CONTENTS

Motto Page ii

PREFACE iii

ABSTRACT x

I Introduction 1
- 1.1 Research Background . . . 1
- 1.2 Research Problem . . . 5
- 1.3 Research Scope . . . 5
- 1.4 Research Objective . . . 5
- 1.5 Research Benefit . . . 6

II Literature Review 7

III Basic Theory 12
- 3.1 Artificial Neural Network . . . 12
- 3.2 Convolutional Neural Network . . . 12
  - 3.2.1 Convolution Layer . . . 13
  - 3.2.2 Non-linearity . . . 15
  - 3.2.3 Pooling Layer . . . 16
  - 3.2.4 Fully Connected Layer . . . 17
- 3.3 Neural Network Committee . . . 17
  - 3.3.1 Static Structures . . . 18
  - 3.3.2 Ensemble Averaging . . . 18
- 3.4 Text Representation . . . 19
  - 3.4.1 Word2Vec . . . 20
  - 3.4.2 GloVe . . . 21
  - 3.4.3 FastText . . . 22
- 3.5 Performance Evaluation . . . 22
  - 3.5.1 Accuracy . . . 23
  - 3.5.2 Precision . . . 23
  - 3.5.3 Recall . . . 23
  - 3.5.4 F1 score . . . 23

IV Research Methodology **24**
4.1 Research Description 24
4.2 Research Phases 25
4.2.1 Literature Study 25
4.2.2 System Analysis 25
4.2.3 System Design 26
4.2.4 Implementation 27
4.2.5 Evaluation 27
4.3 Convolutional Neural Network Ensemble Architecture 28
4.3.1 CNN for Natural Language Processing 28
4.3.2 CNN Ensemble Architecture 33
4.4 Evaluation Design 34

V Implementation **35**
5.1 Main Program Stages 35
5.2 Pre-trained Word Embedding Trimming 35
5.2.1 Dataset Loading and Cleaning 35
5.2.2 Tokenizing 37
5.2.3 Pre-trained Word Embedding Vectors Loading 38
5.2.4 Pre-trained Word Embedding Vectors Trimming and Saving . 38
5.3 Multiple CNN Training 39
5.3.1 Dataset Preprocessing Phase 40
5.3.2 Training Phase 41
5.4 CNN Ensemble 45
5.4.1 Model, Tokenizer, and Test Data Loading 45
5.4.2 Prediction and Evaluation 46

VI Results and Discussions **49**
6.0.1 Individual CNN Training Process 49
6.0.2 Evaluation Stage 52
6.0.3 Final Results 54

VIIConclusion **57**
7.1 Future Works 57

References **61**

A Full Implementation **62**

LIST OF TABLES

2.1 Literature Table 11

6.1 Average Performance Evaluation 52
6.2 Precision and Recall of both CNN with GloVe and FastText 53
6.3 Evaluation Accuracy of models 53
6.4 Evaluation F1-Score of models 54
6.5 Accuracy of CNN and CNN ensemble and its differences for 10 iterations 55

LIST OF FIGURES

1.1 A glimpse of deceptive opinion spam in the context of hotel review by Ott *et al.* (2011) . . . 3

3.1 Illustration by Nielsen (2019) of a basic ANN having five nodes in its input layers and one node in the output layer, also two hidden layers . 12
3.2 Illustration of a basic CNN architecture from Saha (2018) . . . 13
3.3 Convolution layer illustrations by Saha (2018) . . . 14
3.4 Stride in convolution depicted by Chen (2017) . . . 14
3.5 Image padding illustration by Saha (2018) . . . 15
3.6 Common types of nonlinearity function by Saha (2018) . . . 16
3.7 Illustration of implementing max pooling into an input from Saha (2018) . . . 16
3.8 Neural Network Committee architecture and training process as illustrated by Cireşan *et al.* (2011) . . . 18
3.9 Illustration of ensemble averaging by Bullinaria (2004) . . . 19
3.10 Illustration of Word2Vec algorithms by Mikolov *et al.* (2013b) . . . 21
3.11 Co-occurrence matrix for the sentence "the cat sat on the mat" with window size 1 by Ganegedara (2019) . . . 22

4.1 Glimpse of the published dataset made by Ott *et al.* (2013) . . . 24
4.2 Convolutional Neural Network architecture for Natural Language Processing by Kim (2014) . . . 28
4.3 Histogram of the number of words per review against the frequency . 29
4.4 CNN Implementation Architecture . . . 30
4.5 Proposed workflow of the research for both types of neural network to assemble the committee . . . 32
4.6 Illustration of CNN Ensemble Architecture . . . 33

6.1 Figures of the training phase of each models with GloVe word embedding . . . 50
6.2 Figures of the training phase of each models with FastText word embedding . . . 51
6.3 Paired t-test result from the R code . . . 56

Listings

5.1 Loading of the dataset . . . 35
5.2 Function to help clean the dataset . . . 37
5.3 Tokenizing the dataset and adding post padding . . . 38
5.4 Pre-trained word embedding convertion and loading . . . 38
5.5 Trimming of the pre-trained word embedding vectors . . . 39
5.6 Saving of the new trimmed word embedding vectors . . . 39
5.7 Label creation . . . 40
5.8 Dataset splitting . . . 41
5.9 Test data and tokenizer saving . . . 41
5.10 CNN architecture initialization function . . . 43
5.11 Model training . . . 44
5.12 CNN ensemble with 30 CNNs model loading . . . 45
5.13 CNN ensemble with 10 CNNs model loading . . . 45
5.14 Tokenizer and test data loading . . . 46
5.15 CNN Ensemble with 30 CNNs prediction process . . . 46
5.16 CNN Ensemble with 10 CNNs prediction process . . . 47
5.17 Individual CNN with GloVe and FastText prediction process . . . 47
5.18 Calculating the confusion matrix of each final model . . . 48
6.1 Declaring R array variables to contain both the accuracy of CNN and CNN ensemble . . . 55
6.2 Using built-in R function to compute the paired t-test of the data . . . 56
1.1 GloveEmbeddingTrimmer.ipynb . . . 62
1.2 MultipleCNNTrainer.ipynb . . . 65
1.3 CNNCommittee.ipynb . . . 73