Meaning-Sensitive Noisy Text Analytics in the Low Data Regime

by

Buddhika H. Kasthuriarachchy

A thesis submitted in partial fulfilment of the requirements for the degree

of

Doctor of Philosophy

School of Engineering, Information Technology and Physical Sciences

Federation University Australia

April 2022

Supervisors:

Principal Supervisor

Associate Professor Madhu Chetty

Federation University Australia

Associate Supervisor

Dr Adrian Shatte

Federation University Australia

Co-Supervisor

Darren Walls

Global Hosts Pty Ltd, Australia

Abstract

Digital connectivity is revolutionising people's quality of life. As broadband and mobile services become faster and more prevalent globally than before, people have started to frequently express their wants and desires on social media platforms. Thus, deriving insights from text data has become a popular approach, both in the industry and academia, to provide social media analytics solutions across a range of disciplines, including consumer behaviour, sales, sports and sociology. Businesses can harness the data shared on social networks to improve their organisations' strategic business decisions by leveraging advanced Natural Language Processing (NLP) techniques, such as context-aware representations. Specifically, SportsHosts, our industry partner, will be able to launch digital marketing solutions that optimise audience targeting and personalisation using NLP-powered solutions. However, social media data are often noisy and diverse, making the task very challenging. Further, real-world NLP tasks often suffer from insufficient labelled data due to the costly and time-consuming nature of manual annotation. Nevertheless, businesses are keen on maximising the return on investment by boosting the performance of these NLP models in the real world, particularly with social media data.

In this thesis, we make several contributions to address these challenges. Firstly, we propose to improve the NLP model's ability to comprehend noisy text in a low data regime by leveraging prior knowledge from pre-trained language models. Secondly, we analyse the impact of text augmentation and the quality of synthetic sentences in a context-aware NLP setting and propose a meaning-sensitive text augmentation technique using a Masked Language Model. Thirdly, we offer a cost-efficient text data annotation methodology and an end-to-end framework to deploy efficient and effective social media analytics solutions in the real world.

Acknowledgements

First and foremost, I would like to express my gratitude to Associate Professor Madhu Chetty for serving as my Principal Supervisor, mentor and friend. He encouraged me, challenged me, treated me with respect and as an equal, provided opportunities, promoted me and my work, and has encouraged great growth in me as a researcher in addition to aiding with the practical, technical and theoretical parts of my study, which has been invaluable. It is immeasurable how much I have learned from Madhu, and I am extremely grateful to have worked with him during these formative years.

I am deeply grateful to my Associate Supervisor, Dr Adrian Shatte, whose expertise, guidance and encouragement have been beyond valuable for both the work of this thesis and my professional development. Adrian worked very closely with me, and his insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I would also like to offer my special thanks to Professor Manzur Murshed, Associate Professor Gaur Karmakar, Associate Professor Feng Xia, Professor Wendy Wright and Paula Di Maria for their mentoring and helpful advice.

My sincere thank also goes to my Co-Supervisor, Darren Walls, the Founder and Chief Executive Officer (CEO) of SportsHosts, a Melbourne-based start-up, for sharing the industry requirements, real-world challenges and his business strategies with me. Darren's support and guidance were instrumental in blending our research findings with real-world applications.

Most importantly, I would like to express my heartfelt gratitude to Dr Tony Sahama and Dr Thilina Halloluwa for encouraging, supporting and guiding me through the PhD application process. Without them, I would not have started this PhD, let alone finished it. I

would also like to show appreciation to Linda Butler of the Graduate Research School at Federation University for managing my enrolment process smoothly.

I acknowledge Global Hosts Pty Ltd (trading as SportsHosts), Australia, for providing me with an industry-funded stipend scholarship, and I thank them for giving me various learning, enriching, networking and professional and personal development experiences. I would like to especially thank Mr Darren Walls, CEO of SportsHosts, for making the time for frequent meetings, discussions and clarifications related to the specifications and requirements related to the on-going PhD research project. Further, I thank Federation University Australia for providing the Research Tuition Scholarship over the three years of my PhD studies.

Capstone Editing provided copyediting and proofreading services, according to the guidelines laid out in the university-endorsed national 'Guidelines for Editing Research Theses'.

There are others not associated with this thesis in an official capacity but who have contributed significantly, nonetheless. I wish to thank my colleagues Ataul Rajin, Dr Mahbub Khoda, Dr Manzurul Islam, Hasitha Hewawasam (soon-to-be Dr) and my housemates Douglas Strand and Muzamal-Ali Ibrahimi for their companionship, empathy and encouragement. I thank my friends and especially my housemates for remaining interested and supportive and giving me many things to enjoy outside the PhD. Paige Saunders has been especially helpful with my accommodation arrangements in Australia.

Most importantly, none of this could have happened without my family and friends. They have been kind and supportive to me over the last several years. This thesis stands as a testament to their unconditional love and encouragement.

Statement of Authorship

Except where explicit reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis by which I have qualified for or been awarded another degree or diploma. No other person's work has been relied upon or used without due acknowledgement in the main text and reference list of the thesis. No editorial assistance has been received in the production of the thesis without due acknowledgement. Except where duly referred to, the thesis does not include material with copyright provisions or requiring copyright approvals.

Buddhika H. Kasthuriarachchy

2022 April 20

Contents

Abstract ii

Acknowledgements iii

Statement of Authorship v

List of Figures x

List of Tables xi

List of Abbreviations xii

Publications xiii

1 Introduction 1
- 1.1 Social Media Analytics . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives of the Study . . . 4
- 1.4 Contributions . . . 5
- 1.5 Thesis Outline . . . 7

2 Literature Review 9
- 2.1 Background Preliminaries . . . 9
  - 2.1.1 Probability . . . 9
  - 2.1.2 Machine Learning . . . 10
  - 2.1.3 Neural Networks . . . 12
  - 2.1.4 Natural Language Processing . . . 13
- 2.2 Natural Language Processing Applications . . . 15
- 2.3 Deep Learning for Natural Language Processing . . . 17
- 2.4 Language Models . . . 19
  - 2.4.1 Masked Language Models . . . 24
- 2.5 Transfer Learning . . . 25
  - 2.5.1 Transfer Learning in NLP . . . 25
  - 2.5.2 Transfer Learning with Language Models . . . 27
  - 2.5.3 Supply Chain for NLP . . . 28
- 2.6 Language Understanding . . . 28
  - 2.6.1 Language Understanding with BERT . . . 29
  - 2.6.2 Probing Tasks . . . 31
- 2.7 Text Augmentation . . . 31

2.8 Summary . . . 34

3 Noisy Text Comprehension **35**
3.1 Introduction . . . 35
3.2 Inductive Transfer Learning with Pre-Trained Language Models . . . 37
3.2.1 Methodology . . . 38
3.2.1.1 Architecture of the Pre-Trained Language Model . . . 38
3.2.1.2 Inductive Transfer Learning . . . 39
3.2.1.3 Fine-Tuning . . . 40
3.2.1.4 Back Translation . . . 41
3.3 Noisy Text Comprehension . . . 41
3.3.1 Sentence Vector Generation . . . 44
3.3.2 Noisy Probing Datasets . . . 47
3.3.3 Sentence Vector Evaluation Framework . . . 49
3.4 Experiments . . . 50
3.4.1 Experiments on Transfer Learning with Language Models . . . 50
3.4.1.1 Dataset . . . 51
3.4.1.2 BERT Fine-Tuning . . . 52
3.4.1.3 Data Augmentation with Back Translation . . . 53
3.4.1.4 Results and Discussion . . . 53
3.4.2 Experiments on Noisy Text Comprehension . . . 55
3.4.2.1 Dataset Development . . . 55
3.4.2.2 Sentence Embedding Generation . . . 55
3.4.2.3 Probing Task Classification . . . 56
3.4.2.4 Results . . . 56
3.4.2.5 Discussion . . . 61
3.5 Summary . . . 64

4 Meaning-Sensitive Text Augmentation **65**
4.1 Introduction . . . 65
4.2 Semantic Data Augmentation . . . 66
4.2.1 Augmentation Methodology . . . 69
4.2.2 Experiments on Semantic Data Augmentation . . . 70
4.2.2.1 Ablation Study . . . 71
4.2.2.2 Discussion . . . 75
4.3 Meaning-Sensitive Data Augmentation with Intelligent Masking . . . 77
4.3.1 IMOSA Methodology . . . 78
4.3.1.1 Conditional Masked Language Model Pre-Training . . . 79
4.3.1.2 Intelligent Masking . . . 80
4.3.1.3 Optimal Substitution . . . 83
4.3.2 Experiments on IMOSA . . . 84
4.3.2.1 Baselines . . . 85
4.3.2.2 Augmentation Settings . . . 86
4.3.2.3 Evaluation Method . . . 88
4.3.2.4 Results . . . 88
4.3.2.5 Analysis and Discussion . . . 90
4.4 Summary . . . 93

5 Social Media Intelligence and Text Analytics **94**
5.1 Introduction . . . 94
5.2 Cost-Effective Data Annotation . . . 95
5.2.1 Text Annotation Framework . . . 97
5.2.2 Meaning-Sensitive Pre-Processing . . . 99
5.2.3 Pre-Annotation with Zero-Shot Classification . . . 100
5.2.4 Specialised Sub-Datasets . . . 101
5.2.5 Handling Promotional Tweets . . . 101
5.2.6 Multi-Phase Approach . . . 102
5.2.6.1 Phase-Wise Annotation . . . 103
5.2.6.2 Two-Stage Batch Annotation . . . 105
5.3 Social Intelligence Framework . . . 105
5.3.1 Business Requirement . . . 107
5.3.2 Data Preparation . . . 107
5.3.3 Model Development . . . 107
5.3.4 Evaluation . . . 108
5.3.5 Model Deployment . . . 108
5.4 Experiments . . . 109
5.4.1 Discussion . . . 113
5.5 Summary . . . 116

6 Conclusion **117**
6.1 Research Summary . . . 118
6.2 Research Findings . . . 119
6.3 Future Directions . . . 121

References **123**

List of Figures

2.1 RNN model architecture. . . . 17
2.2 CNN architecture. . . . 18
2.3 Model architecture of the Transformer. . . . 20
2.4 A taxonomy for transfer learning for NLP. . . . 26
2.5 AI supply chain for NLP. . . . 28

3.1 The architecture of the BERT model. . . . 40
3.2 Dissection of BERT layers. . . . 45
3.3 Framework. . . . 50
3.4 Data splitting for fine-tuning. . . . 51
3.5 Heat map of probing task accuracy. . . . 60

4.1 Semantic data augmentation framework. . . . 67
4.2 Probability density distributions of semantic similarity scores. . . . 72
4.3 Probability density distribution of semantic similarity of the mixture of all the back-translated sentences. . . . 72
4.4 Evaluation accuracy. . . . 73
4.5 Average test accuracy. . . . 75
4.7 Preparation of BERT inputs for masked sentences. . . . 80
4.8 Key steps in the intelligent masking phase. . . . 81
4.9 Generation of the most probable synthetic sentences from masked token predictions. . . . 82
4.10 Frequency distribution of the number of synthetic sentences generated per original example in the training set. . . . 91
4.11 Distribution of percentage of masked tokens. . . . 92

5.1 Annotation framework. . . . 98
5.2 Two-stage mini-batch annotation process. . . . 104
5.3 High-level framework. . . . 106
5.4 MTurk project–annotation labels. . . . 113
5.5 MTurk project–annotation instructions. . . . 113

List of Tables

2.1 Pre-trained language models. 22

3.1 Strategy to generate sentence embeddings from each region. 46
3.2 Grouping of probing tasks. 49
3.3 Composition of the dataset. 51
3.4 Noisy text classification accuracies. 53
3.5 Probing task datasets. 55
3.6 Pre-trained language models. 56
3.7 Parameters for the classifiers. 56
3.8 Sentence vector sizes derived using different pooling strategies. 57
3.9 Average classification accuracy for different sentence vector sizes. 58
3.10 Mean classification accuracy for sentence vectors derived using different pooling strategies. 59
3.11 Classification accuracies for different sentence vectors. 62

4.1 The F1 results of individual categories. 74
4.2 Examples of synthetic sentences and semantic similarity scores. 76
4.3 Datasets. 84
4.4 Dataset splits. 85
4.5 IMOSA configurations for each dataset. 86
4.6 Accuracies. 89

5.1 Characteristics of phases. 103
5.2 Hypotheses for pre-annotation. 110
5.3 Annotations schemata. 111
5.4 MTurk annotation quality. 112
5.5 The class distribution in specialised sub-datasets. 112
5.6 Dataset annotation summary. 114

Acronyms

AI Artificial Intelligence. 2

BERT Bidirectional Encoder Representations from Transformers. 15, 23, 38

BRNN Bidirectional Recurrent Neural Network. 18

CNN Convolutional Neural Network. 18

ELMo Embeddings from Language Models. 21

GPT Generative Pre-training Transformer. 23

GRU Gated-Recurrent Network. 18

LISA Linguistically-Informed Self-Attention. 27

LSTM Long Short-Term Memory. 18

NLP Natural Language Processing. 1, 2, 9, 13

NLU Natural Language Understanding. 2, 17, 41

ResNet Residual Network. 18

RNN Recurrent Neural Network. 17

SMA Social Media Analytics. 1, 2

SMT Statistical Machine Translation. 18

SRL Semantic Role Labelling. 27

SVM Support Vector Machine. 27

Publications

The work in this thesis primarily relates to the following peer-reviewed articles.

Journal publications

- B. Kasthuriarachchy, M. Chetty, A. Shatte and D. Walls, 'From general language understanding to noisy text comprehension', *Appl. Sci.*, vol. 11, no. 17, p. 7814, 2021, https://www.mdpi.com/2076-3417/11/17/7814/htm

Peer-reviewed conference papers

- B. Kasthuriarachchy, M. Chetty, A. Shatte and D. Walls, 'Cost effective annotation framework using zero-shot text classification', in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1-8. [Online]. Available https://ieeexplore.ieee.org/document/9534335

- B. Kasthuriarachchy, M. Chetty, G. Karmakar and D. Walls, 'Pre-trained language models with limited data for intent classification', in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom, 2020, pp. 1-9. Available https://ieeexplore.ieee.org/document/9207121

Submitted papers

- B. Kasthuriarachchy, M. Chetty, A. Shatte and D. Walls, 'Meaning-sensitive text augmentation with intelligent masking', *ACM Trans. Intell. Syst. and Technol.*

Chapter 1

Introduction

1.1 Social Media Analytics

As communities frequently express their wants and desires on social media platforms such as Twitter and Facebook, people are increasingly adopting social media for a variety of social interactions across numerous channels, thereby increasing its reach, popularity and relevance. Thus, organisations are keen to understand user behaviour to help them adapt their business strategies and goals. As a result, a new stream of analytics, known as Social Media Analytics (SMA), has emerged and focuses mainly on the study of social media data. To this end, understanding individual user behaviour through social media text data using Natural Language Processing (NLP) techniques–such as sentiment analysis, intent analysis [1, 2, 3] and opinion mining–has been an active area of research in the last decade [4, 5, 6]. These NLP applications have captured the attention of both the industry and academia due to their commercial and non-commercial significance in a plethora of real-world use cases. These NLP applications have captured the attention of both the industry and academia due to their commercial and non-commercial significance in a plethora of real-world use cases.

Through engagement with Federation University Australia, SportsHosts expressed its need, as part of a funded PhD research, for a data-driven solution to boost its global customer base. After a careful study of the SportsHosts business model and relevant background details, we proposed to leverage publicly available social media data to automati-

cally identify potential customers (i.e., sports fans) from their behaviour by using text mining and Artificial Intelligence (AI) methodologies. SportsHosts, as the client, and Federation University Australia decided to go beyond naive social media marketing approaches that are based on keywords or hashtags[1] and instead identified potential user behaviour by leveraging advanced NLP techniques, such as context-aware text representations blended with recent advancements.

In addition to these objectives, the industry has also been keen on maximising returns on investment (ROIs) by boosting the accuracy of NLP models in the real world, particularly with social media platforms. Further, as with other projects in the industry, cost and time optimisation for NLP applications in social media are crucial to attract investments or stay within budgets.

1.2 Motivation

SMA enables organisations to monitor social behaviours to make data-driven business decisions. The successful outcomes of the proposed research can enable SportsHosts to leverage the proposed framework and identify potential sports fans, significantly expanding its global customer base and business. However, the performance of many NLP tasks, including text classification, relies heavily on the machine's ability to understand the meaning conveyed in a sentence or document. For any common individual, understanding what is being written is practically subconscious and intuitive. We rely on what we already know about a language and the concepts in a text to comprehend its meaning. However, capturing the meaning conveyed through text is a challenging NLP problem for machines, particularly with noisy text data obtained from social media platforms, such as Twitter.

The study of a machine's ability to understand the human language is the focus of Natural Language Understanding (NLU) [7], a branch of NLP. NLU uses syntactic and semantic analyses of texts to determine the meaning of a sentence or phrase. This context-aware approach to understanding meaning is a new wave and evolution in NLP and AI that can ef-

[1]Introduced by the octothorpe symbol '#', the hashtag is a type of metadata tag used on social networks, such as Twitter and other microblogging services, to apply dynamic, user-generated tagging that helps other users easily find messages with a specific theme or content.

fectively be used in many real-world scenarios. Context-aware techniques use the *context* in which the words appear, which we instinctively know, to play a huge role in conveying meaning. The idea of integrating semantics into word representations is theoretically explained by the distribution hypothesis [8, 9], which states that contextually comparable words have similar semantics. To this end, since 2013, word embeddings [10] have been popular as a de facto starting point to represent the meaning of words. However, static word embeddings–such as Word2Vec [11], GloVe [12] and FastText [13]–generally generate fixed word representations for a given word in a vocabulary. Fundamentally, these techniques cannot be easily adapted to the contextual meaning of a word. Conversely, recent discoveries of dynamic pre-trained representations, such as ELMo [14] and BERT [15], produce dynamic representations of a word based on the context.

However, it should be noted that the success of most state-of-the-art language understanding techniques, such as pre-trained language models, is heavily dependent on the availability of an abundance of training data [15]. In other words, in general, NLP models need a substantial amount of training data to detect patterns to produce acceptable results, as language understanding is a challenging and complex task. To this end, given that labelled datasets are often manually annotated, text classification may suffer significantly from a lack of accurately labelled training data, imposing a significant challenge in real-world scenarios. Moreover, SportsHosts, and any other organisation, may look for a diverse set of text classification use cases over time, making the task of manually labelling a sufficient amount of training data expensive, time-consuming and error-prone.

Further, as social media data are noisy and diverse, this creates additional challenges in the context of linguistic comprehension. The diversity of the vocabulary used in microblogging sites such as Twitter might lead to a higher percentage of *out-of-vocabulary words*, thereby affecting the accuracy of the word representation. Apart from that, as pre-trained language models are trained on datasets containing structured sentences, these models might not scale well to the unstructured and noisy text of social media platforms. The syntactic and grammatical structure of noisy sentences can deviate significantly from standard English sentences (i.e., the training dataset), making it difficult for pre-trained

language models to understand noisy text effectively.

In this thesis, we focus on developing algorithms to derive meaning from noisy text in a low data regime. To this end, we develop novel techniques to improve the NLP model's ability to comprehend noisy text in a limited labelled data scenario, boosting their performance in the real world. We further enhance the accuracies of the NLP models by augmenting text data in a context-aware setting. Finally, we combine our findings, including a cost-effective data annotation methodology, to develop an effective and efficient framework for NLP applications in social media.

1.3 Objectives of the Study

This thesis studies the problem of noisy text classification in the low data regime, focusing on its suitability for application in a real-world environment. This study focuses on addressing the challenges in comprehending noisy and unstructured social media data using a context-aware analysis while reducing the impact of limited labelled data on algorithm performance in a real-world setting.

Firstly, the proposed work focuses on improving text classification accuracies in the low data regime by transferring prior knowledge from pre-trained language models. Further, we explore strategies for improving noisy text representations, focusing on the overall meaning of a particular sentence, thereby improving the accuracy of noisy text-based NLP tasks.

Secondly, to address the data scarcity problem, we focus on generating synthetic sentences to augment the training datasets. However, in a context-aware NLP regime, we cannot effectively use existing text augmentation techniques–such as synonym replacement, random insertion or random swap–to generate more training data. These methods tend to create synthetic sentences with different semantic meanings, thereby introducing many noisy examples to meaning-sensitive NLP tasks, such as sentiment analysis. Analogously, text augmentation techniques, such as back translation, generate noisy examples due to the non-standard spellings and non-grammatical constructions in the social media data.

Most importantly, these text augmentation techniques are limited by their ability to add more diversity and variety to the training data, which is critical when applying machine learning models to real-world use cases.

Thirdly, the automated classification of interest groups via social media has significant commercial value for organisations such as SportsHosts. However, developing accurate and efficient classification methods requires a large amount of annotated training data, which can be costly and time-consuming to create. Thus, methods for reducing annotation costs while maintaining accuracy are also explored in the research reported in this thesis.
Thus, throughout this thesis, we plan to fulfil the following objectives:

- derive meaning-rich word embeddings and sentence representations for noisy and unstructured text to boost the performance of downstream NLP tasks in the low data regime
- improve the accuracy of meaning-sensitive text classification tasks in a low-resource setting by augmenting the training dataset with label-compatible and quality synthetic sentences
- design and develop a cost- and effort-efficient method to annotate social media text data to train NLP models for different text classification tasks across various industry use cases.

1.4 Contributions

Throughout this thesis, we focus on three main challenges of social media text analytics in a real-world environment:

1. performance impact due to the noisy nature of the text data
2. poor accuracies due to extremely limited availability of labelled data
3. high cost and effort required to annotate training data.

Our contributions to addressing these challenges are listed below.

Performance impact due to the noisy nature of the text data.

- We propose using the prior knowledge from pre-trained language models as a strategy to improve the performance of noisy text classification under an extremely low-resource setting. Our findings include a recommended set of hyperparameters for a similar scenario.

- We propose a systematic approach to derive generalisable sentence representations for noisy text, comprising the most important linguistic characteristics, using latent layers of multilayer pre-trained language models. We achieve state-of-the-art performance with a sentence vector obtained with the proposed approach. Further, we release a new probing dataset that can serve as a benchmark dataset for future researchers to study the linguistic characteristics of unstructured and noisy text.

Poor accuracies due to extremely limited availability of labelled data.

- We present a novel text augmentation method that extends the back-translation technique for meaning-sensitive text classification tasks. Further, our reported findings reveal the importance of maintaining the quality of synthetic sentences while adequately increasing the diversity of the augmented dataset to reduce the impact due to overfitting.

- We propose **I**ntelligent **M**asking with **O**ptimal **S**ubstitutions Text Data **A**ugmentation (IMOSA), a state-of-the-art text augmentation method that focuses on progressively generating high quality synthetic data rather than augmenting every single sentence in the original dataset, improving the overall quality and diversity of the augmented dataset. We demonstrate the superiority of the proposed text augmentation technique by evaluating the performance of multiple downstream NLP tasks using a state-of-the-art Transformer-based pre-trained language model as a classifier.

High cost and effort required to annotate training data.

- We propose a novel framework to annotate text data specifically for text classification use cases at a significantly lower cost using crowdsourcing platforms. The

framework consists of multiple steps to address data annotation-related challenges, such as data imbalance, poor annotation quality and annotation cost and effort. A real-world text data annotation experiment using the proposed framework reported over 80% reduction in cost.

- We combine our research findings–pre-trained language models as a strategy to improve the performance of noisy text classification under an extremely low-resource setting, improved sentence representation for noisy texts, state-of-the-art text augmentation technique and a cost-effective text annotation methodology–to develop an end-to-end framework to enable data-driven decision-making based on social media data in a real-world setting.

1.5 Thesis Outline

In Chapter 2, we present an overview of the background information that is necessary to comprehend the contents of this thesis. We review the fundamentals of NLP and machine learning. Further, we discuss language models, pre-trained language models and data augmentation in NLP.

In Chapter 3, we study the feasibility of improving the accuracy of noisy text classification under the low data regime through inductive transfer learning and context-aware word embeddings using pre-trained language models. We then propose a systematic approach to derive generalisable meaning-rich sentence representations for noisy text, comprising the most important linguistic characteristics, using latent layers of a pre-trained language model.

In Chapter 4, we first explore the impact of text augmentation in an extremely low-resource setting and present the importance of label compatibility and the diversity of the augmented dataset to improve text classification accuracy. Next, using a Masked Language Model (MLM), we present a novel text augmentation technique that outperforms state-of-the-art text augmentation techniques.

In Chapter 5, we focus on developing an end-to-end framework to deploy a social me-

dia intelligence solution, in particular, to identify interested target groups effectively and efficiently in a real-world setting. To this end, first, we propose a cost-effective annotation framework to obtain the necessary labelled text data. Next, we present a step-by-step approach to deploying a social media text mining application, combining our findings in Chapters 3 and 4.

Finally, in Chapter 6, we present a summary of our findings and an outlook into the future.

Chapter 2

Literature Review

2.1 Background Preliminaries

This chapter presents a detailed and in-depth review of the literature and current state-of-the-art research that provides the necessary background knowledge and thus sets the stage for the work reported in the remaining chapters. First, concepts related to probability, machine learning and neural networks are introduced. Next, a thorough review of NLP and its applications in social media platforms is summarised. This is followed by an introduction to deep learning and transfer learning in NLP. We subsequently delve into language models and pre-trained language models. Finally, we give an overview of text augmentation.

2.1.1 Probability

The term 'probability' simply refers to the likelihood of something occurring. In the context of probability, a 'trial' or 'experiment' is a procedure that leads to a well-defined set of possible outcomes that can be repeated indefinitely. The 'sample space' defines the collection of all the possible results of an experiment. An 'event' is a non-empty subset of the sample space. Thus, in technical terms, probability is the likelihood of a specific outcome or event occurring as a result of an experiment.

According to the definition, if A is an event of an experiment with n outcomes and S is

the sample space, then the probability of the event A is,

$$P(A) = \sum_{i=1}^{n} P(E_i) \tag{2.1}$$

where E_i denotes the outcomes in A. If each outcome of an experiment occurs with equal probability,

$$P(A) = \frac{\text{No. of outcomes in A}}{\text{No. of outcomes in S}} \tag{2.2}$$

Independent events. Two occurrences, *A* and *B*, are independent if the knowledge that one occurred does not affect the chance that the other occurs. Two events are considered independent if the following conditions hold:

- $P(A|B) = P(A)$
- $P(B|A) = P(B)$
- $P(A \cap B) = P(A)P(B)$

Mutually exclusive. Any two events with non-overlapping outcomes are said to be mutually exclusive. For instance, if *A* and *B* are mutually exclusive, then $P(A \cap B) = 0$. Further, as A and B are disjoint, $P(A \cup B) = P(A) + P(B)$.

2.1.2 Machine Learning

In 1959, Arthur Samuel described machine learning as 'the study that gives computers the ability to learn without being explicitly programmed' [16]. Machine learning focuses on developing mathematical models from a dataset. A dataset is composed of *examples*. An example is a particular instance of data, typically represented as a vector $\mathbf{x} \in \mathbb{R}^d$, where an example consists of d features. Each feature contains the value for one of the data's attributes. In this case, a dataset can be represented as a matrix $\mathbf{X} \in \mathbb{R}^{nxd}$, where n is the number of examples.

Learning and inference are the two main goals of machine learning. To discover patterns, a machine learning model uses a dataset, which is generally known as the training

dataset. This step is commonly known as training the model. Once the model is trained, the model can be used to transform newer data and output predictions.

Machine learning can be divided into two categories: supervised and unsupervised learning. A supervised learning algorithm learns the link between given inputs and outputs (labels) using a training dataset. Each input x_i is typically mapped to a separate target y_i, which is usually represented as a vector of labels y. In contrast, for unsupervised learning, there are no target labels assigned. There are primarily two types of supervised learning: classification task and regression task. The label y_i belongs to one of a predetermined number of classes or categories in classification. y_i is a continuous number in regression.

Linear regression. The formula for a simple linear regression is:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{2.3}$$

where Y_i is the predicted value for the ith instance, $\beta_0 + \beta_1 X_i$ represents a linear function connecting X to Y, and ε_i is an error of the estimate. By searching for the regression coefficient (β_1) that minimises the overall error (ε) of the model, linear regression identifies the line of best fit through training data.

Logistic regression. It is possible to extend the linear regression to perform classification. In the context of binary classification, we deal with two classes: class 0 and class 1. In this case, instead of modelling a straight line, the linear regression is modified to obtain a probability by restricting the output of a linear equation between 0 and 1 using a logistic regression function. The logistic function, also known as the sigmoid function, is defined as:

$$\sigma(z) = \frac{1}{1 + e_{-z}} \tag{2.4}$$

Gradient descent. Gradient descent [17] is one of the most widely used optimisation techniques for training machine learning models by minimising the difference between desired and estimated outcomes. In this context of machine learning, optimisation is the task of minimising the cost function with respect to the parameters of the model. The primary

goal of gradient descent is to use the iteration of parameter updates to minimise a convex function.

Evaluation metrics. Machine learning models are generally evaluated for their performance on a particular task using a test dataset. Accuracy is a common evaluation measure for binary classification tasks and is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

where TP, TN, FP and FN represent the true positive, true negative, false positive and false negative counts, respectively. Additionally, the F_1 score, which is the harmonic mean of precision and recall, is used for multi-class classification:

$$F_1 = 2\frac{PR}{P + R}, P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \tag{2.6}$$

2.1.3 Neural Networks

Neural networks, also known as artificial neural networks (ANNs), are designed to mimic the human brain using a set of algorithms. At its most basic level, a neural network consists of four key components: inputs, weights, a bias or threshold, and an output. In general, a neural network is composed of an input layer, an output layer and zero or more hidden layers (non-output layers). The number of hidden layers in a neural network is commonly used to name it. A one-layer feed-forward neural network, often known as a multilayer perceptron, is a model with one hidden layer. Mathematically, a neural network can be considered a combination of simple functions, such as linear regression, together with an *activation* function, such as the softmax or sigmoid, as follows:

$$\begin{aligned} h &= \sigma_1(W_1 x + b_1) \\ y &= softmax(W_2 h + b_2) \end{aligned} \tag{2.7}$$

where σ_1 denotes the activation function of the first hidden layer. As shown in Equation (2.7), each layer has its own weight matrix W and bias vector b. The calculated output of one layer, h, is provided as an input to the next layer. The last layer, also known as the output layer, produces the neural network's overall output y. This process is known as *forward propagation*. Further, to produce a categorical and Bernoulli distribution, the softmax and sigmoid functions are commonly utilised in the output layer of a neural network.

Back-propagation. Back-propagation [18], short for 'backward propagation of error', is the process of fine-tuning the weights of a neural network based on the error between the desired output (y_i) and the calculated output ($\hat{y_i}$) in the previous iteration using gradient descent. As the name suggests, the gradient calculation propagates backwards through the network. This approach calculates the gradients of the error function with respect to the weights of the neural network as follows:

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial E(X, \theta^t)}{\partial \theta} \tag{2.8}$$

where θ^t represents the error between the actual output y_i and the estimated output $\hat{y_i}$ of the neural network with a particular set of parameters θ at iteration t, and α is the learning rate. The learning rate controls the magnitude of the change propagated in response to an observed error.

The mean squared error is a commonly used error function in back-propagation.

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y_i} - y_i)^2 \tag{2.9}$$

2.1.4 Natural Language Processing

NLP is a technique for teaching computers to understand human speech. We focus on developing models to map an input x consisting of a sequence of words or tokens to an output y using different machine learning tools and techniques. To this end, we discuss the commonly used NLP-specific terminologies and concepts related to our work throughout

this thesis.

Bag-of-words. In some contexts, we use the bag-of-words technique to represent text data as an input for machine learning models. In this approach, we represent a unit of text (phrase, sentence or document) using a vector $x \in \mathbb{R}_{|V|}$ where V is the vocabulary. Each element x_i indicates the number of occurrences (term frequency) of the corresponding vocabulary word in a given text document. *Term frequency-inverse document frequency (tf-idf)* is an extension to the above approach, where we calculate a weighted frequency to measure how important the word is to a document.

Tokenization. Tokenization is the task of splitting up a character sequence into units called *tokens*. Tokenization is commonly applied to split a sentence into tokens. In general, these tokens are referred to as terms or words. A simple strategy to tokenize a sentence is to split all white-space characters. Tokens are frequently used as input for operations such as parsing and text mining. Tokenization is significant because the meaning of a sentence can be easily deduced by examining the words in the text.

Part-of-speech (POS) tagging. The technique of categorising a word in its context is known as POS tagging. To categorise a word into its class, a probability model and extra factors are employed. These classes are known as word classes or lexical categories. A tagset is a collection of such classes that are used for a certain NLP task. Tags representing the most common components of speech may be limited in basic tagsets (e.g., N for noun, V for verb and A for adjective). It is, however, more customary to distinguish between single and plural nouns, verbal conjugations, tenses, aspect, voice, and other details.

Stop words. Stop words are words that contribute only slightly to a sentence in any language. Articles and pronouns are typically categorised as stop words. Examples of a few stop words in English are ‘the’, ‘a’, ‘an’, ‘so’, ‘with’. Bag-of-words-based techniques tend to safely ignore stop words without sacrificing much information from a sentence. However, context-aware NLP approaches want to keep the stop words intact to understand a word’s context accurately.

Dependency parsing. Extracting a dependency between the words of a phrase or a sentence that describes its grammatical structure using a tree structure containing head words and their relationship with child words is known as dependency parsing. Dependency parsing can be used to approximate the semantic relationship between words to improve automatic text understanding.

Zero-shot text classification. Zero-shot learning was traditionally used to classify instances into unseen classes using a model trained on a different set of classes. In the context of NLP, Veeranna *et al.* [19] proposed using a semantic embedding of label and document words and basing the prediction of previously unseen labels on the similarity between the label name and the document words in this embedding. A given sequence x can be classified using a sequence embedding model M_{sent} and a set of possible class names C, as shown in Equation (2.10).

$$\hat{c} = \underset{c \in C}{\arg\max} \cos(M_{sent}(x), M_{sent}(c)) \tag{2.10}$$

Alternatively, Yin *et al.* [20] proposed using a pre-trained Multi-Genre Natural Language Inference (MNLI) [21] sequence-pair classifier as an out-of-the-box zero-shot to perform text classification. Natural language inference focuses on determining whether a 'hypothesis' is true ($entailment$), false ($contradiction$) or undetermined ($neutral$) given a 'premise'. Models based on Bidirectional Encoder Representations from Transformers (BERT) [15] can be used to feed both the premise and the hypothesis through the model as separate segments and train a classification head predicting $entailment$, $neutral$ or $contradiction$.

2.2 Natural Language Processing Applications

NLP applications have recently gained much traction in the industry. While there are many NLP-related applications, in this thesis, we primarily focus on the challenges and limitations of NLP applications in social media data, with a particular focus on the unstructured and

noisy nature of the textual data. Among the plethora of NLP use cases that can be utilised to gain insights into social media data, sentiment analysis and intent classification are widely used in the industry to analyse consumer attitudes and behaviour and make data-driven decisions on business strategies.

Sentiment analysis. Sentiment analysis focuses on extracting user opinion on an entity (e.g., product, person, organisation and place) from text, helping organisations understand the social sentiment of their products, services or brand. To this end, during the last decade, social media platforms have become an invaluable place to monitor online conversations to understand user behaviour for various use cases, such as service quality [22, 23], business performance [24] and tourism [25]. Existing approaches for sentiment analysis can be broadly categorised into three categories: machine learning approach, lexicon-based approach and hybrid approach [26], with machine learning as the most widely used approach. Though machine learning techniques yield high accuracies, sentiment analysis remains an open research field due to several challenges, including negation handling, word sense disambiguation and a lack of sufficient labelled data [26]. Conversely, lexicon methods are popular for sentiment analysis in social media [27]. This method has several drawbacks. For example, the existence of more favourable terms in customer reviews or any other online text source does not always imply that a review is positive or vice versa. In most circumstances, using the same lexicon for scoring texts from different domains is problematic. Thus, throughout this thesis, we propose to focus on improving the machine learning approach with noisy social media data.

Intent classification. Intent, in the simplest term, can be defined as a purpose for action. Intent analysis is the idea of identifying intentions present in textual content and recognising a corresponding intent category for every action indicative of intent in a particular text [28]. Intent Analysis has emerged in a variety of application domains in recent years [29, 30, 31], including market intelligence, advertising and political vote prediction.

Intent classification primarily attempts to capture a plausible future outcome [28] and is different from well-known text mining, such as opinion or sentiment classification, where

they approximate the current state. For example, the sentence *'I like the colour of iPhone 7'* reflects a positive sentiment, but no intention exists. In contrast, the sentence *'I want to buy an iPhone 7'* shows a firm buying intention in the near future. Therefore, verbs and keywords in a piece of text are considered essential features for intent identification. Hence term-based intent analysis [1, 32, 33] has been a popular approach to detecting intent. Further, with the advancement of NLP, researchers have employed more advanced techniques, such as neural networks, attention and transfer learning, as summarised in [34]. Nevertheless, as highlighted by [34], intent analysis using social media data is still challenging and requires deeper analysis. Moreover, according to the authors, the effectiveness of transfer learning and deep learning techniques in this context is still underexplored.

2.3 Deep Learning for Natural Language Processing

For a long time, shallow machine learning models and heuristic-based features or shallow features were used by most techniques that examine NLP problems. These models were limited in their ability to support NLU. Many recent works related to NLP tasks showed that neural models can be effectively used in a variety of tasks in NLP, including, but not limited to, language modelling [35, 36], sentiment analysis [37], machine translation [38], word-embedding extraction [11, 39] and transfer learning with language models [15, 40].

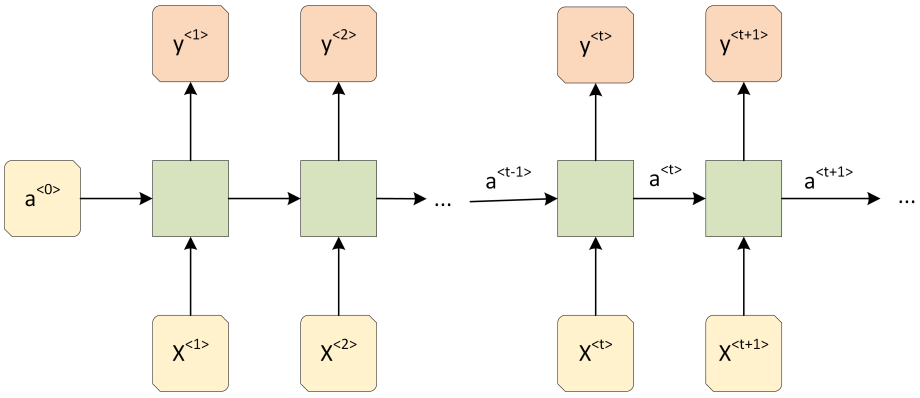

Figure 2.1: RNN model architecture.

Among many neural models, a Recurrent Neural Network (RNN) [41] is a specialised neural-based method that processes sequential information effectively [42]. The main strength of an RNN is the ability to memorise and use the outcomes of previous steps

in the current step; hence, the capacity to model aspects of linguistic structure and meaning based on the preceding sequence (see Figure 2.1). Subsequently, to overcome various shortcomings [43, 44], such as the vanishing gradient problem, different variants of RNN namely Long Short-Term Memory (LSTM) networks [45], Gated-Recurrent Network (GRU) [46] and Residual Network (ResNet) were later introduced. Further, extensions such as Bidirectional Recurrent Neural Network (BRNN) [47] have been proposed to enhance its ability to capture rich linguistic features by considering both past states and future states [48] in the current state.

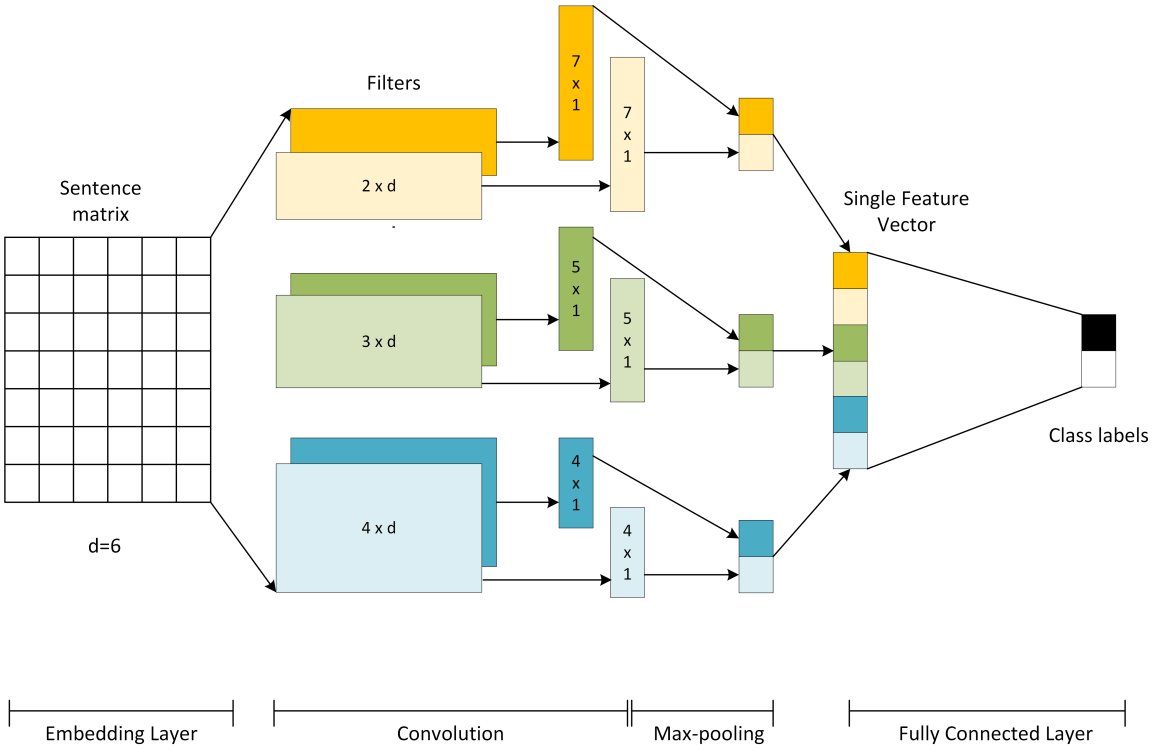

Figure 2.2: CNN architecture proposed by [49] for binary sentence classification.

Conversely, the Convolutional Neural Network (CNN), originally invented for Computer Vision (CV), was first introduced for NLP by Collobort and Watson [50]. Subsequently, CNN models have been shown to be effective at NLP tasks, including semantic parsing [51] and sentence classification [52]. CNN models have demonstrated accuracy in identifying effective clues in contextual windows but preserve no sequential order or long-distance contextual information as illustrated in Figure 2.2. Apart from that, for NLP tasks that require sequence-to-sequence mapping–such as Statistical Machine Translation (SMT), network models that learn to encode a variable-length sequence into a fixed-length vector repre-

sentation and decode a given fixed-length vector representation back into a variable-length sequence, known as Encode-Decoder–were found to be effective [38, 53].

Recently, inspired by the visual attention mechanism found in humans, attention [54] has become one of the most influential ideas in the NLP community. It allows neural models to pay attention only to important information in a sequence, rather than relying on the fixed-length vector representation of a complete sequence, and helps to overcome the inability of the models to remember longer sequences. Attention has been successfully applied in many NLP tasks, including machine translation [55, 56], image captioning [57] and aspect-based sentiment analysis [58]. Further, Zhou *et al.* [59] proposed Attention-Based Bidirectional Long Short-Term Memory Networks (AttBLSTM) to automatically capture the most important semantic information in a sentence, achieving improved performance over classification tasks.

Despite the success of CNN and RNN, the Transformer–a simple yet powerful network architecture based solely on the attention mechanism introduced by Vaswani *et al.* [60] in 2017–was proven to be superior to both RNN- and CNN-based neural models. It uses an Encode-Decoder architecture, as shown in Figure 2.3, primarily based on the attention mechanism to map important linguistic features to the decoder at once rather than sequentially, allowing it to learn long-term dependencies easily. Transformer architecture has been a major contributor to the significant increase in deep learning research in NLP over the past two years. Groundbreaking pre-trained language models–such as Open AI's GPT [61, 62], Google's BERT, Google and Carnegie Mellon University's XLNet [63], Facebook's RoBERTa [64], and Google and Toyota's ALBERT [65]–were mainly based on the Transformer architecture.

2.4 Language Models

Language Model (LM), which provides the word representation and probability indication of word sequences, is the basis for most common NLP tasks, such as sentiment analysis and machine translation. The initial application of LM was based on the time-consuming

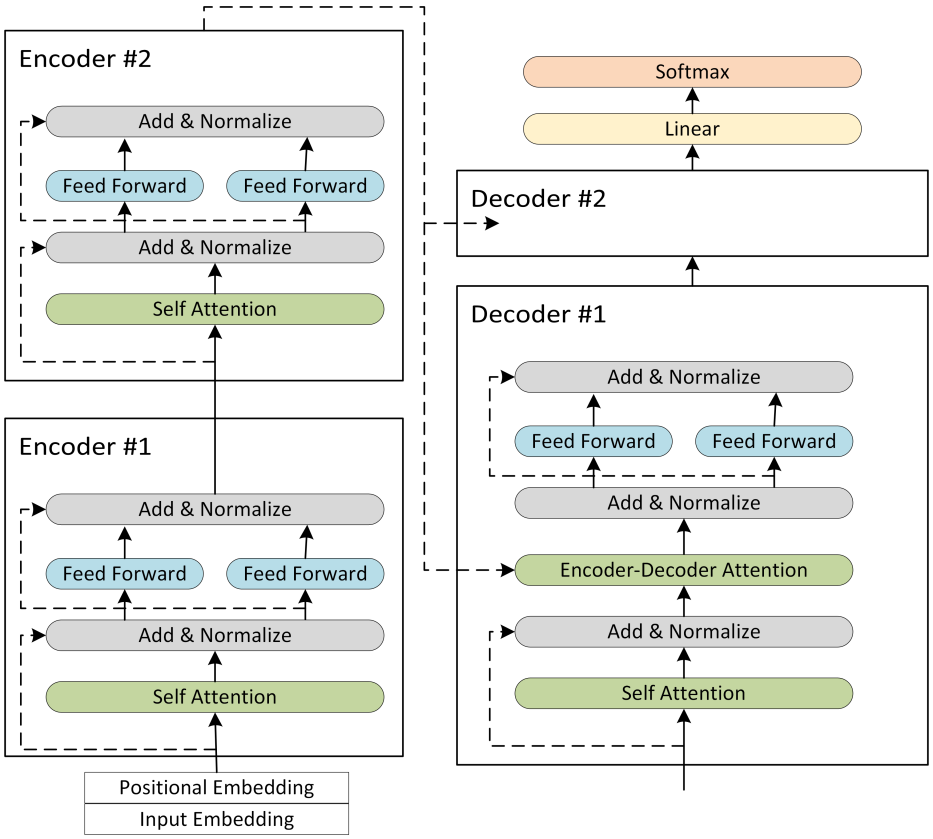

Figure 2.3: Model architecture of the Transformer.

and laborious task of manually writing rules and yet mostly captured only a minimal set of linguistic features. In contrast, static language models focus on developing probabilistic models [66, 67] to retrieve linguistic information by predicting the next word based on a set of words that precedes it. Given a sentence s with a sequence of words $w_1, w_2, \ldots, w_n$, a statistical language model can be modelled as follows:

$$\begin{aligned} P(s) &= P(w_1 w_2 \ldots w_n) \\ &= P(w_1)P(w_2|w_1)\ldots P(w_n|w_1 w_2 \ldots w_{n-1}), \end{aligned} \tag{2.11}$$

Since the model requires an extremely large number of parameters, it was necessary to use an approximation method. N-gram model is a commonly used approximation method and was the state-of-the-art model before the introduction of language models based on neural networks. N is the number of words in the window that is considered the context of a given word. The approximation of N-gram assumes that the current word depends only on previous k words (Markov assumption), which is

$$P(w_t|w_1 \dots w_{t-1}) \approx P(w_t|w_{t-k} \dots w_{t-1}) \tag{2.12}$$

The probabilistic approach to the N-gram model poses major drawbacks, such as the curse of dimensionality [68] and difficulty in handling long-distance dependencies. In 2000, Xu and Rudniky [69] attempted to introduce neural networks into LMs. While their model performed better than standard statistical techniques, it lacked the ability to capture context-dependent features. In 2003, Bengio *et al.* [35] proposed to overcome the curse of dimensionality by learning a distributed representation (i.e., a real valued vector), also known as embedding for words, while introducing the idea to use RNN for LMs to capture contextual information.

Subsequently, many different types of models were proposed for representing words as continuous vectors, including Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation. Word2Vec , proposed by Mikolov *et al.* [11] in 2013, is the most popular word-embedding model which produces distributed representation of words that capture a large number of precise syntactic and semantic word relationships based on continuous bag-of-words (CBOW) or continuous Skip-gram architecture. CBOW is a neural approach to word embedding, and its goal is to calculate a target word's conditional probability given the context words in a given window size. Conversely, Skip-gram is a neural approach to word embedding, where the goal is to predict the context words surrounding it based on a central target word. Word-embedding models [10], including Word2Vec [11], GloVe [12] and FastText [13], have become popular among researchers as a de facto starting point for representing the meaning of words in NLP.

Since early 2018, a paradigm shift has taken place from fixed word embedding for each word to context-based embedding based on language modelling. Table 2.1 summarises the popular context-based language models.

Embeddings from Language Models (ELMo). ELMo [14] was the first deep contextualised word representation model that attempted to address the challenge of modelling complex linguistic characteristics, such as syntax and semantics, and context-based word represen-

Table 2.1: Pre-trained language models.

LM	Architecture	Year
ELMo [14]	Bi-LSTM	2018
UMLFit [70]	AWD-LSTM	2018
Open AI GPT [61]	Transformer	2018
BERT [15]	Transformer	2018
GPT-2 [62]	Transformer	2019
XLNet [63]	Transformer-XL [71]	2019

tations using the internal states of a deep bidirectional language model (biLM). The ELMo vector assigned to a word is a function of the entire sequence containing that word allowing the same word to have different word vectors under different contexts, unlike traditional word embeddings. The model consists of two bi-LSTM layers with 4,096 units and 512 output dimensions, while residual connections are applied between LSTM layers to improve the gradient flow and the model is trained to minimise the negative log-likelihood in both directions, as given in Equation (2.13).

$$\mathcal{L} = \sum_{k=1}^{N} (\log p(t_k|t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k|t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)) \tag{2.13}$$

where $(t_1, \ldots, t_n)$ is a sequence of N tokens, and Θ_x, Θ_{LSTM} and Θ_s are the parameters for the token representation, the softmax layer that separates the two LSTMs and the parameters of the LSTM model, respectively.

To evaluate the linguistic features captured by the different layers of the model, ELMo is used for semantic- and syntax-intensive tasks leveraging representations in different layers of the biLM. The results showed large improvements across a broad range of NLP tasks. Further, experiments revealed that the lexical meaning is better captured in the higher layers, while the linguistic structure is better represented in the lower layers of the model [14].

Universal Language Model Fine-Tuning for Text Classification (ULMFiT). ULMFiT [70] introduces transfer learning for NLP tasks using a pre-trained LM with task-specific fine-

tuning, similar to CV tasks [72, 73, 74]. The model is based on the AvSGD Weight-Dropped LSTM(AWD-LSTM) [75] architecture with an embedding size of 400, three layers and 1,150 hidden activations per layer. ULMFiT follows three steps to ensure good transfer learning results on downstream text classification tasks:

- General LM pre-training–pre-training the language model to capture general properties of language using 28,595 pre-processed Wikipedia articles and 103 million words.
- Target task LM fine-tuning–fine-tuning the LM on the data of the target task by exposing the pre-trained model to the data distribution of the target task.
- Target task classifier fine-tuning–fine-tuning the classifier with two additional linear layers, where the last layer outputs a probability distribution using a softmax activation.

Generative Pre-training Transformer (GPT). Using an approach similar to ELMo, GPT [61] introduces an unsupervised language model, which uses a huge dataset of 800 million tokens of BookCorpus [76] to train an LM, and the model architecture is based on a multilayer unidirectional Transformer model. The key limitation of GPT is its uni-directional nature, making it suitable for only the left-to-right context.

BERT. BERT is the first fine-tuning-based language presentation model that achieves state-of-the-art performance on a large suite of sentence- and token-level tasks, outperforming many task-specific architectures [15]. BERT architecture includes a multilayer bidirectional Transformer [60] and an attention mechanism that learns contextual relations between words (or sub-words) in a text. The Transformer consists of two separate mechanisms–an encoder that processes the input and a decoder that generates a prediction for the task. Since BERT is designed to generate a language model, only the encoder mechanism is used.

BERT is trained bidirectionally on a large corpus of unlabelled text, including the entire Wikipedia and BookCorpus, allowing its models to understand the meaning of a language more correctly. Thus, it could be used effectively for various target tasks, such as sentiment classification and intent detection. Two pre-trained BERT models were first introduced–the

'$\text{BERT}_{\text{BASE}}$', that includes 12-layer bidirectional Transformer encoder block with 768 hidden units and 12 self-attention heads, and the '$\text{BERT}_{\text{LARGE}}$' consisting of 24-layer bidirectional Transformer encoder blocks with 1,024 hidden units and 16 self-attention heads.

Compared to GPT, the most significant contribution of BERT is to enable bi-directional training, and the ablation study confirms the impact of this improvement.

GPT-2. In February 2019, OpenAI published the GPT-2 [62] language model, a successor to GPT with slight modifications to the model, which contains 10 times more parameters (1.5 billion), is trained on 10 times the amount of data and achieves state-of-the-art results on seven out of eight common NLP tasks. Most noticeably, this performance was witnessed with no task-related fine-tuning and trained only in a zero-shot setting.

XLNet. XLNet [63] is a generalised autoregressive pre-training method that introduces a variant of a language modelling, called permutation language modelling, to overcome the limitations in the BERT model due to the masked ([MASK]) training procedure and parallel independent predictions. XLNet showed significant improvement upon BERT across 20 NLP tasks.

2.4.1 Masked Language Models

MLMs predict a 'masked' word based on all past and future words in the sequence. An MLM is traditionally trained by randomly selecting words to be masked, using a special token [MASK], and substituted with a random token. This allows the model to capture bidirectional information to make predictions. The training objective is to recover the original tokens at the masked positions: $\sum_i m_i \log(P(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n); \theta_T)$, in which $m_i \in 0, 1$ indicates whether x_i is masked or not, and θ_T are the parameters in a Transformer encoder.

BERT [15] and RoBERTa [64] are two popular MLMs. These MLM models use self-attention to attend to all past tokens in both directions to learn an embedding for a specific token, mimicking the behaviour of an autoregressive model and stacking many Transformer encoder layers to learn sophisticated and meaningful representations. However, the non-

autoregressive nature allows the computations to be parallelised, thereby significantly reducing the inference time. Given an input sentence $x = (x1, x2, \ldots, x3)$, MLMs such as BERT first choose a fraction (typically 15%) of tokens in x at random and replace them with a special token [MASK], then predicts the masked tokens based on the remaining tokens. Let x_m be the masked tokens and x_r be the remaining tokens; the objective function of MLM can be written as

$$L_{MLM}(x_m|x_r;\theta_{enc}) = -\sum_{t=1}^{|x_m|} \log P(x_t^m|x_r;\theta_{enc}) \tag{2.14}$$

where $|x_m|$ indicates the number of masked tokens.

2.5 Transfer Learning

Data mining and machine learning techniques have already achieved remarkable success in many areas of information engineering, including classification, regression and clustering. However, in general, machine learning models assume that the training data and test data are drawn from the same distribution. When the data distribution changes, mostly, the models need to be redeveloped from scratch. A significant challenge hindering the real-world application of machine learning models is the expensive and tedious task of recollecting necessary training data and rebuilding the models. Transfer learning [77], which focuses on transferring knowledge across domains, is a promising machine learning methodology to solve the problem mentioned above. Transfer learning research is motivated by the fact that previously learned knowledge can be intelligently applied to solve new problems effectively and efficiently.

2.5.1 Transfer Learning in NLP

While deep learning models achieve state-of-the-art accuracy across many NLP tasks, their performance is limited by the availability of a significant quantity of data and the demand for huge computing resources, forcing the exploration of knowledge transfer possibilities. Figure 2.4 shows the complete taxonomy for transfer learning for NLP developed by Ruder

[78], based on the transfer learning taxonomy compiled by Pan and Yang [79]. Throughout this thesis, we mainly focus on two inductive transfer learning techniques: sequential transfer learning and multi-task learning.

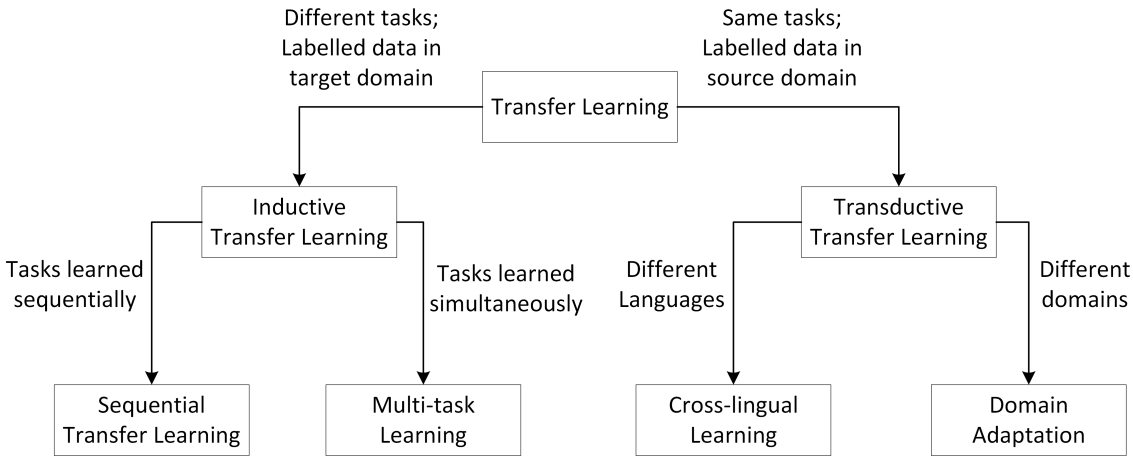

Figure 2.4: A taxonomy for transfer learning [78] for NLP.

Sequential transfer learning. The primary objective of sequential transfer learning [80] is to transfer knowledge from the model trained on a source task to improve the performance of the target model. Sequential transfer learning consists of two steps: pre-training, where a model is trained on the source data, and adaptation, where the knowledge is transferred from a previously trained source model to the target model. This transfer learning approach has shown promising results in CV on the ImageNet Dataset [81]. UMLFiT, ELMo, the Open AI Transformer and BERT have enabled a sequential transfer learning capability for language.

Multitask learning. Inspired by the ability of people to learn a new task effectively, often applying knowledge gathered from previous tasks, multitask learning aims to learn jointly from different tasks, assuming that knowledge captured from a task can be leveraged by another task.

Definition 'Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related, multitask learning aims to help improve the learning of a model for $\mathcal{T}_i$ by using the knowledge contained in all or some of the m tasks' [82].

2.5.2 Transfer Learning with Language Models

With the success of distributed representations [10, 11, 13] of words, neural network models have achieved superior results compared to traditional approaches, such as SVM and logistic regression, mainly due to their ability to capture a linguistic structure of the language and the lexical semantics of words.

In early 2018, Howard and Ruder [70] introduced the ULMFiT method, which was the first transfer learning technique applied to NLP, similar to the transfer learning paradigm in CV. Interestingly, ULMFiT was able to match the performance of models trained from scratch with 100 times more data, only with 100 labelled instances. The paper presented novel techniques to retain previous knowledge and avoid catastrophic forgetting [83, 84, 85, 86] during the transfer learning process. Motivated by the fact that various architectures identify different layer-specific language representations [87], ULMFiT proposed *discriminative fine-tuning*, a technique to tune the model using different learning rates for each layer.

ELMo feeds embeddings as additional features to a customised model of a target task to transfer the previously captured knowledge, while GPT fine-tunes the same base model end-to-end for all target tasks [61]. Strubell and McCallum [88] revealed that the performance of strong neural network models can be further improved, incorporating linguistic structures by combining ELMo with the Linguistically-Informed Self-Attention (LISA) [89] model, a strong, linguistically-informed neural network architecture for Semantic Role Labelling (SRL). Further, Zhang *et al.* [90] proposed an improved language representation model, Semantics-aware BERT (SemBERT), to incorporate explicit contextual semantics from pre-trained SRL, obtaining new state-of-the-art or substantially improving results on 10 reading comprehension and language inference tasks.

Yosinki *et al.* [91] revealed that features learned by different layers of a deep neural network transit from general to task-specific from the first layer to the last layer. To this end, Howard and Ruder [70] introduced discriminative fine-tuning, where they apply different transfer learning rates (gradually decreasing learning rate for lower layers), allowing them

to achieve better results.

2.5.3 Supply Chain for NLP

The advancements in transfer learning techniques for languages have enabled a new AI supply chain for NLP tasks, as illustrated in Figure 2.5.

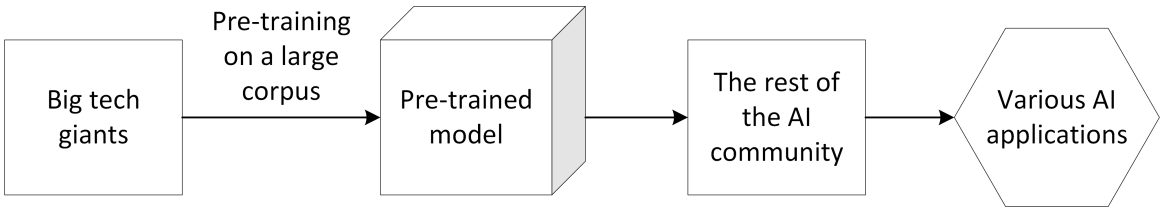

Figure 2.5: AI supply chain for NLP.

Large technology firms, such as Google and Amazon, have published massive pre-trained language models that require a great deal of time and effort to train. The research community and organisations can use transfer learning techniques to extract knowledge captured in a pre-trained model and easily apply it to another task. Numerous experiments have revealed that various NLP tasks achieved state-of-the-art performance along with this emerging paradigm. Thus, the AI community has adopted pre-trained language models as the backbone for downstream tasks instead of developing and training models from scratch. In this case, organisations can use pre-trained language models to solve business problems effectively at a significantly low cost and time.

2.6 Language Understanding

Recently, word embedding [10] has become popular as a de facto starting point for representing the meaning of words. However, static methods–such as Word2Vec [92], GloVe [12] and FastText [13]–generally generate fixed word representations in a vocabulary. Hence, these techniques cannot easily be adapted to identify the contextual meaning of a word. Recent discoveries of dynamic, pre-trained language representations–such as ELMo, a deep contextualised word representation [14] and BERT [15]–produce dynamic representations of a word based on its context. The BERT architecture includes a multilayer bidirectional

Transformer [54] and an attention mechanism that learns contextual relations between words (or sub-words) in a text. The Transformer consists of two separate mechanisms–an encoder that processes the input and a decoder that generates a prediction for the task. BERT–which is trained bidirectionally on a large corpus of unlabelled text, including the entire Wikipedia and BookCorpus–allows its models to understand the meaning of a language more correctly.

Further, several other Transformer-based language models perform well at a broader range of tasks beyond document classification, such as common sense reasoning, semantic similarity and reading comprehension. Transformer-XL [93], a Transformer-based autoregressive model, enables the capture of longer-term dependencies in a sentence and achieves better performance on NLP tasks with both short and long sequences. Generative Pre-trained Transformer 3 (GPT-3) [94], the third-generation language prediction model in the GPT-n series created by OpenAI, is an auto-regressive Transformer model that performs reasonably well on unseen NLP tasks.

These recent models capture many facets of language relevant for downstream tasks–such as long-term dependencies, context and hierarchical relations–to provide state-of-the-art performance [95, 96]. Further, previous research [97, 98, 10] has demonstrated that deep learning models with complex architectures that leverage the contextual meaning of words can significantly improve learning abilities.

2.6.1 Language Understanding with BERT

Goldberg [95] assessed the extent to which the BERT model captures the syntactic structure of a sentence using three stimuli tasks related to subject-verb agreement. Though the results were not directly comparable with previous work due to the BERT's bidirectional nature, the results suggest that purely attention-based BERT models are likely capable of capturing syntactic information at least as well as sequence models, if not better.

Jawahar *et al.* [99] performed a series of experiments using conventional and standard English sentences extracted from books to identify the linguistic information learned by BERT. These experiments were based on the probing datasets developed by [100] using

the Toronto BookCorpus [101], which was one of the two data sources used to train the BERT model. They showed that BERT's intermediate layers encode a rich set of linguistic characteristics, with surface features at the bottom, syntactic features in the middle and semantic features at the top. This indicates that specific regions or layers of BERT are better suited to comprehending different aspects of the English language.

Similarly, Liu *et al.* [102] examined the linguistic knowledge captured by contextual word representations derived from different layers of large-scale neural language models. They showed that the frozen contextual representations are competitive with state-of-the-art, task-specific models in many cases but fail on tasks requiring fine-grained linguistic knowledge. These studies focused only on structured and clean English sentences. They paid little attention to combining the layer representations based on linguistic knowledge to derive a meaning-rich sentence vector. Tenny *et al.* [103] introduced 'edge probing' tasks covering syntax, semantic meaning and dependency relations to study how contextual representations encode sentence structures. Their results using BERT and a few other pre-trained language models concluded that these models encode syntactic phenomena strongly but demonstrate comparable minor improvements on semantic tasks compared to a non-contextual baseline. However, these experiments focused only on the top layer activations of the BERT model, so they may not reveal the full nature of BERT's ability to encode syntactic features. Further, Hewitt and Manning [104] showed that the contextual word representations provided by pre-trained language models, such as BERT, embed syntax trees in their vector representations.

Conversely, Clark *et al.* [96] analysed the BERT's attention mechanism and showed that a specific set of attention heads correspond well to linguistic notions of syntax and coreference. Further, they demonstrated the ability of BERT's attention heads to capture important syntactic information using an attention-based probing classifier.

However, Wang *et al.* [105] more recently concluded that complex pre-trained language models like BERT do not necessarily translate noisy text to better representations. Further, they highlighted that more exploration is needed in this area.

2.6.2 Probing Tasks

Shi *et al.* [106] and Adi *et al.* [107] introduced general prediction tasks to understand the language information captured by sentence vectors. Shi *et al.* [106] investigated whether Neural Machine Translation systems learn source language syntax as a by-product of training by analysing the syntactic structure as a by-product of training. Adi *et al.* [107] proposed a framework that facilitates a better understanding of the encoded representations using, tasks to predict a sentence's length, detect a change in word orders and identify the words in a sentence.

Extending the work of [106] and [107], [100] has introduced ten classification problems known as *probing tasks*. A probing task is a text classification problem that focuses on a grouping of sentences based on the simple linguistic characteristics of sentences. The performance of this classification model depends on the richness of the linguistic information packed into a sentence representation. Further, these probing tasks have been assigned to three groups–surface information, syntactic information and semantic information–based on the primary linguistic feature required to perform the task effectively. The surface information tasks can rely only on surface properties (e.g., sentence length) to perform the classification successfully, and no linguistic knowledge is required. The tasks grouped under syntactic information are sensitive to a sentence's syntactic properties (e.g., depth of the syntactic tree). In contrast, semantic information-related tasks require some understanding of the meaning of a sentence and the semantic structure.

2.7 Text Augmentation

'Data augmentation' is a term used to describe ways of increasing the variety of training examples without explicitly collecting new data. Data augmentation is an essential element for tasks where good generalisation is challenging, primarily when working with small datasets. Further, manually labelling data to develop models in the context of evolving business requirements can be time-consuming and expensive. Thus, accurate and efficient data augmentation techniques could provide a cost-effective method for obtaining

more training data without losing accuracy. Moreover, advancements in neural-model architectures generally demand more data to deliver the expected performance [108, 109]. In contrast to regularisation techniques–such as dropout [110], batch normalisation [110], transfer learning [111] and pre-training [112], and one-shot and zero-shot learning [113, 114]–data augmentation attempts to address the root problem of limited labelled data.

Data augmentation is a widely studied topic in CV, though the usage of data augmentation in NLP has been limited. Since it is non-intuitive to use the signal transformation-based augmentation techniques used in CV for natural languages, text data-specific approaches have been adopted. To this end, rule-based text data augmentations provide efficient methods to generate text sequences based on predefined transformations, such as random insertions and deletions. Easy Data Augmentation [115] is such a method that generates new sentences using token-level perturbation operations, such as random insertions, deletions and swaps. Further, researchers have proposed more advanced methods to replace words using adjacent words in a continuous representation or synonyms from lexical databases, such as WordNet [116, 117]. Zhang *et al.* [118] proposed replacing words with randomly chosen synonyms selected from WordNet based on the semantic closeness to the most frequently seen meaning. Wang *et al.* [119] augmented the training dataset by replacing each word in a Tweet with neighbouring words in a continuous representation. These techniques pay very little attention to the syntax and semantics of sentences, making it hard to maintain invariance. Further, synonyms are very limited, and synonym-based techniques struggle to add sufficient diversity to augmented datasets.

With the popularity of deep generative models, such as GANs [120] and VAEs [121], researchers have attempted to develop novel text generation techniques, combining deep generative models with a sequential decision-making process. Zhang *et al.* [122] proposed using LSTM and CNN models as generators and discriminator, respectively. Further, Yu *et al.* [123] proposed a novel technique combining GANs with reinforcement learning to bypass the generator differentiation problem by directly performing gradient policy updates, achieving significant improvements over strong baselines at that time. However, it is a challenging task to generate meaningful and grammatically correct synthetic sentences while

maintaining label compatibility.

With the rise of context-aware representations, NLP models are more sensitive to the syntactic and grammatical correctness of a sentence [124]. To this end, Kobayashi [125] proposed a novel data augmentation technique by stochastically replacing words based on the context using a bidirectional LSTM-RNN [45] language model. Further, Kobayashi introduced a conditional constraint to ensure label compatibility by embedding the label of a sentence with a hidden layer of the feed-forward network in the bidirectional language model. Nevertheless, the performance improvements by the proposed method were sometimes marginal. Wu *et al.* [126] proposed a text data augmentation technique with more varied substitutions using BERT [15] as an MLM. Additionally, the authors modified the BERT model to embed label information, introducing Conditional BERT, a new conditional MLM, to maintain the label compatibility of the synthetic sentences. Ng *et al.* [127] proposed using MLMs as a denoising autoencoder (DAE) [128] to reconstruct randomly corrupted input tokens by sampling from the underlying natural language distribution. To label preservation, the authors proposed to preserve the original label or use a teacher model trained on original data to obtain a label for more sensitive tasks.

These data augmentation approaches have something in common, in that they generate a predefined number of synthetic sentences for each original sentence provided and pay no or little attention to identifying the most favourable tokens to mask or perturb. The focus is more on augmenting individual sentences with the proposed techniques than improving the overall quality and diversity of the augmented dataset. Apart from that, in most of the experiments [125, 126, 115], the effectiveness of text data augmentation has not been evaluated in a context-aware environment using state-of-the-art Transformer models such as BERT. In this case, the results might not reflect the model's ability to generate meaningful synthetic sentences. However, to some extent, attempts have been made to use bidirectional RNN and CNN models to evaluate the performance of downstream tasks [125, 126].

2.8 Summary

In this chapter, we have conducted a detailed review of existing literature and highlighted the strengths and weaknesses of current state-of-the-art techniques of text comprehension in the low data regime and data augmentation in the domain of NLP. Starting with the background knowledge in probability and machine learning basics necessary for the subsequent chapters, we proceeded to introduce neural network methods and NLP tasks. Further, recent developments in transfer learning in NLP, pre-trained language models and language understanding were discussed in detail.

Section 2.7, on text augmentation, in particular, sets the scene for Chapter 3. We emphasise that the context-aware NLP in a noisy text environment, especially under limited labelled data conditions, is an important research area. Moreover, we explore avenues to further improve the comprehension of noisy text. In the next chapter, we discuss these issues in greater detail and present novel methods to deal with these problems.

Chapter 3

Noisy Text Comprehension

3.1 Introduction

In the previous chapter, we carried out an in-depth review of the methods related to text augmentation in low data regime for NLP. In this chapter, we focus on the intricacies of noisy text comprehension and transfer learning.

Text comprehension occurs when readers derive meaning as a result of intentionally interacting with the text. To this end, in the recent past, word embedding [10] has become popular as a de facto starting point for representing the meaning of words. However, static methods, such as Word2Vec [11], GloVe [12] and FastText [13], generally generate fixed word representations in a vocabulary, and hence these techniques cannot easily be adapted to a contextual meaning of a word. Recent discoveries of dynamic pre-trained representations such as ELMo, a deep contextualised word representation [14], and BERT, a language modelling framework [15], produce dynamic representations of a word based on the context. They capture many facets of language relevant for downstream tasks, such as long-term dependencies, hierarchical relations and context, to provide superior performance [129, 130]. Deep learning techniques with superior algorithms and complex architectures that leverage the contextual meaning of the words [97, 98, 10] can significantly improve learning abilities. It may, however, be noted that the success of these dynamic representation models–to a greater extent–is heavily dependent on the availability of a massive

volume of labelled training data [131, 132].

Conversely, the labelled datasets being often manually annotated, real-world supervised text classification tasks may suffer significantly from a lack of sufficient labelled training data, imposing a significant challenge. Further, different organisations may be interested in entirely different use cases, requiring them to redo the manual annotation process. Moreover, the same organisation may look for a diverse set of target classes over time. These real-world business requirements have exacerbated the lack of training data for NLP tasks, thereby significantly impacting the performance of NLP models in real-world settings.

Apart from that, noisy and diverse social media data creates further challenges. Most NLP tools, such as tagger and parser, including the latest pre-trained language models, have been designed to operate on structured and grammatically correct text. As unstructured and noisy text significantly deviates from the distribution of the original text data used to train these models, the performance of such models reduces drastically [133]. Further, given the diversity and evolving nature of social media content, handling out-of-vocabulary (OOV) words is crucial for the performance of social media content based NLP models.

Considering the aforementioned challenges, this chapter proposes techniques to improve noisy text classification accuracies under a low data regime while exploring the impact of text data augmentation on the proposed techniques. We study the feasibility of transferring prior knowledge from language models pre-trained using clean and structured text for noisy text classification to overcome the problems related to data scarcity. Further, we pay attention to the additional challenges due to the difference between the distribution of the training data and the test data. As the first step, we study the effectiveness of encoding words in a noisy sentence with context-aware representation. Next, we propose a novel technique to improve the model's ability to provide a better encoding to boost the performance of downstream NLP tasks. Additionally, we explore the usefulness of text data augmentation in the context of large-scale language models in a limited training data scenario due to their data-hungry nature. These improvements will enable SportsHosts to effectively use popular microblogging platforms, such as Twitter, which mainly contains

noisy and unstructured content, to understand consumer behaviour and opinion more accurately.

The rest of this chapter is organised as follows. Section 3.2 proposes a mechanism to improve text classification accuracies in the low data regime by transferring prior knowledge from pre-trained language models. Further, it examines the role of text data augmentation in the context of large-scale pre-trained language models. Section 3.3 presents a new generic technique proposed to derive meaning-rich sentence representation for noisy texts using pre-trained multilayer language models. Further, in this section, we present new *noisy* probing datasets that can serve as a novel benchmark dataset for NLP researchers to study the linguistic characteristics of unstructured and noisy text representations.

3.2 Inductive Transfer Learning with Pre-Trained Language Models

In this section, we focus on transferring prior knowledge from pre-trained language models, as a strategy, to overcome the challenges posed by the availability of extremely low training social media data in a text classification task. To this end, the proposed research in this section focuses on an intent classification task, which is a subset of text classification, and enhancing its performance further with pre-trained language models.

Currently, various classifier designs and techniques, incorporating the complexities of the automated intent classification task, have been reported using both heuristic methods and machine learning strategies. Recently, Hollerit *et al.* [1] proposed a binary classification method to identify the commercial intent of a Tweet, applying supervised learning models using word n-grams and part-of-speech n-grams, as features. However, this method fails to capture the semantic representations of the words. Pandey *et al.* [3] presented and evaluated an intent classification model for Twitter posts using semantic features with the help of a convolutional neural network. However, the method uses only static word representations, and the model architecture makes it difficult to disregard the noise and focus on its relevance [60]. Most of these approaches [1, 3, 33] leverage bag-of-words rep-

resentations or static embeddings learned from shallow neural networks, limiting these techniques since they suffer from the absence of dynamic representations of the words in a sentence. The dynamic representation is crucial as it enables understanding of the human intentions contained in a sentence.

3.2.1 Methodology

Our methodology proposes using an attention-based pre-trained language model with sequential transfer learning, a type of inductive transfer learning, to address the scarcity of data when working with noisy social media data in a context-aware setting. The proposed methodology uses BERT, a Transformer-based language model, to derive context-aware representations for the words in a noisy text, while its sub-word-based tokenization helps deal with misspelling and OOV words in a social media text. We leverage a sequential fine-tuning process to transfer the prior knowledge learned from a large corpus. Apart from that, we propose using the back-translation method to augment text data to study its impact in the low data regime, particularly in a context-aware learning setting involving complex deep learning models such as BERT.

The following sections present the main components of the proposed technique and the process to transfer the prior knowledge.

3.2.1.1 Architecture of the Pre-Trained Language Model

BERT is the first fine-tuning based language presentation model that achieves state-of-the-art performance on a broad suite of sentence-level and token-level tasks, outperforming many task-specific architectures [15]. BERT architecture includes a multilayer bidirectional Transformer [60] and an attention mechanism that learns contextual relations between words (or sub-words) in a text. The Transformer consists of two separate mechanisms–an encoder that processes the input and a decoder that generates a prediction for the task. Since BERT is designed to generate a language model, only the encoder mechanism is used.

BERT is trained bidirectionally on a large corpus of unlabelled text, including the entire Wikipedia and BookCorpus, allowing the model to understand the meaning of a language

more accurately than a static language model. Thus, it could be used effectively for various downstream NLP tasks, such as sentiment classification and intent detection. Two pre-trained BERT models were first introduced–'$BERT_{BASE}$', that includes 12-layer bidirectional Transformer encoder block with 768 hidden units and 12 self-attention heads and also '$BERT_{LARGE}$' consisting of 24-layer bidirectional Transformer encoder blocks with 1024 hidden units and 16 self-attention heads.

The processes of the tokenization of an input sentence for the BERT model involves splitting the input text into a list of tokens that are available in the vocabulary. To deal with the words not available in the vocabulary, BERT uses a technique called byte-pair-encoding (BPE) [134] based WordPiece tokenization[135]. '$BERT_{BASE}$-uncased' version of the BERT models convert all the words of an input sentence to lower-case and uses a vocabulary of 30,522 words.

The input layer representation is a summation of WordPiece embeddings [135], positional embeddings and the segment embedding. Since Transformers do not encode the sequential nature of an input sentence, positional embedding is used to introduce a temporal property. Segment embedding is used to distinguish a sentence pair, and it has no impact on a task based on a single sentence, such as text classification. A special classification embedding ([CLS]) is prefixed as the first token of a sentence, and a special token ([SEP]) is appended as the final token. The final hidden state corresponding to the [CLS] token is used as the aggregate sequence representation for classification.

3.2.1.2 Inductive Transfer Learning

Transfer learning refers to the improvement of learning of a particular task by infusing the knowledge from prior learnings of a related task. Transfer learning has played an essential role in many NLP applications [136, 11], and the learning strategy improves the performance on the target task by leveraging the knowledge gained from a different but a related concept or skill [137, 138]. Recently, Universal Language Model Fine-tuning (ULMFiT), introduced by Howard and Ruder [70], was seen as an effective inductive transfer learning method that can be applied to any task in NLP. However, with a similar approach,

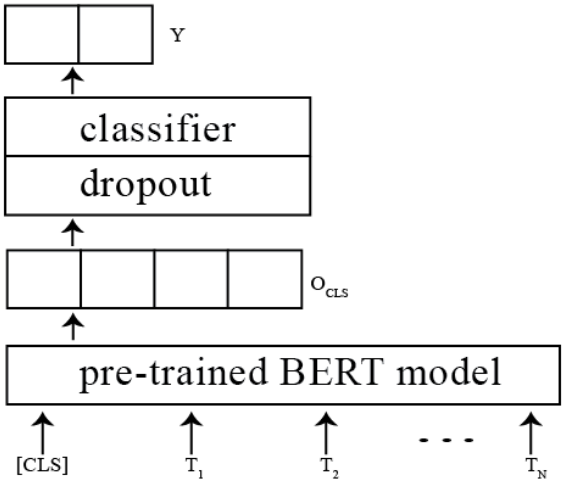

Figure 3.1: The architecture of the BERT model extended for multi-class classification. T_i represents the WordPiece tokens of an input sentence. [CLS] is the special token introduced for classification tasks. O_{CLS} is the final hidden state corresponding to [CLS]. Y is the classification probability vector.

the BERT model achieved superior state-of-the-art results [15]. In the proposed method, we use sequential transfer learning, the most frequently used inductive transfer learning technique, where we transfer information from the pre-trained BERT model to improve the performance of the intent classification task in the low data regime, as discussed in Section 3.2.1.3 below.

3.2.1.3 Fine-Tuning

An intent classification, viewed as a multi-class classification problem with a predefined set of intent categories, can be accurately modelled using BERT. As shown in Figure 3.1, each Tweet can be fed into the BERT model after tokenizing the Tweet into WordPiece tokens $T = [[CLS], T_1, T_2, \ldots, T_N]$, to obtain the output $O = [O_{cls}, O_1, O_2, \ldots, O_N]$.

By leveraging the hidden state of its first special token ([CLS]), denoted $O_{cls} \in \mathbb{R}^H$, where H is the number of hidden units in the BERT model, the intent of each sentence S_i is predicted [15] as

$$Y^i = \text{softmax}(WO^i_{cls} + b) \tag{3.1}$$

The only new parameters to be added [15] during the fine-tuning are for the classification layer $W \in \mathbb{R}^{K \times H}$ and also for $b \in \mathbb{R}^K$, where K is the number of classifier labels and H

is the number of hidden units. Further, a dropout layer is added before the classification layer, with the dropout probability set to 0.1. This extension of the BERT model for multi-class classification is shown in Figure 3.1.

To train the model, first, the standard Softmax function is applied to normalise the output of the classification layer $Y \in \mathbb{R}^K$ into a probability distribution of K probabilities. Then, the model is fine-tuned simultaneously, by considering all the parameters of BERT along with the classification layer weights W for minimising the negative log-likelihood objective function.

3.2.1.4 Back Translation

We use back translation to create synthetic sentences to augment the training data to reduce the impact of labelled data scarcity in a context-aware NLP regime where complex and huge deep learning models are in action. In NLP, back translation is a text augmentation method, first proposed by [139] for context-aware Neural Machine Translation, which works as follows:

1. consider an input text from a source language (e.g., English)
2. translate the input text to an intermediate target language (e.g., English $\rightarrow$ Germany)
3. translate back the previously translated text into the source language (e.g., Germany $\rightarrow$ French)

In this step of data augmentation, we combine sequential transfer learning with text data augmentation to tackle the data scarcity while improving the accuracy of the NLP task through context-aware word and sentence representations.

3.3 Noisy Text Comprehension

In this section, we focus on the complex problem of machine reading comprehension to improve the performance of NLP tasks using meaning-rich sentence representations. This problem has been traditionally studied by researchers as a problem of NLU, a sub field of

NLP. Among several challenges facing NLU, the representation of sentences incorporating all their linguistic elements is considered highly complex. Due to the benefit of accurate sentence representations (e.g., sentence classification, text summarisation and machine translation), it has become necessary to develop new NLU methods that incorporate all linguistic elements to improve accuracy. While a plethora of techniques have already been proposed, representing sentences as vectors of real numbers in high dimensional continuous space is attracting considerable attention [140, 141].

For vector representation, both word and sentence embeddings have influenced the representation following the rapid rise of Word2Vec [92]. Recently, unsupervised, pre-trained language models, such as BERT [15], have been successful in achieving state-of-the-art results in various NLP tasks (e.g., sentence-level text classification), thereby introducing a major paradigm shift in sentence representations. It may be noted that, unlike the shallow word vector models (i.e., Word2Vec [92] and GloVe [12]), the deep models, such as BERT, are contextual.

Widespread use cases, such as sentiment analysis and intent analysis, mandate sophisticated sentence representations since these models essentially involve identification of intricate linguistic patterns [142, 143]. With the increasing proliferation of social media data, such as Tweets, it has further become inevitable to represent noisy texts as vectors to improve the model performance. For this reason, the BERT model is being applied extensively to Tweets to achieve state-of-the-art accuracies [144, 145, 146, 147].

However, the application of pre-trained language models, such as BERT, in such scenarios is not easy because the Tweets follow a different distribution [148, 149] than the training inputs. While the BERT model is pre-trained on BookCorpus and English Wikipedia, the Tweets exhibit a significant deviation from this mainstream English language usage. Further, such challenges become extremely overwhelming as Tweets cover different domains (e.g., day-to-day activities, sports, politics and science), and hence are significantly different. For these reasons, the language representation should clearly express non-task-specific general-purpose priors for solving AI-tasks [150].

Although BERT is a general-purpose language model, the reason behind its overall suc-

cess has not been understood clearly. In [95] and [99], efforts were made to understand BERT's ability to learn the structure and syntax of the English language. It was observed that different layers and regions of BERT capture different traits of the English language. However, it is not reported how these findings can enhance the quality of word or sentence embeddings. Further, given that BERT is trained on datasets containing structured sentences, it is not clear whether BERT's sentence representations can scale to unstructured, noisy text data on social media. Apart from this, there is also a recent emergence of various pre-trained language models comprising of multilayer architectures [151]. Thus, a technique based on the latent representations of multilayer models is vital to optimising the vector representations so that these can be used for use cases involving unstructured and noisy texts.

To address these research gaps, we use BERT as the multilayer pre-trained language model and appropriate Tweets to represent noisy texts. We propose a systematic approach to derive a diverse set of sentence vectors combining and extracting various linguistic characteristics. For this, we have developed new probing datasets using noisy texts based on the definition of specific probing tasks in [100] to analyse BERT's behaviour across different linguistic territories centred on noisy texts. We derive generalisable sentence representations for noisy texts, comprising the most important linguistic characteristics. More specifically, our key contributions for enabling BERT in deriving meaning-rich sentence representation from the noisy text are as follows:

- New *noisy* probing datasets can serve as benchmark datasets for future researchers to study the linguistic characteristics of unstructured and noisy texts. These datasets are available in the public domain (`https://bit.ly/3rK0g7P`) and available on request.

- The proposed novel methodology allows researchers to dissect the BERT model and systematically combine the latent layers and token embeddings to derive various sentence representations capturing different linguistic characteristics. This allows studying the linguistic comprehension of multilayer language models effectively.

- Being generic, the proposed technique can be to generate sentence vectors using a pre-trained multilayer language model.

The sections below discuss the probing dataset generation approach and the strategy to generate sentence embeddings focusing on linguistic richness. The proposed method uses probing tasks to efficiently validate the BERT's ability to capture linguistic information and to derive meaning-rich sentence representations for noisy and unstructured text.

In this methodology, we propose a novel technique to generate sentence embeddings by bisecting BERT into regions and then combining the hidden layers and token vectors using two pooling operations. This allows us to analyse a diverse set of sentence vectors and their ability to capture linguistic information effectively. Next, we discuss our approach to generate probing datasets covering five probing tasks under noisy text conditions. These noisy probing datasets are crucial in determining sentence vector's ability to capture necessary linguistic patterns to classify sentences to the target classes of each probing task.

Further, we propose a systematic approach to study the linguistic behaviour of multilayer, pre-trained language models by dividing the layers into multiple regions. This framework can be easily extended to study the language comprehension capabilities of similar multilayer language models.

The details of the methodology and its components are presented below.

3.3.1 Sentence Vector Generation

Our proposed methodology uses pre-trained language models to generate sentence representations. We use the '$\text{BERT}_{\text{BASE}}$-uncased' model [15] to obtain word embeddings from different hidden layers to produce sentence vectors. This allows for exploration of the linguistic features of unstructured and noisy text, such as Tweets, as learned by different regions (see Figure 3.2) or different hidden layers of the BERT model.

Further, apart from this, we use pre-trained Word2Vec [92] and Stanford's GloVe [12] models to derive sentence vectors. In contrast to BERT, although these models are shallow and non-contextual, they offer 10 to 100 times more vocabulary, thereby providing a vibrant vocabulary to outweigh the benefits of a context-aware pre-trained model with

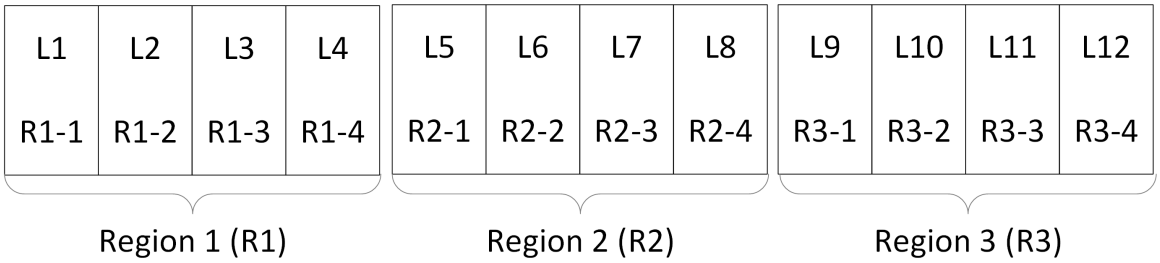

Figure 3.2: The twelve layers (L1 to L12) of the $\text{BERT}_{\text{BASE}}$ model have been partitioned into 3 regions. Rn-i represents the i^{th} layer in the n^{th} region.

a minimal vocabulary (e.g., BERT) in a noisy language setting. Moreover, a word vector trained with the GloVe algorithm using two billion Tweets also enables the performance impact of Twitter-specific pre-trained language models to be evaluated.

The following section explains the strategy to generate multiple sentence embeddings using the pre-trained $\text{BERT}_{\text{BASE}}$-uncased model. It may be noted that for the remaining paper the term BERT is used to represent $\text{BERT}_{\text{BASE}}$-uncased.

Sentence representations using multilayer pre-trained language models. An input sentence is represented as set of input tokens $T = [t_0, t_1, \ldots, t_n]$, where t_o is the special [CLS] token that needs to be prepended for the out-of-the-box pooling schema to work. BERT produces a set of hidden layer activations $H^0, H^{(1)}, \ldots, H^{(L)}$, where $H^{(l)} = [h_0^{(l)}, h_1^{(l)}, \ldots, h_n^{(l)}]$ are the activation vectors of the l^{th} hidden layer. We have ignored the H_0 which consists of non-contextual WordPiece embeddings to generate sentence representations.

To generate a sentence representation based on multiple hidden layers, we propose to generate token representation vector w_i for each token t_i in T using a *layer pooling* strategy. For this, three layer pooling strategies are studied–(i) SUM-layer-strategy, (ii) MEAN-layer-strategy and (iii) CONCAT-layer-strategy. The SUM-layer-strategy and the MEAN-layer-strategy calculate the sum and mean of all the activation vectors $h_i \in \mathbb{R}^d$ of the selected hidden layers, respectively, producing $w_i \in \mathbb{R}^d$, where d is the size of the hidden vector h. Thus, for each sentence, the Mean-layer-strategy and SUM-layer-strategy produces a matrix $W \in \mathbb{R}^{n \times d}$. Conversely, the CONCAT-layer-strategy concatenates the corresponding hidden activation vectors h_i in the order of the layer numbers to generate $w_i \in \mathbb{R}^{kd}$, where k is the number of BERT layers selected to generate the sentence representation.

The CONCAT-layer-strategy produces a sentence representation $W \in \mathbb{R}^{n \times nd}$.

Then, to derive the sentence vector $S = [s_1, s_2, ..., s_{||w_i||}]$, we apply multiple *token pooling* strategies for the sentence representation W (obtained after applying the layer pooling strategy), where each token representation w_i is a row. For this, we study two token pooling operations–(i) MEAN-token-strategy and (ii) MAX-token-strategy. MEAN-token-strategy and MAX-token-strategy are calculated as $s_j = \underset{1 \leq j \leq n}{\mathbb{E}} W_{ij}$ and $s_j = \max_{1 \leq j \leq n} W_{ij}$, respectively. Further, the proposed MEAN-MAX-token-strategy concatenates the MEAN-token-strategy and the MAX-token-strategy output vectors to derive a sentence vector of twice the size of w_i.

Table 3.1: Strategy to generate sentence embeddings from each region (see Figure 3.2) of the BERT model.

Layers	No. of Layers	Layer Pooling	Token Pooling
Rn-1	1	-	mean, max
Rn-2	1	-	mean, max
Rn-3	1	-	mean, max
Rn-4	1	-	mean, max
Rn-1, Rn-2	2	sum, mean, concat	mean, max
Rn-3, Rn-4	2	sum, mean, concat	mean, max
Rn-1 to Rn-4	4	sum, mean, concat	mean, max

Note: Rn-i represents the ith layer in the nth region. We combine each layer pooling strategy with every token pooling strategy across identified layers to generate multiple sentence embeddings. Layer pooling is not applicable for the sentence embeddings generated using a single vector.

As shown in Figure 3.2, for each region Rn ($n \in 1, 2, 3$) different combinations of four layers have been considered to generate sentence embeddings. We apply the layer pooling and token pooling strategy combinations, as listed in Table 3.1, across each BERT region Rn to systematically generate a diverse set of sentence embeddings using the pre-trained BERT model.

Sentence-BERT. Our experiments also utilise the state-of-the-art sentence embedding model Sentence-BERT (SBERT) [152], which uses Siamese and triplet network structures to derive semantically meaningful sentence vectors from the pre-trained BERT model. We propose using a pre-trained model optimised for Semantic Textual Similarity (STS), as this

model is recommended for general-purpose use. SBERT uses a mean pooling strategy to derive sentence vectors from word embeddings.

Static embeddings. We propose using two shallow pre-trained models, namely Word2Vec and GloVe, to generate sentence vectors for unstructured and noisy sentences. These language models are rich in vocabulary compared to BERT. Social media data such as Tweets frequently lack grammatical structure and contain misspelled words and acronyms. Hence, a language model that ensures a lower percentage of OOV words may provide better sentence representations than a deep pre-trained model with a smaller vocabulary. In contrast, BERT uses a technique called byte-pair-encoding (BPE) [153] based WordPiece tokenization [154] to deal with OOVs.

We use the MEAN-token-strategy to derive sentence embeddings using Word2Vec and GloVe.

3.3.2 Noisy Probing Datasets

Probing datasets have a crucial role in the proposed study, as they validate the model's ability to comprehend linguistic characteristics. Studies reported earlier (e.g., [100]) have focused only on language comprehension of structured and grammatical sentences. Hence, the existing probing datasets [100] contain structured and grammatical sentences and rely on the pre-trained Probabilistic context-free grammar (PCFG) model [155] and part-of-speech, constituency and dependency parsing information provided by the Stanford Parser. Although the PCFG model reported close to 87% accuracy for regular English sentences, it is poorly suited for noisy texts [156]. Further, the available Twitter-specific dependency parsers reported a low overall accuracy level with further reductions if the test set topics differed from the training dataset. Thus, we propose using noisy datasets that have been manually annotated with the required linguistic labels to generate quality probing datasets from noisy texts. While the dataset's specifications are essentially based on those proposed by [100], we avoid using automatic part-of-speech or automatic dependency parsing as suggested by them. We use 'Tweebank v2', a collection of English Tweets annotated in

Universal Dependencies [157], as it can be exploited to generate the required noisy probing datasets.

Earlier research [157] did not focus on specific aspects of linguistics, such as dependency parsing information. Due to the unavailability of these linguistic labels, in this study, we work only with key five probing tasks. This does not cause any major disadvantage on the analysis as the key probing tasks are selected such that they continue to cover the three important linguistic categories (i.e., surface, syntactic and semantic), thereby enabling us to analyse the richness of the sentence vectors across all three levels of linguistic information and ensuring the quality of the findings. Further, we have introduced additional criteria explained below to adapt the dataset to noisy conditions. The probing tasks in this study are explained in the following sections:

Sentence length. In this classification task, the goal is to predict the sentence length in eight possible bins (0–7) based on their lengths; 0: (5–8), 1: (9–12), 2: (13–16), 3: (17–20), 4: (21–25), 5: (26–29), 6: (30–33), 7: (34–70). These bins are the same as those proposed earlier [107]. This task is referred to as 'SentLen'.

Word content. We consider a 10-class classification task with 10 words as targets considering the available manually annotated instances. The aim is to predict which of the target words appears in the given sentence. Words that are not part of vocabulary are split by BERT into sub-words and characters. In this case, word embeddings might not reflect the best meaning of the word. Hence, we propose to use only the words that appear in the BERT vocabulary as target words. We constructed the data by picking the first 10 lowercased words occurring in the corpus vocabulary ordered by frequency and having a length of at least four characters. This restriction helps to improve the reliability of the dataset as this is a noisy dataset. Further, each sentence contains only a single target word, and the word occurs precisely once in the sentence. The task is referred to as 'WC'.

Bigram shift. The purpose of the Bigram Shift task is to test whether an encoder is sensitive to legal word orders. Two adjacent words in a Tweet are inverted, and the classifier

performs a binary classification to identify inverted and non-inverted Tweets. The task is referred to as 'BShift'.

Tree depth. The Tree Depth task evaluates the encoded sentence's ability to understand the hierarchical structure by allowing the classification model to predict the depth of the longest path from the root to any leaf in the Tweet's parser tree. The dataset contains six different classes (i.e., two to seven) based on the tree depth. The task is referred to as 'TreeDepth'.

Semantic odd man out. The Tweets are modified by replacing a random noun or a verb o with another noun or verb r. The task of the classifier is to identify whether the sentence gets modified due to this change. The task is called 'SOMO' in the paper.

These five probing tasks, covering the three key linguistic information levels are presented in Table 3.2.

Table 3.2: Grouping of probing tasks.

Group	Probing Tasks
Surface information	SentLen, WC
Syntactic information	BShift, TreeDepth
Semantic information	SOMO

3.3.3 Sentence Vector Evaluation Framework

The most commonly used approach to generate sentence vectors is to average the BERT output layer (BERT embeddings) or by using the output of the first token (the [CLS] token). We extend the common sentence vector generation with our sentence embedding generation technique and combine it with the new probing datasets to develop a sentence vector evaluation framework, as shown in Figure 3.3. This framework enables us to assess the ability of various sentence vectors to capture linguistic information that can be useful for various downstream tasks. Probing datasets consist of the noisy datasets we developed using manually annotated Tweets. As discussed in Section 3.3.1, the Embedding Generator shall generate a diverse set of sentence vectors based on the BERT model while generating

sentence vectors using various other pre-trained models. Next, sentence vectors are forwarded to a classification model. We propose to use a logistic regression (LR) model and a multilayer perceptron (MLP) model to analyse the relationship between different sentence vectors and the shallowness or the deepness of the network.

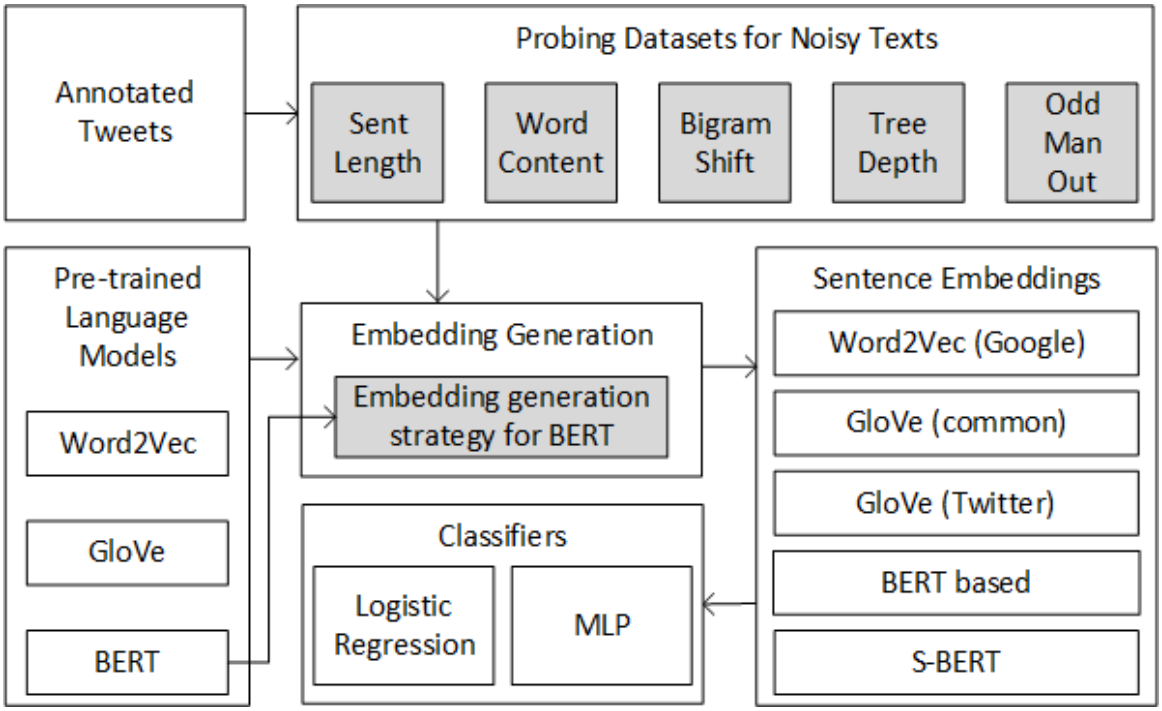

Figure 3.3: Framework.

3.4 Experiments

This section presents various experiments to evaluate the proposed methods' effectiveness in improving noisy text classification accuracy under a low data regime. Section 3.4.1 presents experiments related to inductive transfer learning technique with the BERT language model. Section 3.4.2 explains the experiments conducted related to new sentence vectors and their linguistic behaviour across different probing tasks. At the end of each experiment, we analyse, discuss the results and provide a conclusion.

3.4.1 Experiments on Transfer Learning with Language Models

The following experiments are carried out to study the effectiveness of transferring prior knowledge from pre-trained language models for noisy text classification tasks. Further, we

investigate the application of text data augmentation to boost the classification accuracies in a limited labelled data context.

3.4.1.1 Dataset

As a benchmark, a dataset developed and studied earlier [33] is considered. This dataset contains 2,130 manually annotated Tweets across seven intent categories, as shown in Table 3.3. Table 3.3 also shows the distribution of the dataset in these seven categories.

Table 3.3: Composition of the dataset.

Intent Category	Number of Tweets
Career	159 (7.46%)
Event	321 (15.07%)
Food	245 (11.50%)
Goods	251 (11.78%)
Travel	187 (8.78%)
Trifle	436 (20.47%)
Non-intent	531 (24.92%)

Let

D_T be the entire labelled data comprising of 50 instances for each intent category were randomly sampled (to simulate limited data scenario)

D_V be the remaining labelled data left unused.

We perform the hyperparameter tuning for the BERT model using the five-fold cross-validation by taking only 10 random instances for each intent class (i.e., D_U) from D_T to train the model, and D_V is used to test the model (similar to [33]), as depicted in Figure 3.4.

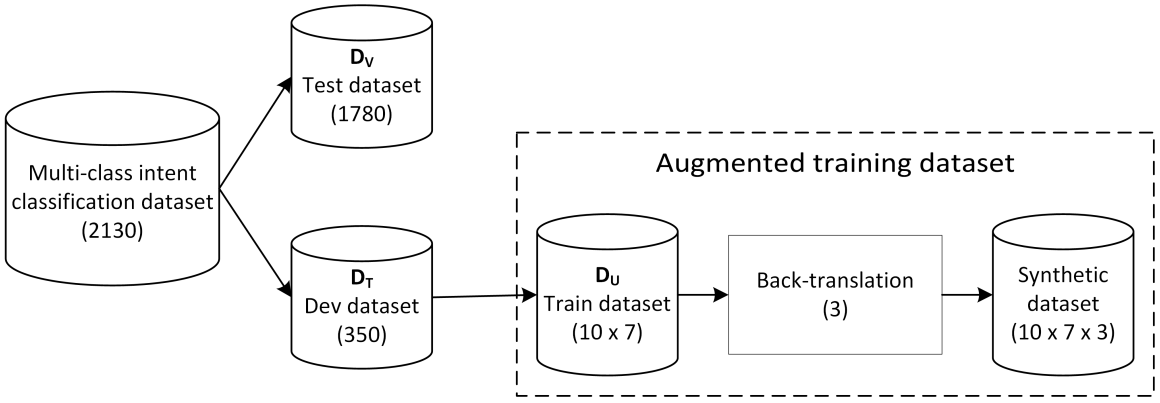

Figure 3.4: Data splitting for fine-tuning.

3.4.1.2 BERT Fine-Tuning

The transfer learning is through fine-tuning the BERT model, as discussed in Section 3.2.1.3. To fine-tune, it is recommended to set most of the BERT model parameters to the original values assigned during pre-training. However, a range of appropriate values for the batch size, learning rate and number of epochs across specific text mining tasks are reported [15, 158]. To meet our requirements, we explore the optimal task-specific hyperparameters for noisy text classification under the limited availability of labelled data.

To derive the optimal batch size, learning rate and number of training epochs, we run an exhaustive search over the following task-specific hyperparameters of the extended BERT model. Apart from the range of possible values recommended for the hyperparameters [15], we also introduce values for tiny batch sizes (i.e., four and eight), since we are using a very small set of labelled-data for fine-tuning. The Adam optimiser ($\beta_1 = 0.9, \beta 2 = 0.999, L2$ weight decay of 0.01) [159] with a learning rate warmup over the first 10% of the training steps, and linear decay of learning rate afterwards (similar to [15]) is used to optimise the objective function. We use *accuracy* as the evaluation metric. The hyperparameter settings chosen for our experiments are

- Batch size: 4, 8, 16
- Learning rate (Adam): 2e-5, 3e-5, 4e-5, 5e-5
- Number of epochs: 3, 4.

To obtain the test set accuracies, the ten-fold cross-validation is carried out with 10 randomly sampled instances from D_T as training data, and using D_V as test data. The cross-validation prevents the model from overfitting the data. As the fine-tuning can sometimes be unstable due to the small training data set, several random restarts for each cross-validation experiment are performed.

For the experiments, we use the pre-trained BERT models provided by the PyTorch-Transformers library.[1] In our simulation experiments for very small training datasets, we

[1]https://github.com/huggingface/transformers

observed the best performance to be consistently obtained for the mini-batch sizes four and eight. We also observed the optimisation difficulties (i.e., a high variance in scores between the folds) associated with large batch sizes during the k-fold cross-validation due to overfitting. In contrast, small batch sizes achieved the best training stability, indicating improved generalisation performance.

3.4.1.3 Data Augmentation with Back Translation

To explore the influence of text data augmentation, we augment the training dataset using the back-translation technique across three target languages to create a new dataset D_S that consists of three synthetic sentences for each sentence in D_U, as shown in Figure 3.4. Next, similar fine-tuning steps, as discussed in Section 3.4.1.2, are carried out using the augmented training dataset as training data.

3.4.1.4 Results and Discussion

Table 3.4 presents the accuracies obtained by transferring prior knowledge through fine-tuning the BERT model with a limited labelled dataset and an augmented dataset. We consider accuracies reported by Wang *et al.* [33] as the benchmark. Experiments with fine-tuning the BERT model, as depicted in Figure 3.1, shows the performance of intent classification with the same dataset and a similar training conditions as [33]. Further, our next experiment shows the performance boost due to text augmentation in a limited data context.

Table 3.4: Noisy text classification accuracies.

Class	Wang's	BERT (fine-tuning)	BERT + Augmentation
Career	45.73	42.62	66.99
Event	27.13	48.8	60.94
Food	54.63	66.79	83.46
Goods	43.25	47.21	60.26
Non-intent	35.56	39.32	40.65
Travel	58.64	56.85	71.4
Trifle	20.04	41.98	49.36
Micro-F1	42.21	50.75	58.69
Macro-F1	40.71	49.08	61.87

The effectiveness of the inductive transfer learning with a pre-trained language model becomes clear with the results of several experiments reported in the previous section. We note that the magnitude of the performance gains using the pre-trained language model is significant, even with minimal data being used for training. Despite using only ten instances from each intent category as training data, the BERT model, fine-tuned only with only four epochs, has performed remarkably well. It resulted in competitive accuracies against the more sophisticated semi-supervised learning models that require complex algorithms [33]. The fine-tuned BERT model obtains a significant absolute accuracy (Macro-F1) improvement of 8.4% over the state-of-the-art semi-supervised learning accuracy reported by Wang *et al.* [33]. Several experiments carried out clearly validate the effectiveness of the proposed technique. These results suggest that the pre-trained language models can have satisfactory performance even with noisy texts, and hence, they can be effectively utilised for other NLP applications having noisy data.

Conversely, it was evident that the accuracy of two classes–Career and Travel–in the benchmark dataset [33] have suffered though the overall accuracy for BERT fine-tuning was higher than the benchmark. However, text augmentation has shown promising results with a remarkable improvement of over 20% over the state-of-the-art benchmark accuracy while showing significant improvement across all the individual categories. The results confirm that while fine-tuning pre-trained models were effective in a limited data context, more data helps to achieve significantly better results.

With intent analysis, while the intended action can be inferred from the text, it may often require some contextual knowledge. The real-world applications of intent analysis are very much challenged by the scarcity of labelled data, hindering its successful application. The above experiments show that significant improvements in prediction accuracy can be achieved by transferring knowledge from the pre-trained language models to the intent analysis model. The pre-trained language model helps on two fronts–it allows the intent analysis model to understand the natural language efficiently and provides relevant knowledge learned from an extensive collection of unlabelled data that can be effectively used when the target task lacks sufficient labelled training data to identify important pat-

terns. This research shows that using a pre-trained language modelling tool and a text data augmentation technique can improve noisy text classification in a low labelled data regime.

3.4.2 Experiments on Noisy Text Comprehension

3.4.2.1 Dataset Development

As discussed in Section 3.3.2, we developed five different probing datasets for these different probing tasks. The probing datasets are developed based on the 'Tweebank v2' dataset[2] developed by [160]. 'Tweebank v2', a collection of English Tweets annotated in Universal Dependencies [157], is useful since it can be exploited for training NLP systems to enhance their performance on social media texts. Table 3.5 shows the distribution of Tweets for training, validation and testing across the five probing datasets.

Table 3.5: Probing task datasets.

Dataset	Train	Validation	Test
SentLen	1530	684	1141
WC	439	186	328
BShift	1639	710	1201
TreeDepth	1071	433	757
SOMO	1003	444	727

3.4.2.2 Sentence Embedding Generation

As shown in Table 3.6, we leverage a few commonly used pre-trained language models and the Sentence-BERT embeddings model under each of the base language models discussed in Section 3.3.1. For training, standard sentences from the Google News dataset and Wikipedia was used for 'GoogleNews' and the 'glove_6b' pre-trained models while $\text{BERT}_{\text{BASE}}$ model was trained using BookCorpus and Wikipedia data. Similarly, the SBERT-NLI-base sentence Transformer was trained on the SNLI [161] dataset, whereas the 'glove_twitter' language model was trained with a large number of Tweets.

[2] https://github.com/Oneplus/Tweebank

Table 3.6: Pre-trained language models.

Base Model	Pre-Trained Model	Vocab. Size	Dimension
Word2Vec	GoogleNews	3M	300
GloVe	glove_6b	400K	300
GloVe	glove_twitter	1.2M	200
BERT	BERT-base	30522	768
S-BERT	SBERT-NLI-base	30522	768

3.4.2.3 Probing Task Classification

We used SentEval toolkit [162] to evaluate different sentence encoders. As in [163], we employed MLP, a deeper network, and a LR classifier to make the findings more practical while reducing the undesirable side effects, such as preference for embeddings of larger size. We used the classifier and the validator provided with the SentEval toolkit[3][162] after modifying it to accommodate proposed sentence embeddings. Following Conneau *et al*. [162], we applied the parameters, shown in Table 3.7, for LR and MLP. However, to cope with the computational constraints, we have modified the value of the 'batch_size' parameter to 32.

Table 3.7: Parameters for the classifiers.

Parameter	LR	MLP
nhid	0	200
optimiser	'adam'	'adam'
batch_size	32	32
tenacity	5	5
epoch_size	4	4

3.4.2.4 Results

This section first analyses the effectiveness of the proposed pooling strategies–layer pooling and token pooling. Next, we analyse the distribution of the language understanding (i.e., surface, syntactic and semantic) across the various regions of the BERT model proposed for this study. Finally, we analyse the performance of the sentence vectors generated by combining these findings along with the existing sentence vector generation mechanisms, including the state-of-the-art techniques.

[3]https://github.com/facebookresearch/SentEval/

Table 3.8: Sentence vector sizes derived using different pooling strategies with four hidden layers.

Layer Pooling	Token Pooling	Embedding Size
concat	max	3072
	mean	3072
	mean_max	6144
mean	max	768
	mean	768
	mean_max	1536
sum	max	768
	mean	768
	mean_max	1536

Pooling strategy analysis. Since the CONCAT-layer-strategy and MEAN-MAX-token-strategy significantly increase the resulting sentence vector size, by four times and two times, respectively, we considered sentence embeddings derived by using all four layers of each BERT region. Table 3.8 shows the resulting sentence vector sizes for each combination of layer and token pooling strategies when applied for four hidden layers of BERT. From the results shown in Table 3.9, we note that the LR model achieves the best results with sentence vectors of size 6,144, whereas the MLP model achieved the best results, in most cases, with 1,536 vector size. From this, it becomes evident that simpler models, such as LR, require huge sentence vectors to identify linguistic patterns, while the complex models can achieve improved results with significantly lower-sized sentence vectors.

Similarly, Table 3.10 shows that the LR model achieved–in most cases–the best accuracy with the CONCAT-layer-strategy. However, one of the syntactic information groups' tasks and the semantic information task obtained the best results with MEAN-layer-strategy. Conversely, the MLP model performed satisfactorily with the MEAN-layer-strategy and SUM-layer-strategy. Both LR and the MLP models prefer MEAN-MAX-token-strategy or MEAN-token-strategy, while MAX-token-strategy performed poorly across all the probing tasks.

In the rest of the analyses, the results derived with MEAN-layer-strategy and MEAN-token-strategy using the MLP classifier are used. This enables easy comparisons of the BERT based sentence embeddings with vectors derived from static pre-trained models by calculating the average of word embeddings. Further, Sentence-BERT internally uses the mean of the token embeddings to generate sentence embeddings.

Table 3.9: Average classification accuracy for different sentence vector sizes derived with four hidden layers using different pooling strategies.

Vector Size	Region	LR					MLP				
		SL	WC	BS	TD	SM	SL	WC	BS	TD	SM
768	1	53.29	70.96	67.36	51.29	68.95	61.11	73.48	68.13	55.65	69.26
	2	52.54	58.69	71.47	51.75	70.29	59.91	60.98	71.38	57.17	70.94
	3	47.40	46.11	71.13	50.93	70.77	55.37	49.47	71.75	55.58	73.14
1536	1	56.75	80.18	69.45	55.42	70.15	**68.41**	84.15	68.99	56.61	70.09
	2	56.18	70.58	73.77	53.57	72.70	65.65	71.19	**74.19**	**58.45**	71.73
	3	48.95	54.27	70.15	51.12	**72.97**	58.07	56.56	71.44	56.28	**74.69**
3072	1	54.74	73.48	69.28	51.79	68.23	63.19	74.70	69.15	56.54	59.22
	2	53.42	61.74	72.53	53.10	70.77	61.49	65.40	72.40	58.39	70.70
	3	48.42	50.31	72.36	51.92	72.56	58.33	52.90	71.44	56.01	72.28
6144	1	**59.33**	**82.62**	69.61	**56.01**	70.29	65.03	**86.59**	71.61	56.94	72.35
	2	56.70	69.21	**75.60**	53.50	72.35	64.59	72.26	72.69	58.39	72.21
	3	51.45	55.79	71.61	51.65	72.90	61.35	56.71	72.61	58.39	71.94

Table 3.10: Mean classification accuracy for sentence vectors derived using different pooling strategies (SL:SentLen, WC:WC, BS:BShift, TD:TreeDepth, SM:SOMO).

Pooling		LR					MLP				
Layer	**Token**	**SL**	**WC**	**BS**	**TD**	**SM**	**SL**	**WC**	**BS**	**TD**	**SM**
concat	max	50.51	55.99	69.50	52.53	69.33	57.35	58.03	69.38	57.11	63.00
	mean	53.87	67.68	**73.27**	52.00	71.71	64.65	70.63	72.61	56.85	71.80
	mean_max	**55.83**	**69.21**	72.27	53.72	71.85	63.66	**71.85**	72.30	57.91	72.17
mean	max	48.85	51.32	67.97	51.78	69.42	56.77	55.08	67.89	56.50	71.43
	mean	54.28	66.67	72.72	52.62	**72.81**	63.02	69.41	72.02	**58.96**	71.34
	mean_max	54.63	68.50	71.77	**54.07**	72.26	63.02	70.33	71.74	56.80	**72.31**
sum	max	47.65	50.00	66.81	49.54	65.75	53.87	50.20	68.83	55.57	70.43
	mean	53.52	66.36	72.44	51.34	72.03	61.53	70.53	**72.94**	53.50	71.25
	mean_max	53.28	68.19	70.47	52.66	71.62	**65.06**	70.94	71.33	57.42	72.03

Region-wise analysis. Figure 3.5 shows a heat map of the accuracies (darker colours equate to higher accuracy) of each probing task with sentence vectors generated using each hidden layer of the BERT model. The SentLen and the WC tasks in the Surface Information group achieved better accuracies with sentence vectors derived from hidden layers in the first region (R1), and the performance gradually decreases as we move towards the last layers of the BERT model. Conversely, higher accuracies were obtained for the Syntactic information tasks–BShift and TreeDepth–with the sentence vectors generated using the hidden layers from the second region (R2). The initial layers of the R2 have shown the most contribution to the accuracy, while the hidden layers from the R1 have contributed poorly to the Syntactic information group tasks. Further, the hidden layers that contribute to increasing the sentence vectors' richness for the Semantic information task were found at the border of the R2 and R3.

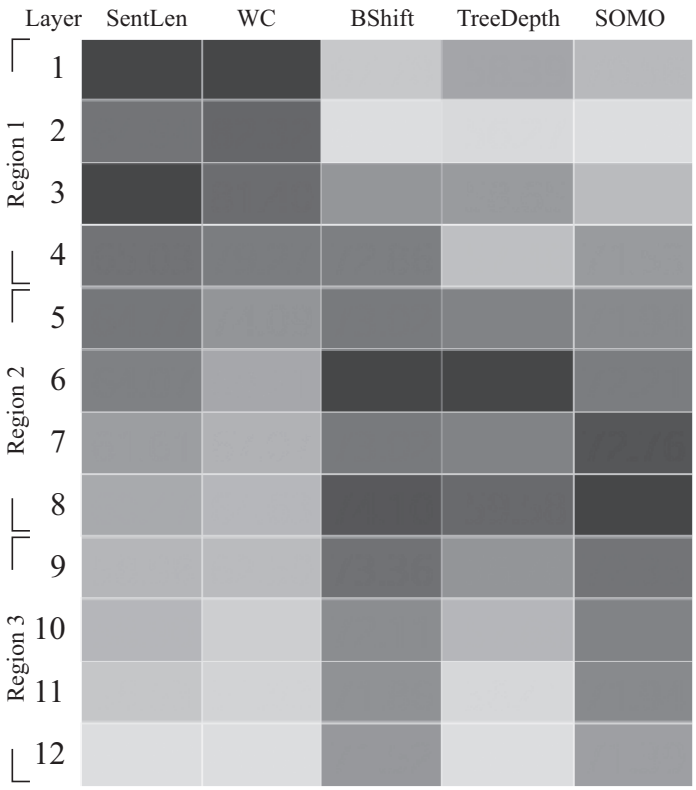

Figure 3.5: Heat map of probing task accuracy.

Overall, in the context of noisy texts, the hidden layers in the region R1 contain most of the linguistic characteristics required to address probing tasks in the Surface group. In contrast, Syntactic and Semantic group tasks were able to identify necessary linguistic patterns

from R1 and R2. Nevertheless, the sentence vectors' performance derived from hidden layers in the last region (R3) ranges from low to marginal, indicating their inability to capture linguistic information from noisy texts.

Overall accuracy. Table 3.11 presents the classification accuracies for probing tasks with sentence vectors derived from GloVe-based pre-trained models, Sentence-BERT and using different hidden layers from $BERT_{BASE}$-uncased model. In the context of BERT-based sentence vectors, we have considered sentence vectors derived based on the last hidden layer, the last four hidden layers, and all 12 layers. Devlin *et al.* [15] achieved comparable results for the feature-based approach by using those layers as the input to an artificial recurrent neural network. Based on our findings, we propose two separate approaches for noisy texts. The first is based on BERT's first hidden layer, while the second combines the first hidden layer of each BERT region–layers 1, 5 and 9 (1-5-9).

The MLP model has achieved the best accuracy for all the probing tasks except for the SOMO task, which is in the Semantic information group. The LR model has reached the Surface information probing tasks' best results with the BERT-based sentence vectors derived only using the first hidden layer. However, LR performed better for Syntactic and Semantic information probing tasks with sentence vectors generated using all 12 hidden layers of the BERT model.

Conversely, the best results for the MLP model were mostly achieved with the sentence vectors derived using the 1-5-9 hidden layers. Only the Semantic information task achieved the best accuracy with all 12 hidden layers. The WC probing task performed well with the first hidden layer, and the second-best accuracy was obtained with the 1-5-9 hidden layers.

3.4.2.5 Discussion

The BERT sentence vectors performed exceptionally well on all the probing tasks and performed better than GloVe and Word2Vec, despite these two representations having rich vocabulary. Specifically, the GloVe model, even though trained on a large corpus of Tweets, performed poorly. Further, sentence vectors derived from BERT's hidden layers achieved significantly better results over the state-of-the-art Sentence-BERT model. The latter hid-

Table 3.11: Classification accuracies for different sentence vectors (SL:SentLen, WC:WC, BS:BShift, TD:TreeDepth, SM:SOMO).

Model	Name	LR					MLP				
		SL	WC	BS	TD	SM	SL	WC	BS	TD	SM
S-BERT	None	39.53	29.57	66.53	49.27	68.36	44.79	29.27	67.36	49.27	65.47
BERT	CLS	33.83	25	65.95	40.29	65.06	38.39	25.61	66.28	48.22	65.2
	Last layer	45.75	44.51	70.27	53.24	71.66	51.8	46.65	71.52	56.01	71.39
	Last four	50.57	52.44	72.77	51.92	73.59	56.79	55.79	72.94	56.54	72.9
	All	55.57	67.68	**73.61**	**54.69**	**74.83**	66.87	71.04	73.77	59.18	**73.18**
	First layer	**58.63**	**82.01**	67.94	52.44	70.01	66.96	**85.67**	67.78	58.39	70.56
	1-5-9	56.35	70.12	72.52	54.43	72.9	**68.8**	74.09	**73.94**	**59.84**	71.53
Glove	glove_6b	22.7	64.02	59.87	38.57	58.18	24.45	64.02	59.2	38.44	57.08
	glove_twitter	26.38	66.77	61.53	40.29	64.65	25.42	69.82	61.37	42.01	63.96
Word2Vec	word2vec_neg	25.24	62.2	61.62	36.2	67.4	26.47	70.43	62.7	40.16	66.16

den layers of BERT performed poorly in capturing linguistic information compared to the shallow layers. The unstructured nature of the Tweets, we observe, benefit more from the initial layers that capture shallow information rather than the later layers, which capture more complex hidden information.

Further, the experiments relating to the length of the sentence vector revealed that the simpler predictive models perform better with large sentence vectors, while complex models prefer significantly smaller vectors. The complex models are better at identifying intricate patterns from compressed vectors that contain rich information. However, simpler models prefer higher dimensions to achieve better results as they cannot identify complex patterns.

We presented a methodology to systematically analyse the knowledge distribution within a multilayer pre-trained language model while generating sentence vectors that can capture various linguistic characteristics. This technique could be directly applied to most multilayer pre-trained language models to understand the linguistic properties captured by latent representations. Further, the noisy probing datasets developed in this study can complement future research in NLU by providing researchers with additional datasets that cover the domain of noisy data. While the current study focused on a representative set of five probing tasks that covered all linguistic attributes, future work could also explore additional probing tasks to further validate our results.

Future research could focus on analysing the impact of pre-processing the Tweets to reduce the noise level on the linguistic knowledge distribution and the derived sentence representations. Moreover, the same probing dataset could be used to examine the relationship between the BERT's attention layers and the meaning-rich sentence embeddings. This could help to derive more meaning-rich sentence vectors.

The research work reported in Section 3.4.2 demonstrates that the general language understanding of pre-trained language models such as BERT can be effectively exploited to comprehend noisy texts. Further, the proposed methodology can effectively generate sentence vectors encoding different linguistic aspects using latent representations of multilayer pre-trained language models. We observe that the shallow layers of the BERT

model are better at capturing linguistic information of noisy and unstructured texts than the deeper layers for general English sentences [99]. Further, it can be noted that simple predictive models prefer large sentence vectors, while complex models are more successful with significantly smaller sentence vectors. It is worthwhile noting that the first layer or a combination of BERT layers from each region can be used to derive generalisable sentence vectors for noisy and unstructured texts.

We believe that our new noisy probing datasets can serve as benchmark datasets for future researchers to study the linguistic characteristics of unstructured and noisy texts.

3.5 Summary

Our research work reported in this chapter shows that the pre-trained language can effectively transfer prior knowledge to noisy text classification tasks to achieve state-of-the-art accuracy under a low labelled-data regime. Further, the meaning-rich sentence representations derived from latent layers of a pre-trained language model proved to be highly effective in capturing the linguistic characteristics of a sentence.

However, although transferring prior knowledge from pre-trained language models helps to improve the performance of low-resource text classification tasks, results also confirm that the text augmentation technique helps boost the performance further. Hence, in the next chapter, we focus on developing a novel text augmentation technique to address the root cause of data scarcity by generating quality synthetic sentences to augment the training dataset.

Chapter 4

Meaning-Sensitive Text Augmentation

4.1 Introduction

In the previous chapter, though we boosted the performance of NLP models through transfer learning and text comprehension, it was evident that the data scarcity significantly affected the accuracy of large-scale NLP models, and data augmentation helped alleviate this problem. Most importantly, data scarcity is a common problem in many real-world applications, including for organisations such as SportsHosts. Thus, it will be challenging to achieve the desired results through NLP-powered solutions without an effective mechanism, such as data augmentation, to address this limitation.

'Data augmentation' is a term used to describe ways to generate various training examples without gathering new data [164]. In computer vision, data augmentation can be easily performed by transformations, such as cropping, rotating, resizing, mirroring and colour shifting, which generate new training samples without losing important information. In contrast, for text-based data, data augmentation is challenging, as universal transformation methods generate random text sequences that tend to lose valuable information about the syntax and semantics of a sentence. Even a minor revision to a sentence could change the overall meaning of the sentence and disturb its grammatical structure, thus, having a significant negative impact on the accuracy of downstream NLP tasks.

In recent times, due to the rapid growth of rich text data (e.g., social media data, re-

views and emails), a plethora of NLP use cases have captured the attention of both the industry and academia. However, the accuracy and scalability of these machine learning approaches are often challenged by two key factors, the scarcity of labelled data and imbalances in the data. Further, due to the sheer volume of text data generated every day and passed through trained models for inference, models are increasingly exposed to unseen data, significantly affecting the model performance. Conversely, with the recent popularity of deep neural network-based models in NLP tasks, attention to text augmentation has increased, as the limited size of training data tends to significantly affect the accuracy of large models due to overfitting. While the state-of-the-art pre-trained language models, such as BERT, trained via MLM, benefit many downstream NLP tasks, the performance of such models can be greatly improved with more training data [143].

The rest of the chapter is organised as follows. Section 4.2 proposes a semantic text augmentation technique based on the back-translation approach. Further, it explores the impact of adding increasingly more synthetic sentences to the accuracy of the downstream NLP tasks. Section 4.3 presents a novel MLM-based text augmentation algorithm, IMOSA, which can significantly optimise the overall quality and diversity of the augmented dataset. Further, we conduct an extensive experiment to ascertain the robustness of IMOSA against two state-of-the-art text augmentation techniques across multiple NLP tasks.

4.2 Semantic Data Augmentation

In this section, we present a novel semantic data augmentation method that generates synthetic sentences to mitigate the impact of small, labelled data problems while preserving the meaning of the synthetic sentence and label compatibility.

In the proposed method, we generate additional (i.e., synthetic) data via the transformation of a specific Tweet. The available sentences are augmented without violating their meaning by applying the back-translation strategy (translating from English to any other language and then back to English) [139], which generates new semantically appropriate sentences that preserve the meaning of the original sentence–thereby synthesising more

data. While generating synthetic sentences, it is necessary to ensure that the synthetic sentence preserves the semantic similarity with the original sentence to reduce the risks of introducing a label noise.

A schematic diagram illustrating the proposed semantic data augmentation method is shown in Figure 4.1. The key components of this semantic data augmentation architecture are back translation, sentence embedding and similarity threshold, as discussed next.

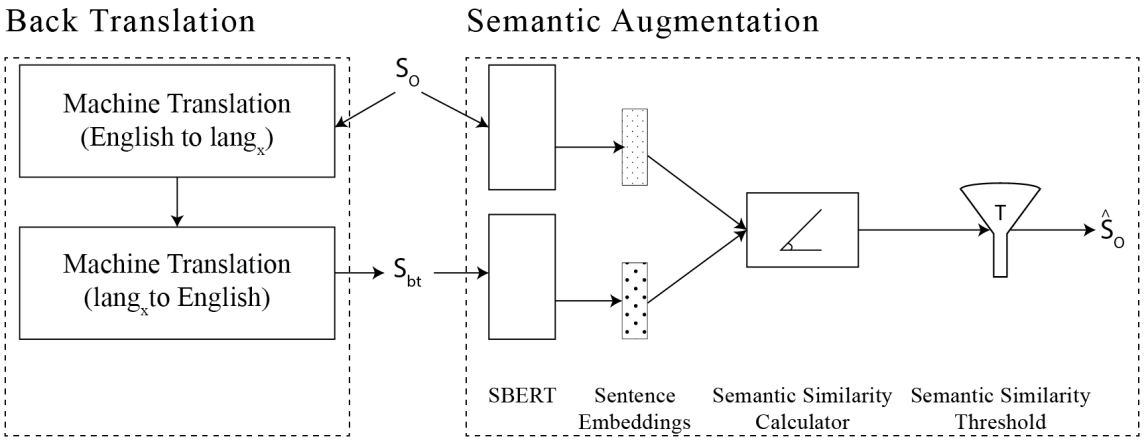

Figure 4.1: Semantic data augmentation using back translation and sentence similarity filtering based on sentence embeddings extracted from the Sentence-BERT (SBERT) [152] model.

Back translation. A new synthetic sentence is obtained by applying back translation, which translates a Tweet in English into any target language, $lang_x$, and then re-translates it back into English. The chosen target languages are those that belong to the Indo-European language family (i.e., same language family as English.) so that multiple target languages can be used effectively. For this research, the three target languages chosen are German, French and Italian.

Sentence embedding. While the diversity of words and sentences is essential, the STS of the sentences is also crucial to underpinning performance improvement, primarily if the meaning of the sentence exists in the downstream segment. The proposed method leverages STS to reduce the distortion or changes in the meaning of the original sentence. For this, we use SBERT, an existing built-in algorithm with the pre-trained BERT network, which uses the Siamese and triplet network structures to derive semantically meaningful

sentence embedding [152].

Similarity threshold. Let

S_o be the source sentence

S_{bt} be the synthetic sentence generated using a second language.

For data augmentation, we propose to use only sentences that are semantically meaningful by comparing the sentence embedding of the original sentence ($\vec{S_o}$) with the transformed example $\vec{S_{bt}}$. S_{bt} is considered a valid sentence ($\hat{S_o}$) if and only if $\cos(\vec{S_o}, \vec{S_{bt}}) \geq T$, where T is a chosen threshold value for semantic similarity.

To determine the similarity threshold level, we propose a novel method based on the probability density function of the cosine-similarity scores between S_{bt}, obtained using all the target languages and corresponding original sentence S_o, as depicted in Figure 4.3. The method uses pth percentile (π_p) to determine a set of candidate threshold values $\{T_{min}, T_p\}$, where $p \in 15, 25, 50$. T_p is calculated using Equation (4.1), whereas the minimum threshold value T_{min}, based on the standard interquartile range (IQR) rule (i.e., 1.5 $\times$ IQR rule), is calculated using Equation (4.2).

$$T_p = \int_{-\infty}^{\pi_p} f(x)\, dx \tag{4.1}$$

$$T_{min} = T_{25} - 1.5(T_{75} - T_{25}) \tag{4.2}$$

Since the augmentation dataset comprises transformed sentences using multiple languages, a question then arises as to whether the different threshold values per target language might be more effective in identifying semantically meaningful sentences. This approach is cumbersome and may be costly when many target languages are used. However, the proposed method is meaningful only if the behaviour of the target languages are approximately similar to each other.

4.2.1 Augmentation Methodology

Based on the above considerations in text data augmentation, the following systematic and novel step-by-step methodology is proposed for applying the linguistic model BERT for intent classification with limited data availability.

1. Translate a source sentence (S_o) into a second language and then back to English. Multiple target languages can be used to generate multiple synthetic sentences from a given S_o. Let S_{bt} be a synthetic sentence generated and A_i be a set of S_{bt} generated by each second language i for a given set of source sentences.

2. Obtain deep contextualised word representation vector S_o^e for each source sentence and S_{bt}^e for corresponding back-translated sentences from A_i using SBERT, a modification of the pre-trained BERT network that uses Siamese and triplet network structures to derive semantically meaningful sentences.

3. Compute the cosine similarity C $(-1 <= C <= 1)$ between the sentence embedding pair (S_o, S_{bt}) for all the sentences in A_i. Steps 4–8 below determine the semantic similarity threshold.

4. Let us propose a hypothesis that the two probability density distributions being compared are equal. Verify the equality of the probability density functions of the semantic similarity scores of synthetic sentences in each A_i against each other and with reference to the probability density functions of the semantic similarity scores of all the back-translated sentences A' based on Equation (4.3).

$$A' = \bigcup_{i=1}^{n} A_i \tag{4.3}$$

5. The null and alternative hypothesis for the Kolmogorov–Smirnov (KS) Test can be formally stated as

$$H_0 : f(C) = f_0(C) \text{ for all } C$$

$$H_1 : f(C) \neq f_0(C) \text{ for at least one } C.$$

Apply the two-sample KS test [165], as it is sensitive to deviations in both location and shape of the empirical cumulative distribution functions of the two samples.

6. As identifying any difference between the probability density distributions of the mixture of all back-translated sentences A' and back-translated sentences generated by a target language A_i is complex, a global threshold value for all the target languages are determined based on the probability density distribution of all back-translated sentences, as depicted in Figure 4.3, without applying individual threshold values for each second language.

7. If there is not enough evidence unavailable to identify any difference between the probability density distributions of the mixture of all back-translated sentences A' and back-translated sentences generated by a target language A_i, we accept H_0. In this case, we determine a global threshold value for all the target languages based on the probability density distribution of all back-translated sentences, as depicted in Figure 4.3, without applying individual threshold values for each second language.

8. Apply a threshold, T, on C to retain only the back-translated sentences semantically close to the source sentence. T is a hyperparameter of the proposed semantic data augmentation model.

9. Finally, fine-tune the extended BERT model using the augmented dataset with the optimal values for task-specific properties of the BERT model (i.e., batch size, learning rate and the number of epochs) and the semantic similarity threshold T, identified during hyperparameter tuning.

4.2.2 Experiments on Semantic Data Augmentation

This section describes the experiments carried out to study the proposed semantic data augmentation technique to overcome the problem of labelled data scarcity and evaluate the effectiveness of pre-trained language models in intent classification, which is a challenging NLP task. We use the same intent related dataset developed by [33] that we studied

earlier in Section 3.4.1 of Chapter 3. Table 3.3 shows the intent categories and distribution of the records across different intent categories.

We extend the same experiment discussed in Section 3.4.1.3 of Chapter 3 to fine-tune the BERT model with a fresh augmented dataset created using the semantic data augmentation technique proposed in Section 4.2 of this chapter. We discuss the extension of the experiment in detail in the following section.

4.2.2.1 Ablation Study

To augment the text data using back translation, we chose three target languages: German, French and Italian. We then apply Google Translate API ('googletrans') to translate the 10 randomly selected instances of each category from D_T. After removing the synthetic sentences that are exactly similar to the original sentence, the augmented dataset D_A is obtained. With this approach, for each training dataset sample, we generated three augmented datasets using the target languages. Let each augmented dataset generated using different target languages be denoted as D_A^i, where $i \in 1, 2, 3$,

Sentence-BERT[1] is applied to generate sentence embeddings for original S_o and synthetic sentences S_{bt} in D_A^i. To evaluate the semantic similarity between S_o and S_{bt}, we chose the cosine-similarity score (cosine of the angle between two embedding vectors), a widely implemented metric in information retrieval. The probability density distributions of the semantic similarity scores for back-translated sentences in each D_A^i and for the mixture of all the back-translated sentences are shown in Figure 4.2.

The hypothesis tests between the empirical distribution function of the mixture of all back-translated sentences and the distribution of the back-translated sentence generated using each of the target languages revealed very high p-values for each KS test. This indicates weak evidence against the null hypothesis, thereby failing to reject the null hypothesis. Therefore, a global threshold value is applied instead of all the back-translated sentences generated using different target languages, as shown in Figure 4.3.

For fine-tuning, with the augmented dataset D_A^i, we follow the same BERT model ar-

[1]https://github.com/UKPLab/sentence-transformers

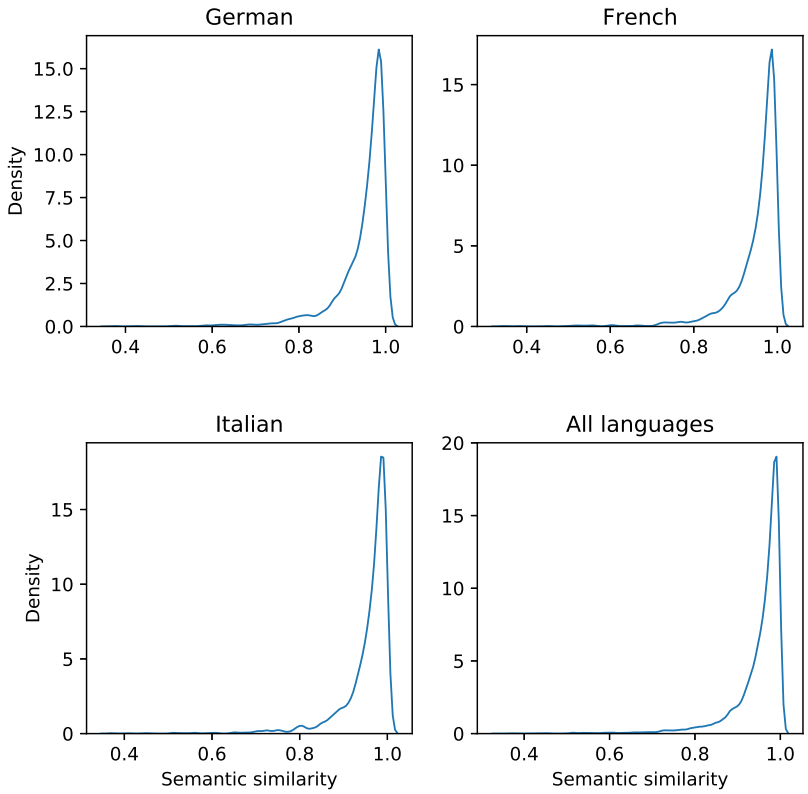

Figure 4.2: Probability density distributions of semantic similarity scores of synthetic sentences.

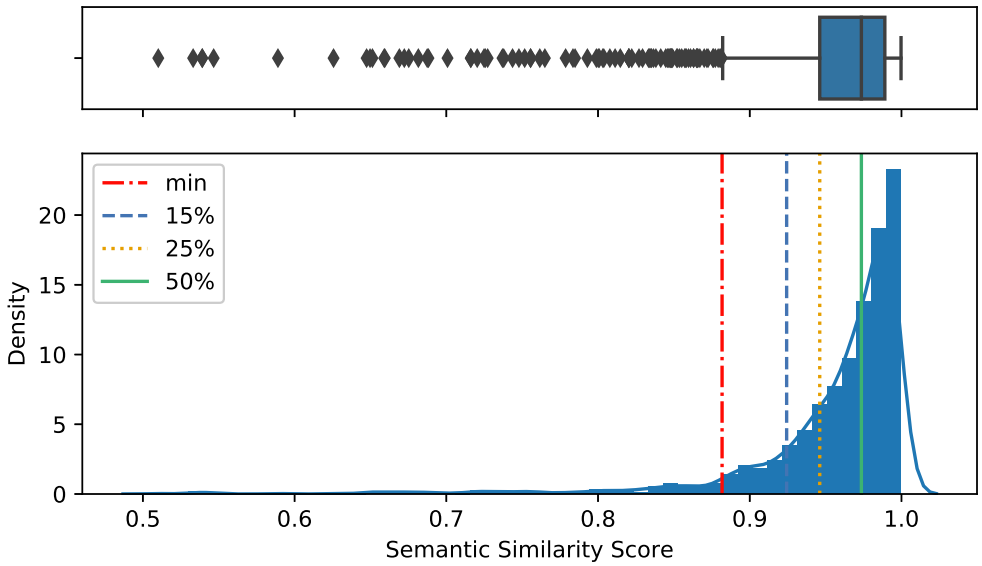

Figure 4.3: Probability density distribution of semantic similarity of the mixture of all the back-translated sentences.

chitecture, as depicted in Figure 3.1. We perform hyperparameter tuning with five-fold cross-validation using D_A^i to obtain the optimal batch size, learning rate, number of training epochs and, additionally, the semantic similarity threshold of the proposed data augmentation technique. As depicted in Figure 4.4, we obtained the best model performance on the training dataset for all D_A^i when the threshold value was set to T_{min}. For each D_A^i, the test accuracies were obtained using ten-fold cross-validation with 10 randomly sampled instances from D_T as training data, with D_V being used as test data.

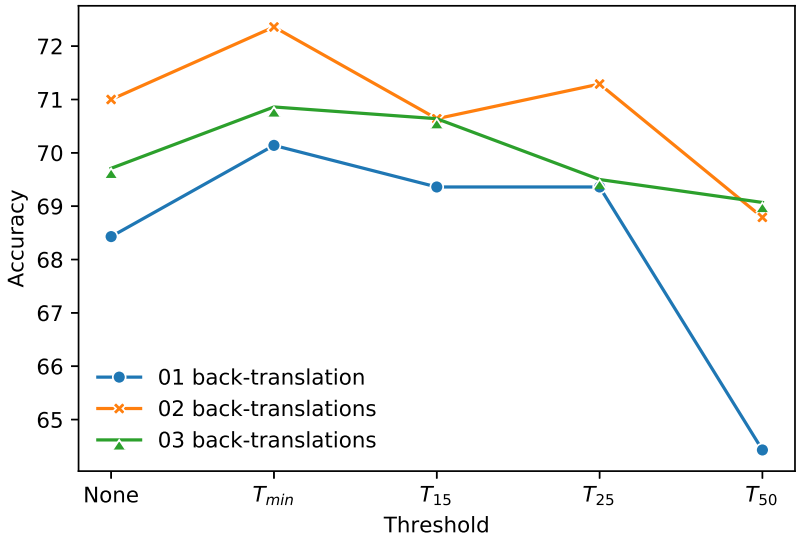

Figure 4.4: Evaluation accuracy.

To evaluate the effectiveness of our approach, we conducted several experiments. Table 4.1 presents the accuracies obtained by different strategies. Experiment 1 gives the baseline accuracies from Wang *et al.* [33]. Experiment 2 shows the performance of intent classification using the model based on BERT, as depicted in Figure 3.1, with the same dataset and a similar training set-up as [33]. Further, Experiments 3, 5 and 7 report the performance of our models, which were fine-tuned with D_A^i ($i = 1, 2, 3$, respectively), without applying any semantic similarity threshold T_p. However, for Experiments 4, 6 and 8, we apply the optimal semantic similarity threshold obtained during hyperparameter tuning to eliminate noisy synthetic sentences.

As shown in Table 4.1, we achieved 1.41%, 0.68% and 0.55% average accuracy (Micro-F1) improvement for Experiments 4, 6 and 8, respectively, which signifies the effectiveness

Table 4.1: The F1 results of individual categories, and Micro-F1 and Macro-F1 over the seven categories.

Expt.#	Model	No. of Back-Tran.	Threshold	Career	Event	Food	Goods	Non-Intent	Travel	Trifle	Micro-F1	Macro-F1
1	Wang's	–	–	45.73	27.13	54.63	43.25	35.56	58.64	20.04	42.21	40.71
2	BERT (fine-tuning)	–	–	42.62	48.80	66.79	47.21	39.32	56.85	41.98	50.75	49.08
3	BERT+back-tr.	1	None	66.99	60.94	83.46	60.26	40.65	71.40	49.36	58.69	61.87
4	BERT+Sem. Aug.	1	T_{min}	65.76	61.86	84.20	61.59	48.74	75.20	45.59	60.14	63.28
5	BERT+Sem. Aug.	2	None	68.48	62.00	83.21	58.76	50.05	77.98	46.20	60.42	63.81
6	BERT+Sem. Aug.	2	T_{min}	68.30	64.45	84.54	60.48	53.88	74.57	45.22	**61.37**	**64.49**
7	BERT+Sem. Aug.	3	None	68.50	61.57	84.28	60.27	47.47	75.07	46.53	59.81	63.38
8	BERT+Sem. Aug.	3	T_{min}	69.03	63.33	84.17	59.50	50.28	74.38	46.82	60.65	63.93

Note: The Macro-F1 weights all the categories equally, whereas the Micro-F1 weights individual Tweets equally favouring the performance of the large class.

of the proposed semantic similarity threshold. This threshold, which controls the amount of noise removed from the synthetic dataset, contributes to the improvements. We observe that the highest accuracy with the proposed semantic data augmentation technique is obtained with two back translations, with a similarity score threshold value of 0.8797 (T_{min}). Figure 4.5 shows the performance of our approach with different threshold values for each D_A^i. As shown in Figure 4.4, the model trained with a semantically augmented training dataset outperforms the model trained with the full augmented dataset in terms of average test accuracy. Interestingly, we observe an overall drop in accuracy when there is an increase in the number of back translations from two to three. This is possibly because the synthetic sentences are not providing any further diversity and variety to the training data, despite adding additional target languages, thereby resulting in the model overfitting the training data.

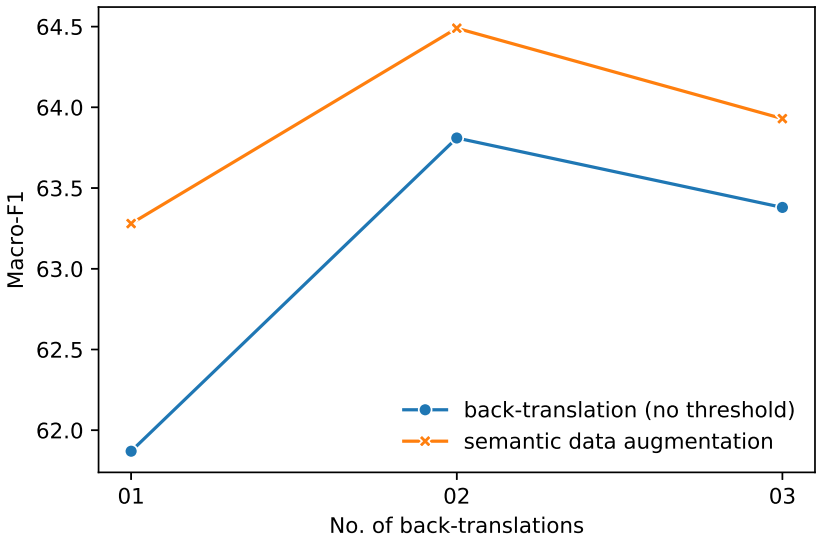

Figure 4.5: Average test accuracy.

4.2.2.2 Discussion

Examples (a)–(d) in Table 4.2, for back-translated sentences with corresponding semantic similarity scores, proclaim that the proposed approach is more effective compared to the naive back translation. Interestingly, as observed in examples (a) and (b) in Table 4.2, our approach was able to easily eliminate the meaningless back-translated sentences gener-

Table 4.2: Examples of synthetic sentences and semantic similarity scores.

(a)	Original sentence	I need ice cream to put out fire -.-	
	Back-tr. (German)	I need ice fire extinguished -.-	0.72
	Back-tr. (French)	I need ice cream to extinguish the fire -.-	0.96
(b)	Original sentence	I need to hit up the mall .	
	Back-tr. (German)	I need the Mall beating.	0.83
	Back-tr. (Italian)	I have to hit the mall.	0.94
(c)	Original sentence	I want to buy i-phone .	
	Back-tr. (German)	I love shopping i-phone.	0.68
(d)	Original sentence.	I want chinese buffet for lunch .	
	Back-tr. (German)	I like Chinese buffet for lunch.	0.90
(e)	Original sentence	I should really get some sleeeep !	
	Back-tr. (French)	I should really sleep!	0.55
	Back-tr. (Italian)	I really should get some sleep '!	0.56
(f)	Original sentence	I would like to get a type writer ...	
	Back-tr. (German)	I want to get a typewriter ...	0.79
	Back-tr. (Italian)	I would like a typewriter ...	0.79
(g)	Original sentence	I should slerp with you . Hmm	
	Back-tr. (German)	I want to sleep with you. Hmm	0.37

ated with German as the target language while continuing to retain the meaningful and diverse synthetic sentences generated using French and Italian languages. Further, as observed in examples (c) and (d), the translations are acceptable in a general context. However, these synthesised sentences express an opinion rather than an intent [28]. The proposed semantic data augmentation technique maintains label compatibility in such situations.

In contrast, examples (e)–(g) in Table 4.2 show valid synthetic sentences, but these had low semantic similarity scores due to repeated sequential letters (e.g., sleeeep), incorrect word separations (e.g., type writer) and spelling mistakes (e.g., slerp) respectively. We may minimise the impact caused by such aberrations by introducing pre-processing tasks, such as spelling correction and removal of additional letters in a word with repeated sequential letters.

4.3 Meaning-Sensitive Data Augmentation with Intelligent Masking

The work proposed in this section focuses on generating synthetic sentences to improve the performance of context-aware NLP tasks while maintaining a good balance between both the overall quality and diversity of the augmented dataset.

With recent advances in NLP technology, researchers have started paying more attention to improving the quality of data augmentation algorithms via context-aware techniques [125]. To this end, researchers have recently studied MLM-based approaches [126, 166, 127] using pre-trained language models to generate context-aware and label-compatible synthetic sentences. These methods generate a fixed number of synthetic sentences for each example in the original dataset. While MLM-based techniques tend to improve the quality of the generated sentences when compared to other augmentation techniques, such as EDA [115], augmenting every single sentence might introduce low-quality sentences to the augmented dataset, affecting the overall accuracy of downstream NLP tasks, especially in low-resource settings. Conversely, the performance improvements afforded by augmented data have only been evaluated using architectures based on recurrent or convolutional layers, which pay less attention to the context of the words than state-of-the-art Transformer-based models [54, 96], thereby being less sensitive to meaning and grammatical structure deviations of the synthetic sentences.

Synthetic sentences that are too similar to the original sentences may lead to greater overfitting, while sentences that are too different may lead to poor performance. While data augmentation techniques such as EDA [115] produce synthetic sentences without paying much attention to the meaning, Conditional BERT Contextual Augmentation (CBERT) [126] and Self-Supervised Manifold Based Data Augmentation (SSMBA) [127] focus on generating more meaningful sentences. However, both CBERT and SSMBA do not concentrate on selecting the most favourable sentences or words to optimise the overall quality and diversity of the augmented dataset. In this section, we propose IMOSA, a novel text augmentation method that focuses on generating more synthetic data progressively where

plausible rather than augmenting every single sentence in the original dataset. IMOSA includes simple yet powerful techniques to control the quality and diversity specific to a particular dataset. The proposed method consists of novel and intelligent mechanisms to identify the most favourable sentences from the training dataset to ensure that the generated synthetic sentences are of high quality. Further, we focus on augmenting the datasets in the low data regime, where the quality and variety of the synthetic sentences play a more significant role in improving the performance of downstream NLP tasks. We demonstrate the superiority of the proposed text data augmentation technique by evaluating the performance of downstream NLP tasks using a state-of-the-art Transformer-based pre-trained language model.

Briefly, the research reported in this section presents the following novel contributions:

- The novel text data augmentation technique can improve the overall diversity of the original dataset by identifying the most suitable sentences to augment while generating an optimum number of quality synthetic sentences from a selected sentence.
- The novel technique proposed adds additional diversity to the augmented dataset while maintaining a constant percentage of masked input tokens.
- Being generic, the proposed intelligent masking and optimal substitution technique can be applied to improve several existing MLM-based algorithms.
- The proposed IMOSA method performs better than several state-of-the-art techniques, as evidenced through rigorous experiments in a context-aware setting.

4.3.1 IMOSA Methodology

In this section, we describe the components and process of the proposed meaning-sensitive text data augmentation strategy in detail. As depicted in Figure 4.6, at a high level, our data augmentation framework consists of three main steps:

1. conditional MLM pre-training

2. generation of masked candidate sentences using intelligent masking

3. generation of synthetic sentences using optimal substitution.

IMOSA is capable of identifying the most favourable tokens to mask and derive multiple masked inputs for a particular sentence (e.g., example no. 1 has two corresponding masked inputs–1a and 1b) using intelligent masking. Further, optimal substitution allows IMOSA to generate multiple high-quality sentences per masked input (e.g., 1a has two synthetic examples–1a.1 and 1a.2). Apart from that, intelligent masking, together with optimal substitution, helps to eliminate source sentences that might not have the potential to generate quality synthetic sentences (e.g., 1b, 3a and 3b). Moreover, Conditional Fine-Tuning ensures that IMOSA generates label-compatible synthetic examples.

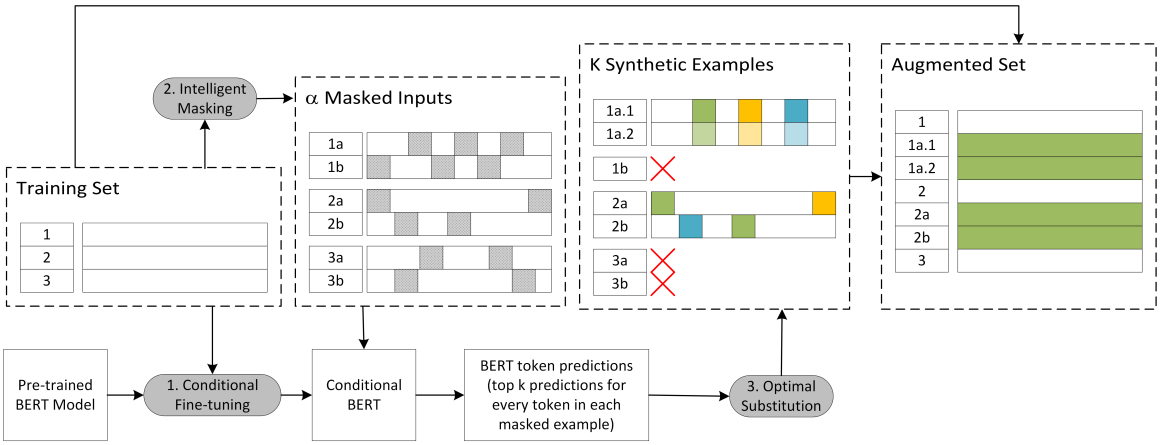

Figure 4.6: High-level steps for generating synthetic sentences using IMOSA.

The proposed methodology primarily uses the BERT model as an MLM for text data augmentation in different stages of the process, as discussed in each section below.

4.3.1.1 Conditional Masked Language Model Pre-Training

The proposed framework leverages the same original architecture of BERT and focuses on fine-tuning BERT for generative tasks to leverage the language model for MLM tasks. Further, to maintain the label compatibility, we fine-tune the BERT model on a particular task using a conditional MLM proposed by [126]. To this end, we modify the input representation and training procedure to perform a conditional MLM task. We blend the label information of a particular text sequence with its input representation and use a labelled

dataset to fine-tune the BERT model to obtain a label-conditional BERT model. To achieve this, as shown in Figure 4.7, we alter the segmentation embedding to label embeddings derived using the annotation label of the sentence. More details on this technique are available in [126].

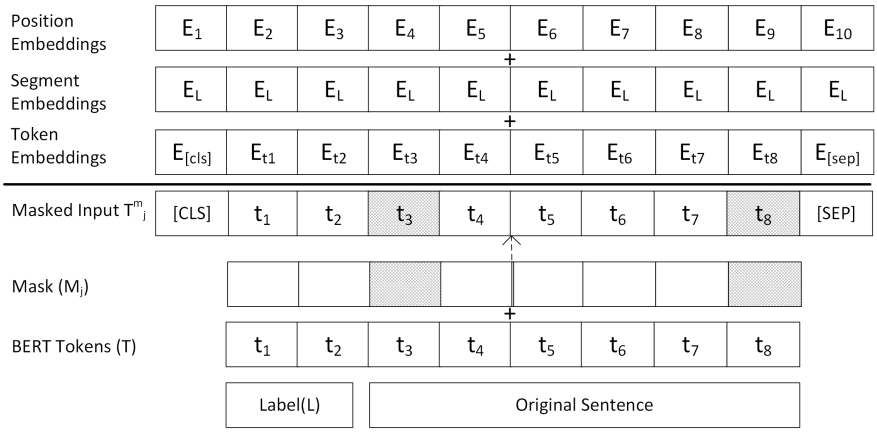

Figure 4.7: Preparation of BERT inputs for masked sentences.

4.3.1.2 Intelligent Masking

The proposed intelligent masking technique is composed of two key steps: Pruned Masking and Masked Multiplier.

Pruned masking. In the pruned masking task, we select candidate tokens to mask a given sentence S effectively. In the context of text data augmentation, replacing words such as proper nouns, pronouns, coordinating conjunctions or determiners (e.g., 'a', 'an' and 'the') in a sentence add little or no diversity to the original dataset. Further, having multiple instances of very similar sentences could cause overfitting. Thus, we select a highly effective set of candidate words or tokens in our approach, eliminating ineffectual words using POS for masking. To optimise the selected candidate tokens, with the help of a stop words list, we further eliminate some common words that would not add required diversity when replaced.

For this, as shown in Figure 4.8, we tokenize S to obtain an array of words $W = [w_0, w_1, \ldots, w_m]$ using a white-space tokenizer. We pass W through POS and a stop-word

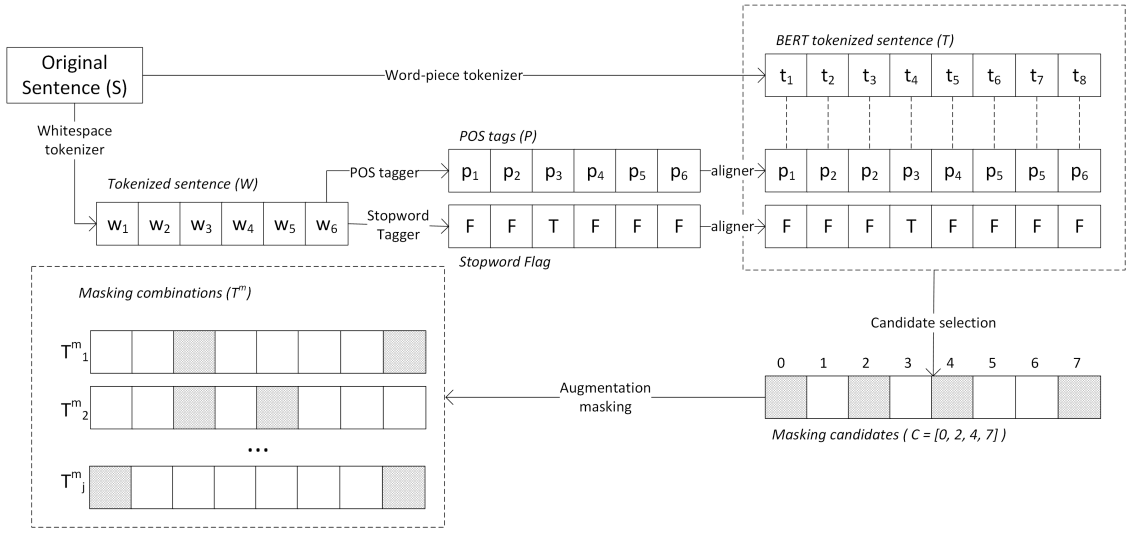

Figure 4.8: Key steps in the intelligent masking phase.

tagger to obtain POS tags $P = [p_0, p_1, \ldots, p_m]$ and stop-word flags $O = [o_0, o_1, \ldots, o_m]$. However, as we use the BERT model, we obtain an array of tokens $T = [t_0, t_1, \ldots, t_n]$ by tokenizing S using BERT's sub-word tokenizer to prepare the inputs. BERT uses a Word-Piece tokenizer, where one word can be broken into multiple tokens producing equal or more tokens (i.e., $|T| >= |W|$) than the white-space tokenizer. Hence, we align P and O with T to map the POS tags and stop-word flags considering sub-words produced by the WordPiece tokenizer. This approach enables us to accurately apply the required tags to each token $t_i \in T$. Next, we select only the tokens tagged with parts of speech (i.e., noun, verb, adjective, adverb, preposition and interjection) as candidate tokens for masking. We disqualify any token flagged as a stop word, optimising the candidate tokens further. For simplicity, we use C to represent these candidate tokens selected for masking. To mask the tokenized sentence for augmentation, we pick a subset $M \in \mathbb{R}^r$ from C, where $r = \min(|T| \times 0.15, |C|)$.

While $|M|$ can contain up to 15% of the tokens, this approach allows IMOSA to consider the most influential tokens concerning text data augmentation. Further, it is worth noting that $|M|$ can be 0 if no candidate tokens were selected (i.e., $C = \emptyset$). This is one of the two methods used by IMOSA to completely eliminate input examples that might not support effective text data augmentation.

Masked multiplier. Though we optimise the masking process to identify the most suitable candidates (i.e., tokens) to replace and generate a synthetic sentence, the pruned masking alone cannot guarantee that we select the most favourable subset to mask the tokens in a sentence. Further, allowing only a single subgroup of masked tokens per sentence limits the data augmentation algorithm's ability to introduce the necessary diversity to the augmented dataset.

Thus, we propose to generate multiple masking subsets M_j with non-overlapping tokens using C. In this way, we create a pool of masked input tokens T_j^m for replacement by masking the corresponding tokens in T based on the tokens in each M_j. This technique allows us to generate multiple distinct masked sequences from the same original sentence to improve the diversity of the dataset while enabling the algorithm to select the most favourable masking to generate a high-quality synthetic sentence. A parameter α controls the maximum number of masked sequences generated per sentence.

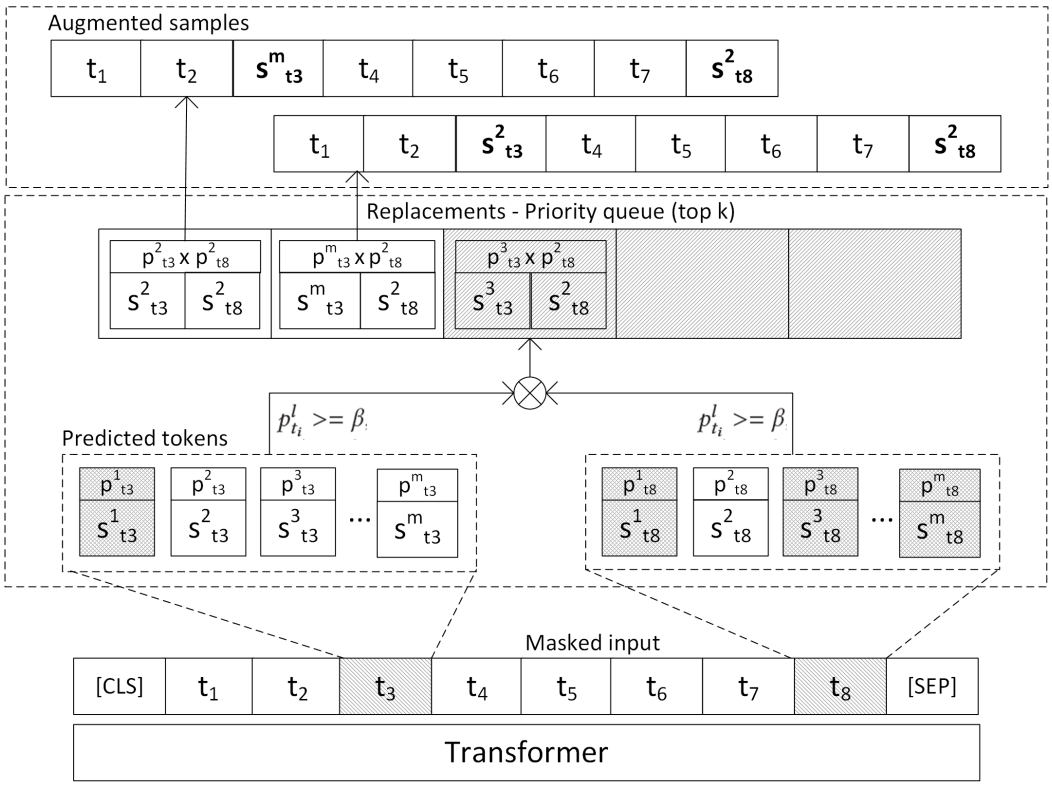

Figure 4.9: Generation of the most probable synthetic sentences from masked token predictions.

4.3.1.3 Optimal Substitution

In this stage, we use the fine-tuned Conditional BERT model to obtain predictions for the masked tokens in an input sentence. Figure 4.9 shows the main steps to obtain synthetic sentences for a particular sentence. We input T_j^m where $j <= \alpha$ and capture the BERT predictions (token and the corresponding probability) for each masked token in T_j^m. We retain only the top-k highest probability predictions for each masked token in T_j^m, if, and only if, the probability of such a token is greater than a threshold, β. We use $s_{t_i}^l$ and $p_{t_i}^l$ to represent lth predicted token and its probability, respectively, where t_i is the corresponding masked token of the original sentence. If $p_{t_i}^l >= \beta$, we regard $s_{t_i}^l$ as an important token. In this case, we may end up with no predicted tokens ($0 \leq l \leq k$) for some masked tokens. For such masked tokens, we consider $s_{t_i}^0 = t_i$. Similar to pruned masking (4.3.1.2), this technique optimises the effectiveness of the proposed algorithm by replacing only the highly potential tokens in a particular sentence. Further, this is the second technique employed by IMOSA to eliminate some less effective input examples entirely.

Next, we consider set possible permutations Q^j of $|T_j^m|$ items by taking one token $s_{t_i}^l$ with $p_{t_i}^l$ at a time from each set of predictions corresponding to each masked token in T_j^m. The ith permutation is represented as an array $Q_i^j \in \mathbb{R}^{|T_j^m|}$ and each rth item in Q_i^j consists of a predicted token q_r^t and the corresponding probability q_r^p. Then, we calculate the likelihood of the token replacements $L_{Q_i^j}$ for a particular sentence by calculating the joint probability between the new tokens q_i^t:

$$L_{Q_i^j} = \sum_{r=0}^{|T_j^m|} \prod q_r^p \tag{4.4}$$

We place each Q_i^j in a priority queue of K elements, corresponding to $|T_j^m|$, where the priority is the $L_{Q_i^j}$. This approach allows the proposed augmentation method to select the most optimal substitutes to generate K synthetic sentences from a source sentence against a particular mask combination T_j^m. The parameter K controls the maximum number of new synthetic sentences generated per mask combination for a particular sentence.

4.3.2 Experiments on IMOSA

To empirically evaluate our proposed text data augmentation technique, we study three benchmark datasets and compare the performances against two state-of-the-art MLM-based data augmentation methods: SSMBA [127] and CBERT [126]. As discussed earlier, the experiments focus on the challenging task of text data augmentation in the low data regime. Apart from that, we evaluate the outcome of the proposed algorithm in alignment with context-aware representations, which are recent advancements in NLP.

Datasets. We select three text classification datasets, as listed in Table 4.3. Since our focus is on generating synthetic sentences under extreme low data settings, we ignore any train/test split in the original datasets and merge all the splits to create a single dataset. Table 4.3 provides summary statistics of the datasets.

Table 4.3: Datasets.

Dataset	Size	Vocabulary	Num. Classes
SST-2	8741	9613	2
SUBJ	10000	21323	2
RT	10662	20287	2

For convenience, a brief description of each dataset is provided:

- *SST-2.* The Stanford Sentiment Treebank (SST) [167] is a corpus of fully annotated parse trees that enables a thorough examination of sentiment in natural language. It contains movie reviews and human annotations of their sentiment. The binary classification (negative and positive) dataset on whole sentences of SST is known as SST-2, or SST-binary [168].

- *SUBJ.* A Subjectivity analysis is a task that is similar to sentiment analysis, with the purpose of classifying an opinion as subjective or objective. The Subjectivity dataset (SUBJ) [169] contains sentences or phrases that are at least 10 words long, drawn from movie reviews or plot summaries.

- *RT-polarity.* A Sentiment polarity dataset is another movie review dataset published

by Pang and Lee [169]. The reviews are extracted from the website RottenTomatoes (RT) and are automatically annotated based on review ratings.

Data splitting. Data splitting is widely used in machine learning to divide data into train, test and validate sets. We use this method to discover the hyperparameters of a model and estimate its generalisation performance. We specifically employ double cross-validation (DCV) [170] to conduct the experiments to improve the reliability of the results. The DVC process comprises two nested cross-validation loops: internal and external. We divide each dataset into two subsets referred to as training and test sets. We use the training dataset in the internal loop of DCV for hyperparameter tuning and model building. The test set is used exclusively for model assessment in the external loop.

To simulate an extremely low labelled data scenario, we use 80% of the dataset as test data (D_S) and use this data only to assess the model. Out of the remaining 20% (training set), we select only a set of 50 records (D_T) to train the model in the inner loop. The rest of the training set examples (D_V) are used as a validation dataset for hyperparameter tuning. Table 4.4 shows the size of the splits for each dataset. For each dataset, using five random seeds, we create five different groups G_i (each containing different examples for D_T, D_V and D_S), where $i \in (1, 2, 3, 4, 5)$.

Table 4.4: Dataset splits.

Dataset	Internal Loop		External Loop
	Train	Validation	Test
SST-2	50	1749	6942
SUBJ	50	2000	7950
RT-polarity	50	2133	8479

4.3.2.1 Baselines

We compare the proposed algorithm against two state-of-the-art text data augmentation methods. CBERT [126] is an MLM-based text data augmentation algorithm that fine-tunes BERT using a label-conditional LM architecture to generate new sentences. SSMBA[127] reconstructs corrupted text with MLMs. SSMBA ensures that the new sentences lie within the manifold neighbourhood of the original example.

4.3.2.2 Augmentation Settings

We augment the training dataset D_T in each group G_i using CBERT, SSMBA, and the proposed technique, IMOSA. We discuss below the details related to each of the data augmentation techniques. We use the pre-trained $\text{BERT}_{\text{BASE}}$-uncased version as an MLM across all the experiments. Additionally, to evaluate the performance in a state-of-the-art context-aware setting, we use the same pre-trained BERT model as a text classifier for the downstream tasks.

CBERT. We fine-tune the Conditional BERT model (see Section 4.3.1.1) using the training dataset D_T, which contains only 50 records, separately for each group G_i. We use three epochs and 5e-5 as the learning rate. Wu *et al.* [126] achieved the best results for the SST-2 dataset with these parameters. Further, we use a batch size of four, as recommended by Kasthuriarachchy *et al.* [124], for fine-tuning BERT under limited labelled data settings.

Next, we use the trained Conditional BERT model to generate synthetic sentences for the original sentences in D_T. For augmentation using CBERT, 15% of the input tokens were randomly masked (i.e., $sample\_ratio = 15\%$). We generate two synthetic sentences per original sentence by setting the $sample\_num$ parameter to two. We keep the default values for the other properties.

To conduct the experiments, we use Wu's [126] implementation in the released code.[2]

Table 4.5: IMOSA configurations for each dataset.

Dataset	IMOSA Parameters				
	$sample\_ratio$	α	K	k	β
SST-2	0.15	4	2	8	0.4
RT-polarity	0.15	4	2	8	0.45
SUBJ	0.15	4	2	8	0.6

IMOSA. We use the same Conditional BERT model trained in Section 4.3.2.2 to generate new sentences using IMOSA. Similar to CBERT, for IMOSA, we set the $sample\_ratio$ to 15%. However, in the context of IMOSA, the actual percentage of masked tokens can vary from

[2] https://github.com/1024er/cbert_aug

0% to 15% due to pruned masking (see Section 4.3.1.2). Additionally, we set the number of masks per sentence α to four and the number of samples K (i.e., the maximum number of synthetic sentences per original sentence) to two, allowing IMOSA to generate a maximum of eight new sentences per original sentence. Further, we set the top-k most probable tokens selected for replacing a masked token to $(\alpha \cdot K)$, the default value for the parameter k. Finally, we set the probability threshold β to different values, as shown in Table 4.5, instructing IMOSA not to replace a masked token if the predicted token has a probability of less than β. Table 4.5 summarises the IMOSA configurations used for each of the datasets.

We select the parameters of IMOSA mentioned above, primarily focusing on keeping the augmented dataset size close to the CBERT output above to ensure comparable results. The optimal parameter values of IMOSA are task-specific; however, we found that the following range of values works well across all tasks:

- Number of masks (α): 2, 3, 4
- Number of samples (K): 1, 2
- Threshold (β): 0.3, 0.4, 0.5, 0.6

SSMBA. In the context of SSMBA, we set the probability for selecting a token for noising ($noise\_prob$) to 0.15. Selected tokens are then masked, randomly replaced or left the same. Further, as suggested by the SSMBA authors, we set the top-k most probable tokens ($topk$) to unrestricted (i.e., -1), as SSMBA works best when it can explore the manifold without any restrictions. In this case, SSMBA draws samples from the full probability distribution of predicted tokens. Finally, we set the number of augmented samples to generate for each input example ($num\_samples$) to two to ensure comparable results. We keep the default values for the other parameters. However, as per the experiments performed by the SSMBA authors, they observed peak performance at 45% corruption. Thus, we carried out an additional set of experiments assigning 0.45 as the new $noise\_prob$ value. This new experiment is identified as SSMBA-45.

To conduct the experiments, we use Ng's [127] implementation in the released code.[3]

[3] `https://github.com/nng555/ssmba`

4.3.2.3 Evaluation Method

Pertaining to performance evaluation, we focus on two crucial aspects: context-aware NLP setting and generalisability. To this end, we fine-tune the pre-trained BERT-base-uncased model to classify text into specific sets of categories.

Next, to guarantee the generalisability of the results while minimising the impact of the generally unstable fine-tuning process, we employ a variation of DCV (see Section 4.3.2) to evaluate the performance of the proposed text data augmentation technique, IMOSA. During the internal loop, we perform a grid search to identify the best parameters for the model and build a model at the end of the internal loop using D_V and the selected parameters. The optimal hyperparameter values of BERT are task-specific. Thus, as recommended in [15], we consider the following range of possible values during the grid search:

- Learning rate: 5e-5, 3e-5, 2e-5
- Number of epochs: 2, 3, 4.

Since we focus on text generation under extremely low labelled data conditions, we use a smaller batch size of four, which is recommended for text classification in the low data regime using pre-trained BERT models [124].

Subsequently, we assess the trained model using D_S in the external loop. The performances of the models are averaged across seeds (i.e., groups) to obtain the overall achievement of a particular model. We use F1-score [171], which emphasises false positives and false negatives, to evaluate the models.

4.3.2.4 Results

This section presents the results of text classification, as summarised in Table 4.6. For each experiment, the table shows the average F1 score across five independent experiments conducted using a different set of examples in each group G_i, where $i \in (1, 2, 3, 4, 5)$. Similarly, the number of sentences and vocabulary size is calculated by taking the averages for five groups.

Table 4.6: Accuracies.

Dataset	Aug. Method	Num. Sentences	Vocab. Size	F1 Score
SST-2	None	50	386	75.15
	CBERT	150	472.2 (+22.3%)	80.08
	SSMBA	149.2	632.2 (+63.8%)	79.77
	SSMBA-45	149.2	912.6 (+136.4%)	77.56
	IMOSA	152	421 (+9.1%)	**81.15**
SUBJ	None	50	517.2	90.0
	CBERT	150	645.4 (+29.9%)	91.72
	SSMBA	150	811.2 (+58.9%)	91.60
	SSMBA-45	150	1161 (+128.3%)	91.15
	IMOSA	138.8	571.4 (+12.7%)	**92.34**
RT-polarity	None	50	448.4	69.53
	CBERT	150	582.4 (+24.8%)	72.66
	SSMBA	150	712.8 (+56.8%)	71.37
	SSMBA-45	150	1023.8 (+124.9%)	71.69
	IMOSA	147.4	505.4 (+10.5%)	**76.45**

We consider classification accuracy with no data augmentation (i.e., None) as the baseline. Further, CBERT and SSMBA provide the benchmark accuracies using the data augmentation techniques proposed by Wu *et al.* [126] and Ng *et al.* [127], respectively. Additionally, SSMBA-45 shows the accuracy of the SSMBA method with the best-performing parameters reported in [127]. All the text data augmentation techniques, including IMOSA, reported considerable performance improvements across all three downstream tasks, confirming the distinct advantage of text augmentation under extremely low labelled data conditions. Across all datasets, models trained with IMOSA outperformed benchmark models. IMOSA obtained a significant absolute accuracy (F1 score) improvement over the state-of-the-art MLM-based data augmentation technique, CBERT, across all three datasets. Nevertheless, CBERT outperformed both SSMBA and SSMBA-45 against all three datasets under extremely low labelled data conditions. Interestingly, SSMBA surpassed SSMBA-45 against SST-2 and SUBJ datasets while showing almost similar performance with the RT-polarity dataset.

The SSMBA method recorded a relative increase of close to 60% over the original vocabulary size of the training dataset in all three downstream tasks, whereas SSMBA-45 reported an upturn of over 120%. These outcomes show the SSMBA's ability to extensively improve the diversity of the augmented dataset. However, the results suggest that diver-

sity alone is not sufficient to guarantee state-of-the-art performance, particularly under the low data regime. Further, it is worth noting that the significant increase in the vocabulary size with SSMBA-45, despite 45% of input tokens being selected for replacement, did not help the model to perform better. Conversely, our text data augmentation technique, IMOSA, increased the vocabulary size by approximately 10%, under multiple strict restrictions applied to maintain the quality of the synthetic sentences. IMOSA considers only a subset of tokens from the entire set of tokens due to pruned masking (see Section 4.3.1.2), and the remaining tokens are ignored. Further, a β value of 0.4 or greater has been considered for IMOSA. In this case, only the predicted tokens with a probability of 40% or more are retained to generate synthetic sentences. Despite the tight restrictions, the results revealed that IMOSA could add sufficient diversity to the augmented dataset. It can be observed that adding diversity under controlled conditions to ensure the quality of the synthetic sentences is highly effective.

The overall accuracy across the three datasets suggests that the RT-polarity task is more challenging. IMOSA achieved an absolute accuracy (F1 score) improvement of 1.06% and 0.62 over CBERT, the second-best performer, in SST-2 and SUBJ datasets, while recording a relatively more remarkable improvement of 3.8% over CBERT in the RT-polarity task. This confirms IMOSA's robustness and its ability to perform under challenging conditions. A more in-depth analysis of the key features in IMOSA that contributes to these achievements is discussed next.

4.3.2.5 Analysis and Discussion

In this section, we analyse the factors that influence IMOSA's performance.

Number of synthetic sentences. The average number of sentences in the augmented datasets in Table 4.3 revealed that both CBERT and SSMBA augmented almost all the original examples (two new sentences per original example). Although SSMBA technically has the ability to ignore sentences by retaining the corresponding original token of a masked token, this aspect of the algorithm did not play a noticeable role. In contrast, IMOSA heavily relied on identifying the most favourable sentences to augment.

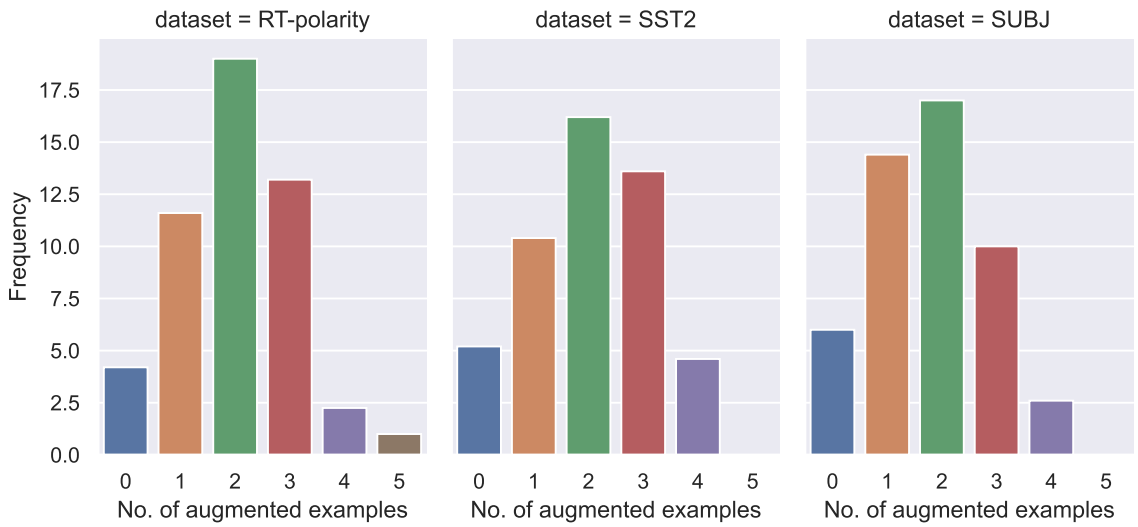

Figure 4.10: Frequency distribution of the number of synthetic sentences generated per original example in the training set.

Figure 4.10 illustrates the frequency distribution of the number of sentences generated by IMOSA against an original example. We have reported the average frequency across the five groups of data splits, G_i. It is important to note that while SSMBA and CBERT were allowed to generate a maximum of two examples for each original sentence, IMOSA's configurations enable it to create a maximum of eight $(\alpha \cdot K)$. The number of actual synthetic sentences generated for each original example varies from zero to five, where zero represents the original sentences that were ignored while replacing words to generate new examples. Approximately 10% of the original sentences in the training dataset have not been considered, showing IMOSA's ability to focus on the most favourable examples to generate synthetic sentences. Further, the distribution reveals how IMOSA focuses on progressively developing more quality sentences using appropriate original examples. While over 30% of the training examples were utilised to generate two synthetic sentences, a few examples were extensively used to create five new sentences.

Diversity. The configurations of IMOSA allow it to mask 15% (i.e., $sample\_ratio = 15$) of input tokens that are considered candidates for replacement. However, due to pruned masking (see Section 4.3.1.2), the percentage of tokens eligible for masking could be less than 15%. Conversely, the substitution threshold, β, discussed in Section 4.3.1.3, might decide not to replace a masked token if the probability of a predicted token is less than β. As

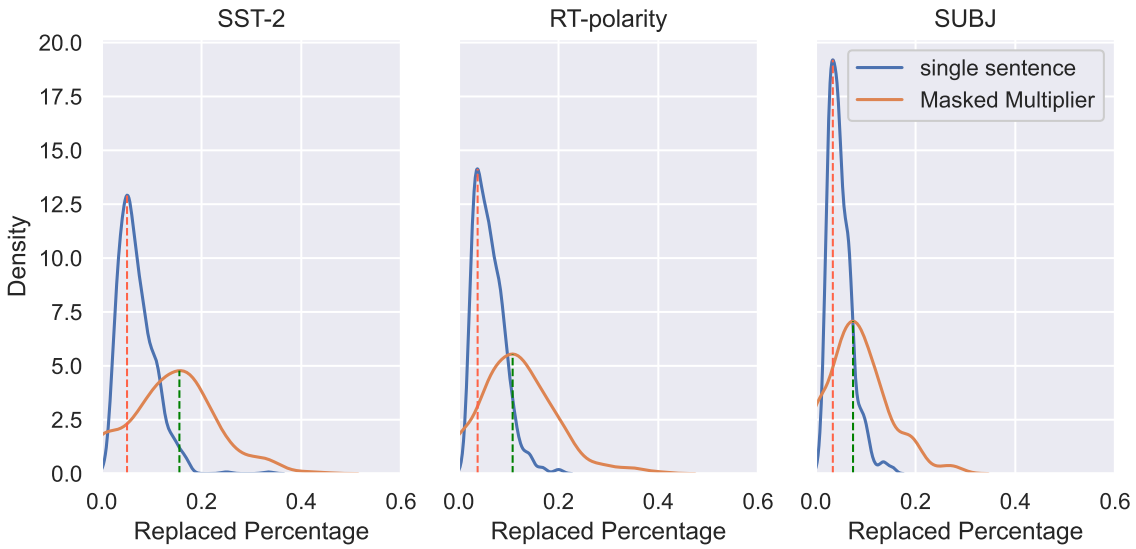

Figure 4.11: Distribution of percentage of masked tokens.

these features were designed to maintain the quality of synthetic sentences, the effective number of tokens replaced can be significantly less than the $sample\_ratio$ value, hindering IMOSA's ability to add diversity to the augmented dataset. As depicted in Figure 4.11, if we consider individual examples, the percentage of tokens replaced was significantly below the expected 15%. Based on the density plot for individual sentences, most values are located at 5.1%, 3.8% and 3.4% for SST-2, RT-polarity and SUBJ, respectively.

While pruned masking and substitution thresholds are essential to generate meaningful synthetic sentences, the right-skewed density plots with very low mode values revealed their significant impact on adding new words to the training dataset. To this end, IMOSA is equipped with Masked Multiplier, a technique that allows IMOSA to add sufficient diversity to the augmented dataset using a particular sentence. As discussed in Section 4.3.1.2, Masked Multiplier is capable of generating multiple distinct masked sentences for a given original sentence, enabling IMOSA to consider a significantly large percentage of favourable tokens for replacement. As shown in Figure 4.11, Masked Multiplier has shifted the density plot to the right drastically with peak values at 15.6%, 10.8% and 7.5% for SST-2, RT-polarity and SUBJ, respectively, confirming its effectiveness.

Further, it is worth noting that, overall, the Masked Multiplier was able to replace a significantly higher percentage of tokens than the specified value for the $sample\_ratio$

while keeping the replacements per synthetic sentences below the $sample\_ratio$ value. Moreover, this technique helps to mitigate the impact on the quality of the predictions due to large $sample\_ratio$ values since BERT neglects dependency among predicted tokens [15]. While this is one of the key reasons behind IMOSA's performance, SSMBA-45, which masked 45% of the input tokens, performed poorly compared to SSMBA.

Nevertheless, IMOSA recorded only a peak replacement percentage of 7.5% with the SUBJ dataset compared to 15.6% with SST-2 dataset. This is mainly due to the high β value in the IMOSA configurations used with the SUBJ downstream task, favouring very high quality over diversity. In this case, IMOSA focuses more on maintaining the meaning and quality of the synthetic sentences, replacing masked tokens only with high probable predicted tokens. In the context of these experiments, we used a significantly high β value, mainly to control the augmented dataset size, enabling us to compare the results against the performance of CBERT and SSMBA.

4.4 Summary

In this chapter, we have presented two novel text augmentation techniques that focus on creating synthetic sentences that maintain similar meaning to the original sentences. Our study on semantic data augmentation proves the importance of maintaining the semantics of synthetic sentences to further contribute to the overall performance of context-aware NLP tasks. Moreover, we evaluated our novel text augmentation technique, IMOSA, on various NLP tasks and found that it outperforms several state-of-the-art text augmentation techniques.

In the next chapter, we introduce novel optimisation for manual data labelling, one of the critical challenges in real-world NLP tasks. For this, we propose a cost-effective text annotation methodology using crowdsourcing platforms. Further, we present an end-to-end framework to build and develop social intelligence solutions using NLP.

Chapter 5

Social Media Intelligence and Text Analytics

5.1 Introduction

In Chapters 3 and 4, we presented novel techniques to significantly improve the performance of NLP models with noisy text content under limited labelled data conditions. Our contributions can be applied to boost the effectiveness of SMA solutions in the real world.

SMA refers to the gathering of data from social media sites and blogs to analyse and inform business decisions. SMA goes beyond the standard monitoring or the rudimentary study of Retweets or 'likes' to generate a more comprehensive understanding of the social consumer. Further, social media listening, which is a branch of SMA, focuses on tracking direct reference to an entity, such as a person, event or organisation, to understand the audience's behaviour around a theme [172]. However, social media intelligence is required to provide a complete and sophisticated understanding of a target audience [172]. In social intelligence projects, organisations use social media data to answer specific questions or solve a problem.

In the context of social intelligence, it is a popular approach in the industry and academia to derive insights from texts to provide solutions across many disciplines, including consumer behaviour, politics, disaster management and sociology [173, 174, 175]. Unstructured

data in microblogging platforms, such as Twitter and Facebook, have seen rapid growth and are among popular text data sources for such use cases, given their popularity with users for the sharing of opinions and thoughts that could be relevant for business decision-making. However, the unstructured and informal nature of social media data can introduce significant noise and diversity, thus making the task very challenging. Further, the text classification frequently suffers from insufficient labelled data because the labelled datasets are often manually annotated.

This chapter aims to develop a social media intelligence framework, focusing on text analytics use cases, to provide a more efficient and effective approach for real-world applications. This framework can help SportsHosts to deploy an AI-driven digital marketing solution that optimises audience targeting and personalisation. SportsHosts can combine social listening with novel AI and NLP techniques to identify prospective sports fans across the globe more efficiently and effectively using popular social media platforms, such as Twitter. Firstly, to address a crucial step, we present a cost-effective data annotation technique to obtain manually annotated training data required for many NLP tasks, including text classification. Secondly, in Section 5.3, we introduce an end-to-end framework for text classification based social intelligence solutions, combining our key contributions discussed in Chapters 3 and 4 and the proposed cost-effective text annotation framework in Section 5.2.

5.2 Cost-Effective Data Annotation

Modern technologies and media, such as social media and smart devices, have made available vast quantities of text-based data, including microblogs, reviews, personal messages and news articles. Further, social media platforms such as Twitter, Facebook and Reddit have captured the attention of both the industry and academia due to the wealth of user-generated information they hold. The popularity of using Twitter data for research and real-world use cases in the NLP domain remains high due to its huge active user base and the tremendous number of text messages (Tweets) every day.

However, any effort to study and generate insights from informal discussions on social media sites will require at least a minimum number of labelled data for evaluation purposes. For instance, SportsHosts will require labelled data related to sports fans to deploy a supervised text classification model to identify potential sports fans using Twitter messages. Traditionally, domain experts or the researchers themselves have annotated the necessary training data, often at considerable costs in terms of time and money. Recently, however, crowdsourcing platforms, such as Amazon's Mechanical Turk (MTurk), have been leveraged to create large training corpora at significantly lower costs [176].

Nevertheless, creating labelled datasets for various use cases remains challenging, as the percentage of useful Tweets is scarce in most of the scenarios. This is mainly due to the high volume of Twitter messages posted by individual users and organisations to promote a product or service. These messages tend to contain a set of keywords (hashtags) similar to useful Tweets, making it harder to differentiate them. Further, in the context of text classification tasks, often, useful Tweets are highly skewed across target classes. Thus, it is inevitable that we annotate a large corpus of Tweets to obtain a sufficient amount of labelled training data for each target class. In the context of data annotation, the quality of the labels received from annotators varies. Some annotators may provide an incorrect label due to a genuine misinterpretation of the definition. Given that annotators are paid a small amount per task completed on MTurk, some annotators may also rush to complete tasks to maximise earnings, thus providing random or poor quality labels at the expense of accuracy.

To overcome these challenges, we propose a novel framework to annotate text data specifically for Twitter-based text classification use cases at significantly lower costs using crowdsourcing platforms. The proposed framework leverages zero-shot text classification to pre-annotate the unlabelled dataset to develop specialised sub-datasets based on the target labels. These sub-datasets can be used to address the challenges of data imbalance across target classes. Further, we propose to introduce additional labels to the annotation task to ensure annotation quality. These additional classes will help to reduce genuine errors due to misjudgement and misinterpretation. Moreover, the annotation framework

consists of a two-stage annotation process to reduce cost while maintaining annotation quality.

The rest of this section is organised as follows. Section 5.2.1 introduces the proposed annotation framework, and the subsequent sections present the key components of the framework: meaning-sensitive pre-processing, novel pre-annotation using zero-shot text classification and specialised sub-datasets. Section 5.2.6 discusses a multi-phase approach to annotating a dataset using the proposed framework.

5.2.1 Text Annotation Framework

The main components of the complete framework are depicted in Figure 5.1. This systematic, step-by-step methodology is proposed to create training datasets for text classification tasks using a crowdsourcing platform. The main steps of the framework are as follows.

Step 1: Tweet extraction. Tweets can be extracted based on suitable keywords using Twitter's standard search Application Programming Interface (API) without any financial cost. However, the number of Tweets that can be extracted per request is limited. Thus, Tweets need to be collected over some time to obtain the required number of Tweets. Additionally, $lang : en$ and $-is : retweet$ parameters are used to obtain only Tweets that are in English and original (i.e. eliminating Tweets that have been re-posted by another user), respectively.

Step 2: Raw dataset preparation. In Step 2, we extract the $user.id$, $idstr$ and $full\_text$ fields from the extracted Tweet objects to create a 'Raw' dataset. Further, we filter out any record with $user.verified = true$ to eliminate Tweets authored by known organisations or companies, who are more likely to post promotional Tweets.

Step 3: Meaning-sensitive pre-processing As the third step, we perform meaning-sensitive pre-processing on the 'Raw' dataset, as discussed in detail in Section 5.2.2. This helps to improve the accuracy of the zero-shot text classification, as we propose to use a state-of-the-art BERT model trained on MNLI. The BERT model is sensitive to the structure of the

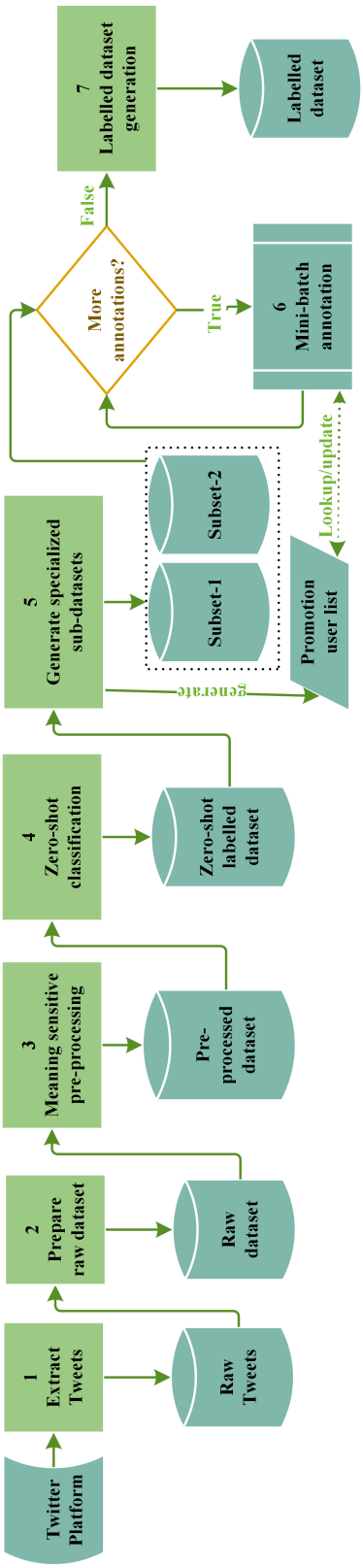

Figure 5.1: Annotation framework.

input sentences, as it was trained on datasets with structured sentences [15].

Step 4: Pre-annotation. Next, we forward the pre-processed instances through a zero-shot text classification model to obtain class probabilities, as discussed in detail in Section 5.2.3.

Step 5: Specialised sub-dataset generation. The probabilities obtained for the identified target classes in Step 4 are used to generate specialised sub-datasets for each target class. As discussed in Section 5.2.4, the majority of the instances in each sub-dataset are expected to be from a single class. These subsets play a key role in reducing the impact of class imbalance.

Step 6: Mini-batch annotation. In Step 6, we generate mini-batches using specialised sub-datasets as discussed in Section 5.2.6.1 and submit them to a crowdsourcing platform for annotation in a two-stage process (see Section 5.2.6.2). Further, the proposed framework recommends a phase-wise approach to perform the mini-batch annotation, as discussed in Section 5.2.6.1.

Step 7: Labelled dataset generation. In the last step, we combine all the 'Reconciled' batches generated in Step 6 to create the final training dataset.

5.2.2 Meaning-Sensitive Pre-Processing

Meaning-sensitive pre-processing is one of the novel components of the proposed annotation framework. In this section, we discuss the details of the meaning-sensitive pre-processing module.

Pre-processing is a preliminary step in text classification, and identifying a suitable pre-processing technique plays an important role in model accuracy [177]. Traditionally, pre-processing techniques–such as stemming, lemmatising and handling negation by adding a prefix, 'NOT_'–have been demonstrated as effective [178]. However, these pre-processing techniques add more noise to a sentence from the perspective of context-aware language

models such as BERT. The BERT model is more sensitive to sentence structure, as it was trained bidirectionally on a large corpus of unlabelled text, including the entire Wikipedia and BookCorpus [15]. Thus, we propose to use only the pre-processing techniques that have the least impact on the sentence structure and meaning, as follows.

Replace elongated words. We replace elongated words with their source words.

Spelling correction. We correct any spelling mistakes using a corrector. While no corrector is perfect, they have some–usually high–accuracy of success [179].

Split hashtag. We propose using word segmentation [180] to split a hashtag of a Tweet into meaningful words. This pre-processing technique is specific to Tweets and helps to improve the intended meaning of a Tweet.

Replace user mentions with a pronoun. Replacing *user mentions* (another account's Twitter username, preceded by the '@' symbol) with a keyword, such as 'AT_USER', is a commonly used pre-processing technique for Tweets. However, this approach interferes with the structure and meaning of the sentence. Thus, we propose to replace user mentions with pronouns. We randomly replace a single *user mention* at the beginning and middle/end of a sequence with 'he/she' and 'him/her', respectively. Further, if it is a sequence of multiple *user mentions*, we replace the set of *user mentions* at the beginning of a sequence with 'they' and with the word 'them' otherwise. Moreover, we avoid using the most common and popular techniques such as stemming, lemmatising and removing stop words, as they can affect the meaning of a sentence.

5.2.3 Pre-Annotation with Zero-Shot Classification

The zero-shot classification module is one of the novel components that we introduce to this framework. This section discusses the zero-shot classification module of the annotation framework in detail.

Hypothesis. As the proposed zero-shot text classification technique is based on a sequence-pair classification, the hypothesis plays an important role in obtaining better results. We propose to identify multiple hypotheses H_c^i based on the business definition of each class $c \in C$. Multiple hypotheses will help to capture most of the relevant data for a selected class c.

Zero-shot text classification. We propose to use a BERT model trained on MNLI, including the last layer, which predicts one of the three labels–$contradiction$, $neutral$, and $entailment$–to pre-annotate the dataset. Kasthuriarachchy *et al.* [181] have shown that BERT-based models can be effectively used with noisy texts such as Tweets.

As we have multiple hypotheses for each candidate label, each sentence/hypothesis pair is forwarded through the model as a premise/hypothesis pair, and we obtain $O_h \in \mathbb{R}^{k \times 3}$, which contains logits for these three categories for each hypothesis. k is the total number of hypotheses across all the candidate labels. Afterwards, we perform a softmax over $entailment$ logits over all the hypotheses H_c^i, where $c \in C$, to derive corresponding probabilities. Finally, we calculate the probability $P_c = \sum_i P_{H_c^i}$, for each candidate label, thereby obtaining $O \in \mathbb{R}^{|C|}$ containing the pre-annotation probabilities for the candidate classes. Finally, we obtain $\hat{c}$ by calculating the $argmax()$ of the calculated probabilities O.

5.2.4 Specialised Sub-Datasets

The pre-annotation probabilities obtained for each text sequence using the zero-shot classification module can be used to generate specialised sub-datasets D_c for each class $c \in C$. For each text sequence, we obtain the predicted class $\hat{c}$ based on the highest pro-annotation probability. Each sub-dataset D_c contains text sequences where $c = \hat{c}$ and $P_c >= 0.5$. Thus, each D_c contains a high proportion of potential class c instances.

5.2.5 Handling Promotional Tweets

A significantly large percentage of Tweets are posted to promote a product or service, but these Tweets typically have no relevance for most text classification use cases. As the pro-

portion of promotional Tweets can be significantly high in an unlabelled dataset, impulsive submission of such promotional Tweets to MTurk for annotation can become very expensive for little gain. Further, these Tweets may easily confuse non-expert annotators, leading to inferior annotation quality. Hence, the proposed framework includes a mechanism to manage promotion Tweets, which is a crucial component in managing the data imbalance and reducing the annotation cost.

We introduce an additional class to the annotation task to identify promotion Tweets. For Tweet-specific use cases, we consider a 'promotion' class to be included in the set of classes C. We use a separate hypothesis during the pre-annotation process to assign a separate promotion-specific probability to each text sequence, as discussed in Section 5.2.3. We generate a promotion user list by identifying the authors of Tweets for which $\hat{c}$ ='promotion' and $P_{promotion} > 0.8$. There is a high probability that these Tweets are promotion Tweets. We use this list of promotion users to filter out potential promotion Tweets while generating mini-batches. Further, we update the promotion user list after each mini-batch annotation based on the promotion Tweets in the 'Reconciled Batch'.

5.2.6 Multi-Phase Approach

We propose using a novel multi-phase approach to annotate a training dataset for improving the quality of the annotation while reducing the cost further. Firstly, a phase-wise approach to annotating the full dataset using the proposed annotation framework is introduced. This approach involves performing an end-to-end annotation of mini-batches in multiple phases as opposed to annotating the full dataset in a single pass. This helps to improve the annotation quality significantly, as discussed below. Secondly, we propose a two-stage process to annotate a single mini-batch to reduce the cost while maintaining the quality of the annotations. We discuss the two-stage batch annotation process in detail in the Section 5.2.6.2 below.

5.2.6.1 Phase-Wise Annotation

Mini-batch annotation. Annotating mini-batches is one of the key concepts in the phase-wise annotation process. We recommend submitting small batches of unlabelled data for annotation and increasing the batch size gradually once the annotation quality and data distribution are reasonably consistent. This approach allows us to manage the issue related to substandard annotation quality and data skewness while significantly reducing the annotation cost. Further, mini-batch annotation allows us to develop a training dataset in multiple phases, enabling further quality improvements. We use different strategies to generate mini-batches in each phase, as discussed in Section 5.2.6.1.

Annotation phases. We propose a phase-wise (three phases) approach to improve the quality of the annotations by revising the annotation schema based on the analysis performed after each mini-batch annotation. The proposed phases are meant to help reduce annotator disagreement rates and incorrect annotation percentages (both the annotators agree, yet the label is incorrect). Table 5.1 shows the key characteristics of each phase based on the purpose of each phase discussed below.

Table 5.1: Characteristics of phases.

Characteristic	Phase 1	Phase 2	Phase 3
batch size	small	medium	large
annotator disagreements	high	low	low
incorrect annotation percentage	high	low	very low
annotation schema revisions	major	minor	none

Phase 1. During Phase 1, a mini-batch from each sub-dataset is generated and submitted for annotation as per the mini-batch annotation process discussed in Section 5.2.6.1. We recommend using a smaller batch size (i.e., 100 to 500 instances) for each mini-batch in Phase 1. Then, we obtain a 'Reconciled Batch' (Step 6.3 in Figure 5.2) for each annotated mini-batch and perform an analysis to obtain the following details:

- proportion of instances from each class c in each sub-dataset

- inter-annotator agreement (IAA)
- incorrect annotations (by manually analysing a sample of 100 annotated instances).

The above information is evaluated to decide the suitability of the annotator schema and the quality of the annotations. We recommend revising the annotation schema to increase the IAA while reducing incorrect annotations. We propose to introduce additional classes to handle incorrect annotations. This is intended to reduce confusion and increase the annotator's attention towards those classes, thereby improving the overall annotation quality. For instance, if most of the promotion-related instances are incorrectly labelled, we can introduce 'promotion' as an additional class so that annotators are obliged to focus more on promotion-related text sequences.

We recommend repeating the above steps multiple times until the results are satisfactory.

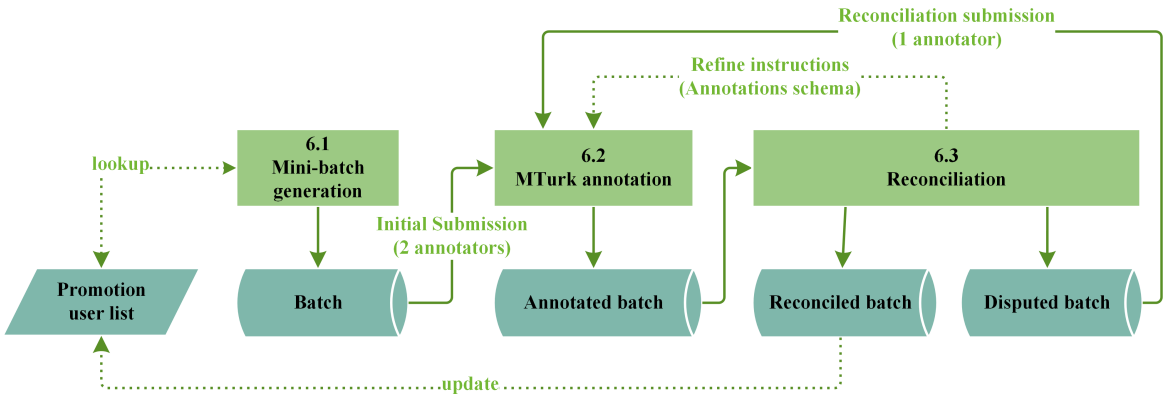

Figure 5.2: Two-stage mini-batch annotation process (Step 6 of the annotation framework).

Phase 2. In Phase 2, we propose generating mini-batches where class labels are uniformly distributed based on the sample distribution of each sub-dataset obtained in Phase 1. Further, this phase focuses on verifying the consistency of the annotation quality by analysing the inter-annotator agreements of each annotated mini-batch. We recommend applying minor modifications to the annotation schema if necessary.

Phase 3. As we have ensured the consistency of the annotation quality and distribution of the class labels, mini-batches with larger batch sizes are generated and submitted to

complete the annotation of the remaining unlabelled data.

5.2.6.2 Two-Stage Batch Annotation

Callison-Burch [182] revealed that MTurk is cheap enough to collect redundant annotations, which can be utilised to ensure annotation quality. However, when multiple labels are necessary, although a single label is cheap, the costs can accumulate quickly. Thus, in our framework, we propose a two-stage annotation process to obtain redundant annotations, reducing the cost further while maintaining the annotation quality. Figure 5.2 illustrates the complete process followed for each mini-batch.

In the first stage, known as 'initial submission' (see Figure 5.2), each instance shall only be annotated by two random annotators. Once the annotations are available, we set aside instances with agreed annotations (i.e., instances with the same annotation from both the annotators) and resubmit the disputed instances to MTurk. In the second stage, known as 'reconciliation submission' (see Figure 5.2), each instance in the disputed mini-batch shall only be annotated by one random annotator, which can be used to break the tie. Finally, we generate a 'Reconciled Batch' containing annotated Tweets from each mini-batch.

Further, for improved annotation quality, we recommend obtaining three and two random annotations in Stages 1 and 2, respectively.

5.3 Social Intelligence Framework

This section presents an overarching framework to deploy social media text analytics models in the real world efficiently and effectively. The proposed framework combines sequential transfer learning and text augmentation to overcome the impact due to data scarcity while leveraging state-of-the-art sentence representations to boost the performance of the NLP task. Further, the framework consists of a text annotation methodology to obtain the required labelled data quickly and cost-effectively. Figure 5.3 shows the key stages involved and the main steps of each stage, including the enhancements proposed throughout this thesis. The following sections discuss the functionality of the key stages of the framework.

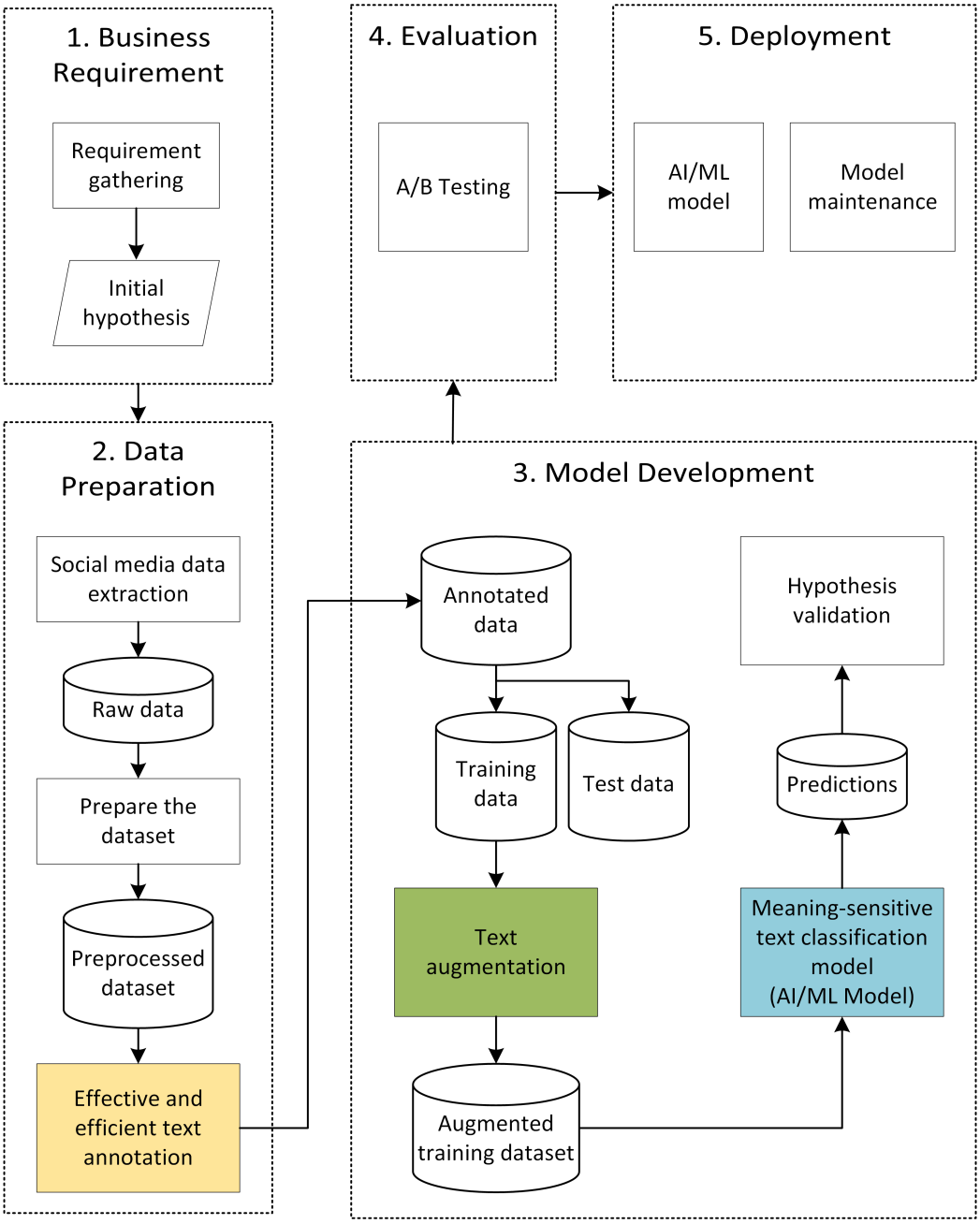

Figure 5.3: High-level framework.

5.3.1 Business Requirement

The first step to achieving social media intelligence using text data is to understand the signals that customers leave every day in billions of text messages and reviews. In this case, organisations must first identify the problem to solve using social intelligence based on their business strategy. Thus, generating a clearly defined business requirement is crucial to ensure the success of a social media intelligence strategy. Next, to evaluate and test the business idea, we must formulate testable, precise and discrete hypotheses. These hypotheses shall drive the next stages of a social media intelligence solution.

5.3.2 Data Preparation

The data preparation stage (Stage 2) focuses on extracting required text data from a social media platform to develop the necessary training data to train AI and ML models. Depending on the platform, APIs or any other mechanism provided can be used to collect the data. According to the hypothesis, information such as keywords, location or language is specified to optimise the collection of useful data. Next, a dataset needs to be prepared after removing duplicates or unnecessary text data for manual annotation. The guidelines for annotations are designed based on the business use case and hypotheses developed. Finally, the annotation framework proposed in Section 5.2 is used to create the labelled dataset at a significantly reduced cost and time.

5.3.3 Model Development

The performance of context-aware NLP models developed using unstructured social media data in a real-world setting is significantly affected by two factors: insufficient labelled training data and difficulty in comprehending the meaning due to the noisy and diverse nature of social media content. To directly address the training data scarcity, the proposed framework leverages a novel text augmentation technique, presented in Section 4.3. This step ensures that the diversity of the training dataset is improved by introducing label-compatible synthetic sentences to the training dataset. Further, in the model development

step, as proposed in Section 3.2, sequential transfer learning techniques are considered to transfer prior knowledge from pre-trained language models to reduce the impact on model performance due to limited labelled data. Apart from that, the noisy texts are represented using the novel sentence embedding technique based on a pre-trained language model, presented in Section 3.3. These sentence embeddings capture essential linguistic features to improve the ML model's ability to comprehend the noisy text, improving text classification accuracy. The use of state-of-the-art pre-trained language models induces context-aware word representations, improving the semantic capabilities of the NLP model, thereby boosting the overall performance of the NLP task.

Finally, the model's performance is evaluated against the hypotheses to verify the effectiveness of the AI and NLP-based approach in solving the identified business problem.

5.3.4 Evaluation

Prior to deploying a machine learning model, its effectiveness needs to be validated in a real-world setting. For this, an A/B test is designed based on the initial hypotheses. When performing an A/B test, the first step is to establish the business objective based on the hypotheses. In the context of social intelligence solutions, we compare the two versions of the solution to the initial business problem: the social intelligence solution and the general solution. This is achieved by splitting the audience into two groups and treating each group with the two versions of the solution. The sample size of each group is estimated based on the expected statistical significance. The performances of both versions of the solution are compared statistically to verify the effectiveness of the social intelligence approach.

After validating the significant improvements of the social intelligence solution, the organisation can roll out new models for production.

5.3.5 Model Deployment

Once the effectiveness of the AI model is established, the solution is deployed to production. Although the model's performance has been verified, the model must continue to be monitored post-deployment to ensure that it continues to perform as expected. As the

model is trained using historical data, ever-changing environments in the real world might produce unfamiliar data, degrading model performance over time. This phenomenon is known as model drift. The AI model must be monitored and maintained along with a re-training strategy to address this issue.

5.4 Experiments

This section describes the results of experiments conducted to evaluate our proposed text annotation framework's effectiveness at pre-annotation and the phase-wise approach in developing an annotated dataset.

Use case. We applied the proposed framework to annotate Tweets to train a model for identifying sports fans. We considered this a binary classification problem and decided to label the Tweets into two classes–'Sports Fan' and 'Other'. For this use case, we restricted the identification of fans to a single selected sport, say X, based on Tweets related to a popular league. As per the business requirement, a sports fan was defined as someone who has attended or wants to attend/watch Sports X. Further, the business is interested in sports fans who positively talk about a game or a related topic.

Dataset. We used the official hashtag to extract Tweets related to the selected league of Sports X using the Standard Twitter API. Further, we applied filters to specify the language as English (*lang:en*) and to ensure that we collected only original Tweets (*-is:retweet*). We collected 48,782 Tweets and filtered out Tweets authored by verified accounts to eliminate accounts of public interest while generating the raw dataset. The raw dataset consists of 45,508 Tweets, and we manually annotated 100 Tweets to understand the distribution of the dataset. There were only 11% 'Sports Fans' Tweets. Most importantly, there were 86% of promotion Tweets in the sample we analysed.

Meaning-sensitive pre-processing. We used Norvig's spelling corrector[1] to automatically correct spelling mistakes. We split the hashtags into meaningful words using a text pro-

[1]http://norvig.com/spell-correct.html

cessor developed by Baziotis, Pelekis and Doulkeridis [183], which is based on the Viterbi algorithm and utilises word statistics (unigrams and bigrams) from an unlabelled dataset. Further, we replaced user mentions in the Tweets with pronouns, as suggested in the proposed framework. Moreover, we replaced elongated words with their source words.

Table 5.2: Hypotheses for pre-annotation.

Class	Hypothesis	
Sports Fan	$H^1_{sportsFan}$:	This text is about watching a game.
	$H^2_{sportsFan}$:	This text is about attending a game.
Promotion	$H^1_{promotion}$:	This text is about a promotion.
	$H^2_{promotion}$:	This text is about gambling.

Pre-annotation. As per the pre-annotation task of the proposed framework, we identified two classes–'Sports Fan' and 'Other'–based on the business requirement. Further, to manage the promotion Tweets, we also included a 'Promotion' class in the pre-annotation classification. Based on the business definition of a sports fan, as recommended in the proposed framework, we identified multiple hypotheses to detect relevant Tweets, as listed in Table 5.2. We did not specify a hypothesis to identify sports fans who positively talk about a game or a related topic since such a hypothesis is generic and would result in an increased amount of false positives. Further, we introduced two hypotheses to detect promotion Tweets. We introduced one hypothesis for the promotion class, specifically to identify gambling-related Tweets, as we observed a large portion of Tweets promoting gambling-related activities. Any Tweet that is not classified as 'Sports Fan' or 'Promotion' was labelled as 'Other'.

Next, we used a BERT model trained on MNLI and forwarded a sentence together with each hypothesis in Table 5.2 to obtain a matrix of logits $O_h \in \mathbb{R}^{4\times3}$. Afterwards, we performed a softmax over $entailment$ logits over all the hypotheses - $H^1_{sportsFan}$, $H^2_{sportsFan}$, $H^1_{promotion}$, and $H^2_{promotion}$ - to obtain probabilities $P(H^1_{sportsFan})$, $P(H^2_{sportsFan})$, $P(H^1_{promotion})$, and $P(H^2_{promotion})$ for each hypothesis respectively. Finally, we calculate the probabilities $P_{sportsFan}$ and $P_{promotion}$ by calculating the $P(H^1_{sportsFan})+P(H^2_{sportsFan})$ and $P(H^1_{promotion})+P(H^2_{promotion})$, respectively.

Specialised sub-datasets. We generated two sub-datasets for 'Sports Fan' and 'Promotion' by selecting Tweets based on the conditions $P_{sportsFan} >= 0.5$ and $P_{promotion} >= 0.5$, respectively. All the remaining Tweets were classified into the 'Other' dataset.

Phase-wise annotation. We used Amazon MTurk as the crowdsourcing platform to annotate the Tweets. In our submissions, we offered US$0.05 for each annotation. As proposed in the framework, we requested two annotations for the initial submission of a mini-batch across all the phases. Any Tweet with non-agreeing annotations in a mini-batch ('Disputed Batch') was resubmitted for a single annotation to resolve the dispute. This section discusses the tasks that we performed using the proposed framework to complete the annotation of the Tweets in a phased approach.

Table 5.3: Annotations schemata.

Version	Annotation Schema	
v1	Positive:	author of the Tweet is a Sports X fan
	Negative:	author of the Tweet doesn't like Sports X
	Neutral:	neither positive or negative (e.g., stating a fact)
	N/A:	promotions, text cannot be understood
v2	Sports Fan:	author of the Tweet is a Sports X fan
	Doesn't like X:	author of the Tweet doesn't like Sports X
	N/A:	promotions, text cannot be understood
v3	Sports X Fan:	author of the Tweet is a Sports X fan
	Promotion/Ad:	Tweets related to promotions
	Doesn't like X:	author of the Tweet doesn't like X
	N/A:	text cannot be understood

As recommended by the proposed framework, during the initial round of Phase 1, we submitted mini-batches of 450 Tweets (i.e., 150 Tweets from each of the three classes). We used Version 1 (v1) of the annotation schema, as shown in Table 5.3, to obtain two annotations for each Tweet. Then we conducted a post-analysis to calculate the IAA. Further, we manually annotated a sample of 100 Tweets from the already-annotated mini-batches to identify incorrect annotations. Table 5.4 shows the post-analysis results for the first round and the subsequent rounds. The IAA was inferior mainly due to confusion between the Neutral and the Positive labels. Further, the incorrect annotation percentage was high as a significant amount of promotion Tweets were either labelled as Positive or Neutral. We as-

sumed that these results were mainly due to misinterpretation of the guidelines provided to the annotators, so we decided to revise the annotation schema to reduce confusion.

Table 5.4: MTurk annotation quality.

Phase	Round	Schema Version	IAA	Incorrect Labels
1	1	v1	31%	26%
1	2	v2	47%	17%
1	3	v3	62%	3%
2	1	v3	64%	4%
2	2	v3	64%	2%

For the second-round submission, we developed a more use case-specific annotation schema (v2), as specified in Table 5.3. Although we improved the IAA and the incorrect label percentage, we still observed a significantly higher percentage of promotion Tweets negatively affecting the annotation quality. Thus, as suggested in the proposed framework, we decided to introduce an additional class for promotion-related Tweets in the third version (v3) of the annotation schema as shown in Table 5.3. We were able to obtain satisfactory results for the third round submissions along with the annotation schema v3. Hence, we decided to progress to Phase 2 of the annotation process with the annotation schema v3. Figure 5.4 and 5.5 show the options provided for the MTurk Worker and the final set of instructions provided in the MTurk project, respectively. At the end of Phase 1, we calculated the class distribution of each specialised sub-dataset using the reconciled mini-batches of each sub-dataset across all three rounds. Table 5.5 shows the identified class distributions.

Table 5.5: The class distribution in specialised sub-datasets.

Label	Specialised Sub-Dataset		
	Sports Fan	Promotion	Other
Sports Fan	76%	5%	13%
Promotion	14%	93%	78%
Other	10%	2%	9%

To generate balanced mini-batches for Phase 2 of the annotation process, we decided to follow a ratio of 0.6:0.1:0.3 between 'Sports Fan', 'Promotion' and 'Other' specialised sub-datasets based on the class distributions. A balanced mini-batch roughly contains an equal percentage of sports fan Tweets and other/promotion Tweets. We completed two

submission rounds of 1,000 Tweets each, and we were able to verify the quality of the annotations through post-analysis, as shown in Table 5.4.

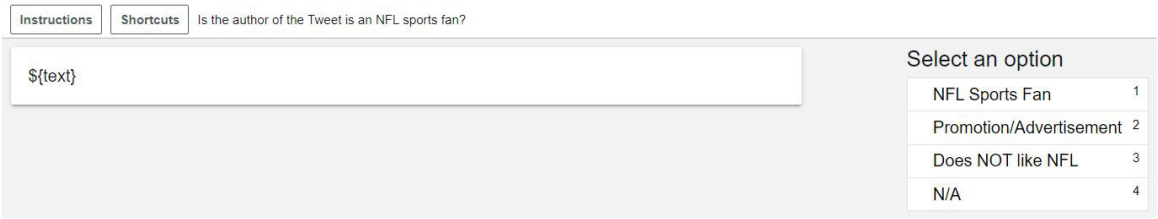

Figure 5.4: MTurk project–annotation labels.

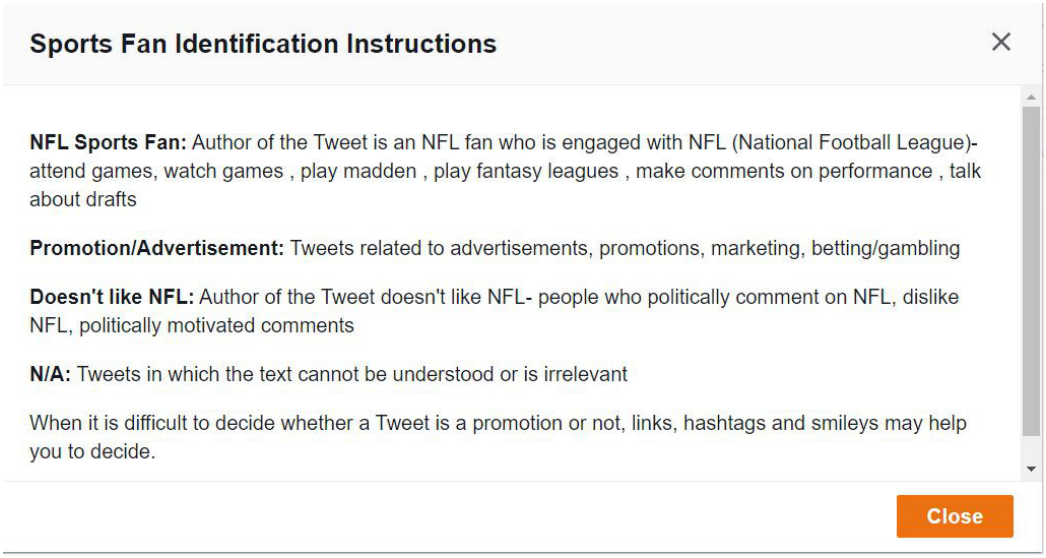

Figure 5.5: MTurk project–annotation instructions.

Finally, we moved into the last phase (Phase 3) of the annotation process to complete the remaining annotations with larger mini-batches compared to the mini-batches used in Phase 2.

5.4.1 Discussion

This section discusses the effectiveness of the proposed framework in developing a training dataset using social media text data to train a classification model. We note that the magnitude of the cost-saving using the proposed annotation framework is significant, even while developing a small training dataset. Further, we focus on the annotation quality improvements we achieved using the proposed framework.

Table 5.6: Dataset annotation summary.

Mini-batch Submission			Reconciliation		MTurk Annotation Cost (US$)			Tweets Distribution	
Phase	Round	Batch Size	Reconciled	Disputed	Initial	Reconciliation	Total	Sports Fan	Other
1	1	450	144	306	45	15.3	60.3	142	308
1	2	450	207	243	45	12.15	57.15	159	291
1	3	450	279	171	45	8.55	53.55	139	311
2	1	1000	639	361	100	18.05	118.05	429	571
2	2	1000	612	388	100	19.4	119.4	437	563
3	1	2500	1803	697	250	34.85	284.85	1174	1326
3	2	2500	1821	679	250	33.95	283.95	1139	1361
3	3	3000	2135	865	300	43.25	343.25	1389	1611
							1320.5	5008	6342

Annotation schema evolution. Table 5.3 summarises the evolution of the annotation schema during Phase 1 of the annotation process. We revised the annotation schema twice to obtain a better version (Version 3) based on the post-analysis of each mini-batch annotated in Phase 1. Table 5.4 shows the improvement in the annotation quality due to the proposed revisions. Introducing a more use case-specific annotation scheme (v2) helped to improve the quality of annotation. Further, as proposed in the framework, introducing additional labels, such as 'promotion', to the annotation schema boosted the annotation quality. This technique helped to direct the attention towards important aspects, thereby improving the annotation quality.

Cost-effectiveness analysis. Considering the distribution of the raw dataset (only 11% sports fan Tweets), to generate a dataset with approximately 5,000 sports fans (identifying sports fans is the key business requirement), we need to annotate almost the complete raw dataset (45,508 Tweets). Further, a minimum of three annotations is required to maintain the same annotation quality. Thus, the total annotation cost is close to US$6,800. Although this general approach allows us to obtain a large number of annotated Tweets, the usefulness of the training dataset is poor due to its highly skewed class distribution with only a small percentage of relevant data.

Table 5.6 shows the mini-batch submissions (including dispute resolution submissions) and the associated MTurk cost (based on the batch size, the number of annotations and the number of disputed annotations) to annotate the dataset using the proposed framework. The total cost to generate a training dataset with approximately 5,000 sports fan Tweets (and 6,342 other Tweets) was around US$1,320. The cost-saving was more than 80%. The cost-saving might vary based on the use case and original data distribution, yet it is evident that the proposed annotation framework leads to significant cost reductions.

In addition, the training dataset development task is highly efficient and effective since we dealt mostly with data that is highly useful for a particular use case or a business requirement.

5.5 Summary

In this chapter, we have proposed a novel text annotation methodology to significantly reduce the time and cost of manually annotating the text data required to train NLP models in the real world. Applying this methodology in the real world showed cost-savings of over 80%. Apart from that, we presented an overarching framework consisting of stages and steps to improve the efficiency and effectiveness of social intelligence solutions using NLP in the real world.

In the following final chapter of this thesis, we summarise our findings and contributions made in this dissertation and present the scope for future work.

Chapter 6

Conclusion

NLP focuses on extracting a comprehensive meaning representation from natural text. Using NLP for social intelligence is a popular strategy adopted by many organisations due to the sheer volumes of publicly available text data on various social media platforms, which can be used to gain a competitive advantage over business rivals. Recent advancements in the NLP domain, such as Transformer-based pre-trained language models and constant advances in processing power, have boosted NLP adoption, providing performance levels acceptable to organisations across various NLP tasks. Nonetheless, NLP continues to be an emerging and evolving technology for real-world applications.

Despite the popularity of NLP techniques applied to social media data to provide social intelligence for business, the noisy and diverse nature of social media data has posed significant challenges. It is difficult to comprehend noisy and unstructured text compared to structured, grammatical sentences. Further, the diverse and fast-changing vocabulary used on social media makes it difficult for machine learning models to perform well in real-world environments. Further, creating the labelled data required for most of the NLP tasks is a time-consuming and costly task, and these manual text annotation activities must be repeated for new use cases to retrain the existing models. Thus, having insufficient labelled data to train NLP models significantly affects the performance of these solutions in real-world applications.

The industry needs to overcome these challenges to use NLP-based social intelligence

solutions effectively and efficiently to gain a competitive advantage over other businesses. Throughout this thesis, we have made several significant and novel contributions to overcome the challenges discussed above, which are summarised in Section 6.1.

6.1 Research Summary

In this thesis, we have made contributions to overcome the three challenges in using NLP for social intelligence solutions. These challenges are time-consuming and costly manual data labelling, low performance in NLP models due to data scarcity and difficulty in comprehending noisy sentences in a context-aware NLP approach. In Chapter 2, we presented a comprehensive overview covering the relevant literature.

Chapter 3 presented approaches to leverage context-aware word representations to improve the performance of NLP models in a low data regime. In particular, we proposed to transfer prior knowledge from pre-trained language models to mitigate the impact on model accuracy due to insufficient labelled data while using context-aware representations to improve overall model accuracy. Further, we presented a novel sentence embedding technique for noisy sentences, condensing highly relevant linguistic characteristics into the sentence representation. Moreover, we released a probing dataset that can serve as a benchmark dataset to study the linguistic characteristics of unstructured and noisy text.

Chapter 4 focused on text data augmentation, particularly MLM approaches. We first explored the necessity of text data augmentation in the context-aware NLP regime and reported the importance of generating sentences with a closer meaning to the original sentence. To this end, we extended the back-propagation technique, combining it with state-of-the-art sentence embeddings to propose a novel text augmentation technique suitable for noisy datasets. Next, we extensively analysed the limitations of existing text augmentation techniques, including MLM-based approaches. We then proposed a novel text augmentation technique, IMOSA, that improves the overall diversity of the original dataset by identifying the most suitable sentences to augment while generating an optimum number of quality synthetic sentences from a selected sentence. We evaluated IMOSA across

several NLP tasks where it outperforms the state-of-the-art algorithms.

Finally, Chapter 5 addressed the problem of manual text data annotation challenges faced by organisations while launching social intelligence solutions using NLP. To overcome these challenges, we proposed a novel framework that combines all our contributions for an effective and efficient end-to-end process to deploy NLP-based social intelligence solutions in the real world.

6.2 Research Findings

Over the course of this study, we have designed and developed multiple novel methods for various challenges in the context of social intelligence solutions using NLP. This section recapitulates how our methods have identified and addressed the identified challenges and further summarises our contributions and findings.

Sequential transfer learning addresses limited labelled data. State-of-the-art pre-trained language models, such as BERT, have shown superior performance across various NLP tasks mainly due to their ability to provide context-aware meaning to the words or tokens of a sentence. Apart from that, our experiments on sequential transfer learning, where we transfer prior knowledge from pre-trained language models to an NLP task under a low data regime, have contributed significantly towards improving the model accuracy (see Section 3.2).

Meaning-rich noisy text comprehension. To obtain meaning-rich sentence representations for the noisy text, we proposed a novel sentence embedding technique based on the linguistic characteristics, boosting the performance of downstream NLP tasks (see Section 3.3). Combining meaning-rich sentence representations with sequential transfer learning has produced significant improvements in context-aware NLP tasks, such as sentiment analysis.

Meaning-sensitive text augmentation. NLP models usually demand a high volume of training data due to the complexity of natural language. Apart from that, massive neural models with millions of trainable parameters, such as Transformer-based models, require more data to deliver the expected performance. Our initial experiments supported this claim and highlighted the importance of text data augmentation to boost performance. To this end, we extended the back-translation technique along with sentence embeddings to augment the training dataset with synthetic sentences only with a meaning closer to the original sentence (see Section 4.2). Further, the experiments confirmed the criticality of adding sentences with a closer meaning to the original sentence while adding diversity to the training dataset. Next, to overcome these challenges and the limitation in existing text augmentation methods, we proposed a novel text augmentation technique, IMOSA, which helps to generate new sentences to augment the training dataset (see Section 4.3). IMOSA focuses on maintaining label compatibility while introducing sufficient diversity to the training dataset through pruned masking. Further, the model adds synthetic sentences with a close meaning to the original sentences using the optimal substitution step. The proposed method outperforms a couple of state-of-the-art text augmentation techniques across multiple NLP tasks, proving its superiority.

Cost-effective data annotation framework. Creating labelled data efficiently and effectively is crucial for the success of any text-based social intelligence solution. The proposed data annotation methodology helps organisations expedite the text data annotation process through crowdsourcing while reducing the cost significantly through innovative techniques, such as zero-shot text classification (see Section 5.2). Further, the text annotation framework consists of multiple strategies to improve the quality of the annotations. This methodology helps eliminate the lack of labelled data that appears to prevent most organisations from using NLP solutions by enabling them to obtain quality annotations quickly and cost-effectively.

NLP for social intelligence. We presented a comprehensive framework to improve the efficiency and effectiveness of social intelligence solutions that rely on NLP models. The pro-

posed framework combines our research contributions, providing improvements in three areas: data annotation, text data augmentation and noisy text comprehension in the low data regime (see Section 5.3). The cost-effective text annotation methodology helps organisations obtain the required labelled data in a shorter time and at a lower cost than before. The novel text data augmentation technique helps address data scarcity by adding high-quality and label-compatible synthetic sentences to the training dataset. The proposed NLP model uses a sequential transfer learning technique to extract prior knowledge from pre-trained language models to reduce the impact of limited labelled data further. Apart from that, the context-aware meaning representations provided by the Transformer-based pre-trained language models combined with the novel noisy text sentence embedding technique boost the machine learning model's ability to comprehend the noisy text, improving the overall performance of the NLP task. These upgrades to the text-based social intelligence solutions will directly enhance the ROI, strengthening any business case using NLP solutions for their competitive advantage.

6.3 Future Directions

The research, carried out over a three-year period of PhD candidature, has covered key and vital research topics in linguistic comprehension and text augmentation with noisy and unstructured text, particularly in the low data regime. However, this area of research is recent and evolving with potential for further research. In this section, we will provide an outlook on potential future research avenues and possible key extensions to the research reported here.

As part of future research, it will be interesting to deep dive into the area of inductive transfer learning to explore the possibilities of using multitask learning, which will allow deep neural networks to learn from related tasks via sharing parameters with other networks. This technique might help to address the difficulties in comprehending social media content due to the diversity and brevity of those messages and posts. Thus, it is possible to study a multitask learning model with multiple decoders covering NLP tasks, such as text

inference, text similarity, named entity recognition and relation extraction.

In the context of social intelligence solutions using NLP, identifying entities in a social media post is vital to enhance the NLP model's ability to comprehend the meaning conveyed by the author. There is a scope to improve existing techniques to deliver improved performance in real-world use cases due to the brevity and noisy nature of social media text. Further, domain-specific knowledge can be embedded into the solution to accurately interpret the entities. For instance, for a use case in the sports domain, the NLP models can understand the sports-related entities, such as team names, players and stadiums. Thus, future work can focus on developing ontologies and knowledge graphs that can evolve over time to improve the overall performance of NLP models.

Finally, further improving the text augmentation technique can lead to improvements in the overall accuracy of the NLP models further, as this might help to reduce the model drift due to the evolving vocabulary on social media. To this end, the use of generative language models for text data augmentation can be considered.

References

[1] Bernd Hollerit, Mark Kröll, and Markus Strohmaier. Towards linking buyers and sellers: Detecting commercial intent on twitter. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 629–632, New York, NY, USA, 2013. ACM.

[2] Bahman Pedrood and Hemant Purohit. Mining help intent on twitter during disasters via transfer learning with sparse coding. In Robert Thomson, Christopher Dancy, Ayaz Hyder, and Halil Bisgin, editors, *Social, Cultural, and Behavioral Modeling*, pages 141–153, Cham, 2018. Springer International Publishing.

[3] Rahul Pandey, Hemant Purohit, Bonnie Stabile, and Aubrey Grant. Distributional semantics approach to detect intent in twitter conversations on sexual assaults. *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Dec 2018.

[4] Rizwana Irfan, Christine K. King, Daniel Grages, Sam Ewen, Samee U. Khan, Sajjad A. Madani, Joanna Kolodziej, Lizhe Wang, Dan Chen, Ammar Rayes, and et al. A survey on text mining in social networks. *The Knowledge Engineering Review*, 30(2):157–170, 2015.

[5] Keith Cortis and Brian Davis. Over a decade of social opinion mining: a systematic review. *Artificial Intelligence Review*, 54(7):4873–4965, Oct 2021.

[6] Chaima Messaoudi, Zahia Guessoum, and Lotfi Ben Romdhane. Opinion mining in online social media: a survey. *Social Network Analysis and Mining*, 12(1):25, Jan 2022.

[7] M Bates. Models of natural language understanding. *Proceedings of the National Academy of Sciences*, 92(22):9977–9982, 1995.

[8] J. R. Firth. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32, 1957.

[9] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.

[10] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Vol-*

ume 2, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct 2014. Association for Computational Linguistics.

[13] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[14] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[16] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3(3):210–229, jul 1959.

[17] Cauchy A. Méthode générale pour la résolution des systémes d' équations simultanées. *Comptes rendus de l'Académie des Sciences*, 25:536–538, 1847.

[18] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986.

[19] Sappadla Prateek Veeranna, Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN-16)*, Bruges, Belgium, April 2016. d-side publications.

[20] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921. Association for Computational Linguistics, 2019.

[21] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[22] Luis Martin-Domingo, Juan Carlos Martín, and Glen Mandsberg. Social media as a

resource for sentiment analysis of airport service quality (asq). *Journal of Air Transport Management*, 78:106–115, 2019.

[23] Sunil Kumar, Arpan Kumar Kar, and P. Vigneswara Ilavarasan. Applications of text mining in services management: A systematic literature review. *International Journal of Information Management Data Insights*, 1(1):100008, 2021.

[24] Melva Hermayanty Saragih and Abba Suganda Girsang. Sentiment analysis of customer engagement on social media in transport online. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 24–29, 2017.

[25] Vallikannu Ramanathan and T. Meyyappan. Twitter text mining for sentiment analysis on people's feedback about oman tourism. In *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–5, 2019.

[26] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 2021.

[27] Zulfadzli Drus and Haliyana Khalid. Sentiment analysis in social media and its application: Systematic literature review. *Procedia Computer Science*, 161:707–714, 2019. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.

[28] Mark Kröll and Markus Strohmaier. Analyzing human intentions in natural language text. In *Proceedings of the Fifth International Conference on Knowledge Capture*, K-CAP '09, pages 197–198, New York, NY, USA, 2009. ACM.

[29] Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. We know what you want to buy: A demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1935–1944, New York, NY, USA, 2014. Association for Computing Machinery.

[30] Malu Castellanos, Meichun Hsu, Umeshwar Dayal, Riddhiman Ghosh, Mohamed Dekhil, Carlos Ceja, Marcial Puchi, and Perla Ruiz. Intention insider: Discovering people's intentions in the social channel. In *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, page 614–617, New York, NY, USA, 2012. Association for Computing Machinery.

[31] Muhammad Roman, Abdul Shahid, Shafiullah Khan, Anis Koubaa, and Lisu Yu. Citation intent classification using word embedding. *IEEE Access*, 9:9982–9995, 2021.

[32] Azin Ashkan and Charles L.A. Clarke. Term-based commercial intent analysis. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 800–801, New York, NY, USA, 2009. ACM.

[33] Jinpeng Wang, Gao Cong, Wayne Xin Zhao, and Xiaoming Li. Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In

Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, pages 318–324. AAAI Press, 2015.

[34] Mohamed Hamroun and Mohamed Salah Gouider. A survey on intention analysis: successful approaches and open challenges. *Journal of Intelligent Information Systems*, 55(3):423–443, Dec 2020.

[35] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[36] Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. Language modelling as a multi-task problem. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2049–2060, Online, April 2021. Association for Computational Linguistics.

[37] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[38] Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

[39] Krutarth Patel and Cornelia Caragea. Exploiting position and contextual word embeddings for keyphrase extraction from scientific papers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1585–1591, Online, April 2021. Association for Computational Linguistics.

[40] Wenjuan Han, Bo Pang, and Ying Nian Wu. Robust transfer learning with pretrained language models through adapters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 854–861, Online, August 2021. Association for Computational Linguistics.

[41] Michael I. Jordan. Serial order: A parallel, distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.

[42] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179 – 211, 1990.

[43] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[44] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[46] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[47] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, Nov 1997.

[48] Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *ArXiv*, abs/1506.00019, 2015.

[49] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[50] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.

[51] Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 643–648, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[52] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[53] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6000–6010. Curran Associates, Inc., 2017.

[55] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[56] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1243–1252. JMLR.org, 2017.

[57] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan

Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2048–2057. JMLR.org, 2015.

[58] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas, November 2016. Association for Computational Linguistics.

[59] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA, 2017. Curran Associates Inc.

[61] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[62] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[63] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[64] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[65] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[66] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[67] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 222–229, New York, NY, USA, 1999. ACM.

[68] Richard E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, 1961.

[69] Wei Xu and Alex Rudnicky. Can artificial neural networks learn language models? In *INTERSPEECH*, 2000.

[70] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[71] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

[72] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

[73] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–813, 2014.

[74] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 97–105. JMLR.org, 2015.

[75] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*, 2018.

[76] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[77] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, oct 2010.

[78] Sebastian Ruder. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019.

[79] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.

[80] Dong Wang and Thomas Fang Zheng. Transfer learning for speech and language processing. *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1225–1237, 2015.

[81] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hier-

archical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[82] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[83] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press, 1989.

[84] James L. Mcclelland, Bruece L. Mcnaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.

[85] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128 – 135, 1999.

[86] Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7):512 – 534, 2016.

[87] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[88] Emma Strubell and Andrew McCallum. Syntax helps ELMo understand semantics: Is syntax still relevant in a deep neural architecture for SRL? In *Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP*, pages 19–27, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[89] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[90] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. Semantics-aware BERT for language understanding. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*, 2020.

[91] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

[92] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges,

L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[93] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics, 2019.

[94] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[95] Yoav Goldberg. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287, 2019.

[96] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[97] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32*, pages 3261–3275. Curran Associates, Inc., 2019.

[98] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, nov 2018. Association for Computational Linguistics.

[99] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Jul 2019.

[100] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, Jul 2018. Association for Computational Linguistics.

[101] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual ex-

planations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA, 2015. IEEE Computer Society.

[102] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[103] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. 2019.

[104] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[105] Lili Wang, Chongyang Gao, Jason Wei, Weicheng Ma, Ruibo Liu, and Soroush Vosoughi. An empirical survey of unsupervised text representation methods on Twitter data. In *Proceedings of the Sixth Workshop on Noisy User-generated Text*, pages 209–214, Online, November 2020. Association for Computational Linguistics.

[106] Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas, Nov 2016. Association for Computational Linguistics.

[107] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR Conference Track*, Toulan, France, 2017.

[108] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[109] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, Feb 2017.

[110] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[111] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.

[112] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal

Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, 11:625–660, March 2010.

[113] Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom M. Mitchell. Zero-shot learning with semantic output codes. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 1410–1418, USA, 2009. Curran Associates Inc.

[114] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly, 2017.

[115] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.

[116] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[117] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA, 2015. MIT Press.

[118] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA, 2015. MIT Press.

[119] William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[120] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[121] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics.

[122] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 4006–4015. JMLR.org, 2017.

[123] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference*

on Artificial Intelligence, AAAI'17, page 2852–2858. AAAI Press, 2017.

[124] Buddhika Kasthuriarachchy, Madhu Chetty, Gour Karmakar, and Darren Walls. Pre-trained language models with limited data for intent classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[125] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, 2018.

[126] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation. In João M. F. Rodrigues, Pedro J. S. Cardoso, Jânio Monteiro, Roberto Lam, Valeria V. Krzhizhanovskaya, Michael H. Lees, Jack J. Dongarra, and Peter M.A. Sloot, editors, *Computational Science – ICCS 2019*, pages 84–95, Cham, 2019. Springer International Publishing.

[127] Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online, November 2020. Association for Computational Linguistics.

[128] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[129] Yoav Goldberg. Assessing BERT's syntactic abilities. *CoRR*, abs/1901.05287, 2019.

[130] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. *CoRR*, abs/1906.04341, 2019.

[131] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias&variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[132] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *Proceedings of International Conference on Learning Representations*, 2021.

[133] Leon Derczynski, Diana Maynard, Niraj Aswani, and Kalina Bontcheva. Microblog-genre noise and impact on semantic annotation accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, page 21–30, New York, NY, USA, 2013. Association for Computing Machinery.

[134] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany, aug 2016. Association for Computational Linguistics.

[135] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[136] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[137] Ricardo Vilalta, Christophe Giraud-Carrier, Pavel Brazdil, and Carlos Soares. *Inductive Transfer*, pages 545–548. Springer US, Boston, MA, 2010.

[138] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 751–760, New York, NY, USA, 2010. ACM.

[139] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics.

[140] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc., 2015.

[141] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, Sep 2017. Association for Computational Linguistics.

[142] O. Coban and G. T. Ozyer. Word2vec and clustering based twitter sentiment analysis. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–5, 2018.

[143] Buddhika Kasthuriarachchy, Madhu Chetty, Gour Karmakr, and Darren Walls. Pretrained language models with limited data for intent classification. In *International Joint Conference on Neural Network (IJCNN)*, 2020.

[144] Marcos Grzeça, Karin Becker, and Renata Galante. Drink2vec: Improving the classification of alcohol-related tweets using distributional semantics and external contextual enrichment. *Information Processing & Management*, 57(6):102369, 2020.

[145] Jonathas G.D. Harb, Régis Ebeling, and Karin Becker. A framework to analyze the emotional reactions to mass violent events on twitter and influential factors. *Information Processing & Management*, 57(6):102372, 2020.

[146] Zhancheng Ren, Qiang Shen, Xiaolei Diao, and Hao Xu. A sentiment-aware deep learning approach for personality detection from text. *Information Processing & Management*, 58(3):102532, 2021.

[147] José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Transformer based contextualization of pre-trained word embeddings for irony detection in twitter. *Information Processing & Management*, 57(4):102262, 2020.

[148] Patrick Jacob and Alexandra Uitdenbogerd. Readability of Twitter tweets for second language learners. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 19–27, Sydney, Australia, Dec 2019. Australasian Language Technology Association.

[149] Arnout B. Boot, Erik Tjong Kim Sang, Katinka Dijkstra, and Rolf A. Zwaan. How character limit affects language usage in tweets. *Humanities and Social Sciences Communications*, 5(76), 2019.

[150] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[151] XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63:1872–1897, 2020.

[152] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, Nov 2019. Association for Computational Linguistics.

[153] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[154] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[155] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*,

pages 423–430, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[156] Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012, Doha, Qatar, October 2014. Association for Computational Linguistics.

[157] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[158] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics*, pages 194–206, Cham, 2019. Springer International Publishing.

[159] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, 2014.

[160] Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. Parsing tweets into Universal Dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 965–975, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[161] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[162] Alexis Conneau and Douwe Kiela. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[163] Steffen Eger, Andreas Rücklé, and Iryna Gurevych. Pitfalls in the evaluation of sentence embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 55–60, Florence, Italy, August 2019. Association for Computational Linguistics.

[164] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, Jul 2019.

[165] Donald Allan Darling. The kolmogorov-smirnov, cramer-von mises tests. *The Annals of Mathematical Statistics*, 28(4):823–838, 1957.

[166] Linqing Shi, Danyang Liu, Gongshen Liu, and Kui Meng. Aug-bert: An efficient data augmentation algorithm for text classification. In Qilian Liang, Wei Wang, Xin Liu, Zhenyu Na, Min Jia, and Baoju Zhang, editors, *Communications, Signal Processing, and Systems*, pages 2191–2198, Singapore, 2020. Springer Singapore.

[167] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[168] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[169] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain, July 2004.

[170] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

[171] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4):427–437, 2009.

[172] Daniel Zeng, Hsinchun Chen, Robert Lusch, and Shu-Hsing Li. Social media analytics and intelligence. *IEEE Intelligent Systems*, 25(6):13–16, 2010.

[173] Daisuke Sakamoto, Naoki Matsushita, Mitsumasa Noda, and Kazuhiko Tsuda. Social listening system using sentiment classification for discovery support of hot topics. *Procedia Computer Science*, 126:1526–1533, 2018. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.

[174] Yogesh K. Dwivedi, Elvira Ismagilova, D. Laurie Hughes, Jamie Carlson, Raffaele Filieri, Jenna Jacobson, Varsha Jain, Heikki Karjaluoto, Hajer Kefi, Anjala S. Krishen, Vikram Kumar, Mohammad M. Rahman, Ramakrishnan Raman, Philipp A. Rauschnabel, Jennifer Rowley, Jari Salo, Gina A. Tran, and Yichuan Wang. Setting the future of digital and social media marketing research: Perspectives and research propositions. *International Journal of Information Management*, 59:102168, 2021.

[175] Zheye Wang and Xinyue Ye. Social media analytics for natural disaster management. *International Journal of Geographical Information Science*, 32(1):49–72, 2018.

[176] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October 2008. Association for Computational Lin-

guistics.

[177] Symeon Symeonidis, Dimitrios Effrosynidis, and Avi Arampatzis. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications*, 110:298 – 310, 2018.

[178] Dimitrios Effrosynidis, Symeon Symeonidis, and Avi Arampatzis. A comparison of pre-processing techniques for twitter sentiment analysis. In Jaap Kamps, Giannis Tsakonas, Yannis Manolopoulos, Lazaros Iliadis, and Ioannis Karydis, editors, *Research and Advanced Technology for Digital Libraries*, pages 394–406, Cham, 2017. Springer International Publishing.

[179] Daniel Hládek, Ján Staš, and Matúš Pleva. Survey of automatic spelling correction. *Electronics*, 9(10), 2020.

[180] Toby Segaran and Jeff Hammerbacher, editors. *Beautiful Data: The Stories Behind Elegant Data Solutions*. O'Reilly, Beijing, 2009.

[181] B. Kasthuriarachchy, M. Chetty, G. Karmakar, and D. Walls. Pre-trained language models with limited data for intent classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[182] Chris Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, page 286–295, USA, 2009. Association for Computational Linguistics.

[183] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August 2017. Association for Computational Linguistics.