# TABLE OF CONTENTS

# MOTIVATION

- Humans can determine sentiments and the logical flow in texts
- Inability to process large amounts of textual data
- A model that can understand logical flow can benefit multiple areas e.g. identifying fake news

# OBJECTIVE

- To predict if a given hypothesis is related to its premise
- Obtain the most accurate model

Contradiction

Entailment

Neutral

# HYPOTHESIS

## Hypothesis 1: Attention Models like BERT will outperform bidirectional LSTM models

- Bidirectional LSTM models can account for either left-to-right or right-to-left context
- Attention Models like BERT can account for both simultaneously

# HYPOTHESIS

## Hypothesis 2: Split Input Model will perform better than a Concatenated Input Model

Method 1:
Concatenated Input

Premise
Hypothesis
Concatenate → Embedding Layer

Method 2:
Split Input

Premise → Embedding Layer
Hypothesis → Embedding Layer
Concatenate

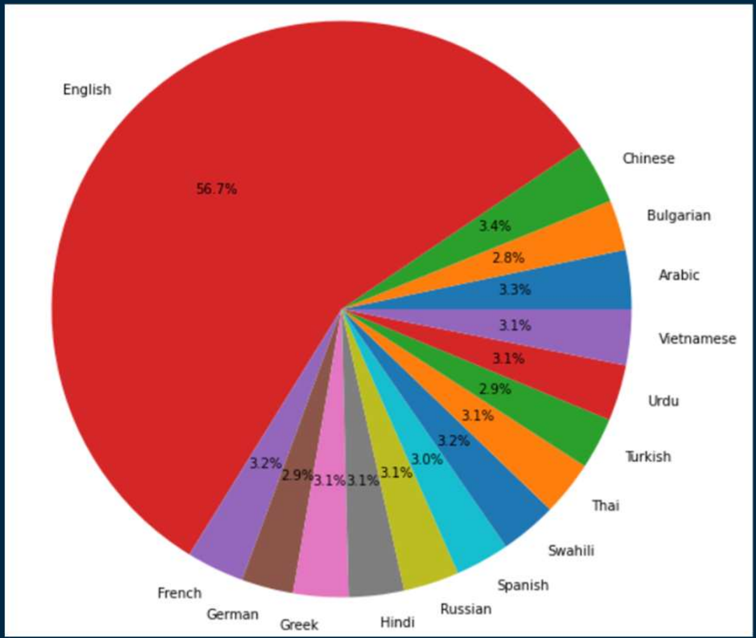
Method 1 produces sentence embeddings that are dependent on the other sentence.

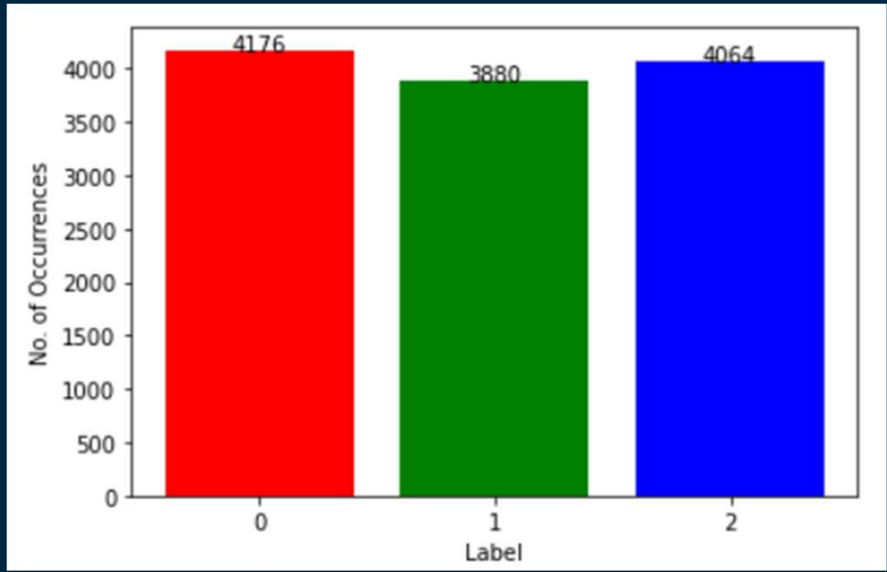
# DATA

- 12120 rows
- 6 columns
- Multiple Languages

| Column | Description |
|---|---|
| ID | A unique identifier for the row. |
| Premise | A starter text, used as context for the hypothesis. |
| Hypothesis | A follow up text. |
| Lang_abv | Abbreviation for the language used in the text. |
| Language | Language used in the text. |
| Label | Classification of the relationship between the Premise and Hypothesis.<br>(0 for entailment, 1 for neutral, 2 for contradiction) |

# EXPLORATORY DATA ANALYSIS




56.7% of rows in English Language

Each label is well represented.

# DATA CLEANING

- No missing observations
- Only kept rows in English Language
- 6870 rows left

# OVERVIEW OF MODELS

1

Generic BiLSTM

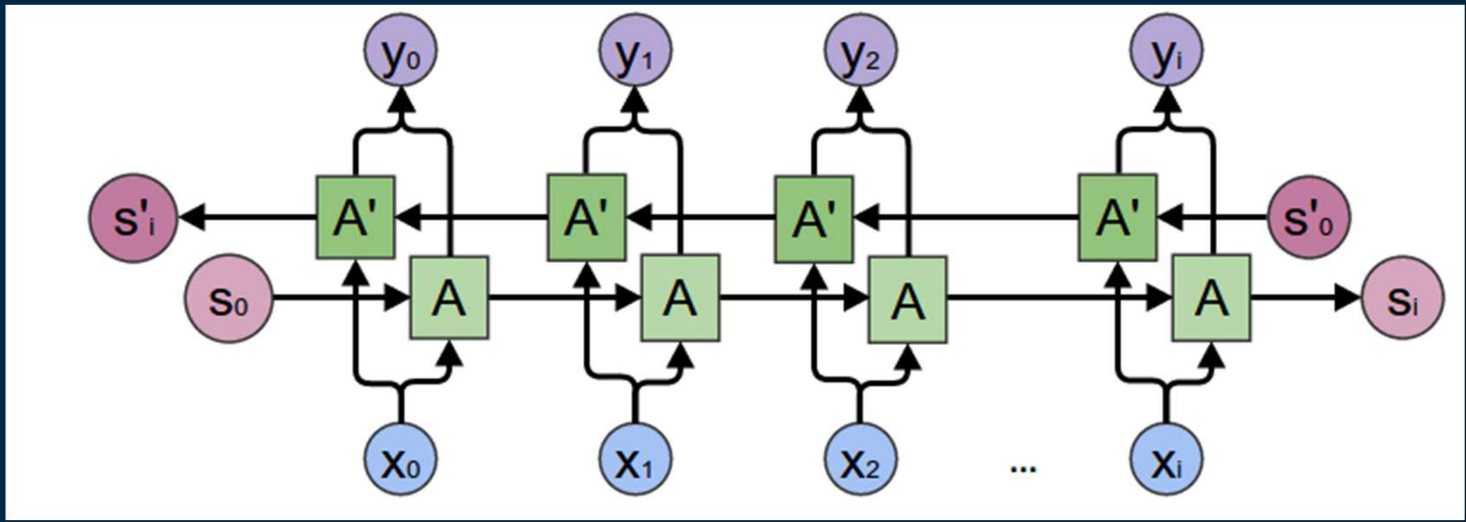
2

ELMo

3

BERT

# Generic BiLSTM (Architecture)



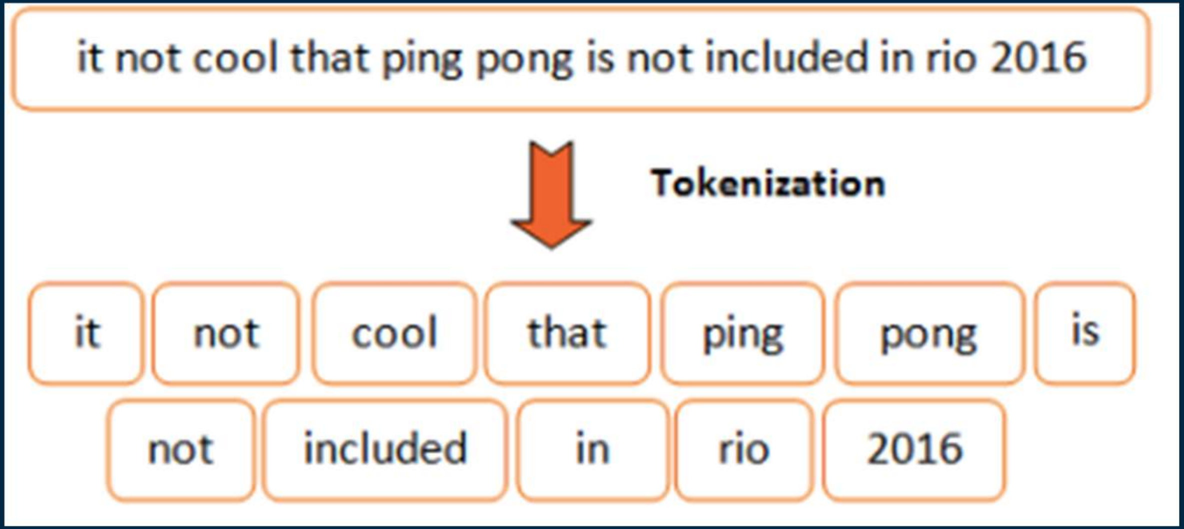

## Flow

Input X > Layer A & Layer A’ > Output Y

# Generic BiLSTM (Tokenization)

**Model**

Tensorflow Keras Tokenizer



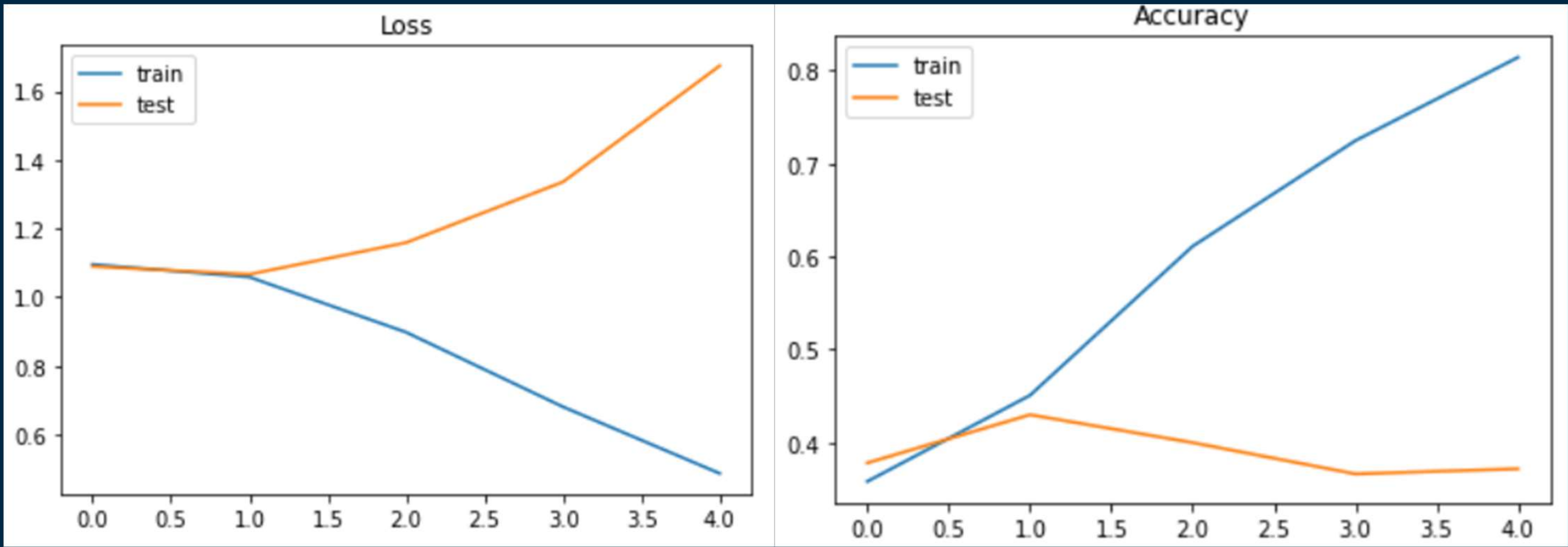
# Generic BiLSTM (Embedding)

**Model**

Embedding Dimension = 32

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 250, 32)           1600000
_________________________________________________________________
bidirectional (Bidirectional (None, 128)               49664
_________________________________________________________________
dense (Dense)                (None, 3)                 387
=================================================================
Total params: 1,650,051
Trainable params: 1,650,051
Non-trainable params: 0
_________________________________________________________________
```

# Generic BiLSTM (Results)

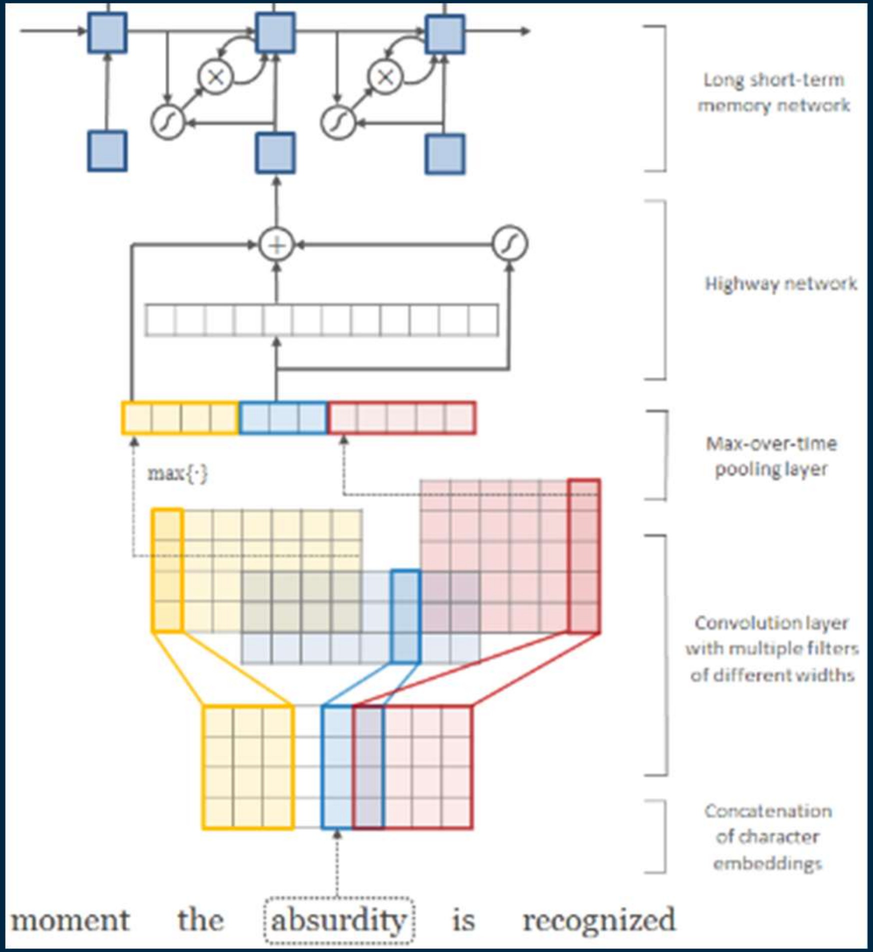



Loss & Model Accuracy

# ELMo (Architecture)

## Contextualized Word Embeddings

*Date* A measurement of time or a romantic engagement?



# ELMo (Outputs)

| Model Outputs | Output Description | Output Shape |
|---|---|---|
| “ELMo” | Word level embeddings | (batch_size, max_length, 1024) |
| “Default” | Sentence Level Embeddings | (batch_size, 1024) |

## Word Level Embeddings

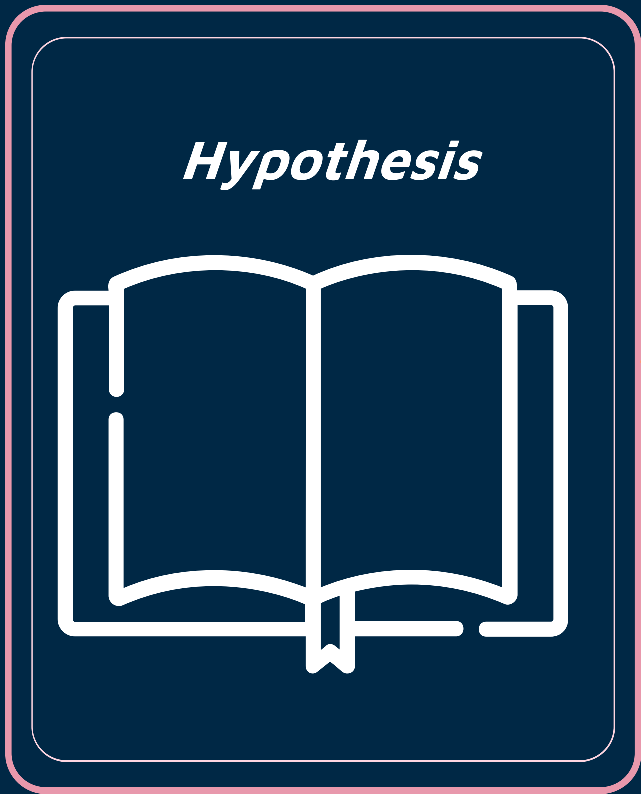
- **1024** dimension embedding vector for each word
- **max_length** as the length of longest sentence in the batch

## Sentence Level Embeddings

- **1024** dimension embedding vector for each sentence
- Vector generated as a **fixed-mean pooling** of the vectors of each sentence

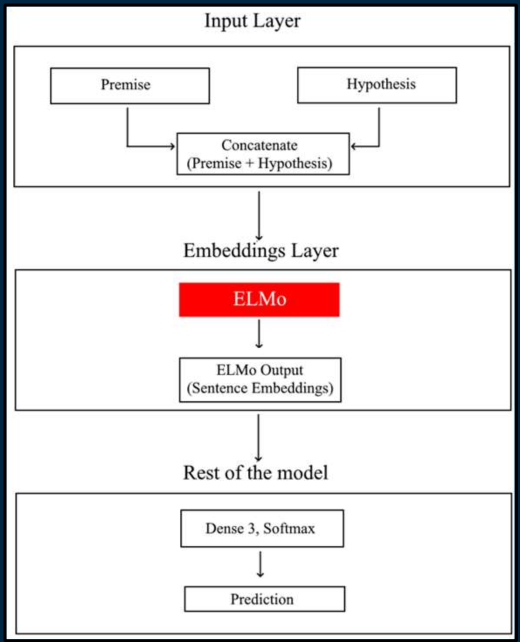
# ELMo (Test Hypothesis 2)

*A split input model will perform better than a concatenated input model.*

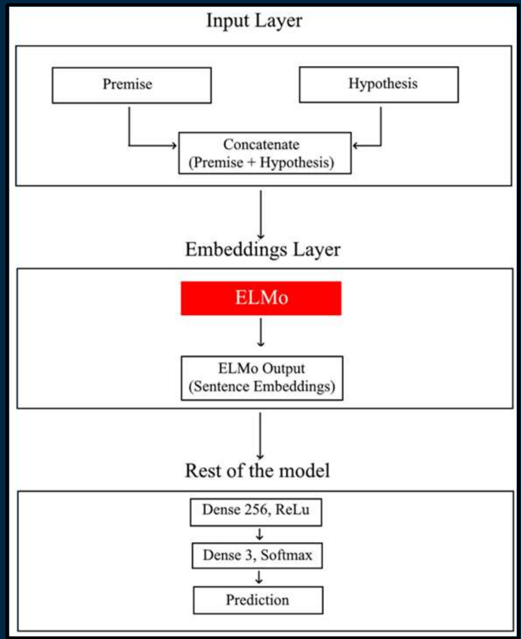


# ELMo (Concatenated Inputs)

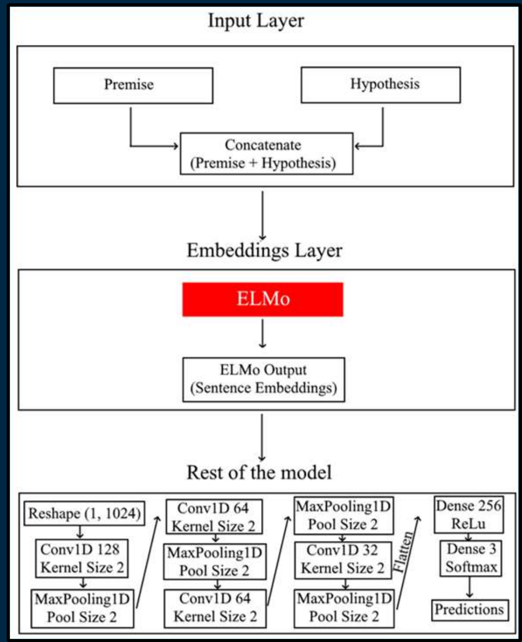
## Direct



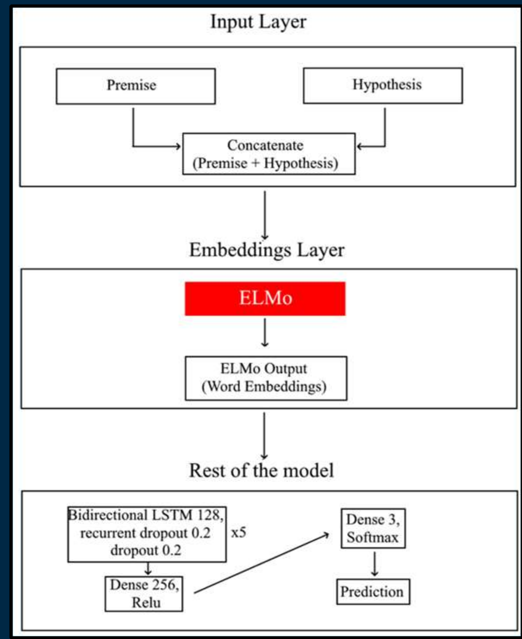

## Dense(256)




## 1D Conv

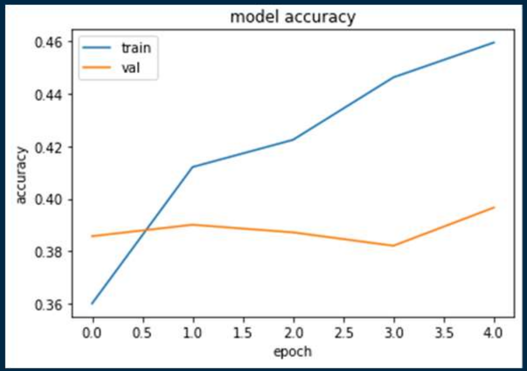



## BiLSTM



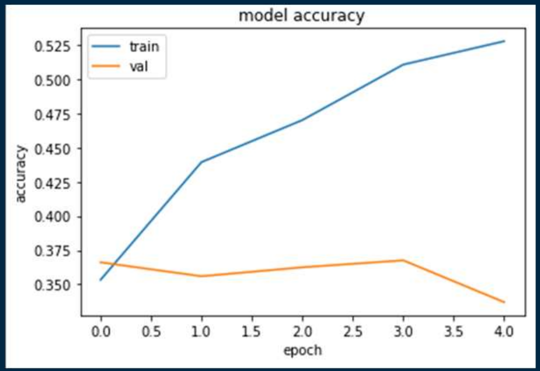
# ELMo (Concatenated Inputs)

**Direct**



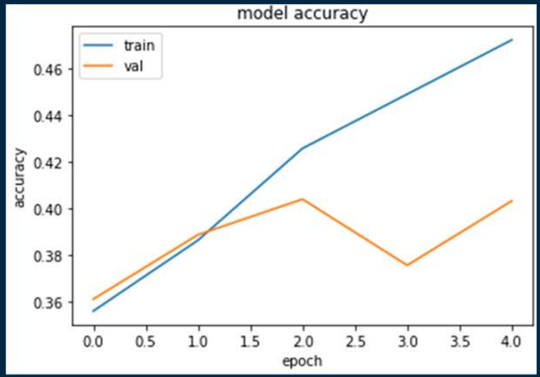

~ 39% validation accuracy

**Dense(256)**



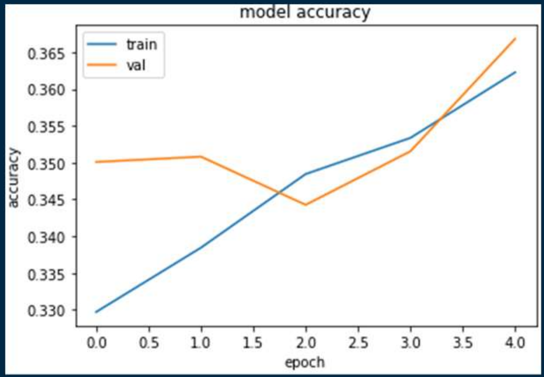

~ 37% validation accuracy

**1D Conv**




~ 40% validation accuracy

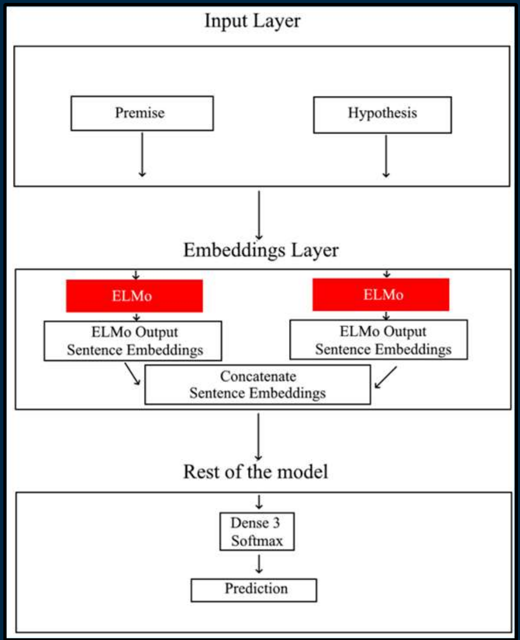
**BiLSTM**



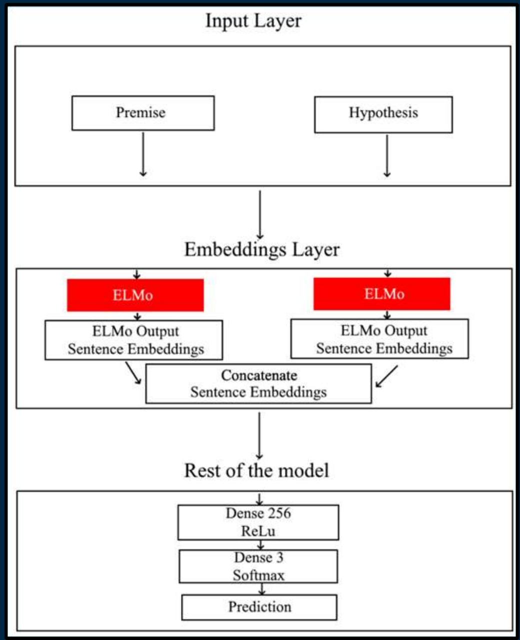

~ 37% validation accuracy

# ELMo (Split Inputs)

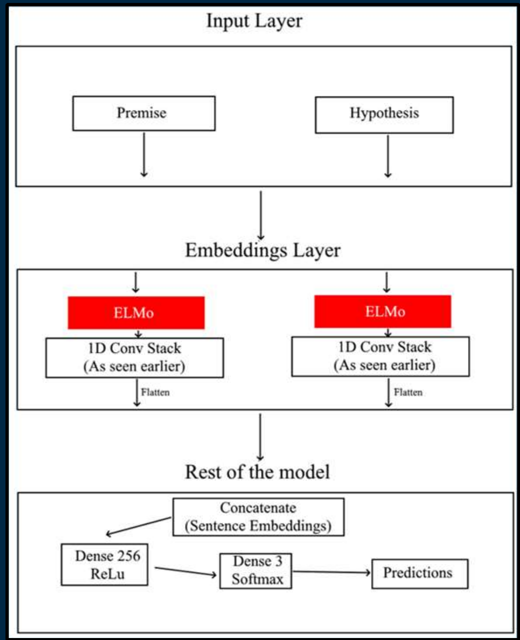
**Direct**



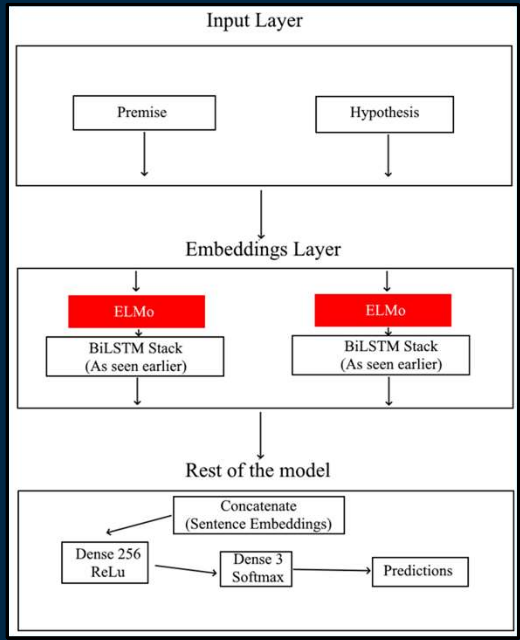

**Dense(256)**




**1D Conv**

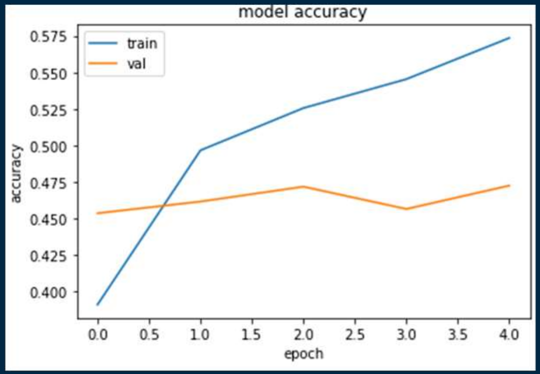



**BiLSTM**



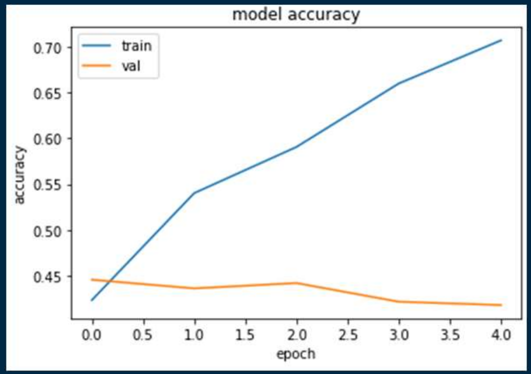
# ELMo (Split Inputs)

**Direct**



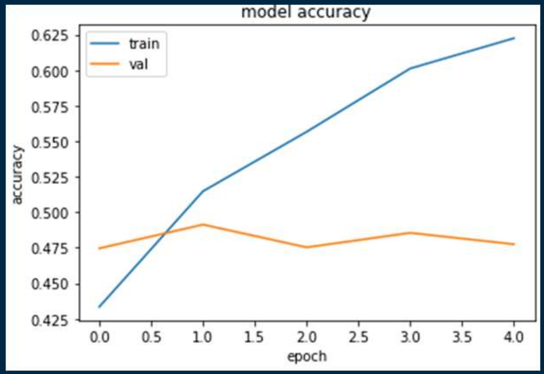

~ 46% validation accuracy

**Dense(256)**



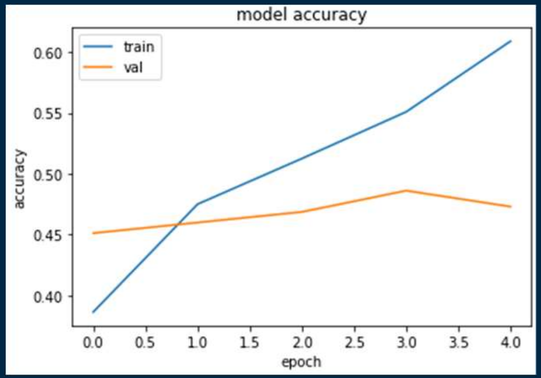

~ 45% validation accuracy

**1D Conv**




~ 49% validation accuracy

**BiLSTM**




~ 48% validation accuracy

# BERT (Description)

- Stands for Bidirectional Encoder Representations using Transformers
- Analyse the words in the sequence as a whole rather than individual
- For this project, we will be using the xlm-RoBERTa-large model

# BERT (Tokenization and Padding)

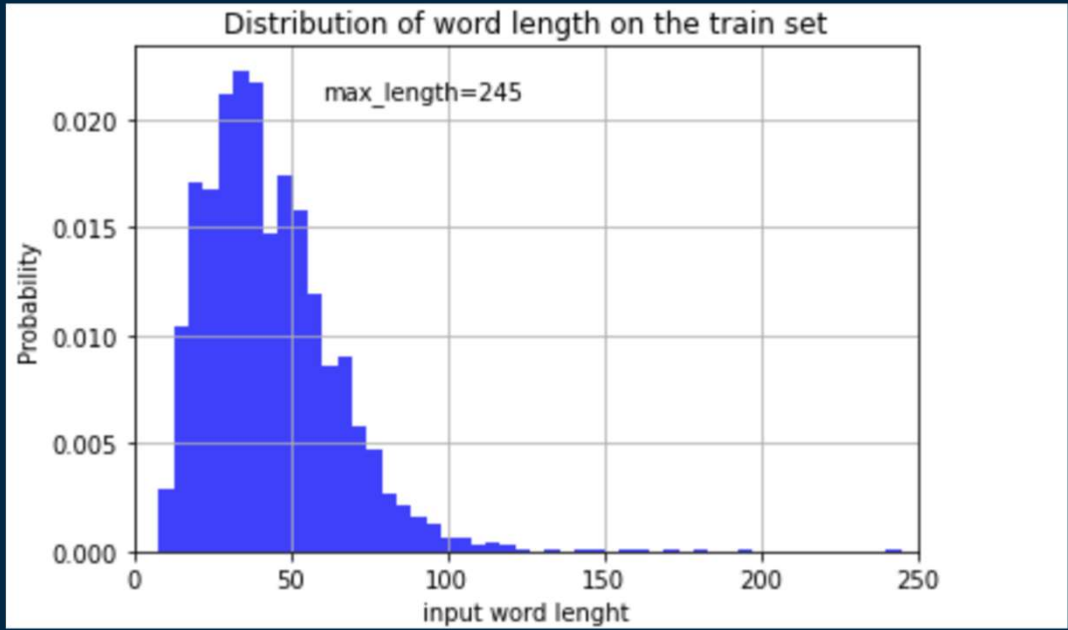
Concatenation of sentences

Labelling and Separation of hypothesis and premise

Padding to ensure equal length

# BERT (Model Building)

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_word_ids (InputLayer)     [(None, 245)]        0           []

input_mask (InputLayer)         [(None, 245)]        0           []

tfxlm_roberta_model (TFXLMRobe   TFBaseModelOutputWi  559890432   ['input_word_ids[0][0]',
rtaModel)                       thPoolingAndCrossAt               'input_mask[0][0]']
                                tentions(last_hidde
                                n_state=(None, 245,
                                 1024),
                                 pooler_output=(Non
                                e, 1024),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

global_average_pooling1d (Glob   (None, 1024)        0           ['tfxlm_roberta_model[0][0]']
alAveragePooling1D)

dense (Dense)                   (None, 3)            3075        ['global_average_pooling1d[0][0]'
                                                                 ]

==================================================================================================
Total params: 559,893,507
Trainable params: 559,893,507
Non-trainable params: 0
```



# RESULTS

| MODEL | VALIDATION ACCURACY |
| --- | --- |
| Bidirectional LSTM | 0.389 |
| ELMo | 0.490 |
| BERT | 0.962 |

# FINAL MODEL

- Pretrained on large amounts of datasets
- Accounts for word context
- Openly accessible to masses

# LIMITATIONS

# LIMITATIONS – Generic BiLSTM

- Small embedding dimension chosen

# LIMITATIONS - ELMo

- Lack of popularity
- Lack of support
- Lack of documentation and examples

# LIMITATIONS - BERT

- Time Consuming
- Uses a lot of memory space
- Cannot handle large text sequences

# LIMITATIONS – Other Issues

- Embedding memory usage
- Different operating systems (MacOS vs Windows)
- Size of dataset used too small
- Server update broke ELMo

# CONCLUSION

# CONCLUSION TO HYPOTHESIS 1

## Hypothesis 1: Attention Models like BERT did outperform bidirectional LSTM models and ELMo

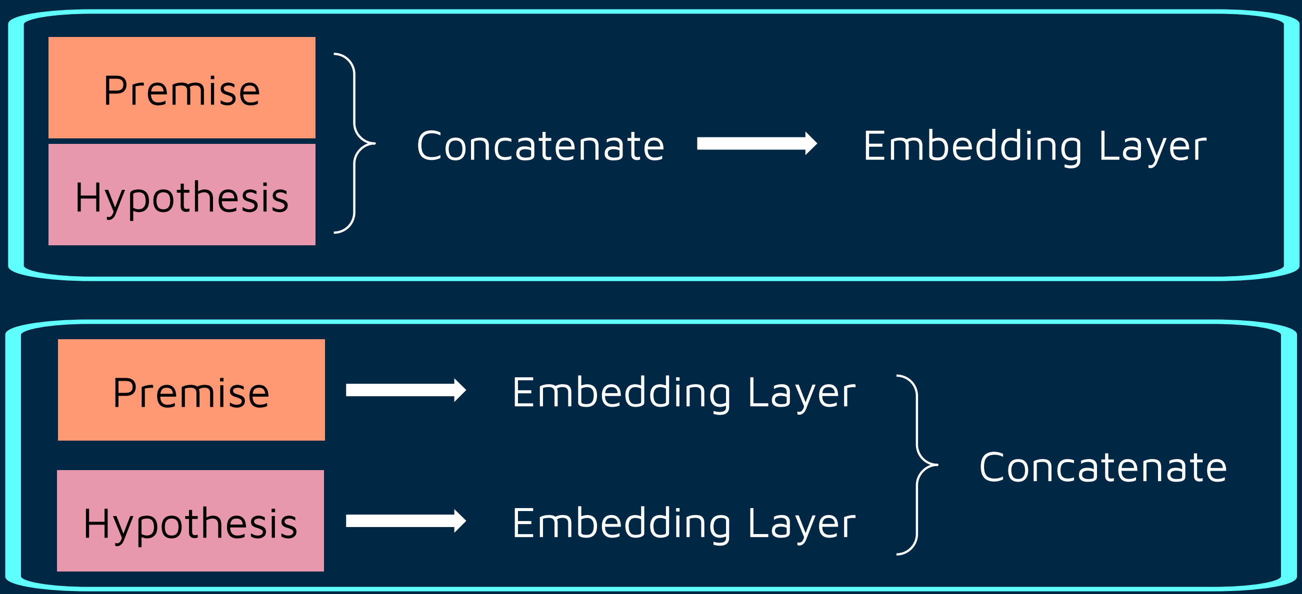
- The BERT model we used is specially fine-tuned for Natural Language Inference tasks

# CONCLUSION TO HYPOTHESIS 2

## Hypothesis 2: Split Input Model <u>did</u> perform better than a Concatenated Input Model

Method 1:
Concatenated Input

Method 2:
Split Input




Method 1 produces sentence embeddings that are dependent on the other sentence.

# CONCLUSION

# THANK YOU

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik
Please keep this slide for attribution