BÁO CÁO
BÀI THỰC HÀNH 1

1. Bài toán: Nhận diện bình luận độc hại trên mạng xã hội

2. Các thành viên:

+ Phạm Trung Tín (21522678)
+ Trần Nguyễn Yến Nhi (21522429)
+ Nguyễn Việt Quang (21522515)

3. Nội dung báo cáo:

3.1. Thu thập dữ liệu

- Tiến hành thu thập các bình luận của các video trên Youtube. Kết quả thu được 4483 mẫu dữ liệu thô.
- Code để crawl data:

```
import os
import pandas as pd
from csv import writer
from googleapiclient.discovery import build
viet = open("datayt4.csv", mode='w', encoding='utf-8-sig', newline = '')
viet.write("Comment\n")
api_key = 'AIzaSyBVPXs5YCHpiZb34AqIvsCOH4cv3zE3-28'
youtube = build('youtube', 'v3', developerKey=api_key)
video_id = 'W5_G2OSuG9E'
next_page_token = None
comments_list =[]
def listToString(s):
    str1 = ""
    for ele in s:
        str1 += ele
    return str1
while True:
  results = youtube.commentThreads().list(
    part='snippet,replies',
    videoId=video_id,
    textFormat='plainText',
    order='relevance',
    pageToken = next_page_token
  ).execute()
  for item in results['items']:
    comment = item['snippet']['topLevelComment']['snippet']
    text = comment['textDisplay']
    texti=['']
    for t in text:
      if t==',' or t=="\n"or t=="\n\n":
        t=""
      texti.append(t)
```

```
            print(listToString(texti))
            w = listToString(texti) + '\n'
            viet.write(w)
        next_page_token = results.get('nextPageToken')
        if not next_page_token:
            break
```

3.2. Tiền xử lý dữ liệu

Gồm các bước:

+ Bỏ trùng lặp
+ Bỏ các ký tự đặc biệt, emoji, dấu chấm,phẩy,… trong câu
+ Loại bỏ stopword
+ word segment
+ tokenize
+ Padding, attention mask

Code:

```
from vncorenlp import VnCoreNLP
rdrsegmenter = VnCoreNLP("/content/drive/MyDrive/ML/vncorenlp/VnCoreNLP-1.1.1.jar", annotators="wseg",
max_heap_size='-Xmx500m')
def standardize_data(row):
    # Xóa dấu chấm, phẩy, hỏi ở cuối câu
    row = re.sub(r"[\.,\?]+$-", "", row)
    # Xóa tất cả dấu chấm, phẩy, chấm phẩy, chấm thang, ... trong câu
    row = row.replace(",", " ").replace(".", " ") \
        .replace(";", " ").replace("“", " ") \
        .replace(":", " ").replace("”", " ") \
        .replace('"', " ").replace("'", " ") \
        .replace("!", " ").replace("?", " ") \
        .replace("=", " ").replace(")", " ") \
        .replace(">", " ").replace("(", " ") \
        .replace("<3", " ").replace("#", " ") \
        .replace("^", " ").replace("%", " ") \
        .replace("[", " ").replace("]", " ") \
        .replace("{", " ").replace("}", " ") \
        .replace("-", " ").replace("<", " ").replace("~", " ")
    return row

def load_stopwords():
    sw = []
    with open('/content/drive/MyDrive/ML/vietnamese-stopwords.txt', encoding='utf-8') as f:
        lines = f.readlines()
    for line in lines:
        sw.append(line.replace("\n",""))
    return sw

def remove_stop_words(corpus,b):
    stop_words=b
    tmp = corpus.split()
    for i in tmp:
      if i in b:
        corpus.replace(i,' ')
    return corpus
def strip_emoji(text):
    RE_EMOJI = re.compile(u'([\U00002600-\U000027BF])|([\U0001f300-\U0001f64F])|([\U0001f680-\U0001f6FF])')
    return RE_EMOJI.sub(r' ', text)

def remove_emojis(data):
    emoj = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
```

```
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U00002500-\U00002BEF"  # chinese char
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"  # dingbats
        u"\u3030"
                      "]+", re.UNICODE)
    return re.sub(emoj, ' ', data)
import pandas as pd
import re
import string
train_text= []
texti = []
data= pd.read_csv("/content/drive/MyDrive/datayt_up.csv")
data=data.drop_duplicates()
for i in data.Comment:
  a=strip_emoji(i)
  a=remove_emojis(a)
  for e in a:
    if e in string.punctuation:
      a.replace(e,' ')
  a=standardize_data(a)
  text = rdrsegmenter.tokenize(a)
  text = ' '.join([" ".join(x) for x in text])
  text = text.strip().lower()
  b=load_stopwords()
  text= remove_stop_words(text,b)
  if text=='':
    continue

  train_text.append(text)

import io
out_data= io.open('out_data.csv','w',encoding='utf-8-sig')
out_data.write('Comment'+'\n')
for i in train_text:
  out_data.write(i+'\n')
out_data.close()
import os
from fairseq.models.roberta import RobertaModel
phoBERT = RobertaModel.from_pretrained('PhoBERT_base_fairseq', checkpoint_file='model.pt')
phoBERT.eval()
from fairseq.data.encoders.fastbpe import fastBPE
# Khởi tạo Byte Pair Encoding cho PhoBERT
class BPE():
  bpe_codes = 'PhoBERT_base_fairseq/bpe.codes'
args = BPE()
phoBERT.bpe = fastBPE(args)
from tensorflow.keras.preprocessing.sequence import pad_sequences
MAX_LEN = 50
train_ids = []
train_ec=[]
for sent in train_text:
```

```
  subwords = phoBERT.encode(sent)
  train_ec.append(subwords)
train_ids = pad_sequences(train_ec, maxlen=MAX_LEN, dtype="long", value=0, truncating="post",
padding="post")
train_masks = []
for sent in train_ids:
    mask = [int(token_id > 0) for token_id in sent]
    train_masks.append(mask)
```

Sau khi thực thi code thu được file dữ liệu với 4346 mẫu đã được làm sạch

Link google colab:
https://colab.research.google.com/drive/1r8NkmORDlY2ktl8rav28BxnO9CWuwO8m#scrollTo=XnjC6VhK3Ksq