Anticipating Hazards in Machine Translations of Public Health Resources via Advanced Text Classification Pipelines

A THESIS SUBMITTED TO
THE FACULTY OF ENGINEERING
OF THE UNIVERSITY OF SYDNEY
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF PHILOSOPHY

THE UNIVERSITY OF
SYDNEY

YIXIONG DING

Supervisor: Associate Professor Chang Xu

School of Computer Science

Faculty of Engineering

The University of Sydney

Australia

2024

Authorship Attribution Statement

This thesis does not contain any material that has been previously published elsewhere, and all the content is original and created by me.

Student Name: Yixiong Ding

Date: 2024

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Supervisor Name: Chang Xu

Date: 2024

Anticipating Hazards in Machine Translations of Public Health Resources via Advanced Text Classification Pipelines

Yixiong Ding (Email: ydin3496@uni.sydney.edu.au)

Supervisor: Associate Professor Chang Xu

School of Computer Science

Faculty of Engineering

The University of Sydney

Copyright in Relation to This Thesis

Statement of Original Authorship

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes. I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Name: Yixiong Ding

I dedicate this thesis to my parents, my wife, and my dog, in appreciation for their various forms of support during my Master's studies. I also wish to express my deepest gratitude to my mentor, Associate Professor Chang Xu, for his academic guidance throughout my studies, and more importantly, his invaluable advice on life.

Abstract

Public health educational resources developed by health institutions aim for high accessibility of information. The translation of these resources, which provide the public with a basic understanding of health risks and diseases, is commonly conducted by professional translators to cater to diverse linguistic and cultural backgrounds. In recent years, the global advancement of information technology has broadened the use of Machine Translation (MT) in online health education and promotion. MT tools such as Google Translate, DeepL, and ChatGPT have significantly improved performance, yet they face challenges posed by the language complexity, content complexity, and formality of professional medical resources.

In this study, we leverage Natural Language Processing (NLP) and Machine Learning (ML) tools to harness the power of text classification, a vital task that assigns text to one or more predefined categories. Our goal is to develop machine learning classifiers within our newly proposed Multi-Dimensional Text Classification Pipeline (MD-TCP) framework. Serving as a risk-prevention mechanism, this approach assists medical professionals with limited knowledge of the patient's language and helps patients who wish to self-navigate. Our model, which operates within the MD-TCP framework, predicts the likelihood of clinically significant mistakes or incomprehensible machine translation outputs based on the features of English source information as input to the machine translation systems.

MD-TCP is a new, comprehensive pipeline for data mining and feature extraction that we developed to achieve this goal. The pipeline has demonstrated significant improvements in both of our datasets. Regarding Accuracy, AUC, Sensitivity, Precision, and Specificity, our method improved by 24% - 33% compared to baseline methods. This underscores the potential of machine learning, mainly when implemented through MD-TCP, in predicting translation errors, thereby ensuring more accurate and understandable translations for health resources across diverse populations.

Keywords

Natural Language Processing, Machine Learning, Data Mining, Text Features Extraction, Text Classification, Public Health Education and Promotion.

Acknowledgements

I am deeply indebted to the numerous individuals who have steadfastly supported, guided, and encouraged me throughout my academic journey. The completion of this thesis could only have been achieved through their unwavering and persistent assistance.

My heartfelt gratitude extends to my supervisor, Professor Chang Xu, for his exceptional mentorship, patience, and guidance. His comprehensive expertise and enthusiasm for research have consistently served as a beacon of inspiration and motivation. I genuinely appreciate the countless hours he has dedicated to scrutinizing my work, offering constructive criticism, and assisting me in navigating the complex maze of academic research.

Furthermore, I am profoundly grateful to my parents and wife for their unconditional love, support, and understanding. Their unwavering belief in my abilities and encouragement during challenging periods have bolstered my resilience. I also extend a special acknowledgment to my faithful canine companion and my delightful feline friend, their vibrant spirits have continuously provided a source of comfort and joy.

I also want to extend my appreciation to my colleagues and peers in the AI lab in Sydney. Their camaraderie, stimulating discussions and invaluable insights have been a significant part of this journey. Their friendship and support have made this endeavour far more enjoyable and rewarding.

To conclude, I express my deepest gratitude to everyone who has contributed to the successful completion of this thesis. This journey would not have been as enriching without these exceptional individuals' unwavering support, guidance, and encouragement. Their invaluable contributions have left a lasting impact, for which I remain eternally grateful.

Table of Contents

CHAPTER 1

Introduction

Public health discourse represents a multifaceted landscape encompassing clinical, research, and educational domains [1]. Clinical resources, characterized by lexical and syntactic nuances introduced by medical professionals into medical records, stand in contrast to research materials known for their linguistic complexity, formal tone, and content intricacy [2]. Both these genres pose formidable challenges for machine translation (MT) technologies, which often struggle with linguistic intricacies and idiosyncrasies [3].

In sharp contrast to these specialized discourses, public health educational resources, curated by health authorities, are meticulously crafted to ensure high information accessibility. These resources serve the vital purpose of informing, guiding, and supporting the general public in acquiring fundamental knowledge about health risks and diseases [2]. Recognizing their crucial societal role, health authorities make extensive efforts to translate original English health resources into multiple languages, benefiting not only English-speaking populations but also diverse linguistic and cultural communities [4].

In recent years, educational resources in the public health sphere, especially those disseminated by the World Health Organization (WTO) and similar entities, exhibit fewer lexical and syntactic irregularities when compared to clinical documents. Furthermore, they maintain a controlled level of language complexity compared to research or policy materials.

Machine learning (ML) and data mining, recognized for their instrumental roles in various systems and applications, from recommendation systems to medical diagnostics, are celebrated for their prowess in handling voluminous data and extracting valuable patterns [5].

However, they present considerable hurdles when tasked with complex and irregular data, such as text.

Specifically, text data, especially in machine translation, is notorious for its irregularities and complexities [6]. Despite the substantial advancements in machine translation technologies, they remain prone to errors, particularly when confronted with specialized or intricate texts. This creates significant obstacles for non-native English speakers dependent on machine translation for information access [7].

In response to these challenges, we crafted a comprehensive machine learning approach, amalgamating several techniques and tools. We leveraged text analysis softwares and USAS (University of Lancaster Semantic Annotation System)[8] for feature extraction from text data and employed data augmentation techniques to bolster our models' robustness. We then transformed the text data into feature vectors combining TF-IDF features for a more practical application of machine learning algorithms.

Centring our efforts around Natural Language Processing (NLP) and Machine Learning, our objective was to harness the potential of text classification—a critical task of assigning text to predefined categories—to develop machine learning classifiers for machine translation error detection. This approach is a risk mitigation strategy for machine translation tool users, like aiding medical professionals with limited patient language knowledge and patients desiring self-navigation. Our proposed model estimates the chances of clinically significant errors or incomprehensible machine translation outputs, using features of English source information as input to machine translation systems.

We introduced a novel data mining and feature extraction pipeline - Multi-Dimensional Text Classification Pipeline (MD-TCP) and applied it to two distinct datasets in the public health discourse area - COVID-19 and Maori Cancer. This approach led to substantial improvements in both datasets, with our methodology outperforming baseline methods by a remarkable 24% to 33% in critical metrics such as Accuracy, AUC, Sensitivity, Precision, and Specificity. The significant enhancement in machine learning classifier accuracy we observed attests to the effectiveness of our approach. Our study not only underscores the potential of machine

learning in preempting translation errors, thereby ensuring more accurate and comprehensible translations for health resources across diverse demographics, but it also aspires to enrich the machine learning and data mining landscape by providing a comprehensive methodology to handle complex text data and demonstrating its efficacy in machine translation.

CHAPTER 2

Literature review

2.0.1 Machine Translation

The rapid development of computer science and the internet has led to the widespread popularity of machine translation software. However, the reliability of machine translation is only sometimes guaranteed. Irvine et al. [6] emphasized the challenges of machine translation, mainly when applied to new domains. While there have been substantial improvements in the field, the authors note that complex linguistic phenomena still need to be solved, often leading to mistranslations and misunderstandings. This is critical in domains such as healthcare and legal affairs, where precise communication is of utmost importance.

Within the realm of healthcare, the implications of inaccuracies in machine translation can be severe. Ji et al. [7] discussed the potential risks when using machine translation tools in healthcare settings. They highlight that errors can cause significant misinterpretations of clinical symptoms, potentially harming patients who depend on translated information for healthcare decisions.

In response to these challenges, Natural Language Processing (NLP) advancements have been leveraged. Prabhakar and Won introduced a novel approach to medical text classification by employing deep learning models. The authors put forth two novel deep-learning architectures for medical text classification, demonstrating promising results in terms of classification accuracy.

Further advancements in machine translation have been seen by incorporating deep learning technologies. Sun Y. [9], in his study titled "Analysis of Chinese Machine Translation Training

Based on Deep Learning Technology", illustrated the efficacy of deep learning models in Chinese-English translation tasks, emphasizing the critical role of large-scale parallel corpora and attention mechanisms in enhancing translation quality.

An exploration of Simple Recurrent Neural Networks (SRNN) in English lexical analysis, a vital part of machine translation, is presented in the "English Lexical Analysis System of Machine Translation Based on Simple Recurrent Neural Network". by Zhu J. [10]. The study demonstrates the significance of SRNN in capturing sequential information of English sentences, thereby improving the accuracy of machine translation systems.

Furthermore, Ren B. [11] discussed the application of the SCN-LSTM (Skip Convolutional Network and Long Short Term Memory) translation model in teaching translation. His study found that this neural network outperforms the traditional N-tuple translation model in terms of translation quality and teaching effect.

In summary, while machine translation software has seen significant improvements, it still confronts challenges in terms of reliability, particularly in critical fields like healthcare. Nonetheless, the rapid progression of NLP and the integration of machine learning and deep learning methods provide promising solutions. The transformative potential of these technologies in reshaping translation teaching methodologies also presents exciting prospects for the future. As these technologies evolve, they are anticipated to play an increasingly central role in developing more sophisticated and accurate machine translation systems. However, their application in critical fields like healthcare must be cautiously approached. The development of machine learning classifiers as a risk-prevention mechanism, as discussed by Ji et al., is of paramount importance. [6]

2.0.2 Natural Language Processing and Artificial Intelligence

Natural Language Processing (NLP) [12, 13] has emerged as a central field in computational linguistics, offering an array of techniques to understand, interpret, and coherently generate human language. Its applications span diverse areas, such as machine translation, sentiment analysis, and text summarization [14]. The effectiveness of NLP is closely linked with the

advancements in Artificial Intelligence (AI) [15, 16], a broader field that strives to emulate human intelligence in machines.

With its numerous subfields, AI has paved the way for a more nuanced understanding and handling of human language. This is particularly evident with the advent of Machine Learning (ML) [17], an AI methodology that has significantly transformed our approach to NLP tasks. Machine Learning employs algorithms capable of learning patterns from data, leading to systems that can adapt, learn, and improve over time [18]. This is especially important in NLP, where ML methods have enabled tasks such as part-of-speech tagging, named entity recognition, and text classification [19].

In NLP, ML techniques such as Naive Bayes, Support Vector Machines, and Decision Trees have been traditionally used to classify text and make predictions [20]. More advanced techniques like Random Forests and Gradient Boosting Decision Tree[21] have further enhanced the capabilities of ML in NLP, enabling more accurate and efficient text analysis.

Deep Learning[22, 23], a subset of ML, takes the concept further by implementing neural networks that mimic the human brain's structure and function. These neural networks can model complex patterns in large datasets, significantly improving NLP applications. Convolutional Neural Networks (CNNs) [24], Recurrent Neural Networks (RNNs)[25], and, more recently, Transformer-based models like BERT [26] and GPT [27] have taken centre stage in the NLP field. These models have demonstrated superior performance in language understanding tasks by leveraging attention and context prediction mechanisms.

The integration of AI, mainly through ML and DL, has revolutionized the field of NLP. We can anticipate more sophisticated language processing capabilities as these AI-driven methods evolve and improve. This progress holds immense potential for developing more effective and contextually aware language-based AI systems, further pushing the boundaries of what we can achieve in NLP.

2.0.3 Text Classification

Text classification, as a fundamental task in Natural Language Processing (NLP), involves categorising text into predefined classes[28]. It is the foundation for many applications, such as spam filtering, sentiment analysis, and topic labelling, thus playing a critical role in deriving meaningful insights from raw text[29].

A particularly significant subset of text classification is Binary Classification, where the objective is categorising text into one of two classes [30]. This is commonly employed in scenarios like email spam detection (spam or not spam) and sentiment analysis (positive or negative). The simplicity of binary classification makes it a popular starting point for many text classification endeavours. However, it poses unique challenges in dealing with imbalanced datasets and distinguishing subtle differences in text.

Machine Learning (ML) has emerged as a powerful tool in text classification, offering a suite of algorithms capable of learning patterns in text data and using these learned patterns to categorise new, unseen text[28]. ML algorithms can be broadly divided into supervised and unsupervised learning. In the context of text classification, supervised learning is often used where a model is trained on a labelled dataset and then used to classify new, unlabeled data [31].

Traditional ML algorithms such as Naive Bayes, Decision Trees, and Support Vector Machines have been widely used for text classification tasks, including binary classification [32]. More recently, ensemble methods such as Random Forests and Gradient Boosting have been leveraged to improve prediction accuracy by combining multiple weaker models to form a stronger one [21].

Further, unsupervised ML techniques, like clustering, are employed for text classification when labelled data is scarce[33]. These methods discover natural groupings in data and can be particularly useful for exploratory data analysis.

The application of ML in text classification has not only automated the process but also increased the efficiency and accuracy of classification tasks. As ML continues to evolve

and improve, we can anticipate more sophisticated text classification models capable of understanding subtle language patterns and contexts, further pushing the boundaries of what we can achieve in text classification.

2.0.4 Text and Feature Mining

Data mining, an interdisciplinary field focused on extracting useful information from large datasets, has become a cornerstone in numerous disciplines and applications, ranging from business intelligence to healthcare [34]. This powerful technology thrives on its ability to manage substantial data and discover valuable patterns and insights. However, the complexity and irregularity inherent in certain types of data, such as text, pose significant challenges to traditional data mining techniques[5].

Text mining, a subset of data mining, specializes in extracting valuable information from unstructured textual data [35]. This form of mining has seen a growing demand due to the vast amounts of unstructured text data generated daily, such as social media posts, news articles, and scientific literature. Despite its complexity, text mining provides an opportunity to extract insights that can be pivotal for decision-making in various domains [36].

Text mining often involves using Natural Language Processing (NLP) techniques, which help transform unstructured text into structured data that can be analyzed. Feature mining, a crucial part of this process, involves identifying and extracting relevant characteristics or 'features' from the text data. This aspect of text mining is also commonly referred to as feature engineering [37].

In conventional NLP, feature mining involves various steps, including tokenization, stop word removal, stemming, and applying techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) and bag-of-words for feature extraction [38]. More advanced methods might incorporate part-of-speech tagging, named entity recognition, or semantic analysis using word embeddings like Word2Vec [39] or GloVe [40]. The selection of features and the techniques used for their extraction can significantly impact the performance of downstream tasks, such as text classification, sentiment analysis, or information retrieval[30].

In sum, the fields of data mining, text mining, and feature mining play an essential role in the extraction of valuable insights from large amounts of unstructured text data. By leveraging advanced NLP techniques for feature mining, it is possible to uncover more profound levels of understanding from the text, paving the way for more sophisticated applications and more informed decision-making processes.

CHAPTER 3

Methods

This chapter will mainly introduce how to construct and manually annotate our dataset. Subsequently, we will elucidate how linguistic tools are employed for text data mining, extracting latent features and information from the dataset. We then present using the USAS (University of Lancaster Semantic Annotation System)[8, 41] to extract features from the plain text data within the dataset, transforming it into a new dataset with multiple dimensions. Despite the increase in the number of features, the majority are likely noise for text classification tasks, necessitating the Recursive Feature Elimination (RFE) method to optimize the number of features, thereby enhancing the performance of text classification tasks. Ultimately, we discovered that the best performance is achieved by extracting features from the original data using the TF-IDF tokenizer, then mixing these extracted features with those extracted by USAS system. The mixed features are then vectorized using techniques such as Principal Component Analysis (PCA) [42]for dimensionality reduction, followed by further optimization using methods like RFE before being fed into the model for training. This process yields the best results for text classification tasks.

3.1 Data Collection and Annotation

We compiled two distinct datasets through manual browsing and searching from the website of the NCBI (Feb 2023) (https://www.ncbi.nlm.nih.gov/, accessed on 06 Feb 2023), the COVID-19 and Maori Cancer datasets. These datasets comprise sentences pertinent to their respective topics, sourced from various medical websites and translated into Chinese and

Maori. The translations were conducted using various translation software, leading to some translation errors.

As illustrated in Figure.3.1, these errors, such as incorrect translation of professional glossary, unclear expressions, and disordered syntax, could potentially lead to misunderstandings for non-native English speakers, encompassing doctors, nurses, and patients. We manually annotated the datasets, marking translations with overt errors as positive and those without errors as negative.

3.2 Text Analysis

Text analysis software encompasses many tools, leveraging natural language processing (NLP), machine learning, and additional computational methods for text data examination. These utilities can execute numerous operations, ranging from sentiment analysis, which determines whether a text conveys positive, negative, or neutral sentiment, to topic modelling, text classification, and named entity recognition, identifying entities such as people's names, locations, and organizations within a text.

Readability Studio (Oleander Software Ltd., Vandalia, OH, USA), meanwhile, is designed to gauge the reading complexity of a text. This type of software employs a variety of algorithms, assessing components such as sentence length, syllable count, and word familiarity to generate a score indicative of the text's readability. Some notable examples of these algorithms are the Flesch-Kincaid Reading Ease [43], Gunning Fog Index [44], and SOMG Index[45]. Such tools find extensive use in education and content creation, ensuring the created content aligns with the intended audience's reading proficiency.

Our study mainly deployed Readability Studio to discern differences between positive and negative samples. For example, positive samples typically exhibited greater text complexity than negative samples.

As evident from Table 3.1, a significant discrepancy exists between positive and negative data. Several reading level tests, including Flesch-Kincaid, Gunning Fog, and SMOG,

	Original Text	Taget Language Translation	Backward Translation	Error
Covid - 19	Patients are given medicine to make them sedated whilst they are on a ventilator.	患者带呼吸机通气时要进行药物镇静。	The patient is sedated while being ventilated by a ventilator.	Professional Glossary
Covid - 19	In 1000 people, 63 fewer would experience a serious unwanted effect compared to placebo or standard care.	与安慰剂或标准照护相比，瑞德西韦治疗使每 1000 人中有不良反应者多减少 63 名或更多。	Treatment with remdesivir resulted in 63 or more fewer adverse reactions per 1,000 people compared with placebo or standard of care.	Opposite Expression
Covid - 19	So, their usefulness may be limited to excluding COVID-19 infection rather than differentiating it from other causes of lung infection.	因此，其有用性可能限于排除 COVID-19 感染，而非将其与其他原因所致肺部感染区分开来。	Therefore, its usefulness may be limited to excluding COVID-19 infection rather than distinguishing it from other causes of pulmonary infection.	Unclear Expression
Covid - 19	Studies generally predicted or observed some benefit from screening at borders, however these varied widely.	研究通常预测或观察到边检筛查会带来一些好处，但是它们差别很大。	Studies often predict or observe some benefit from border screening, but they vary widely.	None
Maori Cancer	Cancers are caused by damage to these genes.	Na te kino o enei ira ka puta te mate pukupuku.	The cancer can be caused by the evil of these dots.	Wrong expression
Maori Cancer	Benign tumours are surrounded by a capsule and do not spread to other parts of the body.	Ko nga pukupuku ngawari ka karapotia e te kapene me te kore e horapa ki etahi atu waahanga o te tinana.	The tender squirrels are surrounded by the squirrel without spreading to other sanitary bodies.	Professional Glossary
Maori Cancer	This is diagnosed as cancer of unknown primary.	Ka kiia tenei he mate pukupuku o te tuatahi kaore e mohiotia.	This is called cancer of the first unknown.	Unclear Expression
Maori Cancer	The aim of the treatment is to control the cancer for as long as possible, improve any symptoms, and improve your quality of life.	Ko te whainga o te maimoatanga ko te whakahaere i te mate pukupuku mo te wa roa, te whakapai ake i nga tohu, me te whakapai ake i to oranga.	The aim of treatment is to manage long-term cancer, improve the indicators, and improve your life.	None

FIGURE 3.1. Translation errors for Covid-19 and Maori Cancer Datasets.

	Positive	**Negative**
Flesch-Kincaid	Post-graduate freshman, first month of class	University senior, seventh month of class completed
Flesch Reading Ease	Text is very difficult to read	Text is very difficult to read
Gunning Fog	University senior, sixth month of class completed	University senior, first month of class
SMOG	Post-graduate freshman, seventh month of class completed	Post-graduate freshman, third month of class completed
Average grade level	17.1	16.8
Average reading age	22.3	22.1
Number of difficult sentences	47.8%	44%
Average sentence length	23.6	21.6
Average number of characters	5.5	5.6
Average number of syllables	1.9	1.9
Number of numerals	3.5%	3.4%
Number of proper nouns	4.5%	4.8%
Number of monosyllabic words	50.9%	49.4%
Number of unique monosyllabic words	6.676%	2.496%
Number of complex words	25.2%	26.4%
Number of unique complex words	8.965%	4.015%
Number of long words	45.6%	47.3%
Number of unique long words	14.725%	6.500%
Number of unique SMOG hard words	10.290%	4.501%
Number of unique Fog hard words	7.813%	3.491%

TABLE 3.1. Readability Studio analysis for Covid-19 dataset.

unambiguously demonstrate that positive data necessitates a higher level of English proficiency from the readers. This discrepancy becomes more conspicuous at the sentence and word levels. In positive data, the percentage of difficult sentences is 3.8% higher. At the same time, indicators for unique monosyllabic words, unique complex words, unique SMOG hard words, and unique long words exhibit even more significant differences of 4.18%, 4.95%, 5.789%, and 8.225%, respectively. This implies that positive data comprises rare, complex, and long words used only once, making it more susceptible to professional translation errors.

3.3 Feature Extraction

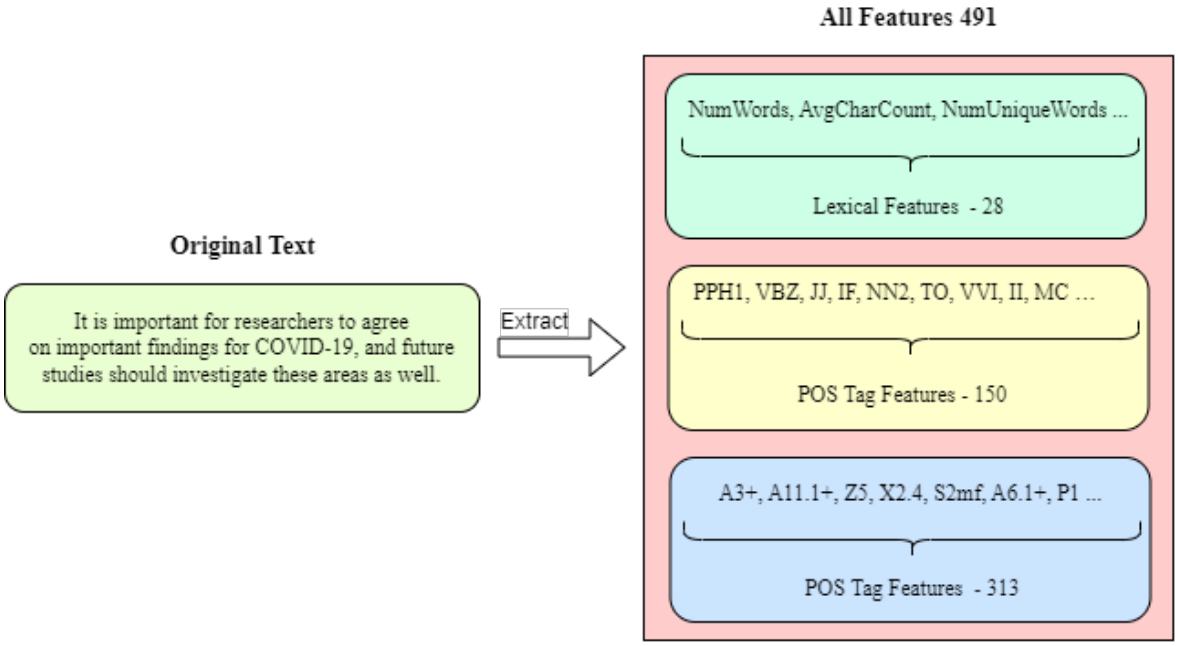

FIGURE 3.2. Extract different features from the original text input of Covid-19 dataset.

USAS (University of Lancaster Semantic Annotation System)[8, 41]is a system for automatic semantic tagging and semantic lexical resources. It assigns to words and phrases a tag from a detailed set of semantic field categories, which provides a rich level of semantic information.

To extract features from the original text, we utilized USAS to analyze all the text in the dataset and then extract the features. For instance, in the Covid-19 dataset, which consists of

1265 samples (619 positives and 646 negatives), we extracted 491 distinct features. These include 28 lexical features, 150 part-of-speech (POS) tag features, and 313 USAS tag features. For the Maori Cancer dataset, which comprises 991 samples (668 positives and 323 negatives), we extracted 458 features, including 30 lexical features, 143 POS tag features, and 285 USAS tag features.

Part-of-speech (POS) tagging [46] is labelling the words in a text with their appropriate part of speech (such as nouns, verbs, adjectives, etc.). It provides grammatical information about the words in a text.

USAS tags [41], on the other hand, provide semantic information. They categorize words into semantic fields or categories, providing a deeper level of understanding of the meaning of words in their context.

Both POS and USAS tags provide valuable information about the words in a text; however, they serve different purposes. For example, POS tags provide grammatical information, while USAS tags provide semantic information. Both types of information can be crucial in tasks such as text analysis and machine learning.

3.4 Machine Learning Algorithms

This section will introduce machine learning algorithms we applied during the experiments.

3.4.1 Naive Bayes

The Naive Bayes classifier is a probabilistic machine learning model used for classification tasks. The classifier is based on the Bayes theorem, which describes the probability of an event based on prior knowledge of conditions that might be related to the event[47].

In the context of text classification, our event is a particular class (or category) we want to assign a given text, and the conditions are the features of the text, which are usually the words it contains.

Given a text t and a class c, the classifier calculates the posterior probability $P(c|t)$ of the text t belonging to the class c. According to the Bayes theorem, this can be computed as:

$$P(c|t) = \frac{P(t|c) \cdot P(c)}{P(t)} \tag{3.1}$$

Where $P(c|t)$ is the posterior probability of class c given a text t. $P(t|c)$is the likelihood which is the probability of text t given class c. $P(c)$ is the prior probability of class c. The prior probability is the proportion of text of class c in the training set. $P(t)$ is the prior probability of text t.

In practice, since $P(t)$ is constant for all classes, we only need to compute $P(t|c) * P(c)$ for each class and choose the class with the highest value.

Assuming that the features in the document are independent (the 'naive' assumption), $P(t|c)$ can be calculated as the product of the probabilities of each feature in the text given the class:

$$P(t|c) = P(f_1|c) \cdot P(f_2|c) \cdot \ldots \cdot P(f_n|c) \tag{3.2}$$

Where $P(fi|c)$ is the probability of feature fi occurring in the text of class c. The frequency of the features in the documents of the training set can estimate these probabilities.

In practice, to avoid numerical underflow from the multiplication of many probabilities, we usually work in log space:

$$\log P(t|c) = \log P(f_1|c) + \log P(f_2|c) + \ldots + \log P(f_n|c) \tag{3.3}$$

Additionally, a smoothing parameter is usually introduced to handle features that did not appear in the training set. A common technique is Laplace smoothing (or add-one smoothing), where one is added to the numerator, and the size of the feature is set to the denominator [48].

Naive Bayes is a simple yet powerful algorithm for text classification. Despite its simplicity and the naive feature independence assumption, it works well in many real-world situations, particularly in text classification tasks where the dataset is large and high-dimensional.

3.4.2 Support Vector Machines

Support Vector Machines (SVM) is a supervised learning model for classification and regression tasks. It is particularly effective in high-dimensional spaces, making it well-suited for text classification, where the data often has a large number of features [49].

The basic idea behind SVM is to find a hyperplane that best divides the data into different classes. In a binary classification problem, SVM tries to find the optimal hyperplane that maximizes the margin between the two classes. Given a training dataset of instance-label pairs (x, y), where $x \in X$ is the feature vector (in our case, the text document represented as a vector), and $y \in Y$ is the class label (which in a binary classification problem can be -1 or 1), the goal of SVM is to find the function $f(x)$ that can predict the class label y for any input x.

The decision function of the SVM can be written as:

$$f(x) = \langle w \cdot x \rangle + b \tag{3.4}$$

Where w is the weight vector. x is the feature vector. b is the bias term. $\langle w \cdot x \rangle$ is the dot product of w and x. The equation $f(x) = 0$ defines the hyperplane that separates the data. The decision boundary is chosen to be the hyperplane that maximizes the margin, which is the distance between the nearest points of the two classes.

In order to handle non-linearly separable data, SVM uses a technique known as the 'kernel trick' [50]. The kernel trick maps the input features into a higher-dimensional space where a linear separation is possible. The decision function with the kernel trick becomes:

$$f(x) = \sum_i \alpha_i y_i K(x_i, x) + b \tag{3.5}$$

Where: α_i are the Lagrange multipliers obtained from the solution of the dual problem. y_i are the class labels. $K(x_i, x)$ is the kernel function. b is the bias term. There are several types of kernel functions, such as linear, polynomial, and radial basis function (RBF) [50]. The choice of the kernel function depends on the nature of the data.

The most common way to create feature vectors in text classification is using the bag of words or TF-IDF methods [38]. These methods transform the text into a vector where each dimension corresponds to a specific word or term in the text.

In conclusion, SVM is a robust algorithm for text classification. Thanks to the kernel trick, its ability to handle high-dimensional data and find complex decision boundaries make it particularly suitable for this task.

3.4.3 Logistic Regression

Logistic Regression is a statistical model that, in its basic form, uses a logistic function to model a binary dependent variable. It is widely used for binary classification problems and can be extended to multi-class classification problems [51].

In text classification, a document is represented as a vector of features, where each feature could be the presence or absence of a word, the frequency of a word, or even more complex features [31]. Given a text document represented as a feature vector $\mathbf{x}$, the logistic regression model computes the probability of the document belonging to the positive class (denoted as $y = 1$) using the logistic function applied to a linear combination of the features:

$$P(y = 1|\mathbf{x}; \theta) = \frac{1}{1 + e^{-(\theta^T \mathbf{x} + \theta_0)}} \tag{3.6}$$

Where $\mathbf{x}$ is the feature vector. θ is the weight vector. θ_0 is the bias term. $\theta^T \mathbf{x}$ is the dot product of θ and $\mathbf{x}$.

The goal of logistic regression is to find the parameters θ and θ_0 that maximize the likelihood of the observed data, which can be written as:

$$\mathcal{L}(\theta, \theta_0) = \prod_{i=1}^{N} P(y_i|\mathbf{x}i; \theta) = \prod i = 1^N \left[P(y = 1|\mathbf{x}_i; \theta)\right]^{y_i} \left[1 - P(y = 1|\mathbf{x}_i; \theta)\right]^{1-y_i} \quad (3.7)$$

To make the computation easier, we usually take the log of the likelihood function, which turns the product into a sum:

$$\log \mathcal{L}(\theta, \theta_0) = \sum_{i=1}^{N} y_i \log P(y = 1|\mathbf{x}_i; \theta) + (1 - y_i) \log \left[1 - P(y = 1|\mathbf{x}_i; \theta)\right] \quad (3.8)$$

These parameters are usually found using optimization algorithms such as gradient descent or more advanced methods like L-BFGS or Newton's [52].

Logistic regression is a powerful and flexible model for text classification. It can handle both linear and non-linear classification problems depending on the features used. It also provides interpretable parameters and can return calibrated probabilities, which can help rank instead of just classification.

3.4.4 Gradient Boosting Decision Tree

Gradient Boosting Decision Trees (GBDT) is a powerful and widely-used machine learning algorithm that produces a prediction model as an ensemble of decision trees. It builds the model in a stage-wise fashion, and it generalizes them by allowing optimization of an arbitrary differentiable loss function [21].

In text classification, documents are typically represented as vectors of features, where each feature could correspond to a word or a set of words in the document. The goal is to predict a target variable, which could be the category of the document [31].

Given a set of n training examples $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where x_i represents the feature vector of the i-th document and y_i is the corresponding label, the GBDT algorithm

starts by initializing the model with a constant value:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3.9}$$

Where $L(y_i, \gamma)$ is the loss function evaluated at the true label y_i and the prediction γ.

Then, for each stage $m = 1$ to M, it performs the following steps:

(1) Compute the pseudo-residuals:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right] F(x) = Fm - 1(x) \quad \text{for } i = 1, ..., n \tag{3.10}$$

These residuals represent the direction to adjust the predictions to minimize the loss. Fit a decision tree to the pseudo-residuals, resulting in terminal regions R_{jm}, for $j = 1, ..., J_m$.

(2) For each terminal region, compute the output value that minimizes the loss:

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{3.11}$$

(3) Update the model:

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \tag{3.12}$$

Where: $I(x \in R_{jm})$ is an indicator function that is 1 if x is in the region R_{jm} and 0 otherwise. ν is the learning rate or shrinkage parameter.

(4) The final model is then given by:

$$F_M(x) = F_0(x) + \sum_{m=1}^{M} \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \tag{3.13}$$

Gradient Boosting Decision Trees is a powerful algorithm for text classification. It has the flexibility of defining an arbitrary differentiable loss function, making it suitable for various problems. It also performs well with high-dimensional data and can handle numeric and categorical data.

3.4.5 Stacking Classifier

Stacking is an ensemble learning technique that combines multiple classification models via a meta-classifier. The base-level models are trained based on a complete training set, and then the meta-model is fitted based on the outputs, or the "meta-features", of the base-level models [53].

For text classification, we often represent documents as vectors of features, where each feature corresponds to a word or a group of words in the document. Given a training set of n text documents represented by their feature vectors $\mathbf{x}_i$ and their labels y_i, $i = 1, ..., n$, we first train our base level models. In our case, we have a Multinomial Naive Bayes classifier and an SVM classifier.

The Multinomial Naive Bayes classifier computes the posterior probability of a document belonging to a class using Bayes' theorem, assuming the features are conditionally independent given the class. The probability of a document $\mathbf{x}$ belonging to class c can be computed as:

$$P(c|\mathbf{x}) = \frac{P(c)\prod_{i=1}^{d} P(x_i|c)}{P(\mathbf{x})} \tag{3.14}$$

The SVM classifier computes the class of a document using a hyperplane that separates the documents of different classes in the feature space:

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \tag{3.15}$$

Once we have trained the base classifiers, we use them to predict the class of the documents in the training set, which gives us a new set of features. If we denote the prediction of the Naive Bayes classifier as $NB(\mathbf{x})$ and the prediction of the SVM classifier as $SVM(\mathbf{x})$, our new feature vector for each document is $(NB(\mathbf{x}), SVM(\mathbf{x}))$.

We then train our meta-classifier, a Logistic Regression classifier in our case, on these new features. The Logistic Regression model computes the probability of a document belonging

to the positive class using the logistic function:

$$P(y = 1|\mathbf{x}; \theta) = \frac{1}{1 + e^{-(\theta^T \mathbf{x} + \theta_0)}} \tag{3.16}$$

Therefore, given a new document, we first compute the features in the final model using the base classifiers. Then we use the Logistic Regression classifier to predict the class of the document based on these features.

Stacking is a robust algorithm for text classification that can combine the strengths of different classifiers to improve performance. The choice of the base classifiers and the meta-classifier can be tailored to the specific problem and the available data.

3.5 Feature Optimization

Recursive Feature Elimination (RFE) is a feature selection method that fits a model and removes the weakest features (those with the least importance) until the specified number of features is reached [54]. It is a type of wrapper feature selection method, meaning it uses a machine learning algorithm and performance metric to evaluate the importance of each feature. This technique can be beneficial in improving the accuracy of machine learning models by eliminating irrelevant features, reducing overfitting, and improving model interpretability [55].

3.6 Multi-Dimensional Text Classification Pipeline

We present the "Multi-Dimensional Text Classification Pipeline (MD-TCP)", a novel and comprehensive multi-stage processing pipeline designed to significantly enhance our model's performance based on the characteristics of the features we have extracted. Emphasizing the pipeline's multi-dimensional and multi-step nature, MD-TCP integrates various strategies, which span from text preprocessing to sample balancing. Each stage of the pipeline has been carefully engineered to improve text classification's quality and performance, offering an advanced solution to complex data processing and machine learning challenges.

Algorithm 1: Multi-Dimensional Text Classification Pipeline (MD-TCP)

Data: Text datasets $\mathcal{D}$
Text Preprocessing: Tokenizing using the NLTK library and filtering out stop words.;
for *Each text* $T \in \mathcal{D}$ **do**
    Feature Extraction: Converting text data to TF-IDF and USAS features;
Dimensionality Reduction:Performing PCA dimensionality reduction on the feature matrix;
Feature Scaling: Encoding categorical features with One-Hot encoding and combining it with continuous features;
Feature Scaling: Scaling the combined features using MinMaxScaler;
Feature Selection:Using RFE for feature selection;
Dataset Splitting: Splitting the dataset into training and testing sets;
Sample Balancing:Using SMOTE to oversample the training set.

Here are the whole process of MD-TCP:

(1) **Text Preprocessing**: Tokenizing using the NLTK [56] library and filtering out stop words. NLTK is an efficient English tokenization library that can help reduce noise.

(2) **Feature Extraction & Fusion**: Converting text data into a TF-IDF matrix. TF-IDF is a common and effective method to transform text into numerical features. It can capture the importance of words in the text. Then TF-IDF features are combined with features that extracted by the USAS system.

(3) **Dimensionality Reduction**: Performing Principal Component Analysis (PCA) dimensionality reduction on the feature matrix. PCA is a method of dimensionality reduction that can help reduce the number of features, alleviate the curse of dimensionality, and speed up model training.

(4) **Feature Encoding**: Encoding categorical features with One-Hot encoding and combining it with continuous features. One-Hot encoding is a standard method to handle categorical features. Combining categorical and continuous features can help the model leverage both features simultaneously.

(5) **Feature Scaling**: Scaling the combined features using MinMaxScaler. Feature scaling can ensure all features are on the same scale, which is crucial to the performance of many machine learning models (such as Support Vector Machines, k-Nearest Neighbors, etc.).

(6) **Feature Selection**: Using Recursive Feature Elimination (RFE) for feature selection. Feature selection can help remove useless or redundant features, alleviate overfitting, and enhance the performance and interpretability of the model. RFE is a commonly used feature selection method.

(7) **Dataset Splitting**: Splitting the dataset into training and testing sets: This is a fundamental step in any machine learning project that can help us evaluate the model's performance on unseen data.

(8) **Sample Balancing**: Using the Synthetic Minority Over-sampling Technique (SMOTE) [57] to oversample the training set: Using SMOTE oversampling is a standard solution for imbalanced class issues.

The optimal strategies for data preprocessing and feature engineering may vary due to the dataset's characteristics, the task's requirements, and the models in use, among other factors. Therefore, the best strategies are often progressively optimized through experimentation and adjustment.

CHAPTER 4

Results

4.1 Binary Text Classification

In Natural Language Processing (NLP) and Machine Learning (MT), the task of text classification plays a crucial role. This task aims to assign pieces of text to one or more predefined categories. For instance, it may involve classifying news articles into predefined topics such as politics, sports, or entertainment [30]. Binary classification, a subtype of text classification, is a scenario with only two potential categories. Being the most straightforward classification task, each instance in binary classification can only fall into one category. An example of this would be sentiment analysis, where a review is classified as either positive or negative [29]. In the context of this paper, we delve into a particular binary classification task: determining whether a segment of English text from the public health field could lead to errors in machine translation.

4.1.1 Baseline Experiments

We executed fundamental baseline experiments on the COVID-19 and Maori Cancer datasets. In this context, we avoided using any feature engineering and directly inputted the datasets' text. This input was then subjected to One Hot and TF-IDF encoding before being incorporated into models for text classification. This experiment offers us a clear insight into the datasets' overall performance. We ran tests using three distinct machine learning models: Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Decision Tree. The results of their 5-Fold cross-validation experiments can be found in Table 4.1 and 4.2, where we have included the Standard Deviation (SD) for each performance indicator.

The experimental indicators are relatively low on both datasets, averaging around 60%. Unusual indicators appeared in both datasets during the experiments. For instance, the sensitivity in the Covid-19 dataset has remained steady at 55.4%, which is at least 5% lower than other indicators on average. On the other hand, the sensitivity in the Maori Cancer dataset is much higher than other indicators, reaching up to 98.5% at its peak. Correspondingly, its specificity is unusually low, hovering around 30%, with the lowest even being 0.22%. These experimental results suggest that there is a data imbalance in both datasets.

	MNB		**SVM**		**Decision Tree**	
Encoding	**One Hot**	**TF-IDF**	**One Hot**	**TF-IDF**	**One Hot**	**TF-IDF**
Accuracy	0.6011	0.6192	0.6192	0.6339	0.6271	0.6282
(SD)	(0.0376)	(0.0407)	(0.0248)	(0.0451)	(0.0262)	(0.0386)
AUC	0.6329	0.6372	0.6549	0.6648	0.6254	0.6299
(SD)	(0.036)	(0.0422)	(0.0262)	(0.0401)	(0.026)	(0.0339)
Sensitivity	0.554	0.5541	0.5541	0.6306	0.554	0.5812
(SD)	(0.0923)	(0.0796)	(0.0626)	(0.0964)	(0.0486)	(0.0453)
Precision	0.613	0.6396	0.6397	0.6364	0.635	0.6485
(SD)	(0.0317)	(0.0436)	(0.027)	(0.0421)	(0.0268)	(0.0231)
Specificity	0.6484	0.6846	0.6848	0.6371	0.6825	0.6394
(SD)	(0.0558)	(0.0607)	(0.0525)	(0.0646)	(0.042)	(0.0874)

TABLE 4.1. Five fold cross-validation results of different machine learning algorithms with two encodings for Covid-19 dataset.

	MNB		SVM		Decision Tree	
Encoding	**One Hot**	**TF-IDF**	**One Hot**	**TF-IDF**	**One Hot**	**TF-IDF**
Accuracy	0.6566	0.6768	0.6753	0.6739	0.6191	0.5858
(SD)	(0.026)	(0.005)	(0.0079)	(0.0177)	(0.0362)	(0.0331)
AUC	0.6579	0.6428	0.5894	0.6315	0.527	0.5143
(SD)	(0.0295)	(0.0231)	(0.0468)	(0.0544)	(0.0467)	(0.0179)
Sensitivity	0.8277	0.9915	0.9851	0.9723	0.7489	0.6766
(SD)	(0.0435)	(0.0089)	(0.0121)	(0.0161)	(0.0409)	(0.0381)
Precision	0.7125	0.6793	0.6799	0.6822	0.7042	0.6906
(SD)	(0.0142)	(0.0032)	(0.0052)	(0.0117)	(0.0108)	(0.0131)
Specificity	0.2963	0.0134	0.0225	0.0452	0.3183	0.777
(SD)	(0.454)	(0.123)	(0.161)	(0.396)	(0.687)	(0.777)

TABLE 4.2. Five fold cross-validation results of different machine learning algorithms with two encodings for Maori Cancer dataset

4.1.2 Full Feature Experiments

We applied the methods described in Chapter 3.3 to extract 491 and 460 distinct features from the COVID-19 and Maori Cancer dataset using the USAS system. These features were then fed into five different models, namely the Stacking Classifier (with MNB and SVM as the base and Logistic Regression as the final estimator), Logistic Regression, Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Gradient Boosting Decision Tree (GBDT). Additionally, we explored the impact of different normalization methods on the same model. The results are presented in Table 4.3 and Table 4.4, where we also included the standard deviation (SD) for each performance indicator.

The results show that the performance did not significantly improve after feature engineering yielded features compared to the baseline experiment. We can attribute this to four possible reasons:

Model	**Stacking Classifier**	**Logistic Regression**	**SVM**	**MNB**	**MNB**	**GBDT**
Features	ALL 491	ALL 491	ALL 491	ALL 491	ALL 491	ALL 491
Normalization	MinMaxScaler	MaxAbsScaler	MaxAbsScaler	L2 Normalization	MinMaxScaler	None
Accuracy	0.6067	0.6106	0.5938	0.6067	0.5998	0.6116
(SD)	(0.0327)	(0.0379)	(0.0427)	(0.0286)	(0.0322)	(0.0403)
AUC	0.6388	0.6489	0.6477	0.6416	0.6389	0.6547
(SD)	(0.0512)	(0.0439)	(0.0395)	(0.0284)	(0.0271)	(0.0368)
Sensitivity	0.5428	0.5626	0.5249	0.5327	0.5307	0.5665
(SD)	(0.0652)	(0.0582)	(0.0649)	(0.0592)	(0.0652)	(0.0604)
Precision	0.6181	0.6187	0.604	0.6212	0.6127	0.6204
(SD)	(0.0357)	(0.0367)	(0.0438)	(0.0301)	(0.0386)	(0.0444)
Specificity	0.664	0.658	0.5345	0.6798	0.668	0.6562
(SD)	(0.003)	(0.0035)	(0.00511)	(0.00319)	(0.00516)	(0.0059)

TABLE 4.3. Five fold cross-validation results of different machine learning algorithms for Covid-19 dataset with all 491 features.

- Feature Redundancy: Among these 491 and 460 features, some might be highly correlated or duplicated. These redundant features might not enhance the model's performance but could make the model overly complex and increase the risk of overfitting.
- Feature Noise: Some features might contain noise rather than information that is truly helpful to the target variable. These features might mislead the model and cause a decrease in performance.
- Curse of Dimensionality: In high-dimensional data, data points might be sparsely distributed in space, making it difficult for the model to learn effective decision

Model	Stacking Classifier	Logistic Regression	SVM	MNB	MNB	GBDT
Features	ALL 460	ALL 460	ALL 460	ALL 460	ALL 460	ALL 460
Normalization	MinMaxScaler	MaxAbsScaler	MaxAbsScaler	L2 Normalization	MinMaxScaler	None
Accuracy	0.6793	0.6414	0.6248	0.6107	0.6003	0.6755
(SD)	(0.0307)	(0.0279)	(0.0257)	(0.0387)	(0.0352)	(0.0304)
AUC	0.6072	0.6008	0.5877	0.5432	0.5491	0.5689
(SD)	(0.0311)	(0.0449)	(0.0335)	(0.0254)	(0.0271)	(0.0368)
Sensitivity	0.9702	0.8382	0.8292	0.811	0.8107	0.974
(SD)	(0.0452)	(0.0381)	(0.0345)	(0.0402)	(0.0662)	(0.0504)
Precision	0.6878	0.696	0.654	0.6433	0.6498	0.6832
(SD)	(0.0457)	(0.0367)	(0.0438)	(0.0301)	(0.0356)	(0.0433)
Specificity	0.0667	0.2244	0.2399	0.273	0.232	0.0432
(SD)	(0.034)	(0.045)	(0.0521)	(0.0419)	(0.0412)	(0.0369)

TABLE 4.4. Five fold cross-validation results of different machine learning algorithms for Maori Cancer dataset with all 460 features.

boundaries. This is the so-called "curse of dimensionality". Reducing the number of features might alleviate this problem and improve the model's performance.

- Overfitting: If the number of features significantly exceeds the number of data points, the model might overfit the training data, learning noise instead of the actual patterns. This could cause the model's performance to decrease on the test data.

Our subsequent experimental results also support the conclusions we have drawn.

4.1.3 Feature Optimization Experiments

Due to the excessive noise in the features, the uncertainty of the model can increase, leading to decreased performance. Therefore, we used the Recursive Feature Elimination (RFE) algorithm to optimize the high-dimensional dataset of 491 and 460 features, thus selecting

Model	Stacking Classifier	Logistic Regression	SVM	MNB	MNB	GBDT
Optimised Features	200	150	200	200	50	100
Normalization	MinMaxScaler	MaxAbsScaler	MaxAbsScaler	L2 Normalization	MinMaxScaler	None
Accuracy	0.7134	0.6887	0.7213	0.6462	0.6818	0.663
(SD)	(0.0422)	(0.0398)	(0.0325)	(0.0175)	(0.0196)	(0.037)
AUC	0.7648	0.7499	0.7662	0.6697	0.7106	0.6637
(SD)	(0.0329)	(0.0216)	(0.034)	(0.0598)	(0.0524)	(0.0531)
Sensitivity	0.6779	0.6422	0.652	0.5686	0.5764	0.5449
(SD)	(0.035)	(0.0372)	(0.032)	(0.0433)	(0.062)	(0.0375)
Precision	0.7288	0.7104	0.7491	0.6452	0.6839	0.6585
(SD)	(0.0346)	(0.0427)	(0.0613)	(0.0637)	(0.05)	(0.0586)
Specificity	0.7621	0.7248	0.7877	0.7033	0.7525	0.7387
(SD)	(0.0498)	(0.0516)	(0.0323)	(0.0418)	(0.0366)	(0.0199)

TABLE 4.5. Five fold cross-validation results of different machine learning algorithms for Covid-19 dataset with optimised features.

the most optimal feature set combination. We still chose five different models, namely the Stacking Classifier (with MultinomialNB and SVM as the base and Logistic Regression as the final estimator), Logistic Regression, Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Gradient Boosting Decision Tree (GBDT), each with different Normalizations. As shown in Table 4.5 and Table 4.6, we list the best performance of different models after feature selection through RFE, and the standard deviation (SD) for each performance indicator is also included.

From the results, we can see that after the features were selected through RFE, all indices significantly increased, on average, by more than 10%. We can also observe that for both two datasets, each model's optimal number of features is around 150 - 200. Next, we will further observe the best-performing Stacking Classifier. As shown in Table 4.7 and Figure 4.1, we use experiments of the Covid-19 dataset for analysing, we can see that all indices continue to rise as the number of optimized features decreases. Finally, it peaks at around 200 and decreases

Model	Stacking Classifier	Logistic Regression	SVM	MNB	MNB	GBDT
Optimised Features	150	150	150	200	150	100
Normalization	MinMaxScaler	MaxAbsScaler	MaxAbsScaler	L2 Normalization	MinMaxScaler	None
Accuracy	0.7374	0.7261	0.7114	0.7054	0.6974	0.6742
(SD)	(0.0336)	(0.0291)	(0.0413)	(0.0345)	(0.0326)	(0.0053)
AUC	0.7665	0.7444	0.7234	0.7133	0.7231	0.5843
(SD)	(0.0338)	(0.0278)	(0.0241)	(0.0287)	(0.0214)	(0.0487)
Sensitivity	0.935	0.9387	0.9301	0.9113	0.9254	0.9814
(SD)	(0.0196)	(0.0167)	(0.0143)	(0.0177)	(0.0142)	(0.0065)
Precision	0.7384	0.7335	0.7288	0.7165	0.7265	0.6804
(SD)	(0.0224)	(0.0217)	(0.0315)	(0.0299)	(0.0301)	(0.0033)
Specificity	0.3145	0.2754	0.2683	0.2533	0.232	0.0236
(SD)	(0.083)	(0.0836)	(0.0773)	(0.072)	(0.0666)	(0.0087)

TABLE 4.6. Five fold cross-validation results of different machine learning algorithms for Maori Cancer dataset with optimised features.

as the number of features decreases, showing a significant decline after 50. One thing worth noting is that when the feature optimization reaches 10, Specificity counterintuitively increases while the corresponding Sensitivity significantly decreases. This indicates that the reduction in the number of features in the dataset has already caused overfitting.

4.1.4 Multi-Dimensional Text Classification Pipeline

In Section 3.6, we introduced a novel framework MD-TCP that merges the previously extracted USAS with new TF-IDF features. This framework undergoes several crucial procedures, including data preprocessing, feature extraction, dimensionality reduction, feature selection, and addressing the class imbalance, all of which contribute to a significant enhancement in the model's performance. As depicted in Table 4.8 and Table 4.9, we chose the Stacking Classifier and Logistic Regression, the two top-performing models from earlier experiments, to test

Optimised Features	300	200	150	100	50	25	10
Accuracy	0.6927	0.7134	0.7065	0.7085	0.6878	0.6423	0.6057
(SD)	(0.0338)	(0.0422)	(0.0196)	(0.0215)	(0.0245)	(0.0209)	(0.0255)
AUC	0.731	0.7648	0.7682	0.7584	0.7328	0.7019	0.6463
(SD)	(0.0423)	(0.0329)	(0.0287)	(0.0452)	(0.0606)	(0.0673)	(0.0485)
Sensitivity	0.6461	0.6779	0.6798	0.664	0.6401	0.5647	0.4651
(SD)	(0.0457)	(0.035)	(0.0426)	(0.0193)	(0.0534)	(0.0395)	(0.0705)
Precision	0.6972	0.7288	0.7282	0.7259	0.6975	0.6809	0.6388
(SD)	(0.0499)	(0.0346)	(0.0373)	(0.0488)	(0.0562)	(0.0606)	(0.0816)
Specificity	0.7329	0.7621	0.7544	0.7564	0.7212	0.7152	0.7231
(SD)	(0.0283)	(0.0498)	(0.0372)	(0.0104)	(0.0465)	(0.0437)	(0.0733)

TABLE 4.7. Five fold cross-validation results of Stacking Classifier for Covid-19 dataset with different optimised features number.

MD-TCP. The results reveal that the methods within MD-TCP have substantially improved performance across all metrics, even in the absence of feature selection and with all features incorporated. Notably, the Stacking Classifier surpasses the old framework that underwent feature selection in nearly all aspects.

In the experiment, we proceeded to utilize feature selection (RFE) with the Stacking Classifier of the new framework. The experiment results of Covid-19 dataset, presented in Table 4.10, indicate that the model's performance reached its peak when optimized to 400 features, with all metrics nearing 90%. A more intuitive understanding of the improvement achieved by our latest framework can be gleaned from Figure 4.2 and 4.3.

It shows that our new approach MD-TCP surpasses initial baseline experiments by over 30% in the overall performance of both Covid-19 and Maori Cancer datasets. Additionally, our

Model	Stacking Classifier	Logistic Regression
Features	All	All
Normalization	MinMaxScaler	MaxAbsScaler
Accuracy	0.7211	0.6511
(SD)	(0.0308)	(0.0398)
AUC	0.7726	0.6941
(SD)	(0.0154)	(0.0216)
Sensitivity	0.7798	0.6031
(SD)	(0.0209)	(0.0372)
Precision	0.6893	0.6707
(SD)	(0.0189)	(0.0427)
Specificity	0.6548	0.6997
(SD)	(0.0334)	(0.0516)

TABLE 4.8. Five fold cross-validation results of Stacking Classifier and Logistic Regression for Covid-19 dataset with all features using MD-TCP.

MD-TCP effectively addresses the class imbalance problem present in both the Covid-19 and Maori Cancer datasets. We previously noted that this issue resulted in consistently low Sensitivity in the experiments of Covid-19 dataset, while feature extraction and selection partially mitigated this problem, Sensitivity remained lower than other metrics. However, the MD-TCP successfully elevated Sensitivity to the same level as other metrics, reaching a high of 88.53%, which represents a significant 33.13% increase compared to the baseline. In the Maori Cancer dataset experiment, the Specificity indicator exhibited a consistent issue. Under the old framework, both in the Baseline and All Features experiments, Specificity fluctuated between 20% and 30%, even dropping to a low of 2.25%. This suggests a significant imbalance in the dataset. However, with the implementation of the MD-TCP method, Specificity surged to over 70%. Furthermore, after the application of RFE screening, Specificity reached a peak

Model	Stacking Classifier	Logistic Regression
Features	All	All
Normalization	MinMaxScaler	MaxAbsScaler
Accuracy	0.8114	0.7006
(SD)	(0.0296)	(0.0249)
AUC	0.8614	0.7699
(SD)	(0.0165)	(0.0272)
Sensitivity	0.8698	0.6946
(SD)	(0.0413)	(0.0295)
Precision	0.7942	0.7005
(SD)	(0.0182)	(0.0301)
Specificity	0.7473	0.7066
(SD)	(0.0398)	(0.0386)

TABLE 4.9. Five fold cross-validation results of Stacking Classifier and Logistic Regression for Maori Cancer dataset with all features using MD-TCP.

of 90.19%. These results strongly attest to the effectiveness of MD-TCP in addressing issues of sample imbalance.

From Figure 4.4, we can extract some fresh insights: MD-TCP retains the traits of the old pipeline, where the model's performance initially increases with the reduction of features via RFE optimization, then starts to decline. Interestingly for both Covid-19 and Maori Cancer datasets, when the Stacking Classifier model's performance reached its peak, the number of optimized features was 400 and 330, double that in the old pipeline. We can analyze this phenomenon as follows:

- Addition of new information: The TF-IDF features introduce a new kind of information distinct from the original features, namely, the significance of words in the text.

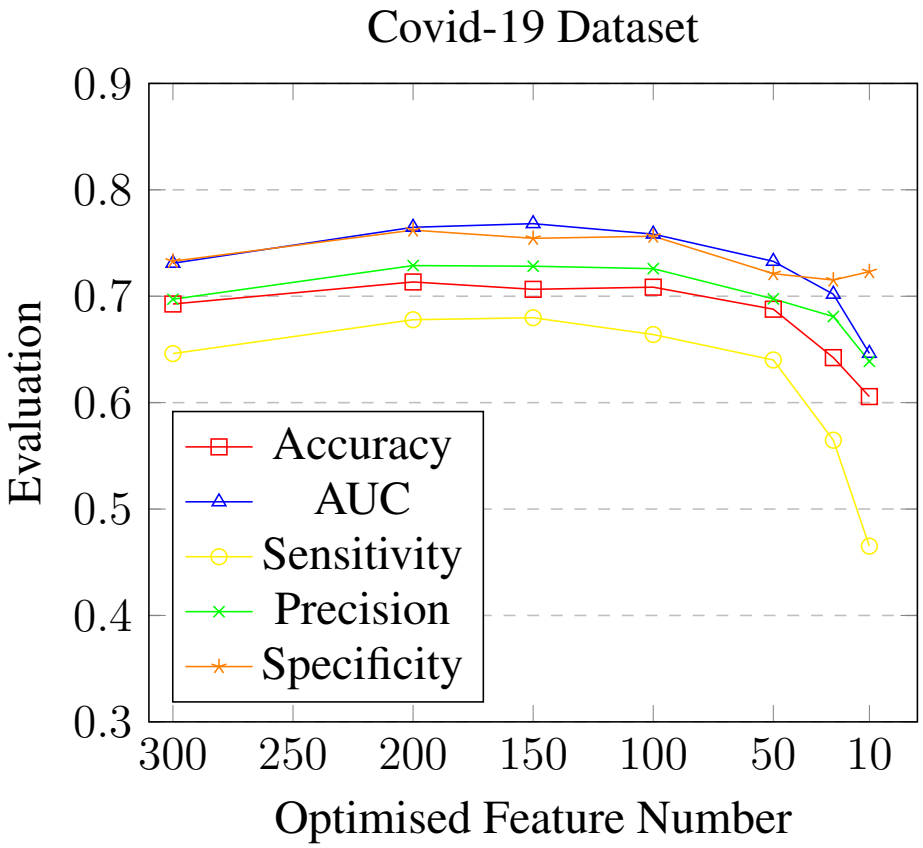

FIGURE 4.1. Evaluation of Stacking Classifier with different Optimised Features for Covid-19 dataset.

This new information might allow the model to better comprehend and classify the text.

- Impact of feature processing: Procedures such as PCA, One-Hot encoding, MinMaxScaler scaling, RFE feature selection, and SMOTE oversampling might have modified the distribution and structure of the features, turning more features into valuable information sources.
- Model complexity: After adding new features and processing them, the model might need to be more complex (i.e., have more features) to achieve the best results. This could be because the new features and processing have added to the data's complexity.

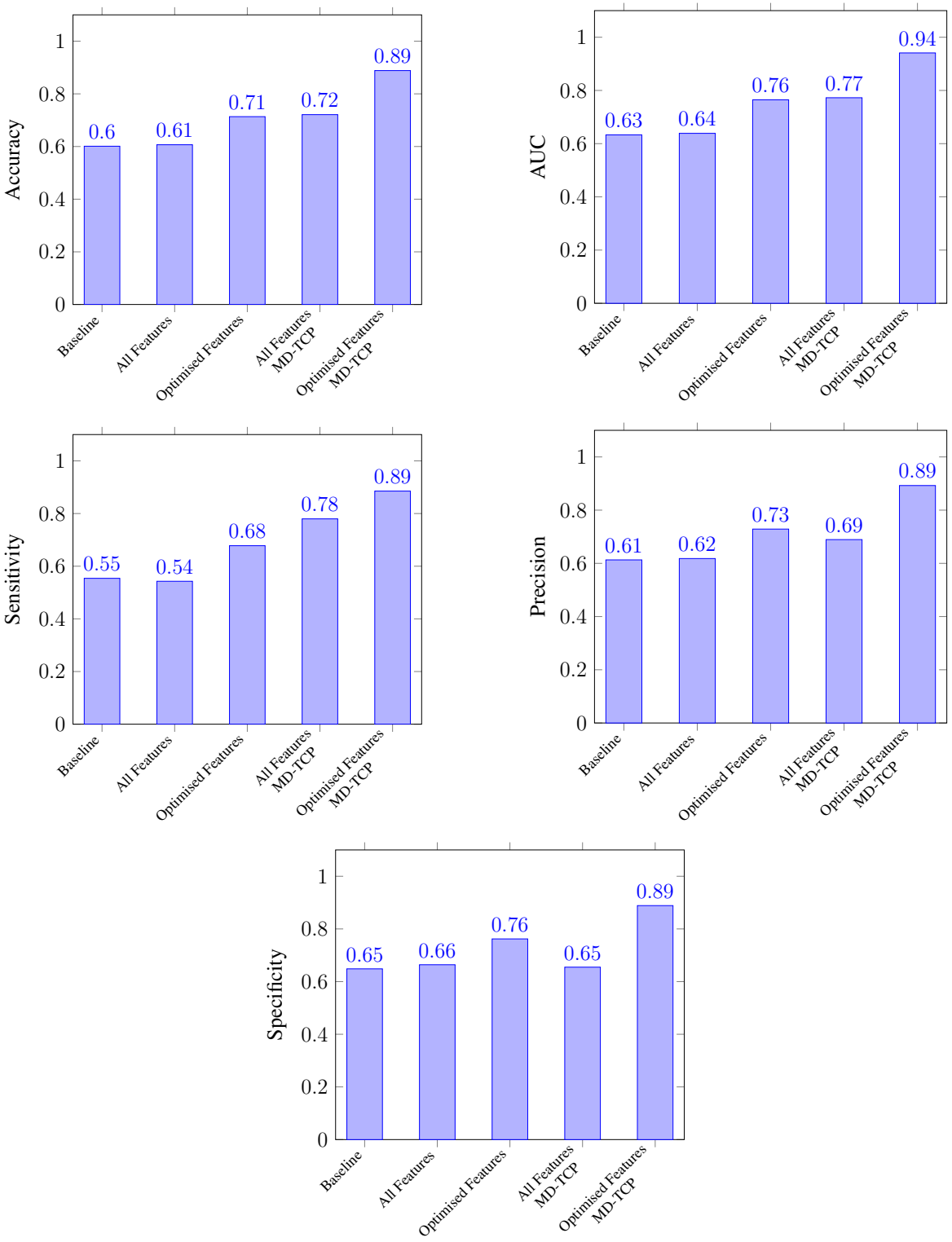

FIGURE 4.2. Performance of Stacking Classifiers using different methods for Covid-19 dataset.

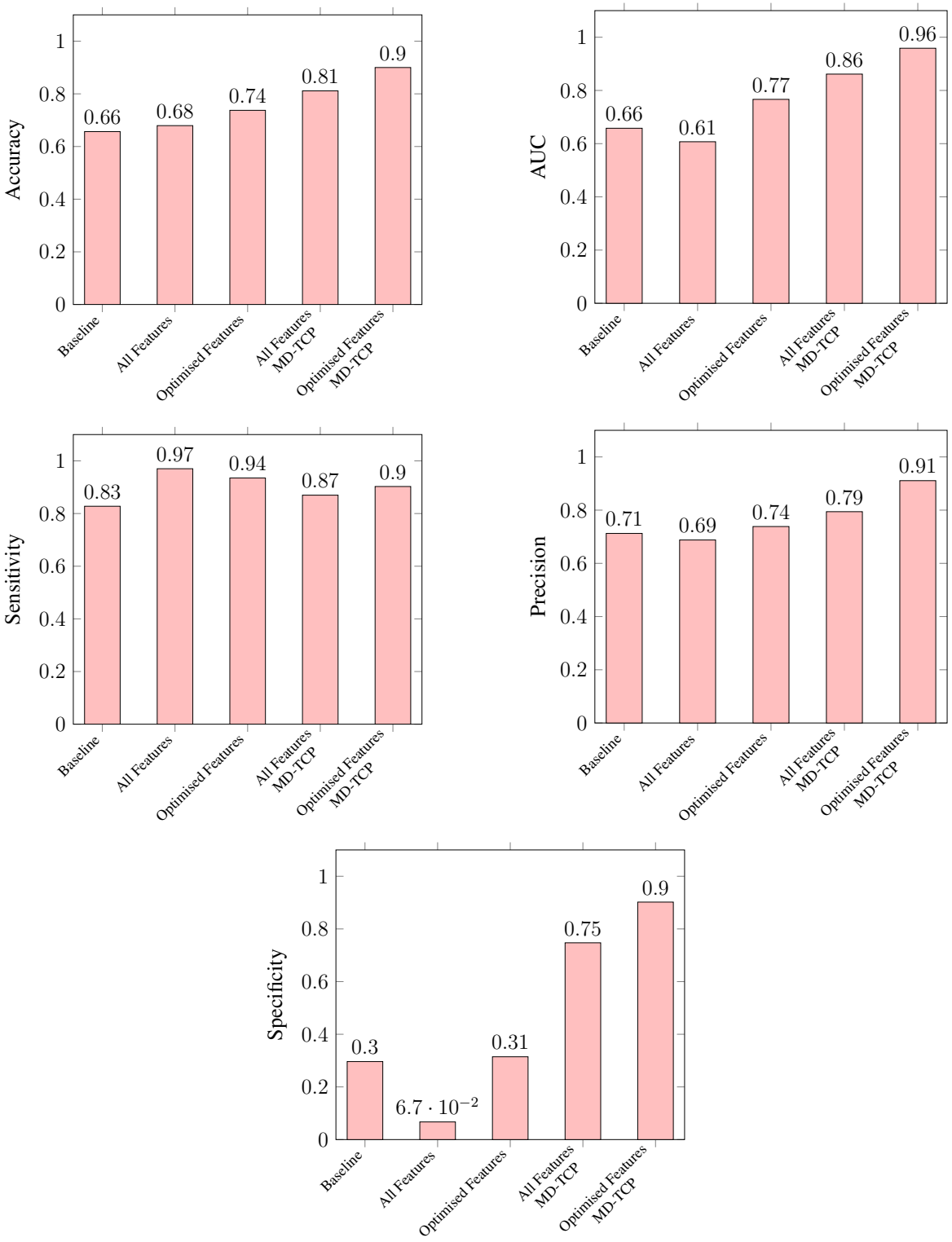

FIGURE 4.3. Performance of Stacking Classifiers using different methods for Maori Cancer dataset.

Optimised Features	420	400	350	300	200	100	50
Accuracy	0.8753	0.8884	0.8637	0.8334	0.8063	0.7679	0.7417
(SD)	(0.0139)	(0.0116)	(0.0121)	(0.0249)	(0.0067)	(0.0336)	(0.0338)
AUC	0.9374	0.941	0.9289	0.9084	0.879	0.8451	0.8099
(SD)	(0.0077)	(0.0104)	(0.0068)	(0.0193)	(0.0071)	(0.027)	(0.0366)
Sensitivity	0.876	0.8853	0.8698	0.8248	0.8153	0.7649	0.7786
(SD)	(0.0164)	(0.0369)	(0.0201)	(0.0374)	(0.0368)	(0.0481)	(0.0223)
Precision	0.8774	0.8925	0.8692	0.8306	0.8098	0.766	0.7375
(SD)	(0.0298)	(0.0126)	(0.0217)	(0.0236)	(0.0183)	(0.0491)	(0.0467)
Specificity	0.8699	0.8886	0.8669	0.8436	0.8003	0.7585	0.712
(SD)	(0.0375)	(0.0208)	(0.0233)	(0.03)	(0.0343)	(0.0494)	(0.0623)

TABLE 4.10. Five fold cross-validation results of Stacking Classifier for Covid-19 dataset with different optimised features number using MD-TCP.

In conclusion, MD-TCP has altered the data's characteristics to a certain extent and augmented the information that the model can leverage.

Another noteworthy point is that we observe the performance of AUC to be approximately 5% superior to other metrics for both datasets. AUC, or Area Under the Curve, typically refers to the area under the Receiver Operating Characteristic Curve (ROC curve). The ROC curve is plotted with the false positive rate (1-Specificity) on the x-axis and the true positive rate (Sensitivity) on the y-axis. A larger AUC value indicates better model performance.

The reasons why AUC might outperform other metrics in certain scenarios:

- Threshold independence: Metrics such as Accuracy, Precision, Sensitivity, and Specificity all depend on the chosen threshold. Different thresholds can yield vastly

Optimised Features	430	380	330	300	200	100	50
Accuracy	0.8802	0.8849	0.9	0.881	0.854	0.8087	0.796
(SD)	(0.0203)	(0.0164)	(0.0216)	(0.0143)	(0.0033	(0.0162)	(0.0217)
AUC	0.948	0.9465	0.9586	0.9448	0.9175	0.8818	0.8553
(SD)	(0.0233)	(0.0089)	(0.0071)	(0.0155)	(0.0068)	(0.0138)	(0.0255)
Sensitivity	0.8787	0.8981	0.9027	0.8847	0.8547	0.8473	0.8308
(SD)	(0.0435)	(0.0276)	(0.0217)	(0.0387)	(0.0409)	(0.0305)	(0.041)
Precision	0.8886	0.8815	0.9108	0.8875	0.8619	0.8093	0.7725
(SD)	(0.0239)	(0.0312)	(0.0403)	(0.0139)	(0.0184)	(0.0087)	(0.026)
Specificity	0.8818	0.8563	0.9019	0.8716	0.8496	0.7854	0.7264
(SD)	(0.0287)	(0.0343)	(0.0471)	(0.0195)	(0.0424)	(0.0183)	(0.0116)

TABLE 4.11. Five fold cross-validation results of Stacking Classifier for Maori Cancer dataset with different optimised features number using MD-TCP.

different results. AUC, however, measures the model's overall performance across all possible thresholds, making it independent of threshold selection.

- Immunity to class imbalance: In situations of class imbalance, a model might achieve high Accuracy by merely predicting the majority class. However, such models often perform poorly when predicting minority classes. AUC considers the model's predictive performance across all classes, making it immune to class imbalance.
- Consideration of both false positives and false negatives: AUC takes into account both the false positive rate and the true positive rate, meaning it considers the quantities of both false positives and false negatives. Other metrics might focus on

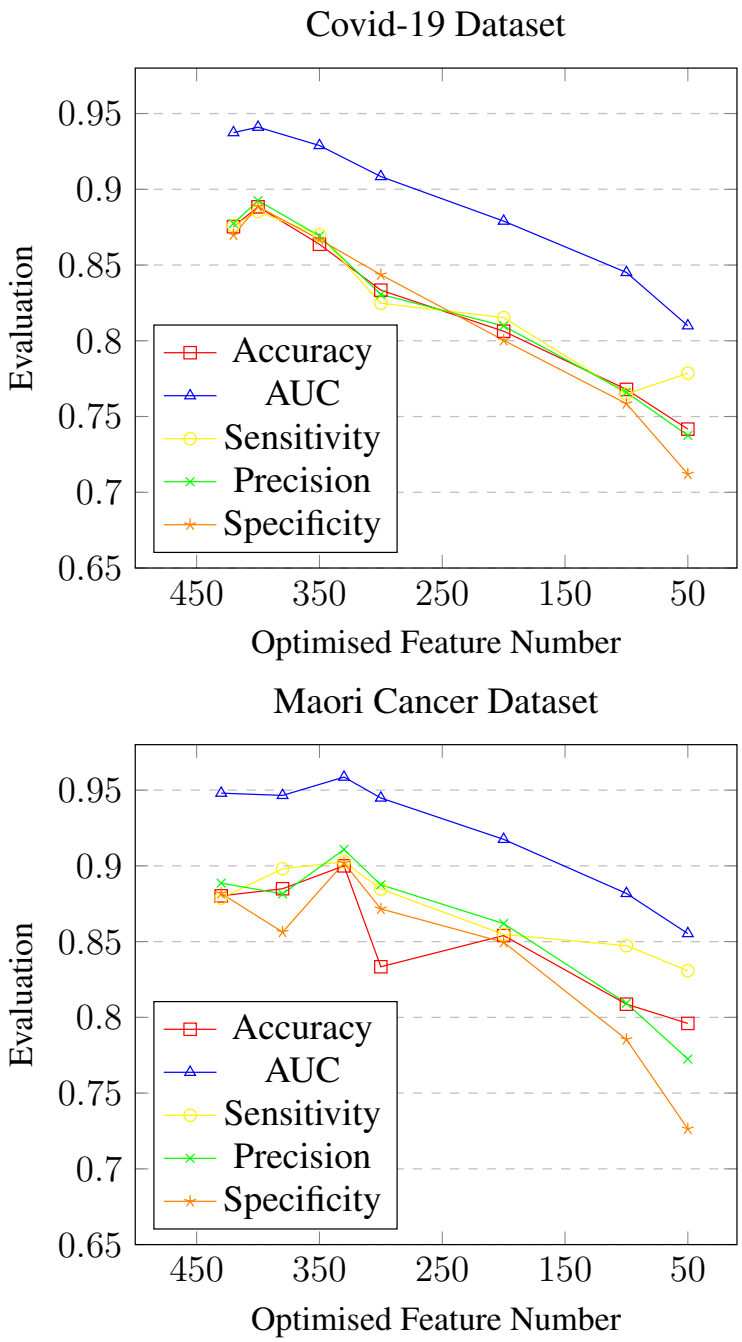

FIGURE 4.4. Evaluation of Stacking Classifier with different Optimised Features for Covid-19 dataset.

only one aspect, for instance, Precision focuses on false positives, while Sensitivity focuses on false negatives.

In summary, AUC is a comprehensive metric. It considers the model's performance across all possible thresholds, is unaffected by class imbalance and threshold selection, and takes into account both false positives and false negatives. These factors might contribute to AUC outperforming other metrics in certain cases.

CHAPTER 5

Conclusions

5.1 Conclusion

The primary objectives of our study were twofold: firstly, to develop machine learning classifiers under the umbrella of our proposed Multi-Dimensional Text Classification Pipeline (MD-TCP), to assist non-native English speakers, including medical professionals and patients, in understanding the risks associated with using machine translation tools for health information. Secondly, within the MD-TCP framework, we aimed to create machine learning classifiers as a risk-prevention mechanism, predicting the likelihood of translation errors based on the features of the original English source texts.

A significant contribution of our study was the development of interpretable machine learning models within the MD-TCP to assess and predict the risks associated with translating health-related content using machine learning tools. Applying these classifiers, including models such as the Multinomial Naive Bayes (MNB) classifier, in clinical or health research settings requires understanding their functionality. These classifiers use linguistic cues to predict whether a specific English health text is likely to be mistranslated, thereby contributing to the risk mitigation aspect of our MD-TCP.

The best-performing classifier within the MD-TCP succeeded in classifying translation-error-prone and non-error-prone English texts using a small number of optimized features. This led to significant accuracy, sensitivity, precision, and specificity improvements compared to the baseline.

Our study challenges the traditional view that translation errors are primarily due to the presence of complex medical jargon. Instead, we found that with the rapid development in computer science and the application of our MD-TCP, the accuracy of machine translation tools is improving significantly, and the translation of medical terminology no longer represents the top challenge. Instead, linguistic phenomena such as polysemy or the context-dependent nature of common words in specialized health and medical domains are causing subtle yet significant errors and confusion in machine translation outputs. This further underscores the importance and effectiveness of our Multi-Dimensional Text Classification Pipeline in tackling these complexities.

Bibliography

[1] P. Hershberg, S. Goldfinger, F. Lemon and W. Fessel, 'Medical record as index of quality of care.,' *The New England Journal of Medicine*, vol. 286, no. 13, pp. 725–726, 1972.

[2] S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead and K. B. Johnson, 'Data from clinical notes: A perspective on the tension between structure and flexible documentation,' *Journal of the American Medical Informatics Association*, vol. 18, no. 2, pp. 181–186, 2011.

[3] K. Kirchhoff, A. M. Turner, A. Axelrod and F. Saavedra, 'Application of statistical machine translation to public health information: A feasibility study,' *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 473–478, 2011.

[4] S. Sharma, 'How to become a competent medical writer?' *Perspectives in clinical research*, vol. 1, no. 1, p. 33, 2010.

[5] R. J. Roiger, 'Data mining: A tutorial-based primer,' 2017.

[6] A. Irvine, J. Morgan, M. Carpuat, I. Daumé Hal and D. Munteanu, 'Measuring machine translation errors in new domains,' *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 429–440, Oct. 2013.

[7] M. Ji, W. Xie, R. Huang and X. Qian, 'Forecasting erroneous neural machine translation of disease symptoms: Development of bayesian probabilistic classifiers for cross-lingual health translation,' *International journal of environmental research and public health*, vol. 18, no. 18, p. 9873, 2021.

[8] S. Piao, F. Bianchi, C. Dayrell, A. D'Egidio and P. Rayson, 'Development of the multilingual semantic annotation system,' in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Association for Computational Linguistics (ACL), Denver, CO, USA, 2015, pp. 1268–1274.

[9] Y. Sun, 'Analysis of chinese machine translation training based on deep learning technology,' *Computational Intelligence and Neuroscience*, vol. 2022, p. 6 502 831, Aug. 2022.

[10] J. Zhu, 'English lexical analysis system of machine translation based on simple recurrent neural network,' *Computational Intelligence and Neuroscience*, vol. 2022, p. 9 702 112, 2022.

[11] B. Ren, 'The use of machine translation algorithm based on residual and lstm neural network in translation teaching,' *PloS one*, vol. 15, no. 11, e0240663, Nov. 2020.

[12] E. D. Liddy, 'Natural language processing,' 2001.

[13] P. M. Nadkarni, L. Ohno-Machado and W. W. Chapman, 'Natural language processing: An introduction,' *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[14] J. Hirschberg and C. D. Manning, 'Advances in natural language processing,' *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[15] P. H. Winston, *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc., 1984.

[16] N. J. Nilsson, *Principles of artificial intelligence*. Springer Science & Business Media, 1982.

[17] T. M. Mitchell *et al.*, *Machine learning*. McGraw-hill New York, 2007, vol. 1.

[18] M. I. Jordan and T. M. Mitchell, 'Machine learning: Trends, perspectives, and prospects,' *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[19] F. Olsson, 'A literature survey of active machine learning in the context of natural language processing,' 2009.

[20] A. Singh, N. Thakur and A. Sharma, 'A review of supervised machine learning algorithms,' in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, Ieee, 2016, pp. 1310–1315.

[21] G. Ke *et al.*, 'Lightgbm: A highly efficient gradient boosting decision tree,' *Advances in neural information processing systems*, vol. 30, 2017.

[22] Y. LeCun, Y. Bengio and G. Hinton, 'Deep learning,' *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[23] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*. MIT press, 2016.

[24] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, 'A survey of convolutional neural networks: Analysis, applications, and prospects,' *IEEE transactions on neural networks and learning systems*, 2021.

[25] L. R. Medsker and L. Jain, 'Recurrent neural networks,' *Design and Applications*, vol. 5, pp. 64–67, 2001.

[26] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, 'Bert: Pre-training of deep bidirectional transformers for language understanding,' *arXiv preprint arXiv:1810.04805*, 2018.

[27] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, 'Improving language understanding by generative pre-training,' 2018.

[28] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes and D. Brown, 'Text classification algorithms: A survey,' *Information*, vol. 10, no. 4, p. 150, 2019.

[29] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu and J. Gao, 'Deep learning–based text classification: A comprehensive review,' *ACM computing surveys (CSUR)*, vol. 54, no. 3, pp. 1–40, 2021.

[30] C. C. Aggarwal and C. Zhai, 'A survey of text classification algorithms,' *Mining text data*, pp. 163–222, 2012.

[31] A. Joulin, E. Grave, P. Bojanowski and et al., 'Bag of tricks for efficient text classification,' *arXiv preprint arXiv:1607.01759*, 2016.

[32] B. Mahesh, 'Machine learning algorithms-a review,' *International Journal of Science and Research (IJSR)*, vol. 9, no. 2020, pp. 381–386,

[33] R. Xu and D. Wunsch, 'Survey of clustering algorithms,' *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[34] D. J. Hand, 'Principles of data mining,' *Drug safety*, vol. 30, pp. 621–622, 2007.

[35] A. Hotho, A. Nürnberger and G. Paaß, 'A brief survey of text mining,' *Journal for Language Technology and Computational Linguistics*, vol. 20, no. 1, pp. 19–62, 2005.

[36] V. Gupta, G. S. Lehal *et al.*, 'A survey of text mining techniques and applications,' *Journal of emerging technologies in web intelligence*, vol. 1, no. 1, pp. 60–76, 2009.

[37] I. Guyon, S. Gunn, M. Nikravesh and L. A. Zadeh, *Feature extraction: foundations and applications*. Springer, 2008, vol. 207.

[38] S. Khalid, T. Khalil and S. Nasreen, 'A survey of feature selection and feature extraction techniques in machine learning,' in *2014 science and information conference*, IEEE, 2014, pp. 372–378.

[39] K. W. Church, 'Word2vec,' *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[40] J. Pennington, R. Socher and C. D. Manning, 'Glove: Global vectors for word representation,' in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[41] P. Rayson, D. Archer, S. Piao and A. M. McEnery, 'The ucrel semantic analysis system.,' 2004.

[42] H. Abdi and L. J. Williams, 'Principal component analysis,' *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[43] R. Flesch, 'A new readability yardstick.,' *Journal of applied psychology*, vol. 32, no. 3, p. 221, 1948.

[44] R. Gunning, *The Technique of Clear Writing*. McGraw-Hill, 1952, ISBN: 9787000014190.

[45] G. H. Mc Laughlin, 'Smog grading-a new readability formula,' *Journal of reading*, vol. 12, no. 8, pp. 639–646, 1969.

[46] D. Kumawat and V. Jain, 'Pos tagging approaches: A comparison,' *International Journal of Computer Applications*, vol. 118, no. 6, 2015.

[47] I. Rish *et al.*, 'An empirical study of the naive bayes classifier,' in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

[48] D. D. Lewis, 'Naive (bayes) at forty: The independence assumption in information retrieval,' in *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings 10*, Springer, 1998, pp. 4–15.

[49] W. S. Noble, 'What is a support vector machine?' *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[50] B. Schölkopf, 'The kernel trick for distances,' *Advances in neural information processing systems*, vol. 13, 2000.

[51] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein and M. Klein, *Logistic regression*. Springer, 2002.

[52] F. C. Pampel, *Logistic regression: A primer*. Sage publications, 2020.

[53] S. Džeroski and B. Ženko, 'Is combining classifiers with stacking better than selecting the best one?' *Machine learning*, vol. 54, pp. 255–273, 2004.

[54] Z. Yin, Y. Wang, L. Liu, W. Zhang and J. Zhang, 'Cross-subject eeg feature selection for emotion recognition using transfer recursive feature elimination,' *Frontiers in neurorobotics*, vol. 11, p. 19, 2017.

[55] H. Jeon and S. Oh, 'Hybrid-recursive feature elimination for efficient feature selection,' *Applied Sciences*, vol. 10, no. 9, p. 3211, 2020.

[56] E. Loper and S. Bird, 'Nltk: The natural language toolkit,' *arXiv preprint cs/0205028*, 2002.

[57] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, 'Smote: Synthetic minority over-sampling technique,' *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.