**NLP Mini Project- BHAVNA: Mixed Language Sentiment Analysis**

***By:***

**Prateek Mehta 32 D17C**

**Alisha Punwani 43 D17C**

**Harsh Sachanandani 49 D17C**

**Title:** BHAVNA - Mixed language sentiment analysis

**Introduction:**

Code-mixed data is an important challenge of natural language processing, as its characteristics widely vary from the traditional structures of standard languages. In such scenarios, context aware sentiment analysis of social media data becomes a great challenge. Thus, a need for analysis tools for code-mixed data arises. Although much research in multilingual and cross-lingual sentiment analysis has used semi-supervised or unsupervised methods, supervised methods still perform better. Only a few datasets for popular languages such as English-Spanish, English-Hindi, and English-Chinese are available. This project is a web application named Bhavna, which aims to bridge this gap and make sentiment analysis easier in mixed languages specifically for English-Hindi mixed text.

**Description:**

The holy grail of artificial intelligence is understanding people's thoughts and emotions in the words they use and reply to them in the same language. This is complicated by the fact that most people in day to day conversation not only do not employ or follow strict grammatical rules and regulations, but many don't even use the same language in a singular thought expression. Forcing people to strictly follow and use a selection of words and phrases defeats the purpose of the naturalness of conversation and language as we know it.   
 This problem however is completely solvable within the realm of machine learning and NLP techniques. Sentiment analysis has emerged as one of the prominent research branches because of its endless usages and applications. Monitoring social media, forums, blogs and other online resources for customer reviews, product competition and survey responses to understand customer insight is of significant importance in business analytics. With the proliferation of informal user generated data online, the use of mixed language has become a common phenomenon. Mixed language arises through the use of linguistic code switching (LCS) or the practice of using more than one language in a single sentence. Such mixed language has rarely been a subject of sentiment analysis before. The lack of a clear grammatical structure renders the previous approaches to sentiment analysis ineffective for such text.

A sentiment analysis system for text analysis utilizes natural language processing (NLP) to assign weighted sentiment scores to the entities, topics, themes and categories within a sentence or phrase.

Code-mixed data is an important challenge of natural language processing, as its characteristics widely vary from the traditional structures of standard languages. In such a scenario, context-aware sentiment analysis of social media data becomes a great challenge. Thus, a need for analysis tools for code-mixed data arises.

Today we aim to solve the task of performing sentiment analysis on a mixed code text. Using a twitter dataset on collated hindi-english dataset with good labels across 3 categories, Neutral, Positive, Negative. LSTMs are the preferred model type used to tackle similar problems in other areas. So we use a couple of LSTM models, one bidirectional and one single flow. Along with two convolutional networks to give a combined highly accurate result.

**Methodology & Application:**

We have trained a total of 4 sequential models on twitter based Hindi-English mixed text with good labels across 3 categories, Neutral, Positive, Negative. They are sequential models with an average of 2.8 million trainable parameters The models are fed spliced and tokenized strings with sequence lengths padded to 50 characters. They were then trained to 10 epochs each. The LSTM models achieved an accuracy of 59%, whereas the CNN models performed slightly better at 62% accuracy. The structure of each model is described below.

1)

Model\_name: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding (Embedding) (None, 50, 128) 2560000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

bidirectional (Bidirectional (None, 256) 263168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 64) 16448

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 32) 2080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 3) 99

=================================================================

Total params: 2,841,795

Trainable params: 2,841,795

2)

Model\_name: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding\_1 (Embedding) (None, 50, 128) 2560000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d (Conv1D) (None, 48, 64) 24640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

global\_average\_pooling1d (Gl (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 64) 4160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_4 (Dense) (None, 3) 195

=================================================================

Total params: 2,588,995

Trainable params: 2,588,995

3)  
  
Model\_name: "sequential\_2"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding\_2 (Embedding) (None, 50, 128) 2560000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_1 (LSTM) (None, 264) 415008

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 64) 16960

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_6 (Dense) (None, 3) 195

=================================================================

Total params: 2,992,163

Trainable params: 2,992,163

4)

Model\_name: "sequential\_3"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding\_3 (Embedding) (None, 50, 128) 2560000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_1 (Conv1D) (None, 48, 64) 24640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_2 (Conv1D) (None, 46, 32) 6176

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_3 (Conv1D) (None, 44, 16) 1552

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

global\_average\_pooling1d\_1 ( (None, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_7 (Dense) (None, 8) 136

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_8 (Dense) (None, 3) 27

=================================================================

Total params: 2,592,531

Trainable params: 2,592,531

Use Cases of the Project:

* Discover negative reviews of your product or service. On blog posts or eCommerce sites or social media. More broadly anywhere on the web.
* Aggregate sentiment on financial instruments. Such as specific stocks. What is the recent market sentiment on stock xyz? Also, aspect-based variants. Such as according to analysts at financial company xyz, stock abc is likely to grow 20% in the coming year.
* Discerning who’s opinion it is provides more information, which may be used to assess credibility or lack thereof.
* Identify which components of your product or service are people complaining about? Especially strongly. For prioritizing tactical or long-term improvements.

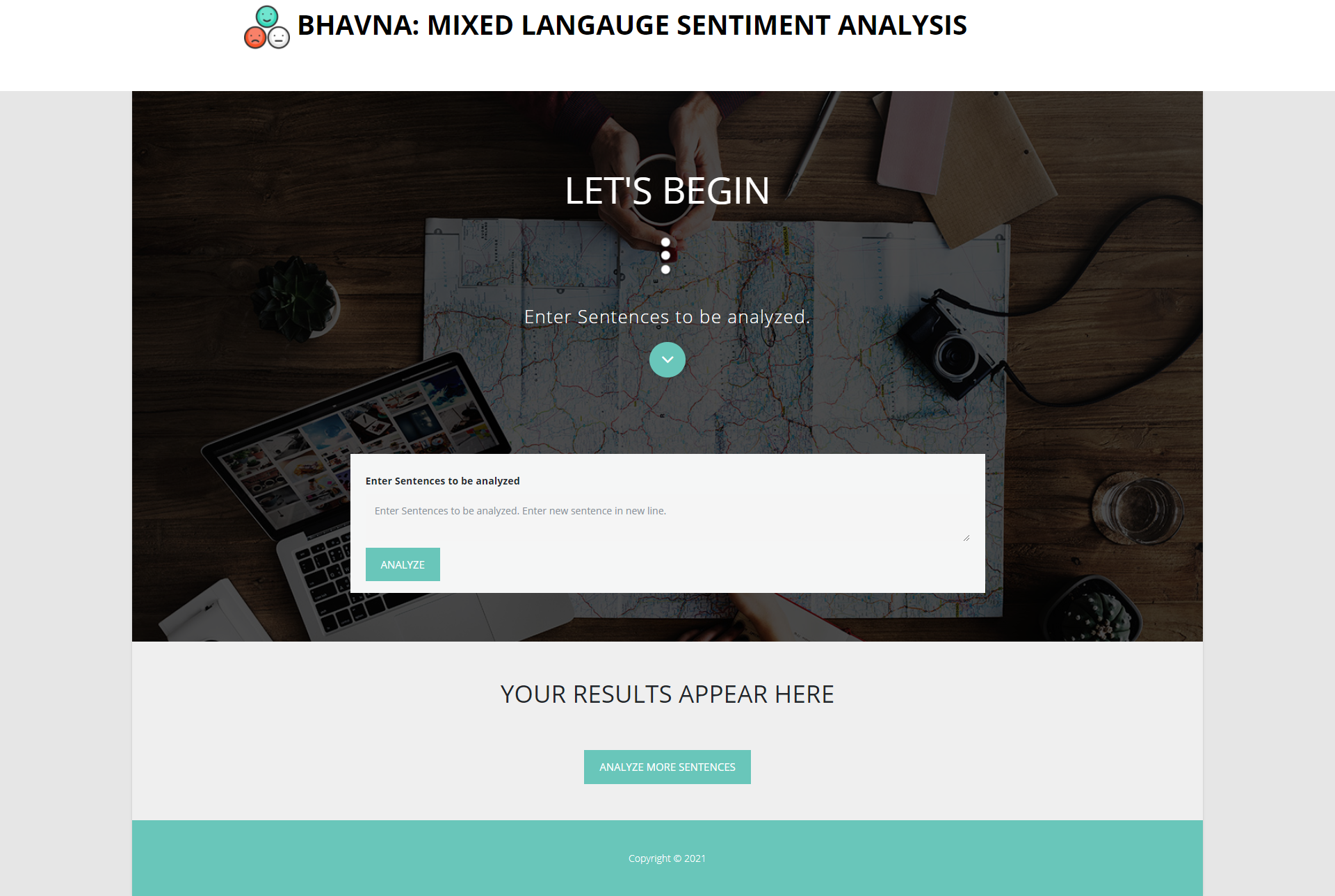
Track shifting opinions of politicians over time. Individuals or groups such as political parties. News media love to do this.  
  
**Code:**   
Flask app.py:-

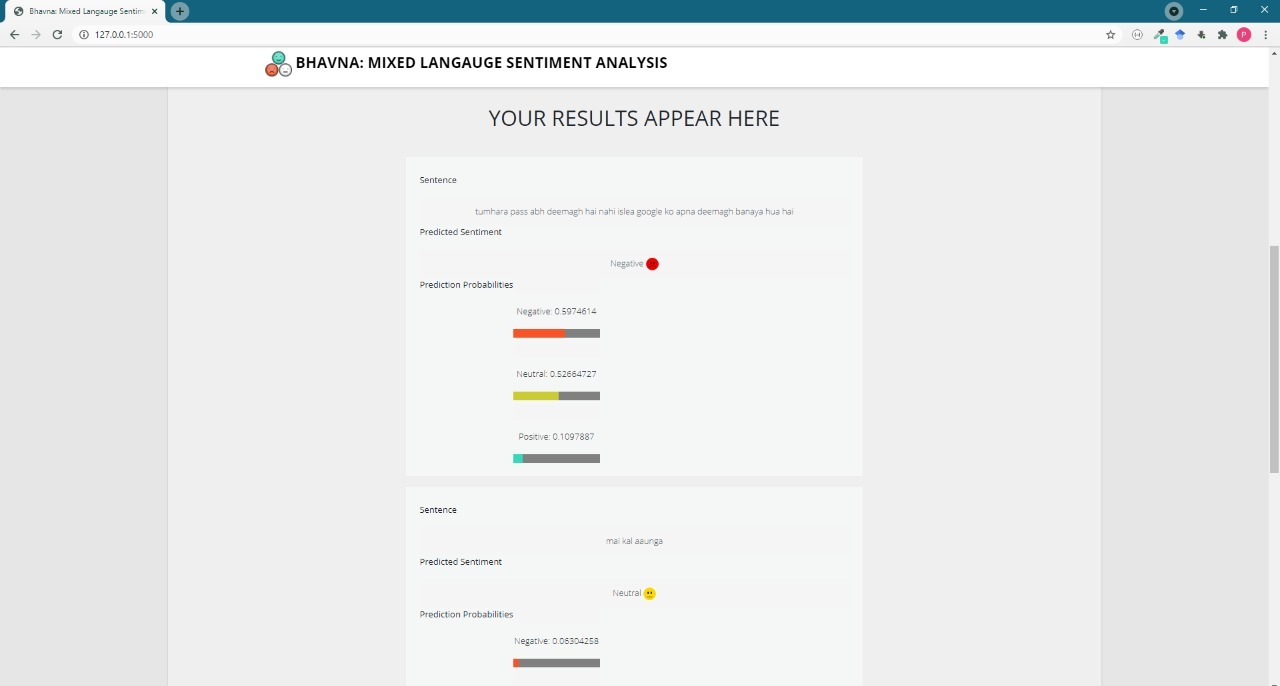
|  |
| --- |
| from flask import Flask, render\_template, flash, request import numpy as np import tensorflow as tf from tensorflow.keras.preprocessing.text import Tokenizer from tensorflow.keras.preprocessing.sequence import pad\_sequences   *# App config.* app = Flask(\_\_name\_\_, template\_folder='html')  gpus = tf.config.experimental.list\_physical\_devices('GPU') if gpus:  try:  for gpu in gpus:  tf.config.experimental.set\_memory\_growth(gpu, True)  except RuntimeError as e:  print(e)  bimodel = tf.keras.models.load\_model('bimodel.h5') clmodel = tf.keras.models.load\_model('clmodel.h5') model = tf.keras.models.load\_model('model.h5') cnn = tf.keras.models.load\_model('cnn.h5')  def predict(data):  tokenizer = Tokenizer(num\_words=2500,split=' ')  tokenizer.fit\_on\_texts(data)   X= tokenizer.texts\_to\_sequences(data)  X = pad\_sequences(X,maxlen=50)   x1 = bimodel.predict(X)  x2 = clmodel.predict(X)  x3 = model.predict(X)  x4 = cnn.predict(X)  ans=[]  for i in range(len(x1)):  b=[]  for j in range(len(x1[i])):  z = x1[i][j] + x2[i][j] + x3[i][j] + x4[i][j]  z = z/len(x1[i])  b.append(z)  ans.append(b)  b=[]    return ans       @app.route("/", methods=['GET', 'POST']) def page():  output = []  if request.method == "POST":  data = request.form.get("data")  data = data.strip()  data = data.split('\n')  for i in range(len(data)):  if data[i][-1] == '\r':  data[i] = data[i][0:-1]  response = predict(data)  for i in range(len(response)):  ans = {  "data" : data[i],  "pred" : response[i],  "pred\_out" : np.argmax(response[i])  }  output.append(ans)  return render\_template("index.html", output=output)  if \_\_name\_\_ == "\_\_main\_\_":  print("Running on ",len(gpus)," GPUs")  app.run(port=5000, debug=True) |
|  |
|  |

Index.html :-

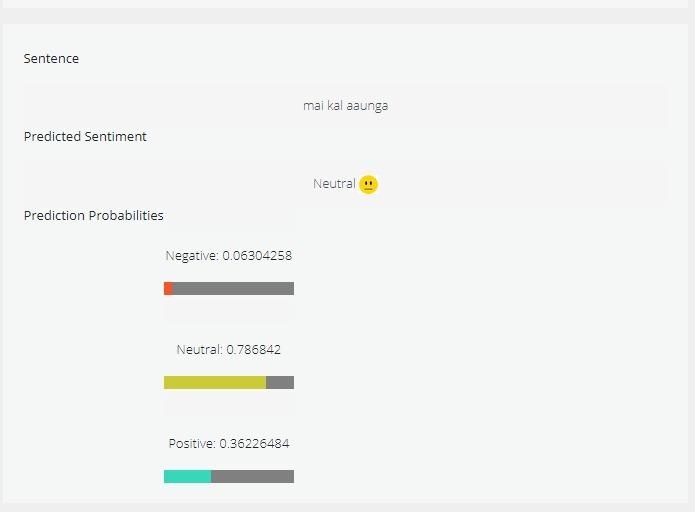
|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">   <title>Bhavna: Mixed Langauge Sentiment Analysis</title>   <!-- load stylesheets -->  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"> <!-- Google web font "Open Sans" -->  <link rel="stylesheet" href="{{ url\_for('static', filename='font-awesome-4.7.0/css/font-awesome.min.css') }}"> <!-- Font Awesome -->  <link rel="stylesheet" href="{{ url\_for('static', filename='css/bootstrap.min.css') }}"> <!-- Bootstrap style -->  <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='css/datepicker.css') }}"/>  <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='slick/slick.css') }}"/>  <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='slick/slick-theme.css') }}"/>  <link rel="stylesheet" href="{{ url\_for('static', filename='css/templatemo-style.css') }}"> <!-- Templatemo style -->   <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->  <!--[if lt IE 9]>  <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>  <![endif]-->  <style>  progress {  -webkit-appearance: none;  appearance: none;   }  progress.negative::-webkit-progress-value{  background:#F95428;  }   progress.neutral::-webkit-progress-value{  background:rgba(255, 255, 0, 0.589);  }   progress.positive::-webkit-progress-value{  background:#35D8B9;  }    </style>  </head>   <body>  <div class="tm-main-content" id="top">  <div class="tm-top-bar-bg"></div>   <div class="tm-top-bar" id="tm-top-bar">  <div class="container">  <div class="row">  <nav class="navbar navbar-expand-lg narbar-light">  <a class="navbar-brand mr-auto" href="#">  <img src="{{ url\_for('static', filename='img/sentilogo.png') }}" alt="Site logo">  Bhavna: Mixed Langauge Sentiment Analysis  </a>    </nav>  </div> <!-- row -->  </div> <!-- container -->  </div> <!-- .tm-top-bar -->   <div class="tm-page-wrap mx-auto">  <section class="tm-banner">  <div class="tm-container-outer tm-banner-bg">  <div class="container">   <div class="row tm-banner-row tm-banner-row-header">  <div class="col-xs-12">  <div class="tm-banner-header">  <h1 class="text-uppercase tm-banner-title">Let's begin</h1>  <img src="{{ url\_for('static', filename='img/dots-3.png') }}" alt="Dots">  <p class="tm-banner-subtitle">Enter Sentences to be analyzed.</p>  <a href="javascript:void(0)" class="tm-down-arrow-link"><i class="fa fa-2x fa-angle-down tm-down-arrow"></i></a>   </div>   </div> <!-- col-xs-12 -->   </div> <!-- row -->  <div class="row tm-banner-row" id="tm-section-search">   <form action="http://127.0.0.1:5000/" method="POST" class="tm-search-form tm-section-pad-2">  <div class="form-row tm-search-form-row">   <label for="input">Enter Sentences to be analyzed</label>  <textarea name="data" class="form-control" id="input" placeholder="Enter Sentences to be analyzed. Enter new sentence in new line."></textarea>  <input type="submit" class="text-uppercase btn-primary tm-btn" value="Analyze" style="margin-top : 0.5rem;">  </div>   </form>    </div> <!-- row -->  <div class="tm-banner-overlay"></div>  </div> <!-- .container -->   </div> <!-- .tm-container-outer -->   </section>   <section class="p-5 tm-container-outer tm-bg-gray">   <div class="container">  <div class="row">  <div class="col-xs-12 mx-auto tm-about-text-wrap text-center">   <h2 class="text-uppercase mb-4">Your Results Appear Here</h2>   {% for n in output %}   <div class="row tm-banner-row" style="padding-top: 1rem;">  <form class="tm-search-form tm-section-pad-2">  <div class="form-row tm-search-form-row">   <p><b>Sentence</b></p>  <p class="form-control" >{{ n["data"] }}</p>  </div>   <div class="form-row tm-search-form-row">   <p><b>Predicted Sentiment</b></p>  {% if n["pred\_out"] == 0 %}  <p class="form-control" >Negative <img src="{{ url\_for('static', filename='img/red.png') }}" alt="Sad Face"> </p>  {% endif %}  {% if n["pred\_out"] == 1 %}  <p class="form-control" >Neutral <img src="{{ url\_for('static', filename='img/yellow.png') }}" alt="Neutral Face"> </p>  {% endif %}  {% if n["pred\_out"] == 2 %}  <p class="form-control" >Positive <img src="{{ url\_for('static', filename='img/green.png') }}" alt="Happy Face"> </p>  {% endif %}  </div>   <div class="form-row tm-search-form-row">   <p><b>Prediction Probabilities</b></p>  <div style="align-items: center;">  <p class="form-control"></p>  <p >Negative: {{ n['pred'][0] }}</p>   <p class="form-control" style="padding: 0rem; margin: 0rem;"></p>  <progress value = "{{ n['pred'][0] }}" max = "1" class="negative">Negative</progress>  <p class="form-control"></p>  <p>Neutral: {{ n['pred'][1] }}</p>   <p class="form-control" style="padding: 0rem; margin: 0rem;"></p>  <progress value = "{{ n['pred'][1] }}" max = "1" class="neutral">Neutral</progress>  <p class="form-control"></p>  <p>Positive: {{ n['pred'][2] }}</p>   <p class="form-control" style="padding: 0rem; margin: 0rem;"></p>  <progress value = "{{ n['pred'][2] }}" max = "1" class="positive">Positive</progress>   </div>   </div>  </form>   </div>   {% endfor %}     <a href="*#" class="text-uppercase btn-primary tm-btn" style="margin-top: 2rem;">Analyze more Sentences</a>*   </div>  </div>  </div>   </section>   <footer class="tm-container-outer">  <p class="mb-0">Copyright (c) <span class="tm-current-year">2021</span></p>  </footer>  </div>  </div> <!-- .main-content -->   <!-- load JS files -->  <script src="{{ url\_for('static', filename='js/jquery-1.11.3.min.js') }}"></script> <!-- jQuery (https://jquery.com/download/) -->  <script src="{{ url\_for('static', filename='js/popper.min.js') }}"></script> <!-- https://popper.js.org/ -->   <script src="{{ url\_for('static', filename='js/bootstrap.min.js') }}"></script> <!-- https://getbootstrap.com/ -->  <script src="{{ url\_for('static', filename='js/datepicker.min.js') }}"></script> <!-- https://github.com/qodesmith/datepicker -->  <script src="{{ url\_for('static', filename='js/jquery.singlePageNav.min.js') }}"></script> <!-- Single Page Nav (https://github.com/ChrisWojcik/single-page-nav) -->  <script src="{{ url\_for('static', filename='slick/slick.min.js') }}"></script> <!-- http://kenwheeler.github.io/slick/ -->  <script src="{{ url\_for('static', filename='js/jquery.scrollTo.min.js') }}"></script> <!-- https://github.com/flesler/jquery.scrollTo -->  <script>    /\* DOM is ready  ------------------------------------------------\*/  $(function(){   // Change top navbar on scroll  $(window).on("scroll", function() {  if($(window).scrollTop() > 100) {  $(".tm-top-bar").addClass("active");  } else {   $(".tm-top-bar").removeClass("active");  }  });   // Smooth scroll to search form  $('.tm-down-arrow-link').click(function(){  $.scrollTo('#tm-section-search', 300, {easing:'linear'});  });   // Slick Carousel  $('.tm-slideshow').slick({  infinite: true,  arrows: true,  slidesToShow: 1,  slidesToScroll: 1  });    });   </script>   </body> </html> |

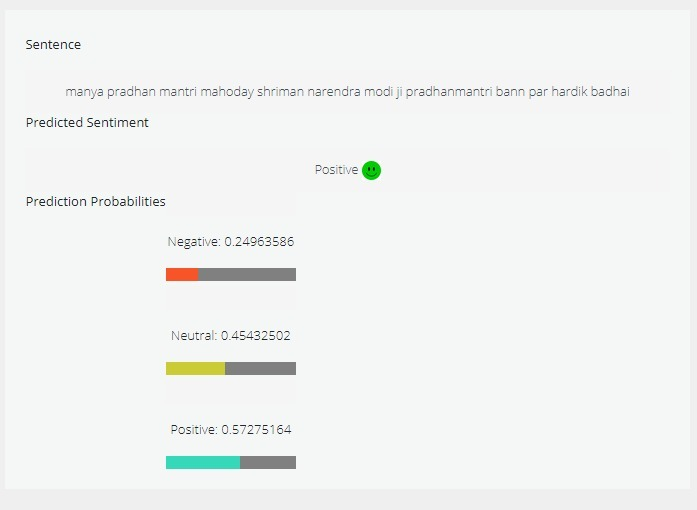
**Output Screenshots:**1. Website UI

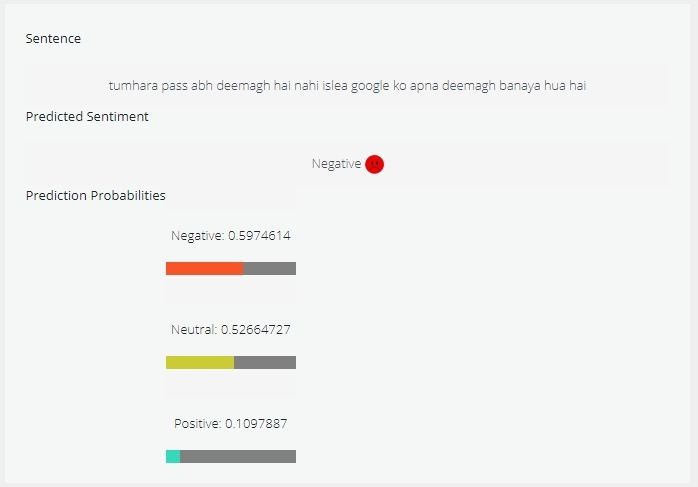




2. Example results of demo sentences: Neutral,Positive and Negative Sentences







**References:**

[1] Bhargava, Rupal, Yashvardhan Sharma, and Shubham Sharma. "Sentiment analysis for mixed script indic sentences." *2016 International conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2016.

[2] Sharma, Shashank, P. Y. K. L. Srinivas, and Rakesh Chandra Balabantaray. "Text normalization of code mix and sentiment analysis." *2015 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2015.

[3] Sitaram, Dinkar, et al. "Sentiment analysis of mixed language employing Hindi-English code switching." *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*. Vol. 1. IEEE, 2015.

[4]Prabhu, Ameya, et al. "Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text." *arXiv preprint arXiv:1611.00472* (2016).

[5] Balahur, Alexandra, et al. "Resource Creation and Evaluation for Multilingual Sentiment Analysis in Social Media Texts." *LREC*. 2014.