DNA SEQUENCE AUTOMATIC CLASSIFICATION—LEARN THE LIFE LANGUAGE USING ARTIFICIAL INTELLIGENCE

Josephine (Hsin) Liu[1,2], Phoebe (Yun) Liu[1,2], Joseph (Yu) Liu[1,2], Emily X. Ding[1], Robert J. Hou[1]

1 Vineyards AI Lab, Auckland, New Zealand
{emily. ding, robert.hou}@vineyardsailab.com
2 Rangitoto College, Auckland, New Zealand
{hsinliu0515, liucloud0515, josephyuliu}@gmail.com

ABSTRACT

This paper explores the applications of Artificial intelligence (AI) techniques for classifying Deoxyribonucleic Acid (DNA) sequences into their corresponding gene families. The paper focuses on presenting how to treat DNA sequences as a human language to be understood and classified. Specifically, we first transformed the DNA sequences into a more human-like format, then we employed Natural Language Processing (NLP) and Multi-layer perceptron (MLP) algorithms to complete sequence classification into 7 gene families. Our research drew DNA sequence data from three organisms, including humans, dogs, and chimpanzees. Finally, various experiments are conducted to prove the classification performance. In addition, to prove the generalization of this solution, we designed experiments that involved cross-domain testing. These experimental results display not only high accuracy and efficiency but also intriguing findings in life sciences.

KEYWORDS

DNA Sequences, Auto Recognition, Natural Language Processing(NLP), Multi-layer Perceptron (MLP)

1. INTRODUCTION

DNA stands for deoxyribonucleic acid, it is a macromolecule made up of nucleotides; phosphate sugar backbone, and nitrogenous bases A, T, C, and G, each in a different order and sequence. DNA can form genetic instructions to guide individual organism development and the functions of individual cells ensuring the survival and growth of the organism. It stores the required information for each cell and micro molecule to function, often described as the “blueprint” of the body. The different sequencing of DNA is the building block determining the structure of the DNA molecule. Segments of specific DNA sequences form genes, and these genes form gene families, genes are then responsible for gene expression or why our body functions the way it does. Scientists discovered that by classifying and identifying gene families from DNA sequences, diagnosis of early diseases can be made and predicted. After initial research, it was shown that DNA sequences are an important part of the biological field as the ability to understand DNA and classify it into families, can cause crucial breakthroughs in scientific and medical fields. Figure 1 displays some examples. Through DNA sequences, scientists can read, understand, and compare genetic information, potentially causing a breakthrough in biological studies and medical fields.[1-5].

David C. Wyld et al. (Eds): ICCSEA, NLAI, SCAI, CSIA, IBCOM, SEMIT, NECO, SPPR, MLDS -2023
pp. 155 - 169, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132413

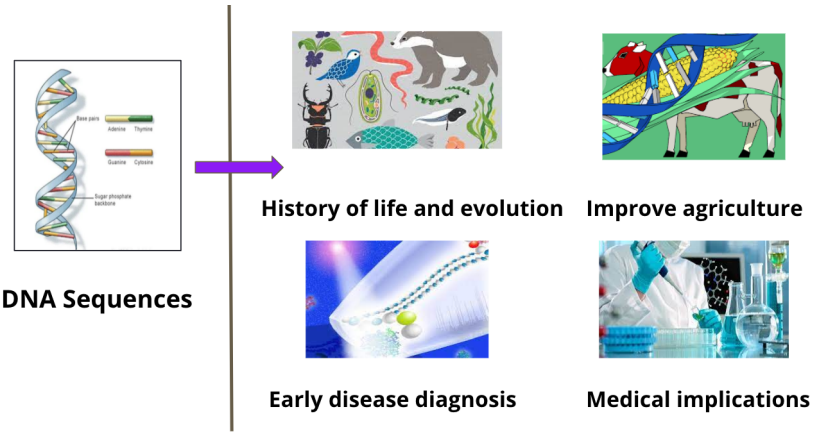

Figure 1. DNA sequences research fields and applications

However, much of this research is based on the premise of annotation of DNA sequences. The annotation is facilitated by classifying DNA sequences into families[6,7]. Predicting the DNA sequence into classes can provide insights into how an organism regulates and expresses genes. For example, if a specific DNA sequence is given, scientists can predict the possible relationship between the DNA sequence's function and its gene family. This would be worthwhile and crucial for genomic sequencing research, therefore, we chose to explore this topic further and predict the classes of DNA sequences.

It is challenging to recognize DNA sequences. Manual marking is time-consuming and error-prone, whereas it is evident that AI technology has the potential to make it far more efficient and accurate. Furthermore, AI is less costly in the long run because it does not require extra costs once the product or model is built, except for nominal costs such as maintenance.

How can AI technology be applied to the project? It is proven that DNA sequences are not only the language of life but also extremely similar to a human language, as it includes the specific "letters" and "phrases" needed to express and communicate information[8]. They are translated into the sequence of amino acids in a protein and can be understood and interpreted by other molecular machines within cells. Natural Language Processing (NLP) is an area of computer science that deals with methods to analyze, model, and understand human language. It performs well on many language-related tasks, such as language translation, sentiment analysis, speech recognition, and text summarization[9-11].

Thus the question arose: How can DNA sequences be treated as one of the human languages to understand and identify them?

In this research, a solution was developed to classify DNA sequences into their respective types with the help of AI technologies such as Natural Language Processing (NLP) and deep neural networks. The project motivation and research plan are shown in Figure 2.

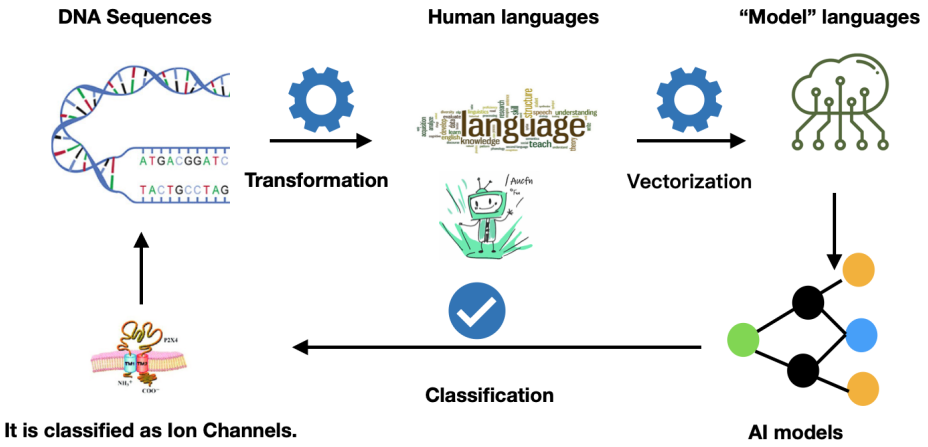

Figure 2. The motivation and research plan for our project

2. Our Method

From Figure 2, there are three main parts for automatically classifying DNA sequences. First, we transformed the DNA sequences into texts similar to human language. Then, we extracted features from the "texts" to transform them into vectors. This part is called vectorization. At this stage, DNA sequences have already been transformed into the language that a model can understand, and are ready to feed the classification model. Finally, we built and trained a deep neural networks (DNNs) classifier based on the extracted features.

2.1 Transforming the DNA sequences into texts similar to human language

DNA sequences are composed of the "letters" A, C, G, and T in a particular order. They, just like human language, communicate the secrets of life waiting for us to understand. Some patterns hidden in these codings decide the gene's functions and structures. It is challenging to use the original sequences for the classification directly. By Finding the letters in DNA sequences, we could treat them like a human language, such as English, and process them as texts composed of several words. The methods applied to NLP could be exploited to classify DNA sequences. The transformation of DNA sequences into a text is shown in Figure 3.

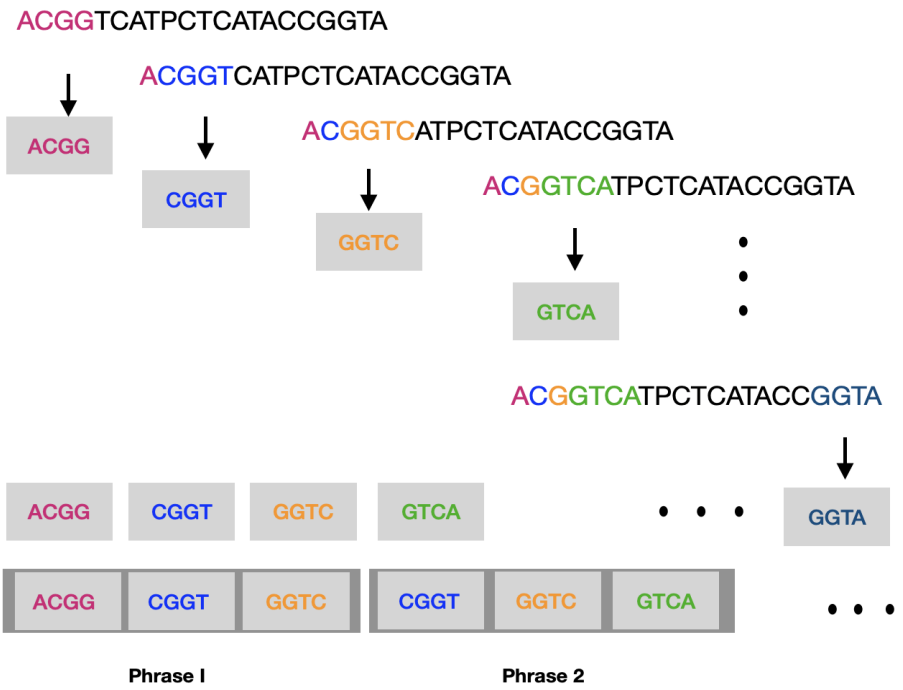

Figure 3. Transformation of DNA sequences into text

As we learned the set concept and how to define a set in maths class, we tried to describe the transformation process using two sets. Here, we defined the first set to describe the DNA sequences in a text with lots of words, as follows:

$$\boldsymbol{X}_{Text} = \left\{ \boldsymbol{X}_{words} | \boldsymbol{X}_{DNA-Seq}(i_n) : \boldsymbol{X}_{DNA-Seq}(i_n + l_{word}),\ i \leq length(\boldsymbol{X}_{DNA-Seq}) - l_{word} + 1 \right\} \quad (1)$$

Where $\boldsymbol{X}_{DNA-Seq}$ is one of the DNA sequences, i_n is the position of nucleotides and l_{word} is the word length. The $\boldsymbol{X}_{Text}$ is composed of words $\boldsymbol{X}_{words}$ with the same size in the **set**.

In addition, we group the words into several "phrases" without punctuation in the text. We also can call these “phrases” as “word bags”. They will be treated as independent targets in the next section and recorded statistically by the times (frequency) they appear in the text. Therefore, we defined the second set as follows:

$$\boldsymbol{X}_{text-new} = \left\{ \boldsymbol{X}_{phrase} | \boldsymbol{X}_{Text}(i_{word}) : \boldsymbol{X}_{Text}(i_{word} + l_{phrase}),\ i_{word} \leq length(\boldsymbol{X}_{Text}) - l_{phrase} + 1 \right\} \quad (2)$$

Where i_{word} is the ith word in the $\boldsymbol{X}_{Text}$ and l_{phrase} is the size of the phrase, different from the l_{word}, it can be a range and also can be fixed.

To this end, DNA sequences are transformed into natural languages, and it is then ready to enter the next section and represent the “text” in vectors where models can identify and process them.

2.2 Vectorization for the “texts”

2.2.1 Background for Vectorization

We have transformed DNA sequences into texts as described above. Then, in this section, we will convert the texts to vectors. Firstly, we should ask, why do we have to conduct this conversion? It could also be helpful to understand what vectors are in general. Finally, a natural question, how is text represented with vectors? We will explore and answer the questions below. The computer can not understand letters or words directly, so the text must be encoded into numeral numbers. Some popular methods for vectorization exist, such as Bag-of-Words(BoW), Word Embeddings, character-level

features, etc. In our project, we chose the classic and simple ones belonging to (BoW). The resulting vector contains the counts or frequencies of each "word bag" in the text. Then we introduced two classic methods we explored; Countvectorizer (CV), and Term Frequency-Inverse Document Frequency (TF-IDF). After the conversion, the numeral data can be used to represent the text, though different results between the two methods.

However, we still wonder why it is called a vector. We seek the definition of vector and try to understand it. It is an object that has both a magnitude and a direction, like an arrow, whose length can be seen as the magnitude of the vector (numeral numbers), and the arrow indicates the directions. We will have a further understanding during the research.

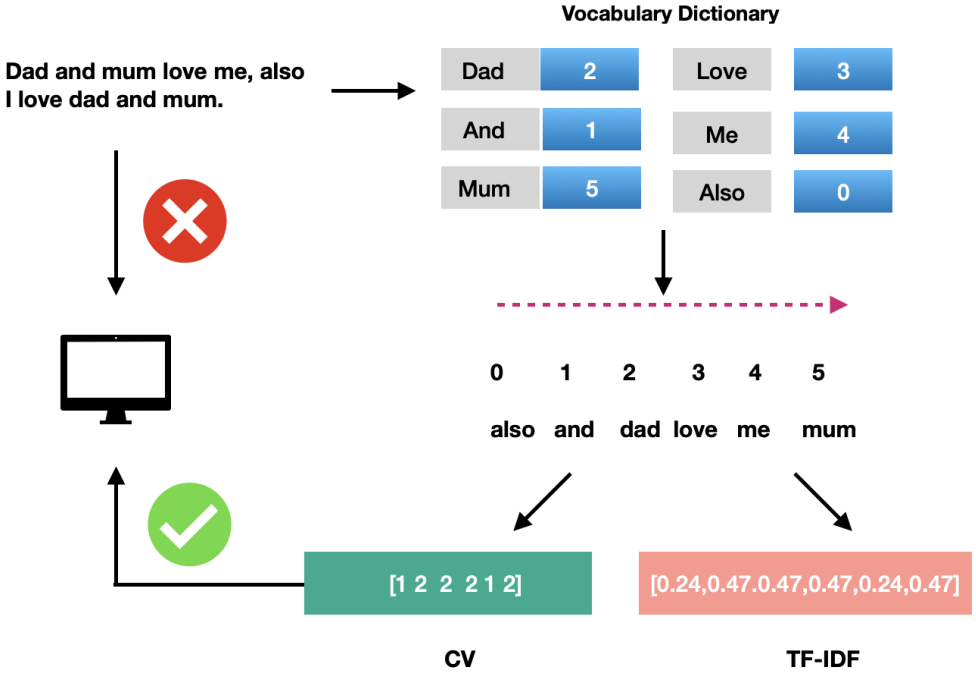

Figure 4. Vectorization for the text

Figure 4 demonstrates the vectorization for the CV and TF-IDF when we set the fixed size of "phrase" as 1. The text "Dad and mum love me also I love dad and mum" is represented by the numeral data. First, the vocabulary in the text is labeled in the dictionary, such as dad is "2" and love is "3". That means their features will be set in positions labeled 2 and 3. Until now, we can understand why we call it a vector. Their features obtained by CV or TF-IDF are the magnitude and arranged according to the vocabulary index for the directions. We can find that the word "I" is missing. Because as a stopping word (common word), such as "is", "are", etc., the stopping words in English are to be neglected. Next, we'll introduce the CV and TF-IDF, what they are, and how to compute the magnitude.

2.2.2 Countvectorizer and TF-IDF

Countvectorizer and TF-IDF are popular methods in NLP to extract features of texts. They are simple and effective in representing text as numerical data. Both measure the importance of words in a text document in different ways. Let's first introduce the Countvectorizer.

As shown in Figure 4, the count vectorizer builds a vector with the same dimension as the size of the vocabulary dictionary first. Then for each word, we calculate its frequency appearing in the text. The numeral or magnitude can be weighted as follows:

$$TF_d(X_{text-new}(d)) = \frac{N_d}{N_{text-new}} \tag{3}$$

Where $X_{text-new}(d)$ is the dth elements in the set $X_{text-new}$, N_d is the frequency of $X_{text-new}(d)$ appeared in the set $X_{text-new}$, and $N_{text-new}$ is the total number of elements in the set $X_{text-new}$.

Compared to the count vectorizer, TF-IDF not only cares for the frequency of the words appearing in this document but also considers how many times the same words appear in other documents. Therefore, there are two parts in TF-IDF described in the equation (4).

$$TF - IDF(X_{text-new}(d)) = TF_d(X_{text-new}(d)) \times IDF(X_{text-new}(d)) \quad (4)$$

Where $IDF(X_{text-new}(d))$ is called Inverse Document Frequency (IDF), and can be calculated as follows:

$$IDF(X_{text-new}(d)) = log\frac{l+1}{l_d+1} + 1 \quad (5)$$

Where l is the number of all texts, and l_d is the text number include the dth elements in set $X_{text-new}$. In equation(5), we could not understand the log function initially. We searched for the definition of this kind of function. But we overcame it and did it, luckily. First, the definition of log function is : if $a^x = b$, then $x = log_a b$. In our situation, we know that $\frac{l+1}{l_d+1} > 1$, smaller l_d, larger $\frac{l+1}{l_d+1}$. That means a larger $\frac{l+1}{l_d+1}$ leads to a larger $\log\frac{l+1}{l_d+1}$ too. So the fewer texts that include this element, the score of IDF of this element is larger. To avoid the zeros appearing in the numerator and denominator $\frac{l+1}{l_d+1}$, we add 1 to them.

2.3 Multi-layer Perceptron (MLP) as Automatic Classifier

In this section, we will build a supervised classification model as the vectorized features are ready.

2.3.1 A brief introduction to supervised learning

Supervised Learning (SL) is the main strategy for learning knowledge from data in AI. Usually, data for SL are paired with their labels and split into training and testing parts. Training data teaches the model how to classify while testing data is used to evaluate the model's performance[12-14]. A couple of analogies could be helpful to explain supervised learning as shown in Figure 5.

In a simplified learning process, babies learn to recognize objects and become mature as taught by their parents. In the very beginning, babies are taught what an object is (data and its paired label). After some repetition (training), babies become confident in recognizing the taught objects.

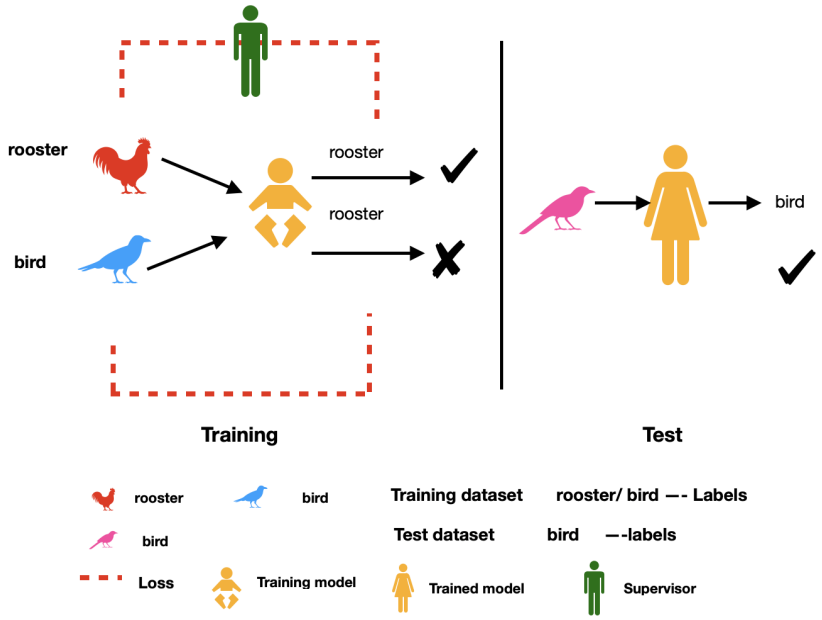

Figure 5. Analogies for Supervised Learning

Therefore, the most important thing is to build a model to describe the learning process from a "baby" to an "adult".

2.3.2 Artificial Neural Network—Multi-layer Perceptron (MLP)

In this part, we'll introduce a classic artificial neural network called multi-layer perceptron (MLP). We spent much time researching the mechanism of MLP and building them in Tensorflow. We tried our best to understand the background theory and explore the parameters for our classification task. Thanks to being a team, we discussed and explained them from various views through many similar analogies in our lives.

As we know, neural networks are inspired by and simplify the functioningof biological neurons. Biological neurons comprise dendrites, cell bodies, axons, and other parts. The dendrites are mainly used to receive signals from other neurons, while the axons output signals from these neurons. Synapses are the gap between the axon and other neurons' dendrites. Tens of thousands of neurons cooperate, enabling us to have advanced thinking and constantly learn new things. Usually, the neurons have two states: fire(active) and rest. When the stimulus received is higher than a certain threshold, it will be fired; otherwise, there is no nerve impulse. Figure 6 shows the biological neurons to a “note” in a neural network[15-17].

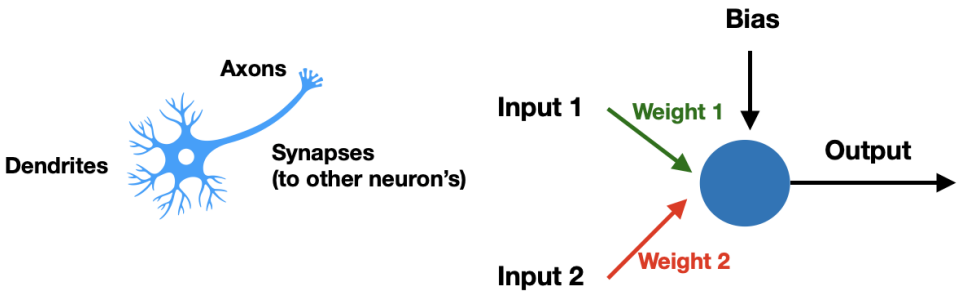

Figure 6. Biological neurons to a “node” in a neural network.

In Figure 6, the red line and green lines are just a synapse, that determines the weights, then the output for every node can be described as follows:

$$output = f(input1 * weight1 + input2 * weight2 + bias) \quad (6)$$

In equation(6), the $bias$ is a constant value, which f is called the activation function, which is a non-linear function. There are some different activation functions, such as Sigmoid, tanh, and ReLU. The Sigmoid function is $\sigma(x) = 1/(1 + exp(-x))$, and the output ranges from 0 to 1, and $tanh(x) = 2\sigma(2x) - 1$, and output is as $[-1, 1]$, the last one is ReLU as $f(x) = max(0, x)$, 0 is the threshold. The bias allows the network to shift the activation function to a different region, and better fit the training data.

Then all the nodes will be fully connected which means that each node in a layer is connected to all other nodes in the next layer. Each connection has a weight. An MLP consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. The structure is shown in Figure 7.

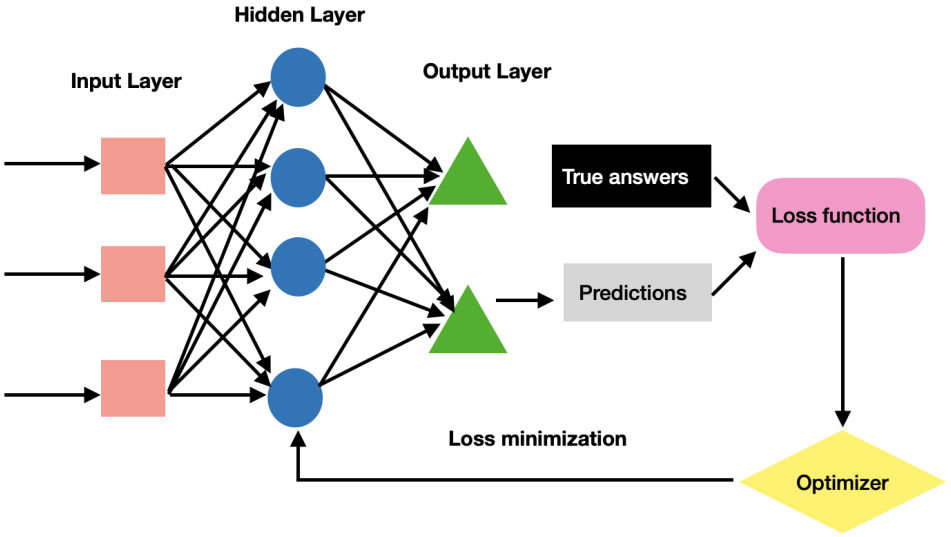

Figure 7. MLP network structure

Figure 7 displays a simple MLP network architecture. This type of network is called a feedforward network since it has no loops (i.e. the output of a neuron never connects to the input of a neuron in the last layer). MLP network includes three parts, the input, hidden, and output layers. How to optimize the weights and biases in the network to make the model smarter, relies on the loss function and the optimizer. The loss function is a method to measure the difference between the actual and predicted outputs. Learning aims to minimize the loss functions by adjusting the weights and biases. In our task, we'll choose the categorical cross-entropy loss function commonly used for multi-classification. The optimizer is an algorithm used to minimize the value of the loss function. Various optimizers are available, such as stochastic gradient descent (SGD), Adam, Adagrad, RMSprop, etc. Among them, generally, Adam is considered one of the best optimizers.

3. Experiments

3.1 Datasets

We downloaded the DNA sequences dataset from Kaggle, an online community where users can find and publish data sets and explore data science. This dataset includes more than 6500 DNA sequences of three organisms, among them, 4380 from humans, 820 from dogs, and 1682 from chimpanzees. They are annotated into 7 classes as shown below. Meanwhile, Figure 8 shows the class distributions of humans, dogs, and chimpanzees.

Table 1. DNA sequence types and the labels in the dataset

Gene family	Class Label
G protein-coupled receptors	0
Tyrosine kinase	1
Tyrosine phosphatase	2
Synthetase	3
Synthase	4
Ion channel	5
Transcription factor	6

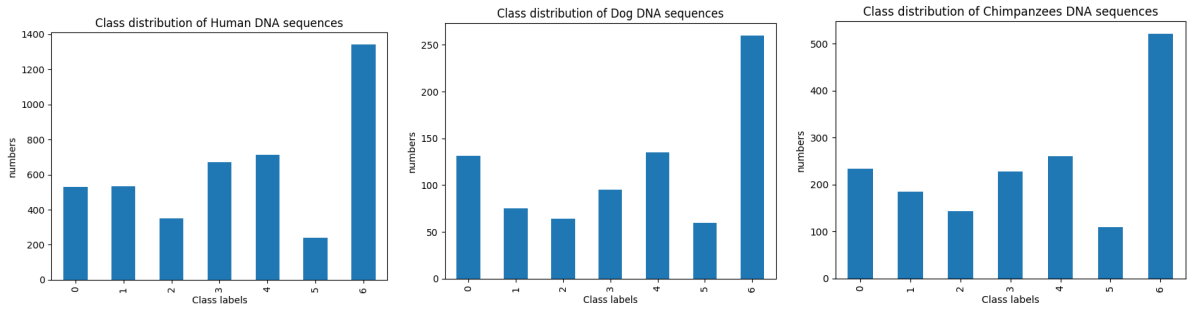

Figure 8 Classes distribution in human, dog, and chimpanzee data

Figure 8 shows the number of DNA sequences of each type (y-axis), human, dog, and chimpanzee, have been distributed against each class (x-axis). The graphs show that the most frequently distributed class is labeled number 6, the Transcription factor, whereas the least distributed class is 5, the Ion channel. Because the ratio of the distributions is roughly the same (despite the actual numbers being different), we can conclude that these three datasets have at least some level of similarity between them. Furthermore, we can see that the number of DNA sequences of the human datasets vastly surpasses that of the other two animals, such as class 6 in the y-axis (1,400) vastly surpasses chimpanzees at 500 and dogs at 250.

3.2 Evaluation of the different lengths of words and phrases

In this part, we evaluate the word and phrase parameters of our model. The parameters are the length of the word and phrase. In this experiment, human DNA sequences are used as our datasets. At the same time, we also record the running time in 100 Epochs and accuracy in these various situations. During these experiments, we fixed the other model settings, as we chose 2 hidden layers of MLP, and neuron numbers are 64 for the first and 128 for the second layer. The counter vectorizer method is used for vectorization, and the activation function ReLU is employed in the network.

Table 2. Performance of different lengths of "word" and" phrase"

phrase's length	Performance	$l_{word} = 2$	$l_{word} = 3$	$l_{word} = 4$
$l_{phrase} = 2$	Acc	0.7648	0.8299	0.8299
	Time	3m	4m	6m
	Dim	84	336	1247
$l_{phrase} = 3$	Acc	0.8550	0.9098	0.9269
	Time	3m	6m	11m
	Dim	336	1247	4459
$l_{phrase} = 4$	Acc	0.9155	0.9224	**0.9586**
	Time	5m	10m	**10m**
	Dim	1247	4469	16834
$l_{phrase} = 5$	Acc	0.9281	0.9521	**0.9532**
	Time	9m	35m	38m
	Dim	4469	16834	65447

As shown in the above table, the accuracy increases along with the gram and word lengths increase, albeit in an uneven manner, and at the cost of more time consumed. Thus, even though increasing the word length and n-grams will increase the accuracy, it is unrecommended to simply choose the one with the highest n-gram and word length, as it is impractical and time-consuming in real life. For example, although the model with a word length of 4 and n-gram of 4 has an accuracy of 0.9586, it also takes too much time. Through this, we can see that a certain balance between accuracy and time taken must be achieved to build a successful model. As such, the model with $l_{word} = 4$, $l_{phrase} = 4$ is recommended as it has a relatively high accuracy and an acceptable time.

3.3 Feature Extraction

In this section, we'll evaluate the performance of two vectorization methods. According to the experiments above, we chose $l_{word} = 4$ and $l_{phrase} = 4$. The network parameters are the same as in the previous experiments.

Table 3. Performance of different vectorization methods

Vectorization methods	Human	Dog	Chimpanzees
Countvectorizer	0.9586	0.7805	0.9139
TF-IDF	0.8505	0.6890	0.8497

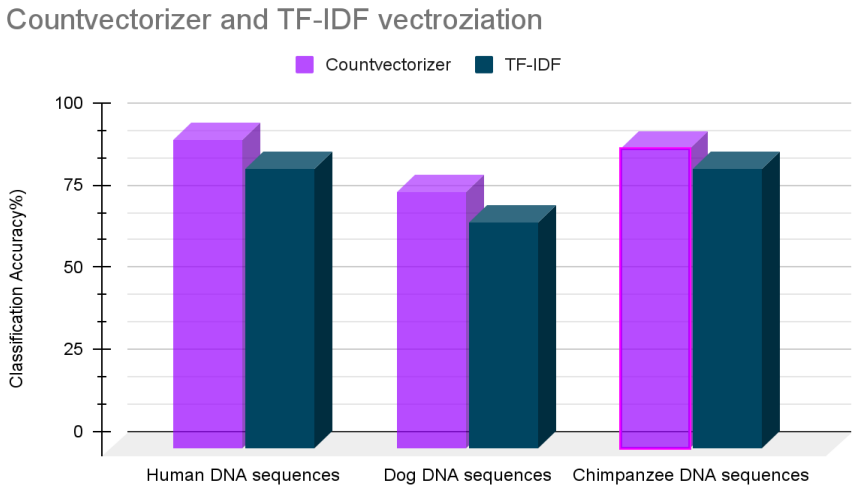

Figure 9 Model performance of Countvectorizer and TF-IDF

The experiment results show that a higher classification accuracy was achieved when we used a count vectorizer as the vectorization method for all three organisms. An interesting finding is that the accuracies were a bit lower with TF-IDF, though a better performance was expected from theory analysis and experiments in some NLP references. A possible reason is that the "words" in our project are not actual words, so it would not be very helpful to introduce IDF.

3.4 MLP performance

We explore the network structure and activation functions in this section. Also, based on the above experiment results, we choose the count vectorizer as the vectorization method, and $l_{word} = 4$ $l_{phrase} = 4$. The human DNA sequences are used as the dataset.

Table 4. Performance under the different network settings

Human dataset	Performance	Hidden Layers =1	Hidden Layers=2	Hidden_layers=3
Sigmoid	ACC	0.9555	**0.9578**	0.9372
	Time	20 min	20 min	19 min
ReLU	ACC	0.9372	**0.9586**	0.9532
	Time	9 min	10 min	11 min
Tanh	ACC	0.9475	0.9532	**0.9578**
	Time	10 min	10min	11min

The experiment results show a small gap in accuracy using different numbers of layers and activation functions. However, the running time is longer when the activation function Sigmoid was used. That means Sigmoid could need more computation. Certainly, as the hidden layers increase, more time is

required. Therefore, in this project, we seek a balance between the accuracy and cost of time. As a result, we choose ReLU as the activation function and two hidden layers in our neural networks.

3.5 Visualisation for the classification performance

To explore more, in this section, we will demonstrate the results with visualization. First, let's introduce the confusion matrix. We can show the classification results in a matrix form. It displays how many samples are classified correctly and incorrectly per class visually. In addition, we show how the accuracy changes with every epoch with a graph. The experiment results are shown in Figure 10.

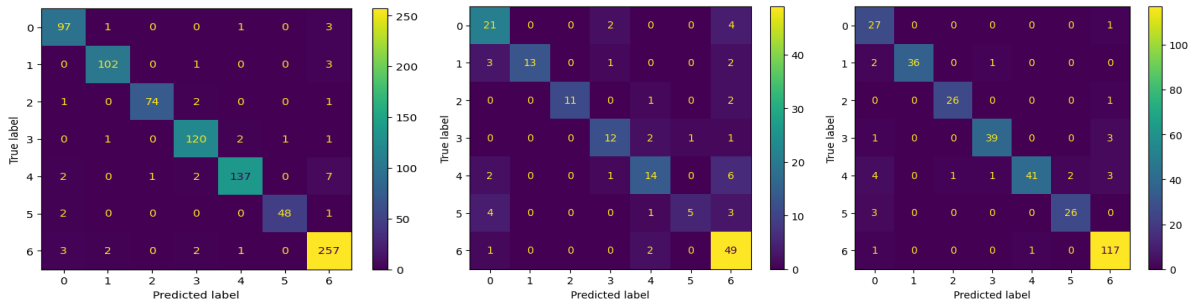

Figure 10 Confusion matrix of testing human, dog, and chimpanzee data

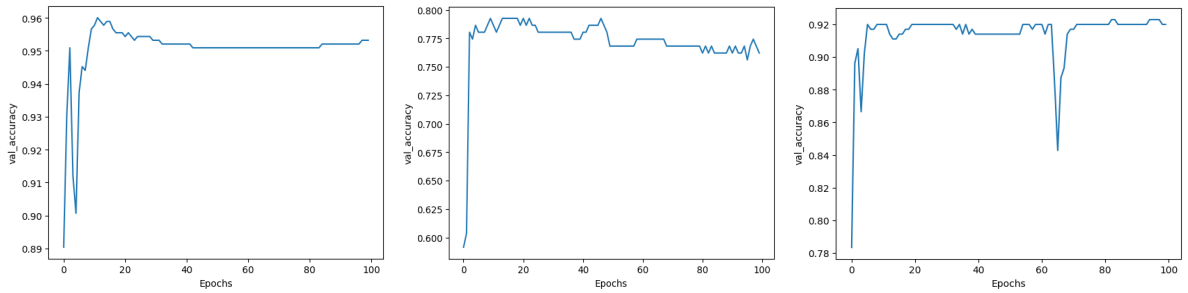

Figure 11 Training history for human, dog, and chimpanzee data

The results show that class '6' can be easily recognized by the model with high accuracy. While class '4' and '5' have a lower accuracy compared with other types, which means they are harder to classify by the model. Figure 11 displays the training history for epochs against val_accuracy, for human, dog, and chimpanzee data, the val_accurancy (accuracy in test data) tends to converge before the 100th epoch.

3.5 Generalization

We have conducted various evaluations to explore the model above. Furthermore, the generalization of AI models is crucial. Generalization is the ability to learn patterns from the training data that can be used for a new, unseen dataset. It is a key ability to build models with robust performance[18-19]. Figure 12 shows an example of model generalization.

Therefore, to evaluate the generalization, we designed our experiments to train the model on dataset A and test it on dataset B. We set the experiment as $l_{word} = 4$ and $l_{phrase} = 4$, Hidden layers =2, the activation function is ReLU, and Count-vectorizer is the vectorization method.

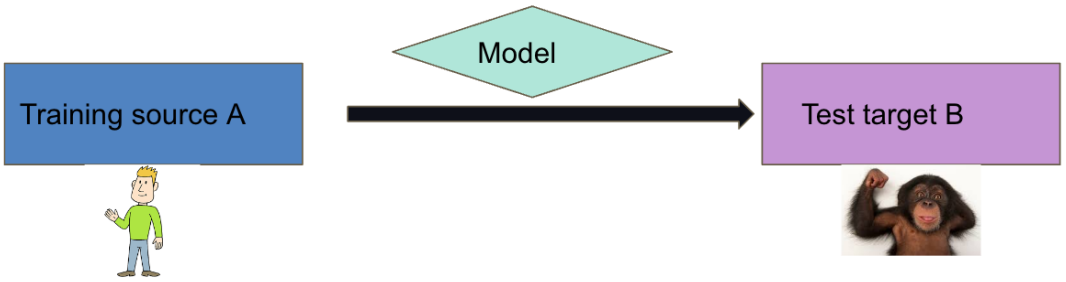

Figure.12 An example of model generalization

Table 5. Generalization performance of the model

Trained in Human	Tested in dog	Tested in Chimpanzees
0.9578	0.9085	0.9899
Trained in dog	Tested in human	Tested in Chimpanzees
0.7805	0.7103	0.8240
Trained in Chimpanzees	Tested in human	Tested in dog
0.9139	0.8758	0.8915

The experiments above show that the tests involving the human and the chimpanzee data are more accurate than those involving the dog's data. Some possible reasons could lead to the results. First, our model/project is data-driven, meaning the more complete and high-quality data we have, the more accurate the result will be. The data for the dog only consists of 820 strands of DNA sequences compared to the human, which has 4380 strands, and the chimpanzee's 1682 strands of DNA sequence. For the model trained by the human data, the test on chimpanzee data showed a higher accuracy than on the human test data, revealing the similarity between the chimpanzees' and human DNA. The distribution gap between human and chimpanzee DNA sequences is small.

Interestingly, the chimpanzees' accuracy is very high on the model trained by the human data. This could be because the human data set has many DNA strands, making its model better for generalization. In addition, we also found that the model trained by the dog data achieved higher accuracy for the chimpanzee's data than humans, while a similar result when trained by chimpanzees achieved higher accuracy for dog data than humans. We reckon that it is because dogs and chimpanzees are mammals, and some hidden patterns are similar. From the interesting findings, we could induce chimpanzees to have various potential connections with both humans and dogs.

CONCLUSIONS

In this research, we used Natural Language Processing (NLP) and neural networks to complete automatic classification for DNA sequences. We transformed DNA sequences to a human-like language and explored count vectorizer and TF-IDF as the vectorization methods. At last, we employed the classic neural network multi-layer perceptron as the classification model. We also developed a demo for users to try.

It is an exciting and thrilling moment for us to finish the project. We spend nearly one year on research, coding, testing, fine-tuning, and writing. During the project, we learned a lot. Firstly, we understand the pipeline of an AI project and adapt it to solve practical problems in our lives with high accuracy. To better understand, we are taught several analogies for AI knowledge, such as supervised

learning and multi-layer perceptron. Secondly, we deeply understand several concepts in our maths class, such as set, vector, matrix, and function. We then proceeded to apply them to our research, not just let them lie in our exam papers. Also, we try to understand some new and challenging knowledge in science. For example, we read the references for more background on this research's different DNA sequence types.

The first touch of AI is enjoyable, and we will explore more. For the next step, we will study different vectorization methods and AI models for classification. Also, exploring the generalization performance in the view of transfer learning could be interesting.

References

[1] D. S. T. Nicholl, *An Introduction to Genetic Engineering*. Cambridge University Press, pp2-10, (2023).

[2] D. S. T. Nicholl, *An Introduction to Genetic Engineering*. Cambridge University Press, p205-225, (2023).

[3] M. Smith, "DNA Sequence Analysis in Clinical Medicine, Proceeding Cautiously," *Frontiers in Molecular Biosciences*, vol. 4, (2017).

[4] J. M. Heather and B. Chain, "The sequence of sequencers: The history of sequencing DNA," *Genomics*, vol. 107, no. 1, pp. 1–8, (2016).

[5] H. Lehrach, "DNA sequencing methods in human genetics and disease research," *F1000Prime Reports*, vol. 5, (2013).

[6] L. C. Bailey Jr., S. Fischer, J. Schug, J. Crabtree, M. Gibson, and G. C. Overton, "GAIA: Framework Annotation of Genomic Sequence," *Genome Research*, vol. 8, no. 3, pp. 234–250, (1998).

[7] J. Zhang, Z.-M. Shang, J.-H. Cao, B. Fan, and S.-H. Zhao, "Manual annotation of the pig whole genomic sequence using Otter-lace software," *Hereditas (Beijing)*, vol. 34, no. 10, pp. 1339–1347, (2012).

[8] A. Wahab, H. Tayara, Z. Xuan, and K. T. Chong, "DNA sequences performs as natural language processing by exploiting deep learning algorithm for the identification of N4-methylcytosine," *Scientific Reports*, vol. 11, no. 1, (2021).

[9] S. Vajjala, B. Majumder, H. Surana, and A. Gupta, *Practical Natural Language Processing: A Pragmatic Approach to Processing and Analyzing Language Data*. O'Reilly Media, (2020).

[10] J. Pustejovsky and A. Stubbs, *Natural Language Annotation for Machine Learning*. "O'Reilly Media, Inc.," (2012).

[11] Y. Zhang and Z. Teng, *Natural Language Processing: A Machine Learning Perspective*. Cambridge University Press, (2021).

[12] S. S. Haykin, *Neural Networks and Learning Machines*. (2016).

[13] M. Coding, *Machine Learning with Python: A Step by Step Guide for Absolute Beginners to Program Artificial Intelligence with Python.* Charlie Creative Lab, (2020).

[14] S. Mendelson, *Advanced Lectures on Machine Learning: Machine Learning Summer School 2002, Canberra, Australia, February 11-22,* (2003).

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, (2016).

[16] S. Pattanayak, "Introduction to Deep-Learning Concepts and TensorFlow," in *Pro Deep Learning with TensorFlow*, Berkeley, CA: Apress, pp. 89–152, (2017).

[17] T. Hope, Y. S. Resheff, and I. Lieder, *Learning TensorFlow: A Guide to Building Deep Learning*

Systems. "O'Reilly Media, Inc.," (2017).

[18] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, *Transfer Learning*. Cambridge University Press, (2020).

[19] R. K. Sevakula and N. K. Verma, *Improving Classifier Generalization: Real-Time Machine Learning based Applications*. Springer Nature, (2022).

Authors

Josephine (Hsin) Liu

Josephine Liu is currently a year 11 student at Rangitoto College in Auckland, New Zealand. Outside of school, her passions include investigating AI and Taekwondo. She is interested in music, writing, law, history, and visual art.

Phoebe (Yun) Liu

Phoebe Liu is currently a year 11 student at Rangitoto College in Auckland, New Zealand. Some hobbies and interests include: playing musical instruments such as drums. She is vastly interested in different fields, such as art, computer science, biology, and Taekwondo.

Joseph (Yu) Liu

Joseph Liu is a student at Rangitoto College in Year 10 in Auckland, New Zealand. His hobbies and interests include music, spatial and product design, Taekwondo, and computer science.