Investigating Multi-label Classification for Bio-inspired Design by Using Text Mining and Natural Language Processing

Siyuan Sun

Department of Mechanical Engineering

McGill University, Montreal, Quebec

April 2022

A thesis submitted to McGill University in partial fulfilment of
the requirements of the degree of Master of Mechanical Engineering

Abstract

Nowadays, bio-inspiration has enhanced the creation of sustainable and innovative solutions to modern engineering problems. Nature is a great source for multi-functional and optimized designs which could inspire mechanical engineers with innovative new ideas. However, it is very difficult to extract desired design knowledge from databases that are primarily text-based and focus on describing nature and biological systems. The main objective of this research is to build a multi-label classification system to classify bio-inspired designs to support design ideation. The method proposed in this study is to fuse text-based techniques such as natural language processing and text mining with machine learning to learn and predict the functionalities of bio-inspired design. Various design multi-functionalities were summarized based on the available resources from the AskNature database, then the main information extracted from the database and papers were labelled with corresponding multi-functionalities. Due to the high complexity of multi-label classification, multi-label classifiers were built based on various combinations of basic classifiers and trained to classify selected examples. One case study was conducted to verify the impact of the proposed system. The results showed that the proposed system is feasible and would be a solution for classifying the bio-inspired design and functional basis knowledge extraction method.

Résumé

De nos jours, la bio-inspiration a amélioré la création de solutions durables et innovantes aux problèmes d'ingénierie modernes. La nature est une excellente source de conceptions multifonctionnelles et optimisées qui pourraient inspirer les ingénieurs en mécanique avec de nouvelles idées innovantes. Cependant, il est très difficile d'extraire les connaissances de conception souhaitées à partir de bases de données qui sont principalement basées sur du texte et se concentrent sur la description de la nature et des systèmes biologiques. L'objectif principal de cette recherche est de construire un système de classification multi-étiquettes pour classer les conceptions bio-inspirées afin de soutenir l'idéation de conception. La méthode proposée dans cette étude consiste à fusionner des techniques basées sur le texte telles que le traitement du langage naturel et l'exploration de texte avec l'apprentissage automatique pour apprendre et prédire les fonctionnalités de la conception bio-inspirée. Diverses multifonctionnalités de conception ont été résumées sur la base des ressources disponibles dans la base de données AskNature, puis les principales informations extraites de la base de données et des articles ont été étiquetées avec les multifonctionnalités correspondantes. En raison de la grande complexité de la classification multi-étiquettes, des classificateurs multi-étiquettes ont été construits sur la base de diverses combinaisons de classificateurs de base et formés pour classer des exemples sélectionnés. Une étude de cas a été menée pour vérifier l'impact du système proposé. Les résultats ont montré que le système proposé est faisable et serait une solution pour classer la conception bio-inspirée et la méthode d'extraction des connaissances de base fonctionnelle.

Acknowledgements

Throughout my graduate studies, I have received invaluable support that has aided me in getting to where I am now. First thanks go to my supervisor, Dr, Yaoyao Fiona Zhao and Dr, Yong Zeng, for their support and guidance, for providing me with invaluable assistance and constructive advice throughout my study at McGill University, and for leading some valuable insights into my research. I would like to thank my ADML lab colleagues, who were always willing to collaborate with me and offer some useful advice. A special thanks are due to Pavan, for his advice and suggestions on the collected dataset and research progress. I must also thank Chonghui, Jiarui and Ying for their valuable suggestions during my study.

Having started as a graduate student at McGill University, I have met lots of great people who have made my time at McGill a wonderful experience, particularly the faculty and staff of the Mechanical Engineering Department.

Finally, I would like to express my great gratitude to my parents for their continuous love and support. They have provided me with love and encouragement while I studied mechanical engineering a thousand miles away from home. As I move on to the next chapter of my life, I am delighted to share my success with all of you.

Table of Contents

Abstract ii

Résumé iii

Acknowledgements iv

List of Tables viii

List of Figures ix

Abbreviations x

1 Introduction 1

1.1 Bio-inspired Design 1

1.2 Bridging Biological Knowledge with Engineering Systems 2

1.3 Research Objectives 4

1.4 Thesis Structure 5

2 Literature Review 6

2.1 Tools and Methods for supporting Bio-inspired Design 9

2.1.1 Database Tools and Methods 9

2.1.2 Modeling Tools and Methods 11

2.1.3 Heuristics Tools and Methods 12

2.2 Text Mining and Natural Language Processing for Knowledge Extraction 13

2.2.1 Text Mining 14

2.2.2 Natural Language Processing 16

2.3 Domain-Integrated-Design method for bio-inspired design concept generation 19

3 Background on Classification Model 22

3.1 Multi-label Classification 24

3.1.1 Problem Transformation Methods 24

3.1.2 Algorithm Adaptation Methods 28

3.2 Classification in Text Mining and Natural Language Processing 30

3.2.1 Tokenization 31

3.2.2 Part of Speech (POS) Tagging 32

3.2.3 Word Stemming and Lemmatizing 33

3.2.4 Natural Language Toolkit (NLTK) 34

4 Bio-inspired Design Classification System 36

4.1 Dataset Collection 36

4.2 Data Preprocessing 38

4.3 Feature Extraction 39

4.3.1 Bag-of-Words (BOW) 39

4.3.2 TF-IDF 39

4.4 Classifier Development 41

4.4.1 Baseline Classifier Choice 41

4.4.2 Ensemble-based Classifier 44

4.5 Classifier Evaluation Metrics 46

4.5.1 Example-based Evaluation 47

4.5.2 Label-based Evaluation 48

4.5.3 Ranking Evaluation 49

5 Implementation and Experiments 51

5.1 Dataset and Experiment Setup 51

5.2 Overall Performance Evaluation 53

5.3 Hyperparameter Tuning Analysis 59

5.4 Feature Extraction Analysis 60

5.5 Case Study 63

5.6 Discussion 64

5.6.1 The Advantages and Limitations of the Proposed System 64

5.6.2 Main Contributions of the Research 65

6 Conclusions and Future Work 67

6.1 Conclusions 67

6.2 Directions for Future Work 68

References 70

List of Tables

Table 1.1: Examples of bio-inspired engineering applications 1

Table 3.1: Classification model parameter representations 23

Table 3.2: Notations for Binary Relevance and Classifier Chains 25

Table 3.3: Mathematical Notations for ML-KNN 29

Table 3.4: POS tagging description 32

Table 3.5: Different constituents for Penn Treebank 33

Table 3.6: Examples of word stemming [124] 34

Table 3.7: Fundamental requirements of using NLTK [127] 35

Table 4.1: Labels of bio-inspired designs 38

Table 4.2: Algorithm adaptation classifiers 45

Table 4.3: Problem transformation classifiers 46

Table 5.1: Dataset statistics 52

Table 5.2: Accuracy of each label of One-versus-All (LR baseline) Classifier based on TF-IDF features 54

Table 5.3: Micro, macro quality number of One-versus-All (LR baseline) Classifier based on TF-IDF features 54

Table 5.4: Classifier performance based on TF-IDF features 55

Table 5.5: Hyperparameter tuning for One-versus-All classifier based on TF-IDF features 59

Table 5.6: Hyperparameter tuning for XGBoost classifier based on TF-IDF features 59

Table 5.7: Hyperparameter for ML-KNN based on TF-IDF features 60

Table 5.8: Model performance based on BOW features 61

Table 5.9: Model performance based on TF-IDF (1-3 grams) features 61

Table 5.10: Performance of different feature extraction methods 62

Table 5.11: Top 15 keywords in the selected AskNature page 64

Table 6.1: Classifiers with great performance 68

List of Figures

Figure 1.1: AskNature database example [20] 3

Figure 2.1: Bio-inspired design interchangeable terms [27] 6

Figure 2.2: Bio-inspired design paradigm [27] 7

Figure 2.3: Knowledge transfer in bio-inspired design (based on Hesse's work) [1, 39] 8

Figure 2.4: Biomimicry Taxonomy, an underlying representation and search structure for AskNature [7] 10

Figure 2.5: SAPPHIRE model to understand natural systems [55] 12

Figure 2.6: Schematic representation of the standard text mining pipeline for information extraction [62] 16

Figure 2.7: Visual summary of bio-inspired design tools and methods [7] 19

Figure 2.8: Bio-inspired suture pin leg/needle [15] 20

Figure 3.1: Supervised learning subdivisions 22

Figure 3.2: One-versus-All and One-versus-One decomposition schemes [107] 26

Figure 3.3: Schematic representation of classification process with supervised learning [80] 30

Figure 3.4: Parse tree of sample sentence [119] 33

Figure 4.1: The overall flowchart of Bio-inspired Design Classification System 37

Figure 4.2: Structure of Naive Bayes classifier [137] 42

Figure 4.3: Logistic Regression example [138] 42

Figure 4.4: Linearly separable samples of SVM [145] 43

Figure 4.5: Decision tree graph example [80] 44

Figure 4.6: Ensemble-based classifier framework [107] 45

Figure 5.1: Dataset information 51

Figure 5.2: Part of the dataset 52

Figure 5.3: Accuracy of classifiers based on TF-IDF features 56

Figure 5.4: Hamming loss of classifiers based on TF-IDF features 57

Figure 5.5: Micro averaged measures of classifiers based on TF-IDF features 57

Figure 5.6: Macro averaged measures of classifiers based on TF-IDF features 58

Figure 5.7: Number of features vs. ML-KNN Accuracy 62

Figure 5.8: Main information before and after preprocessing 63

Abbreviations

ADML	Additive Design and Manufacturing Lab
ML	Machine Learning
NLP	Natural Language Processing
UI	User interface
DID	Domain Integrated Design
DDD	Data-driven Design
DART	Discovery and Aggregation of Relations in Text
LaSIE	Large Scale Information Extraction
SBLD	System based on Linked Data
IE	Information Extraction
ST	Semantics Technology
FOBIE	Focused Open Biology Information Extraction
TF-IDF	Term Frequency-Inverse Document Frequency
BOW	Bags-of-words
Corpus	A sample of written language used for the purpose of linguistic analysis
POS Tagging	Words are separated into verbs, nouns, adjectives, adventives, etc.
AP	Average precision
BR	Binary Relevance
CC	Classifier Chain
ECC	Ensemble of Classifier Chain
HL	Hamming Loss
LP	Label Powerset
KNN	K-Nearest Neighbors
ML-KNN	Multi-label K-Nearest Neighbors
RL	Ranking Loss
LR	Logistic Regression
NB	Naïve Bayes
DT	Decision Tree
OVA	One-versus-All classifier

OVA	One-versus-One classifier
SVM	Support Vector Machines
LSA	Latent Semantic Analysis
LDA	Latent Dirichlet Allocation
P	Precision
R	Recall
HL	Hamming Loss
O	One-error
RL	Ranking Loss
F	F1-measure

1 Introduction

1.1 Bio-inspired Design

Bio-inspired design (also known as biomimetics, biomimetic design, or biomimicry) is the design based on nature inspiration. It is the interdisciplinary field of knowledge of the biological system and technological innovations in the engineering system [1, 2]. Stone et al. [3] state that the objective of bio-inspired design is to apply biological knowledge to tackle design challenges systematically. Therefore, bio-inspired design can enhance the creation of sustainable and innovative solutions to modern engineering problems. Bio-inspired structures and products often outperformed conventional engineering systems since biological systems have developed their solutions to deal with natural obstacles and challenges, and the solutions have been tested by millions of years of evolution: biological species have either adapted, optimized their living tactics, or they have vanished [1, 2, 4, 5]. Table 1.1 shows various engineering applications inspired by biological systems. Traced back to the 19th century, many products have arisen from this bio-inspired strategy of design and process-based biological inspiration [6, 7], including barbed wire, Tiffany lamps, the Wright glider, and the design of Central Park in Manhattan [7, 8].

Table 1.1: Examples of bio-inspired engineering applications

Biological System	**Engineering Applications**
Sharkskin	Aerodynamic drag reduction [9]
Lotus leaf	Superhydrophobicity [10]
Desert scorpion skin	Anti-erosion [11]
Snakeskin	Effective friction management [12]
Penguin spindle shape	Fluidic drag reduction [13]
Iris leaves	Stiffness [14]

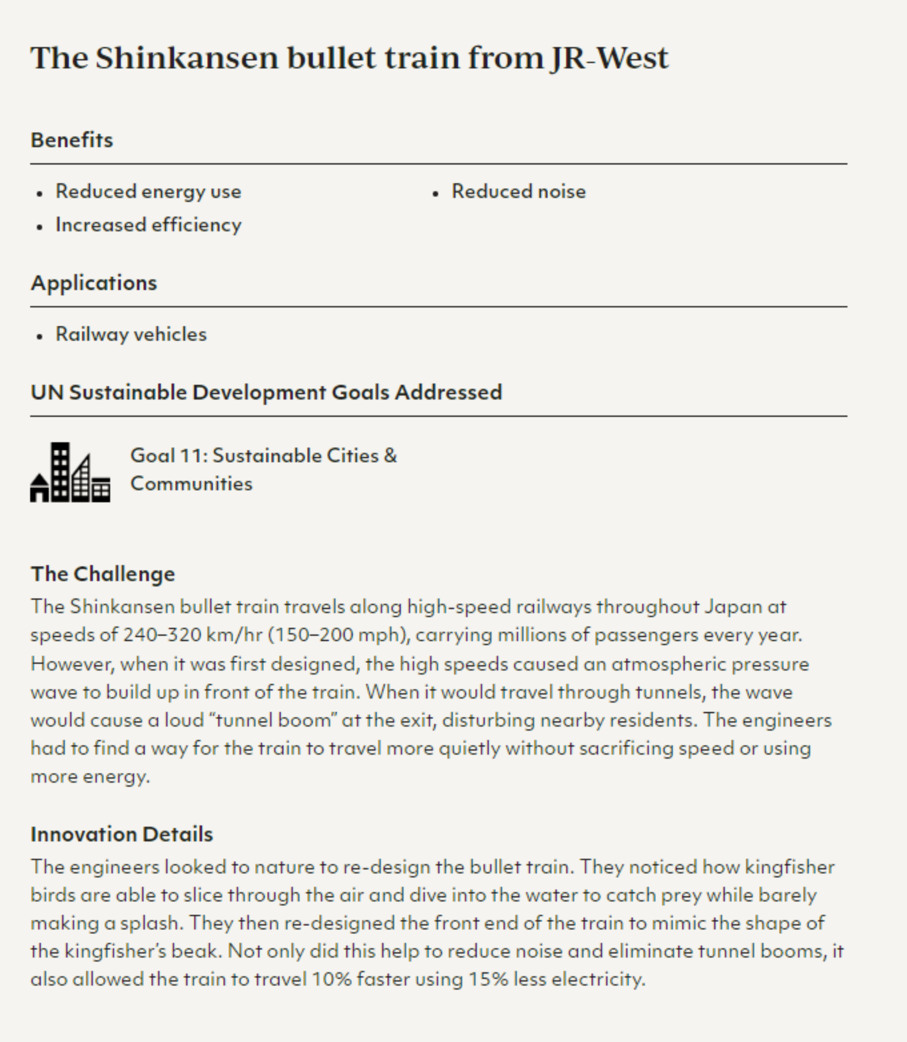
However, bio-inspired design is very challenging since it combines the knowledge of biological and engineering systems, which are two distinguished domains where research is typically carried out independently [1]. Throughout recent years, several definitions of bio-inspired design and its related terms have been reported in the recent literature. Most literature defines bio-inspired design as an approach that uses analogies of biological systems to derive innovative solutions to complicated engineering systems [15]. The collaboration between biologists and engineers requires an interdisciplinary understanding, which leads to the research on collaboration on biological knowledge transfer and ideation [1] as biological solutions require a step to achieve biological knowledge transfer to apply in the technical domain [1, 16-18]. However, most engineers do not have a background in biology. Therefore, it is important to equip engineers with biological knowledge as collaboration in this interdisciplinary field is growing. Some teaching approaches are explored by some researchers [1, 18, 19]. On the other hand, some researchers have worked on developing tools and techniques to support bio-inspired design without a strong background in biology, as understanding knowledge of biological systems requires too much time and effort for researchers [7].

Generally, there are two main directions for bio-inspired design research: the search for biological knowledge and the transfer of biological analogies [1]. Some databases for searching biological information are available online. One example of an online database is AskNature [1, 20-22]. AskNature is the primary online database used for this research. These databases aim to provide researchers with access to biological knowledge, as the understanding of biological knowledge is important. Figure 1.1 shows one example of the AskNature database. The inspiration for biological analogies in the development of engineering solutions is based on biological knowledge transfer. Salgueiredo and Hatchuel [23] state the significance of learning the biological and technical knowledge at the same time to develop bio-inspired designs [1], which is essential for the collaboration between engineers and biologists.

1.2 Bridging Biological Knowledge with Engineering Systems

Fu et al. [7] have characterized the existing state of engineering knowledge regarding biological analogies, aiming to establish a context in which bio-inspired design as an area of investigation should be explored. The importance of understanding the advantages and disadvantages of applying biological knowledge in the engineering design process is also discussed [23]. Take the examples listed in Table 1.1, the majority of engineering applications are problem-driven designs, as they combine bio-inspired design with a function-based method to develop a function-based bio-inspired design [3]. The function can serve as an analogy for the relationship between what the task of an engineering system is and how a biological system accomplishes that task. Biological systems can be modelled by using the standard functional modelling approach [3]. This modelling approach requires the study of the transfer of biological knowledge between engineering and biological system. One of the challenges of this approach is that engineers and biologists employ different terminologies when discussing functional solutions; thus, an engineering-to-biology thesaurus is required. The majority of contemporary bio-inspired designs make use of functional terms that refer to biological systems [3, 24].

Recently, the collaboration between mechanical engineers and biologists has been examined by some researchers [1]. These researchers observed how teams of biology and engineering students worked on a comprehensive engineering design project. Some observed that collaboration resulted in a more comprehensive understanding of biological knowledge. On the other hand, some discovered that the knowledge transfer between engineers and biologists can be hampered by disparate information and knowledge [1, 18, 25, 26].



The Shinkansen bullet train from JR-West

Benefits

- Reduced energy use
- Increased efficiency
- Reduced noise

Applications

- Railway vehicles

UN Sustainable Development Goals Addressed

Goal 11: Sustainable Cities & Communities

The Challenge

The Shinkansen bullet train travels along high-speed railways throughout Japan at speeds of 240–320 km/hr (150–200 mph), carrying millions of passengers every year. However, when it was first designed, the high speeds caused an atmospheric pressure wave to build up in front of the train. When it would travel through tunnels, the wave would cause a loud “tunnel boom” at the exit, disturbing nearby residents. The engineers had to find a way for the train to travel more quietly without sacrificing speed or using more energy.

Innovation Details

The engineers looked to nature to re-design the bullet train. They noticed how kingfisher birds are able to slice through the air and dive into the water to catch prey while barely making a splash. They then re-designed the front end of the train to mimic the shape of the kingfisher’s beak. Not only did this help to reduce noise and eliminate tunnel booms, it also allowed the train to travel 10% faster using 15% less electricity.

Figure 1.1: AskNature database example [20]

1.3 Research Objectives

Numerous solutions have been proposed and developed to support bio-inspired designs as well as to assist engineers or designers in gaining an understanding of bio-inspired design. These tools and methods include database, modelling, and heuristics tools and methods. However, these tools require specific training and some prior knowledge of biological systems, and they are less helpful in terms of knowledge extraction and tend to overwhelm designers with an abundance of irrelevant data. Bio-inspired design primarily focuses on emulating the biological principles observed in nature to solve modern complex engineering problems. Most of the bio-inspired design methods are designed for a single label function at a time. However, little work was done on methods to aid in emulating multiple functions at a time. The purpose of this research is to build a bio-inspired design classification system to classify the bio-inspired design from a multi-functional perspective to save the time of classifying the bio-inspired design and knowledge extraction. This research limits the searches on bio-inspired design to the AskNature corpus to reduce the scope of the biological system. Some knowledge extraction tools including natural language processing and text mining are integrated into the process of extracting useful information, filtering the data, extracting keywords from the corpus, feature extraction, and feature extraction. After the data has been processed, the training is done by presenting to the machine learning model a set of input data with corresponding output labels, and the classification system classifies the input data into the appropriate label.

The main objectives of this research are:

- To build a multi-label classification system to classify the bio-inspired design from a multi-functional perspective
- To conduct a supervised machine learning model on the bio-inspired design classification problem
- To integrate natural language processing, text mining techniques with a machine learning model for the bio-inspired design classification system

Some measures of success are set to access the quality of the proposed system, and they are listed below:

- Classifier evaluation metrics including accuracy, hamming loss, micro averaged precision, micro averaged recall, micro averaged F1-measures, macro averaged precision, macro averaged recall, and macro averaged F1-measures.

1.4 Thesis Structure

This thesis is organized in the following structure. Chapter 2 discusses the literature review on bio-inspired design and the various tools and methods for supporting bio-inspired design. The use of NLP and text mining to extract related information to support the bio-inspired design classification system is presented as well. In addition, the Domain-Integrated-Design (DID) method for bio-inspired design concept generation is addressed. Chapter 3 introduces the background of classification model as well as the development of classification in NLP and text mining techniques. Chapter 4 proposes the bio-inspired design classification system. This system combines multiple domains to extract engineering knowledge from bio-inspired designs and classify bio-inspired designs corresponding to their functions. In Chapter 5, several experiments are conducted on the collected dataset to evaluate the performance of the proposed system. The dataset, experiment setup, and various classifier algorithms used in the proposed system are described in detail, along with their associated results. Additionally, a discussion and analysis of the performance of the proposed system from various perspectives are presented. Finally, Chapter 6 summarizes the major findings of this research and points the way forward for future work on the bio-inspired design classification system.

2 Literature Review

This chapter reviews related research on bio-inspired design and the application of text mining and natural language processing to bridge the knowledge gap in bio-inspired design. The literature review begins with related works around and related directly to bio-inspired design. Tools and methods for supporting bio-inspired design are identified, including database, modeling, and heuristic tools and methods, the findings are synthesized into a summary of what is known and unknown in the current research. Finally, a discussion of the future direction for bio-inspired design is included.

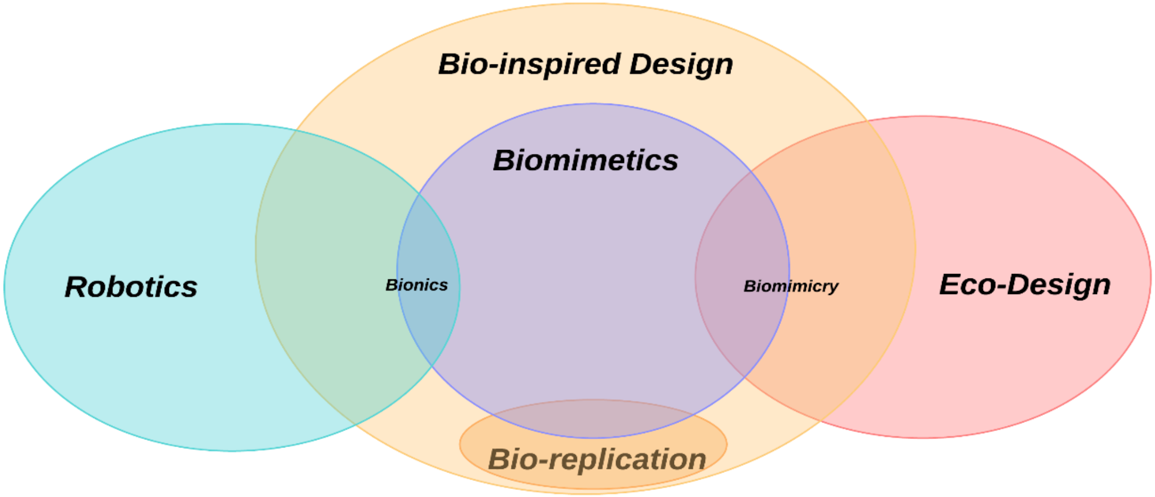


Figure 2.1: Bio-inspired design interchangeable terms [27]

Bio-inspired design, biomimetics, biomimicry, bionics, and bio-replication are all terms that refer to the process of using inspiration from living organisms to create technical artefacts. These bio-inspired design interchangeable terms are shown in Figure 2.1. While they are often used interchangeably, they are not the same, and they have their own definitions. The most general term amongst them is bio-inspired design, which is frequently used interchangeably with biomimetics. But bio-inspired design focuses more on the process of design [27]. Bio-inspired design makes use of biological analogies to discover answers for engineering problems in the field of engineering design [18, 19, 28] as it benefits from millions of years of natural selection and variety [29, 30].

Based on 346,000 papers found for the bio-inspired design and its interchangeable terms, bio-inspired design can be approached using various paradigms including: problem-driven bio-inspired design, solution-driven bio-inspired design, biomimicry, and bioreplication. Figure 2.2 shows the paradigms for bio-inspired design [27]. There are 81% unknown approaches for bio-inspired design in the research projects that need to be further investigated.

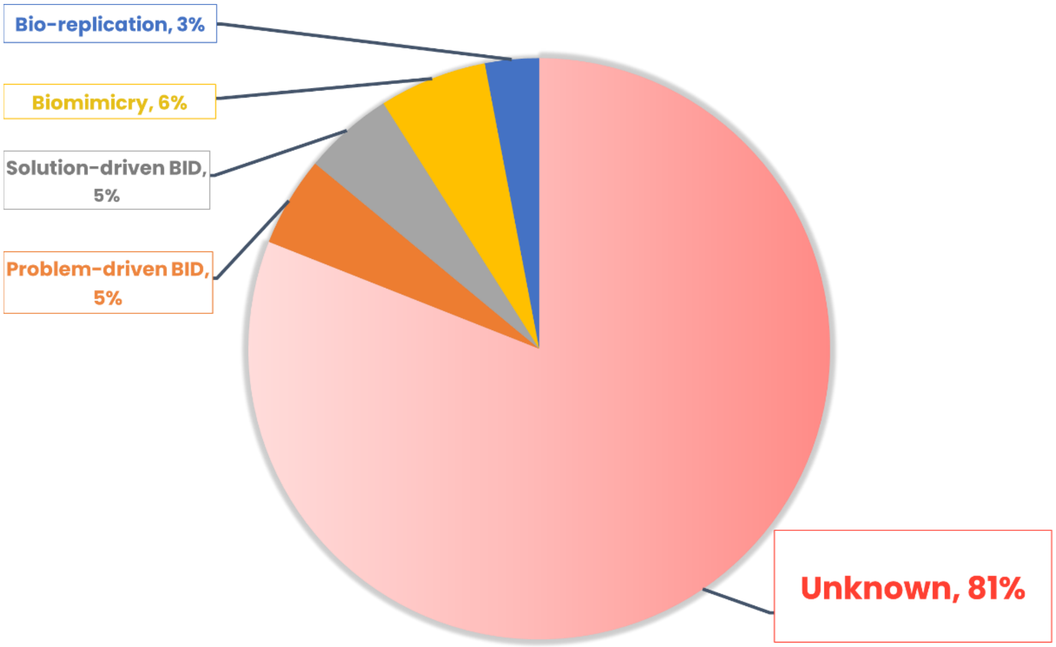


Figure 2.2: Bio-inspired design paradigm [27]

Two main different approaches to bio-inspired design consist of solution-driven approach and problem-driven approach [31, 32]. The solution-driven bio-inspired design begins with a specific biological system that provides a function that can be applied to technical solutions, such as Velcro's innovation [28, 33]. The problem-driven bio-inspired design starts from a real engineering problem, such as developing a new product or optimizing the existing product. Designers would transfer the biological knowledge into the engineering field and apply it to technical solutions. The problem-driven bio-inspired design is commonly used in design practice since it follows the problem-to-solution process of traditional engineering design [32, 34, 35]. Therefore, bio-inspired design is critical for problem-solving and conceptual design in the engineering system [36-38]. These two different approaches of bio-inspired design are also named as Top-Down approach and Bottom-Up approach [15].

Some researchers have studied the knowledge transfer between the biological system and engineering system since learning knowledge transfer is necessary for the collaboration between two fields [1, 23]. The ISO standard establishes the following criteria for determining whether the product is bio-inspired design: firstly a functional analysis of an existing biological system is conducted, then the biological knowledge transfer is abstracted into a model, finally the model is used to design the engineering product [27]. This indicates the biological knowledge transfer is critical for the collaboration between two fields. Figure 2.3 shows the knowledge transfer in bio-inspired design from Hesse's perspective [38]. The bio-inspired solution starts from establishing a similar problem in the biological system after finding out a problem in the technical system, a 'horizontal' relationship is built between biological and technical systems. Based on the existing solution to the biological system, the analogy abstracted solution is applied into the technical domain to develop a solution to the technical system, which is finally named as bio-inspired design. Bio-inspired design often consists of two domains of abstraction, including abstraction of the biological solution and abstraction of the technical problem [1, 39]. The example shown in Figure 2.3 is how praying mantis catches the prey inspires the solution to the system of attachment of pump cables, hose, and rope. The 'vertical' relationship is built from the problem to the solution level, the bio-inspired solution uses an analogy abstraction solution of spikes – cable strap with deformable elements [39].

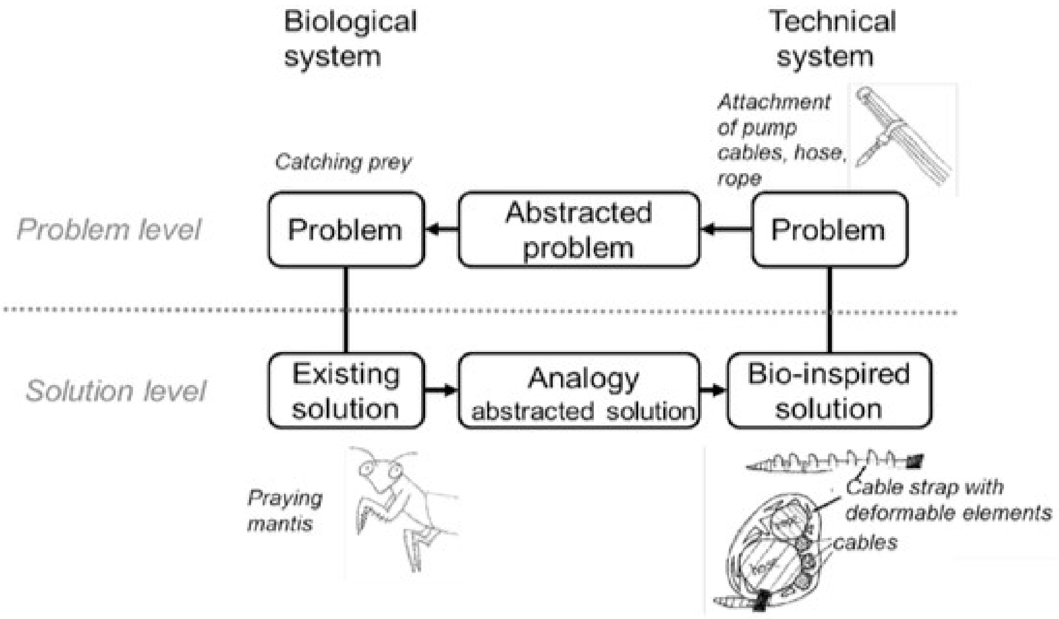


Figure 2.3: Knowledge transfer in bio-inspired design (based on Hesse's work) [1, 39]

2.1 Tools and Methods for supporting Bio-inspired Design

When designers make use the problem-driven bio-inspired design, there are some challenges including:

- Identifying the biological information related to engineering problems
- Extracting design knowledge from biological information

These challenges are mainly due to engineers lacking of biological knowledge or biologists lacking of design knowledge [1]. Some tools and methods for supporting bio-inspired design are developed for researchers to overcome these challenges. They are summarized here to understand how bio-inspired design is supported, to understand the underlying principles and concerns, and to make predictions on the extension of future bio-inspired design methods and tools. These tools and methods can be classified into three categories based on the features of application scenarios: database, modelling, and heuristics [28]. They are discussed in detail in the following sections.

2.1.1 Database Tools and Methods

Regarding the database tools and methods, AskNature [3] is an open-source online database developed by Biomimicry Institute. It provides over 1,700 strategies from biological systems. These strategies provide hyperlinks to Google Scholar publications or biologist Web pages to help the user to research the strategies. This database organizes biological knowledge according to the Biomimicry Taxonomy [40], which classifies the biological system based on its function. Identifying the function is a critical component of approaching bio-inspired design as function is one of the techniques for knowledge transfer between biological and engineering systems [18, 41, 42]. Figure 2.4 depicts the underlying structure of Biomimicry Taxonomy. It organizes biological knowledge into high-level, intermediate-level and granular functions, and some physical principles [7]. However, AskNature enables the user to search for biological knowledge by using specific keywords [31], rather than using the Biomimicry Taxonomy. Thus, it does not provide a mechanism for applying biological knowledge to the technical solution [43]. AskNature is still under development for tailoring to the needs of users [20]. In addition, the examination of using the Taxonomy to aid user in searching needs to be further analyzed [7].

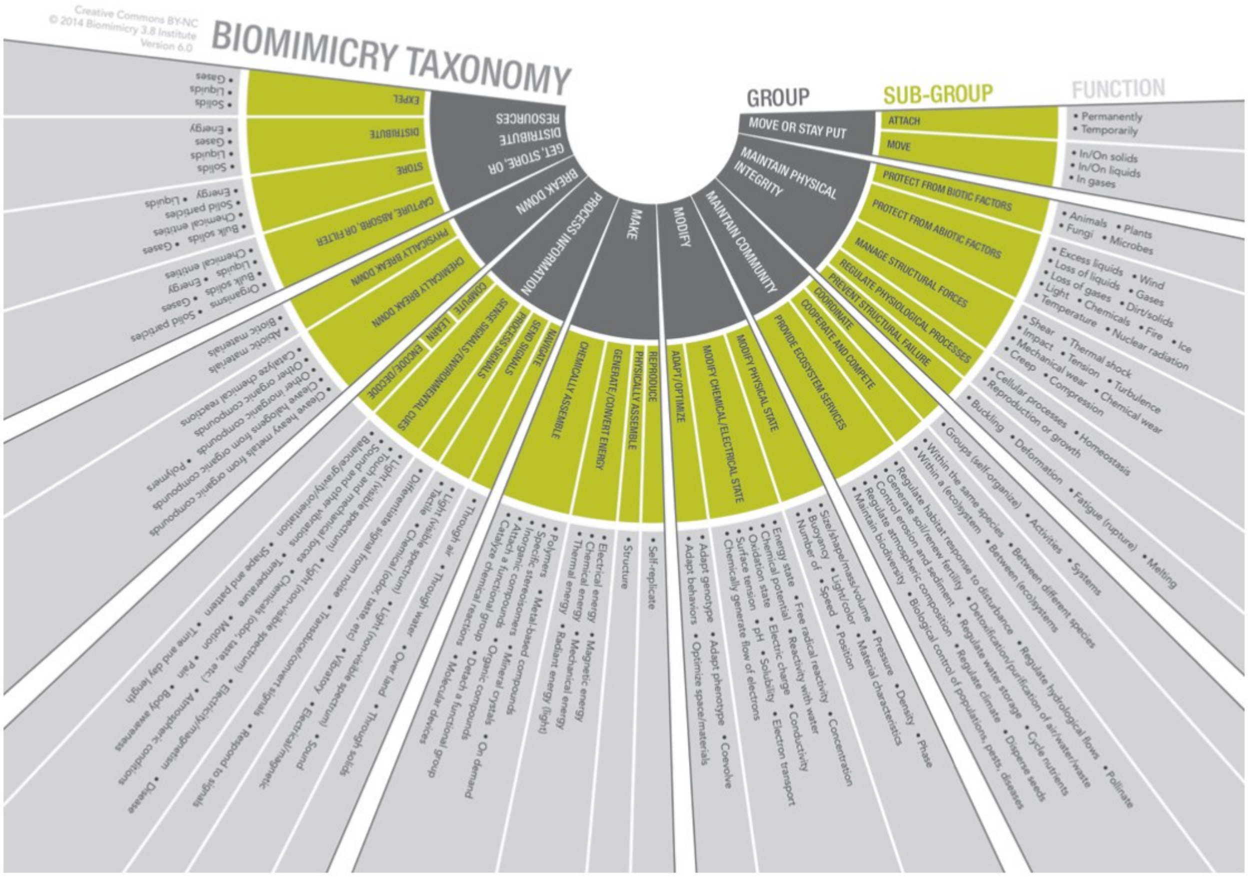

Creative Commons BY-NC
© 2014 Biomimicry 3.8 Institute
Version 6.0
BIOMIMICRY TAXONOMY
GROUP
SUB-GROUP
FUNCTION
GET, STORE, OR DISTRIBUTE RESOURCES
EXPEL
• Solids • Liquids • Gases
DISTRIBUTE
• Solids • Liquids • Gases • Energy
STORE
CAPTURE, ABSORB, OR FILTER
• Energy • Solid particles • Chemical entities • Bulk solids • Liquids • Gases
BREAK DOWN
PHYSICALLY BREAK DOWN
• Chemical entities • Liquids • Bulk solids • Energy • Organisms • Gases • Solid particles
CHEMICALLY BREAK DOWN
• Biotic materials • Abiotic materials
• Catalyze chemical reactions • Other organic compounds • Other inorganic compounds • Cleave halogens from compounds • Polymers • Cleave heavy metals from organic compounds
PROCESS INFORMATION
COMPUTE
LEARN
ENCODE/DECODE
SENSE SIGNALS/ENVIRONMENTAL CUES
• Balance/gravity/orientation • Sound and other vibrations • Touch and mechanical forces • Light (visible spectrum) • Light (non-visible spectrum) • Chemicals (odor, taste, etc.) • Temperature • Shape and pattern • Motion • Pain • Time and day length • Body awareness • Atmospheric conditions • Electricity/magnetism • Disease
PROCESS SIGNALS
• Differentiate signal from noise • Transduce/convert signals • Respond to signals
SEND SIGNALS
• Tactile • Chemical (odor, taste, etc) • Light (visible spectrum) • Light (non-visible spectrum) • Vibratory • Electrical/magnetic • Sound
NAVIGATE
• Through air • Through water • Over land • Through solids
MAKE
CHEMICALLY ASSEMBLE
• Polymers • Metal-based compounds • Specific stereoisomers • Inorganic compounds • Mineral crystals • Attach a functional group • Organic compounds • Catalyze chemical reactions • Detach a functional group • On demand • Molecular devices
GENERATE/CONVERT ENERGY
• Electrical energy • Magnetic energy • Chemical energy • Mechanical energy • Thermal energy • Radiant energy (light)
PHYSICALLY ASSEMBLE
• Structure
REPRODUCE
• Self-replicate
MODIFY
ADAPT/OPTIMIZE
• Adapt genotype • Adapt phenotype • Coevolve • Adapt behaviors • Optimize space/materials
MODIFY CHEMICAL/ELECTRICAL STATE
• Energy state • Free radical reactivity • Chemical potential • Reactivity with water • Oxidation state • Electric charge • Conductivity • Surface tension • pH • Solubility • Electron transport • Chemically generate flow of electrons • Concentration
MODIFY PHYSICAL STATE
• Size/shape/mass/volume • Pressure • Density • Phase • Buoyancy • Light/color • Material characteristics • Number of • Speed • Position
MAINTAIN COMMUNITY
PROVIDE ECOSYSTEM SERVICES
• Regulate habitat response to disturbance • Regulate hydrological flows • Pollinate • Generate soil/renew fertility • Detoxification/purification of air/water/waste • Control erosion and sediment • Regulate water storage • Cycle nutrients • Regulate atmospheric composition • Regulate climate • Disperse seeds • Maintain biodiversity • Biological control of populations, pests, diseases
COOPERATE AND COMPETE
• Within the same species • Between different species • Within a (eco)system • Between (eco)systems
COORDINATE
• Groups (self-organize) • Activities • Systems
MAINTAIN PHYSICAL INTEGRITY
PREVENT STRUCTURAL FAILURE
• Buckling • Deformation • Fatigue (rupture) • Melting
REGULATE PHYSIOLOGICAL PROCESSES
• Cellular processes • Homeostasis • Reproduction or growth
MANAGE STRUCTURAL FORCES
• Shear • Thermal shock • Impact • Tension • Turbulence • Mechanical wear • Chemical wear • Creep • Compression
PROTECT FROM ABIOTIC FACTORS
• Excess liquids • Wind • Loss of liquids • Gases • Loss of gases • Dirt/solids • Light • Chemicals • Fire • Ice • Temperature • Nuclear radiation
PROTECT FROM BIOTIC FACTORS
• Animals • Plants • Fungi • Microbes
MOVE OR STAY PUT
MOVE
• In/On solids • In/On liquids • In gases
ATTACH
• Permanently • Temporarily

Figure 2.4: Biomimicry Taxonomy, an underlying representation and search structure for AskNature [7]

To improve the outcome of knowledge transfer, DANE (Design by Analogy to Nature Engineer) [44] is constructed by Geol and Vattam [45] to utilize a platform that includes algorithms for representing, indexing and retrieving biological information in the form of a structure-behavior model. This database requires manual data entry. The representations of biological systems are subjective and they are dependent on the designer's knowledge on modelling [28]. Users may access the database using information included in DANE via a functional representation located in the library and the search results can be in a variety of multi-media formats [7]. DANE is useful to the problem-driven designers since they can use DANE to research the function addressed and related to the solution to a biological system. Biologue is another tool proposed by Goel [3], which uses a structure-behavior-function model. DANE and Biologue both require the users to access the software, and the expansion of the database library is still needed.

Vincent et al. [46] developed a Bio-TRIZ database, which is based on the framework of TRIZ [47]. They built the database by analyzing the patterns of conflict resolution discovered in approximately 500 biological occurrences. Then a contradiction matrix is developed by using the framework of TRIZ [28, 48]. The advantage of the Bio-TRIZ database is that biological analogies can be mapped to TRIZ concepts [48], and the conflict elements can change based on the user needs. However, the Bio-TRIZ database requires professional knowledge about biological systems and special training on TRIZ contractions/conflicts matrix.

There are also some works on developing database methods and tools for supporting bio-inspired design. Mak and Shu [49] developed a taxonomy of verbs that describes the relationships between biological and engineering designs. Nagel et al. [50] built a database of function-based models of biological systems. Linsey et al. [51] discovered that annotating graphics with functional information increases the likelihood of knowledge transfer from biological systems to engineering design [18]. These two approaches are distinct from traditional design approaches since it begins with a biological system and extract analogical elements from a biological system. More specialized biological texts should be investigated rather than the general text of biological systems.

2.1.2 Modeling Tools and Methods

For the modeling tools and methods, Goel et al. [52] constructed a Structure-Behavior-Function (SBF) model to understand complicated biological systems [28]. Nagel et al. [24] built the Engineering-to-Biology thesaurus, which translates the functional basis terms into biological knowledge and is based on the database of function-based model [28, 50]. This approach is different from the traditional approach since it begins with extracting analogies from biological systems into technical solutions. It enables engineers with limited biological knowledge to leverage nature's brilliance during the design process and increases the likelihood of developing technical solutions [24]. Engineering-to-Biology uses functional representations [53] to understand biological systems and enable designers to bridge biological knowledge with engineering systems. The functional basis for engineering design is referred to as the functional representation to support functional modeling, a consistent set of functional vocabulary enables this representation to produce comprehensive results [54]. The formalized representation in function-based is important for a variety of reasons. The first reason is that it can reduce ambiguity at the level of modeling. The second reason is that a corpus of standard terminology can reduce the ambiguity since the same term could have multiple meanings. The third reason is that it ensures the information exchange inside the function-based model is consistent, which simplifies the process of retrieving data for the function-based model searches [54]. These two methods both require knowledge and training in functional modeling, and the complicated relationships between biological and engineering systems still need to be investigated.

2.1.3 Heuristics Tools and Methods

Concerning the heuristics tools and methods, Chakrabarti et al. [55, 56] developed a computation tool SAPPHIRE to arrange the unstructured information for providing descriptions of structures, behaviors, and functions of biological and engineering systems used in bio-inspired design. This model can obtain seven layers of abstraction and verbs describing the design problems [18] to inspire bio-inspired design ideas and boost the likelihood of generating ideas [3]. Knowledge transfer of this model can be divided into four steps: identify search objectives, search for biological analogies, analyze knowledge transfer, and transfer targeted knowledge [3]. Figure 2.5 shows the seven levels of the model: State-Action-Part-Phenomenon-Input-Organ-Effect, which has been implemented in the software IDEA-INSPIRE [7]. However, not all layers of abstraction of SAPPHIRE have been investigated yet, and these tools require some learning to formulate design problems in terms of SAPPHIRE.

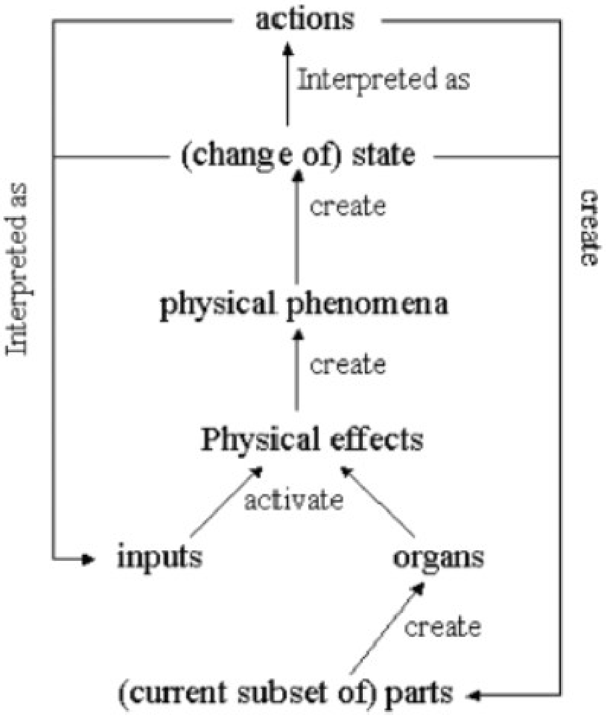


Figure 2.5: SAPPHIRE model to understand natural systems [55]

Helms and Goel [57] introduced the Four-box diagram, which is a two-by-two grid that enables the selection of biological systems in four dimensions. Moreover, Cheong and Shu [58] proposed a causal relation template that identifies how one function is enabled by another, this template aids in designers identifying causal relations in biological systems [28]. These heuristic tools enable the efficiency of knowledge extraction from bio-inspired designs and the selection of biological systems, also enables the knowledge transfer at various levels of abstraction using natural inspirations. However, these tools all require some prior knowledge of models or biological systems to learn or use.

2.2 Text Mining and Natural Language Processing for Knowledge Extraction

Knowledge extraction is the process of identifying unknown structures and valuable information from enormous volumes of data [59]. The objective of knowledge extraction is to extract implicit and potentially beneficial information from the data [60]. Extracting information from unstructured or semi-structured articles is becoming increasingly important as the number of articles is growing [61]. The extracted information should enable the machine and human to be able to understand the knowledge [61]. Most of the research on knowledge extraction has been conducted in the disciplines of biomedical, medicinal, chemical engineering, and material engineering [62]. Not much research has been done in the domain of bio-inspired design.

Several applications are using the knowledge extraction technique. Jebbor and Benhlima [59] summarized various knowledge extraction techniques that allow to find new correlations and analyze trends and models in question-answer (Q/A) systems, including IntelliServe, LaSIE (Large Scale Information Extraction), Quantum, and SBLD (System based on linked data) systems. The advantages of Q/A systems and the extraction algorithm and its limitations were also examined. IntelliServe can provide feedback and respond to customers automatically based on their requests [63], and classify emails and messages from the client using Bayesian classification and regular expressions. Bayesian classification assumes that the words that appear in a text are independent of each other, however, there is always a correlation between words or phrases in messages [59]. LaSIE is a system that consists of three processing phases: lexical preprocessing, parsing and semantic interpretation, and interpretation. This system has been used to generate semantic representations of the sentences using semantic rules that manage the components of a sentence’s structure identified by parsing [64]. At the stage of lexical preprocessing, a file containing a single article is passed to the lexical processor, which outputs a list of lexical graphs that the parser can utilize, as well as a symbolic representation of the original text in bytes that can be reconstructed later using markup [59]. LaSIE is an effective tool that uses a semantic network and translates the knowledge into the Domain Model. Quantum is a Q/A system that delivers the automatic response to questions and manages trans-linguistic documents [65]. This system is simpler to classify questions since it limits the number of classes, and the classes are strongly related to the syntax of questions [59]. In addition, an ontology-based knowledge extraction method called SBLD is developed to convert questions into structured queries based on ontologies [66]. This system consists of three steps: extraction of triple patterns, extraction of entities and properties, and answer extraction. RDF data model and related ontologies are used to convert the sentences in triples form: [subject:] [predicate:] [object:]. RDF model has been widely used for NLP. Most of the techniques for knowledge extraction are the integration of NLP, text mining, IE (Information Extraction) and ST (Semantics Technology). The systems mentioned above demonstrate how knowledge extraction techniques work for various scenarios and they can be served as the guideline for knowledge extraction for bio-inspired designs in the future. Since all of these systems are based on the classification system at different levels of functionalities and techniques used in knowledge extraction. Bio-inspired design classification system can use these techniques to extract useful and important information related to functionalities in order to classify the bio-inspired design according to their corresponding functionalities. Description of text mining and NLP techniques that mainly used for the proposed system in this research will be introduced in the following sections.

2.2.1 Text Mining

Text mining is a technique that extracts useful information or knowledge from various text documents. It is similar to data mining, except that data mining is concerned with structured data, whereas text mining is concerned with semi-structured and unstructured data [67]. Text mining is a multidisciplinary field with different techniques developed through the recent years, including association mining [68, 69], decision tree [70], Machine learning and rule induction method. Text mining can be summarized into three categories: keyword-based, statistically based, and linguistic-based methods. With the input of the keyword-based method being the character strings based on the keyword chosen from the text, the statistics method builds a machine learning model to manage the text while the linguistic-based method is part of the NLP [71]. According to recent surveys, approximately 80% of data in the industry is saved in textual format [72]. While most of the data is available online, it is not completely utilized since the data might not be available at the right time and in the right manner due to the enormous number of collected data. This problem is named as rich data, poor information [73]. Text mining is one of the techniques to filter the knowledge that can enhance productivity to solve this issue.

The core methodologies for text mining are information extraction and natural language processing [71]. Some key research efforts are summarized as follows. Rajman et al. [74] combined probabilistic association of keywords with prototypical document examples to extract knowledge from text collections. Peter Clark et al. [75] proposed a method that constructs tuples datasets based on the simple word knowledge extracted from text documents. Discovery and Aggregation of Relations in Text (DART) system is used in this method. This system can improve parsing, the assessment of the plausibility of textual entailment rules and the guideline of the disambiguation [61]. However, the DART system only produced the output as triples form (subject-verb-objective phrases), and in less normalized form since triple forms are always unprocessed as words. Therefore, great performance on the post-preprocessing is required for the DART system [75]. Cheong and Shu [76] proposed a method that used syntactic parsing to extract functional-related knowledge from biological sources. The causal-relation retrieval method allows designers to be able to extract biological analogies as the sentence structure from enormous bio-inspired design resources. The advantage of this method is that it can fasten the process of identifying valuable biological analogies from bio-inspired design and support automatic knowledge extraction from biological systems into the engineering field. However, there are still some limitations. This method has poor performance on identifying causally related functions from multiple sentences. It cannot distinguish the causally related functions that involve the conjunction 'and'. Furthermore, the proposed methodology used for this research is based on this knowledge extraction method [74].

Some scientific documents, unlike standard publications or articles, are written in a specific language that requires domain knowledge to understand the structure. When designers want to apply text mining in the bio-inspired design domain, it requires customization of the text mining model and the construction of an appropriate training dataset to meet customer requirements. A general scientific document text mining consists of the following steps: document retrieval and conversion from PDF into plain text, text preprocessing, data structure and normalization, as seen in Figure 2.6. The output of the text mining pipeline can be used in further sectors of the application and analysis [62].

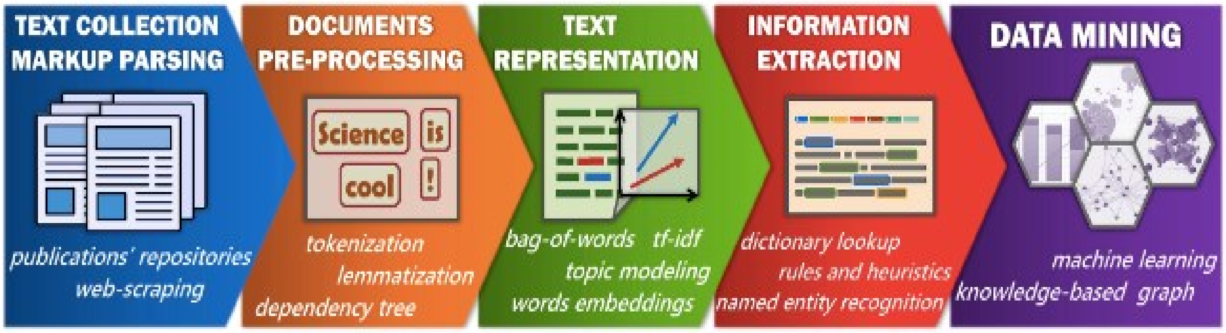


Figure 2.6: Schematic representation of the standard text mining pipeline for information extraction [62]

The goal of bio-inspired design text mining is to transfer the burden of information overload from researchers to the computer allowing researchers to identify required information more efficiently by applying data management methods to the biological and engineering knowledge in the literature [77]. The text classification system is one typical task of text mining, which aims to automatically identify whether a document or part of a document possesses particular interests based on a document discussing a specific topic [77]. In the related work, Donaldson et al. [78] used a support vector machine (SVM) trained on the terms in MEDLINE abstracts to reduce the size of data before locating the protein-protein interactions data into their Biomolecular interaction network database (BIND). The bag-of-words approach is used with an SVM classifier to classify the 100 abstracts. Their system achieves a precision of 96% with a recall of 84% and can reduce the number of abstracts that the curators have to read [77]. Alkahtani et al. [79] extracted manufacturing faults from customer feedback and warranty databases using an ontology-based data mining strategy which is a self-organizing map approach was used to extract information from the database and associate it with the manufacturing data [28]. They enhanced the quality of product and decreased expenditures. These examples demonstrate how text mining has lowered the knowledge threshold for researchers and provided a guideline for the future direction for bio-inspired designs text mining [28].

2.2.2 Natural Language Processing

Natural language processing (NLP) is a technique to show how the computer understands the human language, and it is a component of text mining that analyzes the linguistics of data to help the machine comprehend, manipulate, and communicate using the natural language [80]. Text mining is different from NLP because NLP aims to comprehend the meaning of the whole text; while text mining focuses on solving a particular problem in a predefined specific domain [77]. The semantics and syntax in the text are not considered in text mining. However, semantics and syntax such as the grammatical part of speech and lexical relations are significantly important in natural language processing. The topic of NLP integrates tools and techniques from various fields, including Artificial Intelligence (AI), linguistics, and computer science [81, 82]. NLP has a wide variety of applications, ranging from speech recognition to cross-language information retrieval [80]. Machine learning techniques are often used for natural language processing tasks. And it is mainly focused on the semantics and syntax of textual data. Different machine learning should be developed to address natural language processing challenges with a variety of objectives. Currently, hybrid machine learning systems are always used for NLP tasks, as different patterns and rules must be introduced to the main machine learning system to accommodate future perspectives from the researchers.

NLP consists of two types of approaches: the rationalist approach and the empirical approach [83]. Most early approaches were relying on sets of hand-coded rules [84]. Noam Chomsky proposed a generative grammars method based on the rationalist approach to NLP in the 1950s [83]. Since the 1990s, the empirical approach is preferred by researchers in which rules are discovered by observing a corpus of the representation knowledge [84]. Statistical or machine learning methods are often used for the empirical approach.

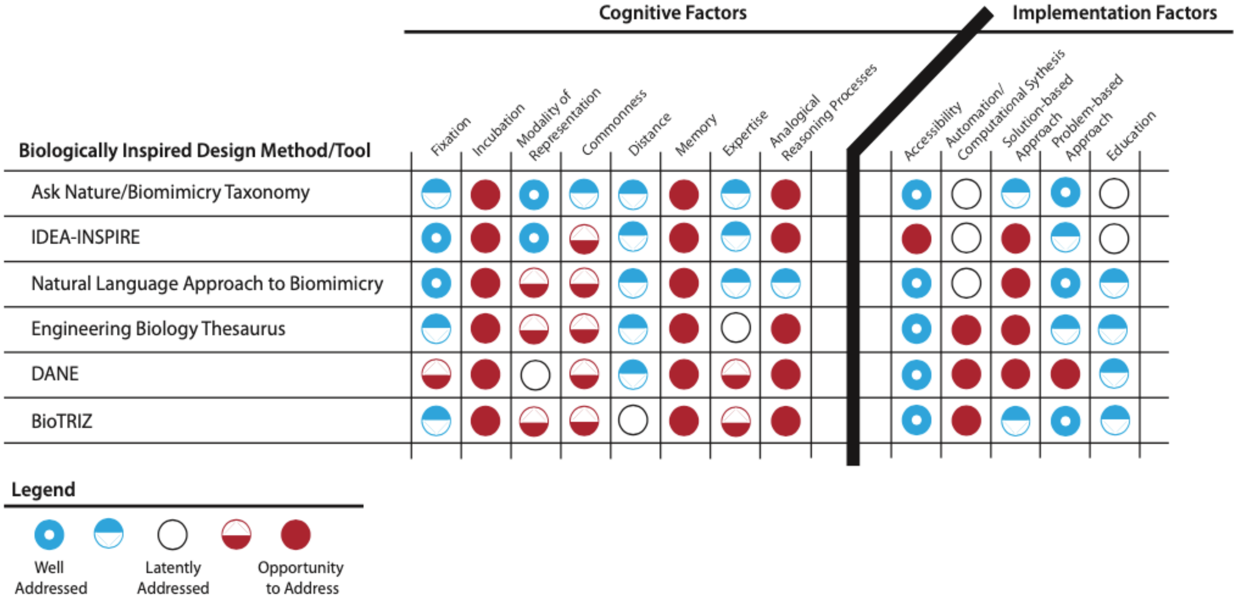
Related to bio-inspired design, Cheong et al. [85, 86] identified the design-by-analogy as an effective strategy for generating new ideas since the biological system provides a variety of analogies. They developed an approach to connect Function Basis terms with biological analogies to provide designers with useful terms that facilitate effective search within the biological knowledge and engineering system. This method allows the original functional keyword to be selected, keywords are expanded using hypernyms, synonyms, and troponyms, and to organize and iterate on search results through the text [85, 87]. There are several challenges using this method. The first is the fixation on certain terms or phrases within biological systems and the second is the difficulty of directly transferring biological functions into technical solutions for designers [7]. Therefore, designers require more explicit guidelines for performing analogy transfer based on the causal relationship in the biological system, as it is critical to identify the related biological knowledge to support the knowledge transfer from biological systems to engineering design problems [7]. To understand the biological knowledge transfer in bio-inspired design, Cheong et al. [88] discovered that novice designers tend to map specific stimuli features, rather than identifying an overall analogy and utilizing them in many ways. Subsequently, Cheong et al. [89] explored the analogical reasoning process of using bio-inspired design in problem-solving, including problem analysis, biological phenomenon discussion, referring to current solutions, developing new solutions, and evaluating solutions. This is different from the discovery found by Cheong et al. in the earlier research because the design evaluation and critical thinking require strategy-level biological knowledge transfer [7].

Several machine learning techniques have been introduced to enhance the design-by-analogy process for bio-inspired designs. Tools such as D-Apps and DRACULA, PAnDA, SEABIRD and Focused Open Biology Information Extraction (FOBIE) use Function Basis and academic as well as scientific data to extract relevant biological terms [90]. Furthermore, tools such as InnoGPS, WordNet, TechNet use NLP techniques to extract relevant terms and components from patent databases, research articles and publications. These strategies are based on the principle of using semantic network representation for ideation generation and conceptual design for bio-inspired designs.

The behavioural biologist George Kingsley Zipf was the first to establish a relationship between behavioural factors and frequencies of the word selection [91]. In 1949, he stated that humans can use the fewest possible words to express the most information by repeating the same keyword to describe a concept. Therefore, the word with a high frequency of occurrence conveys the most meaning [92]. Hans P. Luhn attempted to automate the process of frequency counting. Current research on the frequency of English words examinations of the British National Corpus (BNC) [93], which is a collection of standard English text that many researchers in linguistics and NLP utilize as a test corpus [83]. Some NLP research has demonstrated that frequency is not the only predictor of text meaning, it is also the most used and great automated indexing strategy in the information system [91].

2.3 Domain-Integrated-Design method for bio-inspired design concept generation

Based on cognitive and implementation factors, some bio-inspired design tools and methods are summarized in Figure 2.7 after the evaluation of each method/tool [7]. The summary of the nomenclature and definitions for these factors are shown in the Abbreviations.



	Cognitive Factors								Implementation Factors				
Biologically Inspired Design Method/Tool	Fixation	Incubation	Modality of Representation	Commonness	Distance	Memory	Expertise	Analogical Reasoning Processes	Accessibility	Automation/ Computational Synthesis	Solution-based Approach	Problem-based Approach	Education
Ask Nature/Biomimicry Taxonomy	◓	●	◉	◓	◓	●	◓	●	◉	○	◓	◉	○
IDEA-INSPIRE	◉	●	◉	◒	◓	●	◓	●	●	○	●	◓	○
Natural Language Approach to Biomimicry	◉	●	◒	◒	◓	●	◓	◓	◉	○	●	◉	◓
Engineering Biology Thesaurus	◓	●	◒	◒	◓	●	○	●	◉	●	●	◓	◓
DANE	◒	●	○	◒	◓	●	◒	●	◉	●	●	●	◓
BioTRIZ	◓	●	◒	◒	○	●	◒	●	◉	●	◓	◉	◓

Legend

◉ ◓ ○ ◒ ●

Well Addressed — Latently Addressed — Opportunity to Address

Figure 2.7: Visual summary of bio-inspired design tools and methods [7]

Bio-inspired design primarily focuses on emulating the biological principles observed in nature to solve modern complex engineering problems. One of the challenges for designers is to extract useful knowledge from bio-inspired designs since the biological system and engineering system can achieve similar functions by completely different mechanisms [1]. Most of the bio-inspired design methods emulate one function at a time. However, little work was done on methods to aid in emulating multiple functions at a time. A new design methodology called Domain Integrated Design (DID) was introduced at the Additive Design and Manufacturing Laboratory (ADML) to aid designers in emulating multiple functions at a time. The DID method involves classifying the identified biological features into various domains, the integration of which aids in designing multifunctional and multiscale structures and products. The domains are surfaces, cellular structures, cross-sections, and shapes. The observed biological feature that performs a particular function is categorized into one of these domains, and the integration of these features from the domains results in multifunctional and multiscale products. For example, a new design of suture pin leg/needle that would reduce the pain in the patients is the result of the integration of the rotational parabolic cross-section inspired from kingfisher's beak from cross-sections domain and the barbs from porcupine quill from the surface domain [15]. Figure 2.8 shows the design of the suture pin leg/needle. This design addresses the importance of the multi-functionality aspect of bio-inspired design.

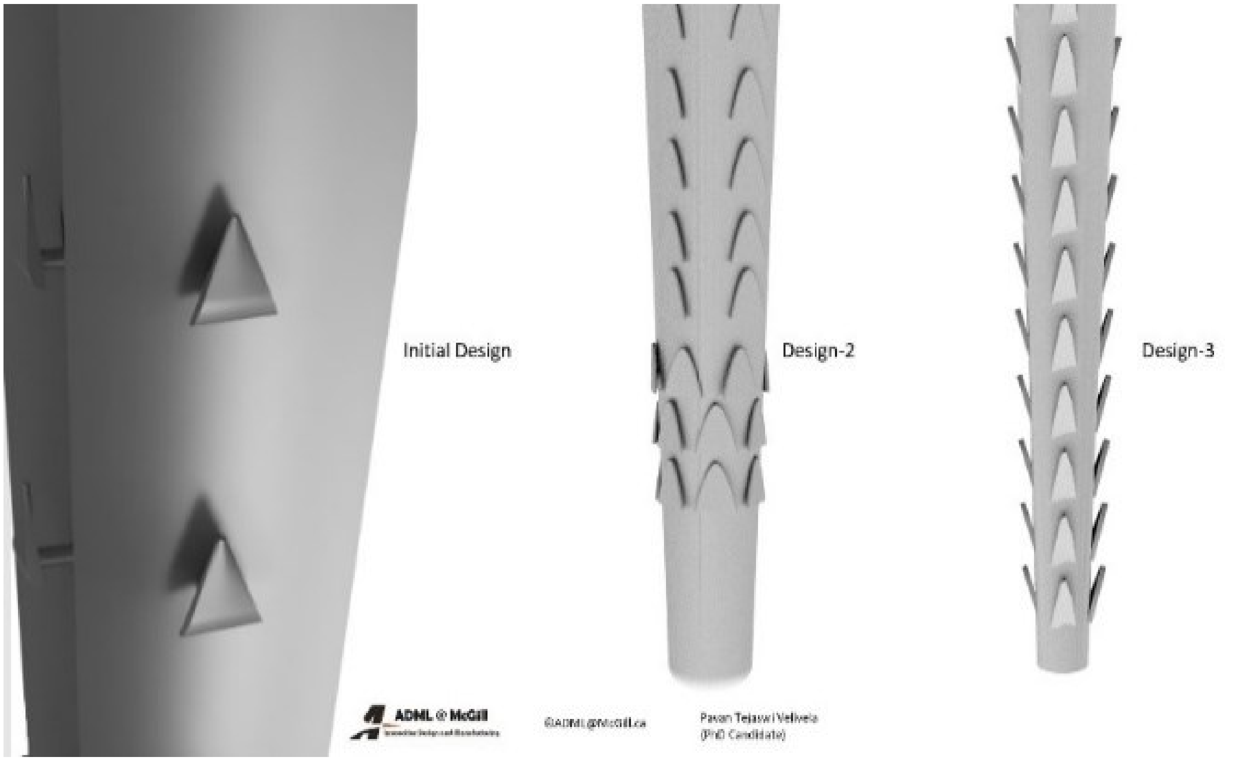


Figure 2.8: Bio-inspired suture pin leg/needle [15]

However, the search and emulation of the relevant biological systems for DID method is often a challenge and requires a human interpretation. Therefore, there is a need to develop a method to extract biological knowledge from bio-inspired designs. NLP and text mining techniques are often used to extract biological functions and mechanisms from patent databases, scientific journals, and articles. These techniques are also used in this study to extract the biological terms and components related to other scientific disciplines from a variety of information sources. The tools and methods for supporting bio-inspired designs demonstrate different tasks in bio-inspired designs and build a strong foundation for supporting bio-inspired designs with diverse approaches. But the majority of tools do not support the automatic extraction of multi-functional knowledge in bio-inspired designs. This research project aims to closes this gap by introducing a knowledge extraction tool based on a multi-label classification system to extract the relevant biological features into engineering domains to enhance the effectiveness of the DID method. The proposed multi-label classification system can classify different bio-inspired designs based on their functions, which leads to the further analysis of engineering domains in the DID method. Also, it can reduce the time of extracting the knowledge from bio-inspired design database. To our best knowledge, our work is the first to integrate the NLP, text mining techniques and machine learning model for multi-functional bio-inspired design to support DID method.

Another challenge is that biological knowledge often incorporates descriptions that are irrelevant for bio-inspired designs, such as common sense [28]. It is necessary to filter out design knowledge extracted from the descriptions. Therefore, a keyword extraction method is developed to collect keywords associated with biological knowledge and utilize them to filter unnecessary information. Based on the above tools and methods, background on classification model is discussed in Chapter 3 to classify the bio-inspired design corresponding to their design functionalities so that designers or engineers can retrieve the desired knowledge from the biological system.

3 Background on Classification Model

This chapter introduces the background of models that are used for the classification task. Also, the development of classification in NLP and text mining are presented with current studies. The algorithm of classification techniques is demonstrated as well.

Classification model belongs to machine learning, and it is one subdivision of supervised learning while another subdivision is regression. The difference between them is that classification predicts categorical values while regression predicts the output as a real value. There are four main types of classification tasks, including binary, multi-class, multi-label, and imbalanced classification as shown in Figure 3.1.

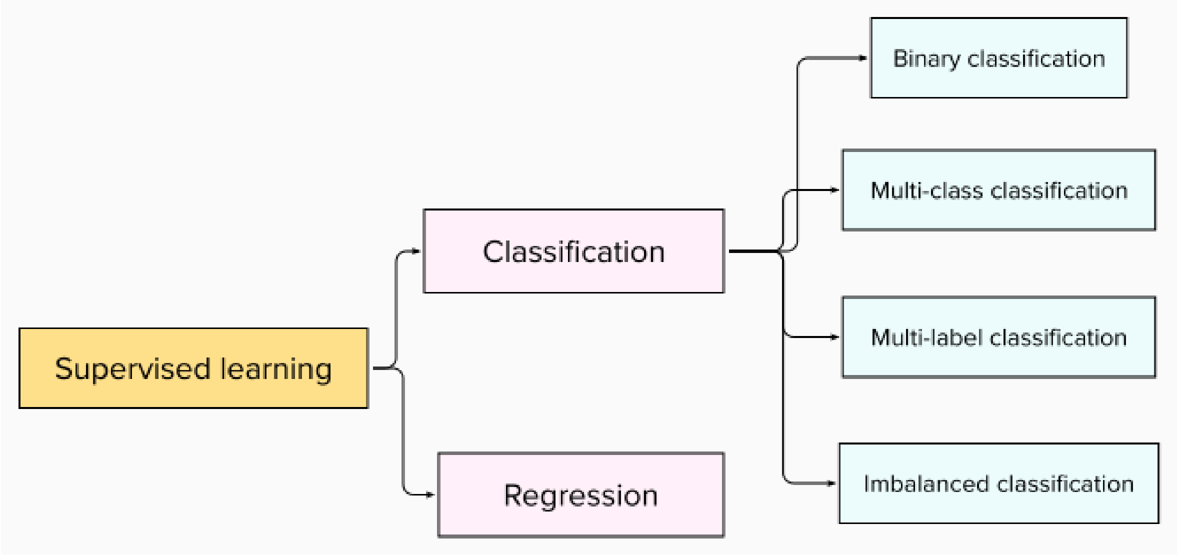

Supervised learning
Classification
Regression
Binary classification
Multi-class classification
Multi-label classification
Imbalanced classification

Figure 3.1: Supervised learning subdivisions

A classification model in machine learning is developed to classify future datasets into specific classes or categories by leveraging machine learning algorithms on the training dataset. The classifier is the algorithm to categorize the input variable into the target class label. Also, the feature vectors in the training sets are assumed to be independently and identically distributed in supervised learning to simplify the task. For the four types of classification, binary classification is a task to predict the output within only two class labels. A typical binary classification task is true/false classification. Multi-class classification is a task with more than two classes, and each feature vector is only assigned to one class label. Multi-label classification is a task to classify one input variable into more than one class label. It is closely linked to the multi-output classification problem. Imbalanced classification is a task with imbalanced data where the distribution of data is not equal.

According to the Oxford English Dictionary (OED) [94], the definition of classification is the action of classifying and categorizing objects according to their common characteristics and assigning them to the appropriate class [94, 95]. A class can be defined as a finite discrete set of objects that are identified by a specific class label which is an arbitrary descriptor for the set. The main objective of classification is to assign labels to the objects or predict categorical or binary values given a set of objects. The objects can be numerical, categorical, and ordinal values. In classification model, the objects are known as instances or features. Some common classification tasks include text classification, image classification, prediction of performance, etc. For example, considering the task of sentiment analysis, the feature vectors might be occurrence frequencies of positive, negative, and neutral words, all possible values of feature vectors are referred to as 'feature space'. The representation of some notations is shown in Table 3.1.

Table 3.1: Classification model parameter representations

Notation	Representation
X	Feature space
Y	Set of class labels
x	Feature vectors
y	Class label

Given the above notations, the classification function is used to map the input variables (feature space) to the output (set of class labels). The form of the mapping function is:

$$f: X \rightarrow Y \qquad \text{Equation 3.1}$$

Clustering is closely related to classification, most techniques used in classification can apply to clustering as well. The difference between classification and clustering is that the set of class labels is identified before the learning procedure of classification while clustering induces the class label during the learning process of clustering [96]. Clustering is an unsupervised learning task because it does not require training data and target function.

3.1 Multi-label Classification

Inspired by the multi-functional remarkable natural principles, the bio-inspired design presents how multi-functionality is addressed in engineering applications [97]. Most of the current bio-inspired designs solve single function problems by searching for biological analogies, there are few types of research about the multi-functional bio-inspired designs. For mechanical engineers, the design problem can be broken into functions with varying degrees of abstraction and means associated with each function [97]. To handle the difficulty of multi-functional means in bio-inspired design, this research proposed a classification system to categorize the functions within bio-inspired designs. Since most bio-inspired designs can realize more than one function, the classification tasks presented in this research are all multi-label classification.

Multi-label classification aims to simultaneously assign attributes into more than one label [98, 99]. Shore and Johnson [100], and De Boer et al. [101] proposed some prediction methods to weigh the training loss function with external knowledge. However, identifying external knowledge for prediction is hard in the real-world multi-label classification task. Yang [102] chooses the thresholds that produce the best evaluation measure on a validation set, whereas Lewis et al. [103] develop a cross-validation method to identify the threshold that gives the best performance. Multi-label classification methods can be divided into two categories: problem transformation methods and algorithm adaption methods [99]. There are a variety of classification algorithms under these two categories, this section will present several classifiers for each method. These classifiers are Binary Relevance One-Versus-All (BR-OVA), Classifier Chains (CC), Label Powerset (LP), and Multi-Label K-Nearest Neighbor (ML-KNN). Each of the classifiers uses different methods to classify the bio-inspired design and they will be explained in the following subsections.

3.1.1 Problem Transformation Methods

The main objective of problem transformation methods is to convert the multi-label classification tasks into one or more single-label classification tasks. Then, conventional single-label methods can be used for feature selection, the chosen features are incorporated into the original multi-label dataset to be evaluated by a multi-label classifier [104]. Some representatives of problem transformation methods consist of Binary Relevance One-versus-One (BR-OVO), Binary Relevance One-versus-All (BR-OVA), Classifier Chains (CC), and Label Powerset (LP).

Before starting the concept of binary relevance, Table 3.2 lists possible notations throughout the multi-label classifier along with their descriptions.

Table 3.2: Notations for Binary Relevance and Classifier Chains

Notations	Descriptions
X	Domain of dataset instance space: $\{x_1, x_2, x_3, \dots x_n\}, 1 \leq i \leq n$
Y	Set of distinct label-sets: $\{Y_1, Y_2, Y_3, \dots Y_m\}, 1 \leq s \leq m$
T	Multi-label training dataset: $\{(x_1, Y_1), (x_2, Y_2), (x_3, Y_3), \dots. (x_n, Y_m)\}, x_i \in X, Y_s \subseteq L$
L	Set of all class labels: $\{y_1, y_2, y_3, \dots y_q\}, 1 \leq j \leq q$
$Rank()$	The ranking function
D	Multi-label dataset
Z	Set of labels that are predicted by the multi-label classifier

- Binary Relevance (BR)

Binary Relevance (BR) is the widely used technique for the problem transformation method [105]. Binary Relevance consists of two approaches: Binary Relevance One-versus-One (BR-OVO) and Binary Relevance One-versus-All (BR-OVA). It converts a multi-label problem into q-binary problems by treating each label's prediction as an independent binary classifier [106]. In other words, it converts the original multi-label dataset into q single-label dataset, each dataset contains all occurrences of the original multilabel dataset, then trains a classifier on each of these datasets. Binary Relevance One-versus-All (BR-OVA) splits a multi-label classification into a binary classification problem per label. One binary classification model can only predict one class label; therefore, BR-OVA makes the predictions based on the most confident model by using binary relevance. Normally, a logistic regression model is fitted in this model as a baseline classifier for multi-label classification. Another common approach in BR is Binary Relevance One-versus-One (BR-OVO). BR-OVO constructs one classifier per each pair of classes and selects the class with the highest prediction score. Figure 3.2 shows the difference between BR-OVA and BR-OVO. The advantage of BR-OVA is its interpretability. Inspecting its matching classifier can provide information about the class, also it is conceptually straightforward and relatively fast [106]. However, it ignores the correlations between labels.

Binary Relevance One-versus-All classifier evaluates the system's performance on each class label separately before returning the mean value across class labels. BR algorithm used in this classifier aims to solve multi-label problems by translating them into well-known learning scenarios. Let $X = R^d$ represents the d-dimensional instance space, $L = \{y_1, y_2, y_3, \dots y_q\}$ represents the class label sets shown in Table 3.2. Given an unseen instance $x^* \in X$, its respective label set Y^* is predicted as $Y^* = f(x^*) \subseteq L$. BR converts the original multi-label dataset into q independent binary datasets. For the single class label y_j, BR creates a binary training set D_j from the original multi-label dataset D shown in Equation 3.2. Each multi-label training example (x^i, y^i) is turned into a binary training example based on its relevance to the class label y_j. For Equation 3.3, the label set for a new instance can be identified by querying the outputs of each binary classifier. If the classifier outputs a negative value, the predicted label set will be an empty set.

$$D_j = \{(x^i, y_j^i) | 1 \le i \le m\} \qquad \text{Equation 3.2}$$

$$Y^* = \{\lambda_j | g_j(x^*) > 0, 1 \le j \le q\} \qquad \text{Equation 3.3}$$

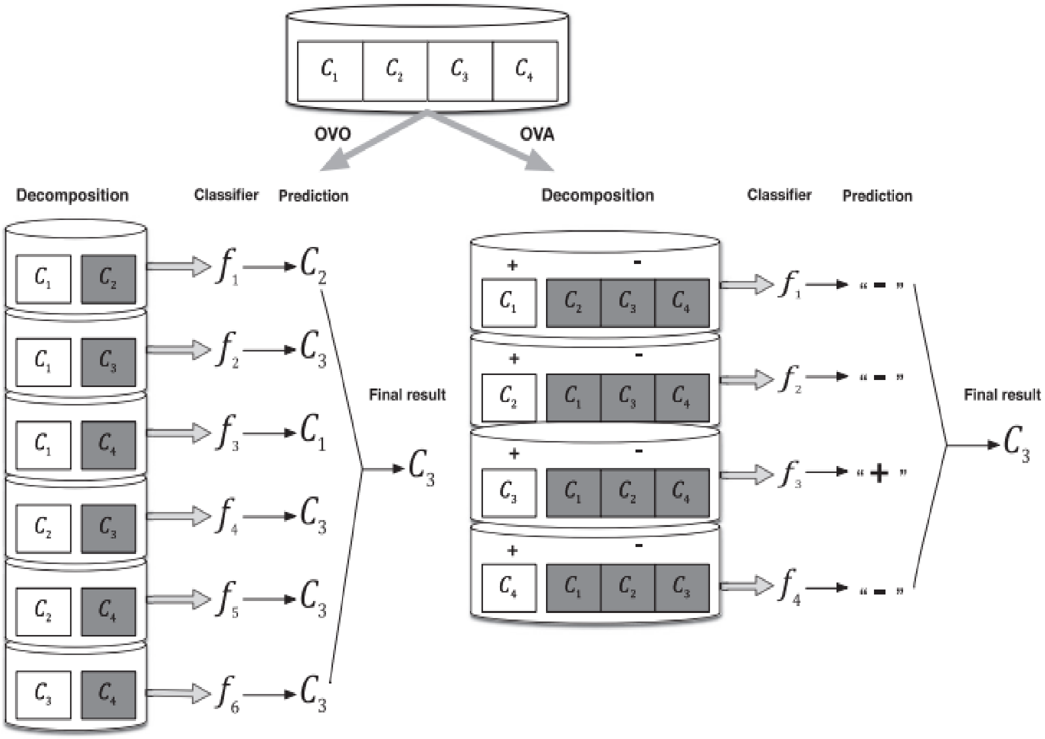


Figure 3.2: One-versus-All and One-versus-One decomposition schemes [107]

- Classifier Chains (CC)

Classifier Chains (CC) employ the same technique as BR. Suppose an instance x with a single label, the multi-label classification task connects each instance with a subset of labels $S \subseteq L$. Classifier Chains use $|L|$ binary classifiers as in binary models. A chain of classifiers is connected with each other, and classifiers on the chain address the BR problem associated with a particular label $l_j \epsilon L$. The 0 or 1 label connections of all preceding links are added to the existing feature space of each link [108]. The training process for dataset D and label set L is shown as Equation 3.4. For a training instance (x, S), where $S \subseteq L$ is represented by binary feature vector $(l_1, l_2, \cdots, l_{|L|}) \in \{0, 1\}^{|L|}$, x is an instance feature vector [108].

```
$Training(D = \{(x_1, S_1), \cdots, (x_n, S_n)\})$
$for\, j \in 1 \cdots |L|$
    $do \vartriangleright single - label\; transformation\; and\; training$
        $D^{'} \longleftarrow \{\,\}$
        $for\, (x, S) \in D$
            $do\, D' \leftarrow D' \cup ((x, l_1, \cdots, l_{j-1}), l_j)$
        $\vartriangleright train\; C_j\; to\; predict\; binary\; relevance\; of\; l_j$
        $C_j : D^{'} \rightarrow l_j \in \{0,1\}$
```

Equation 3.4

Therefore, a chain of binary classifiers C_1, $\cdots$, $C_{|L|}$ is identified. Each classifier C_j in the chain is in charge of predicting the binary association of label l_j given the feature space, with the help of all previous binary relevance predictions in the chain $l_1, l_2, \cdots l_{j-1}$ [108]. The classification process of CC starts from C_1 and follows along the chain, C_1 will predict $P_r(l_1|x)$, and $C_{|L|}$ will predict $P_r(l_j|x_i, l_1, l_2, \cdots l_{j-1})$. The predication phase of classifier chain for a test instance x is shown in Equation 3.5.

```
$Classify\,(x)$
  $Y \leftarrow \{\}$
  $for\, j \leftarrow 1\; to\; |L|$
      $do\, Y \leftarrow Y \cup (l_j \leftarrow C_j : (x, l_1, \cdots, l_{j-1}))$
  $return\,(x, Y) \vartriangleright the\; classified\; example$
```

Equation 3.5

Label information is distributed along the chain, therefore CC considers label correlations and overcomes the label independence problem of BR. The sequence of the chain affects accuracy.

- Label Powerset (LP)

Label Powerset is a straightforward and less frequently used technique for the problem transformation method [106, 109]. The concept of this technique is to treat each label from a multi-label dataset as a single label to convert a multi-label dataset into a single multi-class dataset by using each label combination as a single class. It assigns a new instance to the most possible class label that is made up of a set of labels to perform the classification task. After the classification task, LP can also do a ranking of labels by computing the sum of probability of all class labels that contain this new instance. LP indirectly considers label correlation, but it does not include all possible label combinations during the model development step, which would result in the issue of overfitting. If the number of label sets is large, LP would suffer from the increasing complexity due to diverse label sets, particularly for large values of instances and labels [106]. Another problem with the technique is that a large number of sets of labels are possible to be associated with a small number of instances, which would result in the issue of imbalance for learning [105].

3.1.2 Algorithm Adaptation Methods

Algorithm adaptation methods are different from problem transformation methods, since they extend certain learning algorithms to directly handle multi-label datasets, and the feature selection process is directly applied to the multi-label datasets. It is an algorithm-dependent method [106].

- Multi-label K-Nearest Neighbors (ML-KNN)

The k-nearest neighbor (KNN) classifier is one of the simplest classifiers available online for algorithm adaptation method, and it is often referred to as an 'example-based classifier, KNN algorithm compares new documents with the training set and identifies the K closest documents in the vector space of document features instead of establishing some abstract relationship between documents [110]. The prediction on the new document will be classified into the class that contains the highest proportion of its k nearest neighbors' documents [111]. Multi-label K-Nearest Neighbors (ML-KNN) is an extension of the KNN learning algorithm using a Bayesian approach. It employs the maximum a posteriori principle to classify the class label for the new instance, based on the prior and posterior probabilities of each label's frequency within the K nearest neighbors. ML-KNN has a relatively high computational costs since each new document must be compared with the training documents.

Table 3.3: Mathematical Notations for ML-KNN

Notations	Descriptions
X	Domain of dataset instances: $\{x_1, x_2, x_3, \dots x_n\}, 1 \leq i \leq n$
Y	Set of distinct label-sets: $\{Y_1, Y_2, Y_3, \dots Y_m\}, 1 \leq s \leq m$
T	Multi-label training dataset: $\{(x_1, Y_1), (x_2, Y_2), (x_3, Y_3), \dots . (x_n, Y_m)\}$, $x_i \in X, Y_s \subseteq L$
L	Set of all class labels: $\{y_1, y_2, y_3, \dots y_q\}, 1 \leq j \leq q$
$Rank()$	The ranking function
D	Multi-label dataset
Z	Set of labels that are predicted by the multi-label classifier

Table 3.3 shows some notations along with their descriptions for ML-KNN. Given an instance x_i and its associated label-set $Y_s \subseteq L$. Let $\overrightarrow{y_x}(l) \in L$ takes the value of 1 if $l \in Y_s$ and 0 otherwise supposing that $\overrightarrow{y_x}$ is the category vector for x_i . Assume N_x represent the set of KNNs of x_i identified in the training set. Based on the label sets of their neighbors, a membership counting vector is identified as:

$$\overrightarrow{C_x}(l) \; = \sum_{a \in N_x} \overrightarrow{y_a}(l)\,, l \in L \qquad \text{Equation 3.6}$$

Where $\overrightarrow{C_x}(l)$ represents the number of neighbors of x_i belonging to the lth class.

For the instance t for testing, ML-KNN starts from identifying its KNNs $N(t)$ in the training set. Assume H_1^l is the occurrence that t has the label l, whereas H_0^l is the occurrence that t does not have the label l. Assume that $E_j^l(j \in \{0,1, \dots, K\})$ is the occurrence that, there are j instances that have label l within the KNNs of t. Based on the membership counting vector $\overrightarrow{C_t}$, using the following MAP principle, the category vector $\overrightarrow{y_t}$ can be defined as [112]:

$$\overrightarrow{y_t}(l) = \underset{b \in \{0,1\}}{\arg max}\; \mathrm{P}\left(H_b^l \middle| E^l_{\overrightarrow{C_t}(l)}\right), l \in L \qquad \text{Equation 3.7}$$

Using the Bayesian rule, the above equation can be written as:

$$\begin{aligned} \overrightarrow{y_t}(l) &= \underset{b \in \{0,1\}}{\arg max} \frac{\mathrm{P}(H_b^l)\mathrm{P}\left(E^l_{\overrightarrow{C_t}(l)} \middle| H_b^l\right)}{P\left(E^l_{\overrightarrow{C_t}(l)}\right)} \mathrm{P}(H_b^l)\mathrm{P}\left(E^l_{\overrightarrow{C_t}(l)} \middle| H_b^l\right) \\ &= \underset{b \in \{0,1\}}{\arg max}\; \mathrm{P}(H_b^l)\mathrm{P}\left(E^l_{\overrightarrow{C_t}(l)} \middle| H_b^l\right) \end{aligned} \qquad \text{Equation 3.8}$$

The prior and posterior probabilities can be easily computed using frequency counting from the training set.

3.2 Classification in Text Mining and Natural Language Processing

Text mining is a broad field consisting of many aspects of traditional data mining and NLP tools [110]. Text classification task is associated with text mining, and text classifiers used for making the classification have been discussed in the last section. For the text classification task, natural language plays an important role in information retrieval [113]. Text classification extends beyond standard text categorization and information retrieval from documents to a variety of real-world problems, such as email classification, sentiment analysis, and various search engines [113, 114].

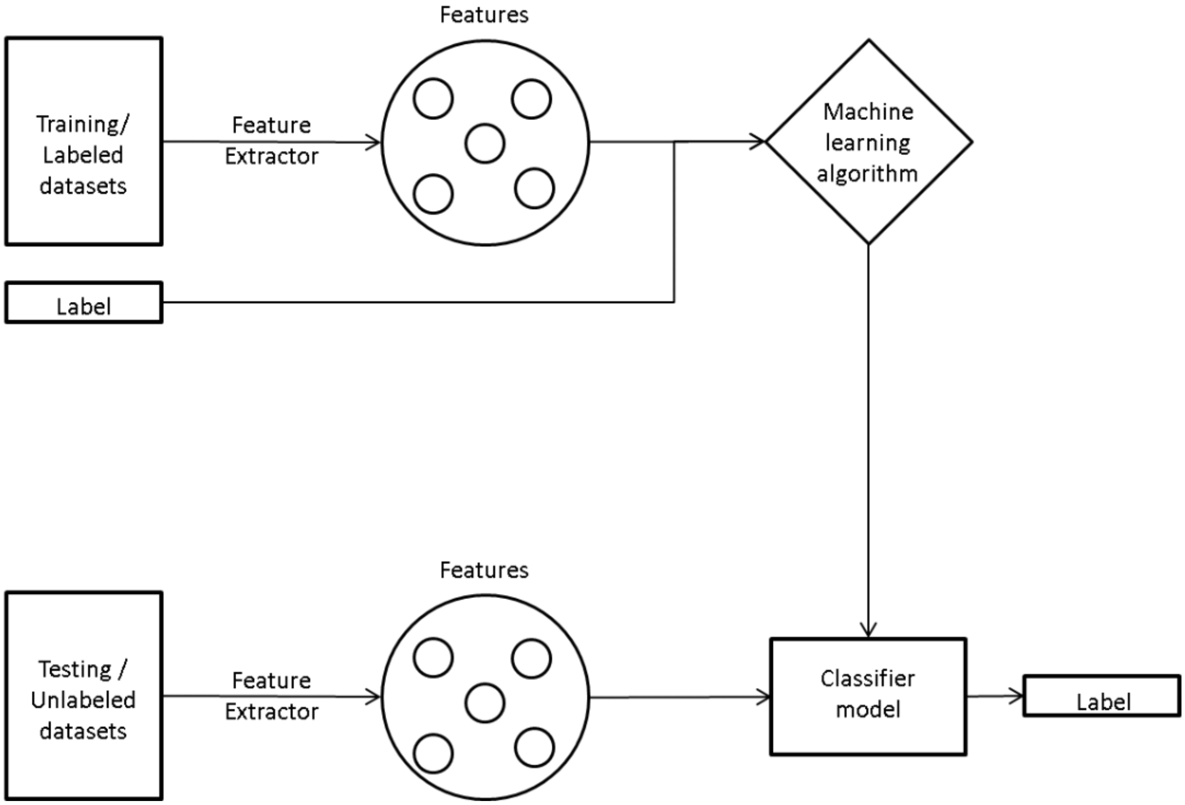

Features
Training/ Labeled datasets
Feature Extractor
Machine learning algorithm
Label
Features
Testing / Unlabeled datasets
Feature Extractor
Classifier model
Label

Figure 3.3: Schematic representation of classification process with supervised learning [80]

As seen in Figure 3.3, several machine learning algorithms are used to train feature vectors that are derived from training/labeled datasets, then a classifier model is used to predict the class label of testing/unlabeled, and the extracted features are used as input to the classifier model. Supervised learning is the classification process that involves training a classifier model on pre-labeled datasets as shown in the schematic representation [80]. Feature extraction is a vital stage in the text classification system since the type of features extracted affects the accuracy of the classifier model. It can improve the performance of the system with minimal additional effort. The following subsections will present the natural language processing (NLP) tools that are used for feature extraction and feature set construction.

3.2.1 Tokenization

The process of tokenization is dividing the continuous text into discrete words, numbers, symbols, or other relevant textual parts [110]. The objective is to represent the continuous text as vectors where each dimension represents one word shown in the whole text. The dimension of the feature vectors represents the number of unique words occurring in the whole text. While tokenization can be used to break text into a variety of formats, this research only uses word tokenization. In word tokenization, all sentences can be broken into their constituent words, numbers, and special characters. This is achieved by using whitespace characters via word tokenization. Tokenization is the first step involved in parsing, as the single entities generated by tokenization will be assigned a part of speech (POS) tagging. This tagging shows the grammar structure of the text, then a parsing tree is generated. Syntax is referred to as the grammatical perspective of a sentence. However, for the input of a parsing process, an efficient way needs to search across parsing trees. There are two types of parsers in the current application, lexicalized and unlexicalized parser. The difference is that the lexicalized parser will annotate both function words and content words with the tags, and it emphasizes the lexical dependencies for parse disambiguation. The unlexicalized parser will not create separate rules for different content words because content words are not part of grammatical structures. No use is made of lexical class to provide monolexical and bilexical probabilities for the unlexicalized parser [115]. Klein and Manning [115] demonstrate that an unlexicalized parser has greater performance while doing PCFG (Probabilistic Context-Free Grammar) parsing, which differentiates the well-known acknowledgment that lexicalized PCFG was the key tool for high-performance PCFG parsing. Several linguistically motivated annotations were described to close the gap between a vanilla PCFG and state-of-the-art lexicalized models.

They trained the unlexicalized PCFG model with horizontal and vertical markovization (external and internal annotations). And the tag splitting, checking the annotations already in the treebank, head annotation and distance were used to optimize the base model of the unlexicalized PCFG model. The advantage of unlexicalized PCFG is easy to estimate, easy to parse with, time-efficient, space-efficient, more compact, easier to replicate and to interpret than a more complex lexical model, and the parsing algorithm are easier to optimize compared to the other model, and can achieve high accuracy of 86.36% [115]. Therefore, an unlexicalized PCFG parser is used in this research.

3.2.2 Part of Speech (POS) Tagging

POS tagging is the process of assigning each word in a document with its tag from a grammatical perspective. Some popular POS tagging algorithms consist of Brill tagger, Hidden Markov model taggers, and neural network classifiers [116-118]. The main objective of POS tagging is to label the words according to their POS tagging. One of challenges of the task of POS tagging is how to distinguish between the tags for ambiguous words. There are various methods to tag, the PCFG parsing is used for our research, and it is available for download as open source online. Parsing can be seen as a search problem to run backwards to discover the grammatical structure of a sentence, CFG represents context-free grammar which specifies a set of tree structures that capture the constituency and ordering of language. Table 3.4 shows the descriptions of some POS tagging.

Table 3.4: POS tagging description

POS Tag	Description
NN	Noun
PRP	Pronoun
JJ	Adjective
VB	Verb
RB	Adverb
IN	Preposition
CC	Conjunction
UH	Interjection

The PCFG parser makes use of the Penn Treebank schema to represent phrasal categories and annotates the text with POS tags [80]. Table 3.5 shows different constituents that function as single units for Penn Treebank. In the Penn Treebank label set, there are 48 preterminal tags including 36 POS tags and 12 other symbols, and 14 nonterminal tags. The Stanford parser takes the text document as input, and output the phrase structure trees, typed dependencies, and plain POS tags [80]. The parse tree of the sentence “I shot the elephant in my pyjamas” is shown in Figure 3.4.

Table 3.5: Different constituents for Penn Treebank

Constituent	Description
S	Simple declarative clause
NP	Noun phrase
VP	Verb phrase
PP	Prepositional phrase

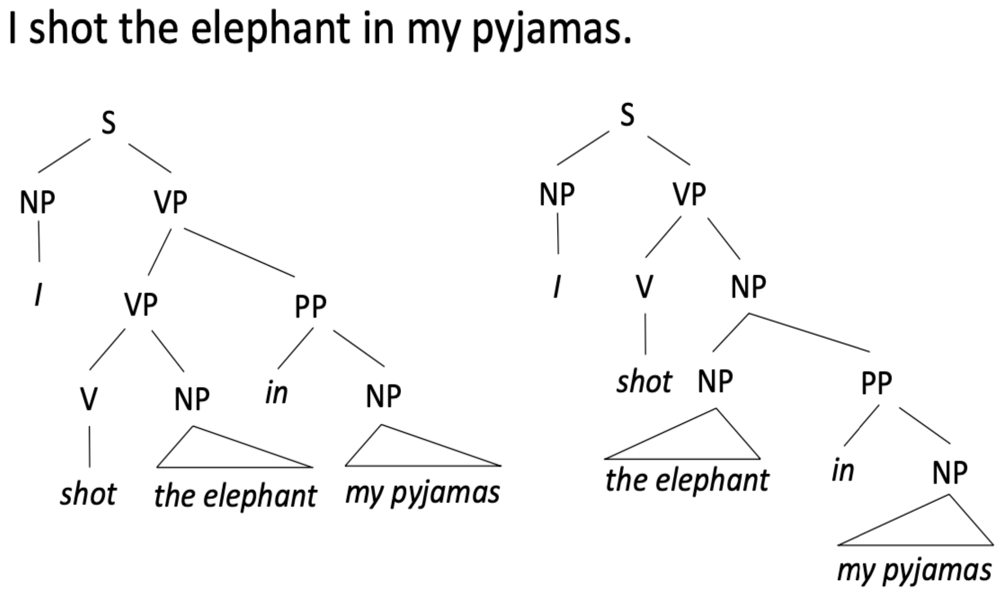


Figure 3.4: Parse tree of sample sentence [119]

3.2.3 Word Stemming and Lemmatizing

Word stemming and lemmatizing are part of text preprocessing and closely linked to each other. The stemming process reduces the word form by removing prefixes and suffixes to the root form, while the lemmatizing process removes the affixes and get the lemma form. Table 3.6 shows the example of word after stemming. If these words are processed with lemmatizing, it will output ‘discover’ since a lemmatizer would not transform the word into a form not found in the standard dictionary. Stemming algorithms have a possibility to muddle words, for example, the words ‘regulate’, ‘regulation’, ‘regulator’ can be mapped into the common stem ‘regul’. However, words ‘region’ and ‘regiment’ maybe mapped to the same stem term [110]. Therefore, stemming cannot predict right root forms for all words. Despite this limitation, stemming algorithms can significantly reduce the dimension of a feature set by transferring similar words into the same stem terms. Word stemming algorithms consist of the Porter, Snowball, and Lancaster stemmers. Porter stemming is a pioneering stemming algorithm that was introduced in 1979 [120, 121]. The Snowball stemmer is a modification of the Porter stemmer, it uses Porter's Snowball string processing language for stemming in various languages [122]. In 1990, the Lancaster stemmer was developed [123].

Table 3.6: Examples of word stemming [124]

Word	Word after stemming
Discovery	discoveri
discovered	discov
discoveries	discoveri
Discovering	discov

Lemmatizing is the process to reduce the word form to linguistically viable lemmas. When dealing with a large dataset or when performance is a major concern, stemming is preferable. If accuracy is the primary concern and the dataset is not enormous, lemmatizing is preferable. For this research, stemming was used for text preprocessing, lemmatizing is not used due to its computationally expensive costs and reliance on look-up tables. Stemming is more flexible than lemmatizing as researchers can add rules to the stemming process.

3.2.4 Natural Language Toolkit (NLTK)

Natural Language Toolkit (NLTK) is an open online Python library for natural language processing tasks. This toolkit is one of the most powerful libraries for NLP tasks, as it features the packages for machine understanding natural language and giving a proper response. It incorporates a variety of text processing libraries, such as tokenization, parsing, classification, stemming, tagging, character count, word count, and semantic reasoning [125]. It provides an easy-to-use interface to over 50 corpora and lexical resources like WordNet. Natural language processing with Python library NLTK provides a hands-on introduction to language processing programming by walking the users through the principles of developing Python programmes, dealing with various corpora, making classification on the text, analyzing linguistic structure, and more [126]. According to some papers, there are some fundamental requirements of using NLTK [127].

Table 3.7: Fundamental requirements of using NLTK [127]

Requirements	Explanation [127]
Simple to implement	The main objective of NLTK is to allow users to concentrate on constructing NLP components and systems. The more time it takes to spend learning to use the toolkit, the less useful it is.
Consistency	The toolkit must use data structures and interfaces that are compatible.
Documentation	The toolkit quickly adapts new components, whether they imitate or extend the toolkit's existing functionality and performance. The toolkit should be organized in such a way that adding new extensions fits the existing structure.
Monotony	The toolkit should cover the ramifications of developing NLP systems rather than dropping them, and every class determined by the toolkit must be assessable to users.
Modularity	The toolkit should cover the ramifications of developing NLP systems and do not drop them.

In this research, NLTK is used for text preprocessing such as tokenization, removing stop words, stemming, analyzing POS tagging and classification. Some detailed explanations and examples for each step will be demonstrated in next chapter.

4 Bio-inspired Design Classification System

According to the analysis in Chapters 2 and 3, a bio-inspired design classification system is presented in detail in this chapter. This system combines multiple domains for the purpose of extracting engineering knowledge from bio-inspired designs and classifying the function of bio-inspired designs. The proposed system can categorize the bio-inspired design according to their functionalities, which also enables future researchers to quickly and efficiently locate the articles or publications based on the design function. Figure 4.1 shows the overall flowchart of the proposed system. It consists of five general steps: Dataset Collection, Data Preprocessing, Feature Extraction, Classifier Development, and Classifier Evaluation and Comparison. The following sections will describe each step of the proposed system in detail.

4.1 Dataset Collection

Most of the existing machine learning applications rely on properly labelled data that has been collected online for a specific purpose. Such prepared data is not available in the field of bio-inspired design. Due to the lack of labelled data on bio-inspired design, the manual labelling abstract/ main information retrieved from the AskNature website was caried out using the general functionalities of bio-inspired designs. AskNature website is a useful database that organizes biological knowledge according to the Biomimicry Taxonomy [40], which classifies the biological system based on its function. Since data interpretation is subjective and different stakeholders would have different interpretations, the data labelling process involved two individuals preparing the data and cross-checking the labels of the bio-inspired designs for verification. Table 4.1 illustrates the potential functionality of a bio-inspired design. The input data will be assigned as a label ‘1’ if the bio-inspired design can accomplish the desired function or property, and it will be assigned as ‘0’ if the design is not associated with the function. For the collected dataset, abstract or main information extracted from the AskNature website is seen as the input. The reason of only using the abstract/ main information is that users can catch the research direction and summarize the research paper based on the abstract/ main information. The output is the label of functions of bio-inspired designs. This research uses 18 categories of bio-inspired design. Then, the modified dataset is constructed and utilized as the input for preprocessing the data.

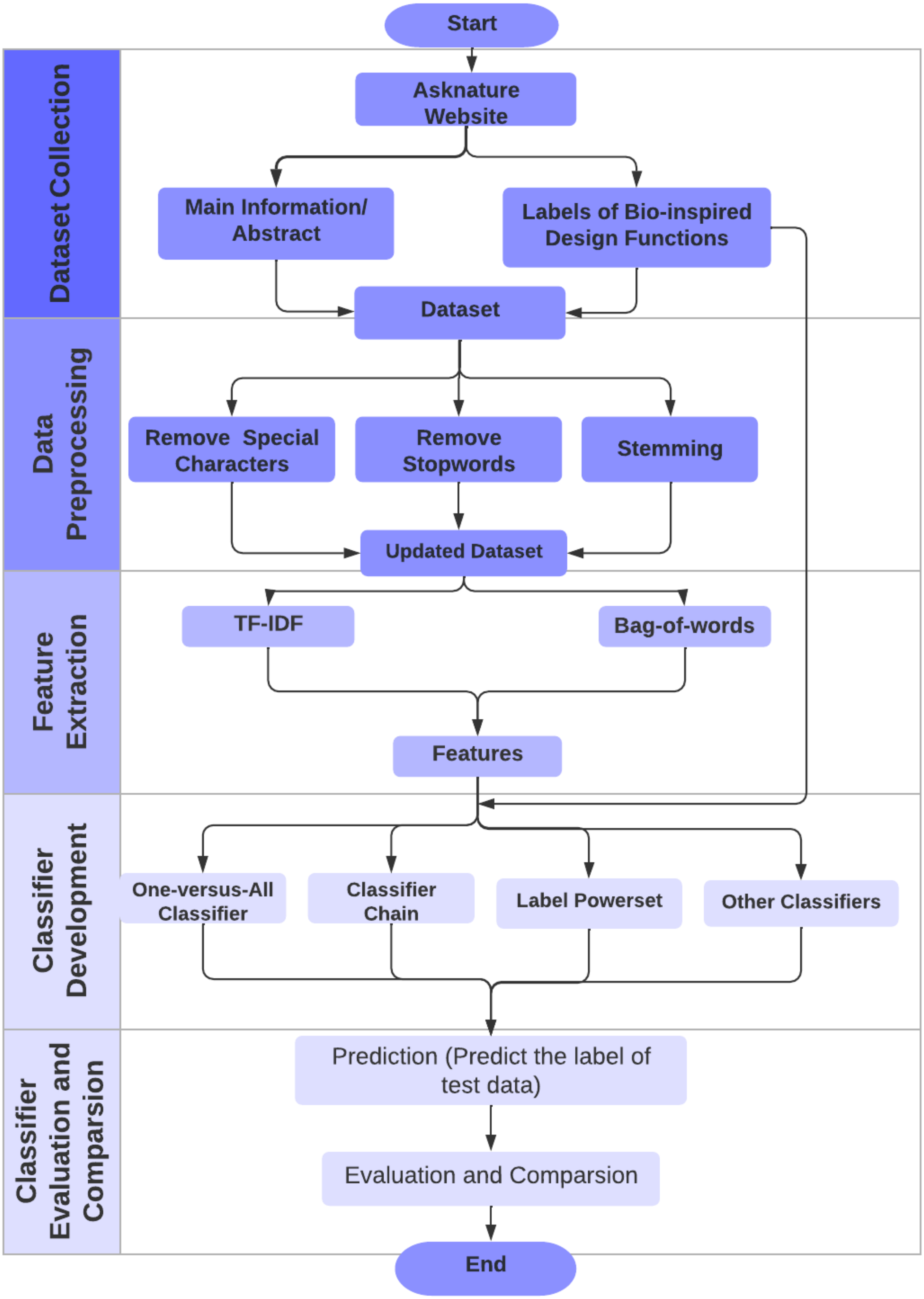


Figure 4.1: The overall flowchart of Bio-inspired Design Classification System

Table 4.1: Labels of bio-inspired designs

Stiffness	Drag Reduction
Self-cleaning	Noise Reduction
Flexibility	Waste Reduction
Lightweight Structure	Adhesion
Increase Strength	Anti-wear
Cost Reduction	Anti-adhesion
Increase Efficiency	Superhydrophobic
Energy Absorption	Impact resistance
Anti-bacterial	Energy Reduction

4.2 Data Preprocessing

The most important step in obtaining a consistent and solid analytical dataset is data preprocessing. It is important to eliminate all digits from the document, which can be accomplished with the aid of a regular expression. After the dataset collection step, it is possible that there could be no information provided for some labels regarding their design functions. Therefore, checking for missing data values is critical for the dataset construction process. The process of gathering data from numerous AskNature pages requires additional data preprocessing to normalize and standardize the results as some unsolved ambiguity could result in future problems. In the bio-inspired design classification system, data preprocessing consists of four main steps: tokenization, removal of special characters, removal of stopwords, and stemming. The data quality was initially determined by checking whether there were any mismatched data, missing, or mixed data values. Then special characters and stopwords were removed, and stemming was employed to remove the inflectional and derivational forms from the word. In addition, all words are turned to lowercase to avoid any possibility of misinterpretation. As a result, data preprocessing is critical to the proposed system, as clean data enables the bio-inspired design classification system to have a better performance. The detail for each phase of data preprocessing was demonstrated in Chapter 3.2.

4.3 Feature Extraction

Following the data preprocessing step, raw textual data was transformed into a format that is comprehensible. Then data transformation began by converting the clean textual data to numerical features for subsequent analysis. This is a process known as feature extraction. The process of feature engineering and extraction is critical for constructing text classifiers [128]. The main objective of feature extraction is to transform the data into features that are more useful for classification while minimizing the computational cost. Several candidate feature sets were developed and examined in this research. The feature extraction and vectorization were performed by using Bag-of-words (BOW) and TF-IDF to train word vectors. Effects of different feature representations along with different classifier models will be compared and assessed in Chapter 5.

4.3.1 Bag-of-Words (BOW)

In Bag-of-Words model, each word in the document represents a feature vector. The Bag-of-Words model produces a set of word features. It is constructed by sentence tokenization, removing special characters, removing stopwords, and stemming words. BOW algorithm is a counting algorithm that encodes a document in a dataset using a vector representation with $|x|$ dimensions, where $|x|$ is the size of the vocabulary selected [129]. These word feature sets are referred to as relatively dense feature sets [110]. Vectors representation developed by BOW does not take the ordering of words in a document into account, therefore, the ordering of word is lost. For example, ‘John is quicker than Mary” and “Mary is quicker than John” would have the same vector representations.” While this model is simple and straightforward to implement, it is deficient in information about the semantic meaning of the sentence [110]. While reducing the feature set speeds up the classifier, it sacrifices some information about the sentences by removing stopwords and conducting word stemming [110]. In this research, BOW is used to count how many times a word appears in the document.

4.3.2 TF-IDF

The TF-IDF (Term Frequency-Inverse Document Frequency) vectorization approach was used for feature representation. Feature extraction using the TF-IDF approach was used to tokenize the trigrams and word grams from the collected dataset in the training phase. TF-IDF feature vectors are used to represent the words in the document. This method involves computing the TF-IDF score for each word in the corpus and storing the results in a vector [130]. It is frequently used to extract keywords from documents, determine search rankings, and compare the degrees of similarity between documents. Additionally, it is a common technique used in text mining and information extraction to determine the significance of a single word in a corpus [130].

The TF in TF-IDF indicates the frequency of certain words occurring in the document. The word with a high TF value indicates that it is more significant in the document. DF indicates the number of times a given term appears in the document collections, not in a single document. Words with a high DF value indicates that it frequently occurs in all documents but is not important. IDF is the inverse of DF, which is used to determine the importance of words throughout all documents. If the word is rare and uncommon, the value of IDF will be high. The calculation of TF is shown below

$$TF(t,d) = \frac{n_{t,d}}{\sum_k n_{k,d}} \qquad \text{Equation 4.1}$$

Where t represents the specific word, d represents each document, D represents the collection of documents. $n_{t,d}$ represents the number of occurrences of word t in document d. $\sum_k n_{k,d}$ represents a total number of occurrences of words in document D. And k represents the number of keywords.

$$IDF(t,D) = log\frac{|D|}{1 + |\{d \in D: t \in d\}|} \qquad \text{Equation 4.2}$$

Where D represents the collection of documents, $|\{d \in D: t \in d\}|$ represents the number of documents that the keyword occurs. “Plus 1” that appears in the denominator is used to avoid the zero division. The higher the $TF - IDF$ value, the more significant the word is in the corpus. Using the $TF - IDF$ value calculated by Equation 4.3, the term frequency of each word in each selected AskNature page could be computed. And the important keywords in each AskNature page could be identified by comparing the $TF - IDF$ value.

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D) \qquad \text{Equation 4.3}$$

After the features are represented by Bag-of-Words and TF-IDF method, the fine-tuning process is applied before training the classifier. Several base classifiers and ensemble of classifiers are presented in next section.

4.4 Classifier Development

After the feature extraction step, TF-IDF would convert words in the document into vectors, which is used as training data for the classification algorithms. The preprocessed dataset was divided into a training set and a test set in a ratio of 67:23. In the total number of 90 AskNature pages, 67 pages is chosen as training dataset, and the remaining 23 pages is chosen as test dataset. Training dataset is used for training the machine learning algorithm to construct a classification model. And the test dataset is used to test the hypothesis of the model and evaluate the performance of the machine learning model. In addition, the training data is vectorized using the scikit-learn library modules.

To demonstrate the effectiveness of the proposed classification system in this research, the following state-of-the-art baseline classifiers are chosen by ensemble-based classifiers to perform the classification task, including Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machines (SVM), and Decision Tree (DT). Ensemble-based classifiers, also referred to as multiple classifier systems [131], are constructed by combining several baseline classifiers to produce a classifier that outperforms each independent classifier [132]. In this section, various classification algorithms that based on combinations of baseline classifiers are going to be discussed. These classifiers will be discussed as baseline classifiers and ensemble-based classifiers. Based on evaluation metrics, their performance will be evaluated, compared, and further analyzed in next chapter.

4.4.1 Baseline Classifier Choice

In this method, some baseline classifiers including Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machines (SVM), and Decision Tree (DT) are explained in detail in the following sections.

- Naïve Bayes (NB)

Naïve Bayes classifier is frequently used as a baseline in machine learning models. It assumes that all features are independent from each other. It is a traditional binary classifier. The Naïve Bayes classifier can adjust the probability threshold to optimize the number of correctly labeled AskNature pages. Naïve Bayes classifier is based on Bayesian network (BN), which is a probabilistic machine learning technique that employs the Bayes theorem to model the conditional dependence of events using a directed edges graph [133]. The structure of Naïve Bayes is shown in Figure 4.2. NB classifier assumes that all of the attributes are conditionally independent from each other. NB classifier has been used in applications by Dinkakar et al. [134], Chen et al. [135], and Kwok and Wang [136]. It can not only classify the text document into its appropriate category, but it can also detect the topic related to the text document. As a result, it serves as a baseline classifier.

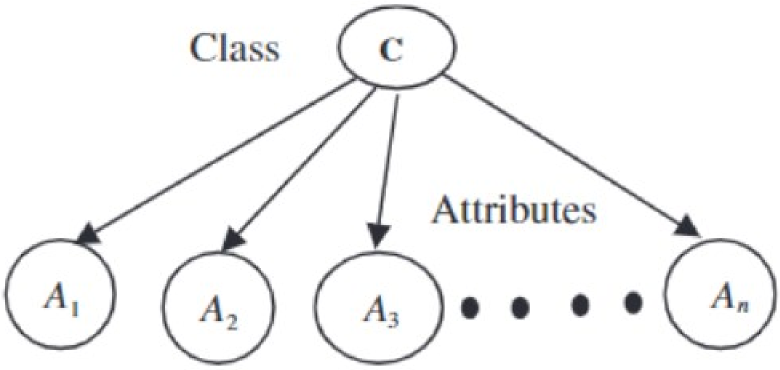


Figure 4.2: Structure of Naive Bayes classifier [137]

- Logistic Regression (LR)

Logistic regression (LR) is well-know supervised learning technique for its simplicity and common use for text classification [128]. LR is often used as a binary classifier, the output is typically binary, but it can also contain three or more target variables.

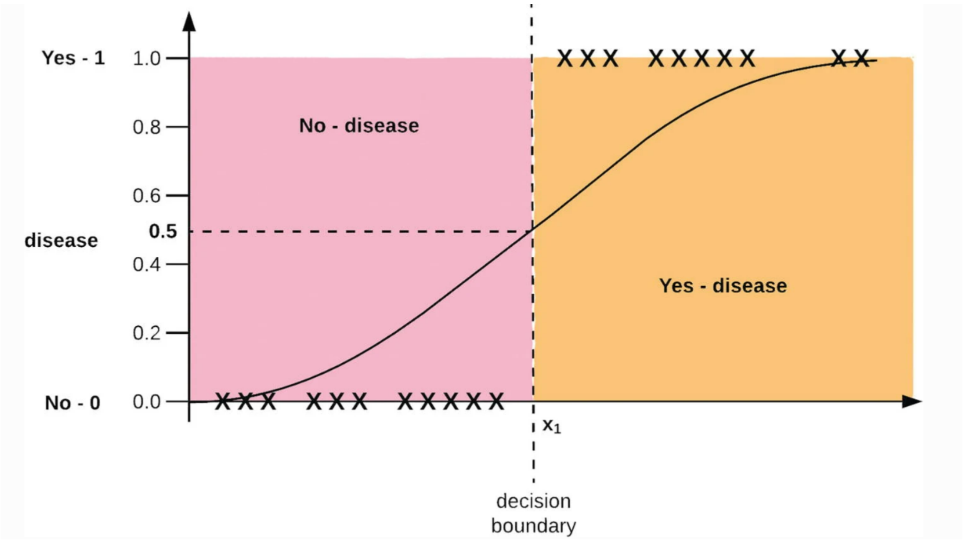


Figure 4.3: Logistic Regression example [138]

Figure 4.3 shows an example of Logistic Regression function. LR function is also known as the sigmoid function, which is used to constrain the output of the cost function between 0 and 1. The LR function looks like an S curve, starts from 0 and approaches to 1. As the time increases to positive infinity, the value of the output becomes closer to 1. LR has been used for sentiment analysis such as online hate detection by Xiang et al. [139], Burnap and Williams [140], Waseem and Hovy [141], Davidson et. al. [142], Wulczyn et al. [143], and Salminen et al. [144]. Therefore, LR is chosen as one of the baseline classifiers to solve the classification task in this research.

- Support Vector Machine (SVM)

Another algorithm frequently used for text classification is support vector machines (SVM). When sentences are expressed in terms of a set of m features, a collection of AskNature pages is a set of m-dimensional vectors in the feature space [110]. SVM will determine the best hyperplane in the feature space that correspond to the predicted label of the selected AskNature page as shown in Figure 4.4. The plane is defined by the collection of AskNature pages that adjacent to it. Support vectors are the terms to describe these pages. A kernel trick is used to aid the space's handling of nonlinearities. SVM is a robust and efficient classifier since it classifies a new AskNature page by determining which side of the hyperplane it lies on [110]. However, SVM is sensitive to missing data values and can be difficult to train on large datasets [107].

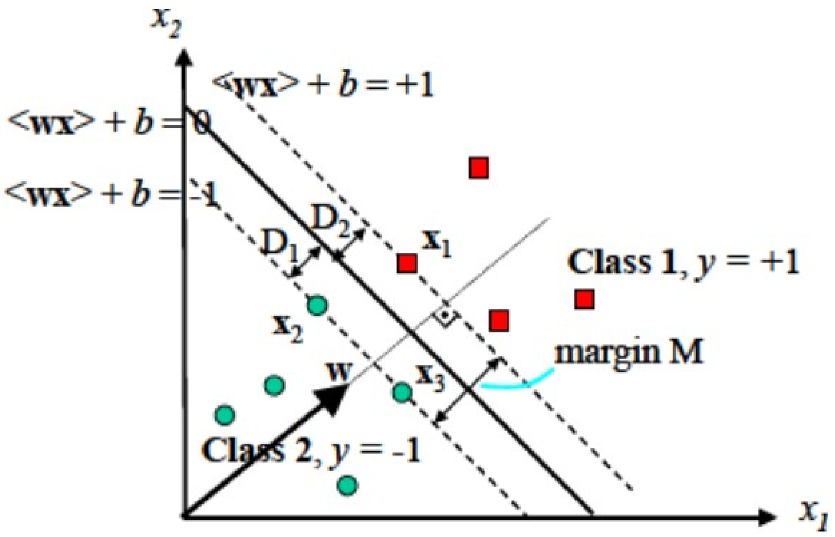


Figure 4.4: Linearly separable samples of SVM [145]

- Decision Tree (DT)

Decision tree is a supervised learning technique that is widely used for classification and prediction. DT is one of the most used decision-based classification algorithms and have great ability to handle both numerical and categorical value [146-148]. It is a tree-like machine learning model that can predict the target or output value given a set of input variables, each internal node corresponds to a test and each branch corresponds to an outcome, the leaf node represents the net result [80]. Rules are defined by the path from the root node to the leaf node. Decision trees have a great performance while handling missing values, however, it struggles to model on small datasets. Figure 4.5 shows an example of tree graph generated by a Decision Tree algorithm.

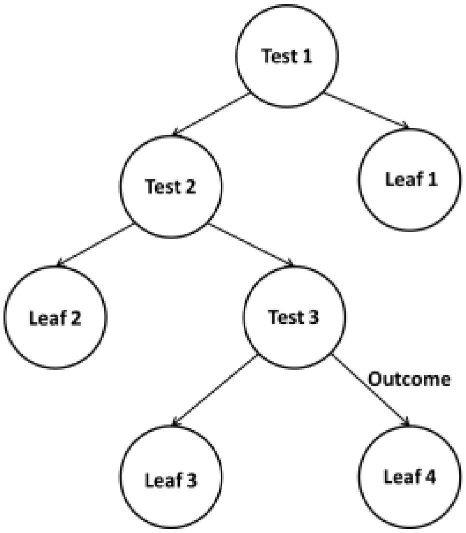


Figure 4.5: Decision tree graph example [80]

4.4.2 Ensemble-based Classifier

Ensemble-based classifier often shows the better performance than baseline classifier, since it is built based on different combinations of several baseline classifiers [132]. Figure 4.6 shows the framework of ensemble-based classifier. Ensemble-based classifiers can be divided into two categories: problem transformation classifier and algorithm adaptation classifiers. Table 4.2 and Table 4.3 show different classifiers used in the proposed system. Problem transformation classifiers are built based on different combinations of multi-label classifiers and baseline classifiers. For Binary Relevance classifier, its performance is compared with combinations of Naïve Bayes, Decision Tree, and Logistic Regression baseline classifiers. For Classifier Chain, its performance is compared with combinations of Naïve Bayes, and Logistic Regression baseline classifiers. For Label Powerset classifier, its performance is compared with combinations of Naïve Bayes, Decision Tree, and Logistic Regression baseline classifiers. The performance of different ensemble-based classifiers is compared based on the classifier evaluation metrics shown in Section 4.5.

Table 4.2: Algorithm adaptation classifiers

	Classifier	**Description**
Algorithm Adaptation Classifiers	ML-KNN	Multi-label K-Nearest Neighbors
	AdaBoost	Boosting algorithm
	XGBoost	Boosting algorithm

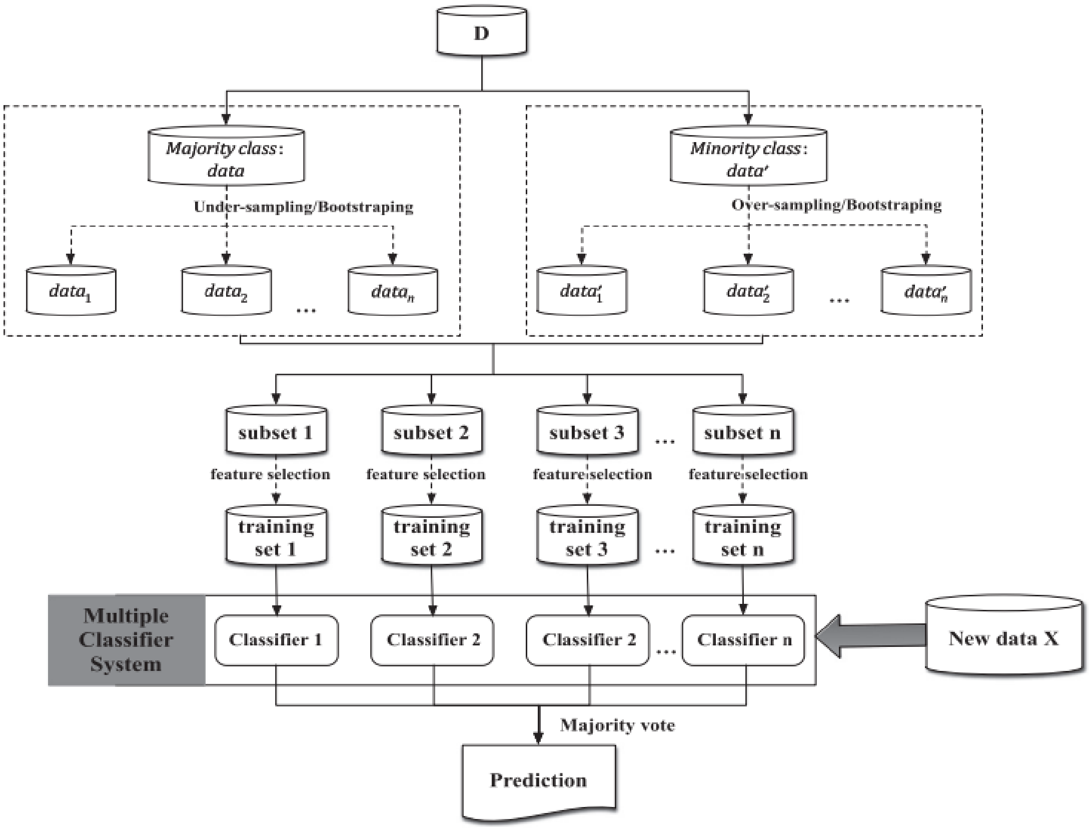


Figure 4.6: Ensemble-based classifier framework [107]

Table 4.3: Problem transformation classifiers

Problem Transformation Classifiers	**Classifier**	**Description**
	BR(NB)	Binary Relevance with Naïve Bayes baseline
	BR(DT)	Binary Relevance with Decision Tree baseline
	DT	Decision Tree
	LP (LR)	Label Powerset with Logistic Regression baseline
	OVA (LR)	One-versus-All classifier with Logistic Regression baseline
	LP (NB)	Label Powerset with Naïve Bayes baseline
	LP (DT)	Label Powerset with Decision Tree baseline
	CC (NB)	Classifier Chain with Naïve Bayes baseline
	CC (LR)	Classifier Chain with Logistic Regression baseline

4.5 Classifier Evaluation Metrics

In the traditional single-label classification task, accuracy, precision, recall, F-measure, and ROC (Receiver operating characteristic) area are the most frequently used evaluation metrics [149]. However, with multi-label classification tasks in this research, predictions for a new instance are a set of labels. Thus, the prediction can be fully correct, partially correct, or completely incorrect [105]. The common evaluation metrics used for single-label classification problems cannot fully convey the levels of correctness. Therefore, it is more difficult and challenging to evaluate a multi-label classifier than a single-label classifier. Depending on the target problem of the multi-label classification task, evaluation metrics for multi-label dataset can be divided into three categories: partition evaluation, ranking evaluation, and a label hierarchy structure evaluation [150]. Partition evaluation is to assess the quality of the classification into classes, ranking evaluation is to evaluate if the classes are ranked according to their relevance, and a label hierarchy structure evaluation is to assess how successfully the learning system can consider the hierarchical structure of existing label [150]. The hierarchical structure of the label is not considered in this research.

Evaluation of a learning system can be defined as a measurement of testing the prediction performance of the learning system on the new instances [105]. Partition evaluation can be divided into two categories: example-based and label-based evaluation.

Referring Table 3.2, let T be a multi-label dataset containing n multi-label instances $(x_i, Y_i), 1 \leq i \leq n, x_i \in X, Y_i \in Y = \{0,1\}^k$, with the label set $L, |L| = k$. Assume h is the multi-label classifier and Z represents the set of labels that are predicted by the multi-label classifier h for the instance x_i, and $Z_i = h_{x_i} = \{0,1\}^k$. The following sections explain these classifiers in detail. All the variables can be found in Table 3.2.

4.5.1 Example-based Evaluation

For the example-based evaluation method, the average difference between the predicted label and the actual label for each text data is assessed [105]. As mentioned previously, evaluating a multi-label classifier is challenging since it incorporates the concept of partially correct. Godbole et al. [151] provided the directions for accuracy, precision, recall, and F1 measure to consider the concept of partially correct in the multi-label problem, and they are demonstrated as Equation 4.4, Equation 4.5, Equation 4.6, and
Equation 4.7.

Accuracy (A) is defined as the ratio of the predicted correct labels to the total number of predicted and actual labels for each instance. Overall accuracy is calculated as the average of all instances.

$$Accuracy, A = \frac{1}{|n|}\sum_{i=1}^{|n|}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$ Equation 4.4

Precision (P) is defined as the ratio of correctly predicted labels to the total number of actual labels over all instances.

$$Precision, P = \frac{1}{|n|}\sum_{i=1}^{|n|}\frac{|Y_i \cap Z_i|}{|Z_i|}$$ Equation 4.5

Recall (R) is the ratio of correctly predicted labels to the total number of predicted labels over all instances.

$$Recall, R = \frac{1}{|n|}\sum_{i=1}^{|n|}\frac{|Y_i \cap Z_i|}{|Y_i|}$$ Equation 4.6

F1-Measure (F) can be defined as the harmonic mean of precision and recall, it is calculated based on precision and recall. The higher the value of accuracy, precision, recall and F1-Measure indicates the better performance of the multi-label classifier.

$$F_1 = \frac{1}{|n|}\sum_{i=1}^{|n|}\frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|}$$ Equation 4.7

Hamming Loss (HL) is used for evaluating the performance of classifier prediction on the test data. It considers the prediction error (when an instance is assigned as a incorrect label) and the missing error (when a related label is not predicted), normalized over the total number of classes and total number of examples [105]. I represents the indicator function. The small value of hamming loss means the classifier would have better performance. The calculation of HL is shown in

Equation 4.8.

$$HammingLoss, HL = \frac{1}{|kn|}\sum_{i=1}^{|n|}\sum_{l=1}^{|k|}[I(l \in Z_i \wedge l \notin Y_i) + I(l \notin Z_i \wedge l \in Y_i)]$$ Equation 4.8

4.5.2 Label-based Evaluation

Label-based measures are used to evaluate the performance on each label independently and then average the results across all labels. It is most frequently used for Binary Relevance classifier. The traditional evaluation metrics including accuracy, precision, recall, F1-measure, and ROC can be used on each independent label. Macro averaged measures and micro averaged measures can be calculated as shown in Equation 4.9 and Equation 4.10 [105].

Macro Averaged Measures:

$$\lambda - Precision, P_{macro}^{\lambda} = \frac{\sum_{i=1}^{n} Y_i^{\lambda} Z_i^{\lambda}}{\sum_{i=1}^{n} Z_i^{\lambda}}$$

$$\lambda - Recall, R_{macro}^{\lambda} = \frac{\sum_{i=1}^{n} Y_i^{\lambda} Z_i^{\lambda}}{\sum_{i=1}^{n} Y_i^{\lambda}}$$ Equation 4.9

$$\lambda - F_{1-macro} = \frac{2\sum_{i=1}^{n} Y_i^{\lambda} Z_i^{\lambda}}{\sum_{i=1}^{n} Y_i^{\lambda} + \sum_{i=1}^{n} Z_i^{\lambda}}$$

Micro Averaged Measures:

$$Precision, P_{micro} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Z_i^j}$$

$$Recall, R_{micro} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j}$$ Equation 4.10

$$F_{1-micro} = \frac{2\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j + \sum_{j=1}^{k}\sum_{i=1}^{n} Z_i^j}$$

Where

$$Y_i^\lambda = \begin{cases} 1 & if x_i \ actually \ belongs \ to \ class \ label \ \lambda \\ & 0 \ \ otherwise \end{cases}$$

And,

$$Z_i^\lambda = \begin{cases} 1 & if x_i \ actually \ belongs \ to \ class \ label \ \lambda \\ & 0 \ \ otherwise \end{cases}$$

Macro averaged F_1 is more affected by the performance of the class labels that has less examples, while micro averaged F_1 is more affected by the performance of the class labels that has more examples [105].

4.5.3 Ranking Evaluation

If the classifier is capable of learning the ranking of predicted labels, the following metrics are often used to evaluate the performance of the classifier [150]. One Error (O) is used to count the number of times of the top-ranked labels does not exist in the instance's list of actual labels [105]. The formula of calculating One-error is shown in Equation 4.11.

$$One - error, O = \frac{1}{|n|}\sum_{i=1}^{|n|} I\left(arg \min_{\lambda \in L} r_i(\lambda) \notin Y_i^l\right)$$ Equation 4.11

Where *I* represents the indicator function, $r_i(\lambda)$ is the predicted rank of class label λ for a new instance x_i. One error is the classification error for the single-label classification task [105]. The smaller value of one error generally indicates the classifier has better performance.

Ranking Loss (RL) is calculated as the average proportion of label pairs that are ranked wrong for an instance [105], which is shown in Equation 4.12.

$$RankingLoss, RL = \frac{1}{|n|}\sum_{i=1}^{|n|}\frac{1}{\left|Y_i^l\right|\left|\overline{Y_\iota^l}\right|}\left|(\lambda_a, \lambda_b): r_i(\lambda_a) > r_i(\lambda_b), (\lambda_a, \lambda_b) \in Y_i^l \times \overline{Y_\iota^l}\right|$$

Equation 4.12

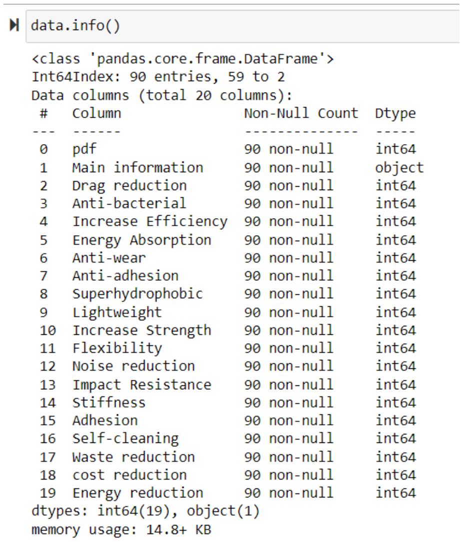
Where $\overline{Y_\iota^l} = L\backslash Y_i$, the smaller value of RL indicates that the classifier has better performance. The classification performance of each classifier is demonstrated in detail in Chapter 5 based on the above evaluation metrics: micro averaged measures, macro averaged measures, accuracy, and hamming loss.

5 Implementation and Experiments

In this chapter, experiments are conducted on the collected dataset in order to evaluate the performance of the proposed system. The dataset, experiment setup, and various classifier algorithms used in the proposed system are described in detail, along with their associated results. A discussion and analysis of proposed system's performance from different perspectives will be demonstrated as well.

5.1 Dataset and Experiment Setup

In the experiment, AskNature is the primary used database. Ninety AskNature pages were selected based on their relevance to the bio-inspired design knowledge and their ability to achieve multi-functionality. The main information is extracted directly from the AskNature page and inserted as the input of the dataset. Two individuals manually labelled the chosen pages based on the bio-inspired design functionalities. The target of the input (the main information) is defined as the labels on the selected page. There are eighteen categories of bio-inspired design functionalities, which includes drag reduction, anti-bacterial, increase efficiency, energy absorption, anti-wear, anti-adhesion, superhydrophobic, lightweight, increase strength, flexibility, noise reduction, impact resistance, stiffness, adhesion, self-cleaning, waste reduction, cost reduction, and energy reduction.



```
data.info()
```

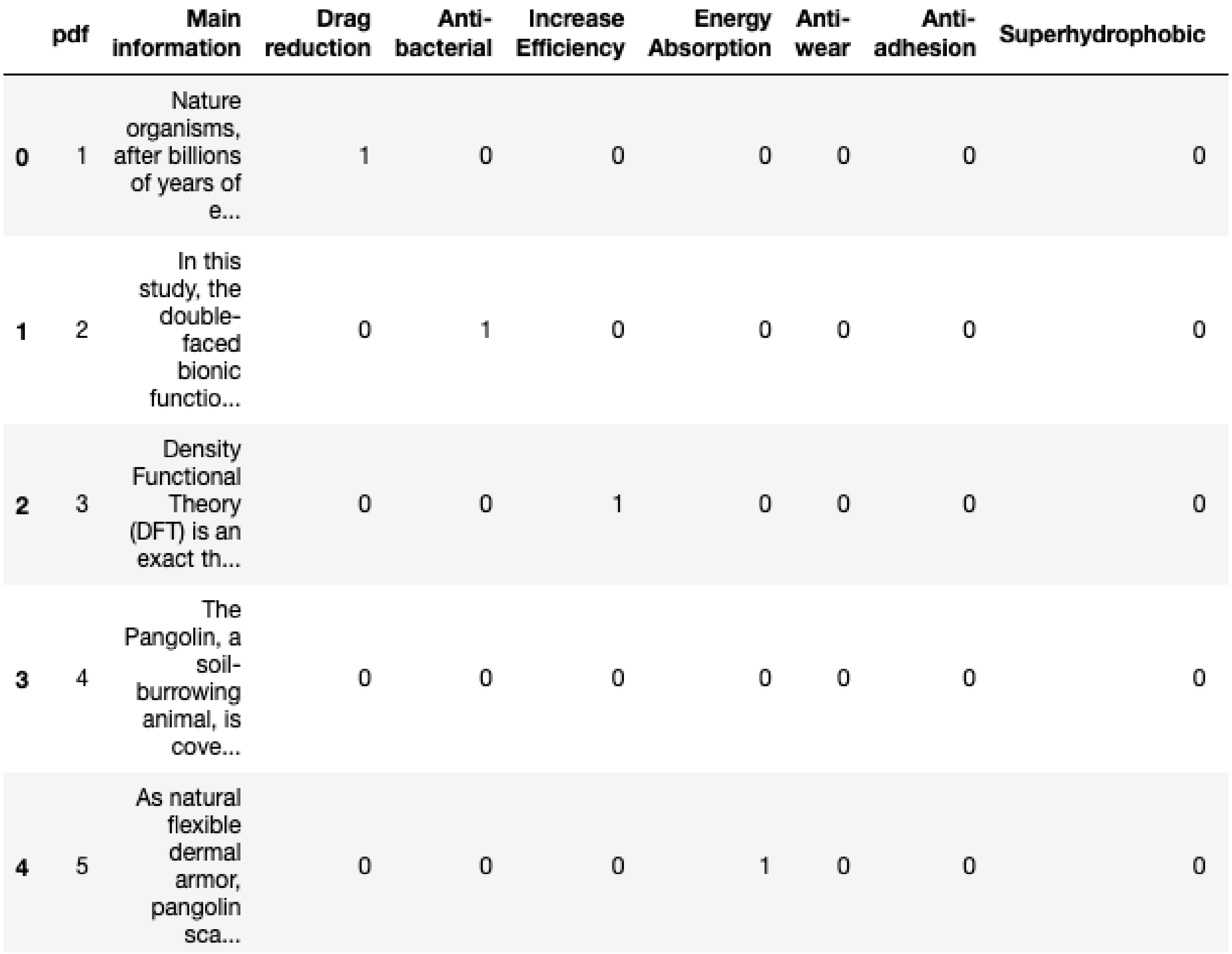
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 90 entries, 59 to 2
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   pdf                  90 non-null     int64 
 1   Main information     90 non-null     object
 2   Drag reduction       90 non-null     int64 
 3   Anti-bacterial       90 non-null     int64 
 4   Increase Efficiency  90 non-null     int64 
 5   Energy Absorption    90 non-null     int64 
 6   Anti-wear            90 non-null     int64 
 7   Anti-adhesion        90 non-null     int64 
 8   Superhydrophobic     90 non-null     int64 
 9   Lightweight          90 non-null     int64 
 10  Increase Strength    90 non-null     int64 
 11  Flexibility          90 non-null     int64 
 12  Noise reduction      90 non-null     int64 
 13  Impact Resistance    90 non-null     int64 
 14  Stiffness            90 non-null     int64 
 15  Adhesion             90 non-null     int64 
 16  Self-cleaning        90 non-null     int64 
 17  Waste reduction      90 non-null     int64 
 18  cost reduction       90 non-null     int64 
 19  Energy reduction     90 non-null     int64 
dtypes: int64(19), object(1)
memory usage: 14.8+ KB
```

Figure 5.1: Dataset information

This dataset is used to train the classification model. The dataset information is illustrated in Figure 5.1 and summarized in Table 5.1. Five samples from the dataset are shown in Figure 5.2.

Table 5.1: Dataset statistics

Total	Train	Test	Labels	Data Type
90	67	23	18	integers



	pdf	Main information	Drag reduction	Anti-bacterial	Increase Efficiency	Energy Absorption	Anti-wear	Anti-adhesion	Superhydrophobic
0	1	Nature organisms, after billions of years of e...	1	0	0	0	0	0	0
1	2	In this study, the double-faced bionic functio...	0	1	0	0	0	0	0
2	3	Density Functional Theory (DFT) is an exact th...	0	0	1	0	0	0	0
3	4	The Pangolin, a soil-burrowing animal, is cove...	0	0	0	0	0	0	0
4	5	As natural flexible dermal armor, pangolin sca...	0	0	0	1	0	0	0

Figure 5.2: Part of the dataset

All experiments are implemented using Jupyter Notebook and executed on an 11th Gen Intel(R) Core-i3 processor with 8 GB memory on Microsoft Windows 11. To obtain stable results, each experiment is repeated through 20 times. In each run, the 5-fold hold out cross-validation is used, 67 instances were randomly chosen as the training set, and the remaining 27 instances were chosen as the test set. The classification results are affected by different word vectors. In this research, all baseline classifiers are compared using TF-IDF and Bag-of-Words (BOW) to train word vectors. At the start of the experiment, the default threshold and parameters are set for the baseline classifier. Then, hyperparameter tuning is used to determine the optimal parameters that result in the highest accuracy of the model. In addition, some hyperparameters adjustments are made to aid the classification model in avoiding the problem of overfitting.

5.2 Overall Performance Evaluation

To compare the performance of classifiers used in the proposed system, they are trained on the randomly chosen 63 AskNature pages. These classifiers are One-versus-All (Logistic Regression baseline) classifier, Label Powerset (Logistic Regression baseline), Label Powerset (Naïve Bayes baseline), Classifier Chain (Naïve Bayes baseline), XGBoost Classifier, Binary Relevance (Naïve Bayes baseline), Classifier Chain (Logistic Regression baseline), AdaBoost Classifier, Label Powerset (Decision Tree baseline), Binary Relevance (Decision Tree baseline), Decision Tree, and ML-KNN. The remaining 27 AskNature pages were used to test the constructed model. Label-based evaluation metrics are used to evaluate the performance of each label independently and then average the results across all labels. Macro averaged measures and micro averaged measures are demonstrated as well. In addition, example-based evaluation metrics hamming loss is used to evaluate the performance of classifier prediction on the test data. The small value of hamming loss means the classifier would have better performance.

The experiments are conducted 20 times on the dataset to obtain stable results. A multi-label classification model is developed to classify the bio-inspired design corresponding to their design multi-functionalities. The performance of the proposed system is more complicated than a traditional binary classification model. The accuracy of each label of One-versus-All classifier (Logistic Regression baseline) based on TF-IDF features is shown in Table 5.2. One-versus-All classifier outperforms all other classifiers. Table 5.3 shows the micro averaged and macro averaged measures of One-versus-All classifier. The micro averaged measure is larger than the macro averaged measure since the micro F1-measure gives equal weight to all classes, and it is affected by class deviations. Micro averaged measure is controlled by the more frequent class, while macro averaged measure reflects the less frequent class better. Therefore, the micro averaged measure of One-versus-All classifier is larger than the macro averaged measure of it. In terms of micro and macro averaged measures, One-versus-All yields the highest micro averaged precision and the highest micro F1-measure, indicating that it has the overall best performance compared to other classifiers.

Table 5.2: Accuracy of each label of One-versus-All (LR baseline) Classifier based on TF-IDF features

Label	Accuracy
Drag reduction	0.9815
Anti-bacterial	0.9877
Increase Efficiency	0.8580
Energy Absorption	0.8518
Anti-wear	0.9877
Anti-adhesion	0.9831
Superhydrophobic	0.9074
Lightweight	0.8642
Flexibility	0.9568
Noise reduction	0.8518
Increase strength	0.8950
Impact resistance	0.9135
Stiffness	0.9815
Adhesion	0.9877
Self-cleaning	0.9753
Waste reduction	0.9012
Cost reduction	0.8580
Energy reduction	0.9444

Table 5.3: Micro, macro quality number of One-versus-All (LR baseline) Classifier based on TF-IDF features

One-versus-All (LR baseline) Classifier	Precision	Recall	F1-Measure
Micro Averaged Measures	0.9408	0.9482	0.9408
Macro Averaged Measures	0.4704	0.4981	0.4865
Hamming Loss	0.0204		

In terms of loss minimization, hamming loss is investigated. Hamming loss is defined as the fraction of labels with incorrectly predicted relevance [152]. One-versus-All classifier performs the lowest hamming loss value. However, it ignores the label correlation, assuming that each class label is independent from each other. Furthermore, the conditional and marginal label dependence between labels is not considered. One possible reason that One-versus-All classifier shows the best performance on hamming loss is that it is well-tailored for hamming loss minimization. It may not have the good performance for other types of loss, such as subset 0/1 loss [152].

Table 5.4 shows other types of classifiers comparison results based on TF-IDF features. XGBoost classifier outperforms three other classifiers. Label Powerset (Logistic regression baseline) has the worst performance. The possible reason is that Label Powerset considers all possible label combinations in the model fitting process, which results in overfitting of the training dataset. However, Label Powerset with NB baseline shows better performance than Label Powerset with LR baseline.

Table 5.4: Classifier performance based on TF-IDF features

Classifier	Accuracy	Hamming Loss	Micro Averaged Measures			Macro Averaged Measures		
			Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Label Powerset (LR baseline)	0.4444	0.0597	0.5357	0.4839	0.5085	0.2767	0.3148	0.2552
Label Powerset (NB baseline)	0.7037	0.0267	0.8214	0.7419	0.7797	0.4296	0.4722	0.4323
Classifier Chain (NB baseline)	0.7007	0.0226	0.9167	0.7097	0.8000	0.4583	0.4583	0.4514
XGBoost Classifier	0.8007	0.0206	0.9200	0.7419	0.8214	0.4643	0.4676	0.4582
Binary Relevance (NB baseline)	0.6296	0.0329	0.9600	0.6154	0.7500	0.4889	0.3212	0.3613
Classifier Chain (LR baseline)	0.6667	0.0267	0.9523	0.6579	0.7937	0.6667	0.4713	0.5312
AdaBoost (DT baseline)	0.5556	0.0535	0.6486	0.6667	0.6316	0.4450	0.5024	0.4685
Label Powerset (DT baseline)	0.6296	0.0556	0.6824	0.6170	0.7632	0.5122	0.5888	0.5324

Classifier	Accuracy	Hamming Loss	Micro Averaged Measures			Macro Averaged Measures		
			Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Binary Relevance (DT baseline)	0.6667	0.0309	0.7761	0.8966	0.6842	0.5356	0.6389	0.4824
Decision Tree	0.5556	0.0453	0.7027	0.7222	0.6842	0.4845	0.5278	0.4778
ML-KNN	0.8235	0.0490	0.8718	0.8500	0.8947	0.8762	0.8889	0.9028

Figure 5.3 shows the accuracy of different classifiers. ML-KNN shows the highest value of accuracy, followed with XGBoost classifier, Label Powerset (NB baseline), and Classifier Chain (NB baseline). The larger value of accuracy indicates the better classification performance of classifiers. However, the smaller value of hamming loss means that the classifier has the better performance. As shown in Figure 5.4, XGBoost Classifier achieves the best classification performance, followed with ML-KNN.

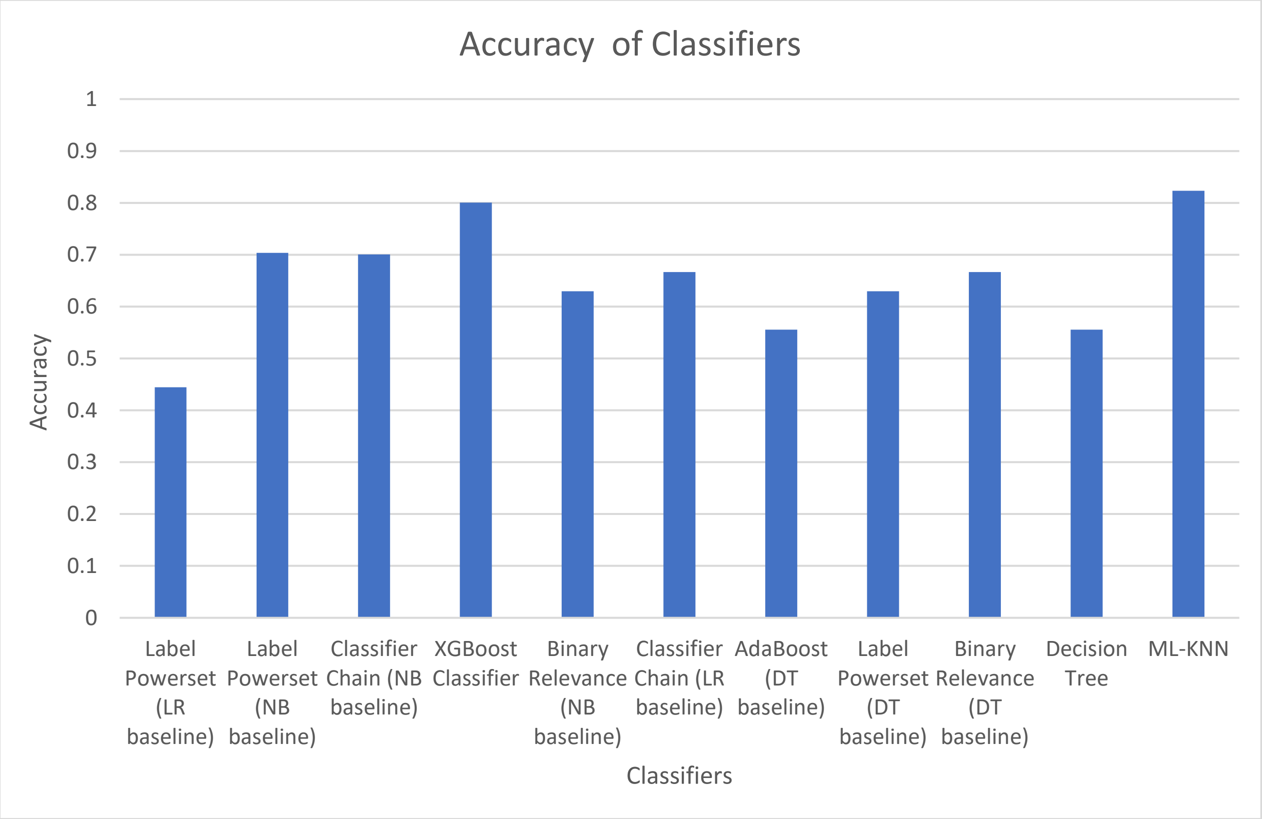


Figure 5.3: Accuracy of classifiers based on TF-IDF features

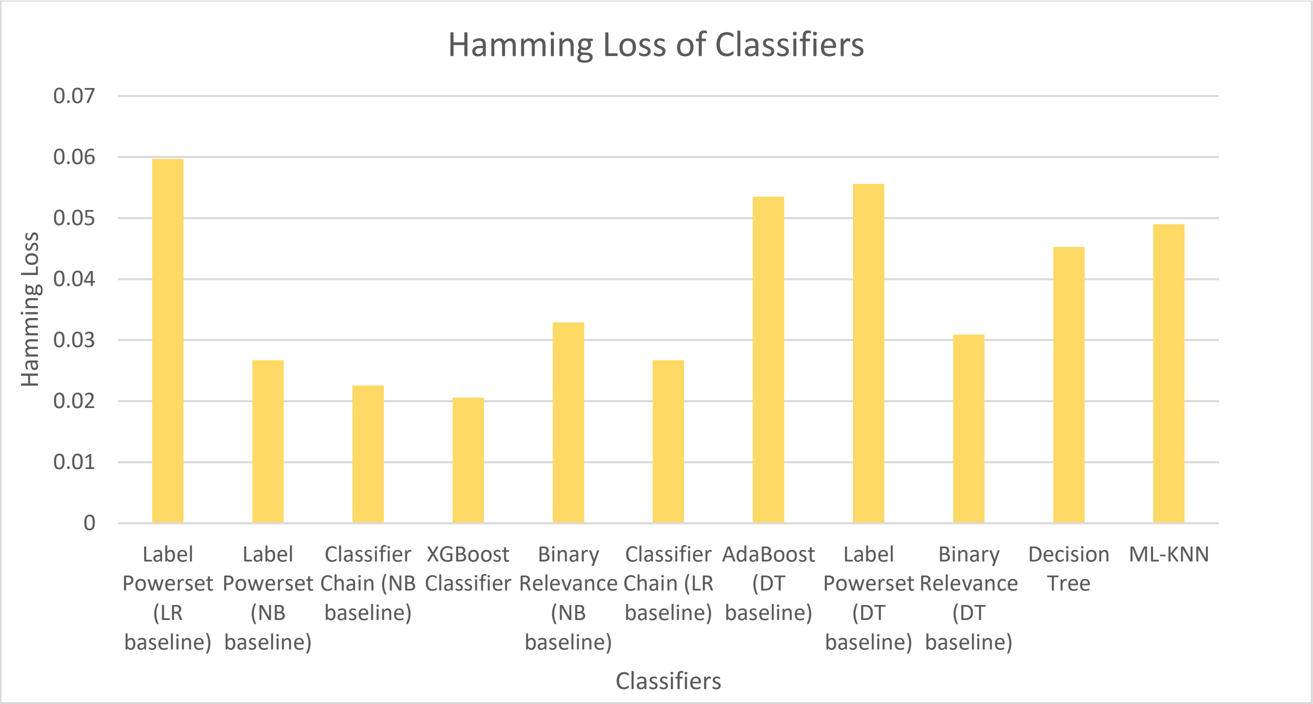


Figure 5.4: Hamming loss of classifiers based on TF-IDF features

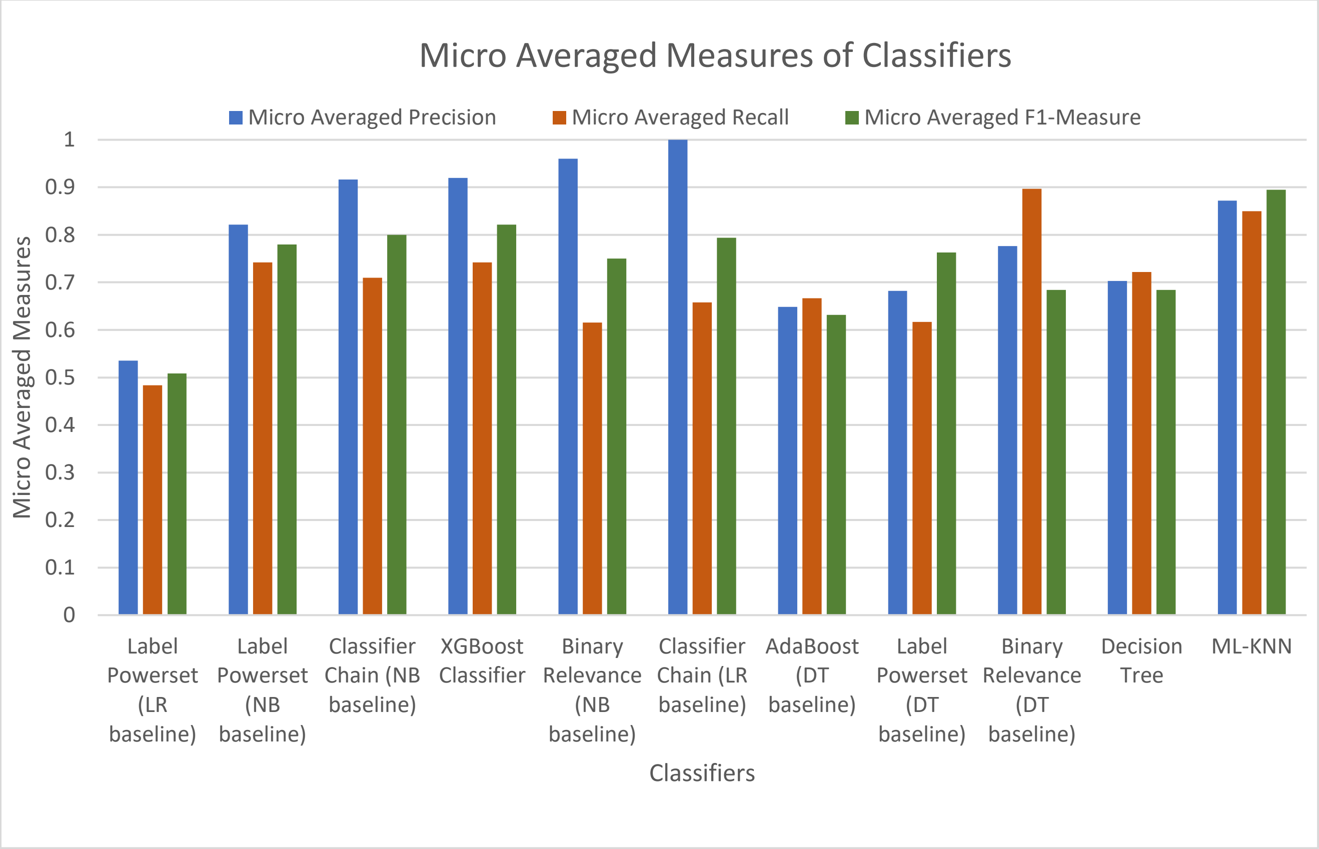


Figure 5.5: Micro averaged measures of classifiers based on TF-IDF features

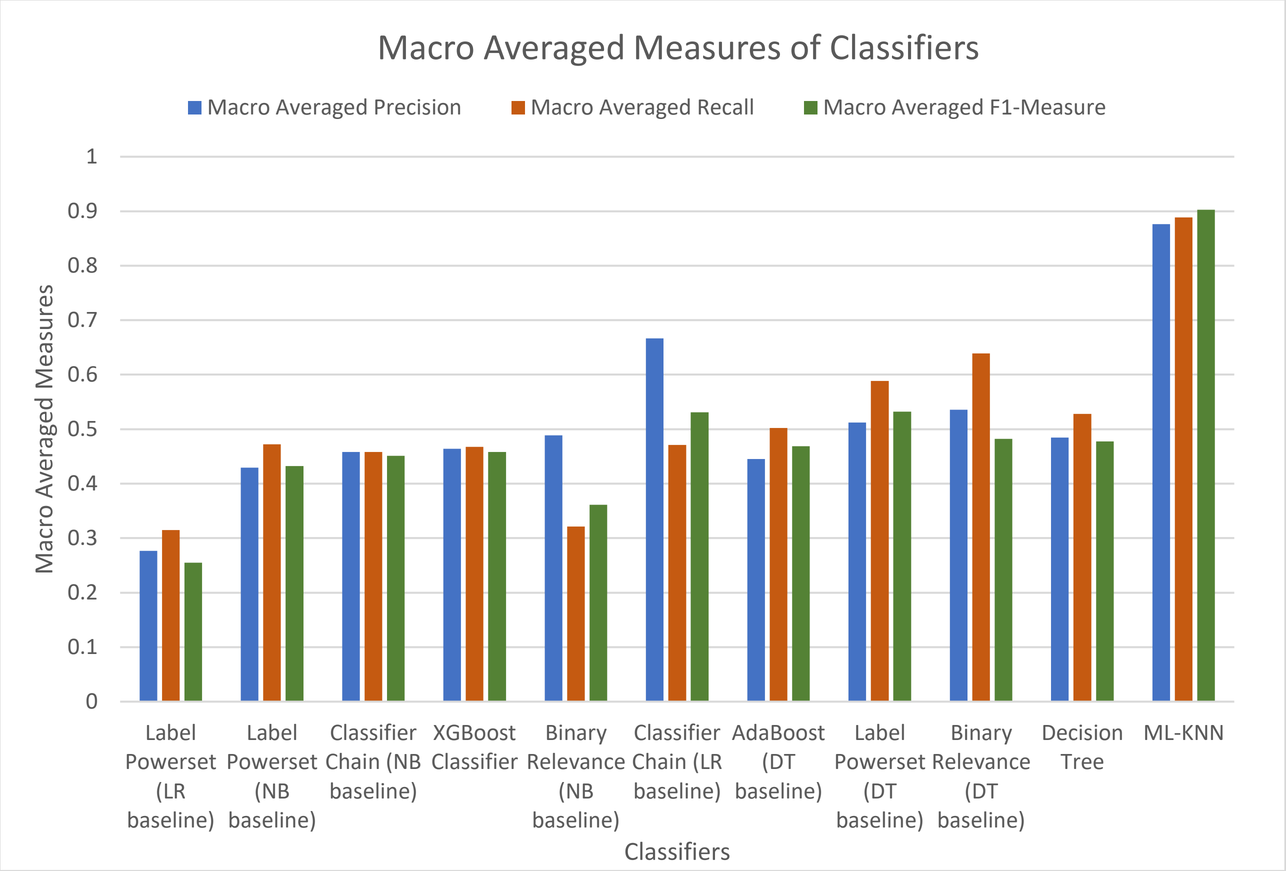


Figure 5.6: Macro averaged measures of classifiers based on TF-IDF features

As shown in Figure 5.5, in terms of micro-averaged precision, recall, and F1-measures, One-versus-All, ML-KNN and XGBoost classifiers outperform other classifiers. The larger value of micro averaged precision, recall, and F1-measure indicates the classifier has better performance. However, ML-KNN has the largest number on the macro averaged measures as shown in Figure 5.6. Most of classifiers have larger values of micro averaged measures than those of macro averaged measures. XGBoost classifier shows the greatest performance in terms of hamming loss. In general, each multi-label classification method has their advantages and disadvantages, and no classification model can be used in all situations. However, there are typically a few classifiers showing the greatest performance across all evaluation metrics. As a result, we can conclude that 1One-versus-All classifier, ML-KNN, and XGBoost outperform other classifiers. In terms of our dataset for bio-inspired design functionalities, One-versus-All classifier is significantly effective and practicable. However, if the label correlation needs to be considered, ML-KNN shows the best performance. In terms of hamming loss, XGBoost classifier outperforms all other classifiers.

5.3 Hyperparameter Tuning Analysis

Hyperparameter tuning is the process of modifying a few initialized parameters that have pretrained on a large corpus to adapt them to the new dataset. To investigate the effects of hyperparameter tuning on the classification model, a hyperparameter metric 'micro averaged measure' is specified. Then the model is tuned to maximize the value of micro averaged precision. RandomizedSearchCV is implemented during the tuning process to perform a randomized search over parameters, which can find the best parameters within the feasible space. To begin, the performance of various multi-label classifiers is compared. The classifiers that have the better performance are chosen from Table 5.4. Since One-versus-All, XGBoost, and ML-KNN classifier shows better performance than other classifiers, hyperparameter tuning is used to determine the optimal parameters for these classifiers. Table 5.5 shows the hyperparameter tuning for One-versus-All classifiers based on TF-IDF features. L2 regularization and the value 1 of C parameter would give the highest accuracy for One-versus-All classifier (Logistic Regression baseline). Table 5.6 shows the hyperparameter tuning for XGBoost Classifier based on TF-IDF features. Learning rate is an important hyperparameter in the classification task, as it determines whether the objective function can converge to a local minimum. Table 5.7 shows the hyperparameter tuning for ML-KNN based on TF-IDF features. When K is set to 3 and S is set to 0.1, the accuracy value of the model is the highest.

Table 5.5: Hyperparameter tuning for One-versus-All classifier based on TF-IDF features

One-versus-All (Logistic Regression baseline) Classifier		
Parameter	Possible Values	Hyperparameter tuning
Penalty	L1, L2	L2
C	0.000001,0.00001,0.0001,0.001,0.01,0.1, 1,10,100,1000,10000	1

Table 5.6: Hyperparameter tuning for XGBoost classifier based on TF-IDF features

XGBoost Classifier		
Parameter	Possible Values	Hyperparameter tuning
Learning rate	0.001-0.2	0.11799778766141077
N_estimators	10,50,100,250,500,750,1000,2000	50

XGBoost Classifier		
Parameter	Possible Values	Hyperparameter tuning
gamma	0-0.02	0.017507386547363852
Subsample	0.2,0.3,0.4,0.5,0.6,0.7,0.8	0.7
Reg_alpha	25,50,75,100,150,200	150
Reg_lambda	25,50,75,100,150,200	200
Max_depth	1-11	3
Colsample_bytree	0.2,0.3,0.4,0.5,0.6,0.7,0.8	0.2
Min_child_weight	1-11	5

Table 5.7: Hyperparameter for ML-KNN based on TF-IDF features

Multi-label K-Nearest Neighbors (ML-KNN)		
Parameter	Possible Values	Hyperparameter tuning
K	1-100	3
S	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8	0.1

5.4 Feature Extraction Analysis

Feature extraction algorithm is predicated on the assumption that important words appear more often in the text. It generates a list of words and converts the words into a feature set for training the classifier in the proposed system. This research uses TF-IDF and Bag-of-Words (BOW) as the main feature extraction method to convert the word into vectors. TF-IDF algorithm effectively balances the term frequency (the frequency with which a word occurs in the document) with the document's inverse term frequency (how often the word occurs across all documents). This means that words like "one" and "“they”" will score extremely low because they occur in every input variable of the dataset. Rarer terms, such as "superhydrophobic," will occur frequently in only a few documents discussing the bio-inspired design functionality. The word related to functionality will receive a higher TF-IDF score, and thus will be identified as the important word by the classification model to classify the bio-inspired design. Bag-of-Words (BOW) generates a set of vectors containing the count of word occurrences in the document, while the TF-IDF algorithm shows the most important word in the whole document. BOW is easier to interpret than TF-IDF. However, TF-IDF performs better in the proposed classification system. These two methods are combined to precisely describe the semantic information of the main information extracted from AskNature database. By selecting useful features from the dataset, the performance of the classification model is improved with minimal additional effort.

Table 5.8: Model performance based on BOW features

Bag-of-Words (BOW)	Binary Relevance (NB baseline)	XGBoost	ML-KNN	Label Powerset (DT baseline)
Micro Averaged Precision	0.8011	0.8323	0.6010	0.6444
Micro Averaged Recall	0.5161	0.6452	0.6771	0.5742
Micro Averaged F1-Measure	0.5961	0.7235	0.6220	0.6844

Table 5.9: Model performance based on TF-IDF (1-3 grams) features

TF-IDF (1-3grams)	Binary Relevance (NB baseline)	XGBoost	ML-KNN	Label Powerset (DT baseline)
Micro Averaged Precision	0.9600	0.9200	0.8718	0.6824
Micro Averaged Recall	0.6154	0.7419	0.8500	0.6170
Micro Averaged F1-Measure	0.7500	0.8214	0.8947	0.7632

Table 5.8 and Table 5.9 show the model performance of Binary Relevance (Naïve Bayes baseline), XGBoost, ML-KNN, and Label Powerset (Decision Tree baseline). They are compared in terms of different feature extraction methods. The micro-averaged precision, recall, and F1-measure of these classifiers based on TF-IDF features are larger than those classifiers based on BOW features. The possible reason is that TF-IDF can catch the semantic information from the text document and shows the most important and least important words in the document. TF-IDF extracts the more identical word than BOW. And BOW only reflects the number of occurrences of words in the document. Therefore, feature extraction based on TF-IDF shows the better performance. Comparing the performance of four classifiers, XGBoost outperforms three other classifiers in terms of micro averaged precision on TF-IDF. However, ML-KNN has better performance in terms of micro averaged recall and micro averaged F1-measure. For Binary Relevance (NB baseline), XGBoost, and ML-KNN, the classification performance is significantly improved by changing from BOW features to TF-IDF features. For Label Powerset (DT baseline), the performance slightly improved by changing from BOW features to TF-IDF features. Therefore, it

can be concluded that single feature extraction method has different effects on various training classifiers.

Table 5.10 shows the effect of number of features on the performance of classifiers. When TF-IDF is based on unigram, it extracts the same number of features as BOW, but it still outperforms BOW. TF-IDF based on only bigram shows the worst performance. The reason could be that the semantic meaning of sentence is changed when there’s only bigram extracted from dataset.

Table 5.10: Performance of different feature extraction methods

Feature Extraction Methods	Number of Features	ML-KNN Accuracy
BOW	991	0.81
TF-IDF_only_unigram	991	0.84
TF-IDF_only _bigram	2048	0.72
TF-IDF_unigram+bigram	3040	0.79
TF-IDF_unigram+bigram+trigram	5183	0.82

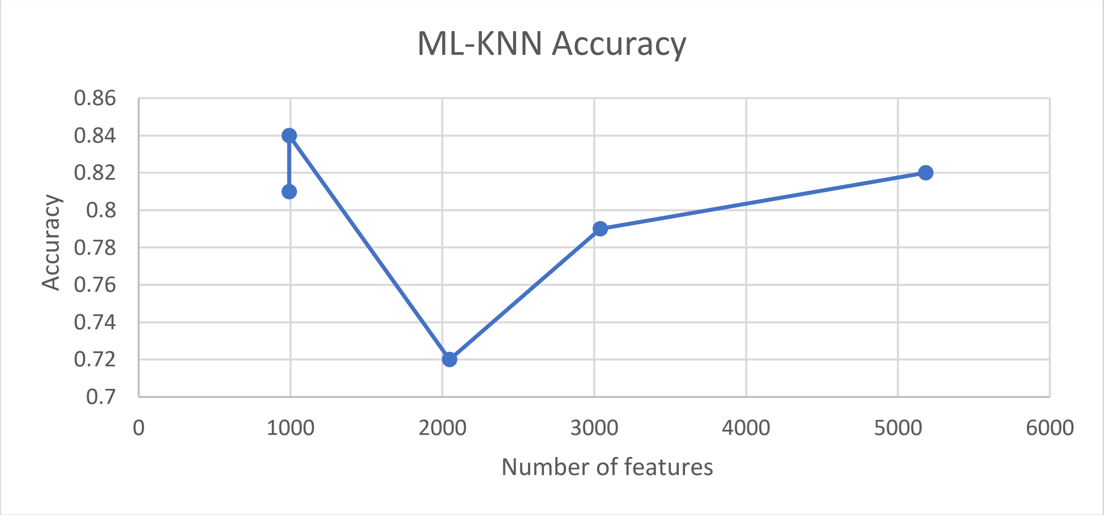
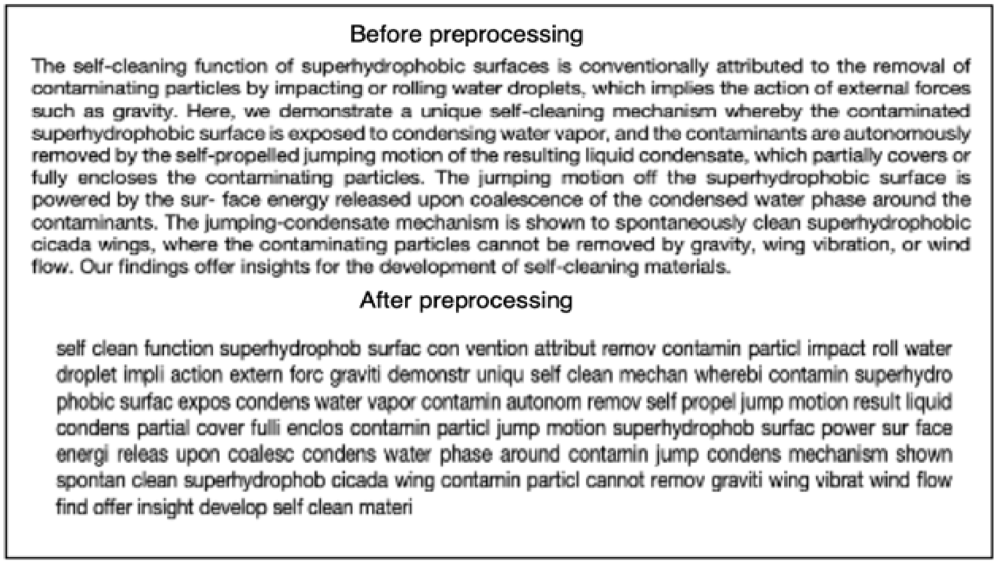


Figure 5.7: Number of features vs. ML-KNN Accuracy

As shown in Figure 5.7, when the number of features increases, the accuracy of the model does not see much improvement. When TF-IDF features are only based on unigram, ML-KNN classifier achieves with the highest accuracy of 84% as the number of features is 991. And ML-KNN shows the best performance when the feature set is smaller, since ML-KNN is an example or instance-based classifier, it performs very well with the limited training data. In many applications, the number of training documents used for KNN classifier is prone to be limited in order to accelerate the classification [153]. Training dataset is usually restricted to a small number of representative examples from each class for KNN classifier. Therefore, ML-KNN shows the best performance when the number of features is the least.

5.5 Case Study

One case study was conducted to validate the proposed system in this section. The case study is about the self-cleaning mechanism of superhydrophobic surfaces, which implies the action of external forces, and it is extracted from one AskNature page. Data preprocessing is critical in the text mining and natural language processing step. Stopwords, punctuations and special characters were removed from the original dataset during the preprocessing to facilitate the future analysis. In addition, stemming was used to convert words-forms to stems. Figure 5.8 shows the main information before and after data preprocessing. Special characters, punctuation such as "," "." "-" are removed, some stopwords such as "the", "of", "is", "of" are removed from the original dataset, then all uppercases are converted into the lowercases. In addition, all word forms are converted to their root forms. For example, the word "superhydrophobic" is converted to "superhydrophob". This step effectively improves the proposed system's performance and reduces the computational cost of the classification model. After the data preprocessing, the number of words decreases, which results in the smaller number of features extracted, which can improve the efficiency of feature extraction and representation before training the classifier.



Before preprocessing

The self-cleaning function of superhydrophobic surfaces is conventionally attributed to the removal of contaminating particles by impacting or rolling water droplets, which implies the action of external forces such as gravity. Here, we demonstrate a unique self-cleaning mechanism whereby the contaminated superhydrophobic surface is exposed to condensing water vapor, and the contaminants are autonomously removed by the self-propelled jumping motion of the resulting liquid condensate, which partially covers or fully encloses the contaminating particles. The jumping motion off the superhydrophobic surface is powered by the sur- face energy released upon coalescence of the condensed water phase around the contaminants. The jumping-condensate mechanism is shown to spontaneously clean superhydrophobic cicada wings, where the contaminating particles cannot be removed by gravity, wing vibration, or wind flow. Our findings offer insights for the development of self-cleaning materials.

After preprocessing

self clean function superhydrophob surfac con vention attribut remov contamin particl impact roll water droplet impli action extern forc graviti demonstr uniqu self clean mechan wherebi contamin superhydro phobic surfac expos condens water vapor contamin autonom remov self propel jump motion result liquid condens partial cover fulli enclos contamin particl jump motion superhydrophob surfac power sur face energi releas upon coalesc condens water phase around contamin jump condens mechanism shown spontan clean superhydrophob cicada wing contamin particl cannot remov graviti wing vibrat wind flow find offer insight develop self clean materi

Figure 5.8: Main information before and after preprocessing

Following TF-IDF vectorization, the TF-IDF score of each word in the document was calculated to determine the keywords of the selected page. Table 5.11 summarizes the top 15 keywords on this page. Based on the extracted keywords, the classifier in the proposed system can classify the selected AskNature page into one of the design functions 'superhydrophobic'. This case study validates the effectiveness and practicability of the proposed classification system for bio-inspired design.

Table 5.11: Top 15 keywords in the selected AskNature page

Top Keywords	
1. function	9. implies
2. surfaces	10. action
3. conventionally	11. forces
4. attributed	12. exposed
5. removal	13. condensing
6. impacting	14. Superhydrophobic
7. rolling	15. water
8. droplets	

5.6 Discussion

In this section, several aspects of this research are discussed, including several advantages and limitations of the proposed system, difficulties associated with dataset collection and main contributions of this research.

5.6.1 The Advantages and Limitations of the Proposed System

The proposed system has the potential to significantly reduce the time of classifying the bio-inspired design corresponding to their design functionalities. This system can be used in conjunction with Domain Integrated Design method for bio-inspired design in the near future. It can also be used to assist the construction of bio-inspired design related database. However, the proposed system has its limitations. Firstly, it is difficult to determine which information is relevant to bio-inspired design and which is not during the process of dataset collection. The lack of standard nomenclature for describing complex bio-inspired design is another challenge during the process of dataset collection, since there is no standard set up within one journal or conference paper. In addition, positive bias exists during the process of dataset collection, as the researchers would tend to omit or leave out the less useful data to the appendix, which possibly prevents the future researcher from determining the boundaries of possible NLP and text mining results. The impact of human bias to the classification accuracy in the dataset collection process has not been thoroughly investigated and still be considered as a challenge. In terms of feature extraction method, TF-IDF approach is a relatively straightforward and fast method for extracting keywords from a corpus and transforming textual data into numerical vectors, but it ignores the contextual information of each word and has poor performance on a large dimensionality of dataset. In terms of the performance of different classifiers, One-versus-All classifier outperforms all other classifiers, and it can provide information about the class by inspecting its matching classifier, however, it ignores the label correlation. Label Powerset considers the correlation between labels, but it does not consider all possible label combinations during the model development step. If the number of labels is large, Label Powerset is possible to have the issue of overfitting. And ML-KNN shows the better performance on a smaller dataset, as it performed well with smaller number of feature sets.

Another limitation of this classification system is the small size of the training data. Although all existing bio-inspired designs on the AskNature database are checked, this results in only 18 labelled bio-inspired design functionalities. This is a very small dataset for classification task. This desire for additional bio-inspired design functionalities, as well as the desire to develop the classifier for additional design functionalities, suggests that this research can be extended in the near future. Bio-inspired design classifiers discussed in this research should be integrated into a publicly search tool/website, which enables the researchers to find the relevant bio-inspired design passages from a bio-inspired database. This search tool should be able to build the sets of training data continuously to train the classifier, as the classifier is adjusted based on the accumulated training data, the output label should be more relevant to the researchers.

5.6.2 Main Contributions of the Research

The proposed system facilitates the interpretability and usability of various classifiers, along with an empirical study of various multi-label algorithms, their applications, and evaluation metrics. Also, the relationships between various algorithms and the advantages and disadvantages of each classifier are introduced as well. Although the collected dataset from AskNature is relatively small, several classifiers including One-versus-All, XGBoost, and ML-KNN classifier can successfully predict the functionalities of bio-inspired design based on the training dataset. The proposed system can effectively and efficiently classify the bio-inspired design corresponding to their design functionalities. In terms of the dataset collected in this research, label correlation does not cause any issue; however, investigating the existence of conditional and unconditional dependency between different labels is another possible extension of future work.

This research may serve as an incentive for engineers and biologists to collaborate, as knowledge transfer between biological and engineering systems is more rapid and efficient based on the classification system. Additionally, this proposed system will gradually reduce the time required to generate a process for resolving engineering problems using naturally occurring biological solutions. It has the potential to be introduced into the engineering field soon as a useful tool to assist designers. One advantage of this proposed system integrates the text mining, natural language processing with machine learning model. Finally, the classification system bridges the biological knowledge with Domain Integrated Design to support the design ideation in terms of multiple domains, such as surfaces, cellular structures, cross-sections, and shapes.

6 Conclusions and Future Work

6.1 Conclusions

Bio-inspired design is a significant innovation that draws inspiration from nature and bridges the biological system with engineering knowledge. Designer can develop innovative engineering solutions by leveraging bio-inspired design knowledge that found from the nature. To solve the difficulties on extracting design knowledge from text-based database about the bio-inspired design, bio-inspired design classification system is developed to classify bio-inspired designs to support design ideation. In the proposed bio-inspired design classification system which consists of multiple classifiers, One-versus-All classifier outperforms than all other classifiers. It can achieve a micro averaged precision of 94.08%, a micro averaged recall of 94.82%, a micro averaged F1-measure of 94.08% and an accuracy of 92.7%. This demonstrates that One-versus-All classifier can classify the bio-inspired design corresponding to their multi-functionalities effectively and efficiently. It is a promising approach to extract useful knowledge from bio-inspired design into solving engineering problems. And the proposed system can be a feasible solution for categorizing the bio-inspired design on a functional basis. One important contribution of this work is that this classification task is tied to the function basis. The highest label accuracy of One-versus-All classifier can achieve 96.3%. The label dependence does not cause any issue in the collected dataset. In addition, different feature extraction methods are compared in terms of classification performance. TF-IDF features have better performance than BOW features. One case study is also proposed to verify and validate the effectiveness and practicability of the proposed classification system. The primary limitation of this research is the small size of the dataset. With searches from the AskNature database, only 90 pages were selected since they are related to multi-functional bio-inspired design. This is a very small dataset for classification task. Thus, the performance of classification system can be improved by increasing the dataset size.

A comparative evaluation of these multi-label classification algorithms on a multi-label dataset composed of 90 instances and 18 class labels is conducted. The results show that the One-versus-All classifier is effective for bio-inspired design classification, followed by ML-KNN and XGBoost classifier. Table 6.1 shows the top three classifiers with their configurations. Problem transformation and algorithm adaptation classifiers perform much better than the baseline classifier. In addition, One-versus-All classifier achieves the lowest hamming loss and the highest micro averaged measures on the dataset. This classification system achieves superior performance on the dataset with TD-IDF as the feature representation. This system extracts the main information from the AskNature database and cleaned the data by using tools and techniques from NLP and text mining. The proposed system integrates NLP, text mining techniques with several machine learning models to make the classification on bio-inspired design. This research can motivate the collaborations between engineers and biologists since the knowledge transfer between biological sand engineering systems is faster and more efficient. Furthermore, this proposed system will gradually reduce the time required to classify the bio-inspired design on a functional basis. It has the potential to be introduced into the engineering field as a knowledge extraction tool to aid designers in the near future.

Table 6.1: Classifiers with great performance

Method	**Classifier**	**Configuration**	**Accuracy**
Problem Transformation Method	One-versus-All Classifier	One-versus-All Classifier with Logistic Regression baseline Penalty = L2, C = 1	92.7%
Algorithm Adaptation Method	XGBoost Classifier	Number of booting rounds parameter set to 50, Lambada = 200, Alpha = 150	80.07%
	ML-KNN	Multi-label K-Nearest Neighbors with K = 3, S = 0.1	82.35%

6.2 Directions for Future Work

In the future, several works need to be carried out. Bio-inspired design classification system should be extended to other types of knowledge in the biological system, such as images, videos, mixed knowledge of textual and image data. In addition, more deep learning classifiers can be developed to improve the classification accuracy, precision, recall and F1-measure. Based on the results in this research, the classification should be further refined by including more semantic information, which may result in more precise and faster results.

Furthermore, developing an integrated search and classification tool represents a very promising area of future research. This tool should enable researchers from diverse fields to retrieve knowledge from biological corpus based on design functionalities and to keep track of the relationship between biological solutions and engineering problems. This search tool should be capable of continuously accumulating training data across multiple uses, which should result in the increasing accuracy of the refining classification system.

In addition, more machine learning packages can be integrated into the system to determine whether the accuracy can be improved, such as BERT developed by Google, MULAN, and WEKA. Incorporating this classification system into a multi-language support classification system is another possible extension of this work. Various coarse, medium, and fine categories for of bio-inspired design labels can be examined to determine the effects of feature complexity on system performance. Additionally, the multi-class and multi-instances approaches, as well as the hierarchical multi-label classification approach can be learned.

Latent Semantic Analysis (LSA) is a natural language processing technique for grouping topics that are similar to one another. This technique can be tested on the collected dataset in this research to see the performance. In the future, a failure analysis will be required to aid the development of a more sophisticated classifier that includes pipelines step such as co-reference analysis. The analysis of false positives and negatives can help understanding why the classification model makes an error. Furthermore, the computational complexity of each classifier within the proposed system should be compared, since the computational and time costs are critical factors when running in a real-world application.

References

[1] H. Hashemi Farzaneh, "Bio-inspired design: the impact of collaboration between engineers and biologists on analogical transfer and ideation," (in English), *Research in Engineering Design,* vol. 31, no. 3, pp. 299-322, Jul 2020, doi: 10.1007/s00163-020-00333-w.

[2] V. VDI, "6220: Conception and Strategy—differences between biomimetic and conventional methods/products," ed: Verein Deutscher Ingenieure, Beuth Verlag: Berlin, 2012.

[3] R. B. Stone, A. K. Goel, and D. A. McAdams, "Charting a Course for Computer-Aided Bio-Inspired Design," in *Biologically Inspired Design*: Springer London, 2014, ch. Chapter 1, pp. 1-16.

[4] W. Nachtigall, *Bionik als Wissenschaft: Erkennen-Abstrahieren-Umsetzen*. Springer-Verlag, 2010.

[5] J. Benyus, "Foreword: curating nature's patent database," *Biologically inspired design—computational methods and tools. Springer, London, p vii–xi,* 2014.

[6] J. M. Benyus, *Biomimicry: Innovation inspired by nature*. Morrow New York, 1997.

[7] K. Fu, D. Moreno, M. Yang, and K. L. Wood, "Bio-Inspired Design: An Overview Investigating Open Questions From the Broader Field of Design-by-Analogy," *Journal of Mechanical Design,* vol. 136, no. 11, p. 111102, 2014, doi: 10.1115/1.4028289.

[8] C. Alberto, "The Bio-Inspired Design Landscape: Industrial Design. BioInspired!," *accessed, Dec,* vol. 28, p. 2013, 2010.

[9] A. G. Domel, M. Saadat, J. C. Weaver, H. Haj-Hariri, K. Bertoldi, and G. V. Lauder, "Shark skin-inspired designs that improve aerodynamic performance," *J R Soc Interface,* vol. 15, no. 139, p. 20170828, Feb 2018, doi: 10.1098/rsif.2017.0828.

[10] M. Zhang, S. Feng, L. Wang, and Y. Zheng, "Lotus effect in wetting and self-cleaning," *Biotribology,* vol. 5, pp. 31-43, 2016.

[11] Z. Han, B. Zhu, M. Yang, S. Niu, H. Song, and J. Zhang, "The effect of the micro-structures on the scorpion surface for improving the anti-erosion performance," (in English), *Surface and Coatings Technology,* vol. 313, pp. 143-150, Mar 15 2017, doi: 10.1016/j.surfcoat.2017.01.061.

[12] C. Tiner, S. Bapat, S. D. Nath, S. V. Atre, and A. Malshe, "Exploring Convergence of Snake-Skin-Inspired Texture Designs and Additive Manufacturing for Mechanical Traction," (in English), *Procedia Manufacturing,* vol. 34, pp. 640-646, 2019, doi: 10.1016/j.promfg.2019.06.116.

[13] C. Yu *et al.*, "Bio-inspired drag reduction: From nature organisms to artificial functional surfaces," *Giant,* vol. 2, p. 100017, 2020/06/01/ 2020, doi: 10.1016/j.giant.2020.100017.

[14] P. L. Gibson. "Lecture 18, Nature Sandwich Notes." https://ocw.mit.edu/courses/materials-science-and-engineering/3-054-cellular-solids-structure-properties-and-applications-spring-2015/lecture-notes/MIT3_054S15_L18_Nat_trans.pdf (accessed.

[15] P. T. Velivela, N. Letov, Y. Liu, and Y. F. Zhao, "Application of Domain Integrated Design Methodology for Bio-Inspired Design- a Case Study of Suture Pin Design," *Proceedings of the Design Society,* vol. 1, pp. 487-496, 2021, doi: 10.1017/pds.2021.49.

[16] U. Lindemann and J. Gramann, "Engineering design using biological principles," in *DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia*, 2004, pp. 355-360.

[17] T. Lenau, A. Dentel, Þ. Ingvarsdóttir, and T. Guđlaugsson, "Engineering design of an adaptive leg prosthesis using biological principles," in *DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia*, 2010.

[18] M. Helms, S. S. Vattam, and A. K. Goel, "Biologically inspired design: process and products," *Design studies,* vol. 30, no. 5, pp. 606-622, 2009.

[19] A. K. Goel, S. Vattam, M. Helms, and B. Wiltgen, "An information-processing account of creative analogies in biologically inspired design," in *Proceedings of the 8th ACM conference on Creativity and Cognition*, 2011, pp. 71-80.

[20] J.-M. Deldin and M. Schuknecht, "The AskNature database: enabling solutions in biomimetic design," in *Biologically inspired design*: Springer, 2014, pp. 17-27.

[21] B. Hill, *Innovationsquelle natur: naturorientierte innovationsstrategie für entwickler, konstrukteure und designer*. Shaker Verlag, 1997.

[22] S. Löffler, *Anwenden bionischer Konstruktionsprinzipe in der Produktentwicklung* (no. 73). Logos Verlag Berlin GmbH, 2009.

[23] C. F. Salgueiredo and A. Hatchuel, "Beyond analogy: A model of bioinspiration for creative design," *Ai Edam,* vol. 30, no. 2, pp. 159-170, 2016.

[24] J. K. Nagel, R. B. Stone, and D. A. McAdams, "An engineering-to-biology thesaurus for engineering design," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2010, vol. 44137, pp. 117-128.

[25] S. S. Vattam, M. E. Helms, and A. K. Goel, "Compound analogical design: interaction between problem decomposition and analogical transfer in biologically inspired design," in *Design computing and cognition'08*: Springer, 2008, pp. 377-396.

[26] K. Helten, S. Schenkl, and U. Lindemann, "Biologizing product development—results from a student project," in *ICORD 11: Proceedings of the 3rd International Conference on Research into Design Engineering, Bangalore, India, 10.-12.01. 2011*, 2011.

[27] T. A. Lenau, A.-L. Metze, and T. Hesselberg, "Paradigms for biologically inspired design," in *Bioinspiration, Biomimetics, and Bioreplication VIII*, 2018, vol. 10593: International Society for Optics and Photonics, p. 1059302.

[28] C. Chen, Y. Tao, Y. Li, Q. Liu, S. Li, and Z. Tang, "A structure-function knowledge extraction method for bio-inspired design," *Computers in Industry,* vol. 127, p. 103402, 2021.

[29] R. Muller *et al.*, "Biodiversifying bioinspiration," *Bioinspir Biomim,* vol. 13, no. 5, p. 053001, Jul 3 2018, doi: 10.1088/1748-3190/aac96a.

[30] J. F. Vincent, "Biomimetics—a review," *Proceedings of the institution of mechanical engineers, part H: Journal of Engineering in Medicine,* vol. 223, no. 8, pp. 919-939, 2009. [Online]. Available: https://journals.sagepub.com/doi/10.1243/09544119JEIM561?url_ver=Z39.88-2003&rfr_id=ori:rid:crossref.org&rfr_dat=cr_pub%3dpubmed.

[31] P. E. Fayemi, K. Wanieck, C. Zollfrank, N. Maranzana, and A. Aoussat, "Biomimetics: process, tools and practice," *Bioinspir Biomim,* vol. 12, no. 1, p. 011002, Jan 23 2017, doi: 10.1088/1748-3190/12/1/011002.

[32] L. Shu, K. Ueda, I. Chiu, and H. Cheong, "Biologically inspired design," *CIRP annals,* vol. 60, no. 2, pp. 673-693, 2011.

[33] D. Vandevenne, P.-A. Verhaegen, S. Dewulf, and J. R. Duflou, "SEABIRD: Scalable search for systematic biologically inspired design," *Ai Edam,* vol. 30, no. 1, pp. 78-95, 2016.

[34] E. Graeff, N. Maranzana, and A. Aoussat, "Biological Practices and Fields, Missing Pieces of the Biomimetics' Methodological Puzzle," *Biomimetics (Basel),* vol. 5, no. 4, p. 62, Nov 18 2020, doi: 10.3390/biomimetics5040062.

[35] G. Pahl, W. Beitz, J. Feldhusen, and K. Grote, "Engineering Design: A Systematic Approach Third Edition," *Berlin, Springer Science+ Business Media Deutschland GmbH, 2007. 632,* 2007.

[36] D. Sysaykeo, E. Mermoz, and T. Thouveny, "Clearance and design optimization of bio-inspired bearings under off-center load," *CIRP Annals,* vol. 69, no. 1, pp. 121-124, 2020.

[37] M. W. Glier, J. Tsenn, J. S. Linsey, and D. A. McAdams, "Evaluating the directed intuitive approach for bioinspired design," *Journal of Mechanical Design,* vol. 136, no. 7, 2014.

[38] E. B. Kennedy, D. J. Miller, and P. H. Niewiarowski, "Industrial and biological analogies used creatively by business professionals," *Creativity Research Journal,* vol. 30, no. 1, pp. 54-66, 2018.

[39] M. B. Hesse, "Models and Analogies in Science. 2. print," *Notre Dame, Ind,* 1970.

[40] G. Hooker and E. Smith, "AskNature and the biomimicry taxonomy," *Insight,* vol. 19, no. 1, pp. 46-49, 2016.

[41] R. B. Stone and K. L. Wood, "Development of a functional basis for design," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 1999, vol. 19739: American Society of Mechanical Engineers, pp. 261-275.

[42] S. S. Vattam and A. K. Goel, "Foraging for inspiration: understanding and supporting the online information seeking practices of biologically inspired designers," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2011, vol. 54860, pp. 177-186.

[43] H. Cheong and L. Shu, "Retrieving causally related functions from natural-language text for biomimetic design," *Journal of Mechanical Design,* vol. 136, no. 8, p. 081008, 2014.

[44] "Design by Analogy to Nature Engine." http://dilab.cc.gatech.edu/dane/ (accessed Feb 18, 2022).

[45] B. Wiltgen, A. K. Goel, and S. Vattam, "Representation, indexing, and retrieval of biological cases for biologically inspired design," in *International Conference on Case-Based Reasoning*, 2011: Springer, pp. 334-347.

[46] J. F. Vincent, O. A. Bogatyreva, N. R. Bogatyrev, A. Bowyer, and A. K. Pahl, "Biomimetics: its practice and theory," *J R Soc Interface,* vol. 3, no. 9, pp. 471-82, Aug 22 2006, doi: 10.1098/rsif.2006.0127.

[47] J. F. Vincent and D. L. Mann, "Systematic technology transfer from biology to engineering," *Philos Trans A Math Phys Eng Sci,* vol. 360, no. 1791, pp. 159-73, Feb 15 2002, doi: 10.1098/rsta.2001.0923.

[48] G. Altshuller, "Suddenly the Inventor Appeared. TRIZ, the Theory of Inventive Problem Solving. 6th Printing. USA. Technical Innovation Center," ed: Inc, 2004.

[49] T. Mak and L. Shu, "Abstraction of biological analogies for design," *CIRP Annals,* vol. 53, no. 1, pp. 117-120, 2004.

[50] R. L. Nagel, P. A. Midha, A. Tinsley, R. B. Stone, D. A. McAdams, and L. Shu, "Exploring the use of functional models in biomimetic conceptual design," 2008.

[51] J. S. Linsey, K. L. Wood, and A. B. Markman, "Modality and representation in analogy," *Ai Edam,* vol. 22, no. 2, pp. 85-100, 2008.

[52] A. K. Goel, S. Rugaber, and S. Vattam, "Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language," *Ai Edam,* vol. 23, no. 1, pp. 23-35, 2009.

[53] J. K. Nagel and R. B. Stone, "A systematic approach to biologically-inspired engineering design," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2011, vol. 54860, pp. 153-164.

[54] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood, "A functional basis for engineering design: reconciling and evolving previous efforts," *Research in engineering Design,* vol. 13, no. 2, pp. 65-82, 2002.

[55] S. Venkataraman and A. Chakrabarti, "SAPPhIRE–an approach to analysis and synthesis," in *DS 58-2: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 2, Design Theory and Research Methodology, Palo Alto, CA, USA, 24.-27.08. 2009*, 2009, pp. 417-428.

[56] A. Chakrabarti, P. Sarkar, B. Leelavathamma, and B. Nataraju, "A functional representation for aiding biomimetic and artificial inspiration of new ideas," *Ai Edam,* vol. 19, no. 2, pp. 113-132, 2005.

[57] M. Helms and A. K. Goel, "The four-box method: Problem formulation and analogy evaluation in biologically inspired design," *Journal of Mechanical Design,* vol. 136, no. 11, p. 111106, 2014.

[58] H. Cheong and L. Shu, "Using templates and mapping strategies to support analogical transfer in biomimetic design," *Design Studies,* vol. 34, no. 6, pp. 706-728, 2013.

[59] F. Jebbor and L. Benhlima, "Overview of knowledge extraction techniques in five question-answering systems," in *2014 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14)*, 2014: IEEE, pp. 1-8.

[60] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge discovery in databases: An overview," *AI magazine,* vol. 13, no. 3, pp. 57-57, 1992.

[61] R. Upadhyay and A. Fujii, "Semantic Knowledge Extraction from Research Documents," presented at the Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, 2016.

[62] O. Kononova, T. He, H. Huo, A. Trewartha, E. A. Olivetti, and G. Ceder, "Opportunities and challenges of text mining in aterials research," (in English), *iScience,* vol. 24, no. 3, p. 102155, Mar 19 2021, doi: 10.1016/j.isci.2021.102155.

[63] Y. Lallemant and M. S. Fox, "IntelliServe™: Automating Customer Service," in *Proceedings of the AAAI-99 Workshop on AI for Electronic Commerce, USA*, 1998.

[64] R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks, "University of Sheffield: Description of the LaSIE system as used for MUC-6," in *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995.

[65] L. Kosseim, L. Plamondon, and G. Lapalme, "La réponse automatique comme solution à la gestion des relations avec la clientèle," *Revues des sciences et technologies de l'information (RSTI) série Ingénierie des systèmes d'information (ISI),* vol. 8, no. 3, pp. 91-114, 2003.

[66] H. Sherzod, T. Hakan, A. Marlen, and D. Erdogan, "Semantic question answering system over linked data using relational patterns," *TOBB University of Economics and Technology,* 2013.

[67] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," *Journal of emerging technologies in web intelligence,* vol. 1, no. 1, pp. 60-76, 2009.

[68] D. H. Goh and R. P. Ang, "An introduction to association rule mining: an application in counseling and help-seeking behavior of adolescents," *Behav Res Methods,* vol. 39, no. 2, pp. 259-66, May 2007, doi: 10.3758/bf03193156.

[69] P. C. Wong, P. Whitney, and J. Thomas, "Visualizing association rules for text mining," in *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, 1999: IEEE, pp. 120-123.

[70] A. F. Damerau and A. Weiss, "Text mining with decision trees and decision rules," in *Conference on Automated Learning and Discovery*, 1998.

[71] S. Jusoh and H. M. Alfawareh, "Techniques, applications and challenging issue in text mining," *International Journal of Computer Science Issues (IJCSI),* vol. 9, no. 6, p. 431, 2012.

[72] A.-H. Tan, "Text mining: The state of the art and the challenges," in *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases*, 1999, vol. 8: Citeseer, pp. 65-70.

[73] Y. Huai, "Study on ontology-based personalized user modeling techniques in intelligent information retrievals," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011: IEEE, pp. 204-207.

[74] M. Rajman and R. Besançon, "Text mining-knowledge extraction from unstructured textual data," in *Advances in data science and classification*: Springer, 1998, pp. 473-480.

[75] P. Clark and P. Harrison, "Large-scale extraction and use of knowledge from text," in *Proceedings of the fifth international conference on Knowledge capture*, 2009, pp. 153-160.

[76] H. Cheong and L. H. Shu, "Retrieving Causally Related Functions From Natural-Language Text for Biomimetic Design," (in English), *Journal of Mechanical Design,* vol. 136, no. 8, Aug 2014, doi: 10.1115/1.4027494.

[77] A. M. Cohen and W. R. Hersh, "A survey of current work in biomedical text mining," *Brief Bioinform,* vol. 6, no. 1, pp. 57-71, Mar 2005, doi: 10.1093/bib/6.1.57.

[78] I. Donaldson *et al.*, "PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine," *BMC bioinformatics,* vol. 4, no. 1, pp. 1-13, 2003.

[79] M. Alkahtani, A. Choudhary, A. De, and J. A. Harding, "A decision support system based on ontology and data mining to improve design using warranty data," *Computers & industrial engineering,* vol. 128, pp. 1027-1039, 2019.

[80] A. Koneru, "Knowledge extraction from work instructions through text processing and analysis," M.S., Clemson University, Ann Arbor, 1551413, 2013. [Online]. Available: https://proxy.library.mcgill.ca/login?url=https://www.proquest.com/dissertations-theses/knowledge-extraction-work-instructions-through/docview/1498532528/se-2?accountid=12339

https://mcgill.on.worldcat.org/atoztitles/link?sid=ProQ:&issn=&volume=&issue=&title=Knowledge+extraction+from+work+instructions+through+text+processing+and+analysis&spage=&date=2013&atitle=&au=Koneru%2C+Abhiram&id=&isbn=978-1-303-69137-9

[81] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition," ed.

[82] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, "Data mining in manufacturing: a review based on the kind of knowledge," *Journal of Intelligent Manufacturing,* vol. 20, no. 5, pp. 501-521, 2009.

[83] I. Chiu Forrest, "Natural language analysis to support biomimetic design," M.A.Sc., University of Toronto (Canada), Ann Arbor, MR07404, 2005. [Online]. Available: https://proxy.library.mcgill.ca/login?url=https://www.proquest.com/dissertations-theses/natural-language-analysis-support-biomimetic/docview/305369598/se-2?accountid=12339

https://mcgill.on.worldcat.org/atoztitles/link?sid=ProQ:&issn=&volume=&issue=&title=Natural+language+analysis+to+support+biomimetic+design&spage=&date=2005&atitle=&au=Chiu+Forrest%2C+Ivey&id=&isbn=978-0-494-07404-6

[84] E. Brill and R. J. Mooney, "An overview of empirical natural language processing," *AI magazine,* vol. 18, no. 4, pp. 13-13, 1997.

[85] H. Cheong, I. Chiu, L. Shu, R. B. Stone, and D. A. McAdams, "Biologically meaningful keywords for functional terms of the functional basis," *Journal of Mechanical Design,* vol. 133, no. 2, 2011.

[86] H. Cheong, L. Shu, R. B. Stone, and D. A. McAdams, "Translating terms of the functional basis into biologically meaningful keywords," in *International design engineering*

technical conferences and computers and information in engineering conference, 2008, vol. 43284, pp. 137-148.

[87] L. Shu and H. Cheong, "A natural language approach to biomimetic design," in *Biologically inspired design*: Springer, 2014, pp. 29-61.

[88] H. Cheong, G. Hallihan, and L. Shu, "Understanding analogical reasoning in biomimetic design: An inductive approach," in *Design computing and cognition'12*: Springer, 2014, pp. 21-39.

[89] H. Cheong, G. M. Hallihan, and L. Shu, "Design problem solving with biological analogies: A verbal protocol study," *AI EDAM,* vol. 28, no. 1, pp. 27-47, 2014.

[90] S. Jiang, J. Hu, K. L. Wood, and J. Luo, "Data-Driven Design-By-Analogy: State-of-the-Art and Future Directions," (in English), *Journal of Mechanical Design,* vol. 144, no. 2, Feb 1 2022, doi: 10.1115/1.4051681.

[91] D. B. Cleveland and A. D. Cleveland, "Introduction to indexing and abstracting. Englewood," *Colorado: Libraries Unlimited,* 1990.

[92] G. K. Zipf, *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.

[93] G. Leech, P. Rayson, and A. Wilson, "Companion Website for: Word Frequencies in Written and Spoken English: based on British Nat. Corpus," ed, 2001.

[94] in *The Oxford English Dictionary*, 2nd ed. OED Online: Oxford University Press, 4 Apr. 2000.

[95] B. W. Medlock, "Investigating classification for natural language processing tasks," University of Cambridge, Computer Laboratory, 2008.

[96] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys,* vol. 34, no. 1, pp. 1-47, 2002, doi: 10.1145/505282.505283.

[97] N. W. Svendsen and T. A. Lenau, "Approaches to analyzing multi-functional problems," *DS 101: Proceedings of NordDesign 2020, Lyngby, Denmark, 12th-14th August 2020,* pp. 1-12, 2020.

[98] J. Wu, W. Xiong, and W. Y. Wang, "Learning to learn and predict: A meta-learning approach for multi-label classification," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2020, pp.

4354-4364. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85084302600&partnerID=40&md5=e6b6168309e03309e678cc78c73e56a6. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85084302600&partnerID=40&md5=e6b6168309e03309e678cc78c73e56a6

[99] G. Tsoumakas, I. Katakis, and I. Vlahavas, "A review of multi-label classification methods," in *Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006)*, 2006: Citeseer, pp. 99-109.

[100] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on information theory,* vol. 26, no. 1, pp. 26-37, 1980.

[101] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research,* vol. 134, no. 1, pp. 19-67, 2005.

[102] Y. Yang, "A study of thresholding strategies for text categorization," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 137-145.

[103] D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of machine learning research,* vol. 5, no. Apr, pp. 361-397, 2004.

[104] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection," (in English), *Swarm and Evolutionary Computation,* Article vol. 64, Jul 2021, Art no. 100892, doi: 10.1016/j.swevo.2021.100892.

[105] M. S. Sorower, "A literature survey on algorithms for multi-label learning," *Oregon State University, Corvallis,* vol. 18, pp. 1-25, 2010.

[106] A. Aldrees and A. Chikh, "Comparative evaluation of four multi-label classification algorithms in classifying learning objects," *Computer Applications in Engineering Education,* vol. 24, no. 4, pp. 651-660, 2016.

[107] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications,* Review vol. 73, pp. 220-239, 2017, doi: 10.1016/j.eswa.2016.12.035.

[108] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-label Classification," in *Machine Learning and Knowledge Discovery in Databases*, (Lecture Notes in Computer Science: Springer Berlin Heidelberg, 2009, ch. Chapter 17, pp. 254-269.

[109] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM),* vol. 3, no. 3, pp. 1-13, 2007.

[110] M. W. Glier, "Machine-learning based classification of textual stimuli to promote ideation in bioinspired design," Ph.D., Texas A&M University, Ann Arbor, 3607471, 2013. [Online]. Available: https://proxy.library.mcgill.ca/login?url=https://www.proquest.com/dissertations-theses/machine-learning-based-classification-textual/docview/1491859844/se-2?accountid=12339
https://mcgill.on.worldcat.org/atoztitles/link?sid=ProQ:&issn=&volume=&issue=&title=Machine-learning+based+classification+of+textual+stimuli+to+promote+ideation+in+bioinspired+design&spage=&date=2013&atitle=&au=Glier%2C+Michael+W.&id=&isbn=978-1-303-65053-6

[111] S. Selim, M. M. Tantawi, H. A. Shedeed, and A. Badr, "A CSP\AM-BA-SVM Approach for Motor Imagery BCI System," (in English), *IEEE Access,* Article vol. 6, pp. 49192-49208, 2018, Art no. 8452948, doi: 10.1109/access.2018.2868178.

[112] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern recognition,* vol. 40, no. 7, pp. 2038-2048, 2007.

[113] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.

[114] D. Bogoradnikova, O. Makhnytkina, A. Matveev, A. Zakharova, and A. Akulov, "Multilingual Sentiment Analysis and Toxicity Detection for Text Messages in Russian," in *Conference of Open Innovation Association, FRUCT*, 2021, vol. 2021-May, pp. 55-64, doi: 10.23919/FRUCT52173.2021.9435584. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107393550&doi=10.23919%2fFRUCT52173.2021.9435584&partnerID=40&md5=10deacb490ea5babee2771a0b5ed5a37

https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=9435584&ref=

[115] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st annual meeting of the association for computational linguistics*, 2003, pp. 423-430.

[116] E. Brill, "A simple rule-based part of speech tagger," PENNSYLVANIA UNIV PHILADELPHIA DEPT OF COMPUTER AND INFORMATION SCIENCE, 1992.

[117] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A practical part-of-speech tagger," in *Third conference on applied natural language processing*, 1992, pp. 133-140.

[118] H. Schütze, "Part-of-speech induction from scratch," in *31st Annual Meeting of the Association for Computational Linguistics*, 1993, pp. 251-258.

[119] "The CKY Parsing Algorithm and PCFGs," ed. McGill University: Jackie CK Cheung.

[120] C. J. Van Rijsbergen, S. E. Robertson, and M. F. Porter, *New models in probabilistic information retrieval*. British Library Research and Development Department London, 1980.

[121] M. F. Porter, "An algorithm for suffix stripping," *Program,* 1980.

[122] M. F. Porter, "Snowball: A language for stemming algorithms," ed, 2001.

[123] C. D. Paice, "Another stemmer," *ACM SIGIR Forum,* vol. 24, no. 3, pp. 56-61, 1990, doi: 10.1145/101306.101310.

[124] J. Jablonski. "Natural Language Processing With Python's NLTK Package." https://realpython.com/nltk-nlp-python/#stemming (accessed Feb 26, 2022).

[125] N. C. Anoop Pillai, Mayuresh Pitale, P V Athira Chandran, "Text Summarization System for English Language," Department Of Computer Engineering PILLAI COLLEGE OF ENGINEERING, 2019-2020.

[126] S. a. N. Theme. "Natural Language Toolkit." https://www.nltk.org/ (accessed Feb 12, 2022).

[127] N. Tyagi. "What is Natural Language Toolkit (NLTK) in NLP?" https://www.analyticssteps.com/blogs/what-natural-language-toolkitnltk-nlp (accessed Feb 24, 2022).

[128] E. F. Unsvåg and B. Gambäck, "The effects of user features on twitter hate speech detection," in *Proceedings of the 2nd workshop on abusive language online (ALW2)*, 2018, pp. 75-85.

[129] J. Salminen, M. Hopf, S. A. Chowdhury, S.-g. Jung, H. Almerekhi, and B. J. Jansen, "Developing an online hate classifier for multiple social media platforms," (in English), *Human-centric Computing and Information Sciences,* Article vol. 10, no. 1, Jan 2 2020, Art no. 1, doi: 10.1186/s13673-019-0205-6.

[130] S.-W. Kim and J.-M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," (in English), *Human-centric Computing and Information Sciences,* vol. 9, no. 1, Aug 26 2019, doi: 10.1186/s13673-019-0192-7.

[131] B. Krawczyk and G. Schaefer, "An improved ensemble approach for imbalanced classification problems," in *2013 IEEE 8th international symposium on applied computational intelligence and informatics (SACI)*, 2013: IEEE, pp. 423-426.

[132] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences,* vol. 250, pp. 113-141, 2013/11/20/ 2013, doi: 10.1016/j.ins.2013.07.007.

[133] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, "Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions," (in English), *Computer Science Review,* Review vol. 38, Nov 2020, Art no. 100311, doi: 10.1016/j.cosrev.2020.100311.

[134] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proceedings of the International AAAI Conference on Web and Social Media*, 2011, vol. 5, no. 3, pp. 11-17.

[135] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, 2012: IEEE, pp. 71-80.

[136] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

[137] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "Intelligent Naïve Bayes-based approaches for Web proxy caching," *Knowledge-Based Systems,* vol. 31, pp. 162-175, 2012.

[138] E. Bisong, "Logistic Regression," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*: Springer, 2019, pp. 243-250.

[139] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 1980-1984.

[140] P. Burnap and M. L. Williams, "Us and them: identifying cyber hate on Twitter across multiple protected characteristics," *EPJ Data Sci,* vol. 5, no. 1, p. 11, 2016/03/23 2016, doi: 10.1140/epjds/s13688-016-0072-6.

[141] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88-93.

[142] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the International AAAI Conference on Web and Social Media*, 2017, vol. 11, no. 1, pp. 512-515.

[143] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1391-1399.

[144] J. Salminen *et al.*, "Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media," in *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[145] C. Shavers, R. Li, and G. Lebby, "An SVM-based approach to face detection," in *2006 Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory*, 2006: IEEE, pp. 362-366.

[146] R. Entezari-Maleki, A. Rezaei, and B. Minaei-Bidgoli, "Comparison of classification methods based on the type of attributes and sample size," *J. Convergence Inf. Technol.,* vol. 4, no. 3, pp. 94-102, 2009.

[147] S. Drazin and M. Montag, "Decision tree analysis using weka," *Machine Learning-Project II, University of Miami,* pp. 1-3, 2012.

[148] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Advances in Space Research,* vol. 41, no. 12, pp. 1955-1959, 2008.

[149] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006, doi: 10.1016/j.patrec.2005.10.010.

[150] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," in *Data Mining and Knowledge Discovery Handbook*: Springer US, 2009, ch. Chapter 34, pp. 667-685.

[151] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Pacific-Asia conference on knowledge discovery and data mining*, 2004: Springer, pp. 22-30.

[152] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence and loss minimization in multi-label classification," *Machine Learning,* vol. 88, no. 1-2, pp. 5-45, 2012/07/01 2012, doi: 10.1007/s10994-012-5285-8.

[153] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.