Shahjalal University of Science and Technology
Department of Computer Science and Engineering

CSE 452

A Probabilistic and Neural Network based Approach for Bengali POS Tagger

Avi Mallick

Reg. No.: 2015331026

4^{th} year, 2^{nd} Semester

Department of Computer Science and Engineering

Supervisor

Enamul Hassan

Assistant Professor

Department of Computer Science and Engineering

27^{th} February, 2020

A Probabilistic and Neural Network based Approach for Bengali POS Tagger

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

Avi Mallick

Reg. No.: 2015331026

4^{th} year, 2^{nd} Semester

Department of Computer Science and Engineering

Supervisor

Enamul Hassan

Assistant Professor

Department of Computer Science and Engineering

27th February, 2020

Recommendation Letter from Thesis Supervisor

The thesis entitled *A Probabilistic and Neural Network based Approach for Bengali POS Tagger* submitted by the students

1. Avi Mallick

is under my supervision. I, hereby, agree that the thesis can be submitted for examination.

Supervisor

Enamul Hassan

Assistant Professor

Department of Computer Science and Engineering

Date: 27th February, 2020

Certificate of Acceptance of the Thesis

The thesis entitled *A Probabilistic and Neural Network based Approach for Bengali POS Tagger* submitted by the students

1. Avi Mallick

on 27th February, 2020 is, hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

Head of the Dept.	Chairman, Exam. Committee	Supervisor
Mohammad Abdullah Al Mumin	Dr Mohammad Reza Selim	Enamul Hassan
Professor	Professor	Assistant Professor
Department of Computer Science and Engineering	Department of Computer Science and Engineering	Department of Computer Science and Engineering

Abstract

POS Tagging is the process of tagging the words in a document or text according to their corresponding Parts Of Speech based on their context and definition. Despite being many types of research on POS tagging conducted, There is no remarkable or pragmatic POS Tagger in Bengali. After exploring the facts and deficiency of current researches, we have proceeded with a probabilistic and neural network-based method. The method is designed to be top-notch with its proposed structure by exploiting the fusion of Dynamic Programming techniques and Prefix Tree. The method introduces the probability to be used as a feature for neural networks and getting the best combination of the tag from all generated combinations. We have worked with 4953 lines, 11,528 unique words out of total 47,594 words and a dictionary of 1,12,382 words. We have worked with the Sequential and Bidirectional LSTM model and got an accuracy of 90% after adding Higher Class Difference and Higher Probability First Technique.

Keywords: POS-Tagger, Neural Network, Probabilistic, Edit Distance, Prefix Tree, Trie, Dynamic Programming, Parts Of Speech, Bengali Natural Language Processing, sequence labeling, Bidirectional LSTM.

Acknowledgements

I would like to thank the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh, for supporting this research.

I wish to express my sincere appreciation to my honorable supervisor Enamul Hassan, who convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Without his persistent help, the goal of this project would not have been realized.

I am very thankful to Md. Saiful Islam for his worthy supports and directions. Without his supports and directions this work couldn't be done.

The contribution of pipilika.com for the dataset is truly appreciated. Without their support, this project could not have reached its goal.

I wish to acknowledge the initiative taken by my seniors, Md. Nazim Uddin and Moudud Khan Shahriar, on this research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

From the current research trends, it has become clear that extracting linguistic information from natural language or text is an powerful method to create a robust and accurate natural language processing system. Parts of Speech(POS) is one of the basic linguistic information that we can extract from texts as a useful feature. Once POS Tagging was done by hand but now is done in the context of computer linguistics. POS Tagging is the process of tagging the words in a document or text according to their corresponding Parts Of Speech based on their context and definition. It is not an easy or straightforward task as the tag of a word varies based on their language, sentence, etc. In the Bengali language, it is harder to define the POS Tag of a sentence as Bengali follows more complex rules. Researchers Have been working on POS Tagging for a long time and in many languages, they have been able to bring a good and accurate POS Tagger. POS Taggers with high accuracy have been developed English, German, French or other European languages[5][6][7][8]. But in Bengali, there had not been any such significant success. Parts of speech tagging in Bengali is harder than just having sentences of words and giving them a tag. Due to complex rules in Bengali, it is hard to design good rule-based tagger for Bengali. POS tagging techniques are mainly of two types, rule-based and stochastics. Various statistical models such as ME, HMM, CRF, SVM has been applied to date but neither of them shows that much good accuracy in tagging. Besides most of them depends on tools like stemmer. That means the accuracy of these methods depends on the accuracy of the stemmers. It is also hard to create a correct dataset because Bengali

is such a complex language that tagging by a human can not always ensure a correct tag. Due to diverse expression in the Bengali language in different platforms, it is also possible for the same sentence to have two different tagset. This also makes this task a bit more challenging.

1.2 Motivation

In Natural Language Processing, most models are based on the bag of words or tokens. So having some features extracted or finding some pattern in the words or sentences always give the research an advantage. Parts of Speech is such kind of features that we may not need or use directly or practically but it can be used as an important feature in several NLP researches.n Natural Language Processing hierarchy[1], at the bottom, is a sentence and word segmentation . POS is at the top of it, so having pos tagged data can help in building parse trees that can be used for several NLP systems.

1.2.1 Word Sense Disambiguation

A natural language text can have multiple meanings or senses according to their context. The task of WSD or Word Sense Disambiguation is to sense the correct meaning of natural language text based on their context [9] [10].POS Tagging plays an important role in Word Sense Disambiguation.

Word	Sense 1 (POS)	Sense 2 (POS)
জাল	Forge (Verb/Adjective)	Net(Noun)
রহিম	Name (Noun)	Kind (Adjective)
গোলা	Mix (Verb)	Shell(Noun)

Figure 1.1: Word Sense Disambiguation example

So from the examples in figure 1.1, we can see that same word can have different senses and they can be distinguished by knowing their Parts of Speech.

[1]https://en.wikipedia.org/wiki/Natural$_{l}anguage_{p}rocessingMajor_{t}asks_{i}n_{N}LP$

1.2.2 Text to Speech Conversion

In Bengali pronunciation of the different words can be the same. They can only be distinguished based on their context or meaning. Parts of speech of a word can help in this case. Based on the parts of speech of a word in a text we can get the real sense of a word in the text and then it can be used to convert into speech.

1.2.3 Named Entity Recognition

Named Entity Recognition (NER) is a Information Retrieval subtask that finds and categorizes the named entity in a sentence in some predefined classes.

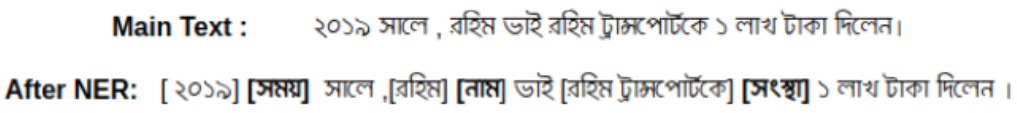


Figure 1.2: Named Entity Recognition example

The figure 1.2 shows a Bengali NER example.
So in order to find the NER, we need to find the nouns in the sentence, only then we can classify them. That is why we need a good POS Tagger.

1.2.4 Sentimental Analysis

In Sentimental Analysis, we basically try to find what the sentence refers to, positive or negative sentiment. In this sector, POS is one of the most important features. POS can help to define the positive or negative sense of a text. Yes it can be possible to do this without POS tagging , but in many cases system will face problems that can be solved with tagged POS.[2]

[2]https://www.quora.com/What-is-the-use-and-need-of-part-of-speech-POS-tagging-in-sentimental-analysis

1.2.5 Question Answering

Question Answering is an NLP Discipline in the field of information retrieval[3]. In Question Answering System POS tagging is used as an important feature. POS can be a piece of useful information for a QA system. It helps understand the context of questions and to generate the answer.

1.3 Overview

In this report, we have discussed some of the previous works conducted on this topic in the Literature Review. We have tried to specify the problems that are still on the system.Then we have discussed our proposed method. In the Methodology chapter, we have disclosed our work methods. In the Dataset Chapter, We have given a brief note on our dataset.

[3]https://en.wikipedia.org/wiki/Question$_{a}nswering$

Chapter 2

Background Study

2.1 Literature Review

As we have discussed earlier, there had been works in English,German and other european languages that show great accuracy in POS Tagging. Researches on Bengali POS Tagging has also been made for a while. But unfortunately, there had not been any significant or practical success in Bengali POS Tagging till now.

- Dandapat, Sarkar, and Basu have used a hybrid model of supervised and unsupervised learning method alongside Hidden Markov Model[11]. They have got 95% accuracy, but that is on a small tagged corpus. We can not be sure how it will work in a practical huge corpus.

- Ekbal and Bandyopadhyay have used a Hidden Markov Model to tag Bengali dataset and got an accuracy of 91.6% [3]

- Ekbal, Haque, and Bandyopadhyay have presented a Maximum Entropy-based POS Tagger that has demonstrated an accuracy of 88.2% for a test set of 20K word forms [4]. This Outperforms their HMM-based Model.

- Ekbal, Haque, and Bandyopadhyay have trained and tested the 72,341 and 20K wordforms in their CRF based POS tagger[12]. Their proposed CRF based POS tagger shows an accuracy of 90.3%. Their CRF Framework Works well in prefix and suffix of length up to three, NE information of the current and the previous words, POS information of the previous word, digit features, symbol features, and the various gazetteer lists.

- Again Ekbal and Bandyopadhyay have modeled an Support Vector Machine framework that has been trained and tested with the 72,341 and 20 K wordforms. This method shows the effectiveness of 86.84%[13].

- Sarkar and Gayen have implemented a supervised Bengali trigram POS Tagger from scratch using a statistical machine learning technique that uses the second-order Hidden Markov Model (HMM)[14].

- Mukherjee and Mandal have worked on an automatic POS tagging for Bengali that uses Global Linear Model (GLM) which learns to represent the whole sentence through a feature vector called Global feature[1].Their experimental accuracy was 93.12 %.

- Chakrabarti has proposed a rule based POS Tagger with 4 levels of layer tagging that also handles multiverb expressions.[15]

- Uddin, Khan, Islam, and Jannat have proposed a Neural Network-based approach for Bengali POS Tagger[2]. They have also proposed to use a dynamic programming technique to reduce time complexity.

2.2 Works Comparisons

While Studying the background we have collected some comparisons of the researchers work.

- Mukherjee and Mandal have made a comparison of how various models work in two different datasets, their own dataset, and the NLTK dataset[1].

Model	GLM Dataset Accuracy	Nltk Dataset Accuracy
ME	89.58 %	73.19 %
SVM	91.19 %	82.44 %
CRF	91.47 %	82.35 %
HMM	84.22 %	65.37 %
GLM	93.12 %	82.53 %

Table 2.1: Tagging accuracy (in %) of different models in GLMD and NLTK dataset [1]

- Uddin, Khan, Islam, and Jannat has given an accuracy comparison of different methods in their paper[2].

Model	Accuracy in %
CRF	88.3 %
CRF+NER	90.4 %
CRF + NER + Lexicon	91.6 %
CRF + NER + Lexicon +Unknown word features	92.1 %

Table 2.2: Tagging accuracy (in %) of different models[2]

- Ekbal and Bandyopadhyay have shown an comparison table on their HMM models.

Model	Accuracy (in %)
HMM	83.04 %
HMM + Unknown feature	86.38 %
HMM + Unknown feature + NER	88.45 %
HMM + unknown word features + NER + Lexicons	91.6 %

Table 2.3: Evaluation results of the POS tagger for the unknown words in the test set in HMM based POS Tagger [3]

- Ekbal, Haque, and Bandyopadhyay had some comparison on their ME- based work [4].

Model	Accuracy (in %)
ME	68.4 %
ME + Lexicon	73.3 %
ME + Lexicon + NER	79.7 %
ME + Lexicon + NER + Unknown features	88.1 %

Table 2.4: Evaluation results of the POS tagger for the unknown words in the test set in ME based POS Tagger [4]

- Sarkar and Gayen have made a comparison between Trigram and Bigram taggers' accuracy [14]

System	Accuracy in %
Trigram Tagger	78.68
Bigram Tagger	74.33

Table 2.5: System Comparisons between trigram and bigram taggers

Chapter 3

Dataset

One of the most challenging part of this work is to collect and process dataset. Thanks to pipilika.com[1]o for providing us with their tagged dataset. We have also collected an open-source XML based Project by Prof. Dr. K. M. Azharul Hasan, Md. Mostafizur Rahman, Md. Abdulla-Al-Sun that contains a huge word tag-set [2]..So we have two sets of data :

- Tagged dataset by Pipilika.com
- XML word tagset

We will discuss them here in brief.

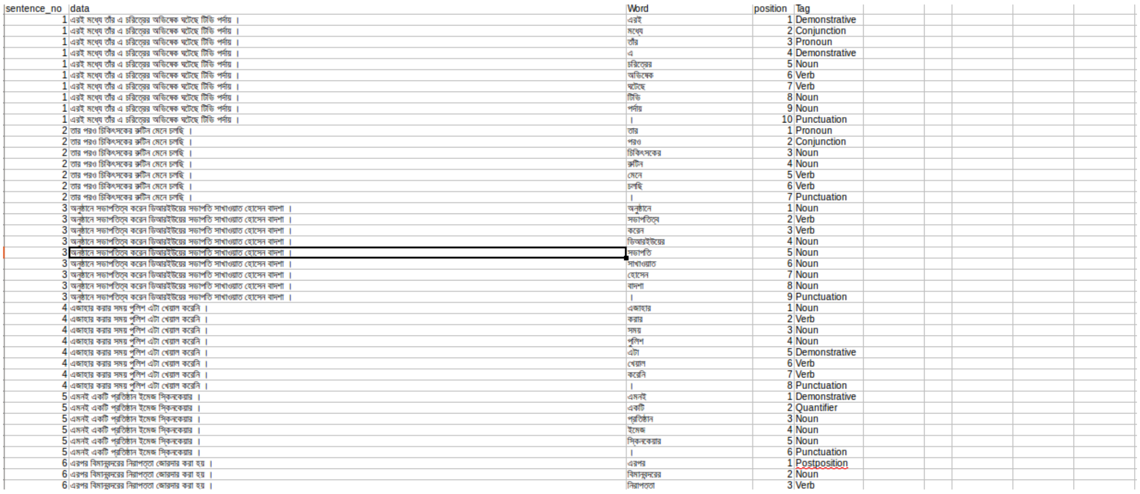
3.1 Tagged dataset by Pipilika.com

The dataset has been collected and processed by pipilika.com for their research purpose. We have collected them and processed them as we need. The dataset has 4943 lines and 47,594 words. The dataset has 12 classes :

- Adjective
- Adverb
- Conjunction
- Demonstrative
- Noun
- Particle
- Postposition
- Pronoun
- Punctuation
- Quantifier
- Residuals
- Verb

[1]pipilika.com
[2]https://github.com/sunkuet02/BanglaPosTagger

- The initial dataset is a csv file with 47,594 rows and 5 columns titled sentence no, data, word, position, Tag. So each row contains the word, the corresponding tag and the position in the sentence.



sentence_no	data	Word	position	Tag
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	এরই	1	Demonstrative
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	মধ্যে	2	Conjunction
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	তাঁর	3	Pronoun
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	এ	4	Demonstrative
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	চরিত্রের	5	Noun
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	অভিষেক	6	Verb
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	ঘটেছে	7	Verb
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	টিভি	8	Noun
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	পর্দায়	9	Noun
1	এরই মধ্যে তাঁর এ চরিত্রের অভিষেক ঘটেছে টিভি পর্দায় ।	।	10	Punctuation
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	তার	1	Pronoun
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	পরও	2	Conjunction
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	চিকিৎসকের	3	Noun
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	রুটিন	4	Noun
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	মেনে	5	Verb
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	চলছি	6	Verb
2	তার পরও চিকিৎসকের রুটিন মেনে চলছি ।	।	7	Punctuation
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	অনুষ্ঠানে	1	Noun
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	সভাপতিত্ব	2	Verb
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	করেন	3	Verb
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	ডিআরইউয়ের	4	Noun
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	সভাপতি	5	Noun
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	সাখাওয়াত	6	Noun
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	হোসেন	7	Noun
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	বাদশা	8	Noun
3	অনুষ্ঠানে সভাপতিত্ব করেন ডিআরইউয়ের সভাপতি সাখাওয়াত হোসেন বাদশা ।	।	9	Punctuation
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	এজাহার	1	Noun
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	করার	2	Verb
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	সময়	3	Noun
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	পুলিশ	4	Noun
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	এটা	5	Demonstrative
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	খেয়াল	6	Verb
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	করেনি	7	Verb
4	এজাহার করার সময় পুলিশ এটা খেয়াল করেনি ।	।	8	Punctuation
5	এমনই একটি প্রতিষ্ঠান ইমেজ স্কিনকেয়ার ।	এমনই	1	Demonstrative
5	এমনই একটি প্রতিষ্ঠান ইমেজ স্কিনকেয়ার ।	একটি	2	Quantifier
5	এমনই একটি প্রতিষ্ঠান ইমেজ স্কিনকেয়ার ।	প্রতিষ্ঠান	3	Noun
5	এমনই একটি প্রতিষ্ঠান ইমেজ স্কিনকেয়ার ।	ইমেজ	4	Noun
5	এমনই একটি প্রতিষ্ঠান ইমেজ স্কিনকেয়ার ।	স্কিনকেয়ার	5	Noun
5	এমনই একটি প্রতিষ্ঠান ইমেজ স্কিনকেয়ার ।	।	6	Punctuation
6	এরপর বিমানবন্দরের নিরাপত্তা জোরদার করা হয় ।	এরপর	1	Postposition
6	এরপর বিমানবন্দরের নিরাপত্তা জোরদার করা হয় ।	বিমানবন্দরের	2	Noun
6	এরপর বিমানবন্দরের নিরাপত্তা জোরদার করা হয় ।	নিরাপত্তা	3	Verb

Figure 3.1: Raw CSV dataset

- we have processed raw dataset and got lines of words with their corresponding tag. There are 4953 lines in the dataset.

পরে Postposition ফোর Noun দুটি Quantifier নিরাপদে Adjective গন্তব্যে Pronoun পৌঁছায় Verb বলে Verb বিআইডব্লিউটিসির Noun কর্মকর্তারা Pronoun জানান Verb । Punctuation
নাদিম Noun মোস্তফাকে Noun আটকের Verb পর Postposition একটি Quantifier বিস্ফোরক Noun মামলায় Noun গ্রেপ্তার Verb দেখিয়ে Verb তাঁকে Pronoun চার Quantifier দিনের Pronoun রিমান্ডে Verb নেয় Verb পুলিশ Noun । Punctuation
স্বজনেরা Pronoun জানিয়েছেন Verb , Punctuation আটক Verb ২০ Quantifier মেয়ে Pronoun ও Conjunction ৪০ Quantifier ছেলেশিক্ষার্থীর Pronoun বিরুদ্ধে Adjective প্রতিবাদ-বিক্ষোভে Noun জড়িত Verb থাকার Verb দায়ে Verb পাঁচ Quantifier ধরনের Adjective অভিযোগ Verb আনা Verb হয়েছে Verb । Punctuation
এ Demonstrative সময়ে Pronoun রোদের Noun তেজ Adjective ভালো Adjective । Punctuation
প্রাকৃতিক Noun গ্যাস Noun উৎপাদনের Verb সময় Pronoun এবং Conjunction পেট্রোল Noun বা Conjunction জ্বালানি Noun তেল Noun পরিশোধনের Verb সময় Pronoun উপজাত Noun বাই-প্রোডাক্ট Noun হিসেবে Adjective পাওয়া Verb যায় Verb এই Demonstrative এলপি Verb গ্যাস Noun । Punctuation
এ Demonstrative সময় Pronoun একজন Quantifier আদিবাসী Noun একটি Quantifier তির Noun তাঁর Pronoun পেটে Noun ঢুকিয়ে Verb দেন Verb । Punctuation
পরীক্ষামূলকভাবে Noun তাঁরা Pronoun একটি Quantifier হাত Noun , Punctuation একটি Quantifier মুখমণ্ডল Noun ও Conjunction একটি Quantifier দণ্ড Noun আড়াল Adjective করতে Verb সমর্থ Verb হয়েছেন Verb । Punctuation
তিনি Pronoun গণহত্যার Noun অভিযোগে Verb অবিলম্বে Adjective খালেদা Noun জিয়াকে Verb গ্রেপ্তার Verb করতে Verb সরকারের Pronoun প্রতি Verb আহ্বান Verb জানিয়েছেন Verb । Punctuation
কারণ Adverb , Punctuation এফটিএ Noun করতে Verb পারলে Demonstrative তারা Pronoun রপ্তানি Verb কয়েক Adjective গুণ Verb বাড়িয়ে Verb ফেলবে Verb । Punctuation
প্রসিডিংস Noun অফ Noun দ্য Noun রয়েল Noun সোসাইটি Noun বি Noun জার্নালে Noun এই Demonstrative গবেষণা Verb প্রকাশিত Verb হয় Verb । Punctuation
জাহিদুল Noun ইসলাম Noun জানান Verb , Punctuation পদ্মা Noun এক্সপ্রেসের Noun দুর্ঘটনার Noun পর Postposition এ Demonstrative পর্যন্ত Adjective বিভিন্ন Pronoun গন্তব্যের Adjective তিনটি Quantifier ট্রেন Noun বাতিল Verb করা Verb হয়েছে Verb । Punctuation
তাদের Pronoun জন্য Adverb কিছু Adjective করার Verb তাগিদ Adjective অনুভব Adjective করেন Verb নিখিল Verb । Punctuation
তাদের Pronoun জন্য Adverb কিছু Adjective করার Verb তাগিদ Adjective অনুভব Adjective করেন Verb নিখিল Adjective । Punctuation
মিসরের Noun কায়রো Noun ফুটবল Noun স্টেডিয়ামে Noun গতকাল Pronoun রোববার Noun ফুটবল Noun খেলায় Noun সহিংস Noun ঘটনায় Noun ২২ Quantifier জন Noun নিহত Verb হয়েছে Verb । Punctuation
এছাড়াও Adverb বিশ্ববিদ্যালয়ের Noun বিভিন্ন Noun সামাজিক Noun ও Conjunction রাজনৈতিক Noun সংগঠনের Pronoun পক্ষ Adjective থেকেও Adjective প্রস্তুতি Verb নেওয়া Verb হয়েছে Verb । Punctuation
তাই Demonstrative মুরগির Noun খাদ্যে Noun খনিজ Noun উপাদান Noun ক্যালসিয়াম Noun ও Conjunction ফসফরাসের Noun গুরুত্ব Verb নিয়ে Verb আজকে Pronoun আলোচনা Verb করবো Verb । Punctuation
শনিবার Noun সন্ধ্যা Noun সোয়া Quantifier ৬টায় Quantifier কাজির Noun গ্রাম Noun এলাকায় Noun এ Demonstrative দুর্ঘটনা Verb ঘটে Verb বলে Verb পুলিশ Noun জানিয়েছে Verb । Punctuation
চার Quantifier মিনিট Noun পরেই Postposition হামেস Noun রদ্রিগেজের Noun ক্রস Noun বাঁ Noun পায়ের Noun আলতো Adjective টোকায় Verb স্কোরলাইন Noun ২-০ Quantifier করে Verb ফেলেন Verb বেল Noun । Punctuation
আরও Conjunction অনেক Pronoun পরিযায়ী Noun পাখির Noun মতো Adjective তারা Pronoun অপেক্ষাকৃত Verb উষ্ণ Verb ইরাকে Noun ফেব্রুয়ারি Noun মাস Noun পর্যন্ত Adjective থাকে Verb । Punctuation
দুই Quantifier ম্যাচ Noun শেষে Adjective দক্ষিণ Noun আফ্রিকা Noun এগিয়ে Verb ১-০-তে Quantifier । Punctuation
ঁ Punctuation উকুন Noun , Punctuation আঁঠালী Noun , Punctuation মশা Noun ও Conjunction মাছি Noun দংশন Verb করার Verb সময় Adjective প্রাণী Pronoun অস্থির Verb ও Conjunction অস্বস্তি Noun বোধ Verb করে Verb । Punctuation
বঙ্গবন্ধুর Noun স্বদেশ Noun প্রত্যাবর্তন Noun দিবস Noun উপলক্ষে Noun আগামীকাল Pronoun সোমবার Noun বেলা Noun দুটায় Quantifier রাজধানীর Pronoun সোহরাওয়ার্দী Noun উদ্যানে Noun বড় Adjective রকমের Adjective সমাবেশ Noun করার Verb প্রস্তুতি Verb নিয়েছে Verb আওয়ামী Noun লীগ Noun । Punctuation
দেশটিতে Pronoun মার্কিন Noun আইরোবট Noun কোম্পানির Noun রুম্বা Noun ও Conjunction জাপানি Noun প্যানাসনিক Noun কোম্পানির Noun তৈরি Verb রোবটের Noun বাজার Noun দখলের Verb লড়াই Verb তীব্র Adjective । Punctuation
ভোলা Noun শহরের Noun খালপাড় Noun সড়কেও Noun গতকাল Pronoun ভোররাতে Noun একটি Quantifier ট্রাক Noun আগুনে Noun পুড়িয়ে Verb দেয় Verb দুর্বৃত্তরা Pronoun । Punctuation

Figure 3.2: Lines with tags

- Then we have created a file containing the frequency of each word being the Parts of Speech.

From this file, we get 11528 different words and their frequency of being each Parts of Speech in the dataset.

words	Adjective	Adverb	Conjunction	Demonstrative	Noun	Particle	Postposition	Pronoun	Punctuation	Quantifier	Residuals	Verb	total
।	0	0	0	0	0	0	0	0	4943	0	0	0	4943
অংশ	17	0	0	0	0	1	0	2	0	1	1	14	36
অংশই	1	0	0	0	0	0	0	0	0	0	0	0	1
অংশগ্রহণ	0	0	0	0	0	0	0	0	0	0	1	4	5
অংশগ্রহণকারী	0	0	0	0	0	0	0	7	0	0	0	0	7
অংশগ্রহণকারীদের	0	0	0	0	0	0	0	2	0	0	0	0	2
অংশগ্রহণমূলক	0	0	0	0	0	0	0	0	0	0	0	1	1
অংশগ্রহণের	0	0	0	0	0	0	0	0	0	0	0	2	2
অংশিদারিত্বের	1	0	0	0	0	0	0	0	0	0	0	0	1
অংশীদার	1	0	0	0	0	0	0	0	0	0	0	0	1
অংশীদারের	0	0	0	0	0	0	0	1	0	0	0	0	1
অংশে	1	0	0	0	0	0	0	1	0	0	0	0	2
অংশেই	1	0	0	0	0	0	0	0	0	0	0	0	1
অংশেও	1	0	0	0	0	0	0	0	0	0	0	0	1
অংশের	3	0	0	0	0	0	0	0	0	0	0	0	3
অংশেরই	1	0	0	0	0	0	0	0	0	0	0	0	1
অঙ্কটায়	0	0	0	0	4	0	0	0	0	0	0	0	4
অঙ্কটাযে	0	0	0	0	1	0	0	0	0	0	0	0	1
অঙ্কটাযের	0	0	0	0	1	0	0	0	0	0	0	0	1
অঙ্কটায়	0	0	0	0	5	0	0	0	0	0	0	0	5
অকল্পনত্ত	1	0	0	0	0	0	0	0	0	0	0	0	1
অক্ষত	1	0	0	0	0	0	0	0	0	0	0	0	1
অক্ষুণ্ন	1	0	0	0	0	0	0	0	0	0	0	0	1
অক্সিজেন	0	0	0	0	1	0	0	0	0	0	0	0	1
অক্সিডেটযুক্ত	0	0	0	0	1	0	0	0	0	0	0	0	1
অখিলেশ	0	0	0	0	1	0	0	0	0	0	0	0	1
অখুশি	1	0	0	0	0	0	0	0	0	0	0	0	1
অগম্য	1	0	0	0	0	0	0	0	0	0	0	0	1
অগ্নি	0	0	0	0	1	0	0	0	0	0	0	0	1
অগ্নিনিরাপত্তা	0	0	0	0	1	0	0	0	0	0	0	0	1
অগ্নিসংযোগ	0	0	0	0	0	0	0	0	0	0	0	1	1
অগ্নিসংযোগের	0	0	0	0	0	0	0	0	0	0	0	1	1
অগ্রগণ্য	1	0	0	0	0	0	0	0	0	0	0	0	1
অগ্রগতি	0	0	0	0	0	0	0	0	0	0	0	2	2
অগ্রগতির	0	0	0	0	0	0	0	0	0	0	0	1	1
অগ্রগী	0	0	0	0	0	0	0	0	0	0	0	1	1
অগ্রাধিকার	2	0	0	0	0	0	0	0	0	0	0	1	3
অগ্রাধিকারভিত্তিতে	2	0	0	0	0	0	0	0	0	0	0	0	2
অগ্রিম	2	0	0	0	0	0	0	0	0	0	0	0	2
অঘোষিতভাবে	0	0	0	0	0	0	0	0	0	0	0	1	1
অঙ্কে	0	0	0	0	0	0	0	1	0	0	0	0	1

Figure 3.3: word tag frequency

3.2 XML word tagset

We needed a huge corpus of words with their tag frequency. Unfortunately, we could not collect or create this huge corpus. So we have taken the open-source XML based Project by Prof. Dr. K. M. Azharul Hasan, Md. Mostafizur Rahman, Md. Abdulla-Al-Sun. This XML file contains 1,12,382 word with corresponding tag[3].

- The XML dataset was in the form presented in the figure 3.4
- The word Dictionary was in XML form, so we had to convert it into text form where the first word of a line contains the word and the second one defines the respective POS.

The figure 3.5 shows the processed samples in the figure 3.4.

[3]https://github.com/sunkuet02/BanglaPosTagger

<অ>adj</অ>
<অংশ>noun</অংশ>
<অংশগ্রহণ>noun</অংশগ্রহণ>
<অংশগ্রহণকারী>noun</অংশগ্রহণকারী>
<অংশতঃ>adv</অংশতঃ>
<অংশাঙ্কিত>adv</অংশাঙ্কিত>
<অংশিত>adv</অংশিত>
<অংশী>adv</অংশী>
<অংশীদার>noun</অংশীদার>
<অংশীদারি>noun</অংশীদারি>
<অংশু>noun</অংশু>
<অংশুক>noun</অংশুক>
<অংশুমান>adv</অংশুমান>
<অংস>noun</অংস>
<অংসকূট>noun</অংসকূট>
<অংসকূট>noun</অংসকূট>
<অংসফলক>noun</অংসফলক>
<অংসল>adv</অংসল>
<অঋণী>adv</অঋণী>
<অকণ্টক>adv</অকণ্টক>
<অকথন>adv</অকথন>
<অকথনীয়>adv</অকথনীয়>

Figure 3.4: XML-based tagset sample

- our dataset is divided into 80:20 for training and test set. The training set contains 3461 lines and the test set contains 1483 lines. We have also processed the training lines suitable for our training features. For each word, we have created a line containing the probabilistic feature of it's previous and next two words and it's own feature. An example is shown in 3.6

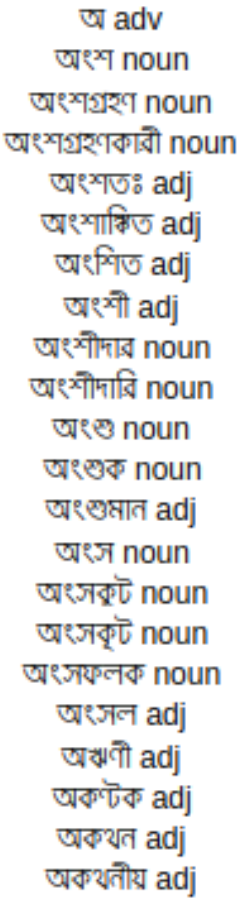


Figure 3.5: Processed Tagset Sample

1	2	3	4	5	6
0	0	11	10.4667	10.5455	5
0	11	10.4667	10.5455	2.25	5
11	10.4667	10.5455	2.25	11	5
10.4667	10.5455	2.25	11	11	1
10.5455	2.25	11	11	10.3636	5
2.25	11	11	10.3636	24.75	5
11	11	10.3636	24.75	24.9946	5
11	10.3636	24.75	24.9946	19	12
10.3636	24.75	24.9946	19	0	12
24.75	24.9946	19	0	0	9
0	0	11	11	24.9928	5
0	11	11	24.9928	19	5
11	11	24.9928	19	11	12
11	24.9928	19	11	11	9
24.9928	19	11	11	24.5556	5
19	11	11	24.5556	24.9864	5
11	11	24.5556	24.9864	2.66667	12
11	24.5556	24.9864	2.66667	25	12
24.5556	24.9864	2.66667	25	11	1
24.9864	2.66667	25	11	19	12
2.66667	25	11	19	0	5
25	11	19	0	0	9
0	0	16.9545	24.8182	24.5556	8
0	16.9545	24.8182	24.5556	16.9091	12
16.9545	24.8182	24.5556	16.9091	25	12
24.8182	24.5556	16.9091	25	6.95666	8
24.5556	16.9091	25	6.95666	25	12
16.9091	25	6.95666	25	25	3
25	6.95666	25	25	16.2857	12
6.95666	25	25	16.2857	19	12
25	25	16.2857	19	0	8

Figure 3.6: Training Features

Chapter 4

Methodology

After reviewing all the previous works in Bengali POS tagging, we have noticed some properties in these methods

- From the comparison by Mukherjee and Mandal[1], we see that all the models show an average accuracy of 85%-90% on their dataset but in case of the NLTK dataset, they show at best 82%.
- Most of the models use other models or tools like NER, stemmer, lexicon development. So the performance of these models depends on the performance of these tools or models.
- Most of the models either use probabilistic methods or use other features for training the neural networks. Neither of them uses probability as features for the network.

From these points, we had some primary goals to fulfill -

- use probability as an feature for training the network.
- Reduce dependency on other tools
- check all possible set of tag to get the best tagset

Most of the models either use probabilistic methods or use other features for training the neural networks. Neither of them uses probability as features for the network. So we started with a new model based on the model proposed by Uddin, Khan, Islam, and Jannat[2].In the model, we will use the words' respected probability as the feature for the neural network We also proposed to use the Prefix Tree (trie) and Dynamic Programming approach to reduce time and space complexity.

4.1 Probability as Feature

In English POS Tagging, different features have been used like :

- If the word starts with a capital letter
- If the word contains digits or hyphen
- Suffixes and prefixes of the word
- Complete capitalized word

In Bengali, there is nothing like the capital letter. It is also hard to find the suffix and prefix of the word in Bengali. So we have decided to use the probability of their perspective tag as feature.

4.1.1 Probability Calculation

At first, we have calculated the frequency of all our words being each POS. Then we have got the probabilities in the method described below :

- Equation to calculate probability

$$P(x, a) = \frac{cnt(x, a)}{cntAll(x)} \tag{4.1}$$

$$P(x, a) = the\ probability\ of\ word\ x\ being\ the\ POS\ a,$$

$$cnt(x, a) = the\ total\ number\ count\ of\ word\ x\ being\ POS\ a$$

$$cntAll(x) = the\ total\ occurrence\ of\ the\ word\ x\ in\ the\ dataset$$

4.1.2 Feature Calculation

For every word and tag, the probability feature will be calculated as :

- the input features will be :
  - **Adjective** : 0 + P(x, Adjective)
  - **Adverb** : 1 + P(x, Adverb)

  - **Conjunction** : 2 + P(x, Conjunction)
  - **Demonstrative** : 3 + P(x, Demonstrative)
  - **Noun** : 4 + P(x, Noun)
  - **Particle** : 5 + P(x, Particle)
  - **Postposition** : 6 + P(x, Postposition)
  - **Pronoun** : 7 + P(x, Pronoun)
  - **Punctuation** : 8 + P(x, Punctuation)
  - **Quantifier** : 9 + P(x, Quantifier)
  - **Residuals** : 10 + P(x, Residuals)
  - **Verb** : 11 + P(x, Verb)

- The output feature will be :
  - **Adjective** : 0
  - **Adverb** : 1
  - **Conjunction** : 2
  - **Demonstrative** : 3
  - **Noun** : 4
  - **Particle** : 5
  - **Postposition** : 6
  - **Pronoun** : 7
  - **Punctuation** : 8
  - **Quantifier** : 9
  - **Residuals** : 10
  - **Verb** : 11

4.2 Prefix Tree (Trie)

] Trie,also known as the prefix tree is a kind of search tree data structure that stores data especially strings in a structured way. It was first introduced by Rene de la Briandais in 1959[16][17].The term trie was coined two years later by Edward Fredkin[18][17]. The term was found from the word retrieve[17].It's searching technique mainly based on the matched prefix of the words.

We have used Trie Data Structure in both Data Preprocessing stem to calculate the frequency and

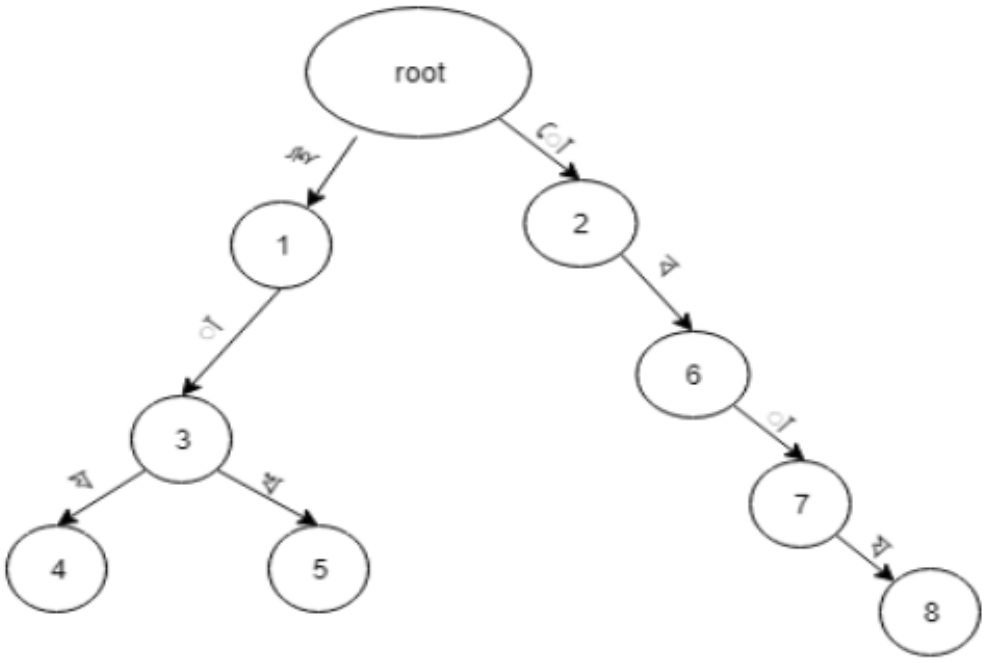


Figure 4.1: Trie Structure

during both the train and the evaluation process. Without Trie, the searching process would take a lot more time complexity. For each word search, it can be O(N), where N is the summation of all the words' length in the dictionary without trie. Using trie it would be only O(n), where n is the length of the searched word. We have stored all the words in the dataset as a string and their frequencies in the leaf nodes.

4.3 Training Process

According to the model proposed in the paper[2], The Neural Network will have a simple structure of 3 layers containing 5,3 and 1 nodes. First layer of 5 nodes will receive the word's, it's previous two words' and the next two words' probability feature as input. And the last layer will provide the output But We have decided to uexperiment with more network architectures for

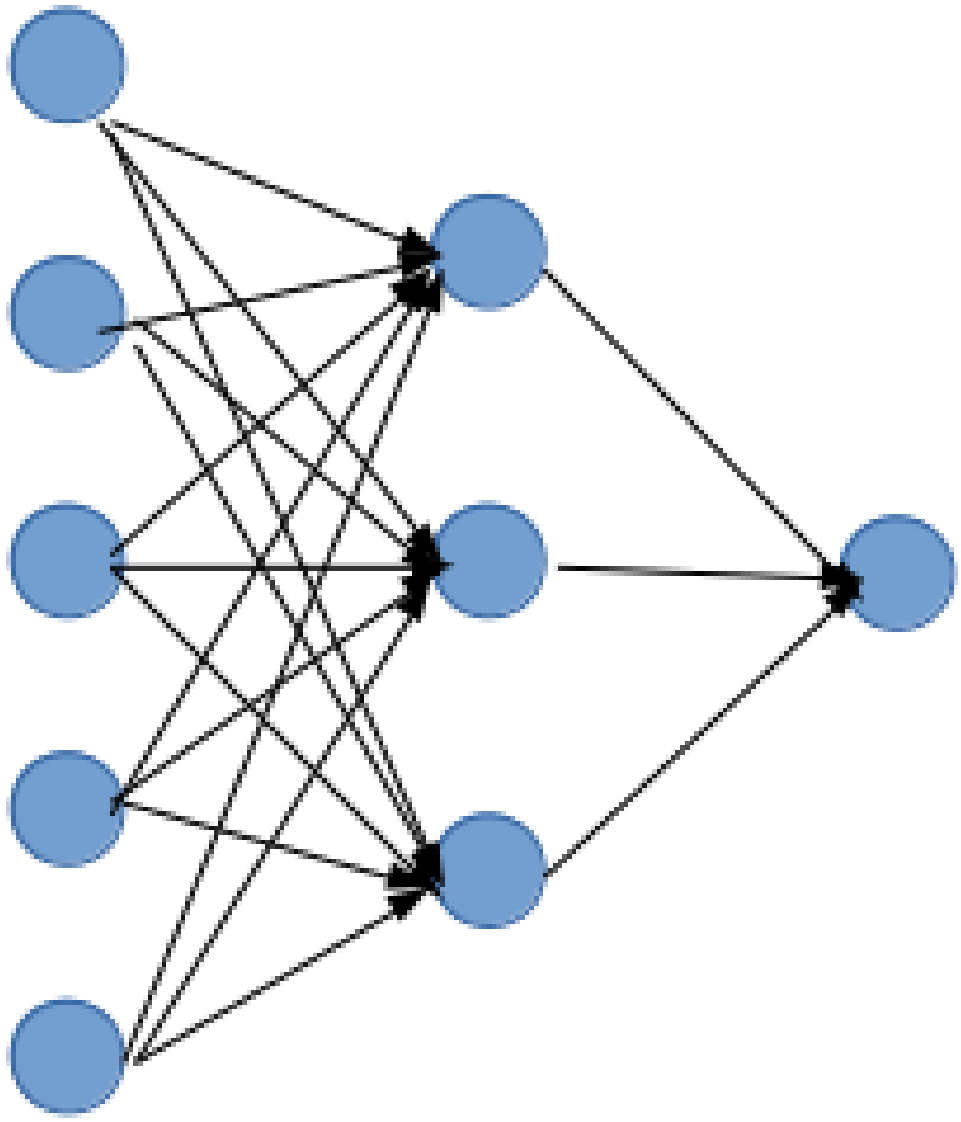


Figure 4.2: Neural Network Structure

training purposes like Sequential Model, Bi-Directional LSTM.

4.4 Tagging Process

One of the most important parts of our research was the tagging process. We would generate all possible sets of tags. For a set, we would calculate the error score by matching it to the networks predicted result. So for each word, we would give the network the probabilistic feature input of the word's, it's previous two words, and the next two words. If they are absent we would provide 0 as input. If the predicted result does not match the generated result we add 1 to the error score. The set with the minimum error score is our result tag set.

4.4.1 Complexity Analysis

As we are generating all sets or combinations of tags of a line then this will generate all 12^n combinations. That means it will have the time complexity of $O(12^n)$. Which is in case of lines having more than 8 words is a huge time complexity to bear for a single line. So we have introduced the Dynamic Programming method to resolve this issue.

4.4.2 Dynamic Programming Method

Here we can divide the problem into subproblems. As we are giving only the five consecutive words' features as input while generating the best answer we only need to care about these five consecutive tags. So we can compute the best result as the DP function :
DP[tag1][tag2][tag3][tag4][pos] = MAX(DP[tag2][tag3][tag4][1][pos+1]+ score(tag1,tag2,tag3,tag4,1),DP[tag2][tag3
. . . . . , DP[tag2][tag3][tag4][12][pos+1]+score(tag1,tag2,tag3,tag4,12))

- tag1 is the tag of word[pos-2]
- tag2 is the tag of word[pos-1]
- tag3 is the tag of word[pos]
- tag4 is the tag of word[pos+1]
- score(tag1,tag2,tag3,tag4,i) = 1 if predicted tag for input set tag1,tag2,tag3,tag4 and i is tag3 otherwise 0.

This has reduced our complexity to O(12^5) for each word. That is O(12^5*n) for each line where n is the number of words in a line which is a huge improvement comparing to the O(12^n)

4.4.3 Unknown Words

While tagging there is a huge chance of countering some unknown words. To sort that out, we have decided to use the XML dictionary. We will consider the word will have the most probability of 0.78 of being the tag in the dictionary. And other tags will have a probability sharing of 0.02

Chapter 5

Experiments

We have run several experiments based on the initial models keeping some properties basic like:

- Basic features for training
- Generating all set and get the best answer

We have experimented with:

- Different neural network architecture
- Class differences for feature
- Choosing higher probabilistic tags for tagging.

We have divided our dataset into 80:20 for training and test set. The training set contains 3461 lines and the test set contains 1483 lines. We have also processed the training lines suitable for our training features. For each word, we have created a line containing the probabilistic feature of it's previous and next two words and it's own feature. An example is shown in figure 3.6

5.1 Initial Model

We have conducted our first experiment exactly based on the model described in section 4.3 and 4.4. We have got an accuracy of only **51.71%.** An example is shown in figure 5.1.

```
words      ---> আহতরা সবাই সাতকানিয়ার রামপুর বণিক পাড়া এলাকার বাসিন্দা ।
tags       ---> Pronoun Pronoun Noun Noun Noun Noun Pronoun Pronoun Punctuation
prediction ---> Pronoun Adjective Noun Noun Noun Noun Noun Adjective Punctuation
accuracy 0.6666666666666666

words      ---> এর দাম ২১ হাজার টাকা ।
tags       ---> Pronoun Noun Quantifier Quantifier Noun Punctuation
prediction ---> Demonstrative Adjective Adjective Adjective Adjective Adjective
accuracy 0.0

words      ---> বললেন , ঘরের ভেতরের প্রদর্শনীর চেয়ে এটাই বরং ভালো ।
tags       ---> Verb Punctuation Noun Verb Adjective Conjunction Particle Conjunction Adjective Punctuation
prediction ---> Verb Punctuation Noun Adjective Adjective Adjective Demonstrative Conjunction Adjective Punctuation
accuracy 0.7

words      ---> অথচ আমাদের প্রয়োজন হলো ২০ হাজার কিউসেক ।
tags       ---> Conjunction Pronoun Verb Verb Quantifier Noun Noun Punctuation
prediction ---> Conjunction Pronoun Verb Adjective Quantifier Adjective Adjective Adjective
accuracy 0.5

total Accuracy    0.5171740201252551
```

Figure 5.1: Initial model tags example

5.1.1 Drawbacks

From the initial model, we can see some major drawbacks :

- As we have discussed our tagging process in section 4.4, we are generating all possible combinations and then taking the one with minimum error score as the result. But there can be some cases where error score can be the same for multiple combinations. We did not handle this case so the program was picking the first best combination as a result. That is not a good idea to implement.

- As discussed in section 4.1 while calculating feature we are adding the probability to the corresponding class value. Where class values are 1,2...12. The difference between the two consecutive class values is only 1. So let for a word, the probability of being class 1 is 0.99999 and for class 2 it is 0.000001 so we will get the feature as 1.99999 and 2.000001 both are very close to each other. So they may create a problem for some models.

5.1.2 modifications

In order to recover the drawbacks discussed, we have decided to run further experiments with some modifications.

- **High Probability First :** In case of the same error score for different combinations, we will take the one with the highest probability. For a combination of words and tags ($word_1$, tag_1), ($word_2$, tag_2) ... ($word_n$, tag_n) the probability of being that combination a good one

is product of the probabilities P(word$_i$, tag$_i$). That means we will take the maximum of $\prod_{i=1}^{len(sentence)} P(word_i, tag_i)$.This technique theoretically increases the chance of getting the best tags.

- **Higher Class Differnce :** For the second drawback, we would increase the class difference by 1 that is the class scores will be now 2,4,6...12. That decreases the chance of two different tags getting the same value.

5.1.3 Experiments Results

We have conducted three different experiments on the initial model. first without any modification, then adding the high probability first and then with a higher class difference. The result is shown in the table 5.1

model	]**accuracy**
initial model	51.71%
initial model + Higher Probability First	72.9%
initial model + Higher Probability First + Higher Class Difference	84.7%

Table 5.1: experiment results with initial model

5.2 Multilayer Architecture

We have conducted some more experiments with keeping every other thing unchanged except the neural network. We introduced the modern multilayer neural network architectures.

5.2.1 Sequential Model

We have used Keras Sequential model[19] to build our multilayer Architecture for training purposes. Sequential is the easiest way to build a model layer by layer. We have experimented on 2 different architectures. We have got 82

- Our first network is described in figure 5.2. We have run two tests, one with Higher Class Difference and another without it. We have kept Higher Probability First in both the experiments. The number of iteration in this network was only 100.

```
model = Sequential()
model.add(Dense(3, input_shape=(5,), activation='relu'))
model.add(Dense(12, activation='softmax'))

#stochastic gradient descent
sgd = SGD(lr=0.001, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
return model
```

Figure 5.2: Sequential model Architecture 1

We have got 82.7% without Higher Class Difference and 89.3% with it. The result is shown in table 5.2

model	]**accuracy**
Sequential model + Higher Probability First	82.7%
Sequential model + Higher Probability First + Higher Class Difference	89.3%

Table 5.2: experiment results with Sequential model Architecture 1

- Our Second network is described in figure 5.3. The differences from the earlier network are that it has 5 dense layers and iteration is increased to 1000. We have run two tests, one with Higher Class Difference and another without it. We have kept Higher Probability First in both the experiments.

```
model = Sequential()
model.add(Dense(5, input_shape=(5,), activation='relu'))
model.add(Dense(12, activation='softmax'))

#stochastic gradient descent
sgd = SGD(lr=0.001, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
return model
```

Figure 5.3: Sequential model Architecture 2

We have got 83.8% without Higher Class Difference and 89.45% with it. The result is shown in table 5.3

model	]**accuracy**
Sequential model + Higher Probability First	83.8%
Sequential model + Higher Probability First + Higher Class Difference	89.45%

Table 5.3: experiment results with Sequential model Architecture 2

5.3 Bidirectional LSTM

LSTM (long short-term memory) is a novel, gradient-based method for sequence labeling[20]. LSTM has been used for POS Tagging in different languages[21][22][23][24].We have used the Keras BiDirectional LSTM model. The model architecture is described in figure 5.4.

```
model = Sequential()
model.add(Bidirectional(LSTM(20, return_sequences=True), input_shape=(5,)))
model.add(Dense(12, activation='softmax'))

#stochastic gradient descent
sgd = SGD(lr=0.001, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
return model
```

Figure 5.4: Bi-LSTM Network Architecture

We have conducted three different tests on the model. The results have been shown in the table 5.4

model	]**accuracy**
Bi-LSTM	55.77%
Bi-LSTM + Higher Probability First	85.56%
Bi-LSTM + Higher Probability First + Higher Class Difference	90(89.98)%

Table 5.4: experiment results with Bi-Direcitional LSTM

Chapter 6

Discussion

Here we will have a short analysis of our works and the places of improvement in this work.

6.1 Result Analysis

So far now, we have trialed with 3 different Architecture and rectified 2 major drawbacks. The whole work's output can be shown in a single table 6.1 From table 6.1, it is clear that without using

model	]**accuracy**
initial model	51.71%
Bi-LSTM	55.77%
initial model + Higher Probability First	72.9%
Sequential model + Higher Probability First	83.8%
initial model + Higher Probability First + Higher Class Difference	84.7%
Bi-LSTM + Higher Probability First	85.56%
Sequential model + Higher Probability First + Higher Class Difference	89.45%
Bi-LSTM + Higher Probability First + Higher Class Difference	90(89.98)%

Table 6.1: experiment results with initial model

Higher Probability First and Higher Class Difference, the accuracy is less than 60% in each model. But using Higher Probability First, it jumps to almost 85%. Adding Higher Class Difference we can achieve the best result of almost 90%. So we can say that without rectifying these two drawbacks we could not get a satisfactory result.

6.2 Scopes of Works

We can not say that the work is anywhere close to the satisfactory works done on other languages. So there are a lot of things that can be improved.

6.2.1 A Corpus of word and tags

We had to use an XML based dictionary for handling unknown words problems in a very naive way discussed in section **??**. So we need a huge data corpus to provide us with more words and tags. That will strengthen the base of this works, feeding probability as a feature.

6.2.2 CRF based Model

The best result in Bengali POS Tagging has been found using the CRF based model. We could not work with that. So we are hopeful of a good result using that technique.

6.2.3 Merging with traditional features

Using probability as a feature along with the traditional morphological features can be a huge advantage in the work of this field. This can enhance the chance to get a better model.

Chapter 7

Conclusion

In the whole report, we have discussed our works and previous works to get an idea about the whole research. As discussed earlier we could reach a satisfactory accuracy of 90%. We could not compare our works with the previous works due to the dataset and working areas. But we can claim that using probability as a feature and getting the best out of all combinations is a wholesome model to proceed with. This could be used to build a practical POS Tagger and an engrossing model to conduct further researches on this type of research fields.

References

[1] S. Mukherjee and S. K. Das Mandal, "Bengali parts-of-speech tagging using global linear model," in *2013 Annual IEEE India Conference (INDICON)*, Dec 2013, pp. 1–4.

[2] M. Nazim Uddin, M. S. Islam, M. Ahmad Khan, and M.-E. Jannat, "A neural network approach for bangla pos tagger," 09 2018, pp. 1–4.

[3] A. Ekbal and S. Bandyopadhyay, "Lexicon development and pos tagging using a tagged bengali news corpus." in *FLAIRS Conference*, 2007, pp. 261–262.

[4] A. Ekbal, R. Haque, and S. Bandyopadhyay, "Maximum entropy based bengali part of speech tagging," *A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal*, vol. 33, pp. 67–78, 2008.

[5] D. Nguyen, D. Nguyen, D. Pham, and S. Pham, "Rdrpostagger: A ripple down rules-based part-of-speech tagger," in *EACL 2014*. Association for Computational Linguistics (ACL), 2014, pp. 17–20.

[6] D. Q. Nguyen, D. Q. Nguyen, D. D. Pham, and S. B. Pham, "A robust transformation-based learning approach using ripple down rules for part-of-speech tagging," *AI Communications*, vol. 29, no. 3, pp. 409–422, 2016.

[7] H. Schmid, "Improvements in part-of-speech tagging with an application to german," in *Natural language processing using very large corpora*. Springer, 1999, pp. 13–25.

[8] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[9] O. Dongsuk, S. Kwon, K. Kim, and Y. Ko, "Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2704–2714.

[10] P. Alva and V. Hegde, "Hidden markov model for pos tagging in word sense disambiguation," in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. IEEE, 2016, pp. 279–284.

[11] S. Dandapat, S. Sarkar, and A. Basu, "A hybrid model for part-of-speech tagging and its application to bengali." in *International conference on computational intelligence*. Citeseer, 2004, pp. 169–172.

[12] A. Ekbal, R. Haque, and S. Bandyopadhyay, "Bengali part of speech tagging using conditional random field," in *Proceedings of Seventh International Symposium on Natural Language Processing (SNLP2007)*, 2007, pp. 131–136.

[13] A. Ekbal and S. Bandyopadhyay, "Part of speech tagging in bengali using support vector machine," in *2008 International Conference on Information Technology*, Dec 2008, pp. 106–111.

[14] K. Sarkar and V. Gayen, "A practical part-of-speech tagger for bengali," in *2012 Third International Conference on Emerging Applications of Information Technology*, Nov 2012, pp. 36–40.

[15] D. Chakrabarti and P. CDAC, "Layered parts of speech tagging for bangla," *Language in India, www. languageinindia. com, Special Volume: Problems of Parsing in Indian Languages*, 2011.

[16] R. De La Briandais, "File searching using variable length keys," in *Papers presented at the the March 3-5, 1959, western joint computer conference*. ACM, 1959, pp. 295–298.

[17] Wikipedia contributors, "Trie — Wikipedia, the free encyclopedia," 2019, [Online; accessed 13-July-2019]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Trieoldid=905392747

[18] P. E. Black, “Dictionary of algorithms and data structures,” Tech. Rep., 1998.

[19] P. Charles, “Project title,” https://github.com/charlespwd/project-title, 2013.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[21] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional lstm-cnns-crf,” 2016.

[22] Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” 2015.

[23] B. Plank, A. Søgaard, and Y. Goldberg, “Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss,” 2016.

[24] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, “A unified tagging solution: Bidirectional lstm recurrent neural network with word embedding,” 2015.