE REVIEW AND RELATED WORK

2.1 Introduction

This chapter defines all the algorithms and techniques that are used to design, implement and evaluate the experiments in this study. All the relevant techniques and algorithms are introduced, which are needed to convert drug review into sentiment dimensions and then again into ratings and to evaluate the results which are used in the coming chapters. There are eight main sections in this chapter out of which the last seven are related to technologies and the first one deals with existing literature on the domain that is drug reviews.

In the first section 2.2, existing standard works in drug review classification into ratings are discussed. Also, it is described how each technique used in this research are selected based on the previous works. Approaches to solving the research problem that is already tried are discussed along with the gaps in these studies. Finally, the base plot for this research is set forth by clarifying all the said gaps.

Section 2.3 deals with various data mining life cycles which are considered as standard in the industry like TDSP, CRISP-DM and KDD. These are compared with each other. In this research TDSP is used.

In section 2.4, the terms data mining and machine learning are compared with each other. Also, various types of data mining algorithms are briefly explained like supervised, unsupervised and reinforcement learning.

Section 2.5 discusses linear and logistic regression algorithms which are examples of supervised machine learning techniques.

In section 2.6, Neural networks are explained which are deep learning techniques and are used in this research. Particular types of neural networks namely CNN and LSTM are also discussed.

Section 2.7 deals with the definition and relevance of Text mining. The main feature in this research is a collection of drug reviews which is in textual form. It also goes through the technologies NLP and NER, and also briefly introduces sentiment extraction.

Model validation techniques are described in section 2.8. This includes techniques for proper fitting of the models, standards in data set splitting and cross-validation methods.

Section 2.9 discusses model evaluation techniques such as RMSE, MAE and Pearson's correlation coefficient. This section briefly differentiates validation techniques used in different types of machine learning models.

2.2 Literature and Existing Works on Text Quantification

Adverse drug reactions (ADR) or side effects of the drug are the main reasons for conducting studies on them and analyzing patients' opinions, which requires these opinions to be converted to numeric form. Some side effects might take months to surface and hence there should be a very long review process, probably as long as the drug is in the market. Lexicon

based systems (Goeuriot et al., 2012; Leaman et al., 2010) are more popular in this field where the reviews are compared to a pre-collected dictionary of words to understand which opinions are projected strongly, which is also called opinion mining (Liu, 2012). All the relevant techniques to extract side effects and reactions are documented in detail by Sarker et al. (2015). Korkontzelos et al. (2016) theorized that when a patient mentions a side effect then it is a negative comment, and thus tried to incorporate sentiment analysis to the lexicon system. The idea proposed by Cavalcanti and Prudêncio (2017) derives different features of the side effects from the reviews using what they call as syntactic dependency paths. It is more of a knowledge-based approach where each word in the review is tagged with a known keyword.

Nikfarjam and Gonzalez (2011); Nikfarjam, Sarker, O'Connor, Ginn, and Gonzalez (2015) looked for strong bigrams and trigrams in the comments that could potentially form a pattern. One among the first sentiment analysis works done on drug reviews (Xia, Gentile, Munro, & Iria, 2009) developed a classification system where sentiment classifiers were applied to each condition. Supervised deep learning models were used by Mishra, Malviya, and Aggarwal (2015); Gopalakrishnan and Ramaswamy (2017) to do sentiment analysis on the drug reviews. Kallumadi et al. (2018) proposed a cross-domain and cross-data sentiment analysis, where machine learning models were generated with reviews on different medical conditions as training and test data and then the same was done across reviews from two different websites. They got an overall Cohen's kappa of 83.99 for in-domain sentiment analysis and could be considered as the best existing approach.

2.2.1 Approaches to Solving Problem

Approaches to analyzing drug reviews principally may be of two forms, lexicon-based rating or sentiment-based classification. Information extraction from the comments and classifying them based on the sentiment is the more relevant approach (Gräßer et al., 2017; Liu et al., 2016). Text mining (Voorhees, 2002) is similar to data mining but has also many differences. Many of the classical data mining tools like decision trees and random forests are not even relevant when considering textual data.

CNN models (Glorot, Bordes, & Bengio, 2011) can be created to extract sentiment or rating from the reviews. These trained models are applied to derive ratings based on review sentiment from the test data. Hence it can be found out how sentiments affect the overall rating of a drug.

Cross-data sentiment analysis (Kallumadi et al., 2018; Al-Moslmi et al., 2017; Bollegala, Mu, & Goulermas, 2016) is a successful method to understand how strong the machine learning models work in the pharmaceutical field, by testing their ability to migrate.

2.2.2 Gaps in Research

The popular lexicon-based systems (Goeuriot et al., 2012; Na & Kyaing, 2015) are inefficient since they largely depend on the existence of these words in the defined vocabulary. User entries are random and unpredictable with no technical basis and may not always adhere to the phonetic rules of the stored language. Lexicon systems have no strategy to deal with misspelt words.

Sentiment analysis for classification of the reviews is a strong approach if a large data set, in terms of hundreds of thousands of reviews, is available and that is a condition very rarely satisfied. To tackle this Kallumadi et al. (2018) suggested cross-domain and cross-data sentiment analysis which was a successful update on the accuracy of the models. But even then, the accuracy levels attained were not satisfactory. The migration capacity of the models between conditions is highly proportional to the similarity between the conditions. Some cross-domain models were more accurate than in-domain models whereas some were less successful. Cross-data sentiment classification models also have drawbacks if strong machine learning models like neural networks are not used. Even with these shortfalls, the intra-model migrating text quantification scheme seems the aptest for classifying reviews and its accuracy can be manipulated by upgrading the models used.

In this proposed work, regional CNN and LSTM neural networks are planned to be used

in these cases: in-domain sentiment analysis, cross-condition and cross-source sentiment analysis to improve the efficiency as compared to the existing works. A hybrid architecture of neural networks will be needed to execute the work.

2.2.3 Existing Works on Drug Reviews

Research	Dataset Source	Model Used	Predicted Output
Na and Kyaing (2015)	WebMD.com	Clause-wise Lexicon approach and Classification using SVM	Sentiment Polarity
Kallumadi et al. (2018)	Drugs.com, Drugslib.com	Classification	Drug Rating
This Work	Drugs.com, Drugslib.com	Regression using Regional CNN-LSTM	Drug Rating/ Sentiment Dimensionality

Table 2.1: A Comparison of works done on Drug Reviews

Two main studies, done on drug reviews, were discovered as part of the research on the literature. They are compared with this work in table 2.1.

The first one, Na and Kyaing (2015) used a dataset extracted from the WebMD website. Their work had two approaches, finding sentiment polarity in the drug reviews using pre-populated sentiment lexicons and by using SVM (Support Vector Machine)model. Additionally, instead of using the entire drug review as such, they divided a review into clauses stating that each clause in the text could contain important information about attributes of the particular drug. This is a meaningful and relevant step and is adapted in this study as well, in a different form. But overall, their work cannot be considered as the standard due to the drawbacks of lexicon and sentiment classification approaches discussed in the above sub-section, 2.2.2. They used Cohen's kappa interrater agreement value to measure the per-

formance of their model.

The second paper by Kallumadi et al. (2018) used the drug review datasets taken from Drugs.com and Drugslib.com. Their principal approach was to convert the reviews into ratings and they used standard classification techniques. But they introduced two extra experiments called cross-source and cross-domain to test the migration capabilities of their model. They also used Cohen's kappa to evaluate their model.

The important features of both the above papers, that are found to be useful, are adopted into this study. The datasets used are the same as the second paper along with the two extra experiments. Cross-condition where train data is taken from one disease condition data and test from other disease conditions. Cross-source experiment where train and test data are from the two different sources. But, unlike their classification approach, regression is employed in this work taking into account that the drug ratings given by users do not necessarily be whole numbers. The clause-wise analysis approach in the first paper cannot be implemented easily into a neural network since it will need a different number of neural network units for each drug review of different size. Instead, a regional approach is planned to be implemented in this work where the drug reviews are divided into an equal number of regions and using dedicated neural network units for each region. The number of regions can be decided by looking at the average sentence size of the drug reviews. the model selected for this work is a regional CNN-LSTM hybrid neural network and the reason for this explained in the next section.

2.2.4 Why Regional CNN-LSTM?

At present there are various techniques used for categorical sentiment analysis like word embedding (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) and deep neural networks such as convolutional neural networks (CNN) (Kim, 2014; Kalchbrenner, Grefenstette, & Blunsom, 2014), recurrent neural networks (RNN) (İrsoy & Cardie, 2014) and long-short term memory (LSTM) (Wang, Liu, Sun, Wang, & Wang, 2015; Liu, Joty, & Meng, 2015). CNN is capable enough to extract local informa-

tion and do proper feature selection but will fail when it comes to long-distance dependency since it has no memory unit. Therefore an LSTM model is added to the CNN since it can sequentially model texts across sentences. These neural networks and word embedding techniques are not well explored for text quantification and dimensional sentiment analysis.

The regional split concept is added to this CNN-LSTM model such that when the input review is fed as regions, CNN will be able to identify important features from each region (Wang et al., 2016), which will add up to the final prediction quality.

2.3 TDSP and Data Mining Life-cycle

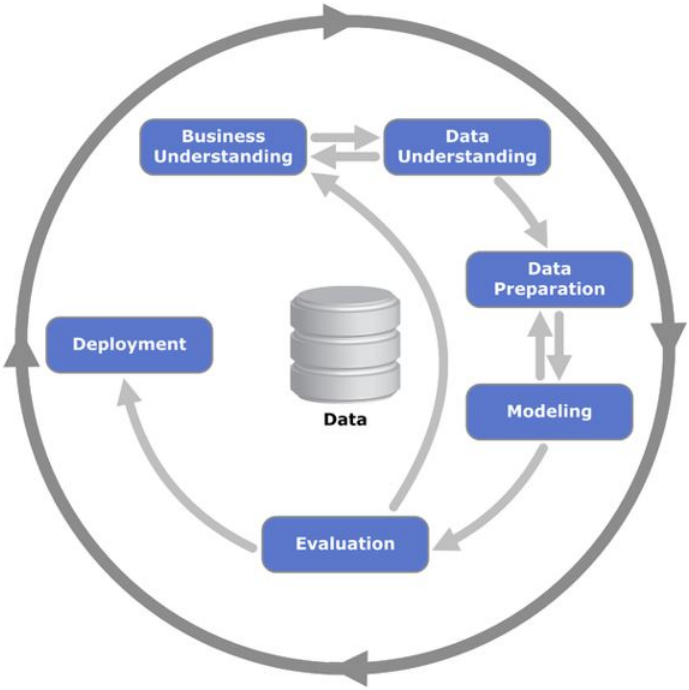


Figure 2.1: CRISP-DM Life cycle

(Source: Data Science)

The Team Data Science Process (TDSP), which can be seen in figure 1.1, is developed by the Microsoft Azure team to guide industrial machine learning tasks. As already mentioned

in chapter 1, it is an advancement on other life cycles like CRISP-DM and KDD process. In fact, TDSP can be considered as a combination of CRISP-DM and SCRUM [1]. Microsoft calls it "an agile, iterative data science methodology to provide predictive analytics and intelligent applications [2]". It is designed to improve teamwork and shared learning and contains the best inputs from Microsoft as well as others from the industry. TDSP is sometimes mentioned as the most evolved of the CRISP processes. There are five main levels in TDSP which are listed in chapter 1.

Cross-Industry Standard Process for Data Mining (CRISP-DM)(Shearer, 2000) is the most common of the data mining life-cycles ever used. It is again an iterative structure with six levels where the real-time inferences from one level can affect previous or future levels. The six levels in CRISP-DM, as seen in figure 2.1, are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. They might seem sequential but are not always. CRISP-DM is developed by Daimler Chrysler in association with SPSS and NCR.

SAS Institute developed SEMMA (Azevedo & Santos, 2008) which stands for Sample, Explore, Modify, Model and Assess. There are five stages in SEMMA as the name suggests. Even though it is an easy, direct to understand model, it lacks support given to the business field. SEMMA can never be considered as business-oriented.

Knowledge Discovery in Databases (KDD) is a traditional step by step process used as a base for discovering patterns or behaviours within collections of data. Knowledge is defined as useful information (Fayyad, Shapiro, & Smyth, 1996). Even though data mining and KDD are terms used synonymous to each other, data mining is also the name of a phase within KDD. This confusion between the two terms was clarified by the creators of the KDD process by stating that knowledge should be the end product of any data mining or pattern-driven experiments. There are five phases in the KDD process, as seen in figure 2.2, called

[1] http://www.datascience-pm.com/tdsp/

[2] https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle

selection, pre-processing, transformation, data mining and interpretation/evaluation.

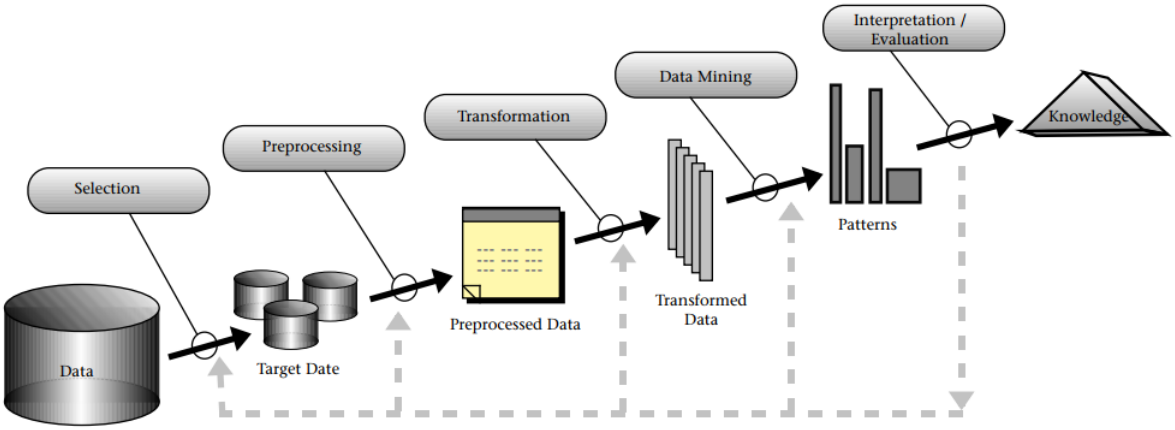


Figure 2.2: KDD Process

(Source: Fayyad et al. (1996))

TDSP	CRISP-DM	SEMMA	KDD
Business Understanding	Business Understanding	-	-
Data Acquisition and Understanding	Data Understanding	Sample	Selection
		Explore	Pre-processing
	Data preparation	Modify	Transformation
Modelling	Modelling	Model	Data mining
	Evaluation	Assess	Interpretation/Evaluation
Deployment	Deployment	-	-
Customer Acceptance	-	-	-

Table 2.2: TDSP, CRISP-DM,SEMMA and KDD: A Comparison

Table 2.2 compares and differentiates all these life cycles. As discussed in this section and as evident from table 2.2, TDSP is the most advanced of the life cycles developed with a customer-centred vision. Therefore TDSP is selected for this research work.

2.4 Machine Learning: Definition and Algorithms

This section looks into the relationship between machine learning and data mining, both of which are sub-fields of data science. Also, different machine learning algorithms are discussed in detail.

2.4.1 Data Mining and Machine Learning

Data mining is an essential part of all of the life cycles discussed in section 2.2. Data mining[3] is the process of digging deep into huge amounts of data to discover previously unknown relationships, features and patterns which are useful for the future.

People often consider machine learning to be synonymous to data mining but technically they are different, even though they share the common core of being data science applications. Machine learning[4] learns from an existing collection of data or learns how the data behaves and teaches the machine how to respond to each behaviour of the data in future.

Essentially, data mining uses machine learning algorithms and machine learning depends on mined information. Data mining extracts useful information from patterns while machine learning uses that information for future iterations. Conventionally data mining requires a third party intervention after each stage but machine learning improves itself with each level of learning. An example would be, e-commerce shops can use data mining to understand what usually goes together into a customers bag and thus give recommendations, but if a machine learning algorithm is implemented, it will go deeper into customers' buying habits and enhance its own recommendation capability[3]. Data mining is good at detecting fraud and machine learning teaches itself to prevent frauds in the financial segment. Figure 2.3 shows a Venn diagram of how data mining and machine learning are derivatives of data science. In spite of these differences data mining and machine learning often overlaps as domains and is used interchangeably in this study.

[3]https://www.netguru.com/blog/machine-learning-vs-data-mining-what-is-the-difference

[4]https://www.import.io/post/data-mining-machine-learning-difference/

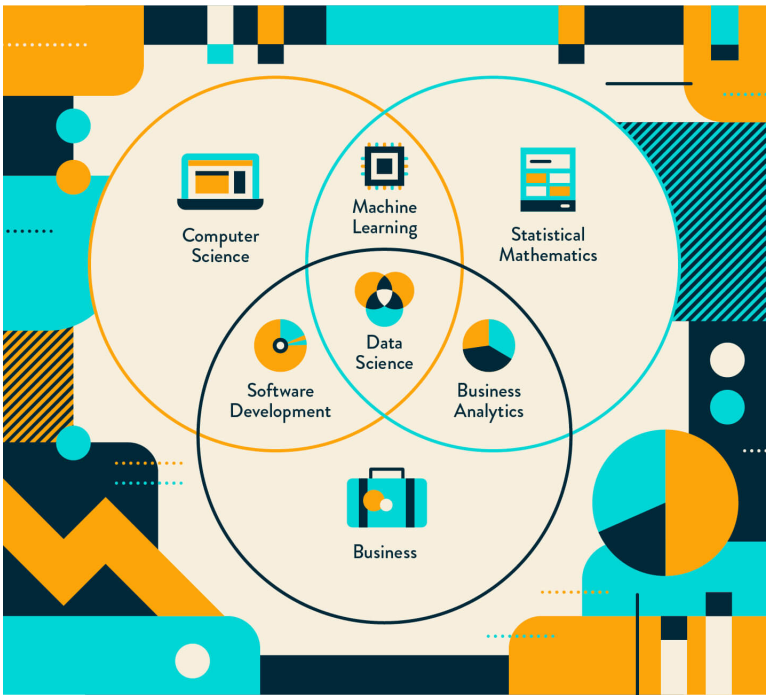


Figure 2.3: Data Science: A Venn Diagram developed by Drew Conway showing how Machine Learning is related to Data Science

(Source: Drew Conway)

Machine learning has applications in many industries and domains. For example (Hastie, Tibshirani, & Friedman, 2001):

- In the health field to predict the occurrence of a condition in a patient from the patient's medical history.
- Stock market predictions of startup companies done by cloud financiers to know safety in investments.
- In converting student records to soft copy by scanning the record documents and identifying the student details from the text in the image.

2.4.2 Types of Machine Learning Algorithms

There are three main types of machine learning algorithms, supervised, unsupervised and reinforcement learning (Ayodele, 2010) as depicted in figure 2.4. taken from IBM Devel-

oper article on machine learning[5].

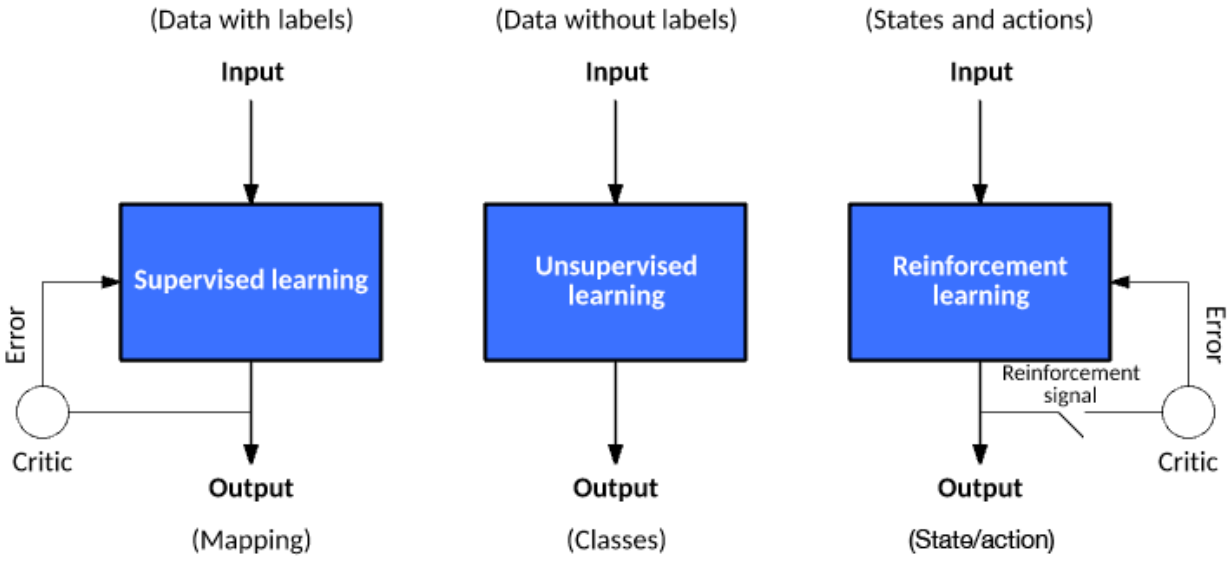


Figure 2.4: The Three Machine Learning Models

(Source: IBM Developer)

In supervised learning both input and output are available. Outputs are also called labelled data or dependent data while inputs are independent data. The algorithm develops a relationship between the input and output, such that new outputs can be predicted or determined from fresh inputs. This is further divided into classification and regression. Classification algorithms groups input into pre-defined classes of outputs while regression algorithms predict distinct output value for each input value. Regression is discussed in detail in the next section of this chapter.

Unsupervised learning has a collection of inputs alone and lacks any labelled output data. The algorithm is exploratory and strives to find new information from the input data like patterns and relationships across them. Clustering is a type of unsupervised learning where the inputs are classified into groups based on their common behaviours and when a fresh input is introduced a class will be determined based on how similar it behaves to others in

[5]https://developer.ibm.com/articles/cc-models-machine-learning/

the class.

Reinforcement learning is an iterative self-learning algorithm. There are inputs called initial states and several outputs or possible solutions. The model starts training at the initial state and will return a solution and will be either rewarded or punished for the solution. In the next iteration, the model will develop a new solution based on whether the reinforcement it got was a reward or a punishment. After all the solutions are presented by the model the best one will be selected which is the solution with the maximum rewards or in other words the least penalised. There are two types of reinforcement[6] learning- positive reinforcement is when an event occurs due to behaviour, then the strength and frequency of that behaviour is boosted and negative reinforcement is when an event is avoided by behaviour and is boosted. Both have their own advantages and disadvantages.

2.5 Linear and Logistic Regression

Regression is a type of supervised machine learning technique. It investigates and establishes relationships between a target[7] (dependent variable) and predictor or predictors (independent variables). Finding causal effect relationships, time series modelling and forecasting are some of the major applications of regression. A curve or a line is discovered such that the distance between the entities and the curve is minimized. The two main features of regression are that it proves the existence of relationships between dependent and independent variables and it also determines how strongly each independent variable affects the dependent variable. It is also not scale-sensitive, in other words, dependent and independent variables need not be of the same scale. There are mainly seven types of regression based on three classification factors number of independent variables, the shape of the regression line and type of dependent variable as seen in figure 2.5. Further, more regression types can be invented by combining these factors. The seven regression types are logistic regression, linear regression, polynomial, stepwise regressions, ridge regression, lasso regression

[6]https://www.geeksforgeeks.org/what-is-reinforcement-learning/

[7]https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

and elasticNet regression. The most commonly used ones are linear regression and logistic regression.

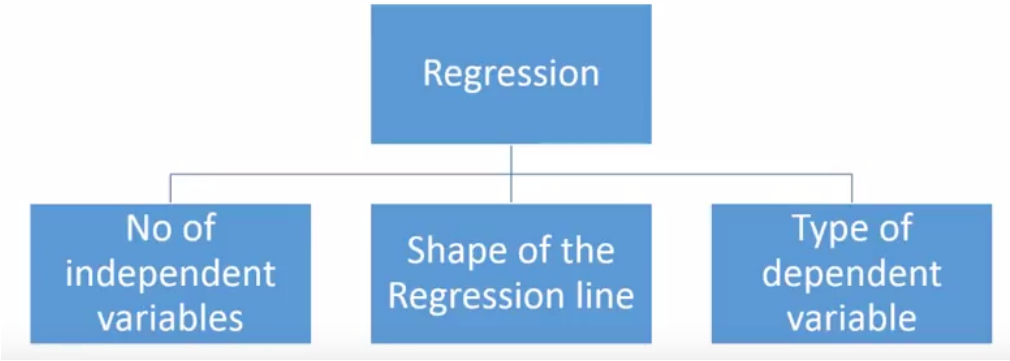


Figure 2.5: Factors to Classify Regression

(Source: Analytics Vidhya)

In linear regression[7], the nature of the regression curve is linear, the dependent variable is continuous and the predictor variables can be discrete or continuous. It discovers a straight line between the target and predictor variables. Linear regression can be described with the below equation where Y is the target variable, a is the intercept of the line on the predictor variable axes, b is the slope of the line and e is the error term that carries the difference of each data point with the line.

$$Y = a + b * X + e \tag{2.1}$$

Linear regression is further classified into simple linear regression and multiple linear regression based on the number of independent variables. But multiple linear regression is often affected by multicollinearity, heteroskedasticity and auto-correlation which can be corrected by selecting the most relevant independent variables using algorithms such as forward selection, backward elimination and step-wise approach. Also, the regression line in any linear regression is susceptible to outliers.

Logistic regression is used when the target variable is binary of the form of (0,1), (TRUE, FALSE), (ON, OFF) etc. It is also defined as the probability of an event both if it is a success

and if it is not. The below equations[7] represent how logistic regression works:

$$odds = p/(1-p) = probabilityofeventoccurrence/probabilityofnoteventoccurrence \tag{2.2}$$

$$ln(odds) = ln(p/(1-p)) \tag{2.3}$$

$$logit(p) = ln(p/(1-p)) = b0 + b1X1 + b2X2 + b3X3.... + bkXk \tag{2.4}$$

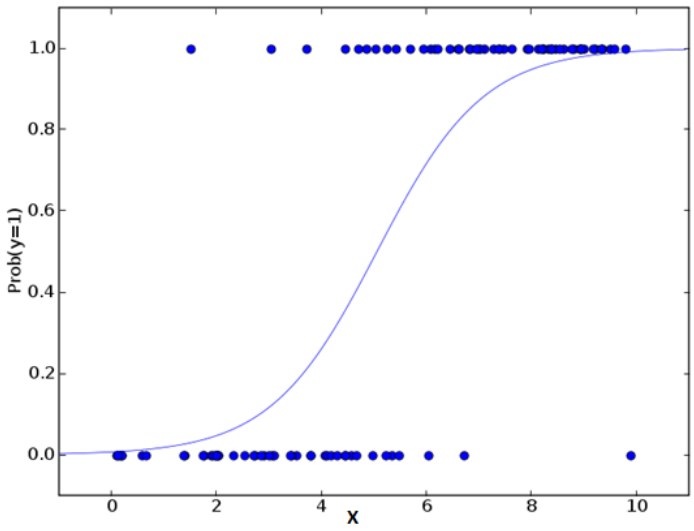


Figure 2.6: Example of Logistic Regression Curve

(Source: Analytics Vidhya)

Figure 2.6 shows an example of a logistic regression curve. Here the goal is maximising the occurrence of data points instead of limiting the error squares, which was the goal in linear regression. Logistic regression is applied mainly in classification requirements. Since a non-linear log equation is used, linearity between target and predictor variables is not necessary. Logistic regression performs better if the data size is large. Further divided into ordinal and multinomial logical regressions based on whether the target variable is ordinal or multi-class.

2.6 Neural Networks

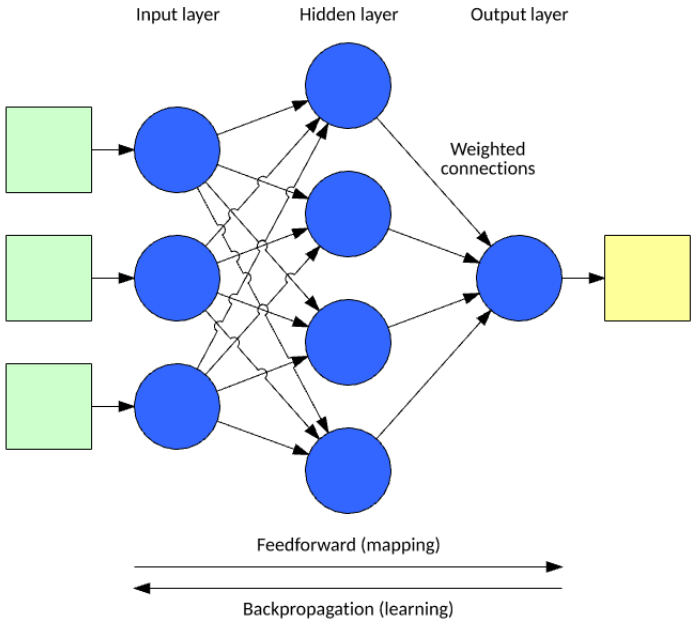


Figure 2.7: Typical Neural Network architecture showing the different types of nodes

(Source: IBM Developer)

Neural networks[8] are advanced machine learning models that have sequential connections from the input layer to the output layer and sometimes vice versa. Between these layers, there are usually an array or arrays of nodes which is inspired by the design of neurons in the human brain. The connections between each such neuron are assigned a weight value such that certain components of the input vector are given more importance in the model training process. Each neuron does a mathematical computation on the weighted inputs, which is called its activation function. The major activation functions ever tried are a step, sigmoid, tanH, and ReLU. The output from a neuron is calculated by applying the activation function on both the input vector and the weight factor of that vector. The total output of the system is generated by feeding the input to the input layer and then calculating outputs of each neuron in between until the output layer. This style of calculation is referred to as feed-forward fashion[8]. Neural networks learn from examples and are not programmed. They are

[8]https://developer.ibm.com/articles/cc-cognitive-neural-networks-deep-dive/

so powerful that they were able to discover patterns in multimedia and images which are not structured data. Neural networks evolved from single layer perceptrons[8] through multi-layer perceptrons, through backpropagation to LSTM(Long Short-Term Memory) which finds patterns across temporal data.

Backpropagation is a popular algorithm used in neural networks where the total losses or errors in the output of the system is propagated back till the input layer and the weight factors of each connection between the nodes are updated to reduce that loss. This is a supervised learning technique. The error at the output is calculated as the difference between actual and desired output. The correction of weights start at the output layer and then proceeds towards the input layer and hence called 'back' propagation.

Deep learning[9] is a prominent field associated with machine learning which makes use of the neural networks to do computational learning. There are different kinds of artificial neural networks and each has its own defined logic in finding the pattern. The different types of artificial neural networks[9] are listed below. Out of them, the two forms relevant to this study- CNN and RNN, are described in detail.

- Feedforward Neural Network – Artificial Neuron.
- Radial Basis Function Neural Network.
- Multilayer Perceptron.
- Convolutional Neural Network(CNN).
- Recurrent Neural Network(RNN) – Long Short Term Memory.
- Modular Neural Network.
- Sequence-To-Sequence Models.

[9]https://www.digitalvidya.com/blog/types-of-neural-networks/

2.6.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks[10] play a major role in the field of image and video processing. The types of hidden layers in CNN (Yamashita, Nishio, Do, & Togashi, 2018) are convolutional layers, pooling layers, fully connected layers and normalization layers. In other words, in CNN the activation functions are convolution and pooling.

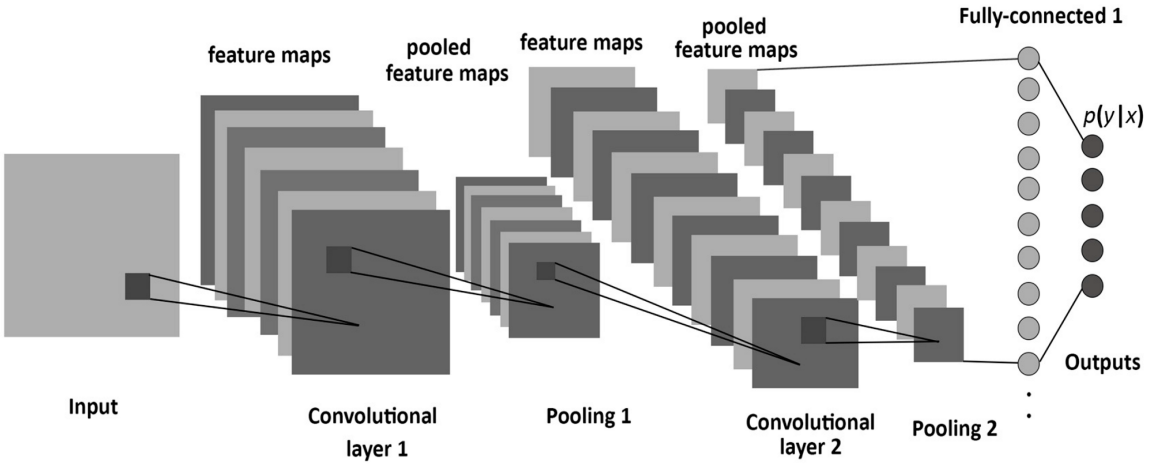


Figure 2.8: Convolutional Neural Network Design shows Convolutional and Pooling layers

(Source: Towards Data Science)

Convolution (Indolia, KumarGoswami, Mishra, & Asopa, 2018) has two inputs and a single output. There are one dimensional and two-dimensional convolutions, wherein signals are used as input for first and images as inputs in the second case. Basically, a convolution node uses the second input, called the kernel[10], as a filter on the first input to get the output. The mathematical form of a convolution function with two input signals x and y is given below, which is same as the dot product of the two inputs.

$$(x * y)(i) = \sum_{j=1}^{m} y(j).x(i - j + m/2) \tag{2.5}$$

[10]https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90

Pooling is a process of down sampling[10] the input, for example, an image or the matrix output of a hidden layer. The dimensionality of the input will be reduced and assumptions will be made about the removed components. There are two types of pooling techniques, max and min pooling. Max pooling considers the maximum value and leaves behind others while min pooling considers the minimum values.
CNN is used in this study to extract the sentiment dimensions from the drug reviews.

2.6.2 Recurring Neural networks and LSTM

Basic neural networks cannot remember past events or their outputs in the past to judge a new event. This is a major shortfall when it comes to advanced deep learning requirements. Recurring Neural Networks are modified algorithms that can preserve memory. RNNs have loops in them such that the data flows through them after each node running. RNNs can be considered as sequential layers of many neural networks in which each neural network will pass its output state to its successor. RNNs are used in a variety of fields such as speech recognition, language modelling, translation, image captioning[11] etc. The memory retained in an RNN neutron is called its internal state, which is used to transform the input signals unlike in typical feed-forward neural networks. The below equations[12] show the current state of an RNN node (eq. 2.6), application of activation function tanh (eq. 2.7) and the output state (eq. 2.8). In the equations W represents weight, W_{hh} indicates weight of previous hidden state, W_{xh} indicates weight of current state and y_t is the value of output state.

$$h_t = f(h_{t-1}, x_t) \tag{2.6}$$

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t) \tag{2.7}$$

$$y_t = W_{hy}h_t \tag{2.8}$$

[11] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[12] https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e

RNNs have many advantages like the input data can be considered as samples that are dependent on previous inputs and RNNs can have convolutional layers for rigorous image processing applications. But there are disadvantages as well like the gradient vanishing and exploding problems and incapability of RELU and tanh as activation functions for large sequences of data.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an advanced form of recurring neural networks which persists previous data more efficiently in memory. LSTM is also a solution for the gradient vanishing issue in RNN. LSTM deals with time-series data, same as RNNs but here the temporal distance or duration can be unknowingly large. Backpropagation algorithm is essential in LSTMs[12]. LSTMs make use of three gates- input, output and forget gates as shown in figure 2.9.

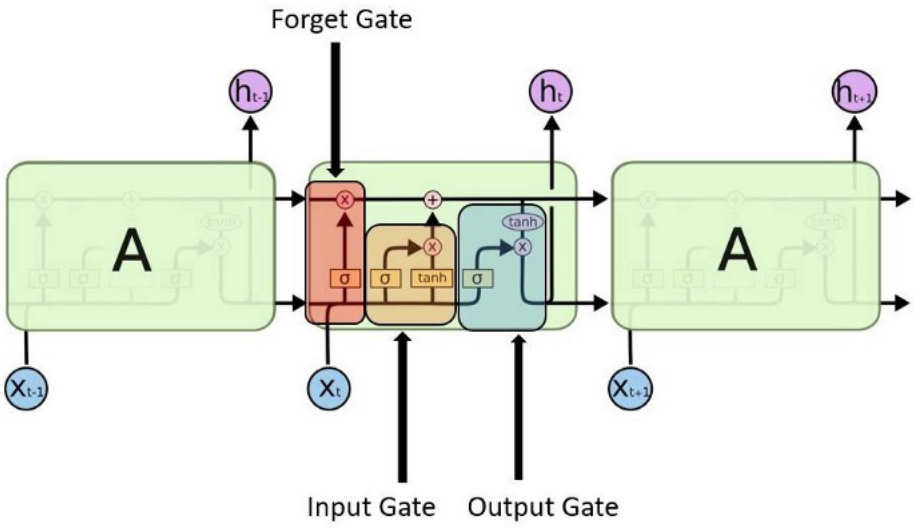


Figure 2.9: LSTM Architecture with the three Gates

(Source: Towards Data Science)

- Input Gate: Determines how the input will affect the memory. It has two parts. Sigmoid function acts like a (0,1) switch and decides which inputs are passed. The tanh

function is used to give further weight to the allowed inputs on a scale from -1 to +1.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2.9}$$

$$\widetilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{2.10}$$

- Forget gate: Again a sigmoid function decides which data to be kept and which to be removed from the unit. For each component in cell state C_{t-1}, a value is generated by forget gate between 0 and 1 based on previous state h_{t-1} and input x_t. If its 0 the value is omitted if it's 1 it is kept in the cell.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2.11}$$

- Output Gate: The input and cell memory together is used to get the output value. Similiar to input gate, output gate also has a sigmoid and a tanh function. Sigmoid selects values and tanh gives them weightage.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{2.12}$$

$$h_t = o_t * tanh(C_t) \tag{2.13}$$

In this work, LSTM is used to understand how one sentence in a review affects the sentiment of a later sentence in the same review.

2.7 Text Mining

Data is the soul in data mining and with the rapid growth in software and hardware technologies helped in creating different types of data in abundance. One such type of data is textual data and is accumulating due to web platforms and social media (Aggarwal & Zhai, 2012). Unlike structured data, textual data does not adhere to any structural framework and hence is difficult to maintain. Due to this reason, search engines are used to maintain text

data as compare to databases in structured data.

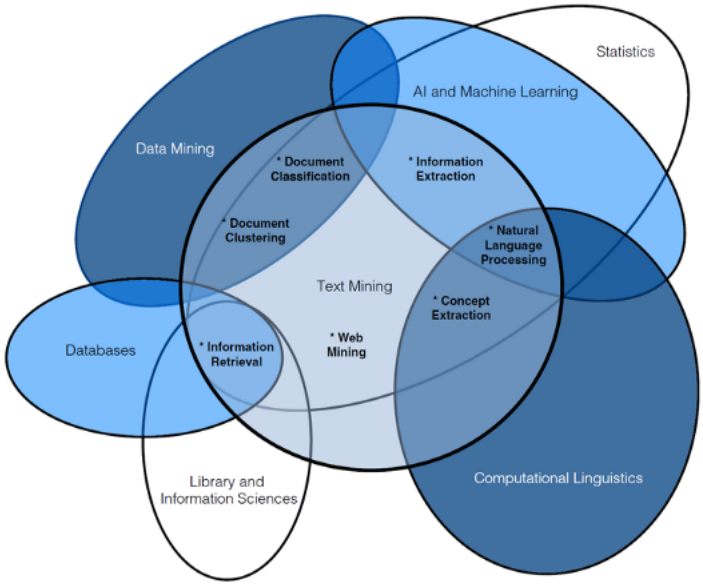


Figure 2.10: Relationship between Text Mining and Data Science

(Source: Miner, 2012)

Textual data are considered to have high dimensionality. Text data can have different types of representations like a bag of words or string of words but it is preferable to represent it semantically like using named entities. A named entity can be any defined unit of knowledgeable data like data about a person, a building, a game, an object etc. Different algorithms relating to text mining field are:

- Information extraction from Text.
- Text Summarization.
- Unsupervised Algorithms like Clustering and Topic Modeling.
- Dimensionality reduction.
- Supervised Algorithms like Automated Text Categorization.
- Transfer Learning.

- Probabilistic Text Models.
- Text Stream Mining.
- Cross-lingual Mining.
- Opinion and Sentiment Mining.

In general text, mining can be defined as the process of extraction of information, relations, facts, hidden messages from textual data and either representing them in a structured manner or quantizing them for mathematical calculations and predictions.

2.7.1 Natural Language Processing and Named Entity Recognition

Natural Language Processing (NLP)[13] is the most important among the methodologies in text mining. It is essentially the process of mimicking the human capability to understand and extract information from natural languages. Both natural language understanding and natural language generation are parts of natural language processing. NLP can generate clean and machine recognizable structured forms from input text or audio data for machine learning purposes. It stands as a bridge between human languages and data science[14]. NLP is the core for technological development in many areas, like:

- Prediction of diseases from patient records and their description of the symptoms.
- Sentiment analysis through social media and review sites by corporate companies.
- IBM developed a cognitive assistant that learns about the customer.
- Stopping spam messages in short text and mailing applications.
- Fake news detection.
- Intelligent personal assistants like Amazon's Alexa and Apple's Siri.
- Stock market and share market traders use NLP to predict rise and fall from news, tweets, chat rooms etc.

[13] https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing

[14] https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1

- Resume classification and other employment-related fields.
- In the legal field for litigation tasks.

Some among the basic algorithms in NLP is a bag of words, tokenization, stop words removal etc.

Bag of words is a method to get the count of all the words in a text. It creates a matrix of word-frequency components and from that matrix, the occurrence rate of each word can be determined. But considering word frequency alone has certain shortfalls including the high number of stop words and least occurrence of defining terms. To overcome this the frequency of words can be penalized by knowing their frequency in all the documents in the corpus. This approach is called Term Frequency/ Inverse Document Frequency (TF/IDF)[14]. The algorithm rewards frequent words in a document but penalizes those words if they are frequent in other documents as well.

Tokenization is the process of splitting text into basic units called tokens like words, sentences etc. Tokenization is easy in languages like English that uses white space for word separation and period for sentence separation but not in all the languages. Again, not all periods can be considered as the end of sentences and thus the tokenization process has shortcomings.

Removing the commonly used words in the language from the text is known as stop words removal. This can bring down processing requirement exponentially. They are removed by comparing with a pre-prepared list of stop words. There are many lists of stop words in different languages but there is no universally accepted list[14].

Affixes are extra additions to words indicate a change in the situation, verb to noun, time, etc. These affixes are often not necessary and should be considered as unwanted extensions of wanted words. Stemming is the process of slicing affixes from words. Both inflectional and derivational affixes must be removed if they do not give any value to the content.

Lemmatization is the process of converting words to their dictionary representation. It also considers identical words that are supposed to have different meanings.

Named Entity Recognition (NER)

Named Entity Recognition is also called entity extraction[15]. NER groups similarly-named entities into categories such as people, places, jobs, objects etc. NER provides a semantic structure for our text. The main python libraries used for NER are spaCy and NLTK, which are used in this study for processing the drug reviews.

2.7.2 Sentiment Extraction

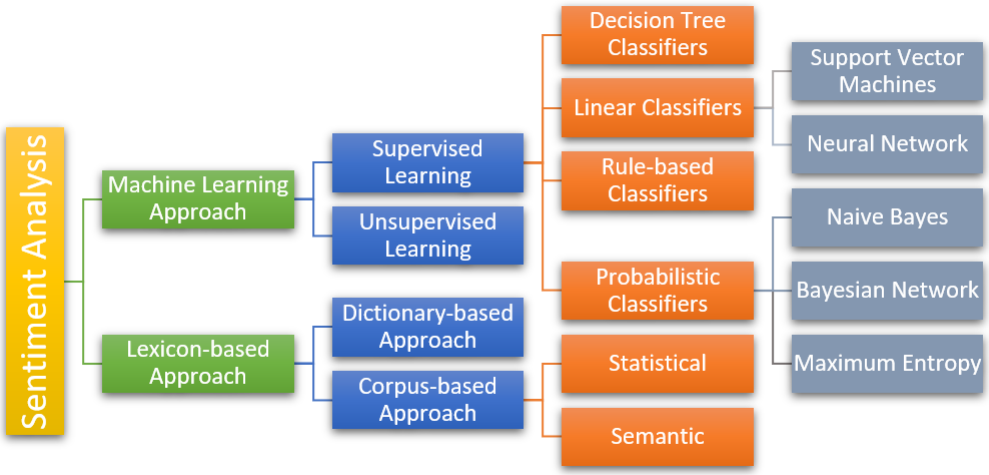


Figure 2.11: Different Approaches and Models used for Sentiment Analysis

(Source: Devopedia)

Sentiment extraction refers to understanding and establishing the sentiment of the person who generated the data, for example, the author of a tweet or the speaker of a dialogue. NLP, statistics and text mining together play important roles in determining whether a customer's sentiment about a particular product or service is positive, negative or neutral. Those

[15]https://medium.com/explore-artificial-intelligence/introduction-to-named-entity-recognition-eda8c97c2db1

are the three polarities of sentiment that are generally considered. In rare cases, two more are considered highly positive and highly negative which are the extremes. This practice is called the category[16] approach to sentiment analysis since it has fixed categories. This is the popular approach among the industry. A disadvantage of sentiment analysis is difficult to obtain labelled data. When the text or sentence has errors, missing punctuation or is too short[17], sentiment extraction will be affected. There can be two types of sentiment extraction- computational learning techniques and semantic approach as seen in figure 2.11.

Using pre-populated dictionaries of words called lexicons is a popular method in semantic type (Turney, 2002). Only the important words are selected using stopwords removal, stemming and lemmatization and then each word is compared to the dictionary of sentiments.

Computational learning techniques (Pang, Lee, & Vaithyanathan, 2002) include using any machine learning classifier model to train on the syntax of the text.

Sentiment Dimensionality

Categorical sentiment analysis is more popular. But a more efficient technique would be considering sentiment into dimensions. Dimensional sentiment analysis helps to fine grain the sentiment value (Wu, Wu, Huang, Wu, & Yuan, 2017). There are many dimensions of sentiment but the first ones to be considered are valence and arousal (VA) pair. VA pair was first introduced as dimensions of sentiment by Dr James Russell (Russell, 1980). Valence axis refers to the polarity or degree of sentiment in the text while the arousal axis shows the intensity. In other words, valence describes how the person is affected and arousal describes how excited the person is about what happened.

The valence-arousal space of sentiment can be seen in figure 2.12 on page 29. Any sentiment can be expressed with a VA pair as coordinates by determining the VA values of the textual

[16]https://algorithmia.com/blog/introduction-sentiment-analysis

[17]https://www.meaningcloud.com/blog/an-introduction-to-sentiment-analysis-opinion-mining-in-meaningcloud

data (Malandrakis, Potamianos, Iosif, & Narayanan, 2011; Paltoglou, Theunis, Kappas, & Thelwall, 2013). In this work as well, valence and arousal values of the sentiment are considered to quantify the drug reviews apart from ratings.

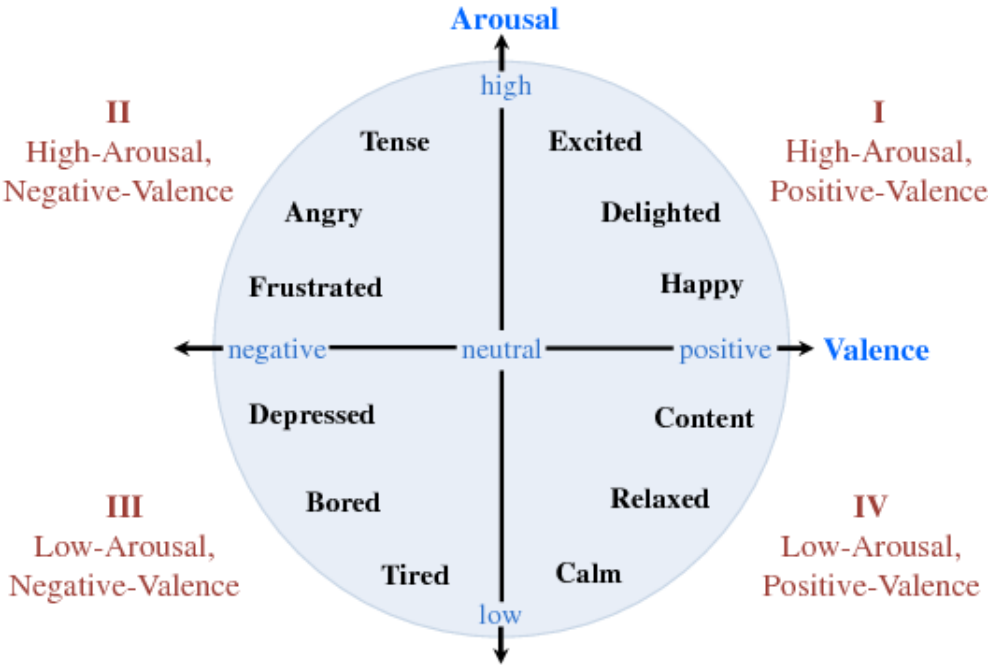


Figure 2.12: Sentiment Dimensionality: Valence and Arousal coordinates

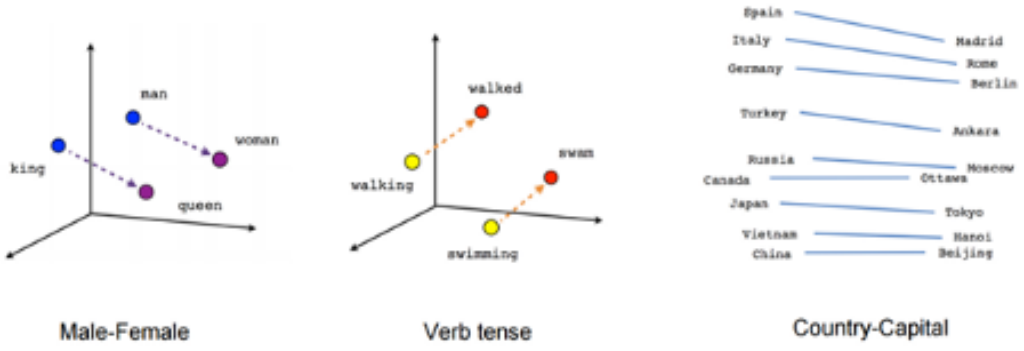
(Source: Wu, 2017)

2.7.3 Word Vectorization

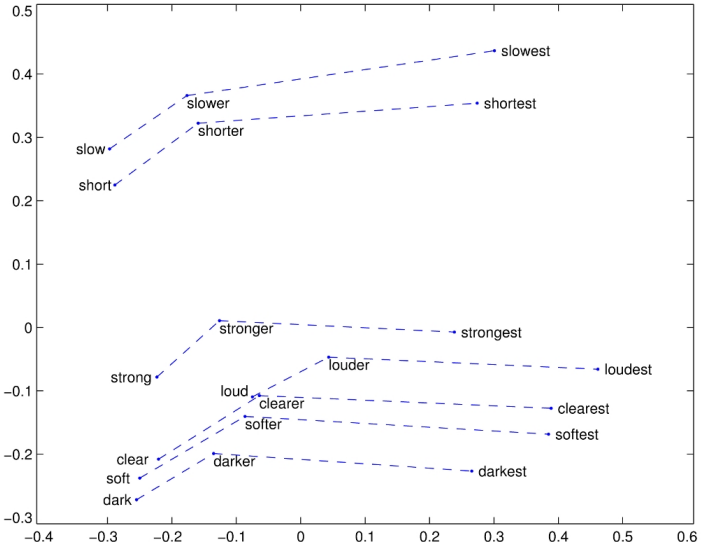
Word vectorization is used to convert the text in words to numerical features, since, machine learning requires numerical features to process upon. There are various methods of doing this vectorization which include:

- Bag of Words-Count of words is calculated and represented as a token with the word.
- Term Frequency/Inverse Document Frequency (TF/IDF)- Term frequency stands for the number of occurrences of the word in the document and inverse document frequency stands for the number of occurrence in the document corpus.
- word2Vec by Google- word2Vec is a pre-trained vectorization model trained on data

from Google and an example is given in figure 2.13(a)[18]. This model records grammar also apart from words. It is widely used for the machine learning scenario in the world.



(a) word2Vec



(b) GloVe

Figure 2.13: Word Vectorization Types

- GloVe- Global Vectors for word representation[19] (GloVe) vectorizes words using an unsupervised model. it uses a high dimensional space to map each word and hence, words with the same context should be mapped closer to each other. An example of

[18]https://medium.com/natural-language-processing-text-data-vectorization-af2520529cf7

[19]https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db

mapping done in GloVe is shown in figure 2.13(b)[20]. This algorithm is used in this study.

2.8 Model Validation Techniques

Thorough model validation is necessary to ensure generalization in the output of the model. There are various validation techniques and each works well only for certain types of models. The different types of validation techniques[21] are:

- Train/test split
- k-Fold Cross-Validation
- Leave-one-out Cross-Validation
- Leave-one-group-out Cross-Validation
- Nested Cross-Validation
- Time-series Cross-Validation
- Wilcoxon signed-rank test
- McNemar's test
- 5x2CV paired t-test
- 5x2CV combined F test

2.8.1 Fitting of the Model

Model fitting is an important requirement to ensure the model represents the real-world data. Overfitting and underfitting are the two main model fitting issues faced. Figure 2.14 shows how model capacity affects its fitting.

[20]https://nlp.stanford.edu/projects/glove/

[21]https://towardsdatascience.com/validating-your-machine-learning-model-25b4c8643fb7

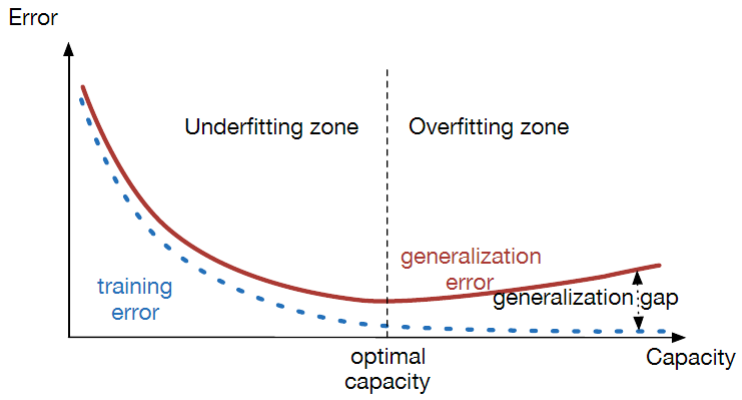


Figure 2.14: Model Capacity vs Model Fitting

(Source: SRDAS.GITHUB)

Model fitting is an important requirement to ensure the model represents the real-world data. Overfitting and underfitting are the two main model fitting issues faced. Figure 2.14 shows how model capacity affects its fitting.

Overfitting occurs when the model fits the data too well, including the noise in the data. When this happens a new real-world, unseen data will not be properly modelled. In overfitting, the model shows low bias, but there will be high variance[22]. Complications in model architecture can lead to overfitting. Using multiple models and cross-validation, overfitting can be prevented.

Underfitting occurs when the model does not fit the data good enough. The model fails to recognize the core patterns in the data. As the simplicity of the model increases chances of underfitting also increases. Models showing high bias but very low variance[22] will lead to underfitting.

[22]https://chemicalstatistician.wordpress.com/2014/03/19/machine-learning-lesson-of-the-day-overfitting-and-underfitting/

2.8.2 Cross Validation

Cross-validation is the general process wherein machine learning models are validated to ensure they are properly fit and that it represents the real-world scenario. Only the main validation techniques are discussed in detail here.

Splitting of Data sets

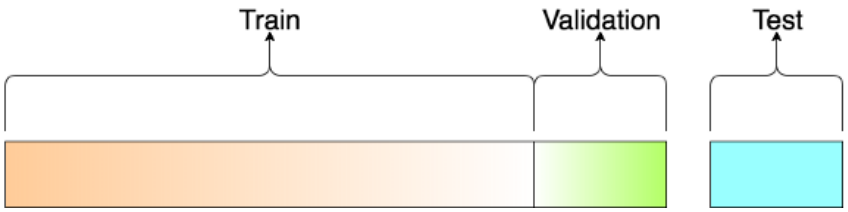


Figure 2.15: Splitting of Datasets into Train, Validation and Test sets

(Source: Towards Data Science)

Splitting datasets is the easiest method of model validation. Datasets may be split into training, test and validation sets. The most common method is dividing the into training and test sets, where the model is trained on the training dataset and then tested on the test set to know how it behaves with previously unknown data. If the dataset is small, to begin with, then this method is not efficient since it keeps apart a portion of the data that could have been used for training the algorithm. This can cause bias in resulting models. A representation of the splitting of a dataset is seen in figure 2.15.

K-Folds Cross Validation

K- folds cross-validation is one among the best methods of validation if the dataset size is limited and splitting cannot be considered. K-folds makes sure that every piece of data is used for both training and testing.

The steps in the K-fold algorithm are:

- Split the entire data into K folds. The value of k should be considerate of the size of the data. practically chosen values are 5 or 10. The splitting should be random.
- Leave the first fold out and use all the k-1 remaining folds for training the model and finally use the first fold to test the model.
- Use all the k-1 folds for training excluding the second fold which is used for testing.
- Repeat these steps until each fold had a chance to act as test data. At that point, the average of the measured values in all the iterations would give the model performance metric.

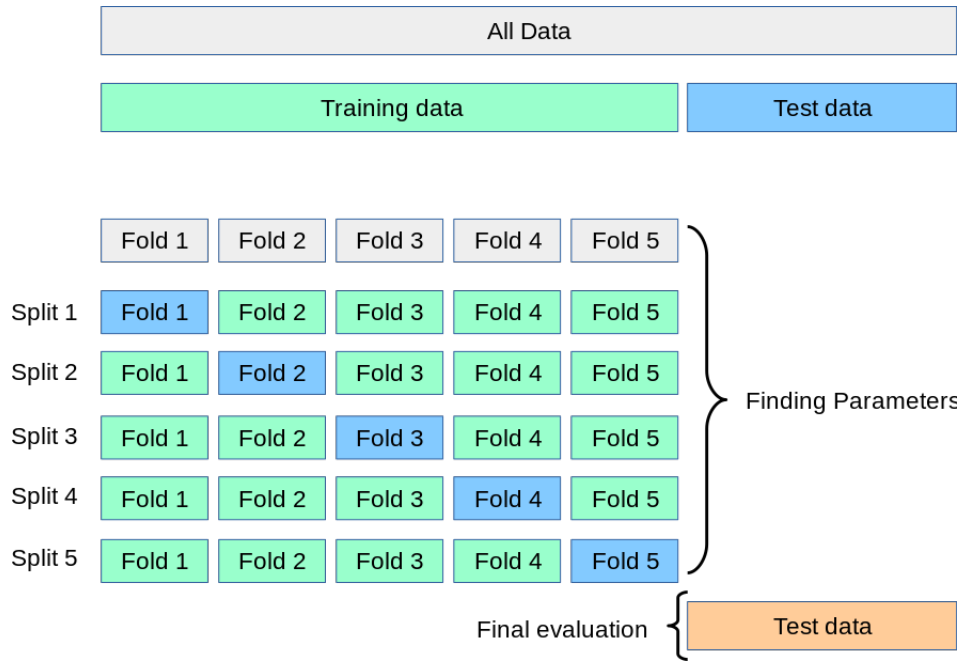


Figure 2.16: k- fold Cross Validation Scheme

(Source: SCIKIT Learn)

This algorithm can be seen in figure 2.16. As the value of k increases biasing of the model will decrease but going beyond a point could cause variance. As k decreases it will then approach the same case as the splitting of data[23]. Another form of k-folds is stratified k-folds cross-validation. The difference is here when the folds are made, in each fold the percentage of all the classes will be the same as in the complete dataset.

[23]https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f

Leave One Out Cross Validation (LOOCV)

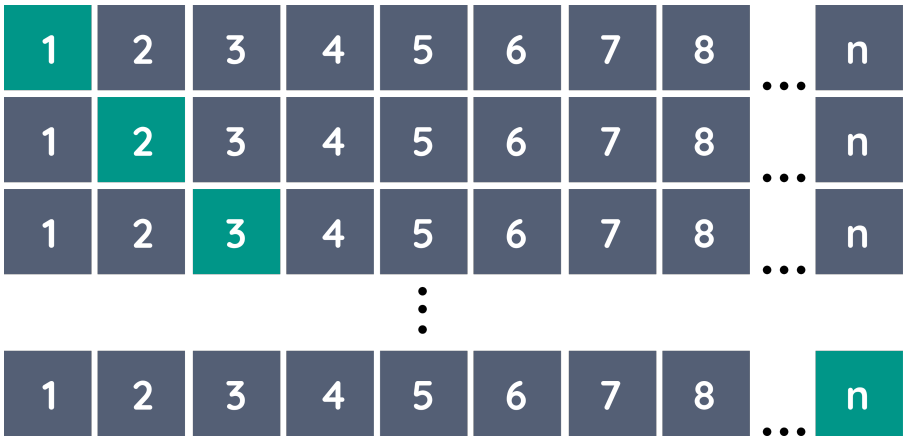


Figure 2.17: Leave one out Cross Validation (k=n)

(Source: Towards Data Science)

Leave one out cross-validation is a modified form k-folds cross-validation. In LOOCV the value of k is taken as n, where n is the total number of elements in the data. Therefore the algorithm will be modified as leaving one sample out for testing in each iteration and using all n-1 remaining for training as can be seen from figure 2.17.

Bootstrap

Bootstrap is a powerful method of including randomness into the data selected for training and testing. Similar to k-fold, this is also an iterative process and the final value is the average of all iterations. The 0.632+ bootstrap (Efron & Tibshirani, 1997) was developed to prevent overfitting of models. The non-information error rate (ˆγ) is defined as the error in classification considering that the independent variables and the data classes are not related. Let φ(x;D) be a classification model made on data D, then:

$$\hat{\gamma} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(c^i, \phi(x^j, D)) \tag{2.14}$$

$$\hat{R} = \frac{\widehat{Err^{(1)}} - \overline{err}}{\hat{\gamma} - \overline{err}} \tag{2.15}$$

$\hat{R}$ is called the relative overfitting rate. The 0.632+ measure comes from the equation

$$\widehat{Err^{(0.632+)}} = (1 - \frac{0.632}{1 - 0.368\hat{R}})\overline{err} + \frac{0.632}{1 - 0.368\hat{R}}\widehat{Err^{(1)}} \tag{2.16}$$

0.632+ bootstrap is better than zero bootstraps and to an extent, overfitting is prevented.

2.9 Model Evaluation Techniques

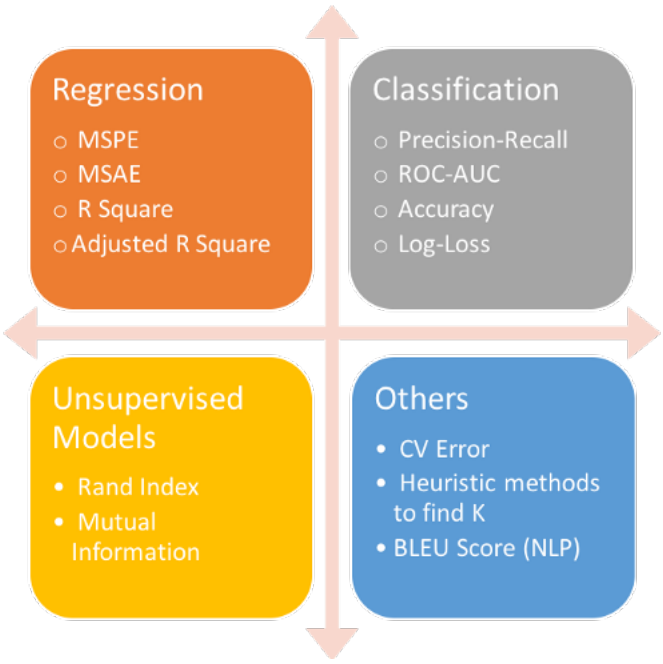


Figure 2.18: Evaluation Metrics vs Types of Models

(Source: USF Data Institute)

While model validation methods determine whether the model trained actually represents the real world, other methods are needed to know if the model output is efficient. Such methods are discussed here. There are many model evaluation techniques but each is better to be used for certain types of models. This is shown in figure 2.18. In this work, neural network regression models are used to extract either drug ratings or valence-arousal pair from the drug reviews. Therefore, we have a regression problem with drug reviews as the independent variable and either rating or VA as the dependent variable. So, the metrics

chosen for evaluation here are Root Mean Square Error (RMSE), Mean Absolute Error (MEA), R^2 error and Pearson's correlation coefficient r.

2.9.1 Root Mean Square Error (RMSE)

The difference between observed and actual values are called residuals[24]. RMSE is defined as the standard deviation of these residuals. RMSE describes how the data points are with respect to the line of best fit. Apart from machine learning applications, RMSE is used in weather forecasting[25]. It can be represented using the mathematical expression:

$$RMSE = \sqrt{\overline{(Exp - Obs)^2}} \tag{2.17}$$

where the factor inside the root symbol is the mean of the square of the differences. This expression can be further expanded to:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{2.18}$$

where n is the sample size in the data.

2.9.2 Mean Absolute Error

MAE is defined as the average of the differences between the predicted and the obtained values.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{2.19}$$

where, again, n is the sample size. MAE becomes same as RMSE when the residual values are either null or the same[24] throughout the data.

[24]https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

[25]https://www.statisticshowto.datasciencecentral.com/rmse/

2.9.3 R Squared (R^2) and Adjusted R^2

Both R squared and adjusted R squared metrics are useful in understanding how the independent variables affected the changes in the dependent variable.

$$\hat{R}^2 = 1 - \frac{\sum_{j=1}^{n}(Y_j - \hat{Y}_j)^2}{\sum_{j=1}^{n}(Y_j - \overline{Y}_j)^2} \tag{2.20}$$

$$R^2_{adj} = 1 - [\frac{(1 - R^2)(n-1)}{n-k-1}] \tag{2.21}$$

where n is the sample size and k refers to the number of independent variables. Adjusted R squared is directly proportional to the number of useful predictors while R squared is proportional to the number of predictors whether they are useful or not.

2.9.4 Pearson's Correlation Coefficient

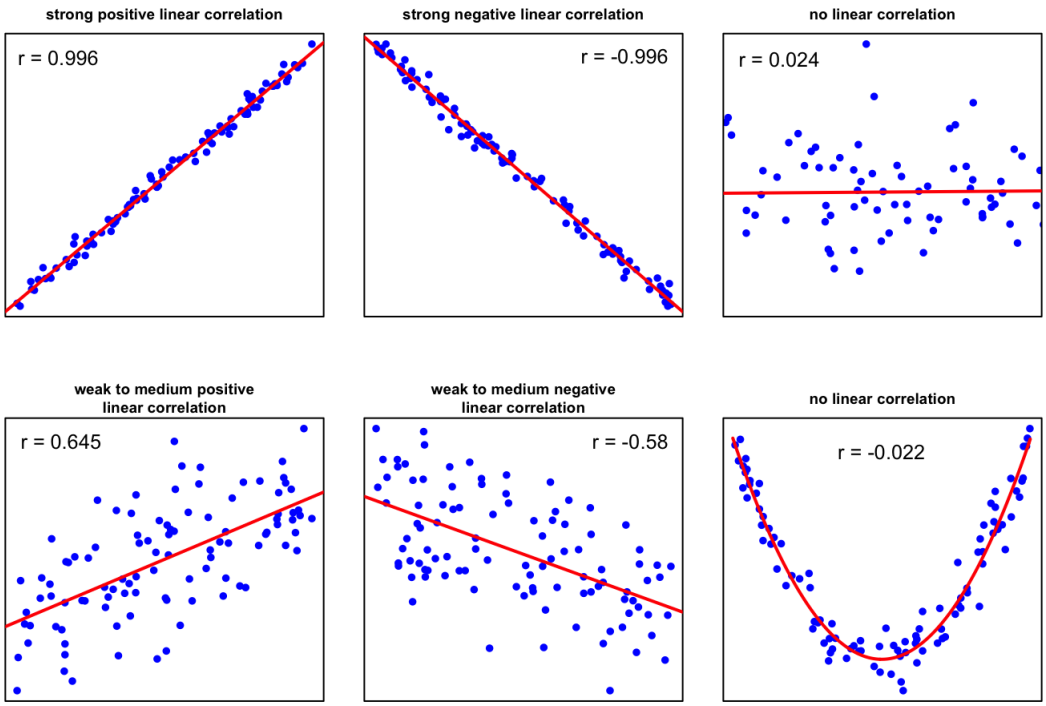


Figure 2.19: Graphs of various correlation coefficient R values

(Source: Dept. of Earth Science, Freie University, Berlin)

Correlation coefficient between two variables shows how strongly dependent are they with each other. In machine learning application, especially in linear regression, the correlation coefficient[26] used is Pearson's R. The value of R ranges from -1 to +1. This is shown in figure 2.19.

- If R=-1, there is a strong negative relationship between the two variables. A unit change in one variable will cause proportional units of decrease in the other.
- If R=0, there is absolutely no relation between the two. Change in one does not affect the other variable.
- If R=+1, it indicates a strong positive relationship. For a unit change in one variable, there will be proportional units of increase in variable two.

2.9.5 Cohen's Kappa Interrater

Kappa value interpretation .
<0 No agreement
0 — .20 Slight
.21 — .40 Fair
.41 — .60 Moderate
.61 — .80 Substantial
.81–1.0 Perfect

Figure 2.20: Interpretation of Kappa values

(Source: (Landis & Koch, 1977))

Kappa measurement is used to understand the agreement level between two rating sources and to establish whether any of those sources biased the results. It is an interrater reliability[27]

[26] https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/correlation-coefficient-formula/

[27] https://towardsdatascience.com/interpretation-of-kappa-values-2acd1ca7b18f

measure. Kallumadi et al. (2018) used kappa measure to compare the ratings made by the users in the drug datasets to that predicted by their model. Hence, to compare this study with their work, Cohen's kappa is used and the comparison is given in chapter 5.

2.10 Conclusion

This chapter has eight main sections. The last seven are dedicated to establishing the literature on all the possible algorithms and methods used in this research work. They are defined and in most cases, it is explained how they are used in this study. The first section deals with the existing papers and publications on the particular domain of information and sentiment extraction from text and in particular drug reviews.

Section 2.2 has discussions about all the relevant research papers referenced to complete this study. Sentiment extraction and conversion to rating of comments, drug reviews, in this case, has been tried out many times and most of them were successful. All the existing approaches to solve the problem are noted down and described. Cross- data and cross-domain approach by Kallumadi et al. (2018) is finally decided to be the best approach in understanding the migration capability of models. All the gaps and issues faced by previous researchers are also explained in this section.

There are various data mining life cycles and practices which are compared with each other. out of the Team Data Science Process (TDSP) is selected for carrying out this work.

Data mining and Machine Learning are two broad subcategories of Data Science and are interrelated. They are dependent on each other but not synonymous. The fields of applications of machine learning are listed in section 2.4. Supervised, Unsupervised and Reinforcement learning types of machine learning algorithms are discussed in detail.

Regression is an example of supervised learning techniques and is used to find relations between target and predictor variables. The types of regression are listed in section 2.5 and

two important ones among them, linear and logistic regression are explained further.

Deep learning is the branch of data science that uses neural networks. Neural networks are advanced models with sequential connections between an input and an output layer. they usually have arrays of neurons between these connections which apply an activation function on the input to get the output. Each neuron has its own activation function. Back-propagation is a popular algorithm in neural networks which is used to reduce the error at the output layer. types of neural networks are listed in section 2.6. Convolutional Neural Network (CNN) has two main activation functions, convolution and pooling. They are used extensively in signal and image processing. Recurring Neural Networks (RNN) are advanced designs that can save data from previous iterations. They have looped nodes that persist a portion of the input. Long Short Term Memory (LSTM) is a further advance form of RNNs. LSTM reduces gradient vanishing which is a drawback of RNNs. LSTMs can remember large sequences of input data by use of three gates- input, forget and output.

Text mining is another branch in data science that is very similar to data mining but uses different algorithms. All the essential algorithms in the field of text mining are listed in section 2.7. natural Language Processing (NLP) is the capability of understanding natural languages while Named Entity Recognition (NER) is the process of determining entities from a text such that they can be grouped and analysed. sentiment extraction is the most important feature in text mining. It is the process of extracting sentiments from users comments or voice. There are two types of sentiment extraction, categorical and dimensional. Categorical sentiment extraction classifies the sentences into pre-defined classes of positive, negative and neutral. Dimensional sentiment extraction, on the other hand, considers sentiment to be a dimensional value.

Model validation is an essential step to determine how effectively the model represents the real-world scenario. The main validation techniques are discussed in section 2.8 like data splitting, k-folds cross-validation, leave one out cross-validation and bootstrap validation.

The evaluation techniques used in this research are explained in detail with equations in section 2.9. They are RMSE, MAE, R squared and Pearson's correlation coefficient.

This section also explains why each deep learning algorithm, used in chapter 3, is selected and which works influenced that decision.

Chapter 3

EXPERIMENT DESIGN AND METHODOLOGY

3.1 Introduction

Chapter 2 discussed all the machine learning algorithms that are used in this research work. This chapter deals with how these algorithms and models are clubbed together or how they are sequenced to get the desired output.

Section 3.2 shows the descriptive design flow of this research and each block in the flow diagram is explained in detail.

Section 3.3. discusses the business understanding phase in TDSP and why business understanding is necessary.

In section 3.4, data acquisition and understanding is the main topic. The data sets used in this work, their source and their components are described. The relevant fields in the dataset that are chosen for this research are described in detail. Data preparation steps are also discussed.

The algorithms used for modelling phase are defined in section 3.5 including model evaluation techniques. The three proposed experiments are described: in-domain, cross-source

and cross-condition research cases.

3.2 Design Flow

Figure 3.1 shows the flow diagram of this entire machine learning experiment in adherence to the TDSP life cycle.

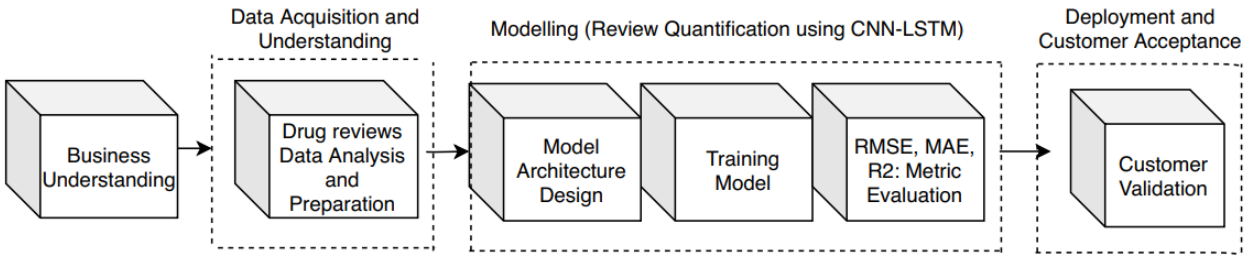


Figure 3.1: TDSP Level Design Flow prepared for this study

At the start, there is business understanding which deals with what the customer or client needs from the proposed experiment.

Data acquisition and understanding is the next phase and includes the pre-processing of the data and analysing.

Next phase is modelling and it has three sub-stages- CNN-LSTM model design for text quantification, model implementation and model evaluation.

Deployment and Customer acceptance phases are related to industry-level applications.

3.3 Business Understanding

The business requirement that supports this research is explained in detail in chapter 1- sections 1 and 2 and chapter 2- section 9.

Business understanding usually deals with formulating aims such that the customer can either solve a problem or increase their financial turnover. Another part of business understanding is selecting useful data to do the analysis. Companies handle lots of unprocessed data and the relevant portion must be selected to do the machine learning. Even though this study is not backed by any company, once successful, this can give huge support in any service or manufacturing company especially the pharmaceuticals. Since there is no company in the background, two general drug databases are considered that hosts data of various pharmaceutical companies.

Essentially, what required is a method that can efficiently convert customers' reviews and comments into product or service ratings. The output of any business understanding phase is hypotheses. The research hypotheses for this study are briefly defined in chapter 1, section 3. They are put in a proper structure here:

- H_0: The efficiency of information extraction from drug reviews **can not be improved** if drug review quantification, using regional CNN-LSTM models and Natural Language Processing (NLP), is implemented to extract either rating on a 10-point scale or valence-arousal value, as compared to using machine learning models alone which had a Cohen's kappa of 83.99 for In-domain analysis.

- H_A: The efficiency of information extraction from drug reviews **can be improved** if drug review quantification, using regional CNN-LSTM models and Natural Language Processing (NLP), is implemented to extract either rating on a 10-point scale or valence-arousal value, as compared to using machine learning models alone which had a Cohen's kappa of 83.99 for In-domain analysis.

3.4 Data Acquisition and Understanding

3.4.1 Data Acquisition

There are two relevant datasets planned to be used for this research. Both are publicly available in the UCI data repository and were initially retrieved from two websites: Drugs.com

and Drugslib.com.

DRUGS.com (215603x6)	**Field**	*condition*	*review*	*rating(dependent)*
	Definition	Health condition	User comment	User rating on 10-point scale
	Type	Textual data	Textual data	Numerical

Table 3.1: Drugs.com: Relevant fields and types

The first dataset from Drugs has 215,603 rows across 6 fields. The ones relevant to this study are drugName, condition, review and rating, where the review is the independent variable and rating is the dependent variable. Condition is also an independent variable in one of the study cases, which are explained in detail in the next section. The field rating ranges on a 10-point scale and review is a textual entry by the users of the drugs.

The second dataset from Druglib has 4,143 instances across 8 attributes, out of which the required ones are commentsReview and rating. Here also rating field ranges on a 10-point scale.

Both the datasets are partitioned to training and test sets with a 3:1 ratio. The relevant fields of drugs.com and drugslib.com are described in tables 3.1 and 3.2 respectively.

DRUGSLIB .com (4143x8)	**Field**	*commentsReview*	*rating(dependent)*
	Definition	User comment	User rating on 10-point scale
	Type	Textual data	Numerical

Table 3.2: Drugslib.com: Relevant fields and types

3.4.2 Data Understanding

Now that the contents of the datasets are known, the next step is to understand how strong and supportive each element of the data is. It is necessary to perform all the mathematical measurements on the data points. The most important measurements are:

- Looking for missing values and duplicate values in the independent as well as dependent variables.
- Looking for intercorrelations between the independent variables but since we have only one independent variable here, this is not necessary.
- Calculating mathematical measures of the variables like measures of central tendency, measures of deviation, skew etc.
- Using visualisation techniques like Tableau and python plotting packages to graphically show how the data is and behaves.

3.4.3 Data Preparation

The behaviour and features of the selected dataset are understood and now it is needed to take actions on whether to keep or discard those behaviours. Since the independent variable is textual, all the text cleaning and preparation modules must be used. They are listed below:

- User must have written the drug's name and the quantity intake in the review which is not necessary for this research work. Therefore drugs' names and quantity measurements are replaced with the same generic words.
- Spelling correction is necessary before analysing text data. The python package named 'pyspellchecker' will be used for this purpose. A NER module is also to be used to exclude names of entities from spell correction.
- Presence of abbreviated words will affect sentiment extraction. Hence words are to be compared with a pre-populated list of abbreviations and expanded.
- Stop words should be removed such that no unwanted processing is carried out.

- Conversion to lower case, stemming and lemmatization are the last few steps.

3.5 Modelling

3.5.1 Model Flow Definition

Figure 3.2 shows the detailed flow diagram of all the steps within the modelling phase. The research design has mainly four levels as shown in the figure. The first level is training the deep learning regression model architecture with drug reviews as the independent variable and drug ratings as the dependent variable. This requires regional CNN and LSTM models-

1. Regional CNN: The review is split into n arbitrary regions (here n is taken as 5). Each region is embedded using GloVe vectorization and fed to a dedicated Convolution layer(regional CNN)to extract affective features.

2. LSTM: It has a memory unit, and hence, can be used to understand the long-distance relationship between the regions and predict the ratings effectively.

In the second level, the trained regional CNN-LSTM model is used to inference the drug reviews in the test data to get the drug rating predictions

The third level is the evaluation of the implemented model. It deals with the comparison of the predicted ratings and the actual drug ratings from the test data. Root Mean Square Error, Mean Absolute Error and Pearson Correlation coefficient will be used as mentioned in chapter 2 section 8.

In this model implementation, instead of providing drug ratings as the data label, if drug review valence-arousal values are given, then VA predictions will be obtained, which is, as explained in the first chapter, another way of quantifying reviews. That is, at this stage, there are two possible ways to proceed, either with drug ratings or with drug review sentiment VA values. But, since the available data labels are drug ratings, it is decided to consider ratings for the remainder of this work. In future, if the alternate is needed to be chosen, then the

drug data can be labelled/annotated with valence-arousal values using the VA lexicon given in Appendix B. However, VA sentiment values are proportional to ratings and vice versa (Yin, Zhang, & Li, 2014) and hence can be considered identical.

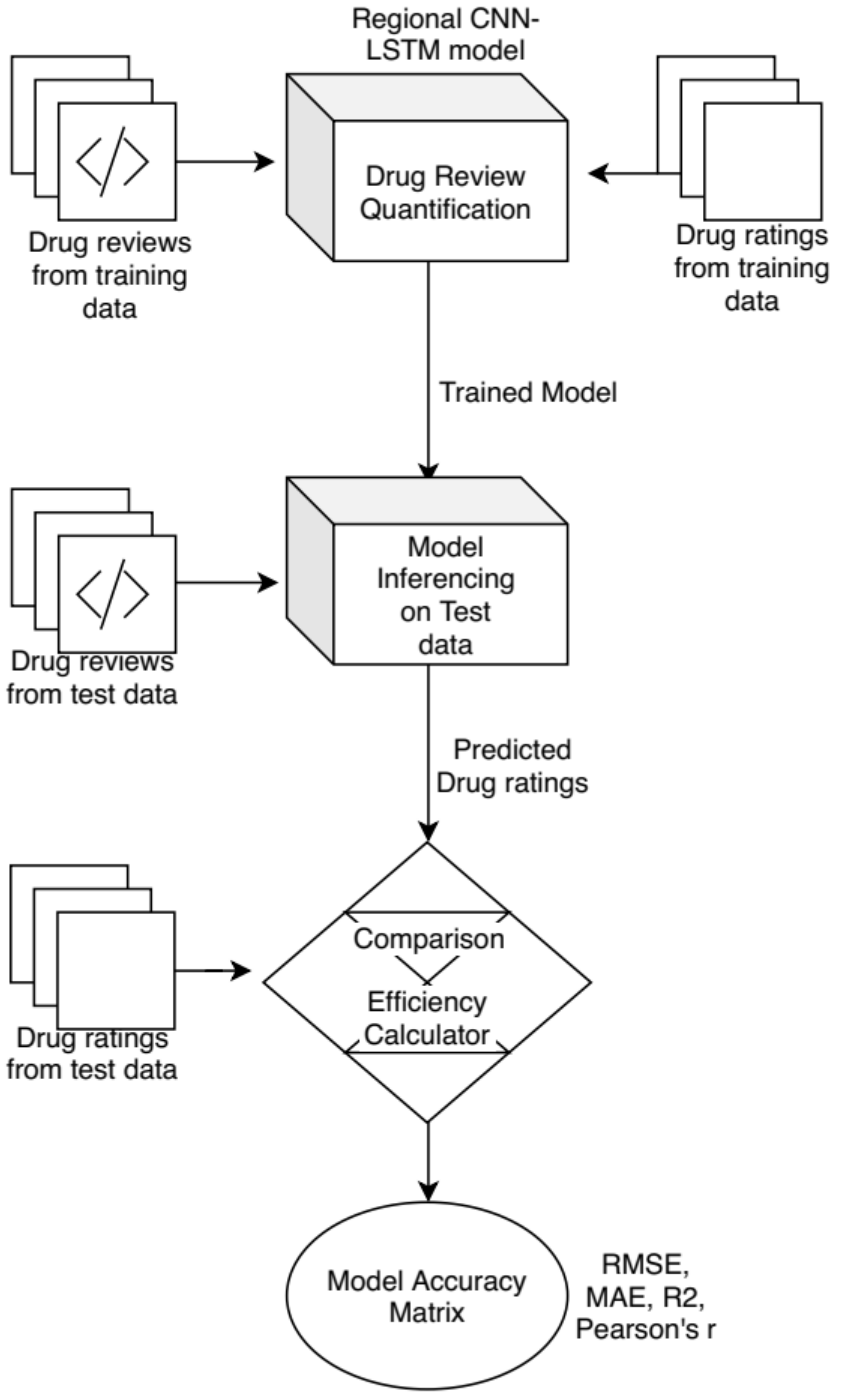


Figure 3.2: Detailed Design Flow of the Modelling phase that shows the three steps

3.5.2 The Three Experiments

There are three experiments in this design, but the steps are the same for all the three except for the data sets used for testing and training. In other words, the entire process flow shown in figure 3.2 on modelling is repeated for each of these three experiments and only the input datasets are to be changed. These experiments are designed to know how efficient the developed models are with respect to changing dataset conditions. Understanding the flexibility of the design will help to generalize the design to the entire medical domain and to other domains as well.

Experiment	**Training Data**	**Test Data**
In-Domain	Drugs.com	Drugs.com
Cross-Condition	Birth Control	Anxiety, Pain and Depression
Cross-Source	Drugs.com	Drugslib.com

Table 3.3: The Three Experiments

Experiment 1: In-Domain

In this experiment, both the training and the test data sets are taken from the same source, that is Drugs.com page. This is the dataset with the maximum available instances and hence both the test and training sets will have enough data-points to assess the models efficiently.

Experiment 2: Cross-Condition

For this experiment, model training is done on drug reviews of birth control and the testing is done on test data of depression, pain and anxiety. In other words, the model is trained in one condition and it is tested in another three conditions. These disease conditions are not selected randomly. They are the ones with the maximum number of drugs and reviews listed in the first dataset.

Experiment 3: Cross-Source

The models, both sentiment extraction and regression are trained on Drugs.com training data and then tested on Drugslib.com data. Hence, only the test data from Drugslib.com is considered and so the issue of non-availability of a large number of instances in Drugslib dataset will not affect the design.

3.6 Conclusion

This chapter dealt with establishing and describing all the experimental steps that need to be carried out for the successful completion of this research work.

Each relevant blocks in the TDSP life cycle are associated with the experimental setup of this study and they are explained in different sections. Business understanding means formulating a hypothesis to base the research on.

Data acquisition and understanding section defined the structure of the two datasets and also listed the necessary actions to be taken to understand, clean and prepare the data. Data preparation is also a part of this section. Modelling is discussed in the next section, which has four stages. The flow diagram depicting the steps in this stage is also explained. Another important part of this section is the three experiments. They are classified based on the training and test data used.

Next chapter, chapter 4, discusses the implementation of all the design steps mentioned in this chapter.

Chapter 4

IMPLEMENTATION AND RESULTS

4.1 Introduction

The last chapter showed the design of this research setup and here, in this chapter, the implementation of all the phases of that design is elaborately discussed. Python is the programming language used to implement the research.

Section two in this chapter discusses steps for data understanding. A thorough analysis is to be done on the dataset to identify patterns and behaviours that can support and oppose this research work. Next section helps in eliminating unwanted behaviours from the data and amplifying the useful patterns. Also, the textual data is cleaned for removing stop words.

In the fourth section, architectures will be defined to implement the model design. The finalised same architecture will be used for the three experiments. The last section in this chapter discusses the results obtained for each of the experiment and reports whether the model is suitable for that type of experiment.

4.2 Data Understanding

Apart from the generic information on the data, it is always important to understand the datasets in detail for deep learning purposes. A number of mathematical and graphical analysis can be done on the datasets to determine whether there are any inter-lying patterns, exceptional behaviours or outliers.

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8.0	April 27, 2010	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5.0	December 14, 2009	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8.0	November 3, 2015	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9.0	November 27, 2016	37

Figure 4.1: Sample training data

The first step is to analyse the data distribution and do feature engineering. Figure 4.1 shows a sample of how the training data is structured. Out of the seven features, the only ones relevant for this study are *condition, review* and *rating*. Hence, it is necessary to look deep into these three columns.

4.2.1 Data Feature: *review*

This is a textual field with the entries of user comments on the drugs they used. The check for duplicates was negative indicating no duplicate instances in this field. Also, there are no null entries. This is the independent feature of this study and most of the data preparation works are to be done on this field.

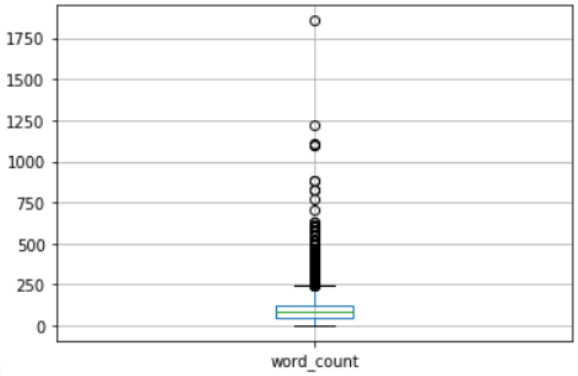
Word Count

The size of each review can be computed based on the number of sentences or words. Since word count gives a more clear picture, it is used here. The total word count of each review

instance is calculated. The summary of the obtained word count and its boxplot can be seen in figure 4.2. There are outliers concerning the word count, as seen in the figure and should be removed from the analysis.

```
count    161297.000000
mean         85.646788
std          45.507389
min           1.000000
25%          49.000000
50%          85.000000
75%         127.000000
max        1857.000000
Name: word_count, dtype: float64
```

(a) Summary



(b) BoxPlot

Figure 4.2: Word Count of Reviews

4.2.2 Data Feature: *rating*

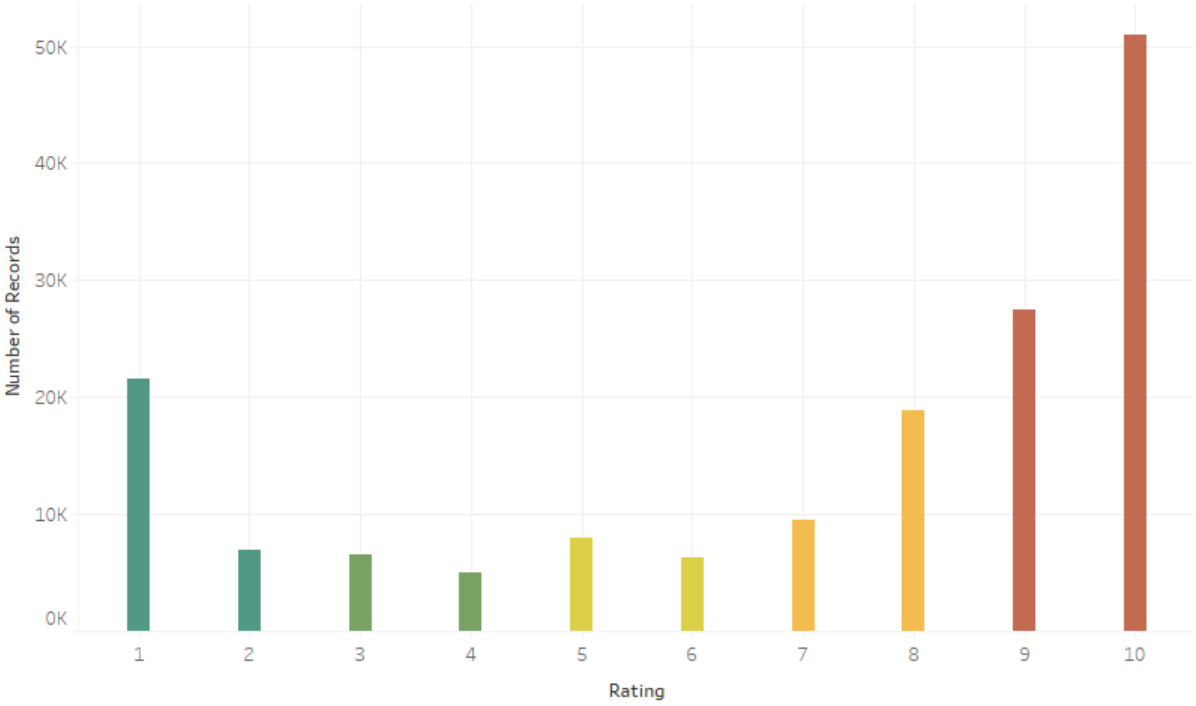


Figure 4.3: Rating distribution

This is a numerical column representing the rating points given by the users for the drugs. The value ranges from 0 to 10 and there are no null entries. This is the target variable for this deep learning application. The distribution of each value on the rating scale, in the training data, is depicted in figure 4.3. It is evident that there is an imbalance in ratings and hence, this needs to be addressed at the data preparation period.

4.2.3 Data Feature: *condition*

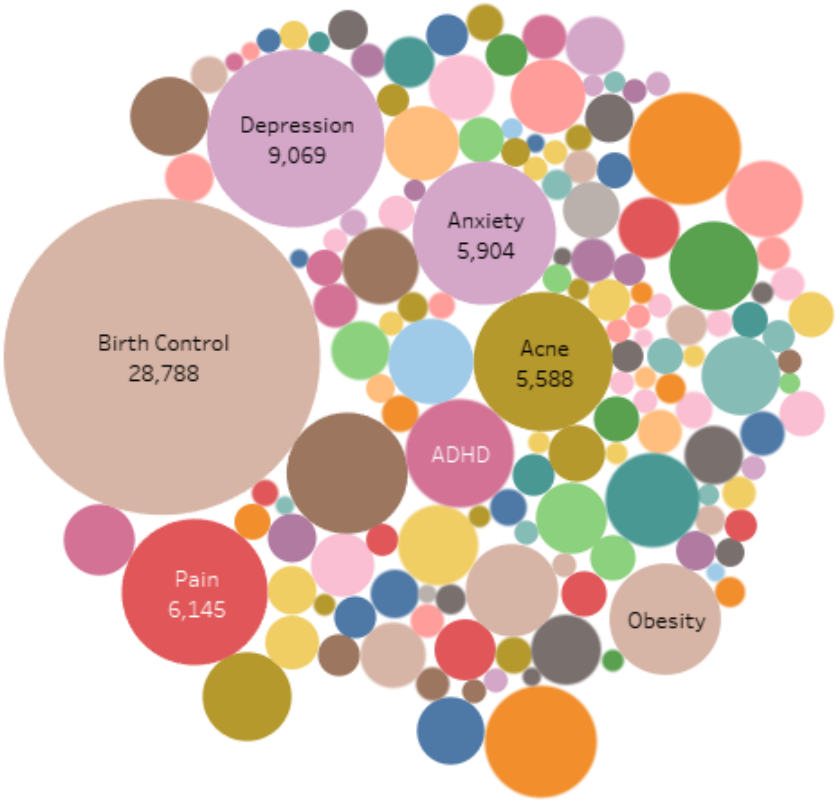


Figure 4.4: Word Cloud of conditions in training data

This is also a textual field which contains the name of the medical condition for which each drug was used. There are no null entries and obviously, there is more than one entry per condition. This field is not directly used for the machine learning purpose but is used in experiment 2 to filter the dataset to select the training and test datasets. The word-cloud of all the conditions in the training dataset is shown in figure 4.4. It can be seen that *BirthControl* has the highest number of entries in the set, followed by *Depression, Anxiety* and *Pain*. Due

to the high representation of these conditions, in experiment 2, the training data will be all the instances of *BirthControl* and test data will be from the other three.

4.3 Data Preparation

Pre-processing of data is the most important step before actual modelling. All the discrepancies discovered during data understanding must be addressed in this section. Different pre-processing steps are described below.

4.3.1 Steps in Data Pre-processing

Selection

Only the fields relevant to the study are kept aside. All the other fields are dropped. The ones remaining are *rating, review* and *condition*.

Null Elimination

As discussed in the previous section of this chapter, there are no null entries in any of the required fields and hence, this step is avoided.

Replacing Drug Names and Quantities

Since the textual data is the collection of reviews of drugs, there are entries of the name of the drug and the quantity of the drug taken. These two entities contribute almost null to the sentiment of the user. Hence, the drug names are replaced with a generic term **drugName** and all the quantities such as *millilitre* and *milligram* with the generic term **qty**. This, in turn, helps to reduce redundancy up to an extent.

Spelling Correction

SpellChecker function in python is used to correct the spellings of all the words in the datasets. Corrected spelling will be helpful in comparing the corpus with the valence-arousal lexicon.

Abbreviation Correction

People often tend to include abbreviations in the reviews which can affect sentiment extraction. So, these abbreviations are expanded by comparing with a public list of abbreviations called *shortforms* extracted from Webopedia[1].

Lowercase Conversion

User review can be in any case, full upper or camel or full lower cases. It is always preferred to bring them to one case before machine learning applications.

Stopwords Removal

It is necessary and useful to remove the stopwords from the textual data which will help in reducing processor load.

Punctuation Removal and Tokenization

All the unwanted punctuation and other characters are removed. Tokenization is the process of splitting the continuous flow of textual data into words, phrases or sentences called tokens. Most of the remaining punctuation marks will be removed during tokenization. Tokenized words are finally fed to the text mining models.

Dummy Variables

The text data reviews are converted to tokens and hence, the numerical data is encoded with dummy variables to match the size of the tokenized independent variable.

4.3.2 Word Embedding

The embedding algorithm used for this study is GloVe word embedding as discussed in chapter 2. The pre-trained GloVe model is prepared by Stanford NLP[2]. An embedding dimension of 50 is used here, that is each word is converted to a vector with length 50.

[1] https://www.webopedia.com/quick_ref/textmessageabbreviations.asp

[2] https://nlp.stanford.edu/projects/glove/

Before embedding, the drug reviews are first split into 5 regions to apply regional modelling. Figure 4.5 shows the before and after versions of sample reviews.

	review	rating	word_count	region1	region2	region3	region4	region5
9	"I was on this pill for almost two years. It d...	2.0	93	"i was on this pill for almost two years. it d...	however my experience at first was it didn&#03...	it my sex drive went down, along with being ve...	cry one second and then get angry with my husb...	a lot worse, i broke out in places i never had...
10	"Holy Hell is exactly how I feel. I had been t...	1.0	131	"holy hell is exactly how i feel. i had been t...	of this medicine coupled with the fact noven w...	drugs, and then fire her- and not replace there...	after calling sebula - you act like you don&#0...	i just decided to wean myself off this and pre...

(a) Before Embedding

	review	rating	word_count	region1	region2	region3	region4	region5
9	"I was on this pill for almost two years. It d...	2.0	93	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	[[0.43833, -0.15784, -0.12393, -0.11901, 0.146...	[[0.61183, -0.22072, -0.10898, -0.052967, 0.50...	[[-0.30965, -0.12453, -0.10142, -1.0104, 0.299...	[[0.21705, 0.46515, -0.46757, 0.10082, 1.0135,...
10	"Holy Hell is exactly how I feel. I had been t...	1.0	131	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	[[0.70853, 0.57088, -0.4716, 0.18048, 0.54449,...	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...

(b) After Embedding

Figure 4.5: Word Embedding done using GloVe on Review split into five Regions

4.4 Modelling

Chapter 3 discussed in detail how the flow of data happens throughout the model training, testing and evaluation process. In this section of chapter 4, more emphasis is given on how the model works internally or what architecture to follow to implement the model.

4.4.1 First Architecture: CNN-LSTM

At first, a single CNN-LSTM model architecture is considered, to understand how the regional approach improves the model. This architecture can be seen in figure 4.6.

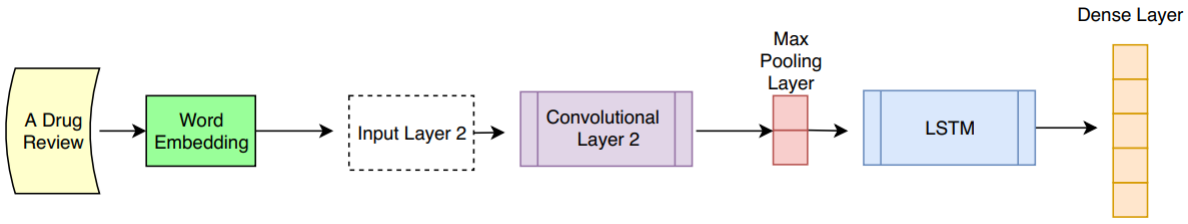


Figure 4.6: First designed Model Architecture with single CNN and LSTM units

The drug review is embedded using GloVe model and is fed to the convolution and max-pooling layers which do feature selection and dimensionality reduction and finally the LSTM layer will find dependencies between the sentences.

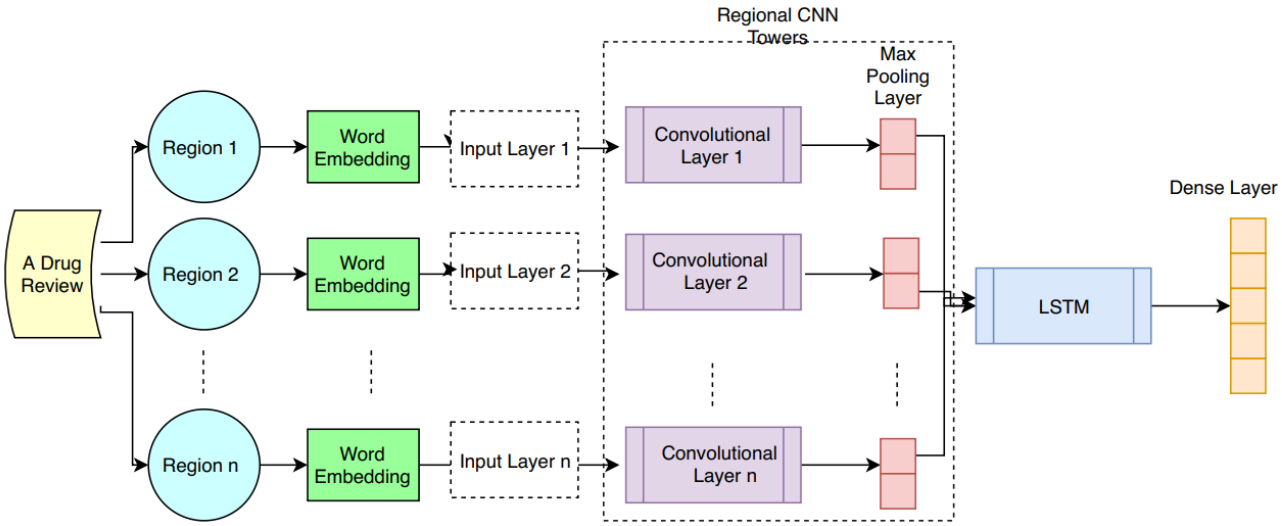
This architecture proved to be unsuccessful since, after a certain number of epochs, the validation loss started increasing due to overfitting of the model and at 50^{th} epoch it crossed the training loss and the validation accuracy at this point was only less than 10 percentage. This was expected, as explained in the literature chapter, since here the model works on the entire drug review and features selected may not be powerful enough to improve accuracy.

4.4.2 Second Architecture: Regional CNN-LSTM

The failure of the first model provided proof that the addition of regional split into reviews could improve the whole situation. Hence, the final architecture is designed where the regional review input is considered. Figure 4.7(a) shows this architecture and is fine-tuned for this research work alone. And figure 4.7(b) shows the summary of the created model printed using *model.summary()* function. In this research, the number of regions is arbitrarily considered to be 5.

A drug review is split into five regions before modelling. Each region is then tokenized and converted to word embedding's with a dimension of 50 using the pre-trained Glove.6B model. These regional vectors are then fed into the convolution layer for feature selection combined with a max-pooling layer for dimensionality reduction of the representation. The outputs of all the five CNNs are then concatenated and fed to the LSTM layer with memory cells to learn the dependencies in the sequence and finally a dense layer to predict the output.

The loss function used in this architecture is Mean Square Error (MSE) which is equivalent to RMSE. The number of epochs run was 475 and had to terminate the model training at this particular epoch, since the validation loss crossed the training loss at that point, the same as in the previous architecture. But, here the accuracy at the terminated point is satisfactory unlike the first architecture and hence this model architecture is finalised for this study.



(a) Final Architecture to Implement the Model

Layer (type)	Output Shape	Param #	Connected to
input_16 (InputLayer)	(None, 300, 50)	0	
input_17 (InputLayer)	(None, 300, 50)	0	
input_18 (InputLayer)	(None, 300, 50)	0	
input_19 (InputLayer)	(None, 300, 50)	0	
input_20 (InputLayer)	(None, 300, 50)	0	
conv1d_16 (Conv1D)	(None, 300, 32)	4832	input_16[0][0]
conv1d_17 (Conv1D)	(None, 300, 32)	4832	input_17[0][0]
conv1d_18 (Conv1D)	(None, 300, 32)	4832	input_18[0][0]
conv1d_19 (Conv1D)	(None, 300, 32)	4832	input_19[0][0]
conv1d_20 (Conv1D)	(None, 300, 32)	4832	input_20[0][0]
max_pooling1d_16 (MaxPooling1D)	(None, 150, 32)	0	conv1d_16[0][0]
max_pooling1d_17 (MaxPooling1D)	(None, 150, 32)	0	conv1d_17[0][0]
max_pooling1d_18 (MaxPooling1D)	(None, 150, 32)	0	conv1d_18[0][0]
max_pooling1d_19 (MaxPooling1D)	(None, 150, 32)	0	conv1d_19[0][0]
max_pooling1d_20 (MaxPooling1D)	(None, 150, 32)	0	conv1d_20[0][0]
concatenate_4 (Concatenate)	(None, 150, 160)	0	max_pooling1d_16[0][0] max_pooling1d_17[0][0] max_pooling1d_18[0][0] max_pooling1d_19[0][0] max_pooling1d_20[0][0]
lstm_4 (LSTM)	(None, 64)	57600	concatenate_4[0][0]
dense_4 (Dense)	(None, 1)	65	lstm_4[0][0]

Total params: 81,825
Trainable params: 81,825
Non-trainable params: 0

(b) Model Summary

Figure 4.7: The Final Model Definition: Regional CNN-LSTM

4.5 Results

As discussed in the previous chapter, there are three experiments to be carried out in this research work. The deep learning model used will be the same as designed in the previous section. The difference will be in the training and test data sets used. Each experiment and the results obtained are explained below.

4.5.1 Experiment 1: In-Domain

This first experiment is the most supportive one for the model since the test data is from the same domain and dataset. Figure 4.8 shows how the model is trained and monitored for experiment 1. The predicted rating is compared with the expected ratings in the test data to get the results which are shown in table 4.1.

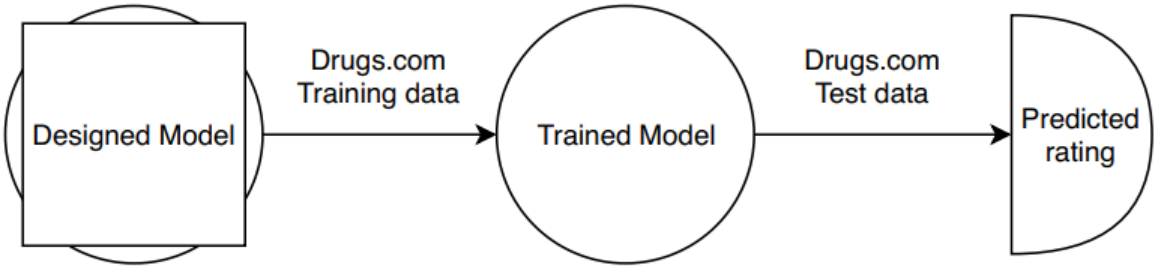


Figure 4.8: Experiment 1- Configuration

Train	Test	RMSE	MAE	R^2	Adjusted R^2	Pearson coeff r
Drugs.com	Drugs.com	20.52%	8.47%	0.363	0.363	0.84

Table 4.1: Experiment 1- Results

RMSE is the square distance measure of the difference between actual and predicted drug ratings. 20.52% indicates a strong closeness between those values.

MAE of 8.47% is a very small value indicating a good performance from the model. It indicates the average of the differences between the actual and predicted values.

The values of R^2 and adjusted R^2 are the same here since there is only one independent variable. Here the first column is responsible for 36.3% variance of expected value around the line of fix.

The correlation coefficient of 0.84 indicates a strong positive correlation between the predicted and actual values.

Overall, the regression done by the model should be reported as successful.

4.5.2 Experiment 2: Cross- Condition

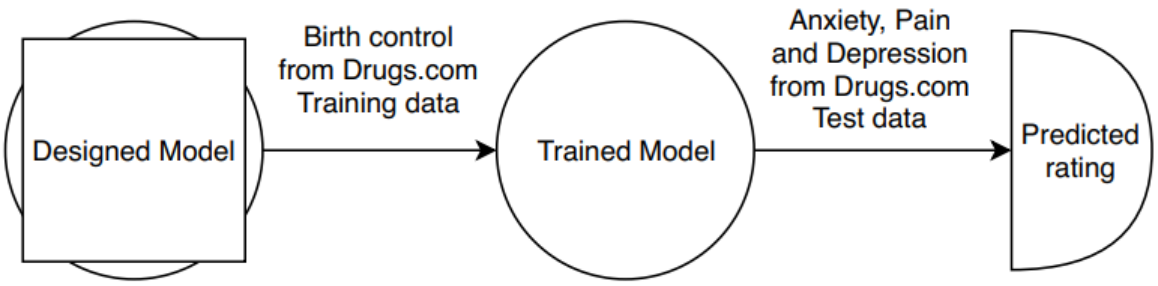


Figure 4.9: Experiment 2- Configuration

In this second experiment *Birth control* data is used for training and *Anxiety, Pain* and *Depression* for testing as evident from figure 4.9. The results are calculated the same as in the first experiment and are displayed in table 4.2.

The result values are similar for all the three parts of this experiment with that of the previous experiment. This could be since the training and testing data used in this experiment are subsets of the training data used in the previous experiment. The experiment can be said to be successful and the model should be reported to have acceptable levels of migration

capability between different conditions within the same dataset. For the same dataset, the premises of data collection will be the same, that is what questions are asked and what exact points were asked by the interviewer to the user. All the reviews will be formulated in a way the interviewer needs it. This should explain why the model can be easily migrated across different conditions.

Train	**Test**	**RMSE**	**MAE**	**R^2**	**Adjusted R^2**	**Pearson coeff r**
Second Experiment: Part One						
Birth Control	Anxiety	23.2%	10.52%	0.301	0.301	0.625
Second Experiment: Part Two						
Birth Control	Pain	23.3%	10.4%	0.311	0.311	0.47
Second Experiment: Part Three						
Birth Control	Depression	23.2%	10.56%	0.327	0.327	0.59

Table 4.2: Experiment 2- Results

4.5.3 Experiment 3: Cross- Source

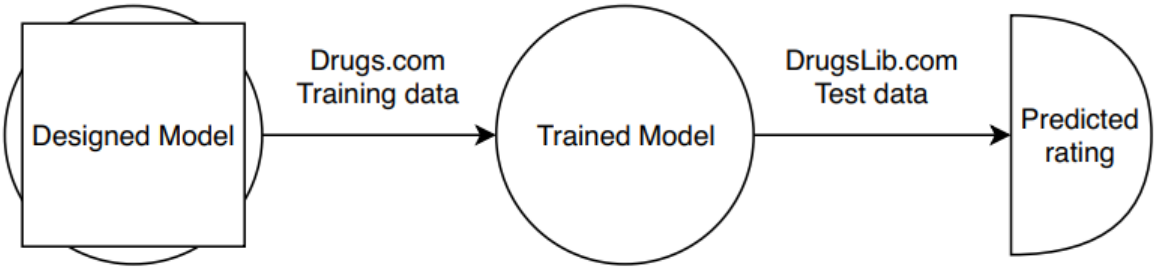


Figure 4.10: Experiment 3- Configuration

The results for the experiment three setup (from figure 4.10) can be seen in table 4.3. The training and test datasets are taken from two different sources. Unlike the first two experiments the R^2 value is low at 0.178. This explains that the model is able to explain only 17.8 % variability of the dependent variable in the test data. The reason could be the environment

difference between the collection of the two datasets even though they belong to the same pharmaceutical field. Hence, it is to be reported that the performance of the model across different sources is averagely acceptable.

Train	Test	RMSE	MAE	R^2	Adjusted R^2	Pearson coeff r
Drugs.com	DrugsLib.com	26.2%	14.2%	0.178	0.178	0.42

Table 4.3: Experiment 3- Results

These obtained results will be evaluated in the next chapter along with validation of the hypotheses.

4.6 Conclusion

This chapter dealt with the physical implementation of all the steps described in chapter 3. The coding and processing were done using Jupyter notebook with python language.

The first section listed all the processes undertaken to analyse the data, and peculiar behaviours are plotted. Total word count in the reviews column is assessed and plotted. there are many outliers indicating reviews with word length very higher than the average in the corpus. Inspection on the rating field showed an imbalance in the representation of each of the ten ratings in the dataset. The disease conditions with the most reviews are found out using a word cloud plot and are *BirthControl, Anxiety, Depression* and *Pain*.

Section 4.3 discussed the implementation stages of the data preparation algorithms. These include selection, replacing, spelling correction, abbreviation expansion, stopwords removal etc. Tokenization of text data and encoding of ratings are also done in this section.

Next section, 4.4, showed two architectures tried to implement the model design and explained why the second one is selected. It is the physical appearance of the model or shows

how the model components are connected.

Section 4.5 depicted the obtained results of all the three experiments and their corresponding setups. Experiment 1 proved that the designed model architecture is good for predicting ratings from reviews. The three parts in experiment 2 showed that the model can be easily migrated across different disease conditions, provided they are from the same dataset. In experiment 3, the results were only satisfactory, indicating the model is not that capable to run on data from a different source.

Next chapter is solely for evaluation of the results obtained in this chapter and for discussing the strengths and weaknesses of this research setup.

Chapter 5

EVALUATION AND DISCUSSION

5.1 Introduction

This chapter is dedicated to evaluation, analysis and discussion of the results obtained in the last chapter. The last chapter discussed what the results are in case of the individual experiments, but here they are analysed on a general basis.

The first section compares the results of all the experiments and comments on why the model behaved such in general as compared to the in-domain experiment. The experiment with the best metrics is also mentioned.

The research hypothesis defined at the beginning of this research work is examined in the next section to conclude whether to accept or reject the null hypotheses. The work is compared with the original research which formulated these datasets.

Section 5.4 deeply discusses the advantages and disadvantages of this research work and the setup. It also lists the issues faced while doing the work and mentions what were done to overcome those issues.

5.2 Evaluation of Results

Parameter	**In-Domain**	**Cross- Condition**			**Cross-Source**
		BirthControl/ Anxiety	*BirthControl/ Depression*	*BirthControl/ Pain*	
RMSE	20.52%	23.2%	23.2%	23.3%	26.2%
MAE	8.47%	10.52%	10.56%	10.4%	14.2%
R^2	0.363	0.301	0.327	0.311	0.178
Adjusted R^2	0.363	0.301	0.327	0.311	0.178
Pearson's r	0.84	0.625	0.59	0.47	0.42

Table 5.1: Evaluation of the Experiments

Table 5.1 shows the performance measures of all the experiments in this research work. It can be evidently seen that the metrics are best for the first experiment that is in-domain. This can be clearly attributed to the proper representation or sampling of the test data and the train data which are taken from the same dataset and belong to the same domain.

Cross-condition analysis shows equivalent performance as experiment 1 and hence it can be stated that model adheres strongly to the domain, when the training and test data are collected together or are subsets of a super collection.

The slight low values of performance in the case of experiment 3 indicate the model cannot be migrated to other data sources with the same efficiency.

Evaluation of the performance metrics clearly proves that, even though the model is efficient, there are still possibilities to improve the working which are addressed in section 5.4.

5.3 Research Hypothesis

The null hypothesis was defined in chapter 3 as,

- H_0: The efficiency of information extraction from drug reviews **can not be improved** if drug review quantification, using regional CNN-LSTM model and Natural Language Processing (NLP), is implemented to extract rating on a 10-point scale, as compared to using machine learning models alone which had a Cohen's kappa of 83.99 for In-domain analysis.

The requirement or goal of this research work was to attain a Cohen's kappa value of 83.99 or greater by using deep learning techniques in converting drug reviews into drug ratings. That benchmark of 83.99 was attained by Kallumadi et al. (2018) using machine learning techniques like logistic regression on the same drug review datasets. Cohen's kappa (Cohen, 1960) is a measure of the interrater agreement, which is used in the original paper, where the user given rating and the computer predicted ratings are considered to be two ratings generated by two sources for a set of drugs. Table 5.2 and figure 5.1 shows the performance metrics specific to both this research and the one compared to.

Performance Metric	**Referred Work**		**This Work**	
	In-Domain	Cross- Source	In-Domain	cross-Source
R^2	-	-	0.363	0.178
Accuracy	92.24	75.29	-	-
Cohen's Kappa	83.99	48.08	65.07	49.3

Table 5.2: Comparing this Study with Existing Standard

It can be very clearly established that this research work did not attain the expected kappa value of 83.99 for In-domain and instead got 65.07. This means that the null hypothesis is to be accepted and the alternate hypothesis should be rejected. This fallback can be attributed to many of the limitations faced during this research work, which are listed in the next section. Also, there was overfitting of the model to the training data which was evident from

the fact that after certain epochs the validation loss started increasing and finally crossed the training loss. The model training was stopped at the epoch where the two losses crossed, to ensure that the model is not over-fitted beyond a point, since, the same model is needed to be used for the cross-source experiment. This step caused an interesting outcome for experiment three which is explained below.

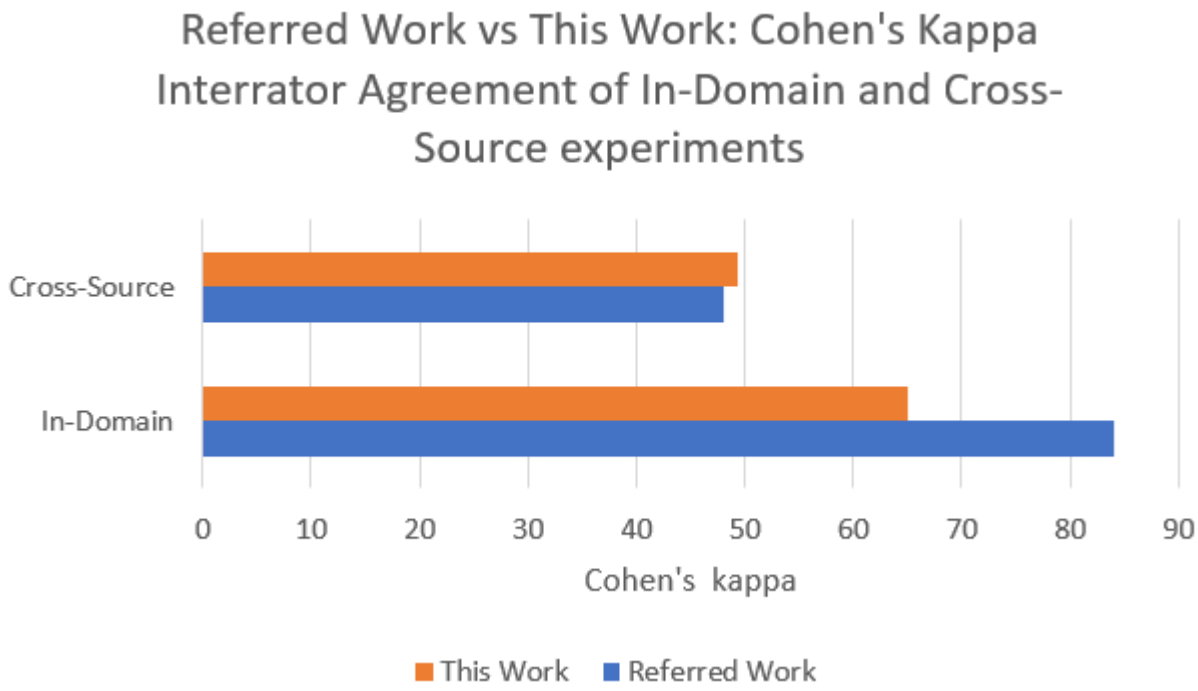


Figure 5.1: Bar Graph Comparing the kappa values: In-Domain and Cross-Source

Even though the kappa value for the in-domain experiment is very low compared to the original work, the kappa value of the cross-source experiment is a bit higher for this study, which is evident from the graph. This points out that the model designed in the original work might be over-fitted to the training data more, compared to this study, which could also attribute to the high kappa values in the first experiment. The overfitting property of the model calls for domain-specific data preparation, which needs to be done in future to improve this work.

5.4 Discussion on Strengths and Weakness of Research

Section 5.2 discusses how well the designed deep learning model performed. The main strengths or advantages of this model are the availability of the two domain-specific drug review datasets. as already mentioned, the model works well for similar or uniform data and not that good for heterogeneous datasets.

This was a typical case of a regression application, and unlike machine learning regression, deep learning models tend to wrap around the generalized domain of the data and not the particular data itself.

The limitations that pulled down the performance of the model in this research work are discussed in detail in chapter 1, section 5. There are many weak points for this experimentation setup, including problems faced during implementation, which are listed below:

- The size of the datasets for training is limited which tends to overfit the model towards the training sample. This was the case for all the experiments. After a certain epoch, the validation loss will start increasing and goes against the trend of the training loss. This forced to discontinue further running of epochs. If the overfitting was not there, more epochs could have been possible, which could have given more accuracy. Especially in experiment 2, since subsets of the original data are used, they are further small in size.

- Data pre-processing is done in a general manner and not domain-specific. Domain-specific vectorization of text data, that is using vectors pre-trained in the pharmaceutical field in this case, will help to improve the efficiency of the model.

- Initial approach to remove stopwords from the text data proved to be of negative impact since it removed rating-affecting words like the negation words. Hence, stopwords removal had to be repeated carefully with a more specific removal bulk.

- Since regional-CNN was to be implemented, different model architectures had to be tried out before finalizing the better one.

5.5 Conclusion

The main content of this chapter was a discussion on the results obtained for each of the three experiments.

Section 5.2 compared the experiment performance metrics of all the experiments and concluded that the behaviour of the model can be explained based on the first experiment in adherence to domain-specific data.

The next section dealt with recalling the research hypothesis defined at the beginning of the report. The Cohen's kappa value obtained for the in-domain experiment is less than that of the original study and hence, it was concluded that the null hypothesis should be accepted and the alternate to be rejected. Even though the research was concluded to be a failure, some important pattern was discovered by looking at the kappa values of the cross-source experiment.

The last section in this chapter was used for listing the positives and negatives of this research work. A list of all the shortcomings and problems faced during the study is also mentioned.

This research work is concluded in the next and final chapter.

Chapter 6

CONCLUSION

6.1 Introduction

This chapter is used to wrap up all the work done in this experimental study. It is a short summary of every step done, every decision taken and every discussion done during the course of this research.

The second section in this chapter will show a brief account of everything done from the beginning of the proposal stage to the final results. Section 6.3 will be a look back on the defined problem, that this work is based on. It will also show the research question.

In 6.4, a brief summary of all the design steps, implementation and evaluation of results will be given. all the stages in between including data understanding, preparation, model designing, implementation architecture, result metrics etc are discussed.

The next section will give a list of contributions this work gave to the existing literature along with how it will impact the business domain. The last section will be focused on listing the important steps to be undertaken to improve this work in the future along with recommendations for future researchers interested in the topic.

6.2 Research Overview

This research is done to evaluate the possibilities of deep learning models in converting text into a situation based numeric equivalent or in other words, in quantifying textual data. Two suggested numeric equivalents are two-dimensional sentiment coordinates and then, user rating, which is chosen for this work. The work is further fine-tuned to the pharmaceutical drugs domain. The requirement is first established, which is called the problem definition. Then most of the relevant literature and related works were found out and studied. All the existing approaches to tackle the problem are compared and their advantages and disadvantages were recorded. One work was selected as the best in the literature.

A deep understanding of different algorithms and technologies associated with machine learning, data analytics and deep learning was acquired by intensive research. A deep learning approach is selected and proposed that it will achieve more success than the current best work. This led to the declaration of the hypothesis and the research question.

The selected deep learning approach is converted into the real world using proper design and architecture. Each step to be done for the research work was framed adequately and implemented. The results calculated for each experiment were evaluated to get a total idea of how well the created model works.

The final outputs were compared with the original hypothesis to decide whether to accept or reject the null hypothesis. Limitations faced during the research work are listed along with further plans and ideas on how to improve this work in the future.

Whatever done during the entire period of the research process is documented and combined into this record in parallel to the work.

6.3 Problem Definition

The research is designed and implemented to quantify textual input into user ratings. The particular domain selected is pharmaceutical drugs and their reviews. There are two datasets extracted from two famous drug-related websites, which are already being worked on by other researchers. There are existing machine learning equivalents. Hence, the quest was to find out alternate methodologies that might be more successful than the existing standard. Thus the problem was defined as whether deep learning models will give a better result in converting text to rating than machine learning regression models.

A research question was also developed which states: **"Can Text quantification techniques, particularly CNN- LSTM models and natural language processing (NLP), be used for drug review regression into a 10-point scale rating and can such models be migrated across different health conditions and different drug review data sources?"**

6.4 Design/Experimentation, Evaluation and Results

The TDSP life-cycle of data science is selected for carrying out this research. The business understanding of the problem definition is done through an analysis of pharmaceutical domain ad campaigning and success stories. The research methods are set to be quantitative and inductive. Python is used to code the work in Jupyter notebook environment.

Data was already available from two sources and were stored in a public repository. Data understanding phase dealt with digging into the datasets to discover patterns and relationship that can either help or pull back the research work. Any such relations found were recorded and worked on to understand how the data behaves in a different environment.

Next was the data pre-processing stage, where the behaviours and features of the datasets are either boosted or negatively weighted based on how they will affect the rest of the work. Data cleaning is also done at this stage along with text data manipulation steps including,

stopwords removal, punctuation removal, tokenization, stemming, lemmatization, spelling correction etc.

The model was designed to reflect the best output from the dataset. The designed model is converted into a deep learning architecture and is trained. Three experiments were suggested where the only difference is the training and test data used and the model architecture remained the same. They are the in-domain, cross-condition and cross-source cases. The last two experiments were used to define the migration capability of the model across different domains and data sources.

The results of three experiments were tabulated and analysed. It was proven that the model works best in the same domain and for the same dataset. overall the model performance is considered to be average.

The strengths of the completed research work were discussed. All the limitations and issues faced are listed down so that the model architecture and performance can be improved in the future.

6.5 Contributions and Impact

The existing standards in quantifying reviews is by machine learning methods. Only a very few researchers used deep learning models and almost none has used regional CNN and LSTM for regression. Additionally, instead of the generally considered sentiment polarity, this research work proposed sentiment dimensionality. All these should contribute to the existing literature.

Even though the research hypothesis did not succeed, the model designed was proved to be strong with respect to the domain and can be made to work more efficiently than the other models by doing domain based embedding. Hence it can easily influence the pharmaceutical businesses to develop much more efficient models using this as a base.

All the research work, done in the pharmaceutical domain, and in drug review to rating conversion are recorded in this report. These will be useful for future researchers.

6.6 Future Work and Recommendations

There are plenty of opportunities to improve this research work in the future. Most of it includes correcting the shortfalls listed in the previous chapter. Especially, since this work did not succeed, a deep analysis should be done on how to improve the efficiency of the same model. Some of the steps include incorporating more data instances which will reduce overfitting of the model, then employing domain-centric data preparation methods.

Data augmentation is an important recommendation for the future of this study. Data augmentation is usually done when the amount of data is not good enough. In principle, it is the process of creating duplicate instances using synonyms in the case of textual data.

Another important step would be to include domain adapted word embedding technique which is proven to be more efficient in text analysis. GloVe embedding can be re-trained to include new words. Generic embedding use large lexicons while domain adapted word embedding use only words or contents related to a particular domain.

Regional CNN is used for this model. It should be checked if a single CNN architecture could perform better with an LSTM than the tower of CNNs implemented here. Possibilities of sentence vectorization instead of word vectorization should be studied.

In future, when datasets are collected, care should be given to improve the balance between the rating values within the data. For all the datasets used here, there were very large differences between instances present for each rating in 1-10.

Instead of spending more hours on sentiment polarity, sentiment dimensions must be ex-

plored in deep to improve sentiment quantification efficiency.

Since more than one neural network algorithm is combined in the model architecture, it is always recommended to use system with enough RAM and GPU because each epoch will take minutes to run and perfect model fit might take nearly thousands of epochs.

With these recommendations considered, it can be hoped that the final output of the designed model in this study can be improved to become satisfactory for business application.

6.7 Conclusion

Chapter 6 is used to conclude the research work in this report. It mainly recalls the initial arguments and proposals and compares them with the obtained outputs.

Section 6.2 gives an overview of the research done. It is a brief description of whatever done during the course of this study. Every step from the planning stage to the discussion on the final results is stated chronologically.

Problem definition was the most important step during the initial stages of any research work. In section 6.3, the problem definition of this research is summarized and along with it the research question is also included.

Everything after problem definition until the evaluation of results is summarized in section 6.4. It starts with the data science life cycle that is used and then states the research methodologies which are quantitative and inductive. Data understanding and pre-processing phases were explained in brief including the names of the algorithms used to clean and process the data. Next, the model design and implementation stages were discussed along with the three experiments that are part of the study. Then the results were discussed and the strength and weaknesses of the model and how it can be improved.

Section 6.5 shows how this research work will contribute to the existing literature and also the positive sides of the study. It also discusses the impact of the study in the pharmaceutical business.

The last section in this chapter concentrated more on how this study can be improved in the future and what recommendations the researcher has to give for future data scientists who are interested in the problem and the domain. More stress is given on how to work around or prevent the limitations faced during this study which affected the final performance.

References

Aggarwal, C. (2007). Data streams: Models and algorithms. In *Springer* (p. 33-59).

Aggarwal, C., & Zhai, C. (2012). An introduction to text mining. *Mining Text Data*, 14-23. doi: 10.1007/978-1-4614-3223-4_1

Al-Moslmi, T., Omar, N., Abdullah, S., & Albared, M. (2017). Approaches to crossdomain sentiment analysis: A systematic literature review. In *Ieee access 5* (p. 16173–16192).

Artstein, R., & Poesio, M. (2008). Inter-coder agreement for computational linguistics. In *Association for computational linguistics* (Vol. 34, p. 555-596).

Ayodele, T. (2010). Types of machine learning algorithms. In *New advances in machine learning* (p. 19-48).

Azevedo, A., & Santos, M. (2008). Kdd, semma and crisp-dm: A parallel overview. In *Iadis european conference on data mining 2008, amsterdam* (Vol. 63, p. 182-185).

Bollegala, D., Mu, T., & Goulermas, J. (2016). Cross-domain sentiment classification using sentiment sensitive embeddings. In *Ieee trans. knowl. data eng.* (Vol. 28, p. 398–410).

Cavalcanti, D., & Prudêncio, R. (2017). Aspect-based opinion mining in drug reviews. In *Progress in artifcial intelligence, eugénio oliveira, joão gama, zita vale, and henrique*

lopes cardoso (eds.). springer international publishing, cham (p. 815–827).

Cohen, J. (1960). A coefcient of agreement for nominal scales. In *Educational and psychological measurement* (Vol. 20, p. 37-46).

Dashtipour, K., Poria, S., & Hussain, A. (2016). Multilingual sentiment analysis: State of the art and independent comparison of techniques. In *Cognitive computation* (Vol. 8, p. 1-15).

Deerwester, S., Dumais, S., Landauer, T., Furnas, G., & Harshman, R. (1990). Indexing by latent semantic analysis. In *Jasis* (Vol. 41, p. 391–407).

Denecke, K. (2015). Sentiment analysis from medical texts. In *Springer international publishing, cham* (p. 83–98).

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, *92*(438), 548–560.

Fayyad, U., Shapiro, P., & Smyth, P. (1996). From data mining to knowledge discovery: an overview. In *Advances in knowledge discovery and data mining* (Vol. 17, p. 38-54).

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classifcation: A deep learning approach. In *Proceedings of the 28th international conference on international conference on machine learning (icml'11). omnipress, usa* (p. 513–520).

Goeuriot, L., Na, J., Kyaing, W., Khoo, C., Chang, Y., Theng, Y., & Kim, J. (2012). Sentiment lexicons for health related opinion mining. In *Proceedings of the 2nd acm sighit international health informatics symposium (ihi '12). acm, new york, ny, usa* (p. 219–226).

Gopalakrishnan, V., & Ramaswamy, C. (2017, February). Patient opinion mining to analyze drugs satisfaction using supervised learning. In *Journal of applied research and technology* (Vol. 15, p. 311 – 319).

Gräßer, F., Beckert, S., Küster, D., Abraham, S., Malberg, H., Schmitt, J., & Zaunseder, S. (2017, August). Neighborhood-based collaborative filtering for therapy decision support. In *Proceedings of the 2nd international workshop on health recommender systems co-located with the 11th international conference on recommender systems italy* (p. 22–26).

Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning: Data mining, inference and prediction. In *Springer series in statistics* (p. 1-764).

Indolia, S., KumarGoswami, A., Mishra, S., & Asopa, P. (2018). Conceptual understanding of convolutional neural network- a deep learning approach. *Procedia Computer Science*, *132*, 679-688. doi: https://doi.org/10.1016/j.procs.2018.05.069

İrsoy, O., & Cardie, C. (2014, October). Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 720–728). Doha, Qatar: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/D14-1080` doi: 10.3115/v1/D14-1080

Jiang, D., Luo, X., Xuan, J., & Xu, Z. (2016). Sentiment computing for the news event based on the social media big data. In *Ieee access* (Vol. 5, p. 2373–2382).

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014, June). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 655–665). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/P14-1062` doi: 10.3115/v1/P14-1062

Kallumadi, S., Gräßer, F., Malberg, H., & Zaunseder, S. (2018). Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 international conference on digital health (dh '18)* (p. 121-125).

Kim, Y. (2014, October). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1746–1751). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D14-1181 doi: 10.3115/v1/D14-1181

Korkontzelos, I., Nikfarjam, A., Shardlow, M., Sarker, A., Ananiadou, S., & Gonzalez, G. (2016, August). Analysis of the eect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. In *Journal of biomedical informatics 62* (p. 148–158).

Kwale, F. M., Wagacha, P. W., & Mwaura, A. (2016). A text clustering comparison methodology. In *International journal of computer applications* (p. 11-19). doi: 10.5120/ijca2016909515

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. Retrieved from http://www.jstor.org/stable/2529310

Leaman, R., Wojtulewicz, L., Sullivan, R., Skariah, A., Yang, J., & Gonzalez, G. (2010). Towards internet-age pharmacovigilance: Extracting adverse drug reactions from user posts to health-related social networks. In *Proceedings of the 2010 workshop on biomedical natural language processing (bionlp '10). association for computational linguistics, stroudsburg, pa, usa* (p. 117–125).

Liu, B. (2012). Sentiment analysis and opinion mining. In *Morgan claypool publishers* (p. 29-35).

Liu, P., Joty, S., & Meng, H. (2015, September). Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1433–1443). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/D15-1168` doi: 10.18653/v1/D15-1168

Liu, Z., Liu, S., Liu, L., Sun, J., Peng, X., & Wang, T. (2016). Sentiment recognition of online course reviews using multi-swarm optimization-based selected features. In *Neurocomputing* (Vol. 185, p. 11–20).

Malandrakis, N., Potamianos, A., Iosif, E., & Narayanan, S. (2011, 01). Kernel models for affective lexicon creation. In (p. 2977-2980).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In (p. 3100-3111).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In (p. 3111-3119).

Miner, G. (2012). Practical text mining and statistical analysis for non-structured text data applications. In (p. 391–407). doi: https://doi.org/10.1016/B978-0-12-386979-1.03001-2

Mishra, A., Malviya, A., & Aggarwal, S. (2015). Towards automatic pharmacovigilance: Analysing patient reviews and sentiment on oncological drugs. In *Ieee international conference on data mining workshop (icdmw)* (p. 1402–1409).

Na, J.-C., & Kyaing, W. (2015, 03). Sentiment analysis of user-generated content on drug review websites. *Journal of Information Science Theory and Practice*, *3*, 6-23. doi: 10.1633/JISTaP.2015.3.1.1

Nikfarjam, A., & Gonzalez, G. (2011). Pattern mining for extraction of mentions of adverse drug reactions from user comments. In *Amia annual symposium proceedings* (p. 1019–1026).

Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., & Gonzalez, G. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. In *Journal of the american medical informatics association* (Vol. 22, p. 671-681).

Paltoglou, G., Theunis, M., Kappas, A., & Thelwall, M. (2013, 01). Predicting emotional responses to long informal text. *Affective Computing, IEEE Transactions on*, *4*, 106-115. doi: 10.1109/T-AFFC.2012.26

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)* (pp. 79–86). Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W02-1011` doi: 10.3115/1118693.1118704

Piryani, R., Madhavi, D., & Singh, V. (2016). Analytical mapping of opinion mining and sentiment analysis research during 2000-2015. In *Information processing management* (Vol. 53, p. 122–150).

Russell, J. (1980, 12). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*, 1161-1178. doi: 10.1037/h0077714

Sarawagi, S. (2008). Information extraction. In *Foundations and trends in databases* (p. 261-377).

Sarawagi, S., & Cohen, W. W. (2004). Semi-markov conditional random fields for in-

formation extraction. In *Advances in neural information processing systems 17* (p. 1185-1192).

Sarker, A., Ginn, R., Nikfarjam, A., O'Connor, K., Smith, K., Jayaraman, S., ... Gonzalez, G. (2015). Utilizing social media data for pharmacovigilance: A review. In *Journal of biomedical informatics* (Vol. 54, p. 202–212).

Seifzadeh, S., Farahat, A. K., & Kamel, M. S. (2015). Short-text clustering using statistical semantics. In *Proceedings of the 24th international conference on world wide web* (p. 805-810).

Shearer, C. (2000). The crisp-dm model: the new blueprint for data mining. In *Journal of data warehousing* (Vol. 5, p. 13-22).

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417–424). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `https://doi.org/10.3115/1073083.1073153` doi: 10.3115/1073083.1073153

Voorhees, E. (2002). The philosophy of information retrieval evaluation. In *Proceedings of the special interest group on information retrieval* (p. 355–370).

Wang, J., Yu, L., Lai, K., & Zhang, X. (2016). Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th annual meeting of the association for computational linguistics* (Vol. 2, p. 225–230).

Wang, X., Liu, Y., Sun, C., Wang, B., & Wang, X. (2015, July). Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th*

international joint conference on natural language processing (volume 1: Long papers) (pp. 1343–1353). Beijing, China: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P15-1130 doi: 10.3115/v1/P15-1130

Wu, C., Wu, F., Huang, Y., Wu, S., & Yuan, Z. (2017). Thungn at ijcnlp-2017 task 2: Dimensional sentiment analysis for chinese phrases with deep lstm. In *Ijcnlp* (p. 47-52).

Wu, F., Huang, Y., & Song, Y. (2016). Structured microblog sentiment classification via social context regularization. In *Neurocomputing* (Vol. 175, p. 599–609).

Xia, L., Gentile, A., Munro, J., & Iria, J. (2009). Improving patient opinion mining through multi-step classifcation. In *Proceedings of the 12th international conference on text, speech and dialogue (tsd '09). springer-verlag, berlin, heidelberg* (p. 70–76).

Yadav, S., Ekbal, A., Saha, S., & Bhattacharyya, P. (2018). Medical sentiment analysis using social media: Towards building a patient assisted system. In *Proceedings of the eleventh international conference on language resources and evaluation (lrec-2018)* (p. 2790–2797).

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018, Aug 01). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629. Retrieved from https://doi.org/10.1007/s13244-018-0639-9 doi: 10.1007/s13244-018-0639-9

Yin, G., Zhang, Q., & Li, Y. (2014). Effects of emotional valence and arousal on consumer perceptions of online review helpfulness. In *Twentieth americas conference on information systems, savannah* (p. 7-9).

Appendix A

Snippets of Code

A.1 Data Pre-processing

```
REPLACE_BY_SPACE_RE = re.compile('[/(){}\[\]\|@,;]')
BAD_SYMBOLS_RE = re.compile('[^0-9a-z #+_]')
STOPWORDS = set()

def clean_text(text):
    """
        text: a string

        return: modified initial string
    """
    text = text.lower()
    text = REPLACE_BY_SPACE_RE.sub(' ', text)
    text = BAD_SYMBOLS_RE.sub('', text)
    text = text.replace('x', '')
    text = ' '.join(word for word in text.split() if word not in STOPWORDS)
    return text
```

Figure A.1: Data Cleaning code

```
[ ]  drug_lst = list(set(train['drugName']))
     drug_lst = [i.lower() for i in drug_lst ]
```

```
[ ]  def replace_entities(raw):
         for eachword in raw.split(" "):
             if eachword.lower() in drug_lst:
                 raw = raw.replace(eachword, 'drugName')
             if eachword[-2:].lower() == 'mg':
                 raw = raw.replace(eachword, 'qty')
         return raw
```

Figure A.2: Replacing drug names and quantity with generic words

```
def do_spell_correction(org_raw):

    raw = re.sub('[^A-Za-z]+', ' ', org_raw)
    doc = nlp(raw)

    word_ints = {}
    for ent in doc.ents:
        word_ints[ent.text] = ent.label_

    tokens = raw.split(" ")
    misspelled = spell.unknown(tokens)
    for each in misspelled:

        if(len(each))<3:
            continue
        if each in word_ints.keys():
            if word_ints[each] in ["ORG", "GPE","PERSON"]:
                continue
        corrected_word = spell.correction(each)
        print(each, "corrected == > ",corrected_word)
        org_raw = org_raw.replace(each, corrected_word)
        del raw
    del doc
    del word_ints
    del misspelled
    del tokens
    return org_raw
```

Figure A.3: Spell correction and NER

```
abbs = pd.read_excel("./shortforms.xlsx")
```

```
abb_dict = dict(zip(abbs.abbrevation, abbs.expansion))
```

```
def replace_abbs(raw):
    for eachword in raw.split(" "):
        if eachword in abb_dict.keys():
            raw = raw.replace(eachword, abb_dict[eachword])
    return raw
```

Figure A.4: Abbreviation Expansion

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
def do_lemmatization(raw):
    for eachword in raw.split(" "):
        raw = raw.replace(eachword, lemmatizer.lemmatize(eachword))
    return raw
```

```
nltk.download('wordnet')
```

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
def do_stemming(raw):
    for eachword in raw.split(" "):
        raw = raw.replace(eachword, ps.stem(eachword))
    return raw
```

Figure A.5: Stemming and Lemmatization

A.2 Model Architecture

```
class DeepCnnLstm():
    def __init__(self, embedding_size, hidden_size, rnn_hidden_size, seq_len, filters=32, kernel_size=3,
                 strides=6):
        self.embedding_size = embedding_size
        self.hidden_size = hidden_size
        self.rnn_hidden_size = rnn_hidden_size
        self.filters = filters
        self.kernel_size = kernel_size
        self.input1, self.tower1 = self.create_cnn_tower(seq_len)
        self.input2, self.tower2 = self.create_cnn_tower(seq_len)
        self.input3, self.tower3 = self.create_cnn_tower(seq_len)
        self.input4, self.tower4 = self.create_cnn_tower(seq_len)
        self.input5, self.tower5 = self.create_cnn_tower(seq_len)
        self.joined = Concatenate()([self.tower1, self.tower2 , self.tower3, self.tower4 , self.tower5])
        self.lstm_out = LSTM(self.rnn_hidden_size, activation="tanh")(self.joined)
        self.outNeuron = Dense(1)(self.lstm_out)

    def create_cnn_tower(self, max_seq_len):
        input_layer = Input(shape=(max_seq_len, self.embedding_size))
        tower = Conv1D(filters=self.filters,kernel_size=self.kernel_size, border_mode='same',activation="relu")(input_layer)
        tower = MaxPooling1D(2)(tower)
        return input_layer, tower

    def create_deep_model(self):
        output= self.outNeuron
        self.model = Model(inputs=[self.input1, self.input2, self.input3, self.input4, self.input5], outputs=[output])
        self.model.compile(optimizer='Adam', loss='mse')

    def train(self, train_data, batch_size, epochs=3500):
        tensorboard = TensorBoard(log_dir="tf_logs/{}".format(time()))
        self.create_deep_model()
        print(self.model.summary())

        region1 = np.array(list(train_data.loc[:, "region1"]))
        region2 = np.array(list(train_data.loc[:, "region2"]))
        region3 = np.array(list(train_data.loc[:, "region3"]))
        region4 = np.array(list(train_data.loc[:, "region4"]))
        region5 = np.array(list(train_data.loc[:, "region5"]))

        self.train_inputs = [region1, region2, region3, region4, region5]
        self.train_outputs = train_data.loc[:, "rating"]

        self.history = self.model.fit(self.train_inputs,
                                      self.train_outputs,
                                      callbacks=[tensorboard],
                                      validation_split=0.05,
                                      batch_size=batch_size,
                                      epochs=epochs)
```

Figure A.6: Definition of model in Python

Appendix B

Valence and Arousal Lexicons

A	B	C	D	E	F
	Word	Valence Mean	Valence SD	Arousal Mean	Arousal SD
1	aardvark	6.26	2.21	2.41	1.4
2	abalone	5.3	1.59	2.65	1.9
3	abandon	2.84	1.54	3.73	2.43
4	abandonn	2.63	1.74	4.95	2.64
5	abbey	5.85	1.69	2.2	1.7
6	abdomen	5.43	1.75	3.68	2.23
7	abdomina	4.48	1.59	3.5	1.82
8	abduct	2.42	1.61	5.9	2.57
9	abduction	2.05	1.31	5.33	2.2
10	abide	5.52	1.75	3.26	2.22
11	abiding	5.57	1.75	3.59	2.26
12	ability	7	1.59	4.85	2.74
13	abject	4	1.29	3.94	2.41
14	ablaze	5.15	1.79	6.75	2.12
15	able	6.64	1.79	3.38	2.25
16	abnormal	3.53	1.22	4.48	2.29
17	abnormal	3.05	1.81	5	2.62
18	abode	5.28	1.27	2.9	1.89
19	abolish	3.84	1.54	4.18	2.07
20	abominab	4.05	1.23	5.45	2.44
21	abominati	2.5	1.65	5.9	2.59
22	abort	3.1	1.37	5.8	2.44

Ratings_Warriner_et_al

Figure B.1: Example of Valence Arousal Lexicons of English words