Chapter 5: Variable Length Character N-Gram Embedding Of Protein Sequences for Secondary Structure Prediction

In this chapter, an effective n-gram based representation of protein primary sequences used for their secondary structure prediction. The proposed method for tokenization of primary sequences includes the variable length of character n-gram words. The most frequent n-gram words in the given dataset used for dense vector representation. Further, a stack of long short term memory networks used to find the complex contextual information to predict the secondary structure. The proposed model performance compared with other states of the art methods on publicly available data.

5.1 Introduction

Proteins are biological molecules that perform necessary functions for all human body processes, such as cell communications, enzymatic activity, and metabolism. The role of a protein is directly related to its three dimensional structure. Often, distinct amino acid sequences incorporate a similar structure, and the resulting structures show the same functionality only due to the similarity in their conformations [81]. Biologists have evolved four levels of the amino acid organization to understand protein structures. These are primary, secondary, tertiary, and quaternary structures [55]. The functions of protein are related to tertiary structure [56]. Protein secondary structure depends on the primary sequences, necessary for tertiary structure prediction [1][10]. Protein secondary structure prediction is one of the important tasks in proteomics [82]. Several computational methods used for protein secondary structure prediction,

which include rule-based [23] methods as well as machine learning-based methods [24] and evolutionary information [25]. Some feature extraction tools have been developed to generate features from primary sequences, such as Pse-in-One [11], BioSeq-Analysis [12], Pse-Analysis [13], and iFeature [14]. Pse-in-One is a webserver using 28 modes to generate feature vectors based on pseudo components. These feature vectors combined with machine learning methods for analyzing biological processes. The BioSeq-Analysis uses amino-acid compositions, autocorrelation, pseudo acid composition (PseAAC), profile-based features, and predicted structure features. BioSeq-Analysis2.0. uses Residue composition (One-hot, One-hot (6-bit), Binary (5-bit), Learn from alignments, Position-specific of two residues), physicochemical property, Structure composition, and Evolutionary information. BioSeq-Analysis2.0 incorporates two classification algorithms (Support Vector Machine (SVM), Random Forest (RF), and a sequence-labeling algorithm (Conditional Random Fields (CRF). Pse-Analysis automatically complete feature extraction, optimizing parameter, model training, cross-validation, and evaluation according to user-provided benchmark dataset for the query sequence. iFeature is a python tool to generate features for protein and peptide sequences. It combines feature clustering, selection, and dimensionality reduction algorithms with machine learning models for analysis and modeling.

Recently, deep neural network approaches have been proposed for the protein secondary structure prediction directly from the primary sequences. Convolution neural networks and recurrent neural networks have the learning capability of complex representation of sequences mostly used in protein secondary structure

prediction. A deep neural network with chained conditional neural network and next step conditioning achieve Q8 accuracy of 71.4% [62]. Protein secondary structure prediction using a deep convolution neural network with multiple layer shifts and snitch achieve Q8 accuracy of 68.4% [63]. Recurrent neural networks with profiles information perform protein secondary structure prediction with Q8 accuracy of 51.1% [64]. The bidirectional long short term memory network performs secondary structure prediction with Q8 accuracy of 67.4% [83]. A 2-D convolutional neural network combines with a two-way recurrent neural network achieves Q8 accuracy of 70.2% [27]. The supervised generative stochastic network[35] utilized both local dependency and long-range dependency for protein secondary structure prediction. DCRNN [65] combines cascaded convolution and recurrent neural networks for protein secondary structure prediction. DeepCNF [26] used the conditional random field for secondary structure prediction with 82.3% Q3 accuracy and 68.3% Q8 accuracy.

The proposed work's main contribution is: (1) Protein primary sequences are represented as the variable-length character n-gram to extract local contexts between amino acid residues. A vector containing counts of these variable-length character n-grams shows each protein sequence. These character n-grams vectors transform into a low dimensional deep embedding representation. (2) Stacked bidirectional Long Short Term Memory networks used for extracting the non-local context between amino-acid residues. (3) The proposed model evaluated three-class secondary structure predictions on publicly available datasets ss.txt, RS126, and CASP9. Experiments demonstrate that the combination of character n-gram embedding vector of primary

sequences and stacked Bidirectional Long Short Term Memory networks capture better features to improve the secondary structure prediction.

5.2 Protein Sequence Representation Method

Character n-grams word representations have a long history as features for several natural language processing applications. Some prior work has found benefit from using character-based compositional models that encode arbitrary character sequences into vectors. Examples include recurrent neural networks (RNNs) and convolutional neural networks (CNNs) on character sequences, showing improvements for several NLP tasks [67–69]. By sharing sub word information across words, character models have the potential to better represent rare words and morphological variants.

Among all the measures, extracting frequencies of character n-grams is a more effective approach that is able to capture nuances of higher level and tolerate the noises such as grammatical errors or misuse of punctuations. Moreover, the procedure of extracting n-grams is language-independent and requires no special tools, especially for oriental languages, where the tokenization procedure is not trivial [6]. The character n-gram approach has been proven to be quite useful to quantify the writing style [7]. Keselj [9] and Stamatatos [10] reported very good results using character n-gram information. Moreover, one of the best performing algorithms in an authorship attribution competition organized in 2004 was also based on a character n-gram representation [11]. An early study in this category of character-based models is to create corpus based fixed-length distributed semantic representations for text. [79]. To train k-gram embeddings, the top character k-grams are extracted from a corpus along with their co-occurrence counts. Then, singular value decomposition (SVD) is

used to create low dimensional k-gram embeddings given their co-occurrence matrix. To apply them to a piece of text, the k-grams of the text are extracted, and their corresponding embeddings are summed. The study evaluates the k-gram embeddings in the context of word sense disambiguation. A more recent study trains character n-gram embeddings in an end-to-end fashion with a neural network [96]. They are evaluated on word similarity, sentence similarity, and part-of-speech tagging. Training character n-gram embeddings have also been proposed for biological sequences [3,4] for various bioinformatics tasks.

An important issue of the character n-gram approach is the definition of n. A large value of n would better capture lexical and contextual information, but it would also capture thematic information and increase the dimensionality of the representation, while a small n would not be adequate for representing the contextual information. The drawbacks of defining a fixed value for n can be avoided by extracting variable-length n-grams [12]. This chapter is aimed at introducing a variable-length character n-gram approach for protein secondary structure prediction. Finally, experiments are conducted to evaluate the effectiveness of the proposed approach.

5.3 Dataset Description

The proposed architecture was trained on a large dataset ss.txt [42]. The training set contains 174,372 protein sequences. In the testing phase of the proposed architecture two datasets namely CASP9 [75] and RS126[44] were used. The CASP9 [75] dataset have 203 non-redundant proteins, derived from the 2016 CASP meeting. The RS126 [44] dataset have 126 non-homologous globular sequences. In RS126[44] average length of sequences are 185, and sequence similarity is below 31%. The maximum

similarity between the two sequences in the training and test sets is 30% to ensure the model performance. All the eight types of the secondary structure of Define Secondary Structure of Proteins (DSSP) [45] and their three-class categorization. The DSSP eight types of secondary structure classes, such as 310-helix (G), alpha-helix (H), and pi-helix (I), beta-bridge (B), and beta-strand (E), high curvature loop(S), beta-turn (T), and coil(C). These eight classes simplified into three significant types' alpha-helix (H), Beta-strand (E), and coil regions (C) by using standard conversion.

5.4 Methodology

5.4.1 Model Details

The proposed model is described in figure 5.1 by representing primary sequences as variable length character n-gram words vectors. These vectors are represented as embedded vectors in a low dimensional space. These vectors are having local contextual information pass to bidirectional long short term memory network. The bidirectional long short term memory captures the non-local contextual information for protein secondary structure prediction. The model learns embeddings of variable length character n-gram words, which produce primary sequences embeddings when all the variable length character n-gram words are summed. Finally, the softmax function is categorized the three class of secondary structure.

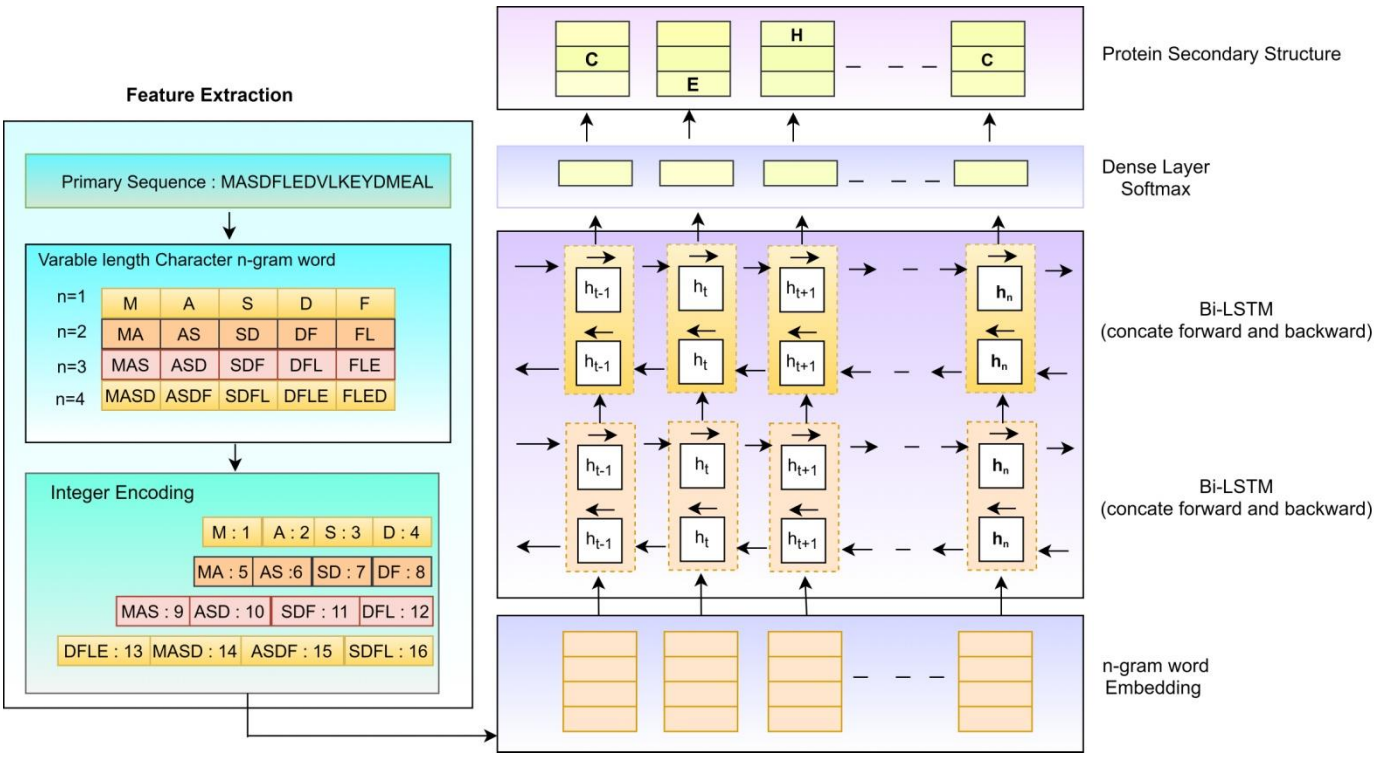

Figure 5.1: Proposed Model for Protein Secondary Structure Prediction

5.4.2 Character N-Gram

We propose a segmentation approach for dividing protein sequences into frequent variable-length sub-sequences. The proposed model extracts a set of character n-grams with variable length instead of fixed length. The collection of n-gram words is too large in preprocessing to eliminate insignificant n-gram words. To select the valuable n-gram word frequency is an essential criterion. Therefore, all the n-grams rank based on their frequency measure and then removes the insignificant ones. Each primary sequence is a bag of character n-gram. The proposed model extracts all the n-grams for n greater or equal to 1 and smaller or equal to 4. The set of n-grams mapped to unique integers in one to N. The model represents sequence A by $G_A \subset \{1. . . G\}$ the set of n-grams appearing in sequence A. We associate a vector

representation z_g to each n-gram g. We represent a sequence by appending the vector representations of its n-grams.

5.4.3 Deep Network: Bidirectional LSTM

Long Short Term Memory uses memory to learn long-range interactions between residues in primary sequences. The BRNN [1] utilizes the past and future residue information to predict the current amino acid secondary structure. In this chapter, a bidirectional long short term memory (Bi-LSTM) is used to extract in-depth features from primary sequences to predict the secondary structure of a protein.

$$h_{forward} = f\left(w_{xh} \cdot h_{t-1} + w_{hh} \cdot x_t\right) \quad (5.1)$$

$$h_{backward} = f\left(w_{xh} \cdot h_{t+1} + w_{hh} \cdot x_t\right) \quad (5.2)$$

$$y = f\left(w_{hy} \times \left[h_{forward}\ ,\ h_{backward}\right]\right) \quad (5.3)$$

Where W_{xh}, W_{hh}, and W_{hy} are weight matrices. x_t is current input and h_{t-1} is previous state.

In the previous section described that the variable-length character n-gram embedding capture local contextual information. After obtaining the primary sequences' deep representation using character n-gram, the proposed architecture uses bidirectional long short-term memory (Bi-LSTM) to capture non-local interaction. The model uses a stacked bidirectional long short-term memory network. The first layer receives input as embedding vectors from the embedding layer, and the second layer bidirectional long short term memory network produces an output vector.

5.4.4 Fully Connected Layer

The model flattened the output vectors obtained from the stacked Bi-LSTM layer to pass in a dense layer. The softmax function categorizes four classes alpha-helix, Beta

Strand, coil, and no class. Softmax having a total sum of probability for each class is one.

$$\sigma(z) = \frac{e^{z}i}{\sum_{i=1}^{k} e^{z}i} \tag{5.4}$$

$$\text{for } i = 1,.....,k \text{ and } z = z_1.....z_k$$

Where z_i is the element of vector z representing the probability of each class, k is equal to four showing the classes alpha-helix, beta-sheet, coil and no-class in vector z.

5.5 Experimental Details

Python version 3.6.7 implement the proposed work. At the front end, Keras[31] and Tensorflow [32] version 1.9 used as a backend. Keras is an open-source API implemented using the Python for machine learning programming. Tensorflow has excellent ability in numerical computation. The proposed model chooses a dropout of 0.1 adopted to avoid overfitting. RmsProp [33] optimization applied with a batch size of 64. The categorical cross-entropy used to update the weights and bias, all the hyperparameters used in the proposed model listed in Table 5.1.

Table 5.1: Hyper-parameters Value

Hyper-parameters	**Value**
Cell	64
Dropout	0.1
Recurrent dropout	0.1
Batch size	64
Optimization method	RmsProp
Loss	Categorical_crossentropy
Output activation	Softmax

5.6 Results and Discussion

In this work, inspired by unsupervised word segmentation in natural language processing, we propose a general-purpose segmentation of protein sequences in frequent variable-length sub-sequences, for machine learning tasks. This segmentation is trained once over large protein sequences (Swiss-Prot) and then is applied to a given set of sequences. In this chapter, we use this representation for developing a protein sequence embedding.

The dataset divided into training and testing in a ratio of 70 to 30 using a statistical sampling method. Data are split between training and testing to produce the generalized model. The secondary structure prediction model using Q3 accuracy for three classes of secondary structure. The model trained on ss.txt with a Q3 value of 92.57%. The plot of Q3 with epochs for dataset ss.txt is in figure 5.3.The plot of loss for training and testing set for ss.txt is in Figure 5.2. The x-axis represents the number of epochs, and the y-axis represents the loss of the proposed model. The model performance is comparable for both pieces of the training as well testing set. Initially, there is a difference between training and testing value, but as the number of epoch increases, the difference decreases. The minimum difference value between training loss and testing loss shows that the proposed architecture with Bi-LSTM is performing well.

Table 5.2: Performance Comparison of Bi-RNN, Bi-GRU and Bi-LSTM

RNN Model	**F-measure**	**Sensitivity**	**Specificity**	**Accuracy**
Bi-LSTM	0.9182	0.9134	0.9746	91.81%
Bi-GRU	0.8471	0.8337	0.9595	86.40%
Bi-RNN	0.7840	0.7778	0.9454	80.45%

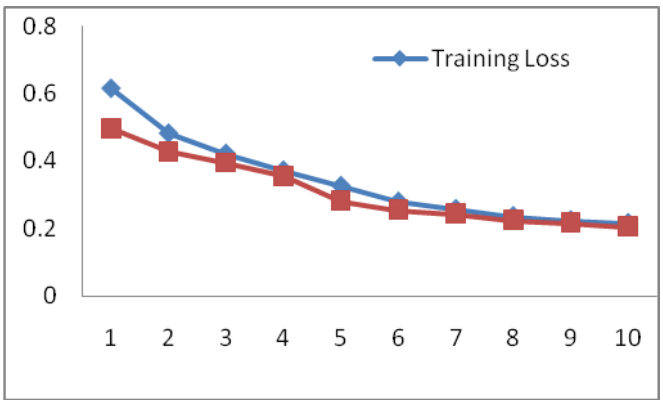

Figure 5.2: Training and Testing Loss over Epochs for ss.txt

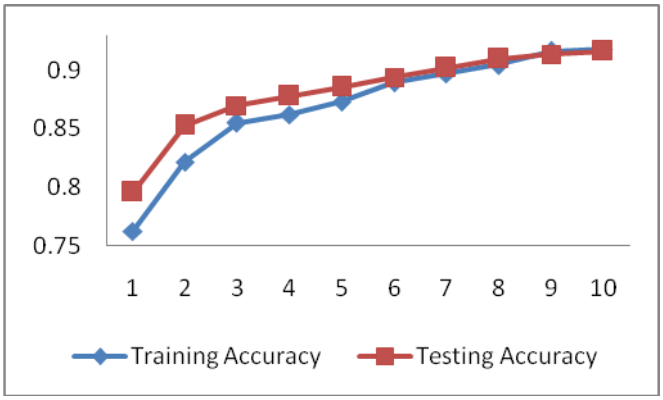

Figure 5.3: Training and Testing Accuracy over Epochs for ss.txt

The proposed architecture implemented and evaluated the three-bidirectional recurrent neural networks such as Bi-RNN, Bi-GRU, and Bi-LSTM for protein secondary structure prediction. Further, the performance metrics used to compare three recurrent neural networks are F-measure, Sensitivity, Specificity, and Accuracy. For training purposes, ss.txt data used, whereas for testing purposes CASP9 and RS126 were used. The results in Table 5.2 show that Bi-LSTM is giving better results in comparison to Bi-RNN, Bi-GRU. Bi-LSTM model has higher sensitivity when compared with Bi-RNN and Bi-GRU.

The F-measure score is the harmonic mean between precision and recall. The model shows a high F-measure value of more than 90%. So the precision and recall are

equally better for secondary structure classification. The sensitivity and specificity greater than 90% imply that secondary structure prediction correctly identifies each class. The prediction accuracy is 91.81%.

The proposed architecture performance for unigram, bigram, and trigram, shown in Table 5.3. The model gives a better result for bigram compared with unigram for training and testing both. Further increasing the value of n-gram, the Q3 value increase for training but decreased for testing. The model also evaluated for 4-gram and 5-gram, showing the same result as a trigram. The better result for bigram shows that the primary sequences have repeated combinations of two amino acids, which better decide their secondary structure.

Table 5.3: Q3 accuracy for different value of n-gram

n-gram size	**Q3(%)**
Unigram	82.74
Bigram	88.45
Trigram	81.96

The model computed secondary structure with a variable-length character n-gram, which consisted of unigram, bigram trigram, and four-gram token representations (i.e., 1 − 4 grams). The result of experiments with a different combination of n-gram value for protein secondary structure prediction is in Table 5.4.

Table 5.4: Q3 accuracy for different combination of n-gram

n-gram combination	**Q3(%)**
Unigram + bigram	**91.86**
Unigram + bigram+ trigram	**92.90**
Unigram + bigram+ trigram + four-gram	**93.87**

The proposed model has tested for two public datasets CASP9 and RS126. The results on both datasets show the effectiveness of the model compared to other state-of-the-art methods. The methods based on deep neural networks for secondary structure prediction methods, such as (SPIDER3)[2], JPred4[3], RaptorX[4], MUFOLD-SS[5], and Ensemble LSTM[6] used to compare the model performance on CASP9. As shown in Table 5.5, the Q3 value of our approach is 89.66%, which is better than SPIDER3, JPred4, MUFOLD-SS, and RaptorX, with Q3 81.9%, 79.3%, 84.82%, and 81.0%, respectively.

The model evaluation result for RS126 compared with the secondary structure prediction methods such as Psipred[7], PHD[8], Sspro[9], Jpred4[3], and MUFOLD-SS[5]. The RS126 test data sets secondary structure prediction results of the proposed architecture reported in Table 5.6.

Table 5.5: Q3 value comparison with other Methods on dataset CASP9.

Methods	**Q3 (%)**
Proposed Model	**89.66**
Ensemble LSTM[79]	83.3
Spider3[4]	81.9
Jpred4[14]	79.3
MUFOLD-SS[78]	84.82
RaptorX[77]	81.0

Table 5.6: Q3 value comparison with other Methods on dataset RS126.

Methods	**Q3 (%)**
Proposed Model	**86.49%**
Psipred[50]	81.01
PHD[53]	76.92
Sspro[36]	77.01
Jpred4[14]	73.82
MUFOLD-SS[78]	74.21

5.7 Conclusion

The segmentation of protein primary sequences in variable length character n-gram words, easily capture the local contextual information of amino acid residues. Further, these variable length n-gram words input to bidirectional long short term memory network, which easily capture complex non-local contextual information. The combination of variable length segmentation and long short term memory network better predict the protein secondary structure form their primary sequences.