TEXT AUGMENTATION FOR EMOTION CLASSIFICATION IN MICROBLOG TEXT USING SIMILARITY SCORING BASED ON NEURAL EMBEDDING MODELS

YONG KUAN SHYANG

UNIVERSITI SAINS MALAYSIA

2022

TEXT AUGMENTATION FOR EMOTION CLASSIFICATION IN MICROBLOG TEXT USING SIMILARITY SCORING BASED ON NEURAL EMBEDDING MODELS

by

YONG KUAN SHYANG

Thesis submitted in fulfilment of the requirements
for the degree of
Master of Science

August 2022

ACKNOWLEDGEMENT

I would like to express my uttermost gratitude to my advisor, Dr Jasy Liew Suet Yan, for her guidance throughout this dissertation. Without her relentless support and scientific knowledge, I would not be able to complete this dissertation on time. I am very grateful for her mentorship and encouragement to me on this long journey. A very special gratitude goes out to all the admin staff members from the School of Computer Sciences especially Ms Sheela Muniandy who assisted me the most on all the admin-related procedures to complete this dissertation. Moreover, I would like to appreciate all the staff members at the Institute of Postgraduate Studies (IPS) in providing me useful information during my candidature period and organizing meaningful research workshops to hone my research ability. Finally, I would like to thank my parents for always giving me unconditional support throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS

t	Threshold value
I	Fixed increment value

LIST OF ABBREVIATIONS

ET	EmoTweet-28
ET-train	EmoTweet-28 train set
ET-seed	EmoTweet-28 seed set
ET-test	EmoTweet-28 test set
DS	Distant supervision tweets set
BoW	Bag-of-Words
W2V	Word2Vec
GloVe	Global Vectors for Word Representation
USE	Universal Sentence Encoder
BiLSTM	Bi-directional Long Short-Term Memory
AUG	Augmented set
PRIMARY	Original proposed text augmentation strategy
CLUSTER	Clustering text augmentation strategy
MISCLASSIFIED	Misclassified text augmentation strategy

LIST OF APPENDICES

PENGEMBANGAN TEKS UNTUK KLASIFIKASI EMOSI DALAM TEKS MIKROBLOG DENGAN PEMARKAHAN PERSAMAAN BERDASARKAN MODEL PEMBENAMAN NEURAL

ABSTRAK

Pengelasan emosi boleh mendapat manfaat daripada suatu kumpulan data latihan yang besar. Namun, pengembangan korpus emosi secara manual banyak memerlukan tenaga kerja dan memakan masa. Selain itu, penyeliaan jarak jauh boleh digunakan untuk mengumpul sejumlah data latihan yang banyak dalam jangka masa yang pendek dengan hashtag perkataan emosi. Namun demikian, data yang dikumpul berkemungkinan mengandungi kehingaran yang berlebihan. Dalam penyelidikan ini, kami mencadangkan strategi augmentasi teks bagi mengembangkan saiz contoh positif untuk enam kategori emosi (kegembiraan, kemarahan, keterujaan, keterdesakan, kebosanan dan sikap tidak peduli) dalam EmoTweet-28 dengan menggunakan tweet yang dikumpul melalui penyeliaan jarak jauh (DS) yang bersamaan dengan contoh benih dalam EmoTweet-28 (ET-seed). Pendekatan pemarkahan serupa digunakan untuk mengira kosinus markah serupa antara setiap tweet DS dan semua tweet ET-seed dalam kategori emosi yang sama. Tujuh perwakilan vektor (USE, InferSent GloVe, InferSent fastText, Word2Vec, fastText, GloVe, and Bag-of-Words) telah diujikaji bagi mewakili tweet dalam pendekatan pemarkahan serupa. Tweet DS dengan markah serupa yang tinggi akan dipilih sebagai contoh augmentasi dan dianotasikan dengan label emosi. Pemilihan tweet DS dibahagikan kepada dua kategori, iaitu pemilihan berasaskan ambang dan pemilihan berasaskan tokokan tetap. Tambahan pula, kami juga telah mengubahsuai strategi teks augmentasi dengan meminda set benih yang digunakan untuk pemarkahan serupa melalui strategi pengelompokan dan

pengklasifikasian silap. Kesemua set augmentasi telah dinilai menerusi latihan menggunakan pengkelas rangkaian neural mendalam secara berasingan untuk membezakan kehadiran atau ketidakhadiran emosi yang tertentu bagi setiap tweet dari set uji. Keputusan menunjukkan USE adalah pendekatan yang lebih berkesan untuk augmentasi berasaskan ambang, InferSent GloVe merupakan pendekatan yang paling stabil untuk augmentasi berasaskan tokokan tetap, manakala strategi pengklasifikasian silap adalah lebih berkesan dalam mengaugmentasikan sampel latihan yang lebih relevan. Sumbangan utama penyelidikan ini termasuk mencadangkan dan menilai strategi augmentasi teks menggunakan pendekatan pemarkahan serupa untuk meningkatkan jumlah data latihan yang diperlukan secara efisien untuk pengkelas emosi mempelajari cara pembezaan emosi yang berlainan dengan lebih baik.

TEXT AUGMENTATION FOR EMOTION CLASSIFICATION IN MICROBLOG TEXT USING SIMILARITY SCORING BASED ON NEURAL EMBEDDING MODELS

ABSTRACT

Emotion classification can benefit from a larger pool of training data but manually expanding the emotion corpus is labour-intensive and time-consuming. Distant supervision can be used to collect large amount of training data in a short period of time using emotion word hashtags, but the collected data may contain excessive noise. In this research, we proposed a text augmentation strategy to efficiently expand the size of positive examples for six emotion categories (happiness, anger, excitement, desperation, boredom and indifference) in EmoTweet-28 by exploiting tweets collected from distant supervision (DS) that are similar to the seed examples in EmoTweet-28 (ET-seed). Similarity scoring approach was used to compute to cosine similarity scores between each DS tweet and all ET-seed tweets under the same emotion category. Seven vector representations (USE, InferSent GloVe, InferSent fastText, Word2Vec, fastText, GloVe, and Bag-of-Words) were experimented to represent the tweets in the similarity scoring approach. DS tweets with high similarity scores were selected to become the augmented instances and annotated with emotion labels. The selection of DS tweets was divided into two categories which are threshold-based selection and fixed increment selection. In addition, we also modified the proposed text augmentation strategy by altering the seed sets used for similarity scoring using clustering and misclassified strategies. All augmented sets were evaluated by training a deep neural network classifier separately to distinguish between the presence or absence of specific emotion in tweets from the test set. The

results showed USE vector representation is the better for threshold-based augmentation, InferSent GloVe vector representation is the most stable for fixed increment augmentation and misclassified strategy is better at augmenting more relevant training instances. The main contribution for this research is to propose and evaluate a text augmentation strategy using similarity scoring approach to efficiently increase the amount of training data required for emotion classifiers to better learn how to distinguish between different emotions.

CHAPTER 1

INTRODUCTION

1.1 Overview

Twitter is a social media platform that provides microblog service for the users to exchange information with each other in a convenient and straightforward manner. The user can broadcast and share important information to their followers by posting a short text message called "tweet" within 280 characters. Moreover, Twitter allows users to search specific topic of tweets based own personal interest with relevant keyword hashtags. Hashtag is a keyword that starts with "#" symbol and followed by an acronym, single word, or concatenation of multiple words without spacing in between such as #tgif, #happy, #anger and #bigdata.

As users tend to express their thoughts and emotions freely on Twitter, it has become a rich data source to gain multiple insights on real life applications such as prediction of stock market trends (Bollen, Mao, & Zeng, 2011), fluctuation indicator in social and economic (Bollen, Mao, & Pepe, 2011), measurement of population's happiness level (Dodds & Danforth, 2010) and disaster management (Vo & Collier, 2013). However, it is impossible for human to identify and analyse millions of tweets generated in a day that may reveal interesting social, economic, and public health trends. Therefore, sentiment analysis and emotion classification are becoming increasingly important to harness these important signals and turn them into meaningful insights.

Sentiment analysis is the classification of the polarity on a given example to determine whether it expresses positive, negative, or neutral opinion. However, emotion classification is beyond sentiment analysis, which not only captures the sentiment value, but it can also recognize the emotional state expressed from a given

example based on a set of emotion categories such as Ekman's six basic emotions (Ekman, 1992) (i.e., joy, sadness, anger, fear, surprise, and disgust). Therefore, it is an undeniable fact that emotion classification can produce more crucial signals for identifying human expression than sentiment analysis when carrying out analysis on data collected from Twitter.

Besides, fine-grained emotion classification (Liew et al., 2016) on microblog text includes a wider range of emotion categories is an advancement of the regular coarse-grained emotion classification and it can detect more detailed insights expressed in the tweet by human such as admiration, amusement, excitement, fascination, gratitude, pride and surprise which are emotions closely related to the happiness. However, availability of quality fined-grained emotion classification corpora is limited because the curation of quality corpus is labour-intensive and time-consuming due to high demand of annotation effort. Without a doubt, insufficient of training data is the most common problem for every classification task which can lead to poor performance. Recently, adaption of deep learning models has shown promising results in various classification tasks, but it required large amounts of training data to be feasible. Therefore, the aim of this research is to identify an efficient solution to generate large amounts of training data for automatic emotion classification in the most economical and the least time-consuming approach. Text augmentation is a popular solution to address the data insufficient problem by generating more data similar to the original data in the existing corpus. There are various methods for augmenting new training data such as word substitution, backtranslation and text generation. In this research, we explored a semi-supervised text augmentation method that has yet to be thoroughly explored to generate more training data similar to the data in an existing

corpus for fine-grained emotion classification using similarity scoring based on neural embedding models.

1.2 Problem Statement

The development of reliable and robust automatic emotion classifier requires large quantities of training data. One common way to collect high quality training data is to use manual annotation to construct a gold standard corpus that would serve as the training and test data. However, manual annotation is labour-intensive and time-consuming because constructing a gold standard corpus requires a great deal of human effort to manually assign the emotion labels for newly collected data. For example, EmoTweet-28 (Liew et al., 2016) is a gold standard corpus consisting of 15,553 tweets annotated with 28 emotion categories which took around ten months to complete. Each tweet was carefully labelled by multiple annotators who had to first undergo training on capturing both implicit and explicit emotions expressed in the tweets. However, the positive examples for 28 emotion categories in EmoTweet-28 are not evenly distributed which will lead to poor performance in classification of certain emotion categories. Therefore, corpus expansion is needed to increase the number of positive examples for emotion classifiers to learn pattern from data more effectively. Corpus expansion can be carried out using the previous manual annotation method or text augmentation which is a method to expand the size of the current corpus by introducing modified or similar example to the existing data.

An alternative method to expand the EmoTweet-28 corpus is to utilize the distant supervision method to collect large amounts of self-labelled tweets using only emotion keyword hashtags as the emotion labels (Abdul-Mageed & Ungar, 2017; Mintz et al., 2009; Purver & Battersby, 2012). This method is fast and cheap since it

only uses emotion hashtags as query words to retrieve tweets from the Twitter API and it does not necessarily have to involve any human annotation. However, tweets collected using distant supervision could be noisy and potentially deteriorate the performance of the classifiers if the text processing on collected tweets is not done properly. Therefore, the motivation for this research is to identify a semi-supervised text augmentation strategy without intensive human interaction that can leverage the scale of distant supervision in expanding the EmoTweet-28 gold standard corpus while maintaining the quality of the corpus for model development.

We minimise noise from the tweets collected using distant supervision by discarding tweets that are less similar to the original (seed) tweets in EmoTweet-28 using similarity scoring based on neural embedding models. Furthermore, threshold-based and fixed increment selection are applied to determine the size of the tweets for the augmentation. Thus, excessive human intervention such as content validation via expert judgement can be avoided to guarantee the quality of the augmented tweets. In addition, three strategies for text augmentation are experimented based on the sampling of the seed samples for augmentation.

1.3 Research Objectives

i. To propose a text augmentation strategy with similarity scoring approach that can leverage tweets collected using distant supervision for expanding the number of training data in an emotion corpus.

ii. To evaluate the performance of similarity scoring approach using different pre-trained neural embedding models on augmenting new training data in different emotion categories.

iii. To identify which text augmentation strategy can efficiently expand the current fine-grained emotion corpus with new relevant training data while preserving the quality of the corpus.

1.4 Research Questions

Specifically, the study addresses the following three research questions:

- R1: How can new training data be augmented using similarity scoring approach to improve the quality of data collected from distant supervision method?
- R2: What is the effect of using similarity scoring approach to augment more relevant training data on the performance of deep learning model for emotion classification?
- R3: Which text augmentation strategy is better at augmenting more relevant training data?

1.5 Research Scope

This research includes two main datasets for conducting the text augmentation experiments with similarity scoring approach which are EmoTweet-28 and a collection of self-labelled tweets collected using distant supervision method from the beginning of December 2019 until the end of May 2020. The six emotion categories chosen for this research from the original 28 emotions in EmoTweet-28 are happiness, anger, excitement, boredom, desperation, and indifference. The six emotion categories are selected because we would like to observe the effect of the proposed text augmentation strategy on high and low-frequency positive seed examples in EmoTweet-28. All the

pre-trained neural embedding models used in this research for computing the similarity scores between the tweets are trained with a wide variety of English sentences in different context. The evaluation strategy for this research utilizes the bidirectional long short-term memory recurrent neural network model (BiLSTM) to perform binary classification on test sets derived from the original EmoTweet-28.

1.6 Thesis Outline

This section provides the organization of the thesis. Chapter 2 is the survey of the related works on the current text augmentation methods and emotion classification approaches. Chapter 3 explains the details of our three-phase methodology for data preparation, text augmentation and emotion classification. Chapter 4 presents the results and discussion from the conducted experiments. Finally, Chapter 5 presents the conclusion of the research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews and compares current text augmentation methods for various classification tasks. In addition, we also surveyed current approaches used for emotion classification in text. The first half of the chapter focuses on four different categories of text augmentation and the second half of the chapter focuses on various emotion classification approaches.

2.2 Text Augmentation

Text augmentation is a solution to overcome the problem of insufficient training data in machine learning by generating more training data based on existing corpus. The existing text augmentation methods can be classified into four main categories which are substitution, backtranslation, text generation and similarity comparison between text units.

2.2.1 Substitution

Substitution in text augmentation generally focuses on substituting words in a sentence with synonyms or words with similar meaning. Kobayashi (2018) augmented new training data by replacing each word in a sentence with words predicted by a Language Model (LM) using the context information around the original words. Bi-directional LSTM-RNN (pre-trained on WikiText-103 corpus) was used as the LM to calculate the word probability for a target word by encoding the words around the target word leftward and rightward. Then, the word probabilities were concatenated and fed into a feed-forward neural network to generate a probability distribution with multiple

words. Numerous new words with the highest probability were sampled from the probability distribution and used to replace the original target word. In addition, a label-conditional LM was introduced to prevent the replacement words from altering the original sentence label by concatenating the label of the original sentence with a hidden layer in the feed-forward network of the LM. LSTM-RNN and CNN models were used as classifiers to evaluate the performance of the three different text augmentation approaches which were synonym-based (1), contextual-based augmentation with (2) and without (3) label conditionals. The results showed that contextual-based augmentation with label conditional in the CNN model achieved the highest average accuracy of 0.782 on all datasets and outperformed other text augmentation approaches and LSTM models.

Wang & Yang (2015) augmented new training data for behavioural analysis on Twitter by replacing each word in a tweet using lexical and semantic word embeddings. The replacement words were determined by the k-nearest-neighbour word vectors from the target word vector using cosine similarity in the embedding space. There were three lexical embedding models trained using Word2Vec on 100 billion words from the Google News dataset, 51 million words from the Twitter dataset and 53 million words from the Urban Dictionary dataset. Also, a frame-semantic embedding was trained with 27 million frames on 3.8 million tweets. A logistic regression model was used to build a multi-class classifier to evaluate the performance of the data augmentation methods. The results showed the classifier with Urban Dictionary lexical and frame-semantic embeddings achieved the highest F1 score of 0.38 which outperformed other lexical embeddings and the baseline model without data augmentation.

Zhang et al. (2015) generated new data by substituting words with their synonyms from WordNet (English Thesaurus). Geometric distribution was used to

decide the number of words to be substituted from a text unit and which words to be chosen from a list of synonyms to substitute the target words. The performance of the text augmentation method was evaluated using a character-level convolution networks on text classification with various datasets. The results showed the model trained with the augmented dataset outperformed model trained without data augmentation on all datasets.

Kolomiyets et al. (2011) created new data for recognition of time expression by replacing temporal expression in given data with predicted words from Latent Words Language Model (LWLM) which was trained on 80 million words from Reuters news article corpus and synonyms from WordNet. The data augmentation was divided into four strategies. The first strategy substituted the phrasal head word with the synonyms predicted from LWLM. The second strategy substituted phrasal head word with the synonyms selected from WordNet. The third strategy replaced the phrasal head word with the largest intersection words between the synonyms predicted from LWLM and the synonym sets (synsets) from WordNet. The final strategy replaced the phrasal head word with the intersection words between LWLM synonyms and the synsets obtained from the third strategy. A logistic regression model was used as the classifier to evaluate the performance of the original dataset and augmented dataset by classifying the types of time expression. The model was trained on the official TempEval 2010 training set (53,450 tokens and 2117 annotated TIMEX3 tokens) and tested on three different corpora which were TempEval 2010 evaluation set (95,99 tokens and 269 annotated TIMEX3 tokens), news article from Reuter corpus (2,960 tokens and 240 annotated TIMEX3 tokens) and the Barack Obama article from Wikipedia (7,029 tokens and 512 annotated TIMEX3 tokens). The results showed the model achieved significant

improvement in the Reuter corpus and Barack Obama article from Wikipedia when trained with augmented data compared to model trained without augmented data.

Wei & Zou, (2019) introduced four methods to augment new data to enhance the performance of text classification. The first method chose a random number of words (not stop words) from a sentence and replaced each selected word with its synonyms from WordNet. The second method inserted new words which were synonyms from WordNet of random words in a sentence (not stop words) into somewhere between the sentence repeatedly. The third method swapped multiple random words in a sentence repeatedly. The final method deleted a random number of words in a sentence randomly. The random number of words changed for these methods were determined by the length of the original sentence and a fixed temperature parameter that specifies the fraction of the words changed in the sentence. CNN and RNN models were used as classifiers to evaluate the performance of the original and augmented datasets. Both classifiers were assigned five different classification tasks on five distinct datasets: Stanford Sentiment Treebank (SST-2), customer reviews, subjectivity/objectivity, question type and pro-con. The results showed the average accuracy for both models on five different classification tasks with augmented dataset is 88.6 which is higher than using the original dataset. Furthermore, Wei et al. (2021) further improved the data augmentation methods by introducing the curriculum data augmentation which is training a classifier on the original dataset initially and then retrain it again on multiple augmented datasets using the same data augmentation methods but with different temperature parameter values. Triplet loss model was used to build the classifiers for evaluating the performance of the curriculum data augmentation strategy using the HuffPost, FewRel, COVID-Q and Amazon product review datasets. The results showed that the triple loss classifiers with hard negative

mining selection strategy achieved average accuracy 34.7 on the four datasets when trained with the curriculum data augmentation strategy which is slightly higher than the average accuracy of 33.3 from the classifiers trained with the original data augmentation strategy.

Neculoiu et al. (2016) introduced three methods to augment new data for a job title normalization task. The first method created typo from a handmade job title taxonomy by randomly substituting and deleting characters across the taxonomy. The second method created new job titles by replacing words in each job title with synonyms from a synonym list created manually. The final method created a new job title by extracting relevant job description from external sources and appending it to the job title. The performance of the data augmentation methods was evaluated using a Siamese Recurrent Neural Network model on job title classification task. The results showed the model trained on data augmented using all the three methods achieved better accuracy than the model solely trained on base taxonomy.

Sharifirad et al. (2018) generated and augmented new training data for sexist tweets using knowledge graph such as ConceptNet and Wikidata. Three methods were used for data generation. The first method substituted every tokenized word from a tweet except the stop words with their "FromOf" and "IsA" relationship example words with the weight higher than 1.0. The second method substituted only the verb in a tweet with synonyms from ConceptNet. The third method substituted only the nouns found in a tweet by words with similar concept from ConceptNet. Two additional methods were proposed for data augmentation. The first method appended the top ten concepts relevant to a target tweet from ConceptNet at the end of the tweet based on the "IsA" relationship. The second method appended the most relevant concept from Wikidata that best described a target tweet together with all the concepts from the previous

method at the end of the tweet. Five multi-class classifiers were used to evaluate the performance of the data generation and augmentation methods: one-versus-all algorithm, Naive Bayes, SVM, LSTM-RNN and CNN. All the models classified the tweets into three different classes: indirect harassment, sexual harassment, physical harassment. The results showed that the first method from data generation which replaced all the words in a tweet achieved the highest accuracy of 0.98 on both LSTM-RNN and CNN models. On the other hand, the results showed that the data augmentation method with ConceptNet and Wikidata achieved the highest accuracy of 0.93 on the LSTM-RNN model.

Raiman & Miller (2017) introduced a data augmentation method with type awareness to enhance the results of question answering systems. The method used knowledge base such as Wikidata to identify all the named entities in a question-document set, and then replaced the named entities with new type-identical entities from Wikidata. First, nominal groups in the training data were identified using Part-Of-Speech Tagger (POS Tagger). Then, the entity types for each entity with identified nominal group were assigned with entities found in Wikidata. After that, new sentences were created by substituting each entity with new entities with the same type as the original entity in Wikidata. The performance of the data augmentation method was evaluated on a question answering model built using Bi-LSTM. The results showed the model trained with the augmented dataset achieved better exact match and F1 score than the model trained with the original dataset.

Vijayaraghavan et al. (2016) augmented new data for stance detection on tweets using Word2Vec to identify new similar terms to substitute original words or phrases in a tweet. First, a tweet was randomly chosen from the original dataset. Then, a random number of words or phrases based on a geometric distribution with p equals to 0.5 were

sampled from the chosen tweet. Next, cosine similarity was used to construct lists of similar terms for each word or phrase based on a pre-trained Word2Vec word embeddings trained on the original corpus, and each term must have a similarity score greater than 0.25. Finally, new augmented data was generated when original words or phrases from the tweets were substituted by the terms sampled from their own respective list using a second geometric distribution with q equals to 0.5. The performance of the data augmentation method was evaluated using a combination of both character and word-level CNN models on stance detection in tweets. The results showed the model achieved a macro F1 score of 0.635 in the test data from SemEval-2016 Task.

Giridhara et al. (2019) proposed five data augmentation methods on text to generate new modified sentence by augmenting the nouns, adjectives and adverbs of the original sentence. The first method focused on the replacement of the target words with high similarity words found in the pre-trained GloVe embedding model. The second method was the replacement of the target words with synonyms based on WordNet Synonym Dictionary. The third and fourth method were the replacement of the target words embeddings using interpolation and extrapolation in which a new word embedding is calculated based on the target word embedding and the centroid embedding from the top three nearest neighbours of the target word embedding. The final method was the augmentation of the target word embedding with random noise value. The deep learning algorithms used to train the classifier for evaluating the performance of the data augmentation methods were CNN and Attention-based Bidirectional LSTM. The datasets used for the classification tasks were specified for ignoring the direction aspect such as SemEval2010 with 10 relation classes and KPB37 with 19 relation classes. The results showed the classifiers trained with the augmented

data did not have any significant improvement or deterioration in terms of F1 score in comparison to the classifier trained with the original dataset.

Li et al. (2022) proposed the learning policy scheduling for text augmentation. The learning policy schedule augmentation is similar to the hyperparameter searching problem. The Population Based Training (PBT) algorithm was used to search an augmentation schedule that has the best policy. First, 16 PBT child models were randomly initialized and all hyperparameters were set to zero. Each child model was tested on the validation set and ranked based on the test error. Then, an exploit-and-explore process was performed in every two epochs to exploit the best hyperparameters and explore potential hyperparameters in the search space by perturbing and resampling the hyperparameters. The exploit process utilized Truncation Selection where the best four models' weights and hyperparameters were cloned to for the worst four models. In the policy search space, four hyperparameters needed to be searched in three augmentation operations. The augmentation operations included replacing words in the sentence with placeholder, surrounding, and random words, swapping order of the words in the sentence and adding random and surrounding words into the sentence. The four hyperparameters in the augmentation operation included probability to conduct augmentation operation, p, magnitude of the operation, m, span of target word, s, and probability distribution of sub policies, w. The datasets experimented in this study were SUBJ (Opinion Analysis), MR (Movie Review), SST1, SST-2 (Sentiment Analysis) and CR (Customer Review). The text classification models used to evaluate the performance of the data augmentation were CNN and Bi-LSTM models with pre-trained GloVe word embeddings with 300 dimensions trained on the Common Crawl corpus. The results showed the best average accuracy from all datasets achieved by the

BiLSTM classifiers was 77.33 and higher than the original average accuracy of 76.95 without augmentation.

Wei et al. (2021) proposed a text augmentation in multi-task view (MTV) which utilized both original and augmented examples in classifier training. Therefore, stronger levels of text augmentation can be conducted using greater strength parameter to create more perturbation in the augmentation process. The experimented text augmentation methods were replacing words in the sentence with synonyms in WordNet (Token Substitution), applying word-level dropout (Pervasive Dropout), inserting synonym of random words in the sentence (Token Injection) and swapping word positions randomly (Positional Shuffling). The datasets experimented in this study were SST2 (Sentiment Analysis), SUBJ (Subjectivity/Objectivity) and TREC (Questions). The BERT model was used to evaluate the performance of the text augmentation. The results showed the four MTV text augmentation methods improved the average accuracy of the baseline models from all datasets without text augmentation by 2.1 (Token Substitution), 2.5 (Pervasive Dropout), 2.2 (Token Injection) and 2.5 (Positional Shuffling).

Substitution is one of the most common text augmentation approaches. However, this approach is most likely to generate new sentences with the same meaning but different interpretation, which may potentially introduce more noise into the training data. This is because the selection of word to be replaced is based on synonyms or related words which can cause the sentence to become unnatural and non-interpretable syntactically. Table 2.1 illustrates the summary of prior studies on the substitution approach for text augmentation.

Table 2.1 Summary of prior studies on substitution approach for text augmentation

Scholar	Domain	NLP Task	Replacement Words Source
Kobayashi (2018)	• Movie Reviews • Short Sentences	• Sentiment Analysis • Opinion Polarity • Subjectivity or Objectivity	• WikiText-103 corpus • Twitter dataset • Urban Dictionary
Wang and Yang (2015)	• Twitter	• Behavioural analysis	• 100 billion words from Google News
Zhang, Zhao and LeCun (2015)	• News Articles • Customer Reviews	• Topic Classification • Polarity Prediction	• WordNet
Kolomiyets, Bethard and Moens (2011)	• Time Expression	• Time Recognition	• 80 million words from Reuters news article corpus and synonyms from WordNet
Wei and Zou (2019)	• Customer Reviews	• Sentiment Analysis • Opinion Polarity • Subjectivity or Objectivity • Pro or Con	• Synonyms from WordNet
Wei et al (2021)	• Headline • Relation Sentences • COVID Questions • Product Reviews	• Topic Classification	• Synonyms from WordNet
Neculoiu, Versteegh and Rotaru (2016)	• Job Title	• Job Classification	• Job title taxonomy
Sharifirad, Jafarpour and Matwin (2018)	• Twitter	• Sexism Detection	• Knowledge graph such as ConceptNet and Wikidata
Raiman and Miller (2017)	• Question & Answer	• Questioning & Answering	• Wikidata
Vijayaraghavan, Sysoev, Vosoughi and Roy (2016)	• Twitter	• Stance Detection	• Words from Twitter corpus
Giridhara et al. (2019)	• Relation Sentences	• Relation Classification	• WordNet • GloVe

Table 2.1 Summary of prior studies on substitution approach for text augmentation (continued)

Scholar	Domain	NLP Task	Replacement Words Source
Li et al. (2022)	• Opinions • Movie Reviews • Customer Reviews	• Opinion Analysis • Sentiment Analysis	• Random unigram in dataset
Wei et al. (2021)	• Movie Reviews • Sentences • Questions	• Sentiment Analysis • Subjectivity or Objectivity • Domain Classification	• WordNet

2.2.2 Back-translation

Back-translation in text augmentation focuses on translating sentence to foreign language and back to the original language to paraphrase the sentence using existing translation models. Risch & Krestel (2018) augmented new training data for aggression identification on comments from Facebook using back translation. First, each English comment with label in the dataset was translated into multiple foreign languages such as French, German and Spanish. Next, all translated comments were re-translated back into English with the initial label preserved. If the wording in the back translated comment remained unchanged from the initial comment, then it would not be added into the augmented dataset. This augmentation method successfully expanded the original dataset from 15,000 labelled comments up to 60,000 labelled comments. The performance of the data augmentation method was evaluated on a multi classification task for aggression identification consisting of three different classes: overtly aggressive, covertly aggressive and non-aggressive. The classification models included GRU-RNN with pre-trained fastText embeddings, logistic regression classifier with word n-grams features, logistic regression classifier with character n-grams features,

logistic regression classifier with hand-picked features such as emoticon, punctuation and capitalization, and an ensemble model combining the outcomes of the four models. The results showed a significant improvement which obtained a F1 score of 0.5846 with the augmented dataset compared to non-augment dataset with F1 score of 0.5722.

Aroyehun & Gelbukh (2018) expanded the dataset for aggression detection in Facebook posts by augmenting new training examples with back translation while preserving the labels of the original post. Each example of the Facebook post in the dataset was translated into four intermediates languages such as Hindi, German, Spanish and French using Google Translate API, and then translated back to English to generate four more new examples with label from the initial example. The data augmentation method was also evaluated on aggression detection predicting three classes: overtly aggressive, covertly aggressive and non-aggressive. The classifiers included Naive Bayes Support Vector Machine (NBSVM) model with character n-grams as baseline and seven deep learning models with pre-trained fastText embeddings. The results showed all classifiers trained with augmented dataset outperformed classifiers trained with non-augmented dataset when tested on the development set especially the LSTM model with improvement of weighted F1 score from 0.494 to 0.568.

Yu et al. (2018) augmented new training data for Stanford Question Answering Dataset (SQuAD) using back translation to paraphrase the sentences in the dataset. The back-translation process involved two translation models. The first model translated English text to a pivotal language while the second model translated the text from pivotal language back to English. The translation model used was a replicate of Google's Neural Machine Translation (GNMT) system and the selected pivotal languages were French and German. Therefore, the translation models were trained on 36 million English-French sentence pairs and 4.5 million English-German sentence

pairs. The number of translated instances by the translation model was determined by the beam width of the model, which was set to 5, so each translation model generated five new instances for each sentence. A total of 25 new instances were generated from two translation models. The training data in SQuAD was a triplet of (*d, q, a*) where *d* represented a document in paragraph form with multiple sentences, *q* was the question related to *d* and *a* was the answer to the question. Therefore, the data augmentation process consisted of two main steps. The first step was to back-translate all the sentences in *d* to produce a new document called *d'*. After each translation, the original sentence, *s* with the actual answer was paraphrased to a new sentence called *s'*. Thus, the second step was to identify the new answer, *a'* from *s'* by computing the character-level 2-gram scores between each word in *s'* with a to identify the position of new answer, *a'*. The data augmentation method was evaluated on a question answering model with convolution network and self-attention. The results obtained from the test set showed the model trained on augmented dataset three times the original dataset achieved Exact Match of 0.762 and F1 score of 0.846.

Data augmentation using backtranslation is quick and simple because it only needs to translate the sentence to a new language and re-translate it back to the original language. However, this approach can only generate a limited number of new sentences with minimal changes in context because it only paraphrases the original sentences instead of adding totally new unseen sentences to the dataset. Also, this approach also highly depends on the translation model used to perform the back-translation. Table 2.2 includes a summary of prior studies on the back-translation approach for text augmentation.

Table 2.2 Summary of prior studies on back-translation approach for data augmentation

Scholar	Domain	NLP Task	Pivotal Language
Risch and Krestel (2018)	• Facebook	• Aggression Detection	• French • German • Spanish
Aroyehun and Gelbukh (2018)	• Facebook	• Aggression Detection	• Hindi • German • Spanish • French
Yu, Dohan, Luong, Zhao, Chen, Norouzi and Le (2018)	• Question & Answer	• Questioning & Answering	• French • German

2.2.3 Text Generation

Text generation in text augmentation focuses on generating new training data similar to the existing data from a dataset. Luo et al. (2021) proposed a data augmentation framework that leveraged sentence compression and data screening to generate new artificial text data based on existing data. Attention-based LSTM model was used to compress the initial training data by removing the adjectives and adverbs in the sentences. The removed words were then replaced with new sentiment words from a sentiment dictionary. The data from the original dataset and the compressed data were subsequently used for training the text generator for generating new artificial data. The text generator comprised of a sequence generative adversarial networks (SeqGAN) model. In addition, a classifier trained on the dataset was used to filter out bad artificial data generated by the text generator while the good artificial data were remained as the augmented dataset. The Stanford Sentiment Treebank (SST) dataset and Hate Speech (HS) dataset collected from Twitter using distant supervision method were used in the experiments. The classifiers used to evaluate the performance of the data augmentation method included logistic regression, SVM, CNN and LSTM. The results showed an

average accuracy score of 80.1 was achieved by all four classifiers trained on the augmented SST dataset, which was slightly higher than the average accuracy score of 79.1 obtained from the classifier trained on the original dataset. Results from the HS dataset also showed similar observation with an average accuracy score of 75.6 across four classifiers using augmented data as opposed to only 74.7 using only the original dataset.

Liu et al. (2020) proposed a text augmentation framework generating new text data from a conditional text generator. The conditional text generator was a modified version of an existing pre-trained language model (GPT-2) with an extra reinforcement learning component between the softmax and argmax functions in the decoding stage. The purpose of reinforcement learning is to guide the text generator to generate modified sentence from the original sentence based on specific class label. The ICWSM 20' Data Challenge dataset (Offense Detection), SemEval 2017 Task 4A dataset (Sentiment Analysis) and SemEval 2018 Task 3A dataset (Irony Classification) were included in the evaluation. The classifiers used to evaluate the performance of the data augmentation method included CNN, Attention-based BiLSTM, Transformer, BERT and XLNet. All classifiers trained with a portion of the original data and augmented data showed significant improvement in contrast with classifiers trained without augmented data.

Yoo et al. (2021) proposed a data augmentation method by combining text perturbation, pseudo-labeling and knowledge distillation to generated new text examples from existing datasets samples using language models such as GPT-3. Text samples were extracts randomly using uniform distribution from training data and embedded in the prompt to generate augmented mixed sentences. Then, each augmented sentence was given a soft-label predicted by the language model. The datasets

experimented in the study were SST-2 (Movie Reviews), CR (Amazon Product Reviews), COLA (Publication Sentences), TREC6 (Questions), MPQA (Opinion Polarity) and RT20 (Movie Reviews). Downstream classification experiments were conducted on artificially data-scarce tasks by sub-sampling the training data. Fifteen different data seeds were sub-sampled for each dataset for the augmentation process and downstream classification. The sub-sample levels experimented were 0.1%, 0.3%, and 1.0%. The classifier models used to evaluate the performance of the data augmentation method were BERT with 109M parameters and DistilBERT with 67M parameters. The results showed the best average accuracy of 79.2 from all datasets using 1.0% sub-sample level for augmentation achieved by the BERT classifier, which was higher than the original average accuracy of 75.5 without augmentation.

Text generation in data augmentation is similar to the substitution and back-translation approaches in term of generating artificial text based on the original text. Therefore, it is highly dependent on the quality of the original data used to train the text generator. However, the choice of generative model is also important in generating grammatically correct sentences and sentences that can preserve the same meaning from the original sentences. Apart from that, extra filtering is necessary to minimize the effect of the noise in the augmented dataset. Table 2.3 illustrates the summary of prior studies on text generation approach for text augmentation.

Table 2.3 Summary of prior studies on text generation approach for data augmentation

Scholar	Domain	NLP Task	Generative Model
Luo, Bouazizi and Ohtsuki (2021)	• Movie Reviews • Twitter	• Sentiment Analysis • Hate Speech Detection	• SeqGAN
Liu et al. (2020)	• Twitter	• Offense Detection • Sentiment Analysis • Irony Classification	• GPT-2
Yoo et al. (2021)	• Movie Reviews • Product Reviews • Publication Sentences • Questions • Opinion Polarity	• Sentiment Classification • Domain Classification • Subjectivity or Objectivity	• GPT-3 • GPT-3Mix

2.2.4 Similarity Comparison

The similarity comparison approach in text augmentation focuses on annotating new unseen data similar to the data in existing dataset. Lu et al. (2006) augmented new training data to improve the performance of text categorization on articles from Text Retrieval Conference (TREC) 2005 which contained articles from four different categories such as allele mutation (A), gene expression (E), gene ontology (G) and tumour (T). The data augmentation method utilised Gaussian random fields and harmonic functions to perform label propagation which probabilistically assigned new label to unlabelled data from labelled data. This method involved the calculation of cosine angle between two document vectors on a weighted undirected graph. The document vectors were represented as vertices and the edge of the two vectors represented the similarity between two document vectors. The label propagation was performed on the unlabelled data using positive training examples with label from each category and 500 manually chosen negative examples which did not belong to any of

the categories. Top 100 newly labelled data for each category with the highest probability to be labelled as positive were added to the augmented dataset. The performance of the augmented data was evaluated using SVM on classification tasks. The results showed a significant improvement for overall performance using the augmented dataset when compared to original dataset. For instance, there was an increment on AUC score of category T from 0.817 to 0.915.

Fürstenau & Lapata (2009) augmented new data for FrameNet-style semantic role annotations by computing the semantic and syntactic similarity between labelled data and unlabelled data. First, all the words in a sentence were lemmatized and the predicate-argument structures of labelled and unlabelled data were extracted using the verb as the frame evoking elements. Also, the semantic and grammatical role for each argument were recorded. Next, the similarity measure between labelled and unlabelled predicate-argument structure with the same frame evoking verb was computed. Each argument from labelled predicate-argument structure was aligned to each argument in unlabelled predicate-argument structure and an alignment score between them was calculated based on syntactic similarity between grammatical roles and semantic similarity between words. After the best alignments were obtained, the role from each argument in labelled predicate-argument structure was transferred to the argument in unlabelled predicate-argument structure. Finally, a predefined number of k nearest neighbours for each labelled data were selected and added into the augmented dataset. The performance of the data augmentation method was evaluated using SVM on semantic role labelling tasks. The results showed the classifier trained on the augmented dataset with k equals to 2 achieved F1 score of 0.409 which was better compared to classifier trained on the original dataset with only F1 score of 0.385.

Gupta et al. (2018) augmented new training examples for emotion detection on textual conversation by calculating the cosine similarity between annotated examples and unannotated examples. First, tweets were collected using Twitter Firehose from 2012 to 2015. The corpus contained conversational pairs composed of tweets and responses. Next, a small part of the corpus was randomly sampled and annotated with three emotion classes (Happy, Sad and Angry) using human judgment. Then, a variant LSTM model was used to generate sentence embeddings for all annotated examples and unannotated examples. Potential examples for each emotion class were picked out from unannotated examples by comparing the cosine similarity between the sentence embeddings of annotated examples and unannotated examples. Finally, all potential examples belonging to each emotion class were further validated by human judgment to discard poor quality examples. The performance of the augmented dataset was evaluated using a LSTM model with 2 embedding layers on a multi-class classification task. The test set used contained 2,226 three-turn conversations obtained from Twitter in 2016 and each conversation was annotated with an emotion class using human judgment. The results showed the model achieved an average F1 score of 0.7134 for three all emotion classes.

Giachanoua et al. (2019) proposed a semi-supervised annotation method by propagating sentiment labels from sentiment positive and negative tweets to sentiment neutral tweets that might implicitly contain sentiment signals. First, all tweets were clustered into multiple topic clusters using the hierarchical agglomerative clustering algorithm. In each cluster, all tweets were vectorized and then each tweet vector was paired with others to calculate the cosine similarity. The propagation of the sentiment for each tweet was defined based on two approaches: 1) the frequency of the sentiment labelled tweets higher than a predefined threshold, and 2) the average cosine score of

each sentiment label. The RepLab 2013 collection tweets with sentiment labels for reputation monitoring was used to perform the propagation. The polar fact filter used to evaluate the performance of the propagation by classifying neutral and sentiment-bearing tweets was built using SVM. The results showed the filter trained on the dataset with frequency-based propagation achieved F-measure score of 0.529, a significant improvement as opposed to the dataset without propagation that only achieved F-measure of 0.368.

The similarity comparison approach shares the same basis to our approach of comparing two text units to identify the most similar examples to be added to the training set. However, the focus of this research is on microblog texts and fine-grained emotion classification. In addition, the proposed method used for similarity scoring is utilizing pre-trained embeddings models to calculate the similarity score between two text units. Furthermore, we proposed another two strategies (clustering and misclassified) to alter the seed set used for text augmentation, unlike prior studies on the similarity comparison approach which only augmented new data based on the instances from the original training set. The clustering strategy focuses on augmenting rare examples in the existing dataset and the misclassified strategy focuses on augmenting examples in the test set misclassified by the baseline model. Table 2.4 presents a summary of prior studies on the similarity comparing approach for text augmentation.

Table 2.4 Summary of prior studies on similarity comparing approach for data augmentation

Scholar	**Domain**	**NLP Task**	**Similarity Comparing Strategy**
Lu, Zheng, Velivelli and Zhai (2006)	• Medical Articles	• Text Classification	Calculation of cosine angle between two document vectors on a weighted undirected graph
Furstenau and Lapata (2009)	• FrameNet	• Semantic Role Labelling	Calculation of alignment score between each argument from labelled and unlabelled predicate-argument structure
Gupta, Catterjee, Srikanth and Agrawal (2018)	• Twitter	• Emotion Detection	Calculation of the cosine similarity score between sentence embeddings from annotated and unannotated examples
Giachanou, Gonzalo and Crestani (2019)	• Twitter	• Sentiment Analysis	Propagation sentiment labels based of cosine similarity between tweets

2.3 Emotion Detection

The two most common approaches used to classify texts into several emotion classes are explored in this literature review. The first approach uses the existing machine learning algorithms to train the classifier for emotion classification with a set of features. The second approach leverages deep learning for emotion classification and learn features directly from labelled data without manual feature extraction.

2.3.1 Machine Learning

The training data for the emotion classification task commonly come from two sources: human-annotated data and distant supervision. Human-annotated data are created by either expert annotators or obtained through crowdsourcing such as Amazon Mechanical Turk (AMT). Therefore, human-annotated data can be more reliable than data collected via distant supervision. However, manual annotation is time-consuming and labour-intensive compared to distant supervision in which large amount of self-labelled data can be collected in a short time using only keyword hashtags. The differences between human annotation and distant supervision in constructing the dataset are the efforts needed for construction and the quality of the outcome dataset. The efforts needed for conducting the human annotation are a lot higher than distant supervision, but it can guarantee to produce a dataset with higher quality standards. In the first half of this section, we discuss previous studies on emotion classification using machine learning approach with human-annotated data while the remaining half of the section focuses on emotion classification using machine learning approach with data collected via distant supervision.

Liew & Turtle (2016) developed 2 sets of classifiers with unigram as features for automatic fine-grained emotion detection on tweets using Support Vector Machine

with Sequential Minimal Optimization (SVM-SMO) and Bayesian Network (BayesNet). The first set of classifiers focused on the multi-class classification while the second set of classifiers focused on building binary classifiers for each emotion category. The corpus used to train the classifiers consisted of 5,553 tweets and annotated with 28 emotions and no emotion by 18 annotators. The results showed SVM-SMO achieved the highest average F1 score of 0.57 in a single multiclass classification while BayesNet achieved the highest average F1 score of 0.57 for all the emotion categories in the binary classification task.

Balabantaray et al. (2012) built a multi-class emotion classifier to differentiate six emotions and neutral tweets based on syntactic, semantic and contextual-based approaches using SVM with various features. The emotion categories were happy, sad, anger, disgust, surprise and fear. The feature set included unigrams, bigrams, personal-pronouns, adjectives, WordNet-Affect emotion lexicon, POS, dependency parsing and emoticons. The corpus used to train the classifier contained 8,150 tweets annotated by five annotators. The results showed the classifier achieved an accuracy of 0.7324 in the multi-class classification task.

Strapparava & Mihalcea (2008) trained a Naive Bayes classifier to identify six emotions (anger, disgust, fear, joy, sadness and surprise) on news headlines. The classifier was trained on 8,761 blog posts labelled with six emotions by the blog authors. The features used were based on WordNet-Affect lexicon. The result showed the classifier achieved the highest F1 score of 0.3287 on the emotion joy.

Alm et al. (2005) trained a multi-class classifier to predict negative and positive emotions from the narrative texts in children fairy tales using Winnow update rule with a wide variety of features. The negative emotion classes consisted of sentences labelled with angry, disgusted, fearful, sad, and negatively surprised while the positive emotion

classes comprised of sentences labelled with happy and positively surprised. The corpus contained 1,580 annotated sentences from 185 children's stories. The results showed average F1 scores of 0.32 was achieved for negative emotion while the average F1 score for positive emotion was 0.13.

Mohammad (2012) trained 6 different binary classifiers to classify six emotions on newspaper headlines using logistic regression and SVM. The six emotion classes were based on Ekman's six basic emotions. The features used were based on NRC emotion lexicon (Mohammad & Turney, 2010) and n-grams. The corpus was collected from SemEval-2007 Affective Text. The average F1 score from the six binary classifiers was 0.524.

Purver & Battersby (2012) evaluated the distant supervision method in collecting large datasets for multi-class emotion classification using Linear SVM and unigram features. The emotion classes used were based on Ekman's six emotion categories (happy, sad, anger, fear, surprise and disgust). Two methods were used to collect the tweets using distant supervision. The first method used emoticons representing each emotion class and the second method used hashtags representing each emotion class. The multi-class classifier was trained separately on the collected corpus and tested on 1,000 tweets labelled using Amazon's Mechanical Turk (AMT). The results showed the classifier trained with the corpus collected using emoticons achieved higher F1 scores on happy, sad and anger classes (0.775, 0.544 and 0.467) while fear, surprise and disgust classes achieved lower F1 score (0.123, 0.257 and 0.125). Moreover, the classifier trained with corpus collected using hashtags achieved similar results from the emoticons corpus but only with slightly lower overall F1 scores.

Davidov et al. (2010) trained two sets of classifiers to differentiate emotional tweets using K-nearest Neighbour (KNN) algorithm. The first set was a multi-class

classifier while the second set consisted of multiple binary classifiers. The classes used for the classification task were based on 50 unique emotion hashtags and 16 smileys. The corpus was collected using distant supervision and each class had 1,000 tweets. The hashtags and smileys used to collect the tweets were removed to avoid bias in the classification task. In addition, 10,000 tweets without any hashtag or smiley were randomly sampled to serve as the negative examples. The features used were unigrams, n-grams, frequent words and punctuation. The results showed the multi-class classifier trained using only emotional hashtags labels achieved an average harmonic F1 score of 0.64 while the multi-class classifier trained using only smileys as labels achieved an average harmonic F1 score of 0.31. On the other hand, the binary classifiers trained using 50 hashtags achieved an average F1 scores of 0.8 while the binary classifiers trained using 16 smileys achieved an average F1 score of 0.86.

Wang et al. (2012) trained a LIBLINEAR classifier to identify seven emotions on tweets. The emotions were joy, sadness, anger, love, fear, thankfulness and surprise. The features used were unigrams, bigrams, WordNet-Affect lexicon and POS tags. The corpus contained 2,488,982 self-labelled tweets with seven emotions collected using distant supervision. Even with a large training set of close to two million tweets, the results achieved the highest F1 score of 0.721 on emotion joy and the lowest F1 score of 0.139 on emotion surprise.

According to the existing studies using machine learning approach to classify text based on emotions, we can conclude that the training size and the features used to train the classifier highly affect the performance of the classification tasks. Classifiers trained using large corpora collected via distant supervision tend to give more promising result compared to using the small corpora with manual annotation. However, a large corpus does not guarantee better performance because sizable data with poor quality of

data can still deteriorate the performance of the classification tasks. Therefore, we need consider the quality over the size of the data when we are expanding EmoTweet-28. Table 2.5 illustrates the summary of prior studies on machine learning approach for emotion detection.

Table 2.5 Summary of prior studies on machine learning approach for emotion detection

Scholar	**Labels**	**Classifier**	**ML Features**	**Result**
Liew and Turtle (2016)	28 emotion categories	SVM-SMO, BayesNet	Unigram	Average F1 score in multi classification: SVM-SMO = 0.57 BayesNet = 0.51 Average F1 score in binary classification: SVM-SMO = 0.53 BayesNet = 0.57
Balabantaray, Mohammad and Sharma (2012)	happy, sad, anger, disgust, surprise and fear	SVM	Unigrams, bigrams, personal-pronous, adjectives, WordNet Affect lexicon, POS, dependency-parsing feature, emoticons	Average accuracy = 0.7324
Strapparava and Mihalcea (2008)	anger, disgust, fear, joy, sadness and surprise	Naive Bayes	WordNet Affect lexicon	F1 score: Anger = 0.1677 Disgust = n/a Fear = 0.563 Joy = 0.3287 Sadness = 0.2143 Surprise = 0.263
Alm, Roth and Sproat (2005)	neutral, negative and positive emotions	Winnow update rule	14 features	F1 score: Neutral = 0.69 Negative = 0.32 Positive = 0.13
Mohammad (2012)	happy, sad, anger, fear, surprise and disgust	Logistic regression and SVM	NRC emotion lexicon and n-grams	Average F1: score = 0.524

Table 2.5 Summary of prior studies on machine learning approach for emotion detection (continued)

Scholar	Labels	Classifier	ML Features	Result
Purver and Battersby (2012)	happy, sad, anger, fear, surprise and disgust	LIBSVM	Unigram	F1 score for emoticon dataset: Happy = 0.775 Sad = 0.544 Anger = 0.467 Fear = 0.123 Surprise = 0.257 Disgust = 0.125
Davidov, Tsur, Rappoport (2010)	50 unique emotion hashtags and 16 smileys	KNN	Unigrams, n-grams, pattern and punctuation	Average harmonic F1 score for 50 hashtags = 0.8 Average harmonic F1 score for 16 smileys = 0.86
Wang, Chen, Thirunarayan and Sheth (2012)	joy, sadness, anger, love, fear, thankfulness and surprise	LIBLINEAR	Unigram, bigram, WordNet-Affect lexicon and POS	F1 score: Joy = 0.721 Sadness = 0.647 Anger = 0.715 Love = 0.515 Fear = 0.439 Thankfulness = 0.571 Surprise = 0.139

2.3.2 Deep Learning

Deep learning utilises neural networks to interpret data features and make predictions by passing the information through several layers of data processing. Deep learning works well with unstructured data because it does not need manual feature engineering as it is capable to carry out feature engineering at its own. In this section, we will discuss previous studies on emotion classification using deep learning approach.

Gupta et al. (2018) applied LSTM-based deep learning model to train a multi-class classifier for emotion detection in textual conversations. The four emotion classes were happy, sad, angry and others. The corpus contained 554,000 utterances which were textual conversations labelled with emotions using the method of similarity scoring and

human judgment. The results showed the F1 score achieved by the LSTM model for happy, sad and angry were 0.5668, 0.8079 and 0.7134. Based on this study, the results obtained from the deep learning model trained with large amount of training data provided a better performance than the machine learning approach from the previous section.

Abdul-Mageed & Ungar (2017) trained a multi-class classifier to detect 24 fine-grained emotions on tweets using Gated Recurrent Neural Networks (GRNN). The corpus contained 1,608,233 self-labelled tweets collected using distant supervision. The results showed the classifier achieved an average F1 score of 0.8747 for all emotion categories which is also better than the overall performance from the machine learning approach in the previous section.

Zahiri & Cho (2018) trained a multi-class classifier to detect emotions on TV show transcripts using sequence-based CNN model with attention mechanism. The emotions classes were neutral, joyful, peaceful, powerful, scared, mad and sad. The corpus used contained 12,606 utterances annotated using AMT. The results showed the classifier achieved an average F1 score of 0.269 for the 7 classes. The poor performance was attributed to the size of the corpus, which insufficient for the deep learning model to learn features accurately. Therefore, we can conclude that the size of the corpus is one important factor affecting the performance of the deep learning model.

Felbo et al. (2017) pre-trained a variant LSTM model with millions of tweets labelled with emojis and transferred the learning to emotion classification. The model was evaluated on three existing datasets with different domains and emotion classes. The first dataset contained 1,250 headlines labelled with three emotion classes and the second dataset contained 1,059 tweets labelled with 4 emotion classes. The final dataset contained 7,480 self-reported emotional experiences constructed by psychologists with

seven emotion classes. The results showed the model achieved F1 scores of 0.37, 0.61 and 0.57 for headlines, tweets and emotional experiences. In this study, transfer learning was used to overcome the problem of small corpora. Therefore, the results obtained were decent although the size of the training data was still relatively small compared to other studies.

Based on existing studies using deep learning for emotion classification, we can conclude that deep learning models can provide similar or even better results in emotion classification compared to machine learning models. However, the amount of the data required to train the classifier using deep learning is also higher than using regular machine learning. Therefore, the size of the EmoTweet-28 corpus should be expanded until an optimal level to apply deep learning in fine-grained emotion classification. Table 2.6 illustrates the summary of prior studies on deep learning approach for emotion detection.

Table 2.6 Summary of prior studies on deep learning approach for emotion detection

Scholar	**Labels**	**Classifier**	**Result**
Gupta, Chatterjee, Srikanth and Agrawal (2018)	happy, sad and angry	LSTM	F1 score: Happy = 0.5668 Sad = 0.8079 Angry = 0.7134
Abdul-Mageed and Ungar (2017)	24 fine-grained emotions	GRNN	Average F1 score = 0.8747
Zahiri and Choi (2017)	neutral, joyful, peaceful, powerful, scared, mad and sad	Sequence-based CNN	Macro F1 score = 0.2690
Felbo, Mislove, Søgaard, Rahwan and Lehmann (2017)	three emotion classes (headlines) four emotion classes (tweets) seven emotion classes (experiences)	LSTM	Average F1 score: Headlines = 0.37 Tweets = 0.61 Experiences = 0.57

2.4 Research Gap

Prior studies for text augmentation have mostly focused on paraphrasing the existing text unit and there are still limited studies that have explored the similarity comparison approach. We proposed a text augmentation strategy to expand the existing fine-grained emotion corpus by adding similar examples from actual data collected using distant supervision instead of generating new artificial examples through substitution, back-translation and text generation. Therefore, the augmented data can be more realistic and have greater diversity as opposed to artificial data. Also, existing studies utilizing the similarity comparison approach for text augmentation generated the vector representations of the sentences for similarity comparison using limited variants of pre-trained neural embeddings models. In this study, we explored six pre-trained neural embedding and Bag-of-Words models to generate the vector representations of the sentences for calculating the similarity between two text units. The six pre-trained neural embeddings models consist of three word embeddings (Word2Vec, GloVe and fastText) and three sentence embeddings (USE, InferSent GloVe and InferSent fastText) models. Word and sentence embeddings models are widely used for representing texts in vector representation in various text mining tasks. In addition, unlike Gupta et al. (2018), we performed threshold-based and fixed increment selection instead of expert judgment to determine the outcome of the text augmentation. Thus, excessive human intervention can be avoided.

2.5 Summary

Text augmentation methods can be generally categorized into four groups: substitution, back-translation, text generation and similarity comparison. Substitution is the most popular and used by many researchers to augment new positive examples from existing labelled corpora. However, our proposed data augmentation strategy focuses on the similarity scoring approach in which we will augment an emotion corpus by selecting new tweets that are the most similar to labelled emotion examples in EmoTweet-28. For emotion classification in text, machine learning and deep learning are two approaches that are becoming more common. Using deep learning in emotion classification can produce similar or even better results compared to using machine learning. However, deep learning models require more training data than machine learning models to be accurate in emotion classification. Therefore, the aim of this research is to explore a similarity-based text augmentation strategy to efficiently expand EmoTweet-28.

CHAPTER 3

METHODOLOGY

3.1 Introduction

To address the goal of the research study in identifying the most suitable text augmentation method to expand the positive examples in the existing gold standard emotion corpus using a combination of distant supervision and similarity scoring approach, the research methodology is divided into three phases: 1) Data Preparation, 2) Text Augmentation and 3) Emotion Classification.

3.2 Methodological Framework

Figure 3.1 illustrates the three phases in the research methodology to address the research questions in this study:

- R1: How can new training data be augmented using similarity scoring approach to improve the quality of data collected from distant supervision method?

- R2: What is the effect of using similarity scoring approach to augment more relevant training data on the performance of deep learning model for emotion classification?

- R3: Which text augmentation strategy is better at augmenting more relevant training data?

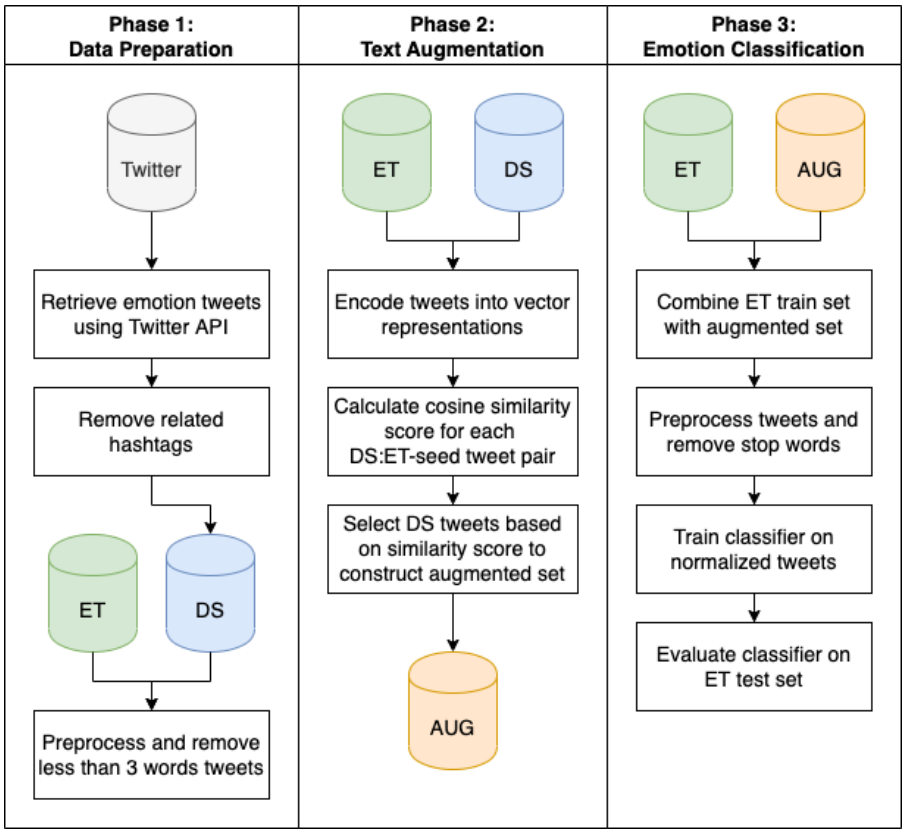

Figure 3.1 Methodological framework

R1 is addressed in Phase 1 and Phase 2 in which new tweets were collected using distant supervision (DS) for each of the six selected emotion categories. Then, each DS tweet was paired with each seed sample from EmoTweet-28 (ET-seed) in the same emotion category. The tweets were vectorized using different vector representations and then the similarity scores were computed for each pair of DS:ET-seed. Next, DS tweets were selected based on the similarity score to construct the augmented set (AUG).

R2 and R3 are addressed in Phase 3 in which emotion classifiers were developed using neural network models and trained on different sizes of augmented sets for each emotion category separately. The classifier performance was evaluated based on the ET test set for each emotion category from EmoTweet-28. Finally, performance results

from the experiments with the augmented sets were compared, and the best vector representation to augment emotion tweets for emotion classification was identified.

3.3 Phase 1: Data Preparation

Two corpora used in this research were 1) EmoTweet-28 (ET) and 2) a corpus containing self-labelled tweets collected from December 2019 until May 2020 using distant supervision (DS). ET served as the gold standard corpus in this study and a portion of the positive examples (80%) from ET were used as the seed examples to augment more training samples by selecting the most similar tweets from DS to each seed example. The remaining positive examples (20%) were used as the test set (ET-test) to evaluate the performance of the augmented data on emotion classifiers.

3.3.1 EmoTweet-28 (ET)

EmoTweet-28 (ET) is an existing fine-grained emotion corpus created by Liew et al. (2016). ET contained 15,553 tweets annotated with 28 emotion categories. The tweets were sampled using four different sampling strategies based on random, topic, average users and active users. Tweets could be labelled with more than one emotion category or absence of emotion (none). Tweets with more than one label were removed from ET because the data augmentation in this research focused on only single labelled emotion tweets. The purpose of ET is to serve as the baseline train set, test set and seed set for text augmentation. All tweets were pre-processed and lowercased. The pre-processing methods included removal of mentions, URL links and punctuation. Also, every contraction and word in a tweet was expanded and each word was lemmatized. Then, tweets with less than three words were discarded because they were not able to provide sufficient context for computing the similarity score. The remaining total

number of single label tweets remained in ET after pre-processed is 13,949. ET tweets labelled with each class were divided into train:test subsets based on an 80:20 ratio. The 80% train subsets for each class were concatenated to become the baseline train set (ET-train). On the other hand, the 20% test subsets for each class were concatenated to become the baseline test set (ET-test). As a binary classifier was developed for each emotion category, for an emotion *x*, we maintained the tweets labelled with emotion *x* as positive examples and converted all other labels as negative examples representing the absence of emotion *x*.

The scope of this research is to expand the positive examples from six emotion categories in EmoTweet-28: *happiness*, *anger*, *excitement*, *boredom*, *desperation* and *indifference*. The six emotion categories were selected because we would like to observe the effect of the proposed text augmentation strategy on high and low frequency of available positive seed examples. Therefore, the 80% train subsets of the tweets for these six emotion categories also respectively served as the seed sets (ET-seed) for text augmentation. The ET-seed set was used as the gold standard to identify the most similar tweets from self-labelled tweets in DS corresponding to the same emotion category.

According to Table 3.1, the sizes of *boredom*, *desperation* and *indifference* seed sets are relatively small compared to *happiness*, *anger* and *excitement*. This imbalance in size may affect the performance of the text augmentation performance because a small seed set is less diverse compared to a big seed set, thus providing limited seed examples for corpus expansion. Therefore, we manually annotated new positive examples called silver sets for the *boredom*, *desperation* and *indifference* seed sets to fix the data imbalanced problem. The silver sets were annotated using content analysis on tweets collected from June 2020 to August 2020 using distant supervision. Two

primary researchers verified if the emotion hashtags from distant supervision accurately reflected the emotion expressed in each tweet in the silver sets. The average Cohen's Kappa score achieved across *boredom*, *desperation* and *indifference* was 0.334 (boredom = 0.345, desperation = 0.228 and indifference = 0.428).

Table 3.1 The number of pre-processed single label tweets for six selected emotion categories in EmoTweet-28 and the addition of silver sets for low frequency emotion categories

Emotion	**Single label tweets (GOLD)**	**80% GOLD**	**20% GOLD**	**Silver tweets (SILVER)**	**Total seeds (80% GOLD + SILVER)**
Happiness	1280	1024	256	-	1024
Anger	971	776	195	-	776
Excitement	522	417	105	-	417
Boredom	37	29	8	112	141
Desperation	49	39	10	46	85
Indifference	52	41	11	56	97

3.3.2 Distant Supervision (DS)

Distant supervision is a method to collect large amount of self-annotated data on microblog such as Twitter using keyword hashtags in short amount of time. This method can retrieve multiple relevant tweets in the same topic based on the keyword hashtags used in the tweets as hashtags are usually used as the labels to specify the content and theme of the tweets. Thus, all tweets collected through this method are considered as the self-labelled tweets and the labels of the tweet are based on the category of the keyword hashtags.

In this research, a total of 577,642 self-labelled tweets with six emotions were collected from December 2019 until May 2020 using the Twitter API. The keyword hashtags used for each emotion category were based on the emotion synonyms or words with similar meaning. Each tweet was automatically assigned with a label corresponding to the category of the keyword hashtag. Then, all the keyword hashtags

were removed from the collected tweets to avoid bias in similarity scoring. Furthermore, all duplicated tweets were removed to prevent redundancy. The DS tweets were pre-processed with the same method used for ET tweets and tweets with less than three words were also discarded. Table 3.2 illustrates the keyword hashtags used to collect the self-labelled tweets for each emotion category while Table 3.3 shows the number of self-labelled tweets collected using distant supervision for each emotion category.

Table 3.2 Keyword hashtags used to collect tweets for each emotion category

Emotion	**Keyword Hashtags**
Happiness	#happiness, #cheerful, #contented, #delighted, #ecstatic, #elated, #joyful, #joy, #jubilant, #happy
Anger	#angry, #annoyed, #enraged, #exasperated, #furious, #offended, #outraged, #rage, #irate, #anger, #displeased
Excitement	#excitement, #excited, #enthusiastic, #thrilled, #animated, #stirred, #stimulated
Boredom	#boring, #bored, #dull, #dullness, #yawn, #boredom
Desperation	#desperation, #despair, #desperate, #hopeless, #nohope, #pointless, #despondent, #despondence
Indifference	#indifference, #indifferent, #apathy, #apathetic, #disinterested, #unconcerned, #dontcare

Table 3.3 The number of self-labelled tweets collected for the six selected emotion categories using distant supervision

Emotion	**ORIGINAL**	**AFTER PRE-PROCESSING**
Happiness	385,579	362,435
Anger	44,197	40,737
Excitement	54,669	51,269
Boredom	78,166	75,634
Desperation	38,871	32,949
Indifference	7,365	7,142

3.4 Phase 2: Text Augmentation

The proposed text augmentation method focuses on finding new tweets (DS) that are similar to the existing tweet (ET-seed) in the gold standard. DS tweets with high similarity score to any ET tweet in the same emotion category are selected as the augmented tweets and assigned with emotion labels.

Figure 3.2 illustrates the similarity scoring approach to obtain the highest cosine similarity score for each DS tweet. *A* represents the vector for each DS tweet and m is the total number of the self-labelled tweets collected through distant supervision method for one emotion category. *B* represents the vector for each ET-seed in the same emotion category, E, and n represents the total number of single labelled tweets from EmoTweet-28. The similarity score, *x*, for each DS:ET-seed tweet pair is computed based on cosine similarity measure as shown in Equation 1. Cosine similarity is chosen as it is the most popular among other existing similarity metrics and is widely used in information retrieval and relevant studies (B. Li & Han, 2013; Rahutomo et al., 2012).

$$cosine\ similarity(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (1)$$

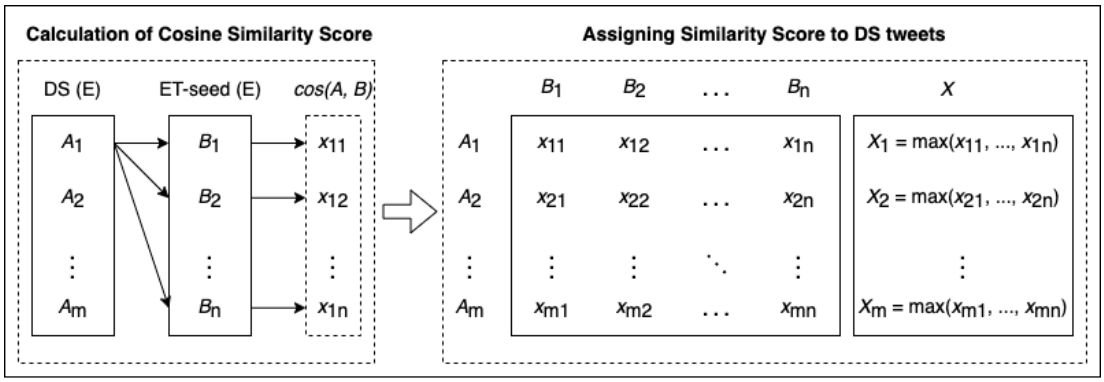

Figure 3.2 Similarity scoring approach

The similarity score ranges from 0 to 1, in which 1 represents the exact same DS tweet to and ET-seed tweet while 0 represents the least similar. The final similarity

score, *X*, for each DS tweet is the maximum value in a sequence of similarity scores computed across all ET-seed tweets under same emotion category. There were seven vector representations experimented in this research for the computation of the similarity score. The seven vector representations encompassed Bag-of-Words (BoW), word embeddings and sentence embeddings.

3.4.1 Bag of Words (BoW)

BoW is the most common approach to convert text into dimensional vector in which each dimension represents the occurrences of specific word based on a predefined list of known words. However, the limitation of this approach is it does not take semantics or ordering of the words into consideration. Therefore, BoW is used as the baseline vector representation in this research. The vocabulary words used to encode the tweets are based on 18,523 distinct terms in the original EmoTweet-28 corpus.

3.4.2 Word Embeddings

This approach is knowns as distributed word representation which means each word within a predefined vocabulary is embedded with a vector representation and words with same meaning should have the vector representations near to each other. Therefore, it can recognize the semantic and syntactics values between the words which can help to identify tweets with similar meaning or intent. There are various methods to train the word embeddings and a commonly used method is called Word2Vec (Mikolov et al., 2013), which is utilizing a simple neural network model to learn the word embeddings from a corpus. A second method, fastText (Mikolov et al., 2018) is an improvised method based on Word2Vec's Skip-gram model, which learns the

representation using character n-grams from words. Third, GloVe (Pennington et al., 2014) is a method to learning word embeddings based on the frequency of global co-occurrence word-to-word pair within a corpus.

In this research, we are not training a new word embeddings from scratch because it requires high amount of data to cover a wide variety of words and learn the word representation accurately. Therefore, we have selected three pre-trained word embeddings models to encode each word within a tweet into its embeddings. The first and second models are obtained from (Godin, 2019), which are pre-trained using Word2Vec (Skip-gram) and fastText method with 400 dimensions on 400 million tweets with approximately 5 billion words. The third model is pre-trained using the GloVe method with 200 dimensions on Twitter 2B[1]. To compute the cosine similarity score between two tweets, all the word embeddings found in each tweet are averaged to become a final embedding representing the tweet and use it for the computation of similarity score.

3.4.3 Sentence Embeddings

Sentence embedding is similar to word embedding but instead of converting word by word into vectors, we can utilize an existing pre-trained sentence encoder to convert a full sentence or tweet into dimensional vector while preserving the syntactic and semantic information. Therefore, sentences with similar meaning may have vectors closer to each other. In this research, Universal Sentence Encoder (USE) (Cer et al., 2018) is chosen one of the sentence encoders to convert tweets into sentence

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

embeddings. USE is a sentence encoder pre-trained with numerous types of data using deep averaging network (DAN) encoder and is available on TensorHub[2].

In addition, we have also included two InferSent sentence encoders (Conneau et al., 2018) pretrained on various datasets using the bi-directional LSTM architecture with max pooling. The main difference between the two InferSent models is the word embeddings model used to train the encoder. The first InferSent model uses GloVe word embeddings trained on Common Crawl 840B (InferSent-GloVe) and the second InferSent model uses fastText word embeddings trained on Common Crawl 600B (InferSent-fastText). In summary, three types of sentence embeddings are included in the experiments.

3.4.4 Augmented Sets

We experimented with two approaches to construct the augmented sets from DS tweets based on the final similarity scores which were threshold-based selection and fixed increment selection. Threshold-based selection picked the DS tweets based on a predefined similarity score threshold, *t.* DS tweets with the final similarity scores higher than *t* are grouped and used to construct the augmented set (AUG-*t*). We tested predefined thresholds of 0.5, 0.6, 0.7, 0.8 and 0.9. Figure 3.3 illustrates the construction of the augmented sets in threshold-based selection.

[2] https://tfhub.dev/google/universal-sentence-encoder/4

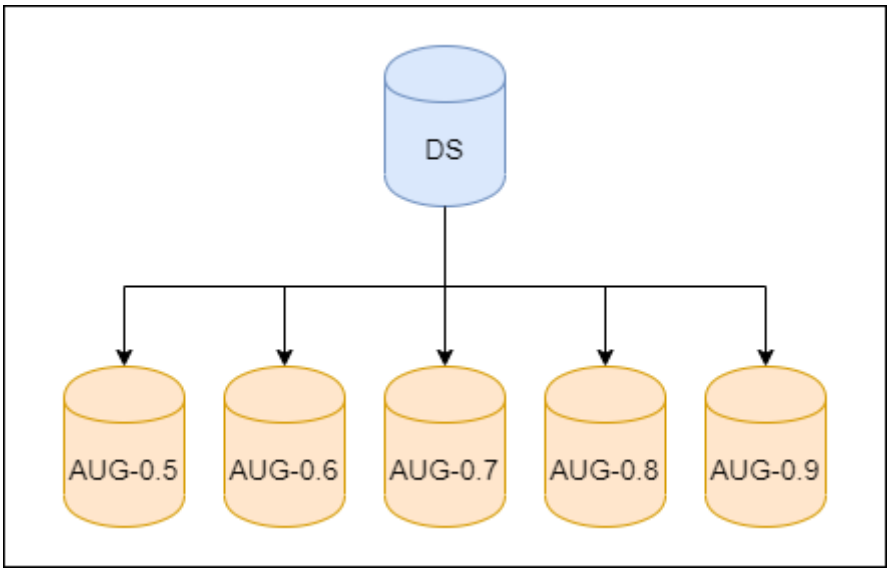

Figure 3.3 Threshold-based selection

The fixed increment selection limits the total number of most similar tweets extracted from DS tweets based on a fixed number of tweets, *I*. First, we rearranged the DS tweets in descending order based on the final similarity score which means that tweets with high scores were located at the top. Then, we extracted *I* number of tweets with the highest similarity scores to construct the augmented set (AUG-*I*) in increments of 400 for 30 times, thus producing 30 augmented sets with the smallest AUG-400 containing 400 most similar DS tweets and the largest AUG-12000 containing 12,000 most similar DS tweets. The fixed increment size of 400 was selected as we observed 400 was the minimum number of augmented data required to improve the performance of the baseline model and any number beyond 12,000 did not show any significant improvement according to the results from threshold-based experiments. Figure 3.4 illustrates the construction of the augmented sets in fixed increment selection. The purpose of increment selection is to investigate the effect of the augmented samples in fixed sizes because the similarity scores obtained from the similarity scoring approach using different vector representations exhibit different characteristics. For example, the similarity scores computed using the BoW representation are usually very low or zero

compared to other approaches because it is based on the exact word matching between two tweets unlike word and sentence embeddings in which all tweets have their own embedding values generated from the pre-trained embeddings models.

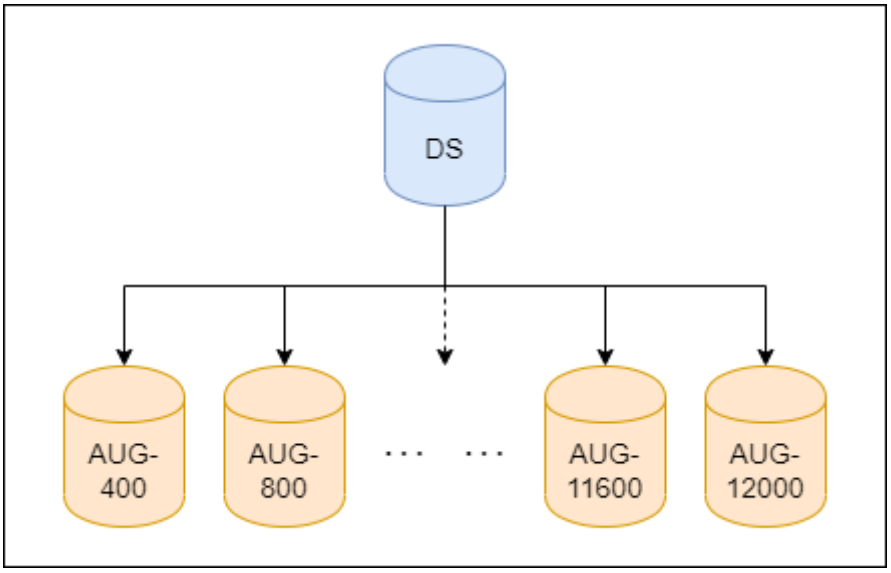

Figure 3.4 Fixed increment selection

3.5 Phase 3: Emotion Classification

The performance of proposed text augmentation strategy on EmoTweet-28 is evaluated using the machine learning approach. The machine learning problem is framed as a binary classification to predict whether a tweet expresses a specific emotion or not. The input for the classifier is the text sequence of word indices where each index maps to a word in a dictionary based on all vocabulary terms found in both ET and DS. The input text sequence is also padded with zeros until the maximum length of 100 to maintain the consistency across all sequences. The target output is a binary value where '1' represents the tweet expresses the specific emotion while '0' represents the absence of specific emotion in the tweet. All training and test data are pre-processed using the same steps in preparing data for similarity scoring but with an extra step, which is removing all the stop words. In this research, we selected Bi-directional Long Short-

Term Memory (BiLSTM) to build the binary classifier for evaluating the effect of adding augmented training samples from different vector representation to baseline model and determine which augmentation strategy is better at augmenting more relevant training data. In addition, we also prepared a baseline classifier using support vector machine (SVM) trained on the ET-train without augmented data.

3.5.1 BiLSTM Architecture

Figure 3.5 illustrates the architecture of the BiLSTM model used in this research which consists of embedding layer, bi-directional LSTM layer, and output layer with sigmoid activation. Adam optimizer with the rate of 0.001 and binary entropy loss function are also applied to the model. For the embedding layer, we used the pre-trained 200-dimensional GloVe embeddings on Twitter 2B instead of training our own word embeddings to convert words to into vectors and we set the embedding layer as non-trainable during the training process to prevent the model from retraining the pre-trained GloVe embeddings. For hyperparameter tuning, we performed 5-fold cross validation on the baseline ET-train for each emotion category using Grid Search and experimented with 64, 128 and 256 number of LSTM hiddent units. Table 3.4 shows the best number of LSTM hidden units for each emotion classifier obtained from Grid Search. We set the dropout rate to 0.5 for both embedding and BiLSTM layers which was the optimal value obtained from the classifiers for all 6 emotion categories. The number of epochs and batch size used for training were both set to 10.

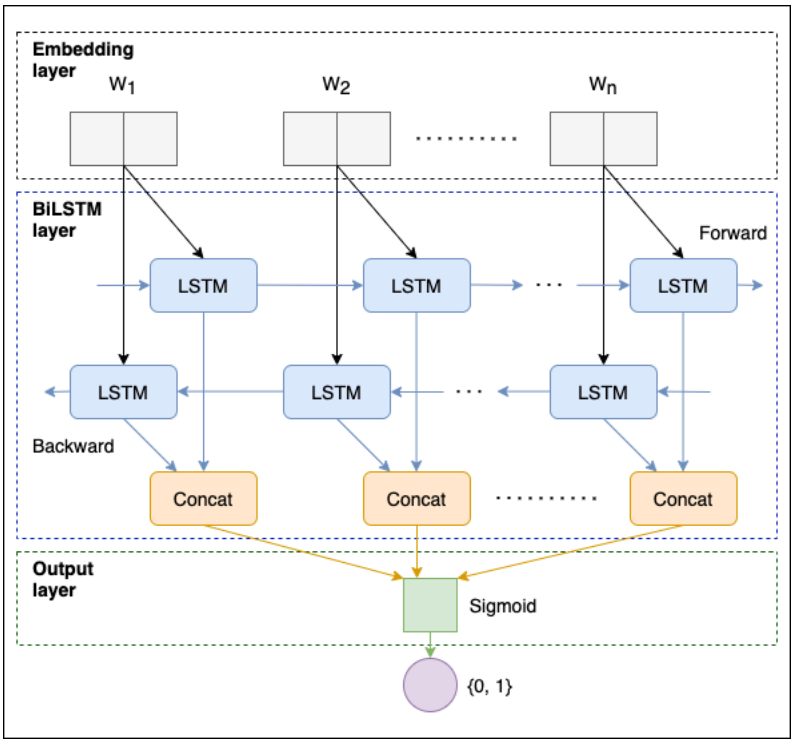

Figure 3.5 BiLSTM architecture

Table 3.4 Number of LSTM hidden units for six different emotion classifiers

Emotion Classifier	Number of LSTM hidden units
Happiness	256
Anger	128
Excitement	128
Boredom	64
Desperation	128
Indifference	128

3.6 Experimental Design

Two sets of experiments were conducted in this research for each emotion category. The first experiments set was to train multiple binary classifiers using the BiLSTM model on ET-train for each emotion category combined with all threshold-based augmented sets (AUG-*t*) from the seven vector representations within the same emotion category. The new training set after combined with ET-train must be shuffled to ensure the positive and negative labels were evenly distributed and not clustered together. The second experiment set was similar to the first experiment set but the ET-

train for each emotion category was combined together with the fixed increment augmented sets (AUG-*I*). In addition, a random augmented set containing tweets randomly sampled from DS was also included in fixed increment selection to compare the performance between random augmentation and the proposed text augmentation in the study. In addition, three baseline classifiers were trained based on ET-train and all DS tweets using SVM and BiLSTM. Finally, all trained classifiers were evaluated on the respective emotion test set (ET-test).

Next, the original text augmentation strategy (PRIMARY) was modified to further investigate the performance of the similarity scoring by altering the ET-seed. The first modified strategy used rare examples as the seed tweets. Hence, we can increase rare examples in the original corpus and prevent the classifiers from only focusing on frequent patterns. The tweets in the original ET-seed for each emotion category were divided into 10 clusters using k-means clustering and the 4 smallest clusters with the least number of tweets were selected to become the seed samples. The second modified strategy selected misclassified tweets from ET-test for each emotion category as the seed samples. The misclassified tweets were actual positive examples of an emotion but predicted by the baseline BiLSTM classifier as being negative examples. The intuition behind the misclassified strategy was to augment more data similar to examples that the classifiers made errors on. The vector representation for similarity scoring applied in these two modified strategies was based on the best approach obtained from PRIMARY.

In summary, we experimented with both threshold-based selection and fixed increment selection for the original text augmentation strategy (PRIMARY). Then, we used fixed increment selection in the clustering (CLUSTER) and misclassified

(MISCLASSIFIED) strategies. Table 3.5 lists the experiment design for each emotion category with the classifiers and training sets.

Table 3.5 Experiment design for each emotion category

Experiment	Classifier	Train set	Test set
Baseline	SVM	ET-train	ET-test
	BiLSTM	ET-train	ET-test
	BiLSTM	ET-train + All DS	ET-test
Threshold-based	BiLSTM	ET-train + AUG-*t* (BoW)	ET-test
	BiLSTM	ET-train + AUG-*t* (Word2Vec)	ET-test
	BiLSTM	ET-train + AUG-*t* (fastText)	ET-test
	BiLSTM	ET-train + AUG-*t* (GloVe)	ET-test
	BiLSTM	ET-train + AUG-*t* (USE)	ET-test
	BiLSTM	ET-train + AUG-*t* (InferSent-fastText)	ET-test
	BiLSTM	ET-train + AUG-*t* (InferSent-GloVe)	ET-test
Fixed Increment	BiLSTM	ET-train + AUG-*I* (BoW)	ET-test
	BiLSTM	ET-train + AUG-*I* (Word2Vec)	ET-test
	BiLSTM	ET-train + AUG-*I* (fastText)	ET-test
	BiLSTM	ET-train + AUG-*I* (GloVe)	ET-test
	BiLSTM	ET-train + AUG-*I* (USE)	ET-test
	BiLSTM	ET-train + AUG-*I* (InferSent-fastText)	ET-test
	BiLSTM	ET-train + AUG-*I* (InferSent-GloVe)	ET-test
	BiLSTM	ET-train + AUG-*I* (Random)	ET-test
Augmentation Strategy	BiLSTM	ET-train + AUG-*I* (CLUSTER)	ET-test
	BiLSTM	ET-train + AUG-*I* (MISCLASSIFIED)	ET-test

3.7 Evaluation Plan

The evaluation plan is to identify which vector representation and text augmentation strategy are better at augmenting relevant training data based on the performance of the emotion classifiers on the ET-test. The metric used to evaluate the performance of the classifiers is F1. For threshold-based experiments, the vector representation and threshold value for text augmentation are identified based on the highest F1 scores obtained from all emotion classifiers.

For the fixed increment experiments, we propose a stability measure to determine which vector representation can achieve the most stable performance in augmenting a fixed number of new training samples. For each emotion category, we set the baseline F1 score achieved by the BiLSTM classifier as the starting threshold for comparison and then count the total number of F1 scores by all classifiers trained on

the increment-based augmented sets (AUG-*I*) given each vector representation that are higher than the starting F1 threshold achieved. Next, we repeat the same process by continuously increasing the F1 from the starting threshold by 0.025 until no classifier is left with F1 score higher than the maximum threshold. The process is also repeated in the opposite direction by decreasing the starting F1 threshold by 0.025 until all F1 scores are higher than the minimum threshold. Finally, the results are used to determine the best and the most stable vector representation based on the distribution of the F1 scores. For example, if all F1 scores achieved by the emotion classifiers with each AUG-*I* from a specific vector representation are relatively higher than the other approaches, then it is considered as the most stable and the best approach. Finally, the best text augmentation strategy from PRIMARY, CLUSTER and MISCLASSIFIED is decided based on which strategy can achieve the highest F1 score for each emotion category.

3.8 Summary

This chapter explains the three-phase methodology to identify the best vector representation and strategy to augment more training samples for EmoTweet-28. Phase 1 (Data Preparation) and 2 (Text Augmentation) addressed how new self-labelled tweets are collected using distant supervision, and then compared to gold and silver standard seed sets to compute the similarity score using seven approaches to select relevant augmented training samples. In Phase 3 (Emotion Classification), we utilized the BiLSTM model to build binary classifiers to evaluate the effect of adding augmented training samples and identify which text augmentation strategy is better at augmenting more relevant training data.

CHAPTER 4

RESULTS & DISCUSSION

4.1 Introduction

The main goal of this chapter is to observe the performance of augmented training samples from seven different similarity approaches on the emotion classification task. In the previous chapter, we address the first research question by explaining the proposed text augmentation strategy.

- R1: How can new training data be augmented using similarity scoring approach to improve the quality of data collected from distant supervision method?

In this chapter, the results from all conducted experiments are used to address the second and third research questions:

- R2: What is the effect of using similarity scoring approach to augment more relevant training data on the performance of deep learning model for emotion classification?

- R3: Which text augmentation strategy is better at augmenting more relevant training data?

First, we present the results for six emotion categories on three different baseline implementations and the outcome from the seven vector representations with DS:ET-seed pair instances. Next, the effect of augmented training sets in threshold-based and fixed increment experiments are discussed and compared to select the most effective vector representation among the seven on augmenting new training data. Finally, we discuss and compare the results from the three proposed text augmentation strategies

using the most effective vector representation to determine the text augmentation strategy that is better at augmenting more relevant training data.

4.2 Baseline Comparison

Table 4.1 illustrates the F1 scores achieved by all baseline emotion classifiers. Baseline BiLSTM classifiers with ET-train for *happiness*, *anger*, *excitement* and *boredom* achieve higher F1 scores compared to baseline SVM classifiers with ET-train and BiLSTM classifiers combining ET-train and all DS. However, F1 scores achieved by baseline BiLSTM with ET-train for *desperation* and *indifference* scored the lowest among all. This is because deep learning algorithm cannot be leveraged when the training data are limited and SVM can perform slightly better than BiLSTM classifiers in a situation with limited training data. To be consistent, BiLSTM with ET-train is selected as the main baseline for this research for comparison in all our experiments because of high the F1 scores for most of the emotion categories. Also, we observed all baseline BiLSTM classifiers with ET-train and all DS have the worst performance across the board mainly because the data from DS is diverse and noisy compared to the original EmoTweet-28. Thus, we can conclude that poor performance is obtained if proper augmentation is not carried out.

Table 4.1 Baseline performance

Emotion	**Baseline SVM (ET-train)**	**Baseline BiLSTM (ET-train)**	**Baseline BiLSTM (ET-train + all DS)**
Happiness	0.499	**0.573**	0.200
Anger	0.308	**0.318**	0.199
Excitement	0.491	**0.563**	0.125
Boredom	0.632	0.**800**	0.011
Desperation	**0.167**	0.000	0.007
Indifference	**0.125**	0.000	0.031

4.3 Similarity Scoring Results

To assess qualitatively if the vector representations are actually returning DS tweets that are similar to the ET-seed, we analyzed the similarity scores returned by different vector representations for selected DS tweets to the ET-seed. Table 4.2 shows the similarity scores calculated for four unique original DS tweets with different context when compared to the same tweets labelled with '*happiness'* extracted from ET-seed using the seven vector representations. All seven vector representations obtain similarity scores close to 1 (most similar) for the first DS tweet, "*What a beautiful day! #joy https://t.co/koSNM2100a*" which almost exactly matches the ET-seed tweet, "*What a beautiful day http://t.co/PFj8H6wNEU*" after removing the related keyword hashtag "*#joy*" from DS tweet and URL from both tweets. The second DS tweet, "*What a wonderful day we had today. #happy #beautiful #day https://t.co/dXxG86ktdP*" conveys the same meaning as the ET-seed tweet but with different choice of words such as "*wonderful*" instead of "*beautiful*" and still obtains high similarity scores for all approaches. However, USE, InferSent fastText and BoW obtain lower scores for the third DS tweet containing slightly more additional information such as "*reading outside*". The final DS tweet, "*#Health #Happy #Life Clean Green Protein 5 pack - discounted 50% https://t.co/vuVZvTcBZT*" which has a totally different context obtains low similarity scores for all the vector representations. Therefore, we can conclude that tweets with similar context have a high similarity score with each other and vice versa.

Table 4.2 Similarity scoring results (examples with similar context and comparable length)

DS	ET-seed	Similarity score						
		USE	**Inf. GloVe**	**Inf. fastText**	**GloVe**	**W2V**	**fastText**	**BoW**
What a beautiful day! #joy https://t.co/koSNM2100a	*What a beautiful day http://t.co/PFj8H6wNEU*	1	0.847	1	1	1	1	1
What a wonderful day we had today . #happy #beautiful #day https://t.co/dXxG86ktdP	*What a beautiful day http://t.co/PFj8H6wNEU*	0.802	0.827	0.824	0.975	0.91	0.903	0.73
Beautiful day for reading outside. #happy #serene #needed https://t.co/k30Lwmgtsz	*What a beautiful day http://t.co/PFj8H6wNEU*	0.586	0.702	0.541	0.916	0.743	0.744	0.471
#Health #Happy #Life Clean Green Protein 5 pack - discounted 50% https://t.co/vuVZvTcBZT	*What a beautiful day http://t.co/PFj8H6wNEU*	0.035	0.433	0.123	0.718	0.485	0.498	0

Table 4.3 illustrated the similarity scores calculated for DS and ET-seed tweets with similar context but a greater difference in the length of sentence. The first DS tweet has 43 words while the paired ET-seed tweet has 16 words and both tweets are referring to the celebrity related context. The second DS tweet has 18 words while the paired ET-seed tweet has six words. Both tweets are about the Easter. USE, InferSent fastText and BoW produce relatively low scores for these two DS:ET-seed pairs compared to InferSent GloVe, GloVe, W2v and fastText. We observed that USE, InferSent fastText

and BoW are more likely to give relatively lower similarity scores compared to the other.

Table 4.3 Similarity scoring results 2

DS	ET-seed	Similarity score						
		USE	**Inf. GloVe**	**Inf. fastText**	**GloVe**	**W2V**	**fastText**	**BoW**
Celebrating her existence in the Kpop World which made all of us see the beauty of life thru her. Whatever hardships ur experiencing , if you stan a person like our bay everything will be vanished! #charisma #energy #happiness #idol #love #LisaIsThat Girl 👑 #NineYears WithLalisa	*Good Morning America of Nicki Minaj was great when she perform Moment 4 Life @NICKIMINAJ #NickiMinaj*	0.33	0.774	0.591	0.946	0.807	0.813	0.165
Have a Happy Easter! May your Easter be blessed with love, joy and abundance. #KMPBeautyAndBodySolutions #Happy Easter https://t.co/mPCBMpR285	*Hangover and Easter egg :) xx*	0.422	0.715	0.543	0.875	0.723	0.729	0.4

4.4 Effect of Augmented Training Set

In this section, the effect of the adding augmented training samples into the emotion classifiers are discussed using the results obtained from the experiments. The results are divided into two sections which are threshold-based and fixed increment experiments.

4.4.1 Threshold-based Experiments Results

The results from threshold-based experiments are presented in two parts. The first part shows the size of all augmented training sets, and the second part focuses on the performance of the augmented sets on the emotion classification task. Table 4.4 illustrates the size of all augmented training sets for the selected six emotion categories. Each augmented set is labelled with the emotion category, vector representation and the threshold value (t) for selecting the potential DS tweets based on the computed similarity score. The augmented sets with USE have relatively smaller sizes compared with other approaches. For instance, the number of *happiness* tweets in USE augmented set with similarity score higher than the threshold value of 0.5 is 26,650 (7.353%) out of 362,435 *happiness* DS happiness tweets. The augmented sets obtained from BoW are the second smallest after USE. For example, the number of augmented *happiness* tweets for BoW when the threshold equals to 0.5 is 30,524 (8.422%). The sizes of the augmented sets obtained from other approaches with 0.5 threshold exceed 80% of the entire DS tweets sample for all other emotion categories.

Table 4.4 Size of augmented training sets

Emotion	*t*	USE	Inf GloVe	Inf fastText	GloVe	W2V	fastText	BoW
Happiness	0.9	14	6	98	248,009	20,711	4,850	22
	0.8	100	92,510	1,596	324,803	203,474	179,512	86
	0.7	670	285,566	18,853	349,379	321,909	324,856	516
	0.6	4,934	338,890	205,370	357,192	357,245	356,671	4,090
	0.5	26,650	355,272	326,112	359,874	361,818	360,903	30,524
	0	362,435						
Anger	0.9	0	1	2	35,388	7,853	3,284	2
	0.8	0	15,374	56	39,181	31,489	30,093	4
	0.7	6	35,166	3,800	40,107	38,704	38,650	37
	0.6	85	39,213	26,958	40,519	40,381	40,355	600
	0.5	1,154	40,305	38,004	40,638	40,678	40,639	4,896
	0	40,737						
Excitement	0.9	0	1	39	41,818	3,400	830	0
	0.8	1	13,155	294	48,723	33,432	28,521	8
	0.7	14	40,015	2,035	50,182	47,123	46,208	69
	0.6	175	47,257	24,302	50,905	50,596	50,155	487
	0.5	1,834	49,784	45,342	51,121	51,141	51,020	4,565
	0	51,269						
Boredom	0.9	21	55	92	59,523	6,586	4,319	128
	0.8	88	15,049	573	70,613	47,375	43,732	327
	0.7	326	51,796	4,033	73,787	67,313	66,426	821
	0.6	1,329	66,938	33,260	74,861	73,963	73,768	1,510
	0.5	5,450	72,920	62,439	75,226	75,311	75,172	4,859
	0	75,634						
Desperation	0.9	0	4	72	28,489	6,542	4,016	0
	0.8	13	9,938	73	31,408	24,022	22,464	9
	0.7	74	22,324	3,906	32,172	29,971	29,477	9
	0.6	120	28,273	15,544	32,524	32,048	31,945	96
	0.5	473	30,865	25,602	32,744	32,813	32,746	826
	0	32,949						
Indifference	0.9	2	7	7	6,161	1,521	968	6
	0.8	3	2,411	41	6,802	5,340	5,012	12
	0.7	10	5,565	984	6,976	6,617	6,574	22
	0.6	34	6,702	3,831	7,075	6,991	7,007	109
	0.5	152	7,025	5,809	7,116	7,117	7,103	544
	0	7,142						

Table 4.5 illustrate the threshold-based experiment results in term of F1 scores generated for each augmented set based on different threshold values and vector representations. F1 scores highlighted in green are the ones higher than the baseline F1 score for the respective emotion category. In *happiness*, the highest F1 score of 0.603 is obtained through the addition of the augmented set from USE with the threshold of 0.7. The F1 score of 0.436 is the highest in *anger*, which is obtained from the USE

augmented set with the threshold value of 0.5. The performance of the *excitement* classifier shows no significant improvement from all augmented sets except the augmented set from BoW with the threshold of 0.7 (F1 = 0.584). In *desperation*, the highest F1 score of 0.182 is obtained in multiple augmented sets using USE, InferSent GloVe and BoW. There is no improvement in performance from all augmented sets for *boredom* because the *boredom* baseline classifier already shows a relatively high F1 score of 0.8. However, a few augmented sets from USE, InferSent fastText and BoW manage to maintain the *boredom* classifier performance at F1 score of 0.8, which is the same as baseline performance. In *indifference*, the augmented set using BoW threshold of 0.6 achieves the F1 score of 0.25, which is the highest among all augmented set in the same emotion category.

According to the threshold-based experiment results, the best vector representation among the seven is USE because it has the most augmented sets from different emotion categories with relatively higher F1 scores. The selection of the best approach is mainly based on *happiness*, *anger* and *excitement* because the number of positive instances in the ET-test for *desperation*, *boredom* and *indifference* are extremely low. Therefore, the F1 scores obtained from these three emotion categories with low frequency are not as promising as the high frequency emotions.

Table 4.5 Threshold-based experiments results

Emotion	*t*	USE	Inf GloVe	Inf fastText	GloVe	W2V	fastText	BoW
Happiness (Baseline = 0.573)	0.9	0.574	0.564	0.569	0.231	0.457	0.533	0.513
	0.8	0.579	0.489	0.533	0.201	0.228	0.259	0.584
	0.7	**0.603**	0.245	0.464	0.168	0.207	0.216	0.563
	0.6	0.582	0.203	0.287	0.202	0.199	0.206	0.509
	0.5	0.445	0.203	0.220	0.193	0.196	0.188	0.369
Anger (Baseline = 0.318)	0.9	-	0.374	0.442	0.212	0.350	0.375	0.413
	0.8	-	0.354	0.372	0.198	0.237	0.254	0.386
	0.7	0.317	0.238	0.389	0.198	0.207	0.199	0.391
	0.6	0.420	0.222	0.284	0.206	0.198	0.212	0.389
	0.5	**0.436**	0.217	0.204	0.196	0.200	0.195	0.325
Excitement (Baseline = 0.563)	0.9	-	0.519	0.568	0.135	0.433	0.503	-
	0.8	0.531	0.311	0.491	0.138	0.149	0.176	0.559
	0.7	0.517	0.191	0.433	0.127	0.131	0.141	**0.584**
	0.6	0.530	0.135	0.198	0.133	0.135	0.133	0.531
	0.5	0.439	0.122	0.145	0.131	0.131	0.125	0.312
Desperation (Baseline = 0.000)	0.9	-	**0.182**	0.174	0.008	0.020	0.011	-
	0.8	0.167	0.026	0.000	0.008	0.010	0.012	**0.182**
	0.7	**0.182**	0.010	0.000	0.008	0.008	0.009	**0.182**
	0.6	0.000	0.010	0.015	0.011	0.008	0.008	0.000
	0.5	0.074	0.008	0.008	0.007	0.007	0.006	0.000
Boredom (Baseline = 0.800)	0.9	0.800	0.750	0.800	0.010	0.126	0.110	0.800
	0.8	0.800	0.085	0.273	0.010	0.015	0.014	0.667
	0.7	0.632	0.018	0.125	0.010	0.011	0.011	0.706
	0.6	0.267	0.011	0.020	0.010	0.010	0.010	0.375
	0.5	0.099	0.011	0.011	0.010	0.010	0.010	0.082
Indifference (Baseline = 0.000)	0.9	0.167	0.000	0.000	0.038	0.108	0.113	0.167
	0.8	0.167	0.118	0.133	0.026	0.036	0.040	0.000
	0.7	0.000	0.030	0.098	0.022	0.032	0.021	0.118
	0.6	0.000	0.028	0.039	0.025	0.019	0.024	**0.250**
	0.5	0.235	0.022	0.025	0.020	0.013	0.025	0.133

The results obtained from threshold-based experiments is observed to have a relation with the size of the augmented set. The augmented set with a smaller size shows higher F1 score compared with the augmented set with a larger size containing more noise, which may deteriorate the performance of the binary classifier during the training phase. For instance, similarity scoring using USE and BoW tend to produce smaller augmented sets even with lower thresholds, thus these two approaches are less affected by noise compared to the five other approaches that have a higher tendency to inflate the similarity scores. However, we do not know if the observation would remain the same when the size of the augmented sets is fixed across the board. Therefore, the fixed

increment experiments are designed to solve this issue by limiting the total number of most similar tweets extracted from DS tweets based on fixed augmented sample sizes.

4.4.2 Fixed Increment Experiments Results

The results of fixed increment experiments are presented in two parts. The first part focuses on the performance of all augmented sets on emotion classifiers. Only the main findings are presented in this section and the full results of all augmented sets in the fixed increment experiments are included in Appendix A. The second part reports the stability measure for each vector representation in the fixed increment experiments.

Figure 4.1 illustrates the F1 scores from adding fixed increments of the DS augmented sets from the seven vector representations and a random set against the *happiness* baseline. Most of the augmented sets obtained using InferSent GloVe used for the *happiness* classifier achieve F1 score higher than the baseline (F1 = 0.573). The augmented sets obtained using random augmentation without similarity scoring approach have the worst performance compared to the other approaches and barely shows any improvement from the baseline. Thus, adding augmented tweets produced by the similarity scoring approach helps the classifiers to learn better as opposed to adding random training data.

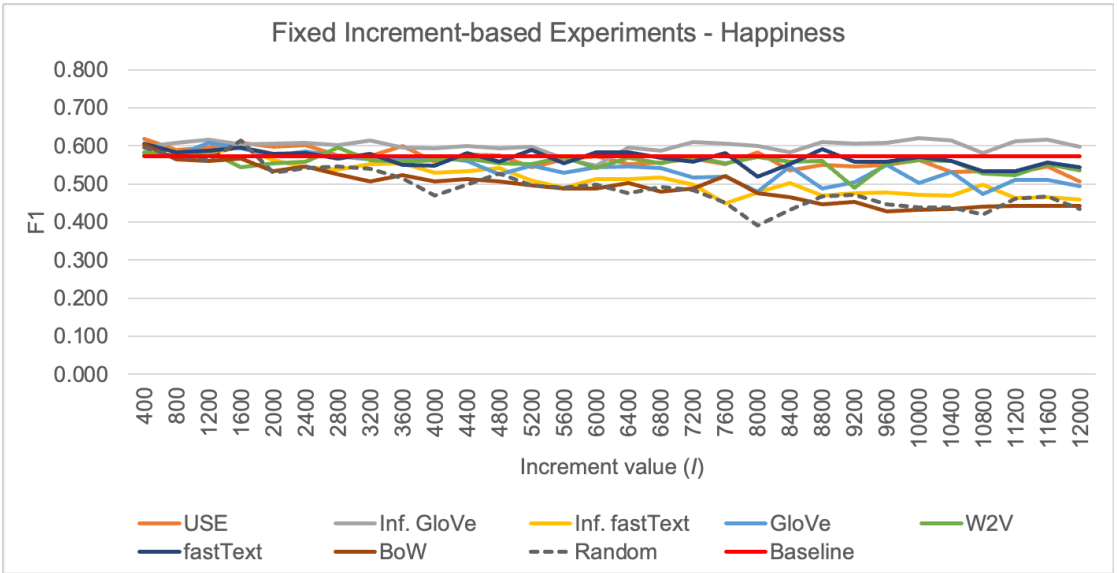

Figure 4.1 F1 scores from fixed increment experiment for happiness

Figure 4.2 illustrates the results for *anger*. The augmented sets obtained using USE produce the most F1 scores higher than the baseline (F1 = 0.318). Similarity, the random augmented sets barely show any improvement from the baseline.

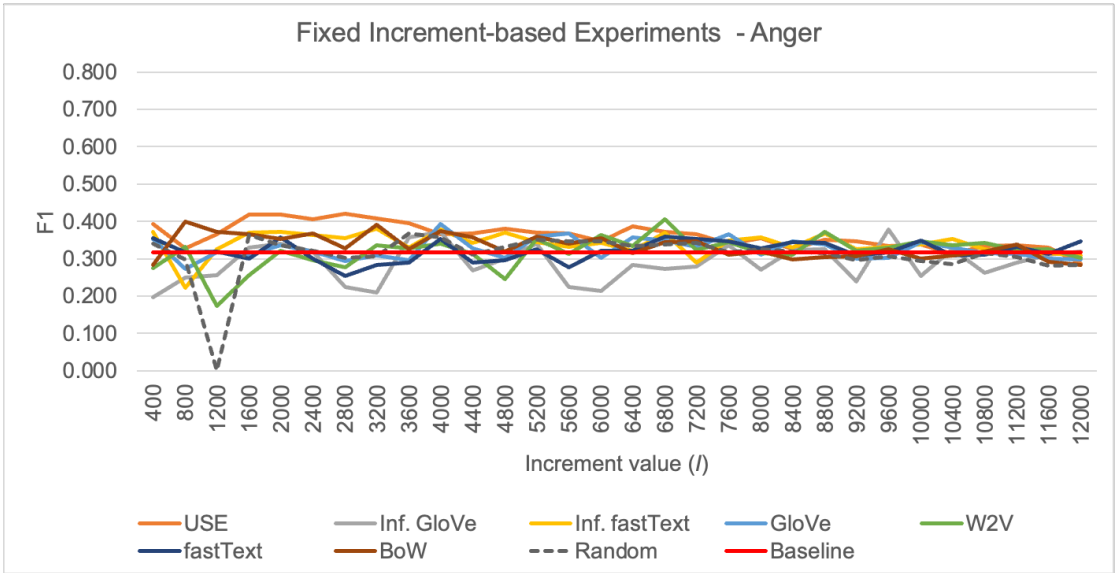

Figure 4.2 F1 scores from fixed increment experiment for anger

Figure 4.3 illustrates the results for *excitement*. In this emotion category, all augmented sets are unable to improve the performance of the classifier as when compared to the baseline F1 score of 0.563. Nonetheless, the augmented sets obtained from InferSent GloVe have the highest F1 scores across all approaches while the

random augmented sets remain mostly the lowest. *Excitement* is one of the high frequency emotion categories, but it does not show any significant improvement from its baseline for all augmented sets when compared to *happiness* and *anger*. Therefore, we further investigate the similarity scoring results for excitement using the InferSent GloVe which reported the most frequent highest F1 score for each increment.

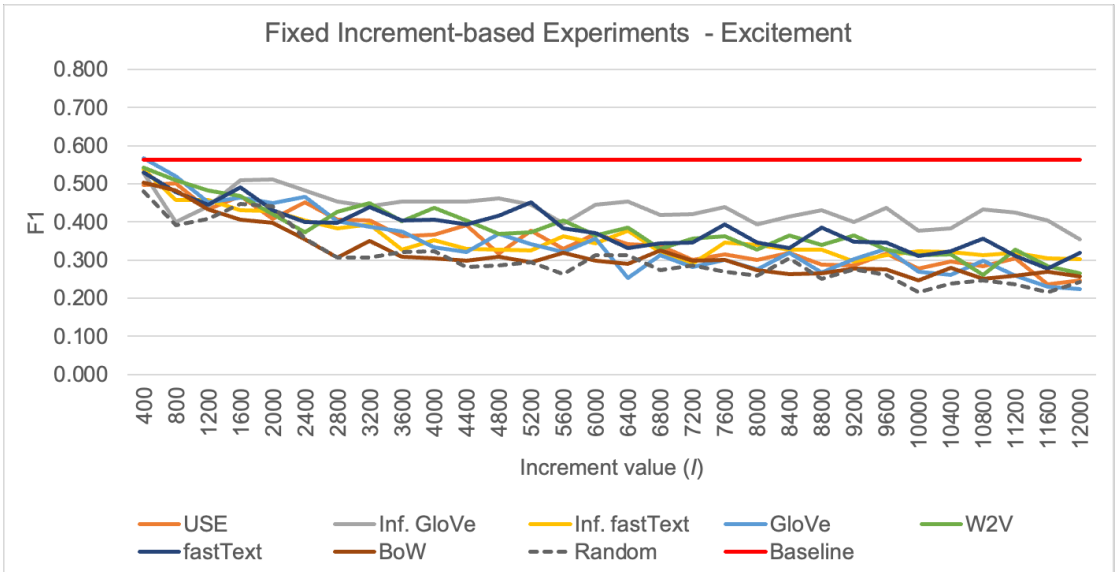

Figure 4.3 F1 scores from fixed increment experiment for excitement

According to Table 4.6, the DS tweet is paired with the most similar tweet in ET-seed for *excitement*. However, we observe the DS tweets contain excessive information when compared to their paired ET-seed tweets might have introduced greater amount of noise into the augmented set. For example, the first DS and ET-seed tweets are both expressing excitement for the college football game, but the DS tweet contains extra information which does not provide insight for the expression of excitement such as "*Don't leave me in an @uber w/ time sitting back here*". Therefore, the emotion classifier for *excitement* cannot appropriately utilize the examples from augmented set to identify the expression of excitement.

Table 4.6 Similarity scoring results for excitement using InferSent GloVe

DS	ET-seed	Similarity score
Don't leave me in an @uber w/ time sitting back here. So excited for tomorrow & the game! #natty #thrilled & will be back on my travel game after this 1 ❤️‍🩹 I love college football but will say watched a pro Kansas City game today which was cra cra like college a bit before I left	*I CANNOT wait until college football! Sept 14th, flying down with my brother to the Florida v Tennessee game, it's gonna be titss*	0.875
Can't wait to play you all the new music coming out of this room. In the mean time check out my other 2 records that were birthed here 😁 #myhappyplace #newmusicontheway #studio #music #excited #dmills https://t.co/n9aY78roZ3	*back in the studio!! creating new music that I can't wait to share with all of you*	0.867
I can not wait to start the year off and welcome everyone one to 2020! Hope you all enjoyed a little break. It is my pleasure to create and design beautiful bloom and no better way to start the year than opening tomorrow (14th of January) #happynewyear2020 #excited #melbourne	*I can't wait for october! 2@justinbieber concerts! :) & I get to visit mexico city for the 1st time! Great way to celebrate my sweet 16!*	0.864
I just want to say #thankyou for all the #support i've gotten for the #release of #YeFennyRemastered Demo, my phone has been going crazy! Can't wait to see people play it and have fun. 👊😎👊 Thank you to all my new followers as well, can't wait to show more updates. #excited. https://t.co/w0gcxciXzh	*Really want the Break Free video to be done already!!! I can't wait for you to see it. So so excited about it.* 💜	0.862

Next, we move on to the results from fixed increment experiment for low frequency emotion categories. Figure 4.4 illustrates the results for *boredom*. As the *boredom* baseline has a fairly high F1 score of 0.8, none of the augmented set in this category is able to beat the baseline score. InferSent GloVe has the most augmented sets with the highest F1 scores compared to the others while the random augmentation approach again has the most augmented sets with the lowest F1 scores.

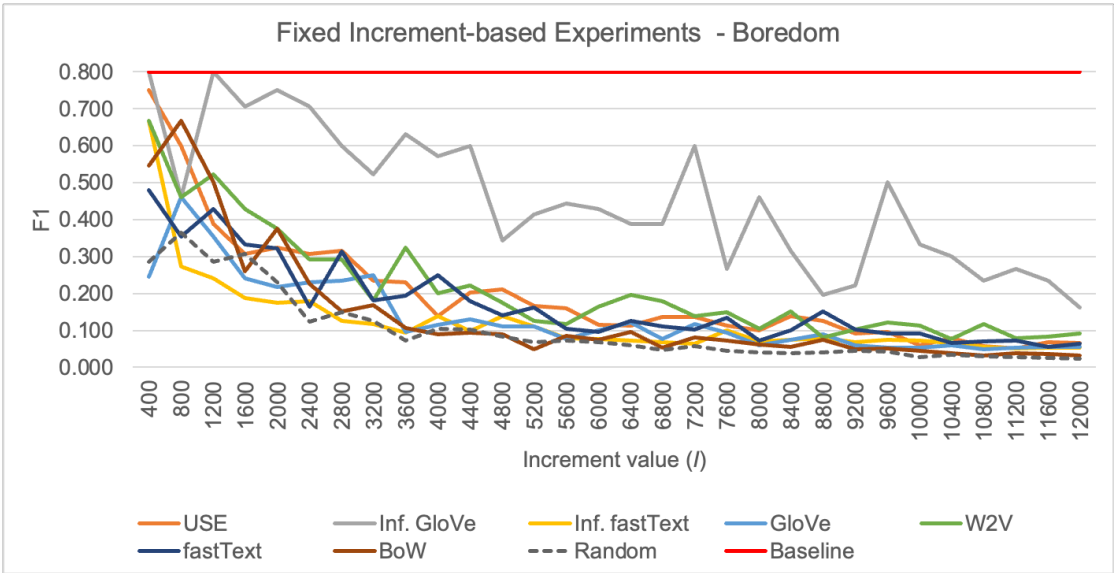

Figure 4.4 F1 scores from fixed increment experiment for boredom

Figure 4.5 illustrates the results for *desperation*. Unlike *boredom*, the baseline F1 score for *desperation* is 0. Therefore, almost all augmented sets show performance improvement against the baseline classifier. The largest improvement can be observed through InferSent GloVe augmented set containing 1200 new augmented instances with the F1 score of 0.125.

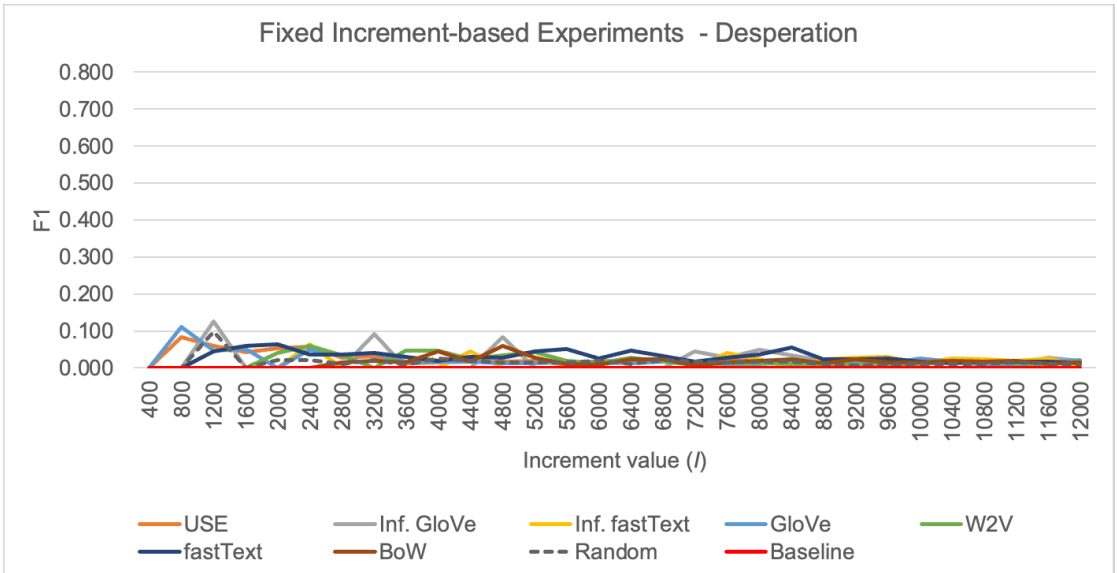

Figure 4.5 F1 scores from fixed increment experiment for desperation

Figure 4.6 illustrates the results for *indifference*. The baseline F1 score for indifference is also 0. Therefore, all augmented sets show improvement from the baseline classifier. The largest increase in performance is from the InferSent GloVe augmented set with 1,600 new instances and the F1 score of 0.25. Ultimately, we observed augmented sets using InferSent GloVe achieve higher F1 score for all emotion categories except *anger*, thus we can conclude that InferSent GloVe is the better vector representation so far. Next, to truly identify the best approach for the similarity scoring with consistent performance, the findings from fixed increment experiments are further evaluated using a stability measure.

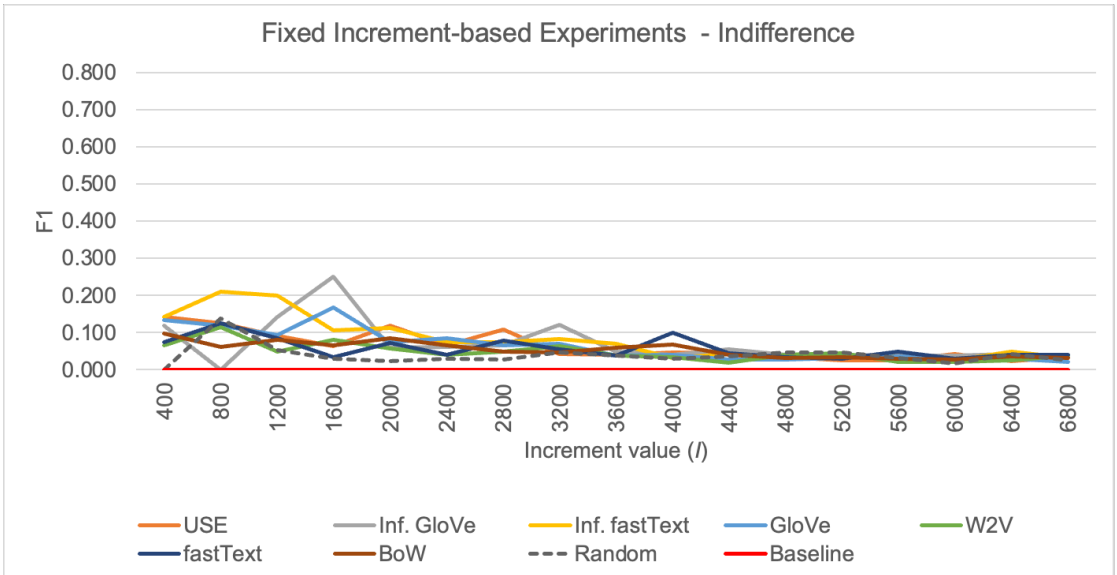

Figure 4.6 F1 scores from fixed increment experiment for indifference

Table 4.7 illustrates the stability of all the augmented sets for happiness using different vector representations. All values presented in Table 4.7 represent the number of the augmented sets for each specific augmentation approach with F1 scores higher than the threshold F1 score. The starting threshold F1 score is first defined by the baseline F1 score (threshold F1 cell highlighted in yellow) for each emotion category, and then the following threshold F1 scores are defined by repeatedly adding and subtracting 0.025. The value 0.025 was chosen is because it can provide a better

observation on the stability of the different vector representations and the performance of the augmented sets as the F1 scores achieved from each augmented set are fairly close to each other. The values highlighted in yellow represent the number of augmented sets that are higher than the baseline F1 score for each augmentation approach and emotion category. The values highlighted in orange are the number of augmented sets with F1 scores higher than the increased threshold F1 scores while values highlighted in green are the number of augmented sets with F1 scores higher than the decreased threshold F1 scores before reaching the maximum number of augmented sets for each approach which is 30. The goal of the stability measure is to identify which approach has the highest number of augmented sets with consistently high F1 scores. The general intuition of the stability visualization is to identify the similarity-based augmentation approach despite different sizes that can consistently produce better or at par performance as the baseline classifier without augmentation and at the same time would not cause significant drop in performance. Therefore, the approach with the most cells highlighted in orange, and the least cells highlighted in green is selected as the most stable approach to augment better training instances.

Table 4.7 Stability measure for happiness augmented sets

Threshold F1	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
0.673	0	0	0	0	0	0	0	0
0.648	0	0	0	0	0	0	0	0
0.623	0	0	0	0	0	0	0	0
0.598	4	20	1	1	0	1	1	1
0.573	11	28	4	6	4	13	1	2
0.548	22	30	7	13	24	25	4	4
0.523	29	30	12	20	29	29	8	9
0.498	30	30	18	26	29	30	14	12
0.473	30	30	23	30	30	30	20	17

In *happiness*, out of 30 augmented set sizes, 28 scored above the baseline F1 score of 0.573 for InferSent GloVe. When the F1 baseline is increased by 0.025, we observe that as many as 20 sets still manage to score above 0.598 (+0.025) although

none is above 0.623 (+0.05). Nonetheless, the *happiness* classifier remains the most stable by adding augmented sets from InferSent GloVe as observed by all 30 augmented sets yielding the F1 score of above 0.548 (-0.025). In short, all the F1 scores generated using the InferSent GloVe augmented sets are within the range of +0.05 (F1 < 0.624) and -0.025 (F1 > 0.548) from the baseline F1 score.

Table 4.8 illustrates the stability of all the augmented sets for *anger* using different vector representations. Out of 30 augmented set sizes, 28 scored above the baseline F1 score of 0.318 for USE. When the F1 baseline is increased to 0.418 (+0.1), only USE still manages to have 2 sets above the increased baseline. In addition, the *anger* classifier remains the most stable by adding augmented sets from USE as observed by all 30 augmented sets yielding the F1 score of above 0.268 (-0.05). In short, all the F1 scores generated using USE augmented sets are within the range of +1 (F1 < 0.443) and -0.05 (F1 > 0.548) from the baseline F1 score.

Table 4.8 Stability measure for anger augmented sets

Threshold F1	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
0.443	0	0	0	0	0	0	0	0
0.418	2	0	0	0	0	0	0	0
0.393	7	0	0	0	1	0	1	0
0.368	11	2	7	2	2	0	5	0
0.343	20	3	15	9	6	10	12	6
0.318	28	12	25	15	19	16	20	15
0.293	29	14	28	29	25	25	27	26
0.268	30	20	29	30	27	29	30	29
0.243	30	24	29	30	29	30	30	29
0.218	30	27	30	30	29	30	30	29

Table 4.9 illustrates the stability of all the augmented sets for *excitement* using different vector representations. None of the augmented sets manage supersede the baseline F1 score of 0.563 except for 1 augmented set from GloVe. However, the *excitement* classifier remains the most stable by adding augmented sets from InferSent GloVe as observed by all 30 augmented sets yielding the F1 score of above 0.338 (-

0.225). The range of all the F1 scores generated using InferSent GloVe are within -0.025 (F1 < 0.538) and -0.225 (F1 > 0.338) from the baseline F1 score.

Table 4.9 Stability measure for excitement augmented sets

Threshold F1	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
0.563	0	0	0	1	0	0	0	0
0.538	0	0	1	1	1	0	0	0
0.513	0	1	1	2	1	1	0	0
0.488	2	3	1	2	2	2	1	0
0.463	3	4	1	4	4	3	2	1
0.438	4	15	3	6	5	6	2	3
0.413	5	22	5	6	8	8	3	3
0.388	9	27	7	7	11	14	5	5
0.363	13	29	9	10	18	17	5	5
0.338	14	30	14	12	21	23	7	6
0.313	20	30	26	18	27	27	9	9
0.288	24	30	30	21	27	29	19	14
0.263	28	30	30	25	29	30	26	21
0.238	29	30	30	28	30	30	30	26
0.213	30	30	30	30	30	30	30	30
0.188	30	30	30	30	30	30	30	30

Table 4.10 illustrates the stability of all the augmented sets for *boredom* using different vector representations. None of the augmented sets can achieve higher performance than the baseline F1 score of 0.8. However, the *boredom* classifier remains the most stable by adding augmented sets from InferSent GloVe as observed by all 30 augmented sets yielding the F1 score of above 0.15 (-0.65). The range of all the F1 scores generated using InferSent GloVe are within -0.65 (F1 > 0.15) from the baseline F1 score of 0.8.

Table 4.10 Stability measure for boredom augmented sets

Threshold F1	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
0.8	0	0	0	0	0	0	0	0
0.775	0	2	0	0	0	0	0	0
0.75	1	3	0	0	0	0	0	0
0.725	1	3	0	0	0	0	0	0
0.7	1	5	0	0	0	0	0	0
0.675	1	5	0	0	0	0	0	0
0.65	1	5	1	0	1	0	1	0
0.625	1	6	1	0	1	0	1	0
0.6	2	9	1	0	1	0	1	0
0.575	2	9	1	0	1	0	1	0
0.55	2	10	1	0	1	0	1	0
0.525	2	10	1	0	1	0	2	0
0.5	2	12	1	0	2	0	3	0
0.475	2	12	1	0	2	1	3	0
0.45	2	14	1	1	3	1	3	0
0.425	2	16	1	1	4	2	3	0
0.4	2	17	1	1	4	2	3	0
0.375	3	19	1	1	5	2	4	0
0.35	3	19	1	2	5	3	4	1
0.325	3	21	1	2	5	4	4	1
0.3	7	23	1	2	6	6	4	2
0.275	7	23	1	2	8	6	4	4
0.25	7	25	2	2	8	6	5	4
0.225	9	27	3	7	8	7	6	5
0.2	11	28	3	8	9	7	6	5
0.175	11	29	5	8	13	10	6	5
0.15	13	30	6	8	17	13	8	6
0.125	18	30	9	9	19	16	8	7
0.1	21	30	11	15	25	20	9	10
0.075	25	30	17	21	30	24	15	11
0.05	30	30	30	29	30	30	22	17
0.025	30	30	30	30	30	30	30	29

Table 4.11 illustrates the stability of all the augmented sets for *desperation* using different vector representations. The baseline F1 score for *desperation* is 0. Therefore, the most stable approach for augmenting instance of desperation only can be selected based on augmented sets with the most number of high F1 scores which is InferSent GloVe. When the F1 benchmark is increased to 0.125 (+0.125), only InferSent GloVe still manages to have one set above the increased benchmark. The range of all the F1 scores generated using InferSent GloVe are within +0.15 (F1 < 0.15) from the baseline F1 score of 0.

Table 4.11 Stability measure for desperation augmented sets

Threshold F1	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
0.15	0	0	0	0	0	0	0	0
0.125	0	1	0	0	0	0	0	0
0.1	0	1	0	1	0	0	0	0
0.075	1	3	0	1	0	0	0	1
0.05	4	4	1	3	1	4	1	1
0.025	7	9	7	7	8	18	3	2
0	-	-	-	-	-	-	-	-

Table 4.12 illustrates the stability of all the augmented sets for *indifference* using different vector representations. The baseline F1 score for *indifference* is also 0. Therefore, the most stable approach for augmenting indifference emotion instances is InferSent GloVe because when the F1 benchmark is increased to 0.25 (+0.25). Only InferSent GloVe still manages to have one set above the increased benchmark. The range of all the F1 scores generated using InferSent GloVe are within +0.275 (F1 < 0.275) from the baseline F1 score of 0.

Table 4.12 Stability measure for indifference augmented sets

Threshold F1	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
0.3	0	0	0	0	0	0	0	0
0.275	0	0	0	0	0	0	0	0
0.25	0	1	0	0	0	0	0	0
0.225	0	1	0	0	0	0	0	0
0.2	0	1	1	0	0	0	0	0
0.175	0	1	2	0	0	0	0	0
0.15	0	1	2	1	0	0	0	0
0.125	2	2	3	2	0	1	0	1
0.1	4	4	5	3	1	2	0	1
0.075	5	4	6	5	2	4	3	1
0.05	7	8	9	8	5	7	8	2
0.025	16	16	16	16	14	17	17	13
0	-	-	-	-	-	-	-	-

Table 4.13 illustrates the most stable vector representation for each emotion category, and we can observe that InferSent GloVe is most stable approach for all emotion category except for anger to augment more relevant training instances. USE is the most stable for approach *anger*.

Table 4.13 The most stable vector representation for each emotion category

Emotion	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
Happiness		✓						
Anger	✓							
Excitement		✓						
Boredom		✓						
Desperation		✓						
Indifference		✓						

4.5 Text Augmentation Strategy

In the section, the results from three different text augmentation strategies are discussed and compared to identify which strategy is better at augmenting more relevant training samples. Table 4.14 illustrates the size of seed sets after using PRIMARY, CLUSTER and MISCLASSIFIED strategies.

Table 4.14 Size of seed sets for primary, clustering and misclassified augmentation strategies

Emotion	PRIMARY-seed	CLUSTER-seed	MISCLASSIFIED-seed
Happiness	1024	292	127
Anger	776	238	153
Excitement	417	77	58
Boredom	141	20	2
Desperation	85	22	10
Indifference	97	21	11

The purpose of generating CLUSTER-seed sets for all emotion category is to have the classifier focus on learning from the rare instances from the PRIMARY-seed sets which is the original ET-seed. The MISCLASSIFIED-seed sets are generated based on misclassified positive label instances in the ET-test for each emotion category from the respective baseline classifier. Therefore, MISCLASSIFIED-seed sets are used to help the classifiers to learn from past mistakes by augmenting more training instances similar to all previously misclassified emotion instances.

Figure 4.7 illustrates the text augmentation strategy results for *happiness*. The vector representation chosen to conduct the experiments is InferSent GloVe because it is proven to be the most stable approach to augment more relevant training instances for most of the emotion categories. PRIMARY indicates all the results from augmented sets generated using the original ET-seed (PRIMARY-seed). CLUSTER and MISCLASSIFIED represent the results from augmented sets generated using CLUSTER-seed and MISCLASSIFIED-seed. The full results from the text augmentation strategy experiments are presented in Appendix B. In happiness, among all the augmented sets, MISCLASSIFIED has the highest F1 score of 0.624 when the size of the augmented set is 7,200. Therefore, misclassified strategy is better at augmenting more relevant *happiness* instances than the other strategies.

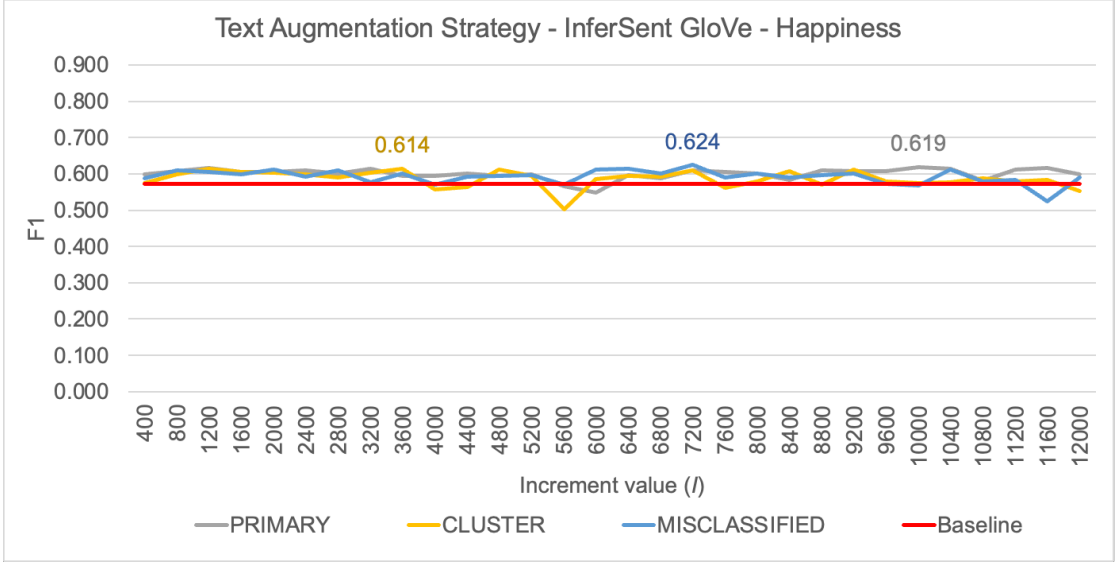

Figure 4.7 F1 score based on three augmentation strategies in fixed increments for happiness

Figure 4.8 illustrates the text augmentation strategy results for *anger*. MISCLASSIFIED has the highest F1 score of 0.408 among all augmented sets when the size of the augmented set is 4,000. Therefore, misclassified strategy is better at augmenting more relevant *anger* instances than the others.

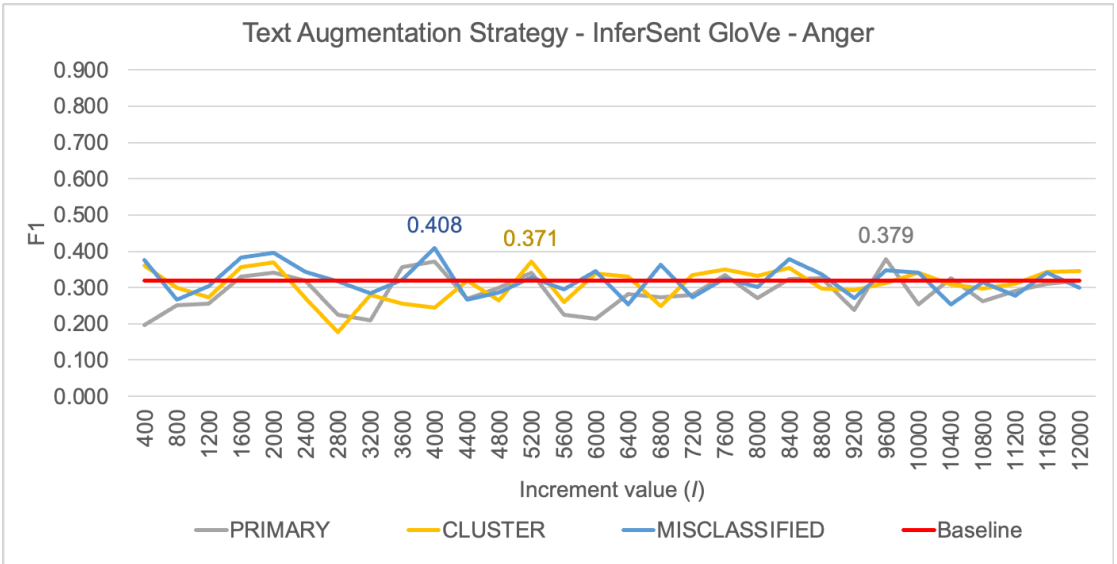

Figure 4.8 F1 scores based on three augmentation strategies in fixed increments for anger

Figure 4.9 illustrates the text augmentation strategy results for *excitement*. All augmented sets from PRIMARY, CLUSTERING, MISCLASSIFIED struggle to improve the F1 score from the baseline performance which is 0.563. The closest F1 score to the baseline is from MISCLASSIFIED with the augmented set size of 3,200 which is 0.551. Therefore, misclassified strategy is still better at augmenting more relevant *excitement* instances than the other strategies.

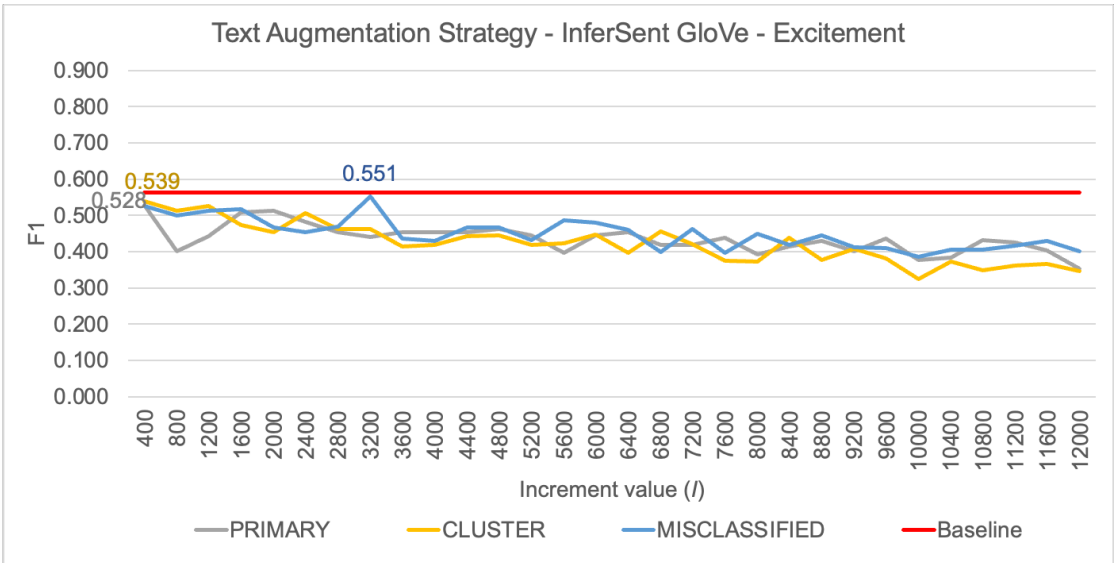

Figure 4.9 F1 scores based on three augmentation strategies in fixed increments for excitement

Figure 4.10 illustrates the text augmentation strategy results for *boredom*. Both PRIMARY and MISCLASSIFIED have the peak F1 score of 0.8 which is same as the baseline. However, the augmented set from CLUSTER with the augmented set size of 2,000 has achieves even higher F1 score of 0.857. Therefore, clustering strategy is better at augmenting more relevant *boredom* instances than the other strategies.

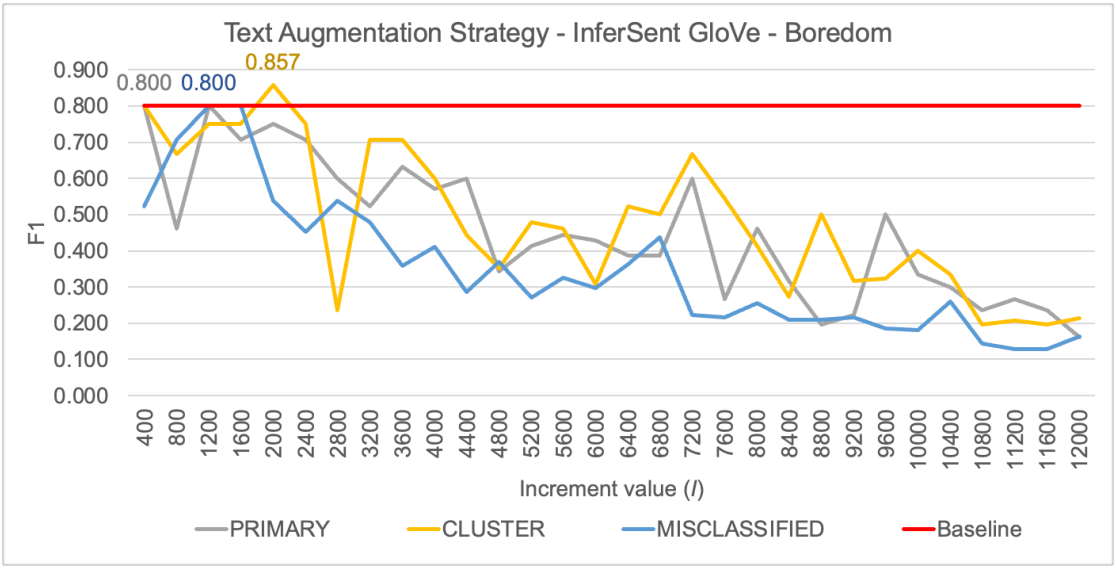

Figure 4.10 F1 scores based on three augmentation strategies in fixed increments for boredom

Figure 4.11 illustrates the text augmentation strategy results for *desperation* with fixed increments of the augmented set size. MISCLASSIFIED augmented set with the augmented set size of 1,200 has the highest F1 score of 0.154. Therefore, misclassified strategy is better at augmenting more relevant *desperation* instances than the other strategies.

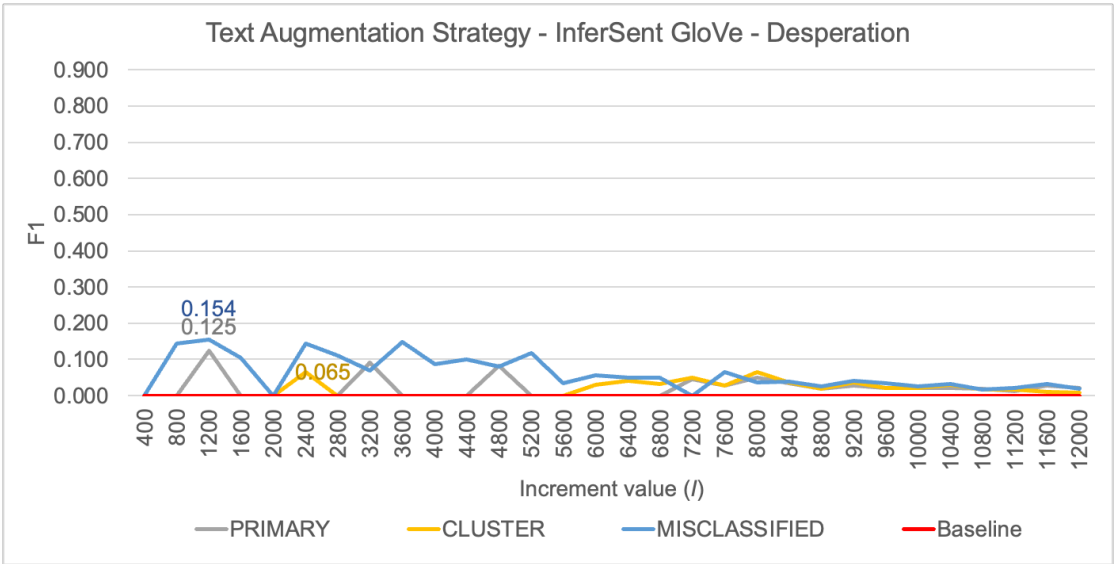

Figure 4.11 F1 score based on three augmentation strategies in fixed increments for desperation

Figure 4.12 illustrates the text augmentation strategy results for *indifference* with fixed increments of the augmented set size. PRIMARY with augmented set size of 1,600 increment value achieves the highest F1 score of 0.25. Both clustering and misclassified strategies are unable to further improve the F1 score from PRIMARY.

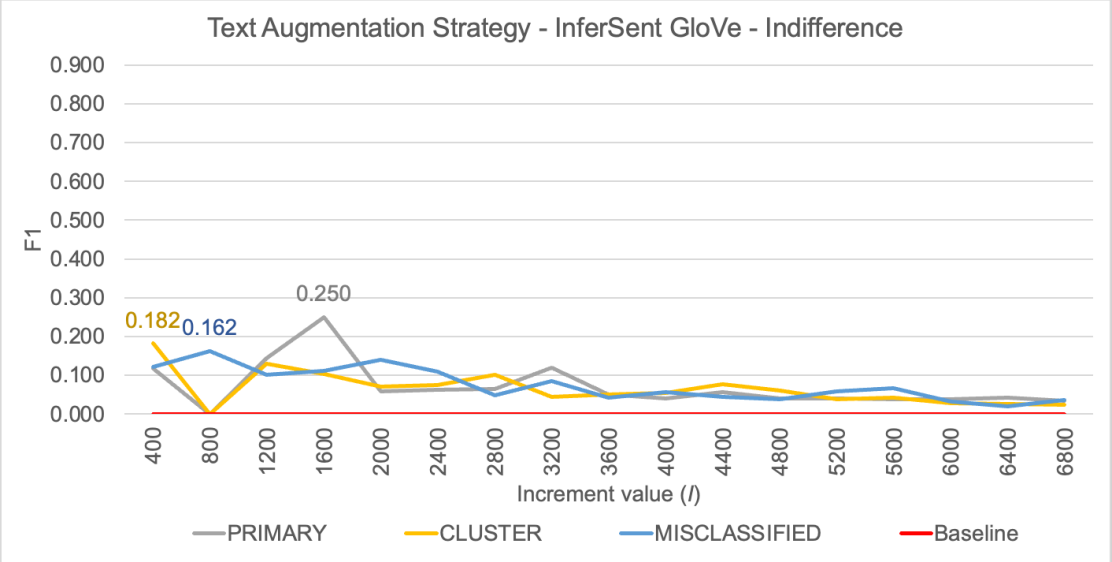

Figure 4.12 F1 scores based on three augmentation strategies in fixed increments for indifference

According to Table 4.15, MISCLASSIFIED obtained four out of six highest F1 scores across the board. Therefore, the misclassified text augmentation strategy is selected as the most suitable to augment more relevant instances.

Table 4.15 The highest F1 score for each text augmentation strategy in six emotion categories

Emotion	PRIMARY	CLUSTER	MISCLASSFIED
Happiness	0.619	0.614	**0.624**
Anger	0.379	0.371	**0.408**
Excitement	0.528	0.539	**0.551**
Boredom	0.800	**0.857**	0.800
Desperation	0.125	0.065	**0.154**
Indifference	**0.250**	0.182	0.162

4.6 Discussion

Three main findings are discussed based on the experiment results. The first finding is the effect of the augmented set size on the performance of the emotion classifiers. The second finding focuses on the differences of the vector representations and their effect on the performance of the emotion classifiers. The third finding throws light on the effect of text augmentation strategy on the performance of the emotion classifiers.

4.6.1 Augmented Set Size

According to the results from threshold-based experiments, augmented sets with relatively higher number of augmented instances have a higher tendency to produce low F1 scores. When a large number of augmented instances are added into the training set, it can also introduce unwanted noise into the training set at the same time. In threshold-based experiment, the happiness USE augmented set with threshold value of 0.7 consists of 670 augmented examples achieved the highest F1 score of 0.603 and

augmented set under the same category with threshold value of 0.5 consists of 26,650 augmented examples achieved the lowest F1 score of 0.445.

4.6.2 Vector Representations

The seven vector representations used in this research are categorized broadly into sentence embeddings, word embeddings and Bag-of-Words. The sentence embedding approach includes USE, InferSent GloVe and InferSent fastText while the word embedding approach covers W2V, fastText and GloVe. According to the results from both threshold-based and fixed increment experiments, the sentence embedding approach outperforms word embeddings and Bag-of-Words in term of classifier performance and stability. In the threshold-based experiments, USE is selected as the best vector representation to augment relevant instances because most of the augmented sets generated from USE approach achieve the high F1 scores among others in the six different emotion categories. The highest F1 scores for emotion happiness (0.603), anger (0.436), excitement (0.531), desperation (0.182), boredom (0.8) and indifference (0.235) are achieved by USE in threshold-based experiment because all augmented sets generated in the threshold-based experiments using USE have relatively smaller sizes compared to the other approaches, which means less noise is introduced into the training set. However, InferSent GloVe is selected as the best approach to augment relevant instances based on fixed increment experiments due to high stability in the F1 scores from six different emotion categories. The fixed increment experiment is designed to limit the number of instances assigned with high similarity scores to be added into the training. Hence, it is suitable to measure the stability of different vector representations in text augmentation. Also, we observe that results from random augmentation without similarity scoring approach show the worst performance compared to the others. Hence,

adding new instances into a corpus without proper augmentation strategy only will deteriorate the quality of the corpus. Finally, we can conclude that USE is more suitable for threshold-based augmentation while InferSent GloVe is the most stable approach for fixed increment augmentation.

4.6.3 Text Augmentation Strategy

According to the results from text augmentation strategy experiments, the misclassified strategy has 4 out of 6 highest F1 score based on the six different emotion categories. This strategy utilizes the misclassified emotion instances in the ET-test as the seed set to augment more relevant training data which is similar to reinforcement learning. Therefore, the classifier can achieve better performance by learning from more instances similar to the past mistakes.

4.6.4 Overall

The text augmentation proposed in this research utilizes the distant supervision and similarity scoring approach to augment new training instances similar to the instances in the original corpus. It is challenging for a machine learning model to learn certain patterns when training data is limited. Therefore, the goal of the proposed augmentation method is to economically increase the size of particular patterns especially the rare examples from existing corpus to improve the model's learning ability. In addition, the augmented instances are more realistic and diverse because all instances are real data collected from the microblog unlike other text augmentation approaches such as substitution, back-translation and text generation that can only generate artificial instances based on seeds from the original corpus. Furthermore, the

proposed text augmentation can be further improved by using the seed set for augmenting new instances with clustering and misclassified text augmentation strategies. Also, all the results obtained from PRIMARY, CLUSTER and MISCLASSIED strategies show significant improvement when compared to results from the baseline model which was trained solely on the original EmoTweet-28 dataset. Thus, we can conclude that our proposed text augmentation strategy has the tendency to expand the size of current existing dataset while preserving the quality of the dataset.

4.7 Summary

First, we compare the results from three different baselines to select a suitable baseline. Then, we present and discuss the similarity scoring results based on seven vector representations leveraging in neural embeddings and compare the effect of augmented sets on the binary emotion classifiers. In addition, we have presented further insights on the stability of each vector representation based on the fixed increment experiments. Finally, the results from three different text augmentation strategies are discussed to identify the better strategy to augment relevant training examples. In conclusion, USE is better approach for threshold-based augmentation, InferSent GloVe is the most stable approach for fixed increment augmentation and misclassified strategy is better at augmenting more relevant training instances.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this research, we proposed a text augmentation strategy utilizing neural embedding models for similarity scoring to expand the number of training instances for six emotion categories based on a set of gold standard seeds from EmoTweet-28 (ET). The motivation of the research is to expand the current existing emotion corpus with minimal human intervention while preserving the quality of the corpus.

The first research objective to propose a text augmentation strategy with similarity scoring approach that can leverage tweets collected using distant supervision for expanding the number of training data in an emotion corpus is achieved through the investigation of six pre-trained neural embedding models (USE, InferSent GloVe, InferSent fastText, W2V, fastText and GloVe) and Bag-of-Words to perform the similarity scoring between instances from ET and DS using different vector representations. All augmented instances were real data collected from Twitter using distant supervision (DS) and selected based on the similarity scores computed from the positive examples in EmoTweet-28.

The second research objective to evaluate the performance of similarity scoring approach using different pre-trained neural embedding models on augmenting new training data in different emotion categories is addressed by comparing the performance of similarity scoring approach based on different vector representations using a BiLSTM model on binary classification of six emotions (*happiness*, *anger*, *excitement*, *boredom*, *desperation* and *indifference*). The results shows that the USE is better for threshold-based augmentation and InferSent GloVe is the most stable for fixed increment augmentation.

The third research objective to identify the text augmentation strategy that can efficiently expand the current fine-grained emotion corpus with new relevant training data while preserving the quality of the corpus is achieved by exploring two additional text augmentation strategies on top of the primary strategy by altering the seed set to include only rare examples from training set (clustering) and another set containing only misclassified examples from the ET-test. The results show that the misclassified text augmentation strategy is better at expanding the current fine-grained emotion corpus with new relevant training data while preserving the quality of the corpus.

5.2 Research Contributions

There are three main contributions in this research. The first contribution is a text augmentation strategy we proposed by utilizing both distant supervision and similarity scoring approach to expand the existing emotion corpus with new relevant training instances while maintaining the quality of the corpus. The second contribution is we discovered USE is better approach for threshold-based augmentation and InferSent GloVe is the most stable approach for fixed increment augmentation. The final contribution is we discovered misclassified strategy works the best for text augmentation.

5.3 Strengths and Limitations

The strength of the proposed text augmentation strategy is the use of distant supervision and similarity scoring to collect and filter the instances similar to the instances in the original corpus. Therefore, all the augmented instances are more realistic and diverse compared to artificial instances generated from substitution, back-translation and text generation. In addition, we also minimise human intervention in the proposed text augmentation strategy using threshold-based and fixed increment

selection instead of human judgment to determine which examples to be added into the augmented set.

One limitation of this research is the corpus used for text augmentation must contain gold standard seeds produced through the manual human annotation and proper verification process. This is because the quality of the augmented instances is highly dependent on the quality of the seed set from original corpus (ET). If the corpus is poorly annotated, then the results of the text augmentation will not be desirable. In addition, the size of the corpus must be sufficiently adequate to be divided into the seed/training set (80%) and test set (20%) with proper size to yield reliable performance scores. A large seed set containing various instances can be used to augment more diverse instances and a sizeable test set is also important to reliably evaluate the performance of the text augmentation strategy. For instance, we can observe significant improvement in high frequency emotions (happiness, anger and excitement) clearly, but not in low frequency emotions (desperation, boredom and indifference) due to the difference in sizes of the seed set for each emotion category.

5.4 Future Work

In the future, we will explore more emotion categories in EmoTweet-28 with the proposed text augmentation strategies. Also, we can expand our similarity scoring approaches by utilizing new pre-trained neural embedding models or train our own neural embedding models with proper emotion corpus. In addition, new instances can be collected and annotated to manually expand the size of all emotion categories in EmoTweet-28 especially the low frequency emotion categories. Therefore, the instances in the EmoTweet-28 can be updated with the latest trend of emotion expressions in tweets, which can then be leveraged to augment more reliable instances. Finally, we can experiment the proposed text augmentation strategies on multi-label classification to fully utilize all instances in EmoTweet-28.

REFERENCES

Abdul-Mageed, M., & Ungar, L. (2017). EmoNet: Fine-grained emotion detection with gated recurrent neural networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 718–728.

Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from text: Machine learning for text-based emotion prediction. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 579–586.

Aroyehun, S. T., & Gelbukh, A. (2018). Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 90–97.

Balabantaray, R., Mohammad, M., & Sharma, N. (2012). Multi-class twitter emotion classification: A new approach. *International Journal of Applied Information Systems*, *4*(1), 48–53.

Bollen, J., Mao, H., & Pepe, A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Fifth International AAAI Conference on Weblogs and Social Media*, 450–453.

Bollen, J., Mao, H., & Zeng, X.-J. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1–8.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., & Kurzweil, R. (2018). *Universal sentence encoder*. 169–174.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 670–680.

Davidov, D., Tsur, O., & Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. *Coling 2010: Posters*, 241–249.

Dodds, P. S., & Danforth, C. M. (2010). Measuring the happiness of large-scale written expression: Songs, Blogs, and Presidents. *Journal of Happiness Studies*, *11*(4), 441–456.

Ekman, P. (1992). Facial expressions of emotion: New findings, new questions. *Psychological Science*, *3(1)*, 34–38.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1615–1625.

Fürstenau, H., & Lapata, M. (2009). Semi-supervised semantic role labeling. *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 220–228.

Giachanoua, A., Gonzalo, J., & Crestani, F. (2019). Propagating sentiment signals for estimating reputation polarity. *Information Processing & Management*, *56*(6).

Giridhara, P., Mishra, C., Venkataramana, R., Bukhari, S., & Dengel, A. (2019). A study of various text augmentation techniques for relation classification in free text: *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*, 360–367.

Godin, F. (2019). *Improving and interpreting neural networks for word-level prediction tasks in natural language processing*. Ghent University.

Gupta, U., Chatterjee, A., Srikanth, R., & Agrawal, P. (2018). A sentiment-and-semantics-based approach for emotion detection in textual conversations.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, *2*, 452–457.

Kolomiyets, O., Bethard, S., & Moens, M.-F. (2011). Model-portability experiments for textual temporal analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *2*, 271–276.

Li, B., & Han, L. (2013). Distance weighted cosine similarity measure for text classification. *International Conference on Intelligent Data Engineering and Automated Learning*, 611–618.

Li, S., Ao, X., Feiyang, P., & Qing, H. (2022). Learning policy scheduling for text augmentation. *Neural Networks*, *145*, 121–127.

Liew, J. S. Y., & Turtle, H. R. (2016). Exploring Fine-grained emotion detection in Tweets. *Proceedings of the NAACL Student Research Workshop*, 73–80.

Liew, J. S. Y., Turtle, H. R., & Liddy, E. D. (2016). EmoTweet-28: A fine-grained emotion corpus for sentiment analysis. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1149–1156.

Liu, R., Xu, G., Jia, C., Ma, W., Wang, L., & Vosoughi, S. (2020). Data Boost: Text data augmentation through reinforcement learning guided conditional generation. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9031–9041.

Lu, X., Zheng, B., Velivelli, A., & Zhai, C. (2006). Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association*, *13*(5), 526–535.

Luo, J., Bouazizi, M., & Ohtsuki, T. (2021). Data augmentation for sentiment analysis using sentence compression-based SeqGAN With data screening. *IEEE Access*, *9*, 99922–99931.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 3111–3119). Curran Associates, Inc.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003–1011.

Mohammad, S. (2012). Portable features for classifying emotional text. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 587–591.

Mohammad, S., & Turney, P. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 26–34.

Neculoiu, P., Versteegh, M., & Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. *Proceedings of the 1st Workshop on Representation Learning for NLP*, 148–157.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Purver, M., & Battersby, S. (2012). Experimenting with distant supervision for emotion classification. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 482–491.

Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012). Semantic cosine similarity. *The 7th International Student Conference on Advanced Science and Technology ICAST 2012*, *4*, 1.

Raiman, J., & Miller, J. (2017). Globally normalized reader. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*,

Risch, J., & Krestel, R. (2018). Aggression identification using deep learning and data augmentation. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 150–158.

Sharifirad, S., Jafarpour, B., & Matwin, S. (2018). Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 107–114.

Strapparava, C., & Mihalcea, R. (2008). Learning to identify emotions in text. *Proceedings of the 2008 ACM Symposium on Applied Computing - SAC '08*, 1556–1560.

Vijayaraghavan, P., Sysoev, I., Vosoughi, S., & Roy, D. (2016). DeepStance at SemEval-2016 Task 6: Detecting stance in tweets using character and word-Level CNNs. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 413–419.

Vo, B.-K. H., & Collier, N. (2013). Twitter emotion analysis in earthquake situations. *International Journal of Computational Linguistics and Applications*, *4*(1), 159–173.

Wang, W., Chen, L., Thirunarayan, K., & Sheth, A. P. (2012). Harnessing twitter "big data" for automatic emotion identification. *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, 587–592.

Wang, W. Y., & Yang, D. (2015). That's So Annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2557–2563.

Wei, J., Huang, C., Vosoughi, S., Cheng, Y., & Xu, S. (2021). Few-shot text classification with triplet networks, data augmentation, and curriculum learning. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5493–5500.

Wei, J., Huang, C., Xu, S., & Vosoughi, S. (2021). Text augmentation in a multi-task view. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2888–2894.

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 6383–6389.

Yoo, K. M., Park, D., Kang, J., Lee, S.-W., & Park, W. (2021). GPT3Mix: Leveraging large-scale language models for text augmentation. *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2225–2239.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining local convolution with global self-attention for reading comprehension. *ICLR*.

Zahiri, S. M., & Choi, J. D. (2018). Emotion detection on TV show transcripts with sequence-based convolutional neural networks. *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 44–52.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 649–657.

APPENDICES

APPENDIX A: FIXED INCREMENT EXPERIMENTS RESULTS

TABLE A.1 FIXED INCREMENT EXPERIMENT RESULTS FOR HAPPINESS

I	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
400	0.618	0.598	0.604	0.577	0.583	0.605	0.604	0.596
800	0.590	0.608	0.585	0.574	0.576	0.584	0.564	0.572
1200	0.596	0.617	0.578	0.608	0.585	0.586	0.561	0.563
1600	0.608	0.604	0.597	0.593	0.544	0.596	0.566	0.614
2000	0.598	0.606	0.563	0.575	0.554	0.579	0.533	0.530
2400	0.602	0.609	0.544	0.585	0.557	0.581	0.545	0.541
2800	0.576	0.601	0.538	0.573	0.596	0.566	0.525	0.547
3200	0.572	0.613	0.552	0.564	0.562	0.578	0.507	0.540
3600	0.600	0.595	0.554	0.564	0.559	0.550	0.523	0.514
4000	0.559	0.593	0.530	0.571	0.562	0.548	0.507	0.469
4400	0.577	0.600	0.533	0.559	0.566	0.582	0.513	0.498
4800	0.574	0.594	0.541	0.525	0.554	0.559	0.507	0.527
5200	0.544	0.598	0.508	0.548	0.552	0.589	0.496	0.496
5600	0.564	0.566	0.490	0.530	0.570	0.555	0.488	0.490
6000	0.573	0.549	0.513	0.544	0.542	0.584	0.488	0.499
6400	0.551	0.596	0.512	0.547	0.569	0.583	0.502	0.475
6800	0.556	0.587	0.516	0.541	0.554	0.569	0.480	0.493
7200	0.567	0.610	0.498	0.517	0.570	0.559	0.488	0.483
7600	0.553	0.606	0.448	0.519	0.554	0.581	0.521	0.451
8000	0.584	0.600	0.478	0.479	0.570	0.518	0.475	0.390
8400	0.535	0.584	0.502	0.545	0.558	0.553	0.465	0.432
8800	0.551	0.609	0.470	0.488	0.560	0.591	0.447	0.467
9200	0.546	0.607	0.474	0.505	0.489	0.558	0.453	0.471
9600	0.550	0.608	0.477	0.551	0.554	0.558	0.427	0.447
10000	0.563	0.619	0.471	0.502	0.563	0.571	0.433	0.438
10400	0.530	0.614	0.469	0.531	0.560	0.561	0.434	0.438
10800	0.532	0.582	0.498	0.474	0.527	0.533	0.441	0.420
11200	0.528	0.613	0.462	0.511	0.523	0.534	0.443	0.462
11600	0.545	0.616	0.465	0.511	0.552	0.556	0.443	0.468
12000	0.506	0.598	0.459	0.493	0.535	0.543	0.442	0.435

TABLE A.2 FIXED INCREMENT EXPERIMENT RESULTS FOR ANGER

I	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
400	0.394	0.197	0.372	0.354	0.276	0.354	0.282	0.341
800	0.327	0.250	0.221	0.273	0.332	0.317	0.400	0.298
1200	0.366	0.255	0.326	0.316	0.174	0.320	0.372	0.000
1600	0.418	0.330	0.369	0.311	0.257	0.300	0.367	0.363
2000	0.418	0.340	0.372	0.335	0.321	0.357	0.354	0.336
2400	0.406	0.319	0.364	0.320	0.297	0.299	0.369	0.322
2800	**0.421**	0.225	0.354	0.294	0.277	0.253	0.328	0.303
3200	0.407	0.211	0.381	0.309	0.335	0.284	0.391	0.306
3600	0.396	0.357	0.329	0.296	0.327	0.289	0.325	0.367
4000	0.366	0.371	0.378	0.392	0.339	0.353	0.373	0.357
4400	0.367	0.269	0.343	0.328	0.314	0.289	0.357	0.312
4800	0.381	0.299	0.370	0.303	0.245	0.297	0.319	0.331
5200	0.371	0.341	0.344	0.359	0.353	0.326	0.359	0.352
5600	0.368	0.225	0.331	0.369	0.313	0.276	0.340	0.347
6000	0.348	0.214	0.342	0.302	0.364	0.322	0.355	0.346
6400	0.386	0.283	0.321	0.358	0.334	0.322	0.315	0.335
6800	0.372	0.273	0.370	0.349	0.405	0.360	0.344	0.338
7200	0.366	0.280	0.291	0.331	0.323	0.354	0.347	0.342
7600	0.341	0.335	0.348	0.366	0.345	0.347	0.311	0.317
8000	0.354	0.270	0.357	0.310	0.315	0.328	0.320	0.325
8400	0.326	0.323	0.331	0.347	0.310	0.345	0.298	0.319
8800	0.351	0.326	0.367	0.339	0.371	0.343	0.305	0.315
9200	0.347	0.238	0.325	0.300	0.322	0.308	0.306	0.295
9600	0.334	0.379	0.331	0.302	0.330	0.317	0.325	0.306
10000	0.337	0.253	0.338	0.344	0.347	0.348	0.300	0.295
10400	0.312	0.327	0.352	0.330	0.336	0.310	0.309	0.286
10800	0.333	0.263	0.326	0.315	0.342	0.310	0.320	0.315
11200	0.336	0.290	0.313	0.313	0.324	0.332	0.337	0.304
11600	0.330	0.310	0.314	0.300	0.326	0.313	0.292	0.281
12000	0.288	0.319	0.309	0.298	0.302	0.346	0.284	0.283

TABLE A.3 FIXED INCREMENT EXPERIMENT RESULTS FOR EXCITEMENT

I	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
400	0.497	0.528	0.538	0.566	0.541	0.529	0.503	0.480
800	0.500	0.400	0.459	0.519	0.508	0.477	0.482	0.392
1200	0.433	0.442	0.457	0.452	0.482	0.446	0.433	0.409
1600	0.468	0.509	0.430	0.463	0.469	0.490	0.405	0.447
2000	0.408	0.512	0.429	0.449	0.420	0.430	0.397	0.442
2400	0.451	0.483	0.403	0.467	0.372	0.400	0.354	0.356
2800	0.407	0.453	0.383	0.401	0.427	0.398	0.306	0.307
3200	0.404	0.440	0.393	0.387	0.448	0.439	0.350	0.307
3600	0.363	0.453	0.327	0.375	0.402	0.403	0.308	0.320
4000	0.367	0.454	0.353	0.333	0.437	0.405	0.304	0.323
4400	0.391	0.454	0.329	0.322	0.403	0.394	0.298	0.282
4800	0.317	0.462	0.326	0.369	0.369	0.417	0.309	0.286
5200	0.378	0.445	0.325	0.342	0.372	0.450	0.293	0.295
5600	0.329	0.396	0.362	0.322	0.405	0.384	0.319	0.264
6000	0.369	0.445	0.344	0.356	0.365	0.372	0.299	0.314
6400	0.343	0.454	0.377	0.252	0.384	0.332	0.290	0.312
6800	0.338	0.418	0.322	0.314	0.330	0.345	0.324	0.274
7200	0.300	0.419	0.290	0.282	0.356	0.346	0.298	0.285
7600	0.316	0.439	0.346	0.300	0.362	0.393	0.300	0.269
8000	0.300	0.393	0.339	0.275	0.327	0.346	0.274	0.258
8400	0.320	0.414	0.326	0.319	0.364	0.331	0.263	0.304
8800	0.288	0.430	0.328	0.267	0.339	0.386	0.266	0.251
9200	0.286	0.400	0.297	0.303	0.364	0.349	0.277	0.277
9600	0.317	0.437	0.313	0.330	0.328	0.347	0.275	0.262
10000	0.278	0.377	0.323	0.269	0.313	0.311	0.246	0.215
10400	0.296	0.383	0.321	0.262	0.316	0.323	0.280	0.238
10800	0.283	0.433	0.314	0.298	0.260	0.356	0.250	0.247
11200	0.304	0.425	0.319	0.258	0.327	0.311	0.260	0.237
11600	0.237	0.403	0.305	0.230	0.284	0.277	0.269	0.215
12000	0.247	0.353	0.303	0.223	0.265	0.318	0.257	0.243

TABLE A.4 FIXED INCREMENT EXPERIMENT RESULTS FOR BOREDOM

I	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
400	0.750	0.800	0.667	0.245	0.667	0.480	0.545	0.286
800	0.600	0.462	0.273	0.462	0.462	0.353	0.667	0.364
1200	0.387	0.800	0.240	0.353	0.522	0.429	0.500	0.286
1600	0.308	0.706	0.188	0.240	0.429	0.333	0.261	0.308
2000	0.324	0.750	0.174	0.217	0.375	0.323	0.375	0.231
2400	0.308	0.706	0.179	0.231	0.293	0.163	0.226	0.124
2800	0.316	0.600	0.125	0.235	0.293	0.313	0.152	0.150
3200	0.235	0.522	0.117	0.250	0.182	0.182	0.169	0.125
3600	0.231	0.632	0.094	0.097	0.324	0.194	0.107	0.073
4000	0.140	0.571	0.138	0.115	0.200	0.250	0.089	0.105
4400	0.203	0.600	0.098	0.130	0.222	0.179	0.094	0.102
4800	0.211	0.343	0.138	0.110	0.174	0.141	0.090	0.083
5200	0.167	0.414	0.110	0.110	0.125	0.161	0.048	0.067
5600	0.160	0.444	0.079	0.077	0.117	0.105	0.085	0.073
6000	0.114	0.429	0.077	0.101	0.164	0.095	0.074	0.068
6400	0.112	0.387	0.072	0.121	0.197	0.125	0.096	0.059
6800	0.136	0.387	0.068	0.077	0.179	0.110	0.054	0.046
7200	0.136	0.600	0.065	0.118	0.140	0.103	0.081	0.058
7600	0.112	0.267	0.100	0.093	0.150	0.133	0.072	0.045
8000	0.099	0.462	0.071	0.061	0.104	0.071	0.061	0.041
8400	0.140	0.316	0.075	0.075	0.152	0.099	0.055	0.039
8800	0.126	0.197	0.078	0.090	0.081	0.152	0.075	0.041
9200	0.092	0.222	0.067	0.060	0.102	0.102	0.048	0.044
9600	0.096	0.500	0.075	0.053	0.121	0.092	0.052	0.042
10000	0.059	0.333	0.073	0.054	0.112	0.092	0.046	0.028
10400	0.078	0.300	0.066	0.059	0.078	0.065	0.039	0.033
10800	0.057	0.235	0.055	0.048	0.117	0.070	0.033	0.029
11200	0.052	0.267	0.050	0.053	0.079	0.071	0.038	0.027
11600	0.068	0.235	0.052	0.056	0.082	0.054	0.035	0.026
12000	0.066	0.162	0.054	0.056	0.092	0.063	0.031	0.023

TABLE A.5 FIXED INCREMENT EXPERIMENT RESULTS FOR DESPERATION

I	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
800	0.083	0.000	0.000	0.111	0.000	0.000	0.000	0.000
1200	0.061	0.125	0.000	0.047	0.000	0.045	0.000	0.098
1600	0.042	0.000	0.000	0.051	0.000	0.061	0.000	0.000
2000	0.053	0.000	0.000	0.000	0.040	0.065	0.000	0.022
2400	0.057	0.000	0.065	0.050	0.059	0.036	0.000	0.022
2800	0.027	0.000	0.000	0.034	0.035	0.037	0.014	0.011
3200	0.033	0.091	0.000	0.041	0.000	0.040	0.019	0.021
3600	0.013	0.000	0.000	0.017	0.047	0.029	0.015	0.008
4000	0.015	0.000	0.000	0.017	0.048	0.019	0.044	0.026
4400	0.017	0.000	0.044	0.017	0.024	0.029	0.016	0.018
4800	0.018	0.083	0.000	0.013	0.034	0.028	0.059	0.014
5200	0.018	0.000	0.000	0.013	0.042	0.045	0.027	0.015
5600	0.008	0.000	0.000	0.016	0.019	0.051	0.011	0.014
6000	0.019	0.000	0.000	0.015	0.013	0.025	0.009	0.018
6400	0.018	0.000	0.000	0.013	0.027	0.047	0.024	0.011
6800	0.018	0.000	0.000	0.017	0.018	0.032	0.024	0.021
7200	0.008	0.044	0.000	0.012	0.009	0.018	0.006	0.016
7600	0.021	0.028	0.041	0.011	0.013	0.028	0.014	0.015
8000	0.018	0.050	0.024	0.011	0.013	0.036	0.019	0.017
8400	0.016	0.035	0.000	0.014	0.010	0.056	0.023	0.017
8800	0.019	0.020	0.023	0.012	0.009	0.023	0.012	0.011
9200	0.023	0.027	0.027	0.013	0.008	0.023	0.022	0.007
9600	0.012	0.022	0.031	0.013	0.009	0.026	0.015	0.008
10000	0.016	0.020	0.014	0.025	0.009	0.018	0.015	0.011
10400	0.021	0.022	0.025	0.016	0.019	0.014	0.019	0.011
10800	0.017	0.019	0.023	0.009	0.018	0.017	0.018	0.010
11200	0.013	0.013	0.019	0.005	0.013	0.017	0.019	0.014
11600	0.011	0.028	0.025	0.016	0.011	0.017	0.011	0.010
12000	0.019	0.020	0.007	0.022	0.017	0.012	0.015	0.009

TABLE A.6 FIXED INCREMENT EXPERIMENT RESULTS FOR INDIFFERENCE

I	USE	Inf. GloVe	Inf. fastText	GloVe	W2V	fastText	BoW	Random
400	0.143	0.118	0.143	0.133	0.065	0.074	0.098	0.000
800	0.125	0.000	0.211	0.118	0.114	0.125	0.061	0.138
1200	0.091	0.143	0.200	0.093	0.049	0.084	0.080	0.053
1600	0.063	0.250	0.105	0.167	0.080	0.034	0.066	0.029
2000	0.118	0.059	0.113	0.074	0.057	0.071	0.085	0.023
2400	0.066	0.062	0.074	0.084	0.040	0.041	0.065	0.030
2800	0.108	0.065	0.074	0.066	0.049	0.078	0.049	0.027
3200	0.041	0.120	0.083	0.070	0.063	0.055	0.047	0.045
3600	0.041	0.049	0.071	0.037	0.038	0.038	0.059	0.037
4000	0.045	0.039	0.033	0.039	0.034	0.100	0.068	0.030
4400	0.045	0.056	0.042	0.027	0.019	0.045	0.039	0.035
4800	0.034	0.041	0.029	0.028	0.040	0.034	0.032	0.046
5200	0.026	0.040	0.035	0.030	0.045	0.029	0.033	0.046
5600	0.025	0.037	0.035	0.036	0.021	0.048	0.030	0.032
6000	0.042	0.038	0.025	0.029	0.021	0.028	0.027	0.018
6400	0.022	0.042	0.048	0.029	0.025	0.041	0.035	0.041
6800	0.034	0.034	0.031	0.021	0.038	0.040	0.031	0.024

APPENDIX B: TEXT AUGMENTATION STRATEGY RESULTS

TABLE B.1 PRIMARY STRATEGY RESULTS (INFERSENT GLOVE)

I	Happiness	Anger	Excitement	Boredom	Desperation	Indifference
Baseline	0.573	0.318	0.563	0.800	0.000	0.000
400	0.598	0.197	**0.528**	**0.800**	0.000	0.118
800	0.608	0.250	0.400	0.462	0.000	0.000
1200	0.617	0.255	0.442	0.800	**0.125**	0.143
1600	0.604	0.330	0.509	0.706	0.000	**0.250**
2000	0.606	0.340	0.512	0.750	0.000	0.059
2400	0.609	0.319	0.483	0.706	0.000	0.062
2800	0.601	0.225	0.453	0.600	0.000	0.065
3200	0.613	0.211	0.440	0.522	0.091	0.120
3600	0.595	0.357	0.453	0.632	0.000	0.049
4000	0.593	0.371	0.454	0.571	0.000	0.039
4400	0.600	0.269	0.454	0.600	0.000	0.056
4800	0.594	0.299	0.462	0.343	0.083	0.041
5200	0.598	0.341	0.445	0.414	0.000	0.040
5600	0.566	0.225	0.396	0.444	0.000	0.037
6000	0.549	0.214	0.445	0.429	0.000	0.038
6400	0.596	0.283	0.454	0.387	0.000	0.042
6800	0.587	0.273	0.418	0.387	0.000	0.034
7200	0.610	0.280	0.419	0.600	0.044	-
7600	0.606	0.335	0.439	0.267	0.028	-
8000	0.600	0.270	0.393	0.462	0.050	-
8400	0.584	0.323	0.414	0.316	0.035	-
8800	0.609	0.326	0.430	0.197	0.020	-
9200	0.607	0.238	0.400	0.222	0.027	-
9600	0.608	**0.379**	0.437	0.500	0.022	-
10000	**0.619**	0.253	0.377	0.333	0.020	-
10400	0.614	0.327	0.383	0.300	0.022	-
10800	0.582	0.263	0.433	0.235	0.019	-
11200	0.613	0.290	0.425	0.267	0.013	-
11600	0.616	0.310	0.403	0.235	0.028	-
12000	0.598	0.319	0.353	0.162	0.020	-

TABLE B.2 CLUSTERING STRATEGY RESULTS (INFERSENT GLOVE)

I	Happiness	Anger	Excitement	Boredom	Desperation	Indifference
Baseline	0.573	0.318	0.563	0.800	0.000	0.000
400	0.575	0.360	**0.539**	0.800	0.000	**0.182**
800	0.598	0.299	0.514	0.667	0.000	0.000
1200	0.613	0.273	0.527	0.750	0.000	0.129
1600	0.605	0.356	0.472	0.750	0.000	0.103
2000	0.603	0.369	0.453	**0.857**	0.000	0.071
2400	0.598	0.270	0.505	0.750	**0.065**	0.075
2800	0.589	0.177	0.462	0.235	0.000	0.100
3200	0.603	0.280	0.462	0.706	0.000	0.043
3600	**0.614**	0.256	0.414	0.706	0.000	0.051
4000	0.557	0.245	0.419	0.600	0.000	0.053
4400	0.564	0.320	0.442	0.444	0.000	0.077
4800	0.613	0.264	0.445	0.353	0.000	0.060
5200	0.595	**0.371**	0.419	0.480	0.000	0.038
5600	0.501	0.259	0.422	0.462	0.000	0.041
6000	0.586	0.338	0.448	0.308	0.029	0.029
6400	0.594	0.330	0.397	0.522	0.041	0.025
6800	0.593	0.248	0.455	0.500	0.033	0.025
7200	0.609	0.335	0.421	0.667	0.050	-
7600	0.561	0.351	0.375	0.545	0.027	-
8000	0.579	0.331	0.373	0.414	0.065	-
8400	0.607	0.355	0.438	0.273	0.036	-
8800	0.571	0.298	0.377	0.500	0.019	-
9200	0.611	0.292	0.408	0.316	0.036	-
9600	0.578	0.313	0.380	0.324	0.021	-
10000	0.575	0.341	0.323	0.400	0.020	-
10400	0.576	0.308	0.371	0.333	0.030	-
10800	0.588	0.298	0.348	0.197	0.017	-
11200	0.579	0.310	0.361	0.207	0.016	-
11600	0.583	0.343	0.365	0.197	0.010	-
12000	0.553	0.346	0.347	0.213	0.009	-

TABLE B.3 MISCLASSIFIED STRATEGY RESULTS (INFERSENT GLOVE)

I	Happiness	Anger	Excitement	Boredom	Desperation	Indifference
Baseline	0.573	0.318	0.563	0.800	0.000	0.000
400	0.589	0.377	0.527	0.522	0.000	0.121
800	0.609	0.266	0.500	0.706	0.143	**0.162**
1200	0.606	0.304	0.512	**0.800**	**0.154**	0.100
1600	0.599	0.383	0.518	0.800	0.105	0.111
2000	0.611	0.395	0.467	0.538	0.000	0.140
2400	0.592	0.344	0.453	0.452	0.143	0.109
2800	0.609	0.317	0.469	0.538	0.111	0.048
3200	0.578	0.283	**0.551**	0.480	0.070	0.086
3600	0.602	0.321	0.436	0.359	0.148	0.043
4000	0.571	**0.408**	0.429	0.412	0.087	0.057
4400	0.593	0.266	0.467	0.286	0.100	0.044
4800	0.593	0.286	0.467	0.368	0.080	0.038
5200	0.597	0.327	0.432	0.270	0.118	0.058
5600	0.571	0.295	0.486	0.326	0.033	0.067
6000	0.611	0.345	0.480	0.298	0.056	0.033
6400	0.615	0.253	0.461	0.364	0.050	0.021
6800	0.602	0.362	0.398	0.438	0.051	0.036
7200	**0.624**	0.274	0.462	0.222	0.000	-
7600	0.591	0.325	0.396	0.215	0.066	-
8000	0.601	0.302	0.449	0.255	0.037	-
8400	0.590	0.379	0.418	0.209	0.038	-
8800	0.596	0.337	0.444	0.211	0.025	-
9200	0.600	0.271	0.413	0.215	0.040	-
9600	0.573	0.348	0.409	0.185	0.035	-
10000	0.568	0.341	0.385	0.182	0.026	-
10400	0.613	0.253	0.405	0.259	0.033	-
10800	0.578	0.314	0.406	0.144	0.017	-
11200	0.584	0.277	0.416	0.128	0.022	-
11600	0.525	0.341	0.429	0.128	0.032	-
12000	0.590	0.299	0.402	0.163	0.019	-

LIST OF PUBLICATIONS

1. Yong K. S., & Liew, J. S. Y. (2020). A text augmentation approach using similarity measures based on neural sentence embeddings for emotion classification on microblogs. In 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) (pp. 1-6). IEEE. [SCOPUS]