NLP J COMP

TestToon

Team MT5:

1. **Guttula Dhanush Ram (20BCE2263)**
2. Prakash Kumar (20BCE0080)
3. Samriddh Prasad (20BCE0131)
4. Clifford Christopher (20BCE2352)

Fields impacted: Small scale business, Regional Language

Colab Link:

1. Train a question generator using T5 transformer model
https://drive.google.com/file/d/17x9nssfMEUIaRC5eDpdtwmSgFJZBJ5WC/view?usp=sharing
2. MCQs Generator
https://drive.google.com/file/d/1wUy2quqiJi5szhYFZpTMNjSFZ3zwR2rB/view?usp=sharing
3. Fill in the Blanks ques Generator
https://drive.google.com/file/d/1BxlHx5w5nFwvzqq3K5QCVaJWX1AN7Rx0/view?usp=sharing

Github Link of Project:
https://github.com/PrakashGavel/NLP-Project

Corpus:
Hindi Corpus:
https://wortschatz.uni-leipzig.de/en/download/Hindi

QA_Dataset:
https://www.kaggle.com/datasets/thedevastator/the-stanford-question-answering-dataset

References:

1. https://github.com/google-research/multilingual-t5

2. https://github.com/ramsrigouthamg/Questgen.ai/tree/master/Questgen

3. https://gist.github.com/avidale/44cd35bfcdaf8bedf51d97c468cc8001

4. https://github.com/KrishnanJothi/MT5_Language_identification_NLP

Description of our project:

The utilization of Multilingual T5-translation, summarization, and language classification can aid in the creation of MCQs and fill-in-the-blanks from regional language text paragraphs. This involves natural language processing algorithms to identify key concepts and generate relevant questions. This can benefit individuals who are not proficient in the regional language, as it improves their language skills and understanding of the content. Additionally, educators and researchers can use this method to evaluate the comprehension and retention of their students or participants. This can be especially helpful for small-scale tuition educators to create assessments.

Section 1: For T5 Traning

Training T5 model:

```
# Training the Model.
args_dict = dict(
    batch_size=4,
)

args = argparse.Namespace(**args_dict)


model = T5FineTuner(args,t5_model,t5_tokenizer)  # model traning

trainer = pl.Trainer(max_epochs = 1, gpus=1,progress_bar_refresh_rate=30)

trainer.fit(model)

print ("Saving model")
save_path_model = '/content/gdrive/My Drive/T5/model/'
save_path_tokenizer = '/content/gdrive/My Drive/T5/tokenizer/'
model.model.save_pretrained(save_path_model)
t5_tokenizer.save_pretrained(save_path_tokenizer)
# after this go to t5/model folder in google drive, where the trained model w
ill be stored and t5/tokenizer folder where our all the tokens will be saved.
```

Test the trained T5 model:

```
[ ] context ="President Donald Trump said and predicted that some states would reopen this month."
    answer = "Donald Trump"
    text = "context: "+context + " " + "answer: " + answer + " </s>"
    print (text)

    context: President Donald Trump said and predicted that some states would reopen this month. answer: Donald Trump </s>
```

Function:

```
model.eval()
beam_outputs = model.generate(
    input_ids=input_ids,attention_mask=attention_mask,  # input: context+answ
er as tokens (input_ids)
    max_length=72,
    early_stopping=True,
    num_beams=5,
    num_return_sequences=3  # top 3 question will be generated.

)

for beam_output in beam_outputs:
    sent = tokenizer.decode(beam_output, skip_special_tokens=True,clean_up_to
kenization_spaces=True)
    print (sent)
```

Result:

```
/usr/local/lib/python3.9/dist-packages/torch/_tensor.py:575: UserWarning: floor
To keep the current behavior, use torch.div(a, b, rounding_mode='trunc'), or fo
  return torch.floor_divide(self, other)
question: Who predicted that some states would reopen this month?
question: Who predicted some states would reopen this month?
question: Who said some states would reopen this month?
```

Section 2: MCQs

2.1 Generate Keywords (We will use these keywords as correct_answer of MCQs).

```
from flashtext import KeywordProcessor


def get_keywords(originaltext,summarytext):
  keywords = get_nouns_multipartite(originaltext)
  print ("keywords unsummarized: ",keywords)
  keyword_processor = KeywordProcessor()
  for keyword in keywords:
    keyword_processor.add_keyword(keyword)

  keywords_found = keyword_processor.extract_keywords(summarytext)
  keywords_found = list(set(keywords_found))
  print ("keywords_found in summarized: ",keywords_found)

  important_keywords =[]
  for keyword in keywords:
```

```
    if keyword in keywords_found:
      important_keywords.append(keyword)

  return important_keywords[:4]


imp_keywords = get_keywords(text,summarized_text)
print (imp_keywords)
```

Result:

```
keywords unsummarized:  ['elon musk', 'dogecoin', 'bitcoin', 'statements', 'tweets', 'transaction efficiency', 'cryptocurrency', 'vehicle maker tesla', 'currency market',
keywords_found in summarized:  ['elon musk', 'musk', 'world', 'bitcoin', 'dogecoin', 'cryptocurrency']
['elon musk', 'dogecoin', 'bitcoin', 'cryptocurrency']
time: 1.39 s (started: 2023-04-15 18:45:19 +00:00)
```

2.2 Generate Questions from Context and Answers.

```
def get_question(context,answer,model,tokenizer):
  text = "context: {} answer: {}".format(context,answer)
  encoding = tokenizer.encode_plus(text,max_length=384, pad_to_max_length=Fal
se,truncation=True, return_tensors="pt").to(device)
  input_ids, attention_mask = encoding["input_ids"], encoding["attention_mask
"]

  outs = model.generate(input_ids=input_ids,
                                  attention_mask=attention_mask,
                                  early_stopping=True,
                                  num_beams=5,
                                  num_return_sequences=1,
                                  no_repeat_ngram_size=2,
                                  max_length=72)


  dec = [tokenizer.decode(ids,skip_special_tokens=True) for ids in outs]


  Question = dec[0].replace("question:","")
  Question= Question.strip()
  return Question



for wrp in wrap(summarized_text, 150):
  translater = Translator()
  hwrp = translater.translate(wrp, dest="hi")
```

```
  print (hwrp.text)
print ("\n")

# for answer in imp_keywords:
#   ques = get_question(summarized_text,answer,question_model,question_tokenizer)
#   print (ques)
#   print (answer.capitalize())
#   print ("\n")

for answer in imp_keywords:
  ques = get_question(summarized_text,answer,question_model,question_tokenizer)
  translater = Translator()
  hques = translater.translate(ques, dest="hi")
  print(hques.text)
  hanswer = translater.translate(answer, dest="hi")
  print(hanswer.text)
  print ("\n")
```

Result:

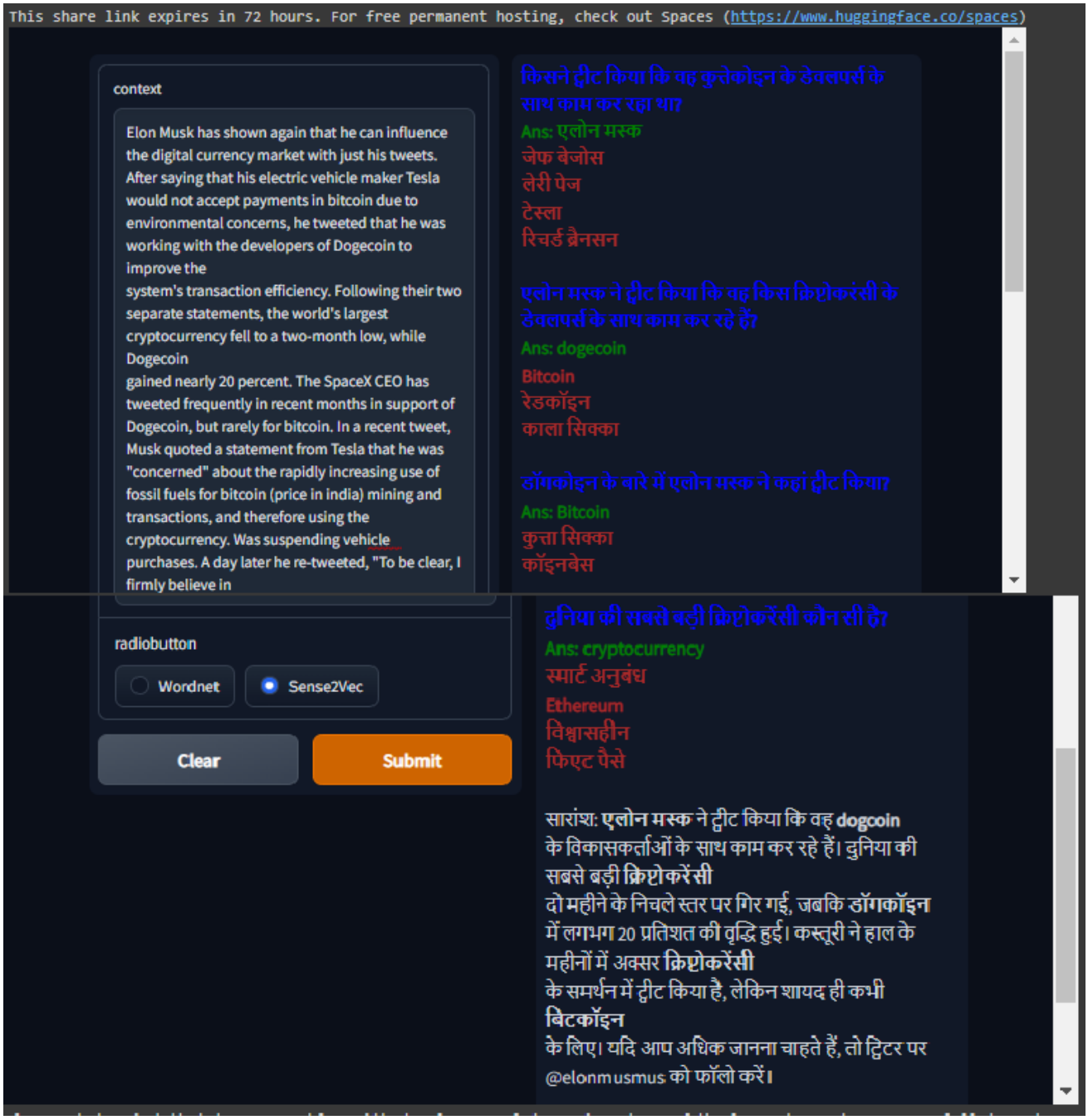
```
एलोन मस्क ने ट्वीट किया कि वह डॉगकॉइन के डेवलपर्स के साथ काम कर रहे हैं। दुनिया की सबसे बड़ी क्रिप्टोकरेंसी दो महीने के निचले स्तर पर आ गई, जबकि डॉगकॉइन में बढ़त हुई
लगभग 20 प्रतिशत। कस्तूरी ने हाल के महीनों में क्रिप्टोक्यूरेंसी के समर्थन में अक्सर ट्वीट किया है, लेकिन बिटकॉइन के लिए शायद ही कभी। यदि आप और जानना चाहते हैं,
ट्विटर पर @elonmusmus को फॉलो करें।


किसने ट्वीट किया कि वह कुत्तेकोइन के डेवलपर्स के साथ काम कर रहा था?
एलोन मस्क


एलोन मस्क ने ट्वीट किया कि वह किस क्रिप्टोकरंसी के डेवलपर्स के साथ काम कर रहे हैं?
dogecoin


डॉगकोइन के बारे में एलोन मस्क ने कहां ट्वीट किया?
Bitcoin


दुनिया की सबसे बड़ी क्रिप्टोकरेंसी कौन सी है?
cryptocurrency
```

2.3 Gradio Visualization of MCQs

```
import gradio as gr
from googletrans import Translator
translater = Translator()

context = gr.inputs.Textbox(lines=10, placeholder="Enter paragraph/content here...")
output = gr.outputs.HTML(label="Question and Answers")
radiobutton = gr.inputs.Radio(["Wordnet", "Sense2Vec"])
```

```
def generate_question(context, radiobutton):
    summary_text = summarizer(context, summary_model, summary_tokenizer)
    for wrp in wrap(summary_text, 150):
        print(wrp)
    np = get_keywords(context, summary_text)
    print("\n\nNoun phrases", np)
    output = ""
    translater = Translator()
    for answer in np:
        ques = get_question(summary_text, answer, question_model, question_to
kenizer)
        if radiobutton == "Wordnet":
            distractors = get_distractors_wordnet(answer)
        else:
            distractors = get_distractors(answer.capitalize(), ques, s2v, sen
tence_transformer_model, 40, 0.2)

        hques = translater.translate(ques, dest="hi")
        output += "<b style='color:blue;'>" + hques.text + "</b><br>"

        hanswer = translater.translate(answer, dest="hi")
        output += "<b style='color:green;'>" + "Ans: " + hanswer.text + "</b>
<br>"

        if len(distractors) > 0:
            for distractor in distractors[:4]:
                hdist = translater.translate(distractor, dest="hi")
                output += "<b style='color:brown;'>" + hdist.text + "</b><br>
"
        output += "<br>"

    summary = "Summary: " + summary_text
    for answer in np:
        summary = summary.replace(answer, "<b>" + answer + "</b><br>")
        summary = summary.replace(answer.capitalize(), "<b>" + answer.capital
ize() + "</b>")
        translater = Translator()

    hsum = translater.translate(summary, dest="hi")
    output += "<p>" + hsum.text + "</p><br>"
    return output


iface = gr.Interface(
    fn=generate_question,
    inputs=[context, radiobutton],
    outputs=output)
iface.launch(debug=True)
```

Result:


This share link expires in 72 hours. For free permanent hosting, check out Spaces (https://www.huggingface.co/spaces)
context
Elon Musk has shown again that he can influence the digital currency market with just his tweets. After saying that his electric vehicle maker Tesla would not accept payments in bitcoin due to environmental concerns, he tweeted that he was working with the developers of Dogecoin to improve the system's transaction efficiency. Following their two separate statements, the world's largest cryptocurrency fell to a two-month low, while Dogecoin gained nearly 20 percent. The SpaceX CEO has tweeted frequently in recent months in support of Dogecoin, but rarely for bitcoin. In a recent tweet, Musk quoted a statement from Tesla that he was "concerned" about the rapidly increasing use of fossil fuels for bitcoin (price in india) mining and transactions, and therefore using the cryptocurrency. Was suspending vehicle purchases. A day later he re-tweeted, "To be clear, I firmly believe in
radiobutton
Wordnet
Sense2Vec
Clear
Submit
किसने ट्वीट किया कि वह कुत्तेकोइन के डेवलपर्स के साथ काम कर रहा था?
Ans: एलोन मस्क
जेफ बेजोस
लेरी पेज
टेस्ला
रिचर्ड ब्रैनसन
एलोन मस्क ने ट्वीट किया कि वह किस क्रिप्टोकरेंसी के डेवलपर्स के साथ काम कर रहे हैं?
Ans: dogecoin
Bitcoin
रेडकॉइन
काला सिक्का
डॉगकोइन के बारे में एलोन मस्क ने कहां ट्वीट किया?
Ans: Bitcoin
कुत्ता सिक्का
कॉइनबेस
दुनिया की सबसे बड़ी क्रिप्टोकरेंसी कौन सी है?
Ans: cryptocurrency
स्मार्ट अनुबंध
Ethereum
विश्वासहीन
फिएट पैसे
सारांश: एलोन मस्क ने ट्वीट किया कि वह dogcoin के विकासकर्ताओं के साथ काम कर रहे हैं। दुनिया की सबसे बड़ी क्रिप्टोकरेंसी दो महीने के निचले स्तर पर गिर गई, जबकि डॉगकॉइन में लगभग 20 प्रतिशत की वृद्धि हुई। कस्तूरी ने हाल के महीनों में अक्सर क्रिप्टोकरेंसी के समर्थन में ट्वीट किया है, लेकिन शायद ही कभी बिटकॉइन के लिए। यदि आप अधिक जानना चाहते हैं, तो ट्विटर पर @elonmusmus को फॉलो करें।

```
Elon musk tweeted that he was working with developers of dogecoin. The world's largest cryptocurrency fell to a two-month low, while dogecoin gained
nearly 20 percent. Musk has tweeted frequently in recent months in support of the cryptocurrency, but rarely for bitcoin. If you want to know more,
follow @elonmusmus on twitter.
keywords unsummarized:  ['elon musk', 'dogecoin', 'bitcoin', 'statements', 'tweets', 'cryptocurrency', 'vehicle maker tesla', 'transaction efficiency', 'currency market', 'fuels', 'musk', 'world',
keywords_found in summarized:  ['elon musk', 'musk', 'world', 'bitcoin', 'dogecoin', 'cryptocurrency']


Noun phrases ['elon musk', 'dogecoin', 'bitcoin', 'cryptocurrency']
word  Elon musk
PERSON
Similar  ['Musk', 'Elon', 'Richard Branson', 'Bill Gates', 'Jeff Bezos', 'Mark Zuckerberg', 'Larry Page', 'Mr. Musk', 'Tesla', 'Steve Jobs', 'Warren Buffett', 'Eric Schmidt', 'Warren Buffet']
distractors  ['Musk', 'Elon', 'Richard Branson', 'Bill Gates', 'Jeff Bezos', 'Mark Zuckerberg', 'Larry Page', 'Mr. Musk', 'Tesla', 'Steve Jobs', 'Warren Buffett', 'Eric Schmidt']
word  Dogecoin
PERSON
Similar  ['Litecoin', 'Bitcoin', 'Reddcoin', 'Blackcoin']
distractors  ['Bitcoin', 'Reddcoin', 'Blackcoin']
word  Bitcoin
PERSON
Similar  ['Litecoin', 'Bitcoins', 'Dogecoin', 'Bitcoin', 'Coinbase']
distractors  ['Dogecoin', 'Coinbase']
word  Cryptocurrency
NOUN
Similar  ['Cryptocurrency', 'Digital Currency', 'Bitcoin', 'Cryptocurrencies', 'Blockchain', 'Cryptocurrencies', 'Crypto Currency', 'Crypto', 'Blockchain Technology', 'Crypto-Currencies', 'Bitcoin',
distractors  ['Digital Currency', 'Bitcoin', 'Blockchain', 'Crypto', 'Blockchain Technology', 'Trustless', 'Smart Contracts', 'Ethereum', 'Fiat Money', 'Bitreserve', 'Crypto World', 'Bitcoin World',
```

Section 3: Fill in the Blanks.

Text Data:

```
htext = """महासागर में अपसारी प्लेट सीमाओं पर अत्यधिक ज्वालामुखी गतिविधि ह
ोती है। उदाहरण के लए, मध्य-अटलांटक के साथ-साथ कई अिर्लीय ज्वालामुखी पाए
र्ाि े हैं
ररर्। यह अपसारी प्लेट सीमा है र्  अटलांटक महासागर के मध्य
से उत्तर- दक्षिण की ओर चलिी है। र्ैसे टेक्टक्षिक
प्लेट्स एक दूसरे से दूर खीचिी हैं
अपसारी प्लेट सीमा पर, िे पपडी में गहरी दरारें, या दरारें बिाि े हैं।
क्षपघलती हुई चट्टाि, क्षरसे मैग्मा कहा र्ा
िा है, इि दरारों के माध्यम से पृथ्वी पर फू टिी है
सिह। सिह पर क्षपघलती हुई चट्टाि क लाि ा कहा र्ाि ा है। यह ठडा ह कर कठर ह
र्ाि ा है, क्षरससे चट्टाि बि र्ाि ी है। अपसारी प्लेट सीमाएँ भी महाद्वीपीय में पाई र्ाि ी हैं
पपडी। इि सीमाओ पर ज्वालामुखी बिि े हैं, लेकि समुद्र की पपडी की
िुलि ा में कम। ऐसा इसलए है क् धक महाद्वीपीय क्रस्ट महासागरीय क्रस्ट से अधिक मटा है।
यह
क्षपघलती हुई चट्टाि क पपडी के माध्यम से ऊपर िके लि ा अधिक कधठि बि ा
देि ा है। कई ज्वालामुखी अधभसरण प्लेट सीमाओ के साथ बिि े हैं रहा एक
टेक्टक्षिक प्लेट ह ि ी है
सबडक्शि र्ि में दूसरे के िीचे खीचा र्ाि ा है। प्लेट का अग्रणी
```

3.1 Keyword Extraction using MultipartiteRank

```
# Extracting keywords like nouns, verbs, adjectives.
def get_noun_adj_verb(text):
    out=[]
    try:
        extractor = pke.unsupervised.MultipartiteRank()
        extractor.load_document(input=text,language='en')
        #    not contain punctuation marks or stopwords as candidates.
        pos = {'VERB', 'ADJ', 'NOUN'}
        stoplist = list(string.punctuation)
```

```
        stoplist += ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
        stoplist += stopwords.words('english')
        # extractor.candidate_selection(pos=pos, stoplist=stoplist)
        extractor.candidate_selection(pos=pos)
        # 4. build the Multipartite graph and rank candidates using random wa
lk,
        #    alpha controls the weight adjustment mechanism, see TopicRank fo
r
        #    threshold/method parameters.
        extractor.candidate_weighting(alpha=1.1,
                                      threshold=0.75,
                                      method='average')
        keyphrases = extractor.get_n_best(n=30)


        for val in keyphrases:
            out.append(val[0])
    except:
        out = []
        traceback.print_exc()

    return out

noun_verbs_adj = get_noun_adj_verb(text)
print ("keywords: ",noun_verbs_adj)
```

Result:

```
keywords:  ['divergent plate boundaries', 'molten rock', 'called magma', 'form rock', 'crust', 'oceans', 'create deep cracks'
```

3.2 Deleting the keywords and replacing those with the blanks.

```
# finding the longest sentence for each keyword and replace those keywords wi
th blanks.
def get_fill_in_the_blanks(sentence_mapping):
    out={"title":"Fill in the blanks for these sentences with matching words
at the top"}
    blank_sentences = []
    processed = []
    keys=[]
    for key in sentence_mapping:
        if len(sentence_mapping[key])>0:
            sent = sentence_mapping[key][0]
            # Compile a regular expression pattern into a regular expression
object, which can be used for matching and other methods
            insensitive_sent = re.compile(re.escape(key), re.IGNORECASE)
```

```
            no_of_replacements =  len(re.findall(re.escape(key),sent,re.IGNOR
ECASE))
            line = insensitive_sent.sub(' _____', sent)
            if (sentence_mapping[key][0] not in processed) and no_of_replacem
ents<2:
                blank_sentences.append(line)
                processed.append(sentence_mapping[key][0])
                keys.append(key)
    out["sentences"]=blank_sentences[:10]
    out["keys"]=keys[:10]
    return out


fill_in_the_blanks = get_fill_in_the_blanks(keyword_sentence_mapping_noun_ver
bs_adj)
pprint(fill_in_the_blanks)
```

Result:

```
    {'keys': ['divergent plate boundaries',
              'molten rock',
              'called magma',
              'form rock',
              'crust',
              'surface',
              'erupts',
              'volcanoes form',
              'found',
              'oceanic crust'],
     'sentences': ['Extreme volcanic activity occurs at  _____  in the oceans.',
                   'This\n'
                   'This makes it more difficult to push  _____  up through '
                   'the crust.',
                   'Molten rock,  _____ , erupts into the earth through these '
                   'cracks.',
                   'It cools and hardens to  _____ .',
                   'as tectonic plates pull away from each other\n'
                   'At the divergent plate boundary, they create deep cracks, or '
                   'fissures, in the  _____ .',
                   'The molten rock on the  _____  is called lava.',
                   'The leading edge of the plate melts as it is pulled into the '
                   'mantle, forming magma that  _____ \n'
                   'Volcano.',
                   'Many  _____  along convergent plate boundaries where a '
                   'tectonic plate\n'
                   'One is pulled beneath the other in a subduction zone.',
                   'For example, many underwater volcanoes are  _____  along '
                   'the Mid-Atlantic.',
                   'Volcanoes form at these boundaries, but less so than in the  '
                   '_____ .'],
     'title': 'Fill in the blanks for these sentences with matching words at the '
              'top'}
```

3.3 Visualization:

```
# Simple Visualization of above things and randomize the order of keyword(ans
wer).
from IPython.core.display import display, HTML
import xml.etree.ElementTree as et
import random

root = et.Element("div")

heading = et.Element("h2")
heading.text = fill_in_the_blanks['title']

keywords = et.Element("ul")
keywords.set('style', 'color:fuchsia;')

all_keys = fill_in_the_blanks['keys']
random.shuffle(all_keys)
for blank in all_keys:
  child=et.Element("li")
  child.text = blank
  keywords.append(child)

sentences = et.Element("ol")
sentences.set('style', 'color:yellow;')
for sentence in fill_in_the_blanks['sentences']:
  child=et.Element("li")
  child.text = sentence
  sentences.append(child)
  sentences.append(et.Element("br"))

heading_content = et.Element("h4")

root.append(heading)
heading_content.append(keywords)
heading_content.append(sentences)
root.append(heading_content)

xmlstr = et.tostring(root)
xmlstr = xmlstr.decode("utf-8")
# display(HTML(xmlstr))

from googletrans import Translator
translater = Translator()
text1 = translater.translate(xmlstr, dest="hi")
xx = text1.text
display(HTML(xx))
```

Result:

इन वाक्यों के लिए रिक्त स्थानों की पूर्ति शीर्ष पर मिलते-जुलते शब्दों से करें

- पिघली हुई चट्टान
- सतह
- पपड़ी
- फटती है
- ज्वालामुखी बनते हैं
- अपसारी प्लेट सीमाएँ
- मिली
- महासागरीय पपड़ी
- जिसे मैग्मा कहा जाता है
- चट्टान का रूप

1. चरम ज्वालामुखीय गतिविधि _____ में होती है महासागर।
2. यह इससे _____ को पपड़ी के माध्यम से ऊपर धकेलना अधिक कठिन हो जाता है।
3. पिघला हुआ चट्टान, _____, इन दरारों के माध्यम से पृथ्वी में प्रस्फुटित होता है।
4. यह ठंडा होता है और _____ तक कठोर हो जाता है।
5. जैसे टेक्टोनिक प्लेट्स एक दूसरे से दूर खींचती हैं अपसारी प्लेट सीमा पर, वे _____ में गहरी दरारें, या फिशर बनाते हैं।
6. _____ पर पिघली हुई चट्टान को लावा कहा जाता है।
7. प्लेट का अग्रणी किनारा पिघल जाता है क्योंकि इसे मेंटल में खींच लिया जाता है, जिससे मैग्मा बनता है जो कि _____ ज्वालामुखी।
8. कई _____ अभिसरण प्लेट सीमाओं के साथ जहां एक टेक्टोनिक प्लेट सबडक्शन जोन में एक को दूसरे के नीचे खींचा जाता है।
9. उदाहरण के लिए, कई पानी के नीचे के ज्वालामुखी मध्य-अटलांटिक के साथ-साथ _____ हैं।
10. ज्वालामुखी यहां बनते हैं ये सीमाएँ, लेकिन _____ की तुलना में कम।

Problem-faced:

Question and answers are not generating in regional language.

```
  Question = dec[0].replace("question:","")
  Question= Question.strip()
  return Question



for wrp in wrap(summarized_text, 150):
  print (wrp)
print ("\n")

for answer in imp_keywords:
  ques = get_question(summarized_text,answer,question_model,question_tokenizer)
  print (ques)
  print (answer.capitalize())
  print ("\n")
```

```
Elon musk tweets that he is working with developers to improve transaction efficiency. The
while dogecoin there was an increase of about 20 percent. Many twitter users welcomed the
crypto is "here to stay" the spacex ceo has tweeted frequently in recent months in support


Who tweeted that he is working with developers to improve transaction efficiency?
Elon musk


What was the largest cryptocurrency in the world?
Dogecoin


What has spacex ceo rarely tweeted about?
Bitcoin


What does Elon musk do?
Tweets
```

Problem-resolved:

Question and answers are generating in regional language.

```
for answer in imp_keywords:
  ques = get_question(summarized_text,answer,question_model,question_tokenizer)
  translater = Translator()
  hques = translater.translate(ques, dest="hi")
  print(hques.text)
  hanswer = translater.translate(answer, dest="hi")
  print(hanswer.text)
  print ("\n")
```

```
किसने ट्वीट किया कि वह लेन-देन दक्षता में सुधार के लिए डेवलपर्स के साथ काम कर रहा है?
एलोन मस्क


दुनिया की सबसे बड़ी क्रिप्टोकरेंसी कौन सी थी?
dogecoin


स्पेसएक्स के सीईओ ने किस बारे में शायद ही कभी ट्वीट किया हो?
Bitcoin


एलोन मस्क क्या करते हैं?
ट्वीट्स
```