
SE 801: Final Presentation

Bengali Word Embedding and Next Word Prediction

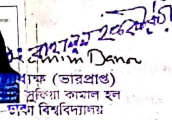
Presented by
Ishita Sur Apan
Exam roll: 0932

Supervised By
Dr. Mohammad Shoyaib
Professor

Institute of Information Technology
University of Dhaka



05-04-2021



(পরীক্ষা নিয়ন্ত্রক অফিস কর্তৃক পূরণীয়)

પ્રશ્નના વિષય : ગ્રહે અથવા ચંદ્રનીચરિ:

ଅବେଶ ଗଳ୍ପ

રેશનિંગ સુર જાપત

ISHITA SUR APAN

माहान्न नाम : सुदि रात्री पक्ष

SE 701: Internship

81 /

21

91

পত্রিকা: সারথ হইবার তারিখ: 28 JAN 2021
(পত্রিকা: সারথের অফিস কর্তৃক প্রকাশিত)

५७: वा. ७७७ २८८८८८८८

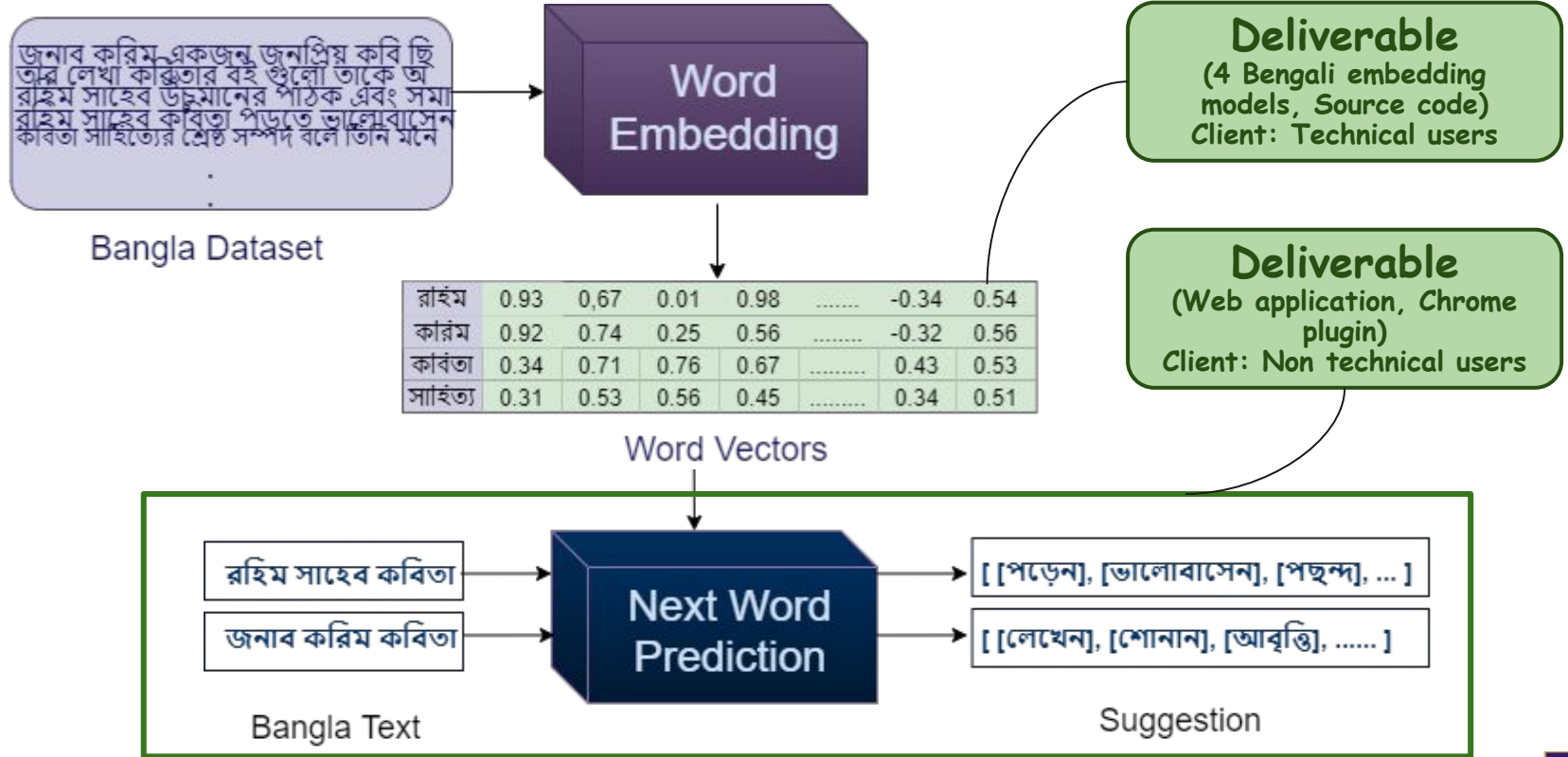
বিঃদ্রঃ ১। প্রডোজট মহোদয় পরীক্ষার্থীর ফটো গীন বোহদবুল্ল সত্যায়িত করিবেন।

২। প্রভোস্ট মহোদয় কর্তৃক নিরীক্ষাকৃত ওখ্যাবলী চূড়ান্ত বনিয়া বিবেচিত হইবে।

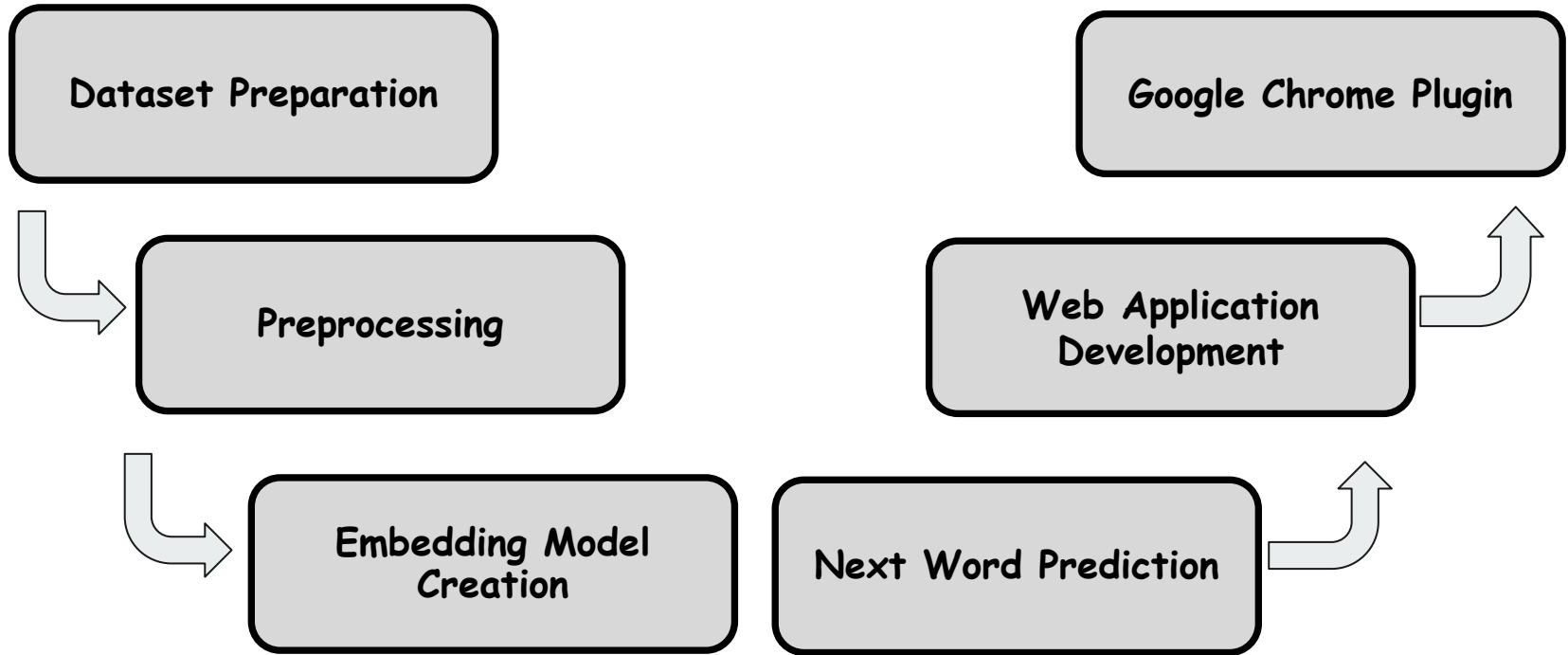
৩। পরীক্ষার পত্র/কোর্স কোন কাটাকাটি বা যবামজা চলিবে না। (অপর পৃষ্ঠা প্রদর্শন)

Bengali Word Embedding & Next Word Prediction

3



Methodology (At a glance)



Dataset

- **Datasource**

- News portals
 - Prothom Alo, Kaler Kantho, Ittefak
- Bengali websites
 - Banglapedia, Bengali wikipedia
- Bengali bolg
 - Sachalayaton
- Bengali Books

- **About dataset**

- 10.2 million sentences
- Total size 5 GB



Banglapedia
National Encyclopedia of Bangladesh



উইকিপিডিয়া
একটি মুক্ত বিশ্বকোষ



প্রথম আলো



Bangla Books PDF
Bangla Ebook & Study Support

Preprocessing

Remove unwanted characters and digits

Remove English words

Remove punctuation

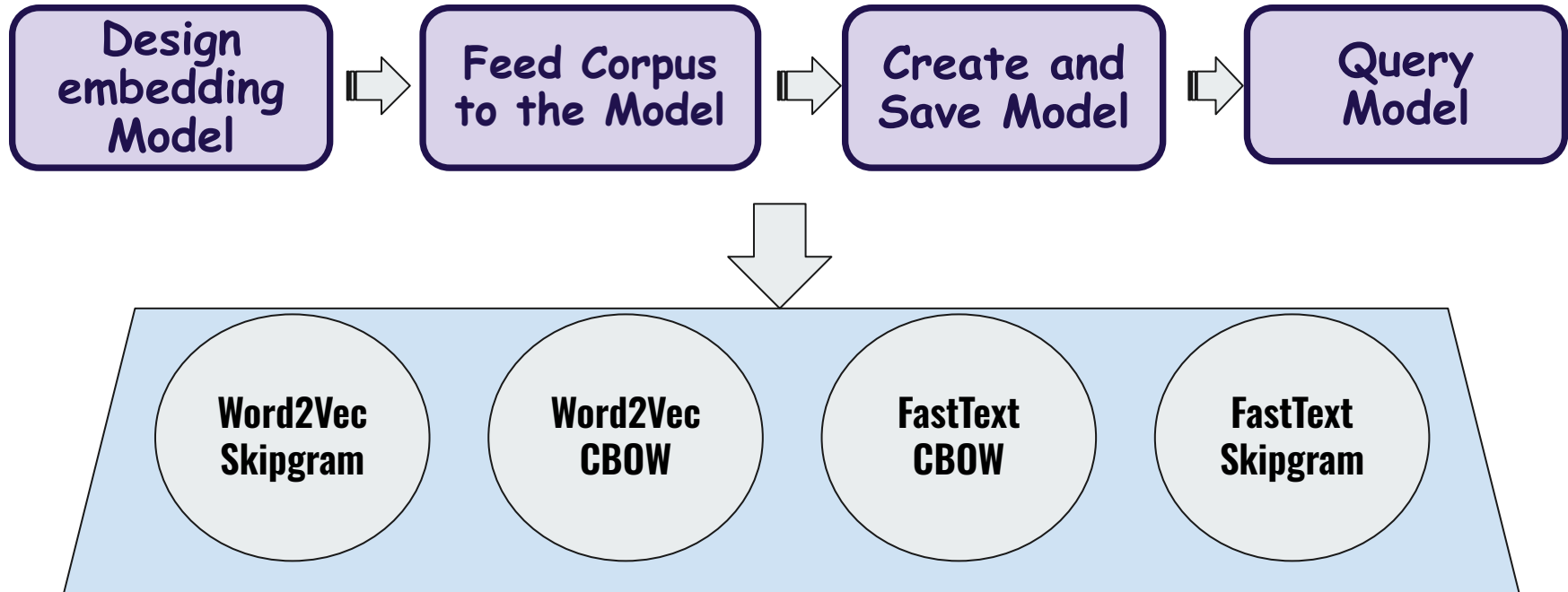
Extract sentences

Tokenize words

Save preprocessed files in pkl

Different approaches
of preprocessing is
required to find out
**Best embedding
model**

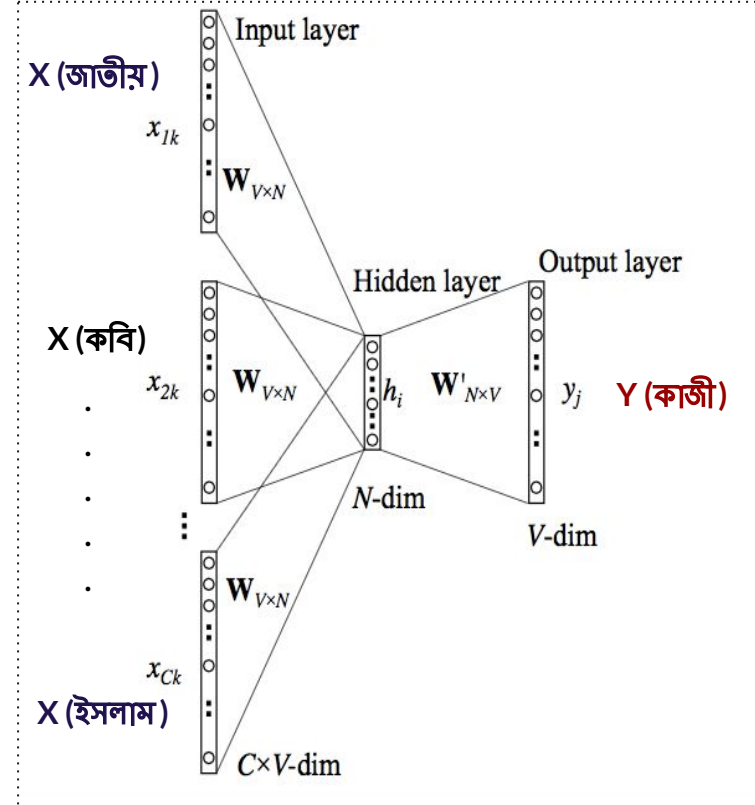
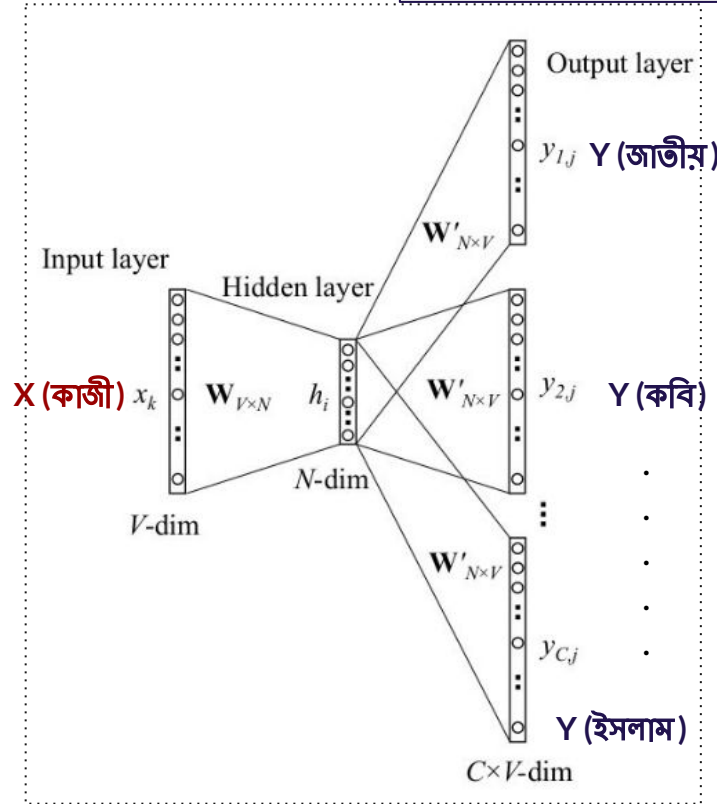
Word Embeddings



Sequence: [জাতীয় কবি কাজী নজরুল ইসলাম]
 Context window size: 5
 Target word: কাজী

SkipGram

CBOW



→ Word2Vec

- ◆ Treats each word as an atomic entity
- ◆ Generates a vector for each word
- ◆ Cannot produce vector for out-of-vocabulary words

[জাতীয় কবি কাজী নজরুল ইসলাম]

["নজরুল"]

নজরুল → [-0.44, 0.957, ..., -0.013, 0.64]

জারুল → **Error**

→ FastText

- ◆ Extension of Word2Vec
- ◆ Treats a word composed of its character **ngrams**
- ◆ Mean of target word vector and its component ngram vectors

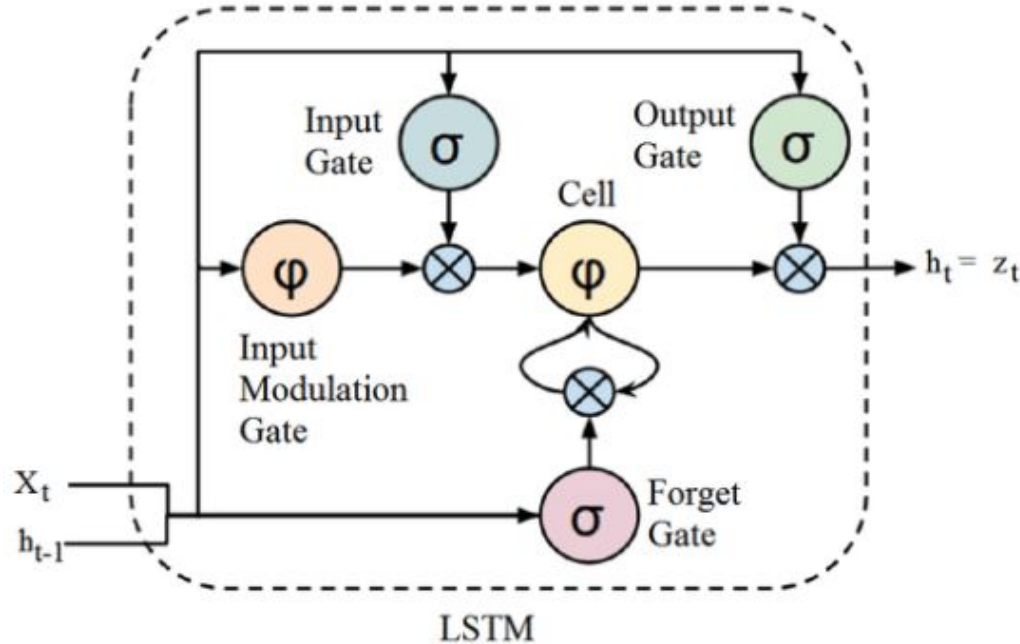
[জাতীয় কবি কাজী নজরুল ইসলাম]

["<নজ", "জর ুল", "র ুল", "ুল>", "নজরুল"]

নজরুল → [-0.57, 0.9, ..., -0.001, 0.934]

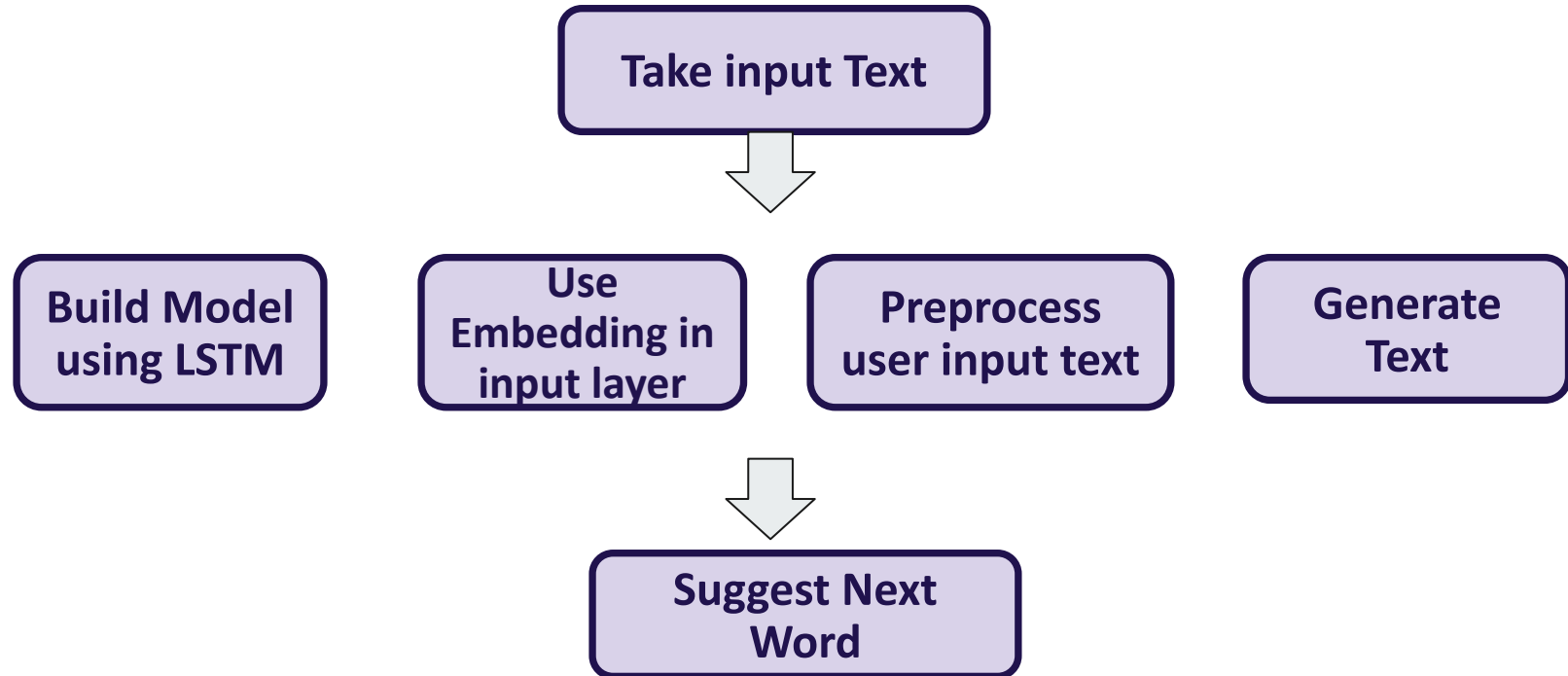
জারুল → [-0.37, 0.84, ..., -0.012, 0.734]

Long Short Term Memory

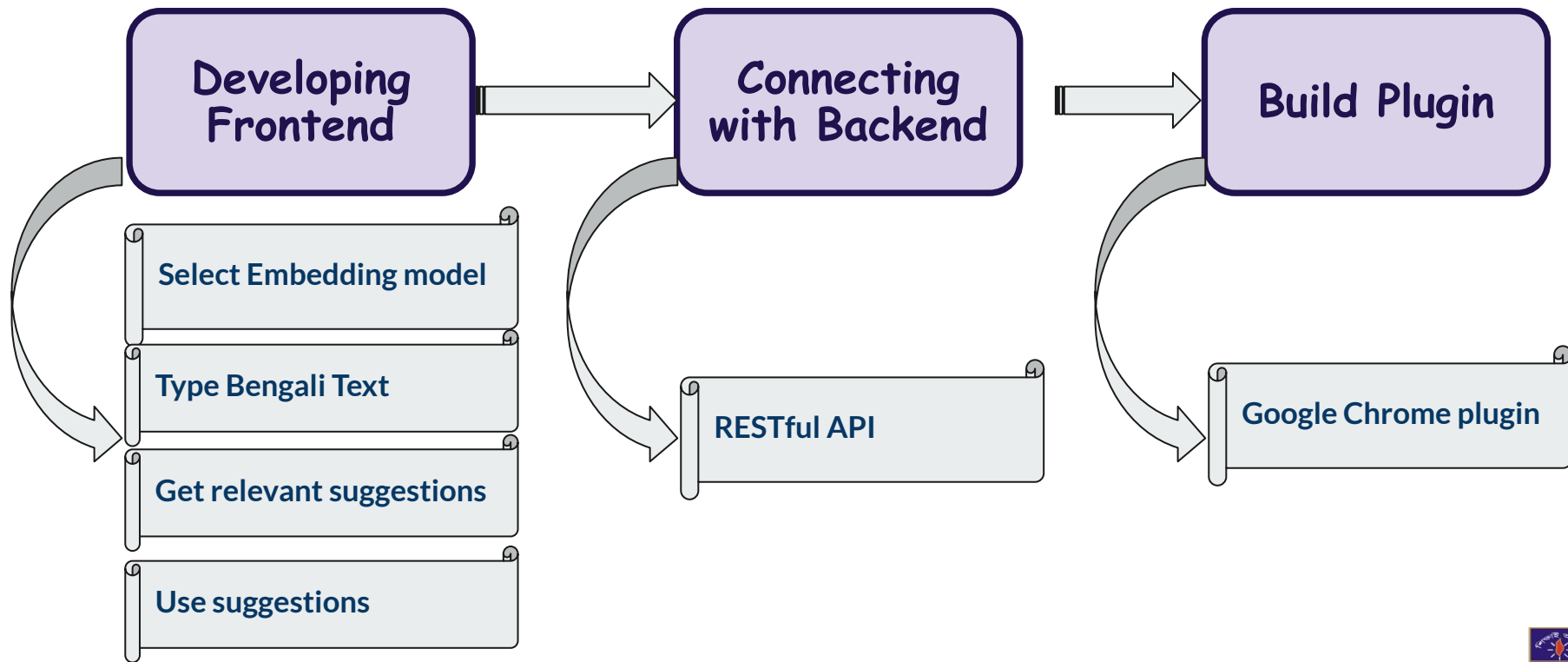


- LSTM
 - Special kind of RNN
 - Input gate, output gate
 - Cell state, forget gate
- Activation Function: SoftMax
- Loss Function: Cross Entropy
- Adam Optimizer
- Metrics: Accuracy

Next Word Prediction



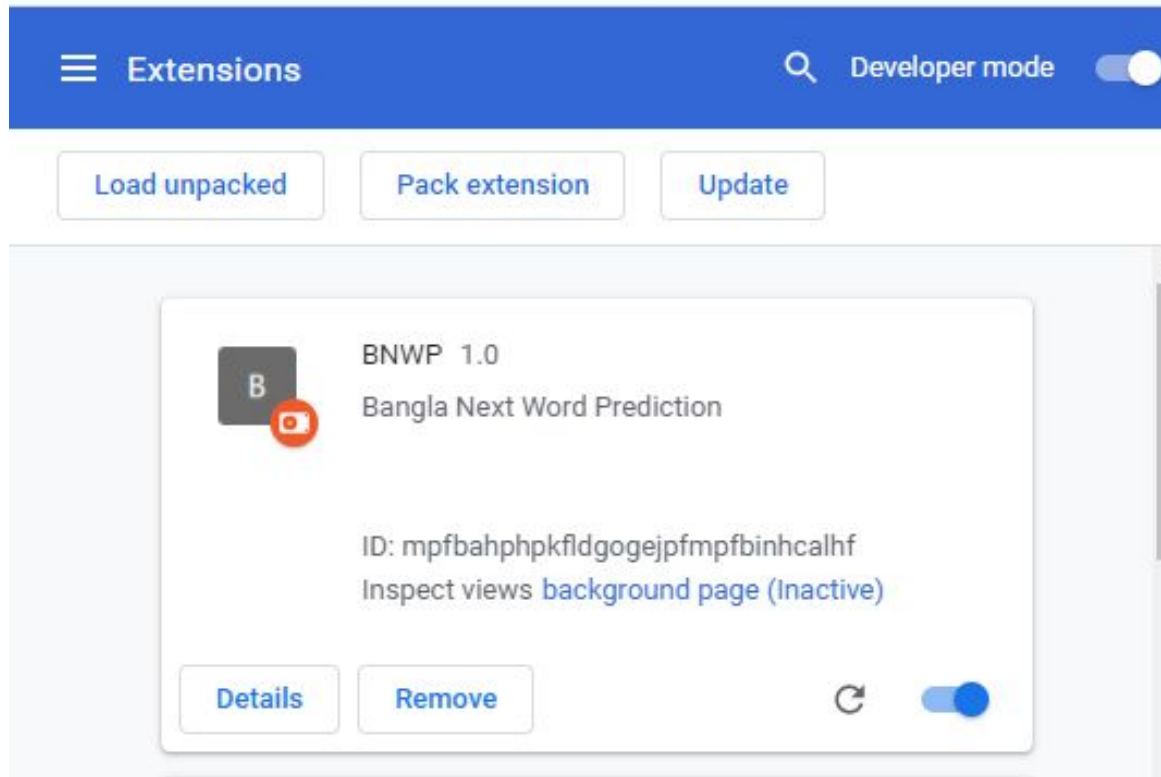
Web Application and Plugin



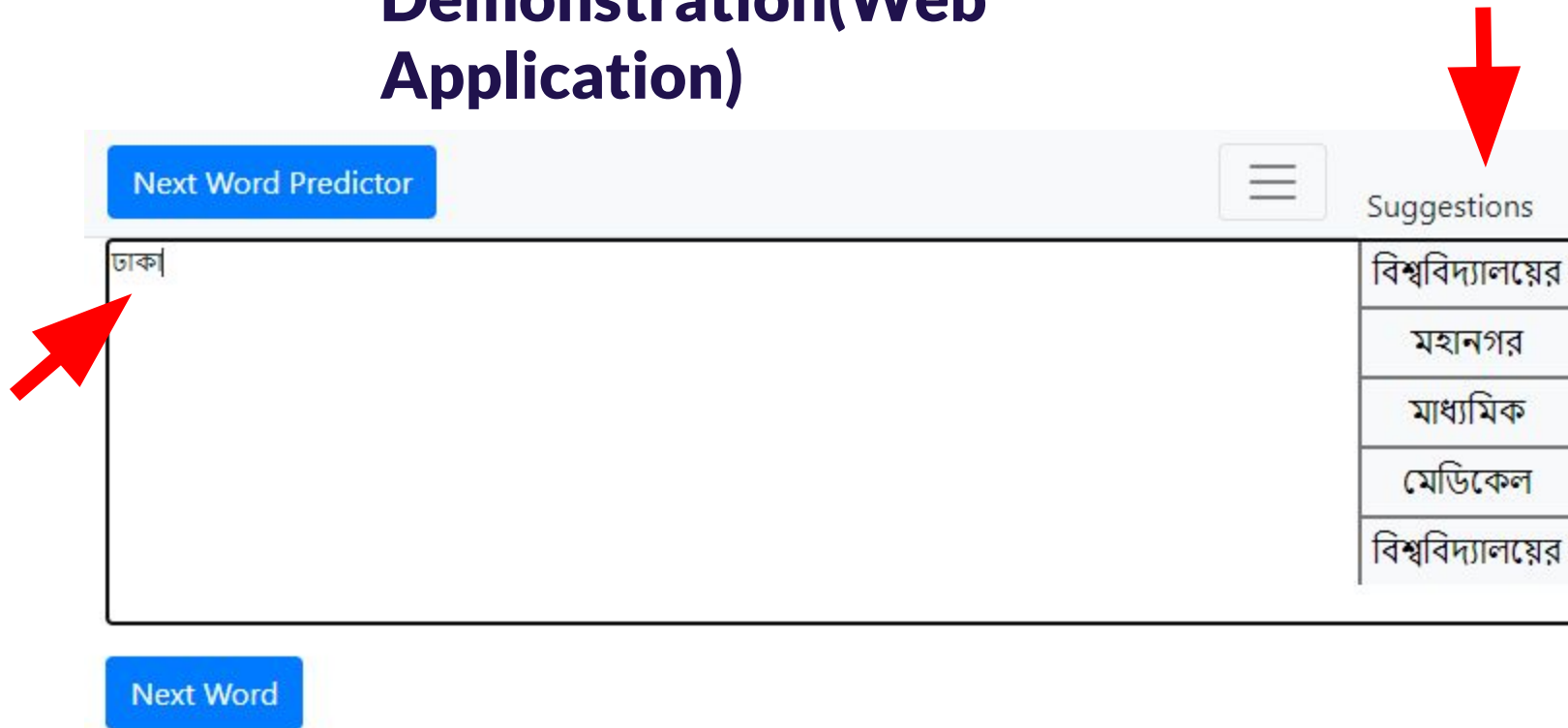
Tools and Technology

- **Backend→ Python 3**
 - **Embedding**
 - Word2Vec, FastText, CBOW, Skipgram
 - Gensim Library
 - **Next Word Prediction**
 - Algorithm: LSTM
 - Keras Library
- **Frontend→ Django**
 - **Web application**
 - Html, Css, Bootstrap, Javascript
 - **Web API**
 - Django RESTful framework
- **Plugin**
 - **Google Chrome Extension**

Demonstration(Chrome Extension)



Demonstration(Web Application)



Next Word Predictor

☰ Suggestions

তাক

- বিশ্ববিদ্যালয়ের
- মহানগর
- মাধ্যমিক
- মেডিকেল
- বিশ্ববিদ্যালয়ের

Next Word

Demonstration(Web Application)

Next Word Predictor

☰

Suggestions

ঢাকা বিশ্ববিদ্যালয়ের	শিক্ষক
	মুক্তিযোদ্ধা
	চাৰি
	উইমেন
	সাবেক

Next Word

Demonstration(Web Application)

Next Word Predictor

☰

Suggestions

ঢাকা বিশ্ববিদ্যালয়ের শিক্ষক

অধ্যাপক

মিজানুর

তপন

শিক্ষক

নির্বাচিত

Next Word

Demonstration(Embedding)

jupyter embedding Last Checkpoint: 15 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Save + Undo Copy Paste Up Down Run Stop Restart Code

```
In [59]: word_equation(model1, positive=['বাবা', 'মেয়ে'], negative=['ছেলে'])  
word_equation(model2, positive=['বাবা', 'মেয়ে'], negative=['ছেলে'])  
word_equation(model3, positive=['বাবা', 'মেয়ে'], negative=['ছেলে'])  
word_equation(model4, positive=['বাবা', 'মেয়ে'], negative=['ছেলে'])
```

```
[('মা', 0.8470655679702759)]  
[('মা', 0.841377317905426)]  
[('মা', 0.8503274321556091)]  
[('মা', 0.8910645246505737)]
```

বাবা + মেয়ে - ছেলে
= মা

Demonstration(Embedding)

jupyter embedding Last Checkpoint: 13 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help



Code



```
In [58]: word_equation(model1, positive=['রাজা', 'মেয়ে'], negative=['ছেলে'])
word_equation(model2, positive=['রাজা', 'মেয়ে'], negative=['ছেলে'])
word_equation(model3, positive=['রাজা', 'মেয়ে'], negative=['ছেলে'])
word_equation(model4, positive=['রাজা', 'মেয়ে'], negative=['ছেলে'])

[('রাণী', 0.6641949415206909)]
[('রানি', 0.6961914300918579)]
[('রাজারাণী', 0.8331283330917358)]
[('রানি', 0.8929707407951355)]
```

রাজা - ছেলে + মেয়ে
= রানি

Demonstration(Embedding)

```
In [104]: word_equation(model1, positive=['লেখক', 'নারী'], negative=[])  
word_equation(model2, positive=['লেখিকা', 'পুরুষ'], negative=['নারী'])  
word_equation(model1, positive=['ঢাকা'], negative=['রাজধানী'])  
word_equation(model1, positive=['রবীন্দ্রনাথ', 'ভ্রাতা'], negative=[])  
word_equation(model1, positive=['শেখ', 'হাসিনা'], negative=[])  
  
[( 'লেখিকা', 0.724480390548706)]  
[( 'ঔপন্যাসিক', 0.618572473526001)]  
[( 'জাহাঙ্গীরনগর', 0.46925368905067444)]  
[( 'দেবেন্দ্রনাথ', 0.7047825455665588)]  
[( 'বঙ্গবন্ধুকন্যা', 0.6981199979782104)]
```

Demonstration(Embedding)

```
: find_similar_words(model1, 'ঢাকা')
   find_similar_words(model2, 'নজরুল')
   find_similar_words(model2, 'কলম')
   find_similar_words(model4, 'সোমবার')
```

```
[('রাজশাহী', 0.8118138909339905), ('চট্টগ্রাম', 0.7809731364250183), ('কুমিল্লা', 0.760356605052948)]
[('কাজী', 0.8067245483398438), ('রফিকুল', 0.7441709637641907), ('ইসলামগীতিকারঃ', 0.7411827445030212)]
[('কলমও', 0.7381693124771118), ('পেন্সিল', 0.6929911971092224), ('কলমের', 0.6923819780349731)]
[('বুধবার', 0.995028018951416), ('মঙ্গলবার', 0.9946527481079102), ('বৃহস্পতিবার', 0.9931206703186035)]
```

```
: find_similarity(model1, 'বিশ্ববিদ্যালয়', 'স্কুল')
   find_similarity(model1, 'বিশ্ববিদ্যালয়', 'কলেজ')
   find_similarity(model1, 'কলেজ', 'স্কুল')
   find_similarity(model2, 'বাবা', 'মা')
```

0.50

0.67

0.74

0.81



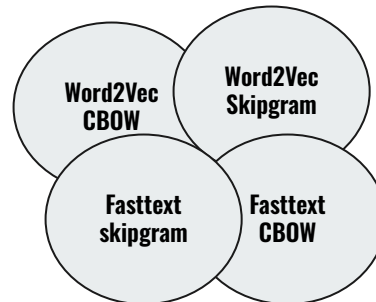
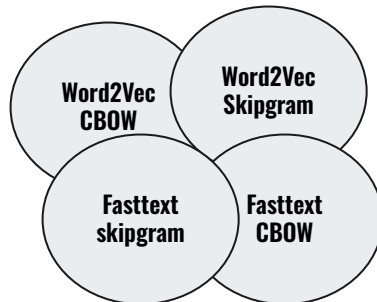
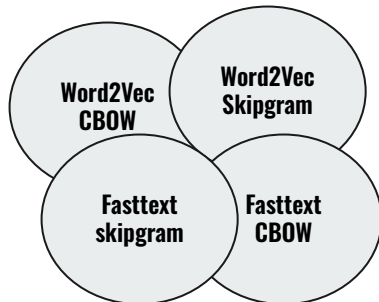
Word Embeddings

Different Approaches of Preprocessing

- Without data cleaning
- With data cleaning
- Extract sentences regarding “ | ”
- Extract sentences regarding “ | ”, “?” and “!”

Different Approaches of Algorithms

- Word2Vec CBOW
- Word2Vec Skipgram
- FastText CBOW
- FastText Skipgram



Methodology (Preprocessing)

Remove unwanted characters and digits

Remove English words

Remove punctuation

Extract sentences

Tokenize words

Save preprocessed files in pkl

Why not
Stemming?



Word Embeddings

Different Approaches of Preprocessing

- Without data cleaning
- With data cleaning
- Extract sentences regarding "I", "am"
- Extract sentences regarding "I", "am", "are"

Different Approaches of Algorithms

- Word2Vec CBOW
- Word2Vec Skipgram
- FastText CBOW
- FastText Skipgram

Which Approach is better?

Preprocessing

Trial and Error

Depends on the task to be performed

**In this Project, the task is
Next Word Prediction**

Word2Vec
CBOW

Word2Vec
Skipgram

Word2Vec

Skipgram

Word2Vec

Skipgram

Fasttext
skipgram

Fasttext
CBOW

Word2Vec vs FastText

The key difference between FastText and Word2Vec is the use of **n-grams**.

- Word2Vec → Google, FastText → Facebook
- Word2Vec is Faster to train
- FastText works better for rare words
- Fasttext can create word vectors for out-of-vocabulary words