JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MULTILINGUAL LANGUAGE TRANSLATOR BY DETECTING VARIOUS LANGUAGES IN THE DOCUMENT / SPEECH USING NLP

Twinkle Dharashive
Dept. of Computer Engineering
Sinhgad Academy of Engineering, Pune
twinkle055@gmail.com

Sudity Khushi
Dept. of Computer Engineering
Sinhgad Academy of Engineering, Pune
sudity7654@gmail.com

Lokesh Kolte
Dept. of Computer Engineering
Sinhgad Academy of Engineering, Pune
lokeshkolte2580@gmail.com

Rohit Mahajan
Dept. of Computer Engineering
Sinhgad Academy of Engineering, Pune
rohitm.official404@gmail.com

Prof. Suvarna Bahir
(Asst. Professor, Department of Computer Engineering, SAE Kondhwa, Pune, India)

Abstract –

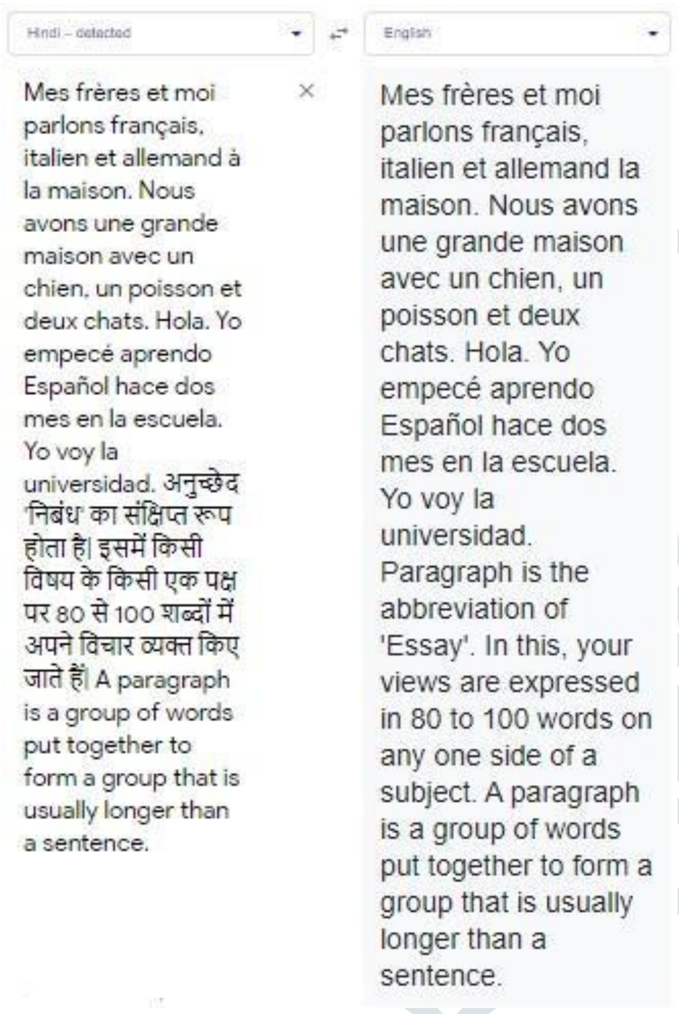
The capacity to speak with each other is a major piece of being human. There are almost 7,000 unique dialects around the world. As our reality turns out to be progressively associated, language interpretation gives a basic social and financial extension between individuals from various nations and ethnic gatherings.

Language discovery and interpretation is the endeavor of normally recognizing the dialects present in a chronicle and making an interpretation of them to the expected language present with the records. In this work, we address the issue of recognizing reports that contain message information present in single/multilingual records as well as a sound record that contain discourse and present a strategy that can perceive a record of specific language, check their relative implications, and further make an interpretation of them to a solitary language of our decision by utilizing NLP. We display the practicality of our procedure over fabricated data, similarly as obvious multilingual chronicles would do to get an appropriate interpretation of our favored language.

Introduction –

Machine interpretation is the method involved with deciphering text from one normal language into another utilizing PC framework. You might have utilized google translation, which is one of popular uses of machine interpretation. So taking a gander at the present world, which is loaded with different normal dialects, it is a significant need to switch the reports from a particular language over completely to another so that individuals can impart effectively with next to no issues.

Image – Google Translator



Mes frères et moi parlons français, italien et allemand à la maison. Nous avons une grande maison avec un chien, un poisson et deux chats. Hola. Yo empecé aprendo Español hace dos mes en la escuela. Yo voy la universidad. अनुच्छेद 'निबंध' का संक्षिप्त रूप होता है। इसमें किसी विषय के किसी एक पक्ष पर 80 से 100 शब्दों में अपने विचार व्यक्त किए जाते हैं। A paragraph is a group of words put together to form a group that is usually longer than a sentence.

Mes frères et moi parlons français, italien et allemand la maison. Nous avons une grande maison avec un chien, un poisson et deux chats. Hola. Yo empecé aprendo Español hace dos mes en la escuela. Yo voy la universidad. Paragraph is the abbreviation of 'Essay'. In this, your views are expressed in 80 to 100 words on any one side of a subject. A paragraph is a group of words put together to form a group that is usually longer than a sentence.

Fig. 1 : Google Translate

Presently we have an effective and generally utilized interpreter for example Google Translator, which can interpret the message from a particular language into another, yet what we saw here that assuming we feed message that contains sentences of different dialects, google decipher, can't make it. As you can find in the picture.

We feed the section with sentences written in Spanish, French, Hindi and English. As may be obvious, Google Translate recognizes the language as Hindi, which isn't so successful as it ought to be and it makes an interpretation of simply Hindi sentence to English yet not others. We have attempted to cover this issue and made an interpreter, which can identify different dialects in a text and afterward make an interpretation of it to a solitary language.

Language Identification (LI), additionally called as language speculating, is the assignment in Natural Language Processing (NLP) that consequently recognize the normal language wherein the substance in given record are written in. Prior to going for a specific normal language application one should distinguish the language of the substance. Normal dialects have different linguistic designs thus many undertaking of NLP, for example, POS labeling, data extraction, machine interpretation, multilingual records handling are language subordinate. Language distinguishing proof is major and vital stage in numerous NLP applications. Thus, there is need to foster a computerized apparatus and procedures for language recognizable proof before utilization of additional handling. For instance, if there should arise an occurrence of machine interpretation to change over an unknown dialect text into required language text, the language in which the first text is composed should be distinguished. Whenever it is distinguished then utilizing a machine interpretation framework it tends to be deciphered in required target language. Because of variety of records on the web, LI is an imperative errand for web search tools during slithering and ordering of web archives. For cross-lingual applications there is a rising interest to manage multilingual reports. Computationally, language distinguishing proof issue is seen as an exceptional instance of text order or characterization.

This will save the expense as well as the energy. We can think how extended this work will be in greater scale assuming that human will do this assignment. For example, to distinguish the language of each and every sentence there is in the message and afterward converting that into an sentence, etc. For that reason our model can assume a significant part to make this assignment in a matter of seconds.

Literature Survey –

1. **Automatic Language Identification in Texts: A Survey – 2018**
   Author(s) - Tommi Jauhiainen,Krister Linden
   This article provides a brief history of LI research,and an extensive survey of the features and methods used in the LI literature.

2. **Language Identification for Multilingual Machine Translation – 2020**
   Author(s) - Arun Babhulgaonkar,Shefali Sonavane
   In this paper, n-gram based and machine learning based language identifiers are trained and used to identify three Indian languages such as Hindi, Marathi and Sanskrit present in a document given for machine translation.

3. **Language Detection using Convolutional Neural Network - 2020**
   Author(s) - A K M Shahariar Azad Rabby, Md. Majedul Islam, Jebun Nahar, Fuad Rahman
   In this paper, we present a lightweight, small footprint convolutional neural network, which detects Bangla and English languages—directly from scanned mixed-language document images. The proposed model achieves 99.98% recognition accuracy for this specific two-language classification problem.

4. **Identification of Languages from The Text Document Using Natural Language Processing System - 2021**
   Author(s) - Manjula S1, Dr. Shivamurthaiah M
   In this article One of the fundamental and significant tasks of data interpretation is language detection from textual data. The current effort is to detect the 22 distinct languages in a multilingual document using the Hybrid Isomap technique.

Proposed System –

The proposed sytem is a Multilingual language translator where we take the contribution from the client and the model deciphers all the discourse/record information into the ideal text. We have utilized the voice recognition package "SpeechRecognition" which has a great deal of stowed classes worked in. One of these classes is the character module that makes the framework knows what the source is attempting to say. As this sound is mp3 or some comparable organization, we use "pydub" package to change it into wav design to get additional information for appropriate transformation into text. The changed over wav audio interrupt with the recurrence dissemination is then interpreted utilizing a form that utilizes different APIs accessible in the "SpeechRecognition" package. Here we have utilized 'SVM' (Support Vector Machine) to prepare the model for language identification. The packages like "Pipeline", "TfidVectorizer", "LinearSVC" are utilized to distinguish the language. The "TfidVectorizer" is utilized to vectorize the information in an arrangement that is utilized to recognize the language. The motivation behind the "Pipeline" is to collect a few stages that can be cross-approved together while setting various boundaries.

Besides, we use the Sequence to Sequence (seq2seq) model for translation. Seq2seq model is a class of Recurrent Neural Networks (RNN) that utilizes the feed forward strategy to figure out information. We utilize the RNN's to prepare the model with the assistance of a multilingual corpus as the base for interpretation. The "tensorflow" package that incorporates sub bundles like "keras" which goes sometime later from "layers" to "TextVectorization" are the primary bundles that are utilized for interpretation. "TextVectorization" is utilized for perceiving the huge words from the less critical ones. The "keras" and "layers" package are utilized to encode and disentangle the information to decipher the sentences. And then we get the result i.e. the language that the user chooses to translate.

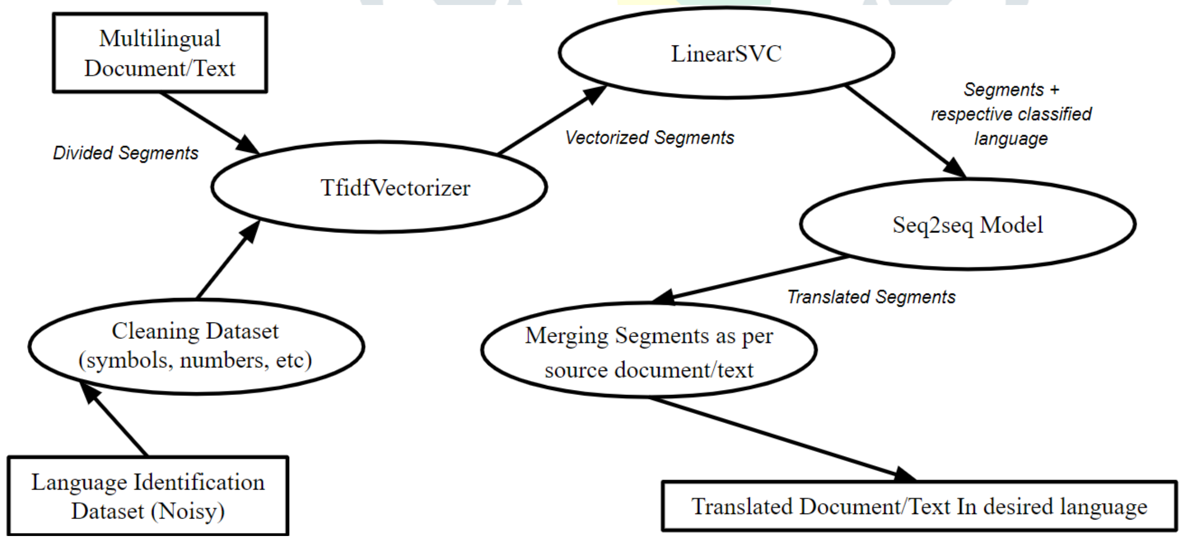

Multilingual Document/Text
Divided Segments
TfidfVectorizer
Vectorized Segments
LinearSVC
Segments + respective classified language
Seq2seq Model
Translated Segments
Merging Segments as per source document/text
Translated Document/Text In desired language
Cleaning Dataset (symbols, numbers, etc)
Language Identification Dataset (Noisy)

Fig 2. : Data Flow Diagram

Algorithm used –

Support Vector Machine

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems.

In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

Seq2seq

A Sequence to Sequence (seq2seq) takes input as a sequence of words (sentence or sentences) and generates an output sequence of words. It does so by use of the recurrent neural network (RNN).

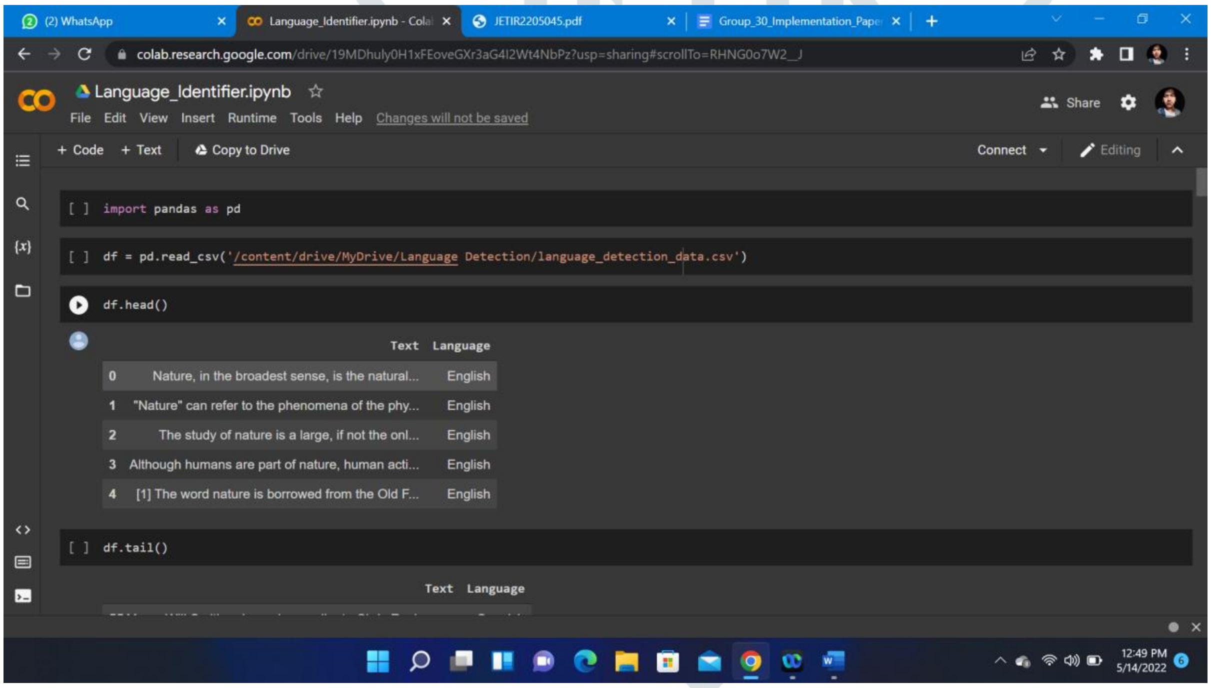
The seq2seq architecture is an encoder-decoder architecture which consists of two LSTM networks: the encoder LSTM and the decoder LSTM.

The input to the encoder LSTM is the sentence in the original language; the input to the decoder LSTM is the sentence in the translated language with a start-of-sentence token.

The output is the actual target sentence with an end-of-sentence token.

Process Flow –

1:=>Read the Dataset



```
import pandas as pd
```

```
df = pd.read_csv('/content/drive/MyDrive/Language Detection/language_detection_data.csv')
```

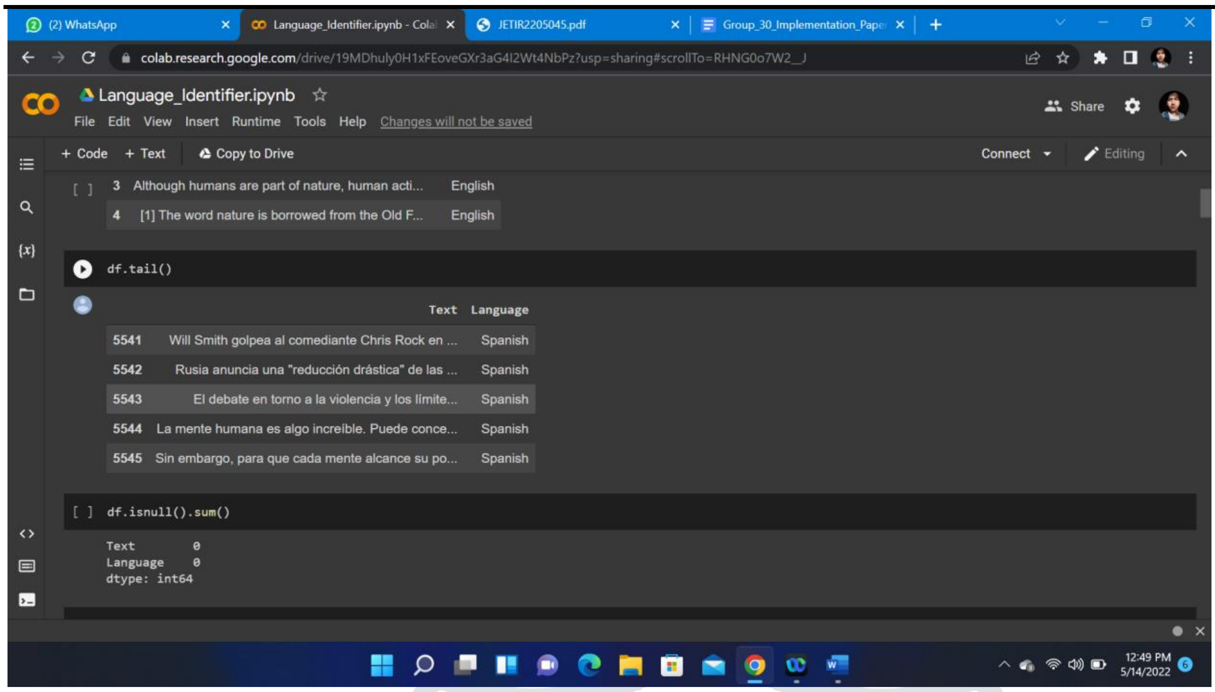
```
df.head()
```

```
                                                Text  Language
0        Nature, in the broadest sense, is the natural...   English
1   "Nature" can refer to the phenomena of the phy...   English
2          The study of nature is a large, if not the onl...   English
3   Although humans are part of nature, human acti...   English
4      [1] The word nature is borrowed from the Old F...   English
```

```
df.tail()
```

```
                                                Text  Language
```

Language_Identifier.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Share



```
3   Although humans are part of nature, human acti...   English
4   [1] The word nature is borrowed from the Old F...   English
```

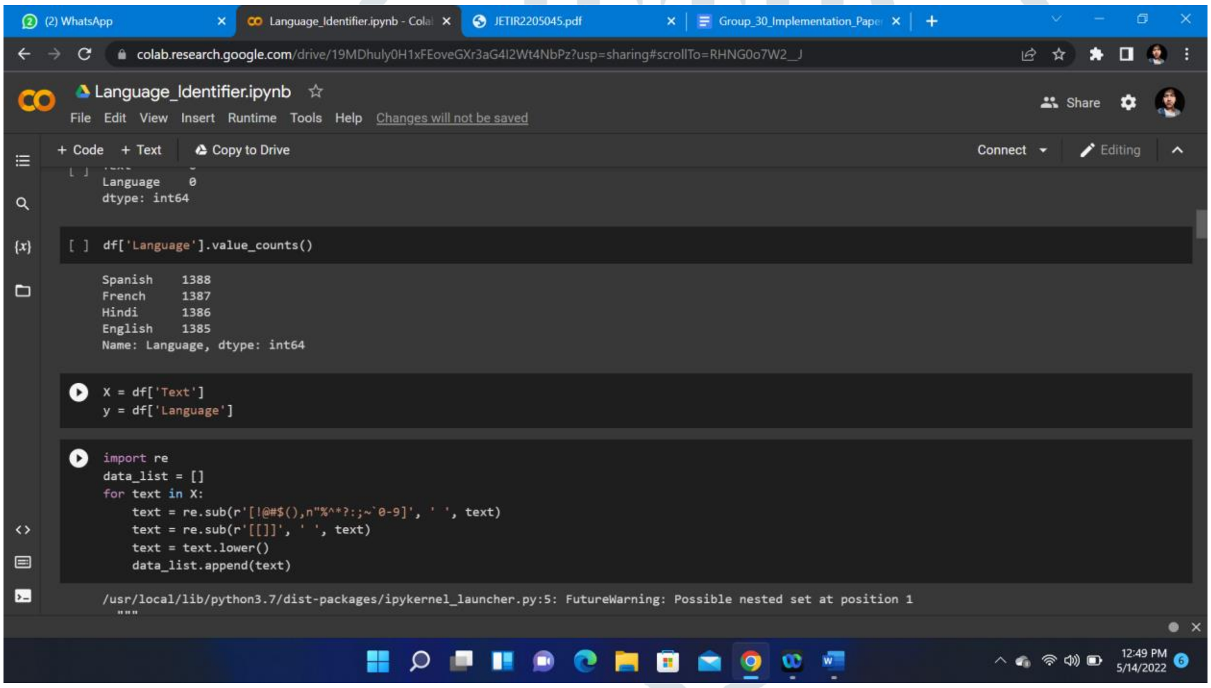
```
df.tail()
```

```
                                               Text  Language
5541  Will Smith golpea al comediante Chris Rock en ...   Spanish
5542  Rusia anuncia una "reducción drástica" de las ...   Spanish
5543     El debate en torno a la violencia y los límite...   Spanish
5544  La mente humana es algo increíble. Puede conce...   Spanish
5545  Sin embargo, para que cada mente alcance su po...   Spanish
```

```
df.isnull().sum()
```

```
Text        0
Language    0
dtype: int64
```

2:=>Remove all special character such as(;",./\[]) from the dataset



```
Language    0
dtype: int64
```

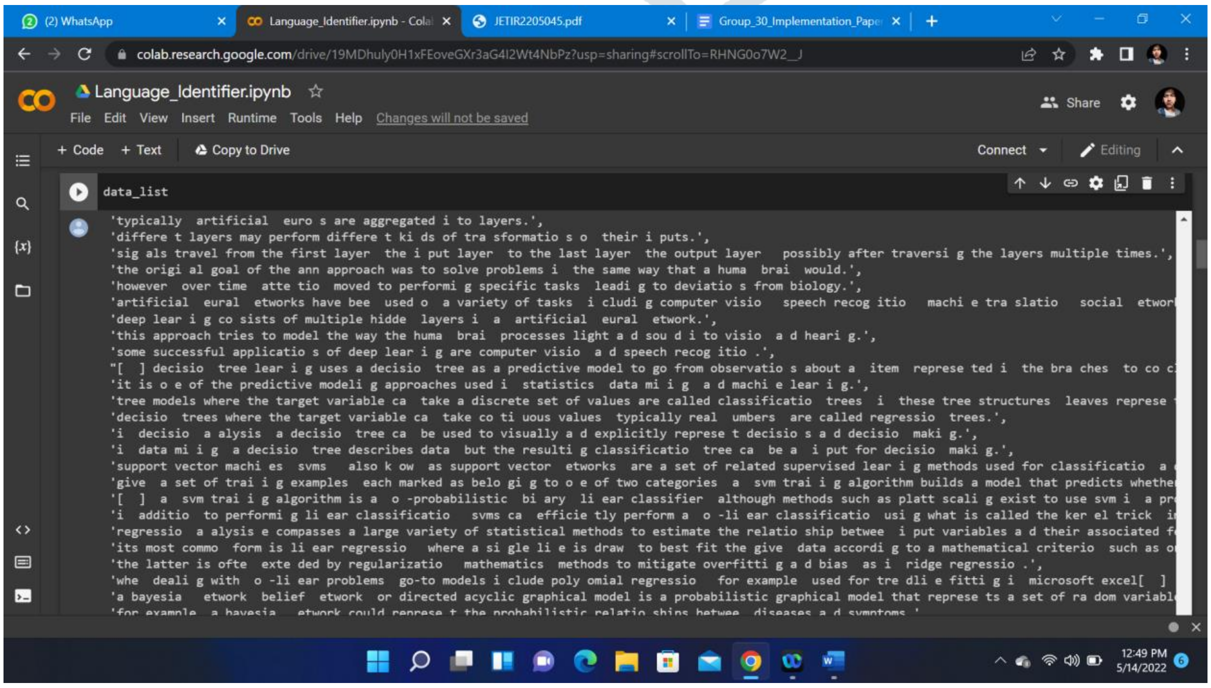
```
df['Language'].value_counts()
```

```
Spanish    1388
French     1387
Hindi      1386
English    1385
Name: Language, dtype: int64
```

```
X = df['Text']
y = df['Language']
```

```
import re
data_list = []
for text in X:
    text = re.sub(r'[!@#$(),n"%^*?:;~`0-9]', ' ', text)
    text = re.sub(r'[[]]', ' ', text)
    text = text.lower()
    data_list.append(text)
```

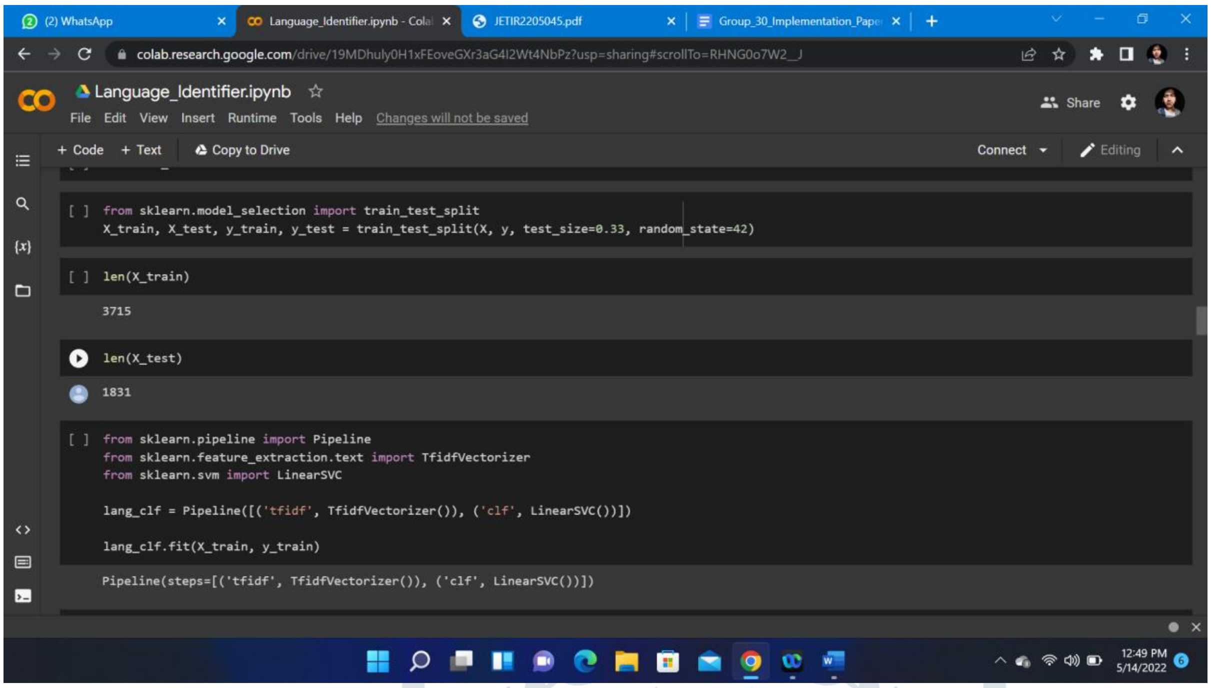
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: FutureWarning: Possible nested set at position 1
  """
```



```
data_list
```

```
'typically  artificial  euro s are aggregated i to layers.',
'differe t layers may perform differe t ki ds of tra sformatio s o  their i puts.',
'sig als travel from the first layer  the i put layer  to the last layer  the output layer   possibly after traversi g the layers multiple times.',
'the origi al goal of the ann approach was to solve problems i  the same way that a huma  brai  would.',
'however  over time  atte tio  moved to performi g specific tasks  leadi g to deviatio s from biology.',
'artificial  eural  etworks have bee  used o  a variety of tasks  i cludi g computer visio   speech recog itio   machi e tra slatio   social  etwor
'deep lear i g co sists of multiple hidde  layers i  a  artificial  eural  etwork.',
'this approach tries to model the way the huma  brai  processes light a d sou d i to visio  a d heari g.',
'some successful applicatio s of deep lear i g are computer visio  a d speech recog itio .',
"[  ] decisio  tree lear i g uses a decisio  tree as a predictive model to go from observatio s about a  item  represe ted i  the bra ches  to co c
'it is o e of the predictive modeli g approaches used i  statistics  data mi i g  a d machi e lear i g.',
'tree models where the target variable ca  take a discrete set of values are called classificatio  trees  i  these tree structures  leaves represe
'decisio  trees where the target variable ca  take co ti uous values  typically real  umbers  are called regressio  trees.',
'i  decisio  a alysis  a decisio  tree ca  be used to visually a d explicitly represe t decisio s a d decisio  maki g.',
'i  data mi i g  a decisio  tree describes data  but the resulti g classificatio  tree ca  be a  i put for decisio  maki g.',
'support vector machi es  svms   also k ow  as support vector  etworks  are a set of related supervised lear i g methods used for classificatio  a
'give  a set of trai i g examples  each marked as belo gi g to o e of two categories  a  svm trai i g algorithm builds a model that predicts whethe
'[  ]  a  svm trai i g algorithm is a  o -probabilistic  bi ary  li ear classifier  although methods such as platt scali g exist to use svm i  a pr
'i  additio  to performi g li ear classificatio   svms ca  efficie tly perform a  o -li ear classificatio  usi g what is called the ker el trick  i
'regressio  a alysis e compasses a large variety of statistical methods to estimate the relatio ship betwee  i put variables a d their associated f
'its most commo  form is li ear regressio   where a si gle li e is draw  to best fit the give  data accordi g to a mathematical criterio  such as o
'the latter is ofte  exte ded by regularizatio  mathematics  methods to mitigate overfitti g a d bias  as i  ridge regressio .',
'whe  deali g with  o -li ear problems  go-to models i clude poly omial regressio   for example  used for tre dli e fitti g i  microsoft excel[  ]
'a bayesia  etwork  belief  etwork  or directed acyclic graphical model is a probabilistic graphical model that represe ts a set of ra dom variabl
'for example  a bayesia  etwork could represe t the probabilistic relatio ships betwee  diseases a d symptoms.'
```

3 Train the model:=>


Language_Identifier.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

```
len(X_train)
```

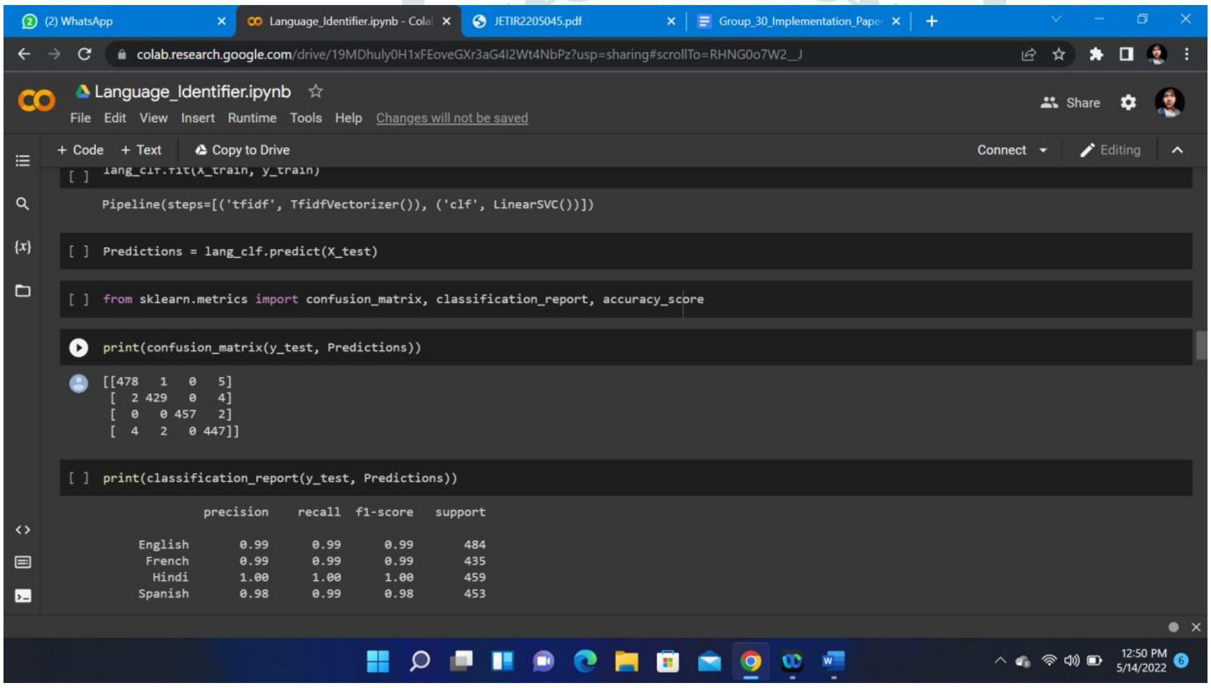
```
3715
```

```
len(X_test)
```

```
1831
```

```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC

lang_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', LinearSVC())])

lang_clf.fit(X_train, y_train)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', LinearSVC())])
```


Language_Identifier.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing

```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', LinearSVC())])
```

```
Predictions = lang_clf.predict(X_test)
```

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

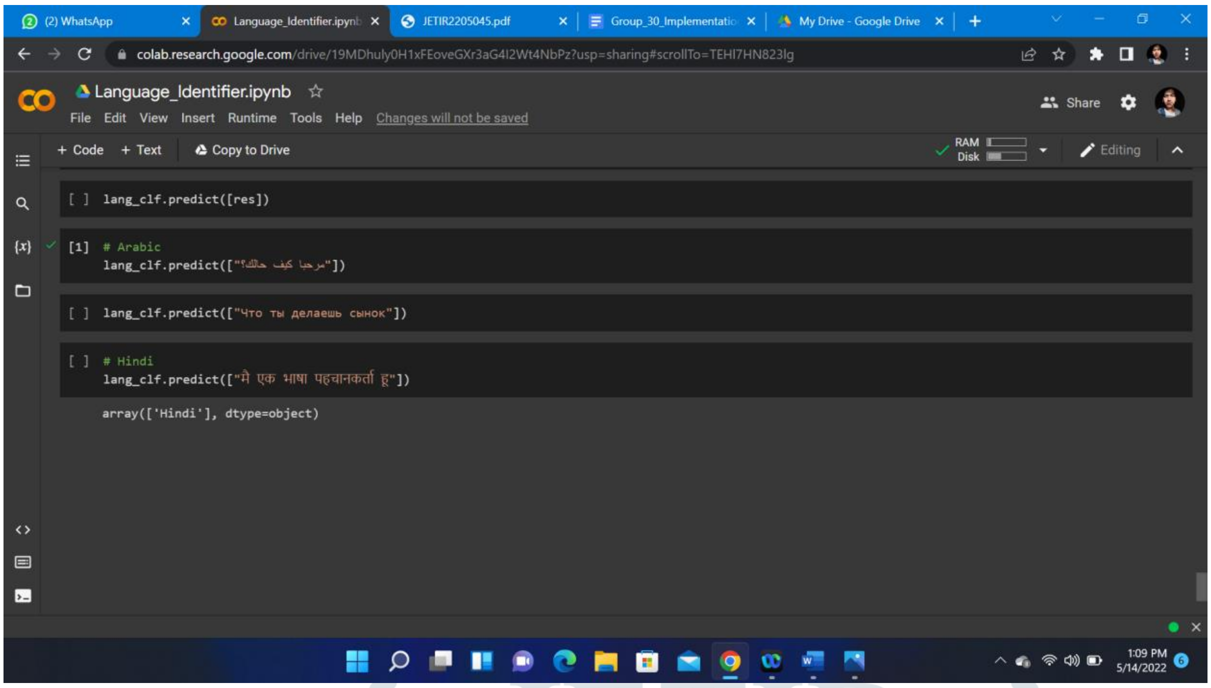
```
print(confusion_matrix(y_test, Predictions))
```

```
[[478   1   0   5]
 [  2 429   0   4]
 [  0   0 457   2]
 [  4   2   0 447]]
```

```
print(classification_report(y_test, Predictions))
```

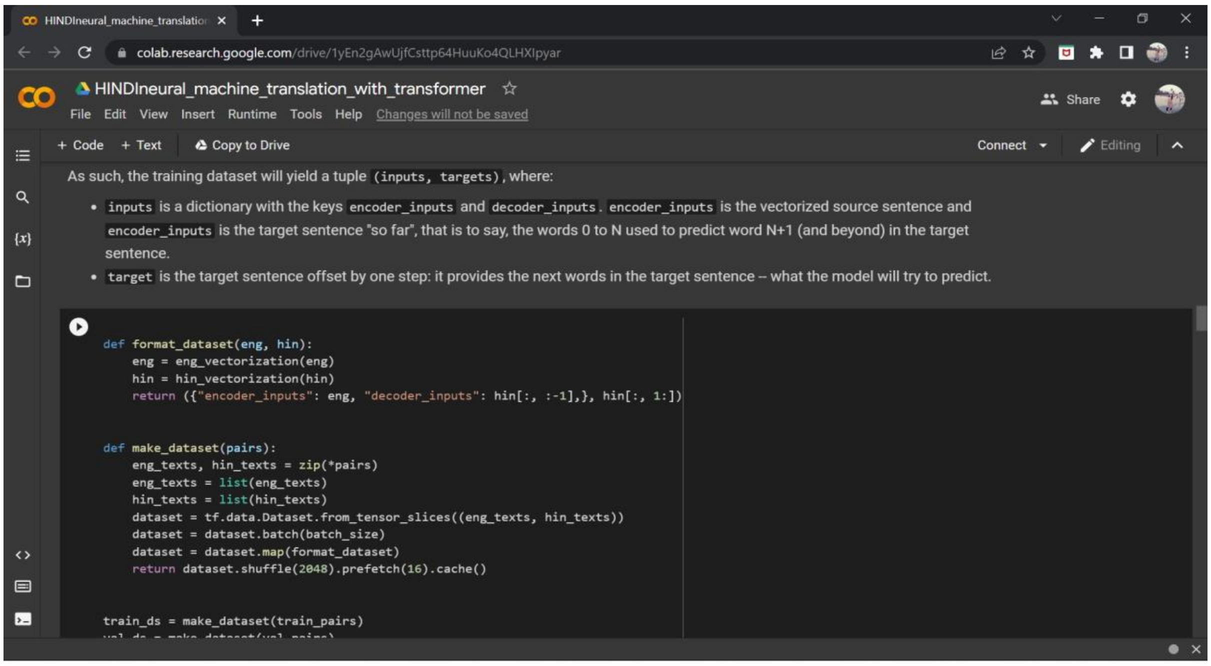
```
              precision    recall  f1-score   support

     English       0.99      0.99      0.99       484
      French       0.99      0.99      0.99       435
       Hindi       1.00      1.00      1.00       459
     Spanish       0.98      0.99      0.98       453
```

4=>This is the result for identification


Language_Identifier.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive

```
[ ]  lang_clf.predict([res])

[1]  # Arabic
     lang_clf.predict(["مرحبا كيف حالك؟"])

[ ]  lang_clf.predict(["Что ты делаешь сынок"])

[ ]  # Hindi
     lang_clf.predict(["मै एक भाषा पहचानकर्ता हू"])

     array(['Hindi'], dtype=object)
```

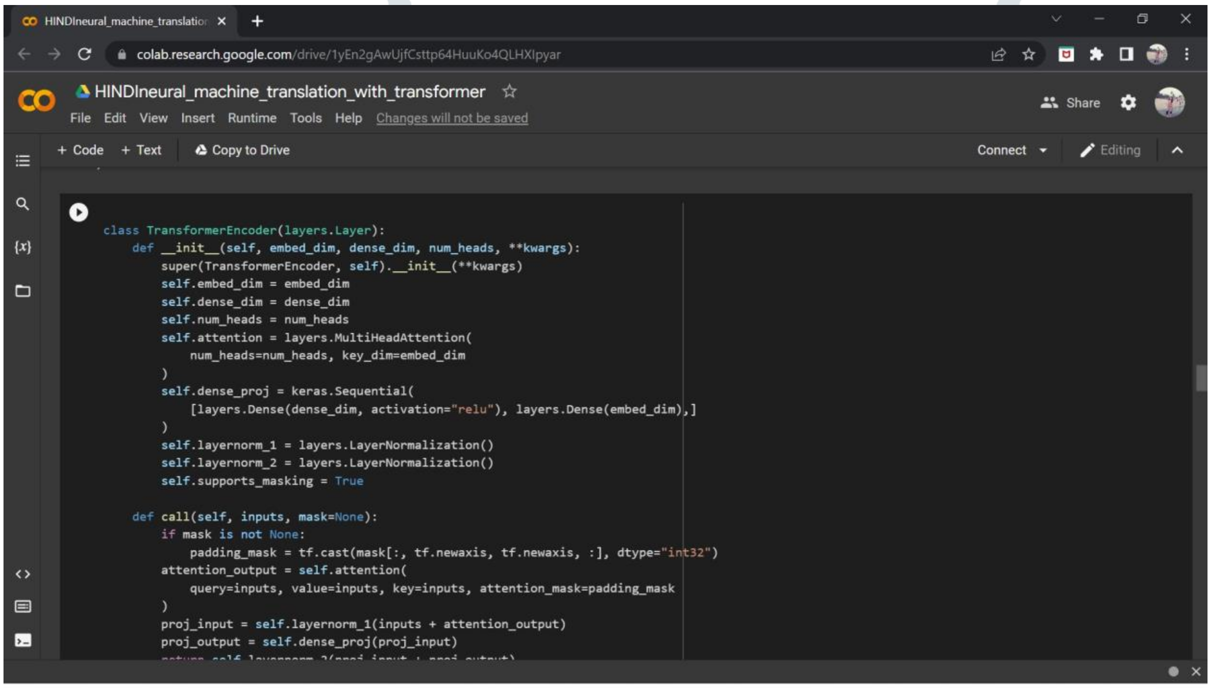
5=Encoding and decoding


HINDIneural_machine_translation_with_transformer
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Share

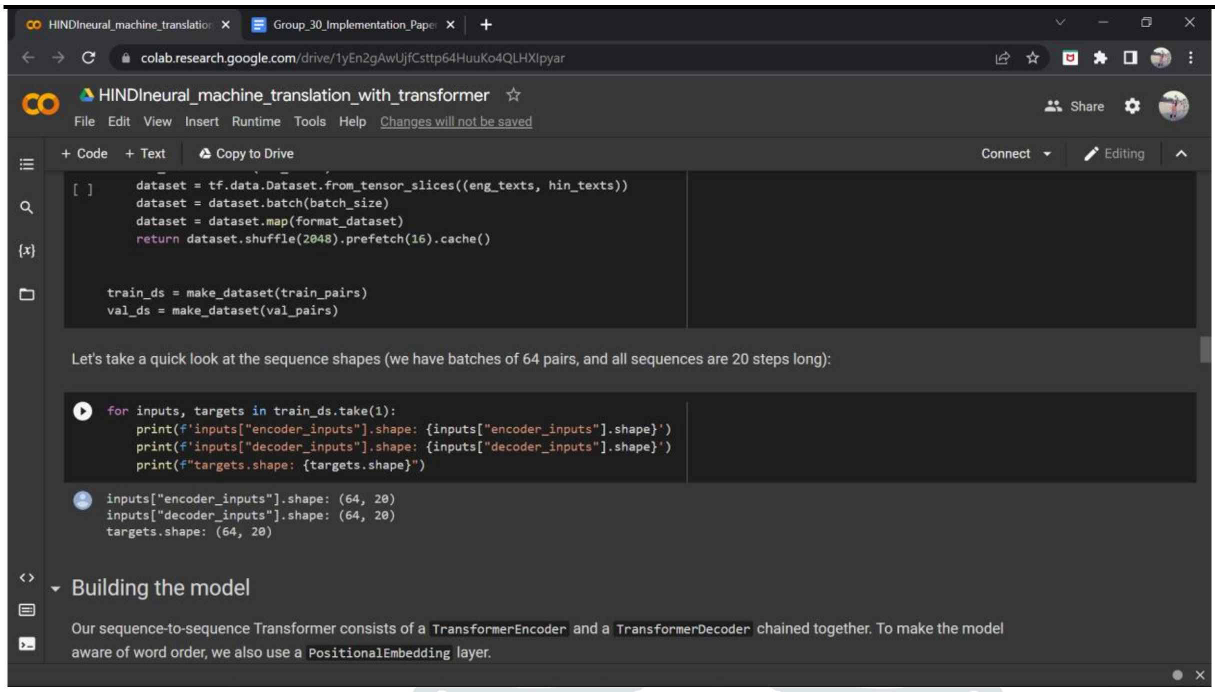
As such, the training dataset will yield a tuple `(inputs, targets)`, where:

- `inputs` is a dictionary with the keys `encoder_inputs` and `decoder_inputs`. `encoder_inputs` is the vectorized source sentence and `encoder_inputs` is the target sentence "so far", that is to say, the words 0 to N used to predict word N+1 (and beyond) in the target sentence.
- `target` is the target sentence offset by one step: it provides the next words in the target sentence – what the model will try to predict.

```python
def format_dataset(eng, hin):
    eng = eng_vectorization(eng)
    hin = hin_vectorization(hin)
    return ({"encoder_inputs": eng, "decoder_inputs": hin[:, :-1],}, hin[:, 1:])


def make_dataset(pairs):
    eng_texts, hin_texts = zip(*pairs)
    eng_texts = list(eng_texts)
    hin_texts = list(hin_texts)
    dataset = tf.data.Dataset.from_tensor_slices((eng_texts, hin_texts))
    dataset = dataset.batch(batch_size)
    dataset = dataset.map(format_dataset)
    return dataset.shuffle(2048).prefetch(16).cache()


train_ds = make_dataset(train_pairs)
```


HINDIneural_machine_translation_with_transformer
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Share

```python
class TransformerEncoder(layers.Layer):
    def __init__(self, embed_dim, dense_dim, num_heads, **kwargs):
        super(TransformerEncoder, self).__init__(**kwargs)
        self.embed_dim = embed_dim
        self.dense_dim = dense_dim
        self.num_heads = num_heads
        self.attention = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim
        )
        self.dense_proj = keras.Sequential(
            [layers.Dense(dense_dim, activation="relu"), layers.Dense(embed_dim),]
        )
        self.layernorm_1 = layers.LayerNormalization()
        self.layernorm_2 = layers.LayerNormalization()
        self.supports_masking = True

    def call(self, inputs, mask=None):
        if mask is not None:
            padding_mask = tf.cast(mask[:, tf.newaxis, tf.newaxis, :], dtype="int32")
        attention_output = self.attention(
            query=inputs, value=inputs, key=inputs, attention_mask=padding_mask
        )
        proj_input = self.layernorm_1(inputs + attention_output)
        proj_output = self.dense_proj(proj_input)
```



```
    dataset = tf.data.Dataset.from_tensor_slices((eng_texts, hin_texts))
    dataset = dataset.batch(batch_size)
    dataset = dataset.map(format_dataset)
    return dataset.shuffle(2048).prefetch(16).cache()


train_ds = make_dataset(train_pairs)
val_ds = make_dataset(val_pairs)
```

Let's take a quick look at the sequence shapes (we have batches of 64 pairs, and all sequences are 20 steps long):

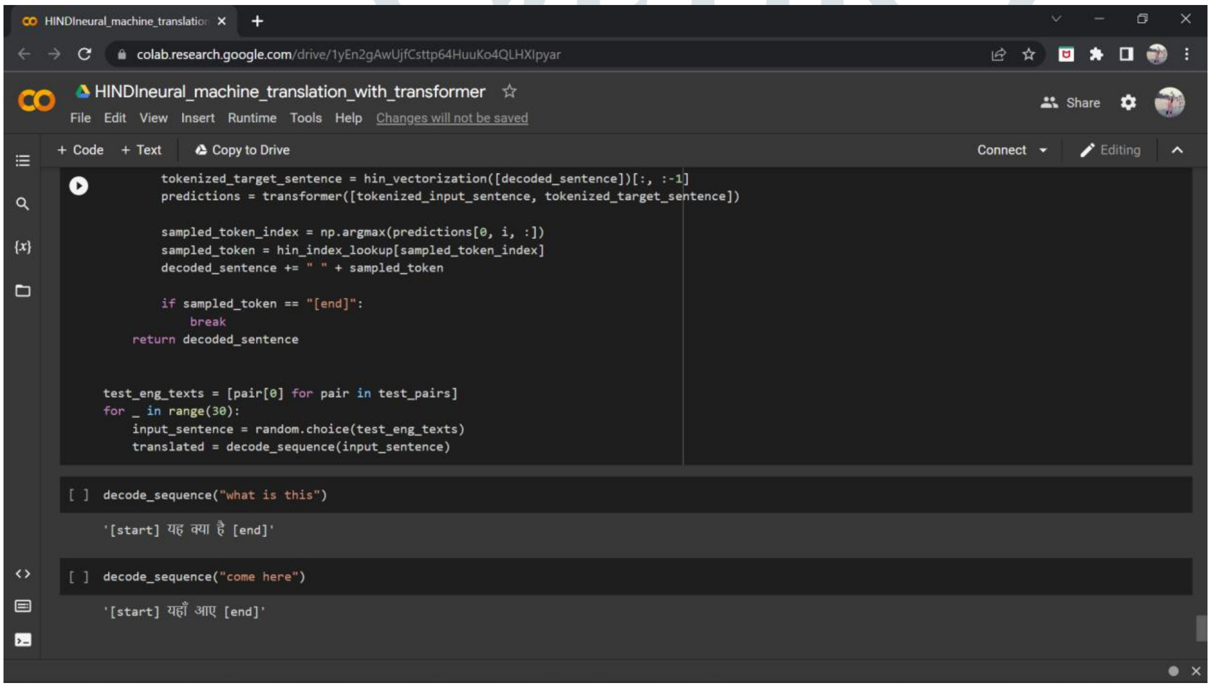
```
for inputs, targets in train_ds.take(1):
    print(f'inputs["encoder_inputs"].shape: {inputs["encoder_inputs"].shape}')
    print(f'inputs["decoder_inputs"].shape: {inputs["decoder_inputs"].shape}')
    print(f"targets.shape: {targets.shape}")
```

```
inputs["encoder_inputs"].shape: (64, 20)
inputs["decoder_inputs"].shape: (64, 20)
targets.shape: (64, 20)
```

Building the model

Our sequence-to-sequence Transformer consists of a `TransformerEncoder` and a `TransformerDecoder` chained together. To make the model aware of word order, we also use a `PositionalEmbedding` layer.

5=> All languages are implemented in the same way.



```
        tokenized_target_sentence = hin_vectorization([decoded_sentence])[:, :-1]
        predictions = transformer([tokenized_input_sentence, tokenized_target_sentence])

        sampled_token_index = np.argmax(predictions[0, i, :])
        sampled_token = hin_index_lookup[sampled_token_index]
        decoded_sentence += " " + sampled_token

        if sampled_token == "[end]":
            break
    return decoded_sentence


test_eng_texts = [pair[0] for pair in test_pairs]
for _ in range(30):
    input_sentence = random.choice(test_eng_texts)
    translated = decode_sequence(input_sentence)
```

```
decode_sequence("what is this")
```

```
'[start] यह क्या है [end]'
```

```
decode_sequence("come here")
```

```
'[start] यहाँ आए [end]'
```

Conclusion –

The solution provided in this paper has tried to overcome the challenge of a different language that can be a barrier to many people. We have created an automated system to detect the language as well as translate it to remove this barrier. With our proposed SVM and seq2seq trained system, language identification and machine translation can compare the translations of full sentences before translating them, which provides you with a high quality and human-sounding output.