Deep Learning based Morphological Analysis for Bhojpuri

An exploratory project by **Ankit** and **Sakshi**

Supervised by :

Dr. Anil Kumar Singh

Mentored by :

Mr. Amit Kumar

Contents

Contents

Chapter 1

What is Natural Language Processing?

Why is it even needed?

NLP

Natural Language Processing is a subfield of *linguistics*, *computer science*, and *artificial intelligence* concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

NLP is needed because:

- It helps **resolve ambiguity** in language and **adds proper numeric structure** to the data for many downstream applications, such as **speech recognition** or text analytics.
- It helps to **train AI models** used in **voice assistants** like Google Assistant, Cortana, Siri, etc,.
- It helps linguists manage low-level language

Chapter 2

What is Morphological Analysis?

Why is it so hyped now-a-days?

Morphological Analysis

Morphological analysis refers to the task of assigning a set of well-defined *morphological tags* and a *lemma* (root) to the data of a language by studying various syntactic attributes such as inflection, derivation and combining forms. In layman terms, it is a study of word formation, i.e. how words are built using smaller parts, and knowing those parts makes it easier to translate a word in one language into another.

Morphological Analysis is hyped because:

➔ It might help us discover new relationships or configurations, which may not be so evident, or which might have overlooked by other.

➔ It helps **resolve ambiguity** in language and **adds proper numeric structure** to the data for many downstream applications, such as **speech recognition** or text analytics.

➔ It encourages the identification and investigation of **boundary conditions**, i.e. the **limits and extremes** of different contexts and factors

Chapter 3

Bhojpuri and its Morphology

Why did we choose Bhojpuri?

Bhojpuri.

Bhojpuri is an Indo-Aryan language spoken in **east-central** region of India and the **Terai region of Nepal**. It is chiefly spoken in western Bihar and eastern Uttar Pradesh.

Sociolinguistically, it is often considered one of several **Hindi dialects**.

This language needs more attention in the NLP fields because of its **morphologically rich**, **non-configurational**, and **agglutinative nature**.

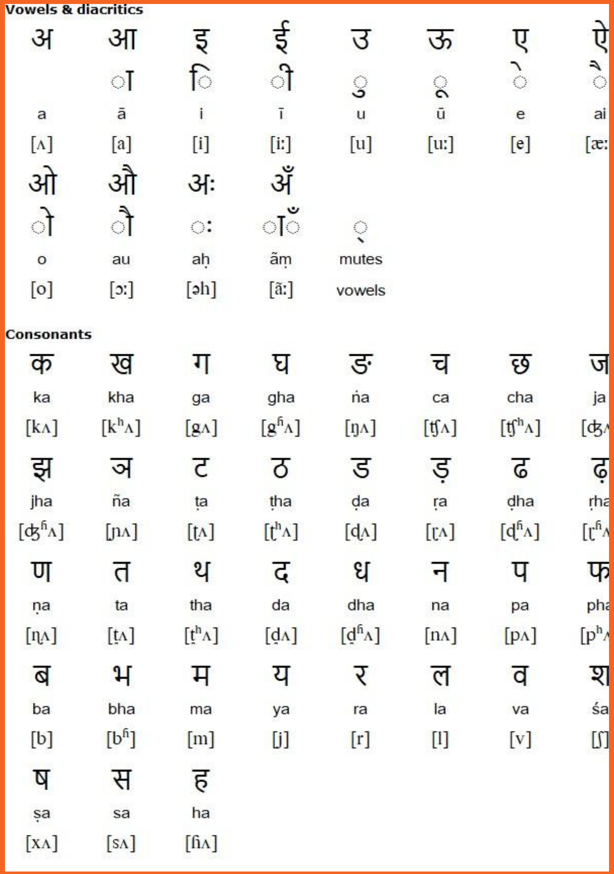
An example.

A word as simple as **speak** (*bolo* in Hindi), has several forms in Bhojpuri depending upon the context.

Literary	**bōl**
Casual and intimate	**bōl**
Polite and intimate	**bōl'** (or **bōla**)
Formal yet intimate	**bōlīñ**
Polite and formal	**bōlīñ**
Extremely formal	**bōlal jā'e**

—

Hence, we thought it would be an interesting project to work on.



WX Notation we used for Bhojpuri Language

Chapter 4

Challenges

And how we resolved them?

Bhojpuri being a low-resource language has these issues:

- ➔ **Unavailability of good dataset-** Even though, there is a dataset on Universal Treebank Dependencies Website, but it has only 8,000 words, which can't be used as Training Dataset.

- ➔ **Less Morphological Research Background-** The morphological research done on Bhojpuri language is way less when compared to other languages with similar resource availability.

So, we used Hindi as the training data for a model that will predict tags and lemma for Bhojpuri. Why?

→ Since, Bhojpuri is a dialect of Hindi sociolinguistically. Hence, there are a few features common in both. And because of this, we can use it to test Bhojpuri Data on Hindi.

And, this method is known as **Unsupervised Domain Adaptation** (UDA) in **Zero Shot Learning Condition** (ZSL).

Chapter 5

Approach

And the way we proceeded?

Preprocessing:

Since, the data we are using for the model is in **CoNLL format**, which needs **two steps** of preprocessing before sending it to the model.

- ➔ The CoNLL data has a **lot of comments** before (every sentence), so it was first taken care of by using **Regex**.

- ➔ CoNLL data has words in the **UTF format**, so we then need to **convert it into WX notation** using a custom **WX converter**.

And, then the final data is sent to the model. Our model consists of two major parts:

- → **Tag Predictor:** It predicts the tags on the basis of training data.
- → **Lemma Predictor:** It predicts the lemma on the basis of training data.

And their functioning is explained in the next few slides.

Tag Predictor:

1. Firstly, the **input data vector v** is passed through the **Dropout layer**, which predicts the target Y, after passing input data vector v through various hidden layers.

2. The target Y predicted from the dropout layer passes through GNL (**Gaussian Noise Layer**), which adds zero-centered Gaussian Noise into Y.

3. The noise-infected data goes through **convolution and pooling** parallelly.

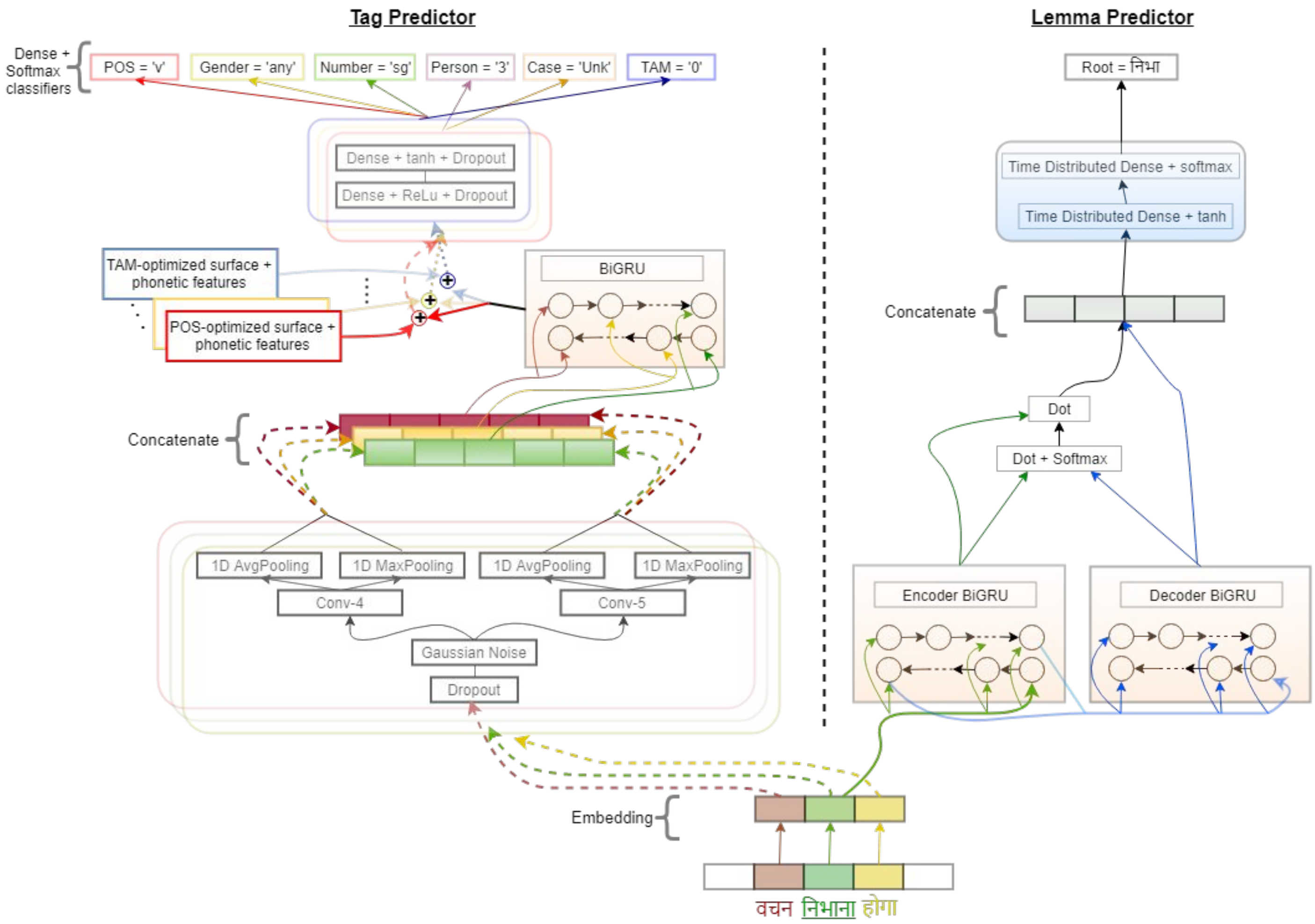
4. After **concatenation**, data passes through the **Bi-GRU layer**.
5. The result from **Bi-GRU** is obtained in a branched format where each branch is **tag-specific**. Hence, **six** such branches are formed.
6. Using **Genetic Algorithm optimization**, the best results for the six branches are passed through **two dense and dropout layers**, which finally gives the **predicted tags**.
7. The **predicted tags** are then returned.

Lemma Predictor:

1. The character embedding space is fed into the encoder. Due to the **sequential processing** of characters, it **captures the summary** of all the previous character sequences.

2. The **decoder** takes input as the hidden state of the last time step of the encoder, hence **captures the whole character sequence**, and finally generates the root.

3. The **generated root** is then returned.



Tag Predictor
Lemma Predictor
Dense + Softmax classifiers
POS = 'v'
Gender = 'any'
Number = 'sg'
Person = '3'
Case = 'Unk'
TAM = '0'
Root = निभा
Dense + tanh + Dropout
Dense + ReLu + Dropout
Time Distributed Dense + softmax
Time Distributed Dense + tanh
TAM-optimized surface + phonetic features
POS-optimized surface + phonetic features
BiGRU
Concatenate
Concatenate
Dot
Dot + Softmax
1D AvgPooling
1D MaxPooling
1D AvgPooling
1D MaxPooling
Conv-4
Conv-5
Gaussian Noise
Dropout
Encoder BiGRU
Decoder BiGRU
Embedding
वचन निभाना होगा

Chapter 5

Results and Plots

Tag\Measure	Accuracy	Precision	Recall	F1-Score
Lemma (Root)	63.88	63.88	63.88	63.88
POS	77.26	77.26	77.26	77.26
Gender	45.54	45.54	45.54	45.54
Number	52.75	52.75	52.75	52.75
Person	56.77	56.77	56.77	56.77
Case	51.21	51.21	51.21	51.21
TAM	56.41	56.41	56.41	56.41

Micro-averaged parameters for the WX model trained on Hindi Dataset and tested on the Bhojpuri Testset

Why same?

Since, we calculated **micro-averaged** precision, recall and F1 score in **multi-class problem**. So, all the values come out to be same. (Here's why)

Tag\Measure	Accuracy	Precision	Recall	F1-Score
Lemma (Root)	63.88	63.32	60.80	61.18
POS	77.26	72.38	62.68	64.65
Gender	45.54	32.84	37.76	35.54
Number	52.75	47.06	40.53	41.27
Person	56.77	24.18	49.27	26.42
Case	51.21	44.42	43.75	44.0
TAM	56.41	28.59	20.83	22.90

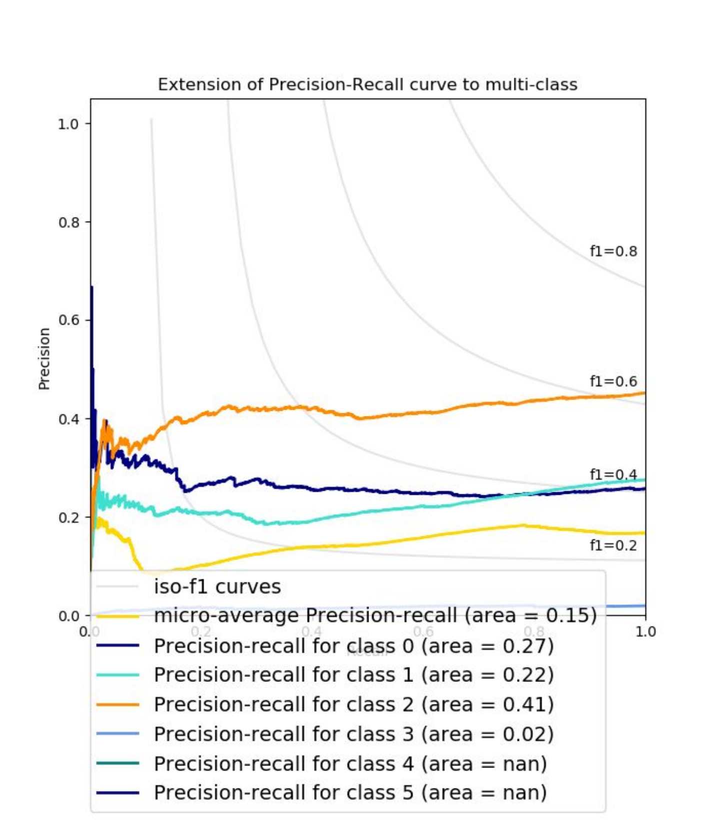
Macro-averaged parameters for the WX model trained on Hindi Dataset and tested on the Bhojpuri Testset

Why Precision is low?

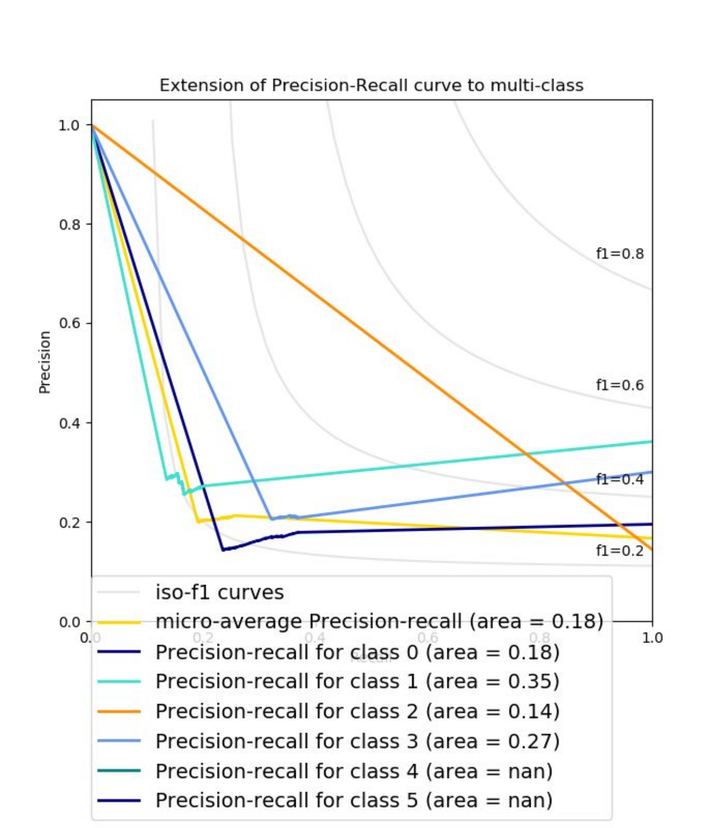
A lot of **False Positives** were generated.

Why Recall is low?

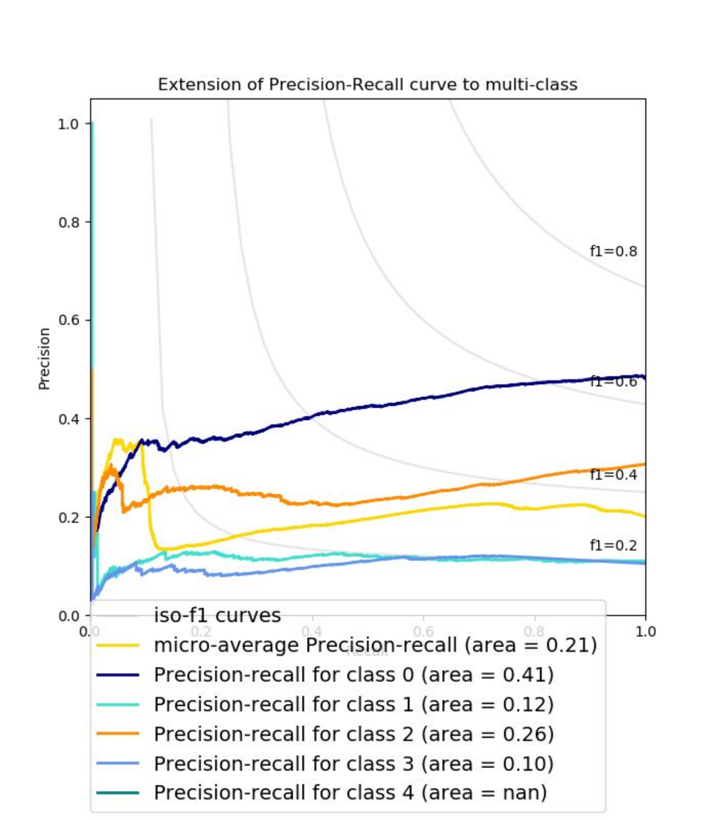
A lot of **False Negatives** were generated.



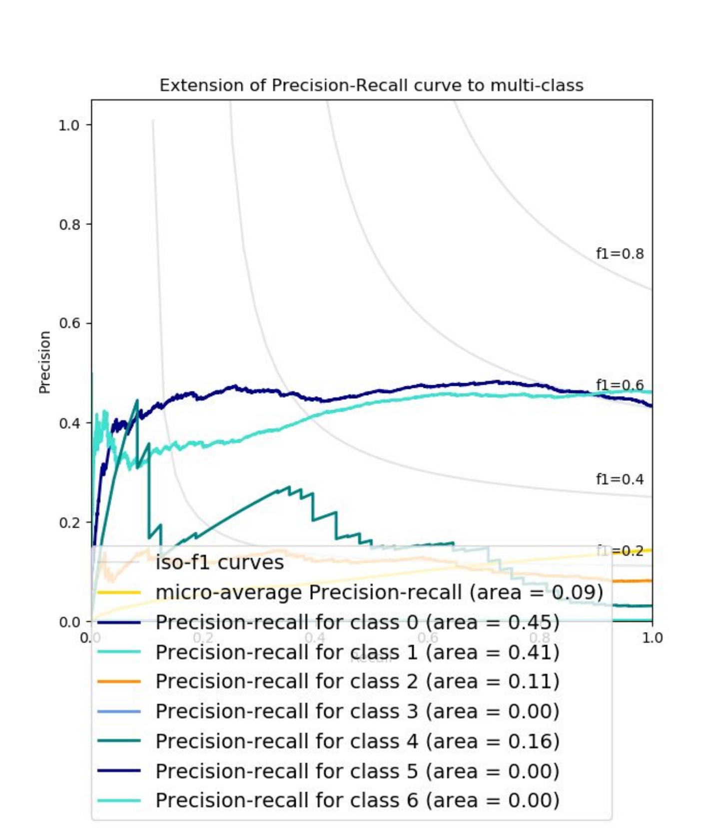
Extension of **Precision-Recall curve** to **multi-class** for feature **Case**



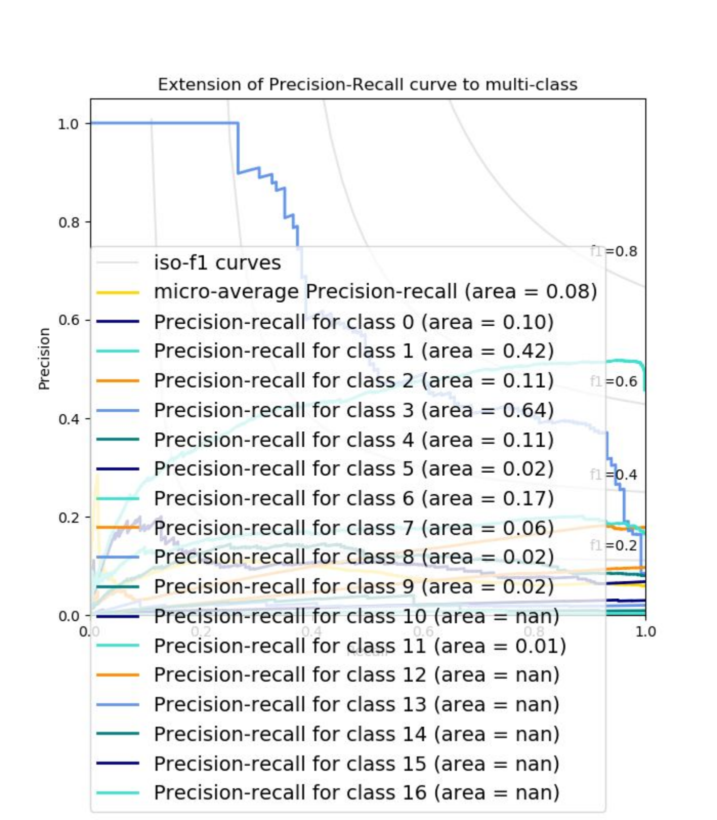
Extension of **Precision-Recall curve** to **multi-class** for feature **Gender**



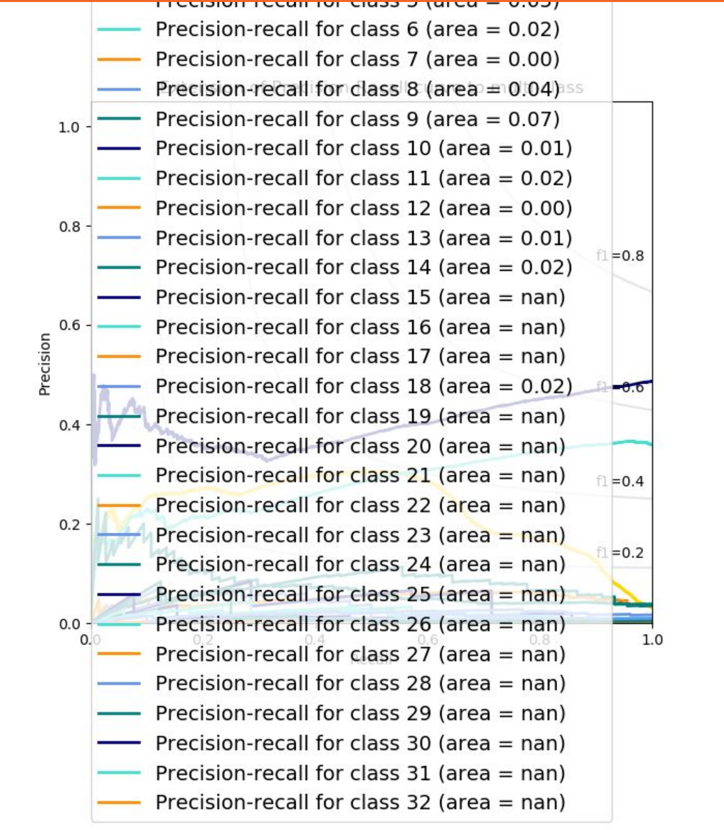
Extension of **Precision-Recall curve** to **multi-class** for feature **Number**



Extension of **Precision-Recall curve** to **multi-class** for feature **Person**



Extension of **Precision-Recall curve** to **multi-class** for feature **POS**



Extension of **Precision-Recall curve** to **multi-class** for feature **TAM**

Chapter 7

Conclusion and Future Work

How is this model helpful?

We conclude that:

Using **unsupervised domain adaptation** in **zero-shot learning condition** on the MT-DMA, we can morphologically analyze any language, just like we did for Bhojpuri, even if there is not enough dataset available for the language for training a model.

This would make morphological analysis possible for **low resource languages**, such as Maithili, Sinhala, etc.

Future Work:

- ➔ In future, if enough well-formatted data is made available, then there is scope to train the model on Bhojpuri training data, which will definitely improve the results.
- ➔ We ran only 5 epochs per training dataset due to unavailability of better computational resources. So, in future, we can increase the number of epochs which will definitely enhance the model.
- ➔ We look forward to see developers build applications like translator for Bhojpuri as well, after the enough training data is made available.

Chapter 8

References

- https://arxiv.org/abs/1811.08619 (Multi Task Deep Morphological Analyzer: Context Aware Joint Morphological Tagging and Lemma Prediction)
- https://www.researchgate.net/publication/259440611_An_empirical_analysis_of_dropout_in_piecewise_linear_networks (An empirical analysis of dropout in piecewise linear networks)
- https://core.ac.uk/download/pdf/162017899.pdf (Detection of Gaussian Noise and Its Level using Deep Convolutional Neural Network)
- https://medium.com/analytics-vidhya/bi-directional-rnn-basics-of-lstm-and-gru-e114aa4779bb (Bi-directional RNN & Basics of LSTM and GRU)
- https://arxiv.org/ftp/arxiv/papers/1407/1407.2989.pdf (HMM based POS tagger for Sinhala language)


Thanks!