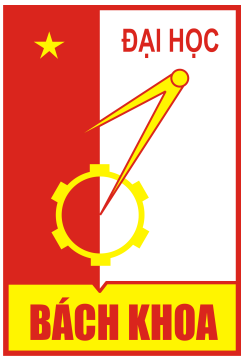
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



BÁO CÁO CUỐI KÌ
HỆ HỖ TRỢ QUYẾT ĐỊNH

ĐỀ TÀI:
XÂY DỰNG MÔ HÌNH PHÂN TÍCH QUAN ĐIỂM ĐÁNH GIÁ PHIM

Giảng viên hướng dẫn: TS. Lê Hải Hà
Sinh viên thực hiện: Đào Bảo Đại
MSSV: 20200126
Mã lớp học: 142300

HÀ NỘI – 2023

Mục lục

Lời mở đầu

Thu thập thông tin phản hồi của khách hàng là một cách tuyệt vời giúp cho các doanh nghiệp hiểu được điểm mạnh, điểm yếu trong sản phẩm, dịch vụ của mình; đồng thời nhanh chóng nắm bắt được tâm ký và nhu cầu khách hàng để mang đến cho họ sản phẩm, dịch vụ hoàn hảo nhất.

Ngày nay, với sự phát triển vượt bậc của khoa học và công nghệ, đặc biệt là sự bùng nổ của Internet với các phương tiện truyền thông xã hội, thương mại điện tử,... đã cho phép mọi người không chỉ chia sẻ thông tin trên đó mà còn thể hiện thái độ, quan điểm của mình đối với các sản phẩm, dịch vụ và các vấn đề xã hội khác. Vì vậy mà Internet đã trở lên vô cùng quan trọng và là nguồn cung cấp một lượng thông tin vô cùng lớn và quan trọng.

Để có thể biết được thái độ, đánh giá và quan điểm của người dùng một cách hiệu quả và nhanh chóng thì ta phải tận dụng phân tích được chính những thông tin mà người dùng để lại qua internet như các bình luận, đánh giá, bài chia sẻ,...

Chương 1

Tổng quan về Sentiment Analysis

1.1 Sentiment Analysis là gì

Sentiment analysis, còn được gọi là phân tích cảm xúc hoặc khai thác ý kiến, là một kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) được sử dụng để xác định cảm xúc biểu đạt trong một đoạn văn bản. Nó bao gồm việc sử dụng các phương pháp tính toán để tự động phân tích dữ liệu văn bản và xác định xem tình cảm là tích cực, tiêu cực, trung lập, hoặc đôi khi là các cảm xúc phức tạp hơn như vui mừng, giận dữ, buồn bã, v.v.

Mục tiêu chính của sentiment analysis là hiểu và giải thích các cảm xúc, thái độ và ý kiến của cá nhân hoặc nhóm đối với một chủ đề cụ thể, sản phẩm, dịch vụ, người, sự kiện hoặc bất kỳ vấn đề nào khác. Phân tích cảm xúc có nhiều ứng dụng trong các ngành công nghiệp khác nhau, bao gồm nghiên cứu thị trường, giám sát mạng xã hội, phân tích phản hồi của khách hàng, quản lý danh tiếng thương hiệu và nhiều ứng dụng khác.

1.2 Phương pháp thực hiện

Phân tích chi tiết (Fine-Grained)

Mô hình phân tích cảm xúc này này giúp xác định được độ chính xác của các tính chất. Các tính chất chính được phân chia thành: rất tích cực, tích cực, trung tính, tiêu cực hoặc rất tiêu cực. Việc phân chia chi tiết như vậy rất phù hợp để đánh giá các cuộc trò chuyện.

Đối với thang điểm đánh giá từ 1 đến 5, bạn có thể coi 1 là rất tiêu cực và 5 là rất tích cực. Hay từ 1 đến 10, bạn có thể coi 1-2 là rất tiêu cực và 9-10 là rất tích cực.

Dựa trên khía cạnh (Aspect-Based)

Trong khi phân tích chi tiết xác định tính chất tổng thể cảm xúc của khách hàng trong cuộc trò chuyện, thì phân tích dựa trên khía cạnh sẽ đi sâu hơn, nhận dạng cụ thể từng khía cạnh trong lời nói của họ.

Chẳng hạn khi khách hàng nói rằng "máy ảnh gặp khó khăn trong điều kiện ánh sáng nhân tạo". Với phân tích dựa trên khía cạnh, chúng ta không chỉ đánh giá được đây là cảm xúc tiêu cực mà còn có thể xác định rằng người dùng đó đã nhận xét tiêu cực về đối tượng "máy ảnh".

Phát hiện cảm xúc (Emotion Detection)

Cảm xúc ở đây có thể bao gồm các sắc thái tức giận, buồn bã, hạnh phúc, thất vọng, sợ hãi, lo lắng, hoảng sợ... Hệ thống phát hiện cảm xúc thường sử dụng từ vựng – một tập hợp các từ truyền tải những cảm xúc nhất định. Một số bộ phân loại nâng cao cũng sử dụng các thuật toán học máy (Machine Learning – ML) mạnh mẽ.

Phân tích ý định (Intent Analysis)

Việc xác định chính xác ý định của người tiêu dùng có thể giúp công ty tiết kiệm thời gian, tiền bạc và công sức, bởi các doanh nghiệp có thể không tiếp tục chăm sóc những khách hàng không tiềm năng và chưa có kế hoạch mua hàng. Phân tích ý định chính xác có thể giải quyết nhiều vấn đề cho doanh nghiệp như vậy.

Phân tích ý định giúp doanh nghiệp xác định mục đích của người tiêu dùng – cho dù khách hàng có ý định mua hàng hay chỉ đang lướt qua. Nếu khách hàng sẵn sàng mua hàng, bạn có thể theo dõi họ và nhắm mục tiêu họ bằng các quảng cáo. Nếu người tiêu dùng chưa sẵn sàng mua, chúng ta có thể tiết kiệm thời gian và nguồn lực bằng cách không quảng cáo cho họ.

1.3 Quy trình thực hiện trong Sentiment Analysis

- **Tiền xử lý văn bản:** Loại bỏ các thành phần không cần thiết khỏi văn bản, chẳng hạn như ký tự đặc biệt, số và dấu câu, và chuyển đổi văn bản thành chữ thường.
- **Tách từ (Tokenization):** Chia văn bản thành các từ hoặc "token" riêng lẻ.

- **Gán cảm xúc:** Gán điểm cảm xúc cho mỗi từ hoặc token trong văn bản bằng cách sử dụng từ điển cảm xúc hoặc mô hình học máy.
- **Tổng hợp:** Kết hợp các điểm cảm xúc cá nhân để có được điểm cảm xúc tổng thể cho toàn bộ văn bản.
- **Xử lý sau khi phân tích:** Xử lý phủ định, cường điệu và các đặc điểm ngôn ngữ khác có thể ảnh hưởng đến cảm xúc của một câu.

Phân tích cảm xúc có thể cung cấp thông tin quý giá để doanh nghiệp đưa ra quyết định dựa trên dữ liệu, hiểu ý kiến của khách hàng và phản hồi một cách hiệu quả. Tuy nhiên, cần lưu ý rằng phân tích cảm xúc không luôn hoàn hảo, vì ngôn ngữ và ngữ cảnh có thể phức tạp và mơ hồ. Do đó, cần cẩn thận xác minh và giải thích kết quả một cách cẩn thận.

Chương 2

Giới thiệu bài toán

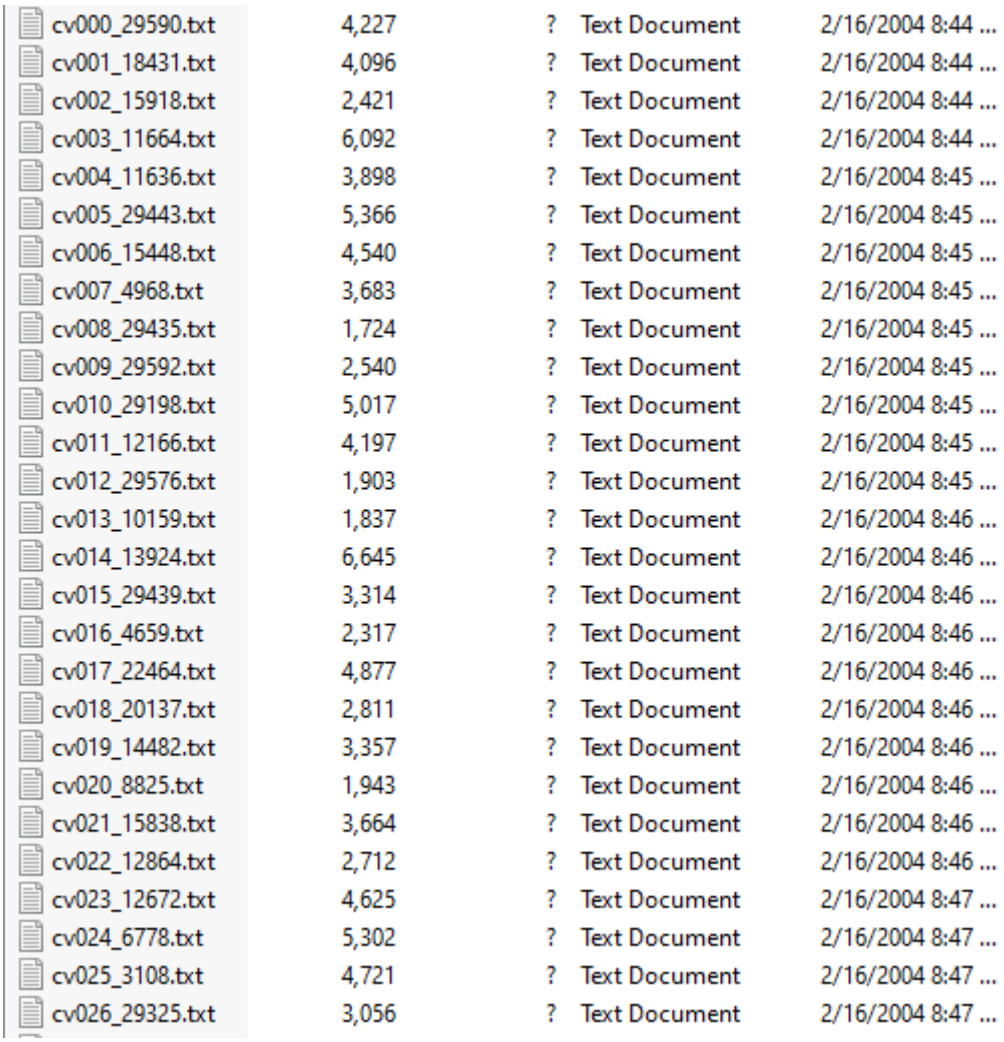
2.1 Bài toán

Dựa vào các đánh giá từ người xem phim, ta cần xây dựng một mô hình phân tích cảm xúc để phân loại xem đó là đánh giá tích cực hay đánh giá tiêu cực.

2.2 Bộ dữ liệu

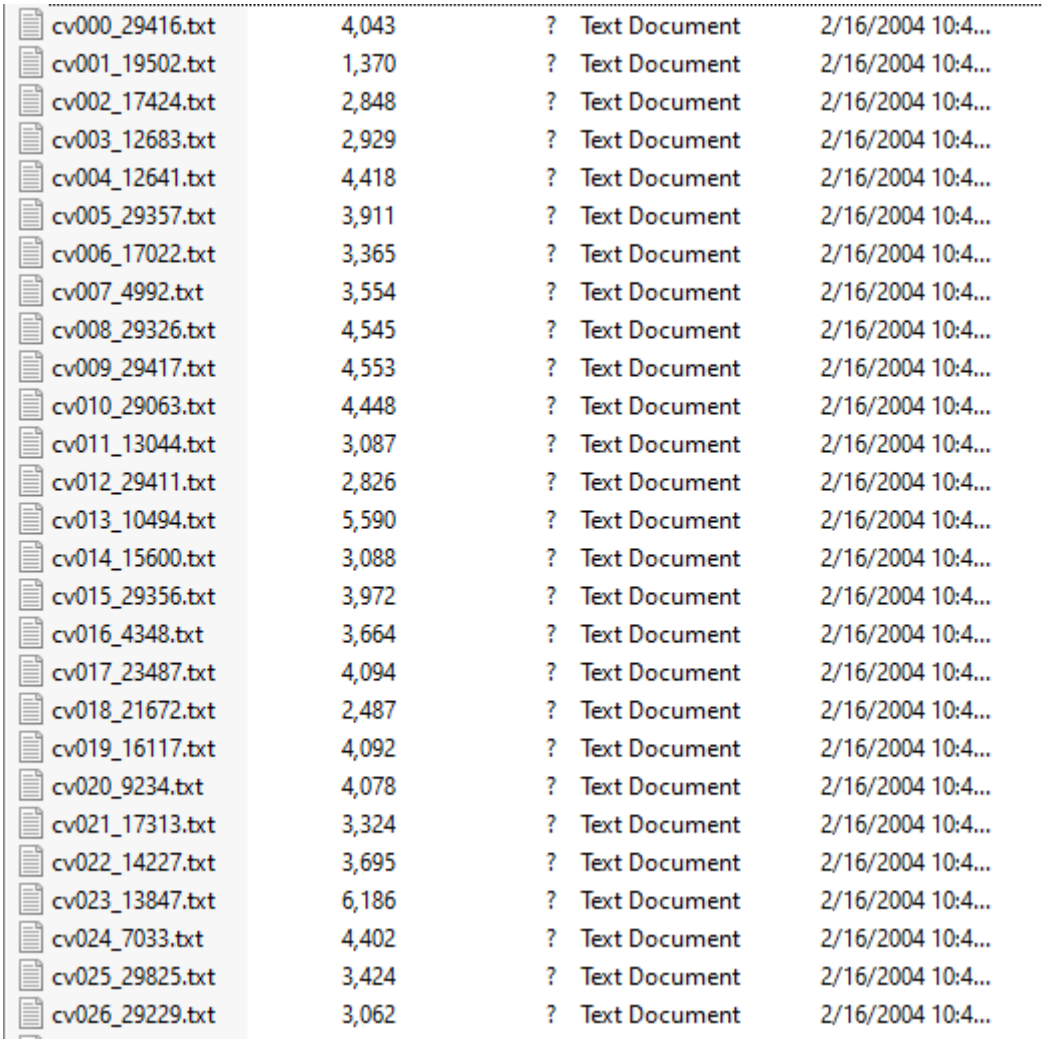
Bộ dữ liệu dùng cho bài toán trên được lấy từ website của IMDb với 1000 bài đánh giá tích cực và 1000 bài đánh giá tiêu cực.

- Bộ dữ liệu chỉ bao gồm các bài đánh giá bằng tiếng Anh.
- Tất cả các văn bản đều được chuyển thành chữ thường.
- Có khoảng trắng xung quanh các dấu câu như dấu chấm, dấu phẩy, dầu ngoặc.
- Mỗi câu nằm trên đúng một dòng.



cv000_29590.txt	4,227	?	Text Document	2/16/2004 8:44 ...
cv001_18431.txt	4,096	?	Text Document	2/16/2004 8:44 ...
cv002_15918.txt	2,421	?	Text Document	2/16/2004 8:44 ...
cv003_11664.txt	6,092	?	Text Document	2/16/2004 8:44 ...
cv004_11636.txt	3,898	?	Text Document	2/16/2004 8:45 ...
cv005_29443.txt	5,366	?	Text Document	2/16/2004 8:45 ...
cv006_15448.txt	4,540	?	Text Document	2/16/2004 8:45 ...
cv007_4968.txt	3,683	?	Text Document	2/16/2004 8:45 ...
cv008_29435.txt	1,724	?	Text Document	2/16/2004 8:45 ...
cv009_29592.txt	2,540	?	Text Document	2/16/2004 8:45 ...
cv010_29198.txt	5,017	?	Text Document	2/16/2004 8:45 ...
cv011_12166.txt	4,197	?	Text Document	2/16/2004 8:45 ...
cv012_29576.txt	1,903	?	Text Document	2/16/2004 8:45 ...
cv013_10159.txt	1,837	?	Text Document	2/16/2004 8:46 ...
cv014_13924.txt	6,645	?	Text Document	2/16/2004 8:46 ...
cv015_29439.txt	3,314	?	Text Document	2/16/2004 8:46 ...
cv016_4659.txt	2,317	?	Text Document	2/16/2004 8:46 ...
cv017_22464.txt	4,877	?	Text Document	2/16/2004 8:46 ...
cv018_20137.txt	2,811	?	Text Document	2/16/2004 8:46 ...
cv019_14482.txt	3,357	?	Text Document	2/16/2004 8:46 ...
cv020_8825.txt	1,943	?	Text Document	2/16/2004 8:46 ...
cv021_15838.txt	3,664	?	Text Document	2/16/2004 8:46 ...
cv022_12864.txt	2,712	?	Text Document	2/16/2004 8:46 ...
cv023_12672.txt	4,625	?	Text Document	2/16/2004 8:47 ...
cv024_6778.txt	5,302	?	Text Document	2/16/2004 8:47 ...
cv025_3108.txt	4,721	?	Text Document	2/16/2004 8:47 ...
cv026_29325.txt	3,056	?	Text Document	2/16/2004 8:47 ...

Hình 2.1: Đánh giá tích cực



cv000_29416.txt	4,043	?	Text Document	2/16/2004 10:4...
cv001_19502.txt	1,370	?	Text Document	2/16/2004 10:4...
cv002_17424.txt	2,848	?	Text Document	2/16/2004 10:4...
cv003_12683.txt	2,929	?	Text Document	2/16/2004 10:4...
cv004_12641.txt	4,418	?	Text Document	2/16/2004 10:4...
cv005_29357.txt	3,911	?	Text Document	2/16/2004 10:4...
cv006_17022.txt	3,365	?	Text Document	2/16/2004 10:4...
cv007_4992.txt	3,554	?	Text Document	2/16/2004 10:4...
cv008_29326.txt	4,545	?	Text Document	2/16/2004 10:4...
cv009_29417.txt	4,553	?	Text Document	2/16/2004 10:4...
cv010_29063.txt	4,448	?	Text Document	2/16/2004 10:4...
cv011_13044.txt	3,087	?	Text Document	2/16/2004 10:4...
cv012_29411.txt	2,826	?	Text Document	2/16/2004 10:4...
cv013_10494.txt	5,590	?	Text Document	2/16/2004 10:4...
cv014_15600.txt	3,088	?	Text Document	2/16/2004 10:4...
cv015_29356.txt	3,972	?	Text Document	2/16/2004 10:4...
cv016_4348.txt	3,664	?	Text Document	2/16/2004 10:4...
cv017_23487.txt	4,094	?	Text Document	2/16/2004 10:4...
cv018_21672.txt	2,487	?	Text Document	2/16/2004 10:4...
cv019_16117.txt	4,092	?	Text Document	2/16/2004 10:4...
cv020_9234.txt	4,078	?	Text Document	2/16/2004 10:4...
cv021_17313.txt	3,324	?	Text Document	2/16/2004 10:4...
cv022_14227.txt	3,695	?	Text Document	2/16/2004 10:4...
cv023_13847.txt	6,186	?	Text Document	2/16/2004 10:4...
cv024_7033.txt	4,402	?	Text Document	2/16/2004 10:4...
cv025_29825.txt	3,424	?	Text Document	2/16/2004 10:4...
cv026_29229.txt	3,062	?	Text Document	2/16/2004 10:4...

Hình 2.2: Đánh giá tiêu cực

plot : two teen couples go to a church party , drink and then drive .
they get into an accident .
one of the guys dies , but his girlfriend continues to see him in her life , and has nightmares .
what's the deal ?
watch the movie and " sorta " find out . . .
critique : a mind-fuck movie for the teen generation that touches on a very cool idea , but presents it in a very bac
which is what makes this review an even harder one to write , since i generally applaud films which attempt to bre
they seem to have taken this pretty neat concept , but executed it terribly .
so what are the problems with the movie ?
well , its main problem is that it's simply too jumbled .
it starts off " normal " but then downshifts into this " fantasy " world in which you , as an audience member , hav
there are dreams , there are characters coming back from the dead , there are others who look like the dead , the
now i personally don't mind trying to unravel a film every now and then , but when all it does is give me the same
it's obviously got this big secret to hide , but it seems to want to hide it completely until its final five minutes .
and do they make things entertaining , thrilling or even engaging , in the meantime ?
not really .
the sad part is that the arrow and i both dig on flicks like this , so we actually figured most of it out by the half-way
i guess the bottom line with movies like this is that you should always make sure that the audience is " into it " eve
i mean , showing melissa sagemiller running away from visions for about 20 minutes throughout the movie is just
okay , we get it . . . there
are people chasing her and we don't know who they are .
do we really need to see it over and over again ?
how about giving us different scenes offering further insight into all of the strangeness going down in the movie ?
apparently , the studio took this film away from its director and chopped it up themselves , and it shows .
there might've been a pretty decent teen mind-fuck movie in here somewhere , but i guess " the suits " decided th
the actors are pretty good for the most part , although wes bentley just seemed to be playing the exact same char
but my biggest kudos go out to sagemiller , who holds her own throughout the entire film , and actually has you f
overall , the film doesn't stick because it doesn't entertain , it's confusing , it rarely excites and it feels pretty redu

Hình 2.3: Đánh giá phim từ người xem

Chương 3

Tiền xử lý dữ liệu

3.1 Bộ dữ liệu đào tạo và kiểm tra

Ta đang phát triển một hệ thống có thể phân tích cảm xúc của một bài đánh giá phim là tích cực hay tiêu cực. Có nghĩa là sau khi mô hình được phát triển chúng ta cần đưa ra dự đoán trên các đánh giá mới. Điều này sẽ yêu cầu tất cả quá trình chuẩn bị dữ liệu được thực hiện trên những đánh giá mới đó giống như được thực hiện trên dữ liệu dùng để đào tạo cho mô hình.

Bằng cách tách bộ dữ liệu đào tạo và kiểm tra khi chuẩn bị dữ liệu ta có thể đảm bảo rằng ràng buộc này được tích hợp vào quá trình đánh giá của mô hình. Như vậy là bất kỳ kiến thức nào trong bộ kiểm tra có thể giúp chuẩn bị dữ liệu tốt hơn đều không có sẵn trong quá trình tiền xử lý dữ liệu và đào tạo mô hình.

Ta sẽ dùng 100 bài đánh giá tích cực và 100 bài đánh giá tiêu cực cuối cùng cho bộ kiểm tra và 1800 bài đánh giá còn lại dùng cho bộ đào tạo. Đây là mô hình sử dụng 90% dữ liệu đầu vào để đào tạo và 10% để kiểm tra.

3.2 Làm sạch các bài đánh giá

Vì nội dung bộ dũ liệu đã khá rõ ràng cho nên ta thực hiện các bước sau để làm sạch dữ liệu:

- Tách các mã thông báo bởi khoảng trắng.
- Loại bỏ các dấu câu ở các từ.
- Loại bỏ tất cả các từ không phải các từ không bao gồm các chữ cái từ bảng Alphabet.

- Loại bỏ các từ dừng.
- Loại bỏ tất cả các từ có độ dài <= 1 ký tự.

```
# turn a doc into clean tokens
def clean_doc(doc):
    # split into tokens by white space
    tokens = doc.split()
    # remove punctuation from each token
    table = str.maketrans('', '', string.punctuation)
    tokens = [w.translate(table) for w in tokens]
    # remove remaining tokens that are not alphabetic
    tokens = [word for word in tokens if word.isalpha()]
    # filter out stop words
    stop_words = set(stopwords.words('english'))
    tokens = [w for w in tokens if not w in stop_words]
    # filter out short tokens
    tokens = [word for word in tokens if len(word) > 1]
    return tokens
```

Sau khi làm sạch ta sẽ được một danh sách dài các từ vựng

```
['films', 'adapted', 'comic', 'books', 'plenty', 'success', 'whether', 'theyre',
, 'toward', 'kids', 'casper', 'arthouse', 'crowd', 'ghost', 'world', 'theres', '
rters', 'created', 'alan', 'moore', 'eddie', 'campbell', 'brought', 'medium', 'wh
hmen', 'say', 'moore', 'campbell', 'thoroughly', 'researched', 'subject', 'jack'
son', 'starting', 'look', 'little', 'odd', 'book', 'graphic', 'novel', 'pages',
otnotes', 'words', 'dont', 'dismiss', 'film', 'source', 'get', 'past', 'whole',
stumbling', 'block', 'hells', 'directors', 'albert', 'allen', 'hughes', 'getting
ludicrous', 'casting', 'carrot', 'top', 'well', 'anything', 'riddle', 'better',
```

3.3 Định nghĩa từ điển

Khi sử dụng mô hình Bag-of-Words thì việc định nghĩa một từ điển từ vựng là vô cùng quan trọng. Càng có nhiều từ vựng dự đoán thì mô hình sẽ càng chính xác hơn.

```
# load doc and add to vocab
def add_doc_to_vocab(filename, vocab):
    # load doc
    doc = load_doc(filename)
    # clean doc
    tokens = clean_doc(doc)
    # update counts
    vocab.update(tokens)

# load all docs in a directory
def process_docs(directory, vocab):
    # walk through all files in the folder
    for filename in listdir(directory):
        # skip any reviews in the test set
        if filename.startswith('cv9'):
            continue
        # create the full path of the file to open
        path = directory + '/' + filename
        # add doc to vocab
        add_doc_to_vocab(path, vocab)

# define vocab
vocab = Counter()
# add all docs to vocab
process_docs('txt_sentoken/pos', vocab)
process_docs('txt_sentoken/neg', vocab)
# print the size of the vocab
print(len(vocab))
# print the top words in the vocab
print(vocab.most_common(50))
```

Ta được một từ điển bao gồm 44276 từ và top 50 từ được sử dụng nhiều nhất trong các bài đánh giá.

```
44276
[('film', 7983), ('one', 4946), ('movie', 4826), ('like', 3201), ('even', 2262),
ms', 1873), ('would', 1844), ('much', 1824), ('also', 1757), ('characters', 1735)
'first', 1588), ('see', 1557), ('way', 1515), ('well', 1511), ('make', 1418), ('r
t', 1288), ('people', 1269), ('could', 1248), ('bad', 1248), ('scene', 1241), ('m
, 1140), ('scenes', 1135), ('man', 1131), ('many', 1130), ('doesnt', 1118), ('knc
14), ('another', 992), ('action', 985), ('love', 977), ('us', 967), ('go', 952),
'still', 936)]
```

Ta có thể lọc các từ có tần suất xuất hiện thấp ra khỏi từ điển vì khi trong các bài đánh giá chúng chỉ xuất hiện 1 tới 2 lần nên sẽ không có ảnh hưởng tới mô hình.

```
# keep tokens with a min occurrence
min_occurane = 2
tokens = [k for k,c in vocab.items() if c >= min_occurane]
print(len(tokens))
```

Số lượng từ vựng của từ điển sau khi lọc.

```
25767
```

Cuối cùng, ta sẽ lưu lại từ điển vào một file mới là "vocab.txt" để sau này có thể sử dụng để lọc các bài đánh giá phim trước khi mã hóa chúng lập mô hình.

```
# save list to file
def save_list(lines, filename):
    # convert lines to a single blob of text
    data = '\n'.join(lines)
    # open file
    file = open(filename, 'w')
    # write text
    file.write(data)
    # close file
    file.close()

# save tokens to a vocabulary file
save_list(tokens, 'vocab.txt')
```

Chương 4

Biểu diễn mô hình Bag-of-Words

Mô hình bag-of-words là một cách trích dẫn các đặc trưng từ văn bản để tập văn bản đầu vào có thể được sử dụng trong các thuật toán học máy.

Mỗi bài đánh giá được chuyển đổi thành một biểu diễn vector. Số lượng mục trong vector biểu diễn cho một tài liệu tương ứng với số lượng từ trong từ điển từ vựng. Các từ được tính điểm và điểm số được đặt ở vị trí tương ứng của biểu diễn.

4.1 Chuyển đổi bài đánh giá thành các dòng mã thông báo

Đầu tiên, ta cần một hàm để chuẩn bị tài liệu

```
32  # load doc, clean and return line of tokens
33  def doc_to_line(filename, vocab):
34      # load the doc
35      doc = load_doc(filename)
36      # clean doc
37      tokens = clean_doc(doc)
38      # filter by vocab
39      tokens = [w for w in tokens if w in vocab]
40      return ' '.join(tokens)
```

Tiếp theo, ta cần một hàm xử lý tất cả các bài đánh giá trong một thư mục để chuyển đổi tài liệu thành các dòng.

```
# load all docs in a directory
def process_docs(directory, vocab):
    lines = list()
    # walk through all files in the folder
    for filename in listdir(directory):
        # skip any reviews in the test set
        if filename.startswith('cv9'):
            continue
        # create the full path of the file to open
        path = directory + '/' + filename
        # load and clean the doc
        line = doc_to_line(path, vocab)
        # add to list
        lines.append(line)
    return lines
```

Cuối cùng, ta tải từ điển từ vựng lên và chạy qua các bài đánh giá.

```
# load the vocabulary
vocab_filename = 'vocab.txt'
vocab = load_doc(vocab_filename)
vocab = vocab.split()
vocab = set(vocab)
# load all training reviews
positive_lines = process_docs('txt_sentoken/pos', vocab)
negative_lines = process_docs('txt_sentoken/neg', vocab)
# summarize what we have
print(len(positive_lines), len(negative_lines))
```

4.2 Mã hóa bài đánh giá với biểu diễn mô hình bag-of-words

Ta sẽ sử dụng Keras API để chuyển đổi các bài đánh giá thành các vector tài liệu được mã hóa.

Đầu tiên, ta cần khởi tạo một Tokenizer, sau đó khớp với các tài liệu văn bản trong bộ dữ liệu huấn luyện.

```
# create the tokenizer
tokenizer = Tokenizer()
# fit the tokenizer on the documents
docs = negative_lines + positive_lines
tokenizer.fit_on_texts(docs)
```

Tiếp theo, ta mã hóa tài liệu theo tần suất xuất hiện.

```
# encode training data set
Xtrain = tokenizer.texts_to_matrix(docs, mode='freq')
print(Xtrain.shape)
```

Làm tương tự với bộ dữ liệu kiểm định ta được

```
(1800, 25768)
(200, 25768)
```

Kích thước của tập dữ liệu huấn luyện và kiểm định sau khi được mã hóa tương ứng là 1800 và 200, mỗi tài liệu này đều có cùng kích thước với từ điển từ vựng mã hóa (độ dài vector) là 25786.

Chương 5

Mô hình phân tích cảm xúc

5.1 Mô hình phân tích cảm xúc đầu tiên

Ta sẽ phát triển một mô hình mạng nơ-ron truyền thẳng đa lớp (Multilayer Perceptron) đơn giản để dự đoán cảm xúc của những bài đánh giá đã được mã hóa.

Mô hình sẽ có một lớp đầu vào bằng với số lượng từ trong từ điển.

```
n_words = Xtest.shape[1]
```

Lớp nhãn cho tất cả các dữ liệu đánh giá huấn luyện và kiểm định

```
ytrain = array([0 for _ in range(900)] + [1 for _ in range(900)])
ytest = array([0 for _ in range(100)] + [1 for _ in range(100)])
```

Ta sẽ sử dụng một lớp ẩn duy nhất với 50 nơ-ron và hàm kích hoạt tuyến tính được điều chỉnh. Đầu ra sẽ là một nơ-ron duy nhất có hàm kích hoạt sigmoid để dự đoán 0 cho các đánh giá tiêu cực và 1 cho các đánh giá tích cực.

```
# define network
model = Sequential()
model.add(Dense(50, input_shape=(n_words,), activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# compile network
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit network
model.fit(Xtrain, ytrain, epochs=50, verbose=2)
# evaluate
loss, acc = model.evaluate(Xtest, ytest, verbose=0)
print('Test Accuracy: %f' % (acc*100))
```

Đánh giá mô hình trên tập dữ liệu kiểm định, ta thấy mô hình hoạt động khá tốt, đạt độ chính xác trên 90%.

```
Epoch 45/50
57/57 - 1s - loss: 0.0166 - accuracy: 1.0000 - 1s/epoch - 26ms/step
Epoch 46/50
57/57 - 1s - loss: 0.0155 - accuracy: 1.0000 - 1s/epoch - 26ms/step
Epoch 47/50
57/57 - 2s - loss: 0.0147 - accuracy: 1.0000 - 2s/epoch - 27ms/step
Epoch 48/50
57/57 - 2s - loss: 0.0138 - accuracy: 1.0000 - 2s/epoch - 36ms/step
Epoch 49/50
57/57 - 2s - loss: 0.0130 - accuracy: 1.0000 - 2s/epoch - 31ms/step
Epoch 50/50
57/57 - 2s - loss: 0.0123 - accuracy: 1.0000 - 2s/epoch - 29ms/step
Test Accuracy: 91.000003
```

5.2 So sánh với các phương pháp chấm điểm từ vựng

Hàm *texts_to_matrix()* cho Tokenizer trong Keras API cung cấp cho ta 4 phương pháp để chấm điểmm từ:

- "binary" trường hợp các từ được đánh dấu là có mặt (1) hoặc vắng mặt (0).
- "count" trong đó số lần xuất hiện cho mỗi từ là số nguyên.
- "tfidf" mỗi từ được chấm điểm dựa trên tần suất xuất hiện của chúng, trong đó các từ phổ biến trên tất cả tài liệu sẽ bị trừ điểm.
- "freq" các từ được tính điểm dựa trên tần suất xuất hiện của chúng trong tài liệu.

Đầu tiền, ta cần tạo một hàm để mã hóa các tài liệu dựa trên mô hình và phương pháp tính điểm đã chọn.

```python
# prepare bag of words encoding of docs
def prepare_data(train_docs, test_docs, mode):
    # create the tokenizer
    tokenizer = Tokenizer()
    # fit the tokenizer on the documents
    tokenizer.fit_on_texts(train_docs)
    # encode training data set
    Xtrain = tokenizer.texts_to_matrix(train_docs, mode=mode)
    # encode training data set
    Xtest = tokenizer.texts_to_matrix(test_docs, mode=mode)
    return Xtrain, Xtest
```

Vì mạng nơ-ron có tính ngẫu nhiên, chúng có thể tạo ra các kết quả khác nhau với cùng một mô hình khớp với cùng một dữ liệu. Điều này có nghĩa là bất kỳ điểm số nào của mô hình đều là không đáng tin cậy cho nên ta cần ước tính điểm số của mô hình dựa trên mức trung bình của nhiều lần chạy.

```
# evaluate a neural network model
def evaluate_mode(Xtrain, ytrain, Xtest, ytest):
    scores = list()
    n_repeats = 30
    n_words = Xtest.shape[1]
    for i in range(n_repeats):
        # define network
        model = Sequential()
        model.add(Dense(50, input_shape=(n_words,), activation='relu'))
        model.add(Dense(1, activation='sigmoid'))
        # compile network
        model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
        # fit network
        model.fit(Xtrain, ytrain, epochs=50, verbose=2)
        # evaluate
        loss, acc = model.evaluate(Xtest, ytest, verbose=0)
        scores.append(acc)
        print('%d accuracy: %s' % ((i+1), acc))
    return scores
```

Đánh giá hiệu suất của 4 phương pháp chấm điểm từ khác nhau.

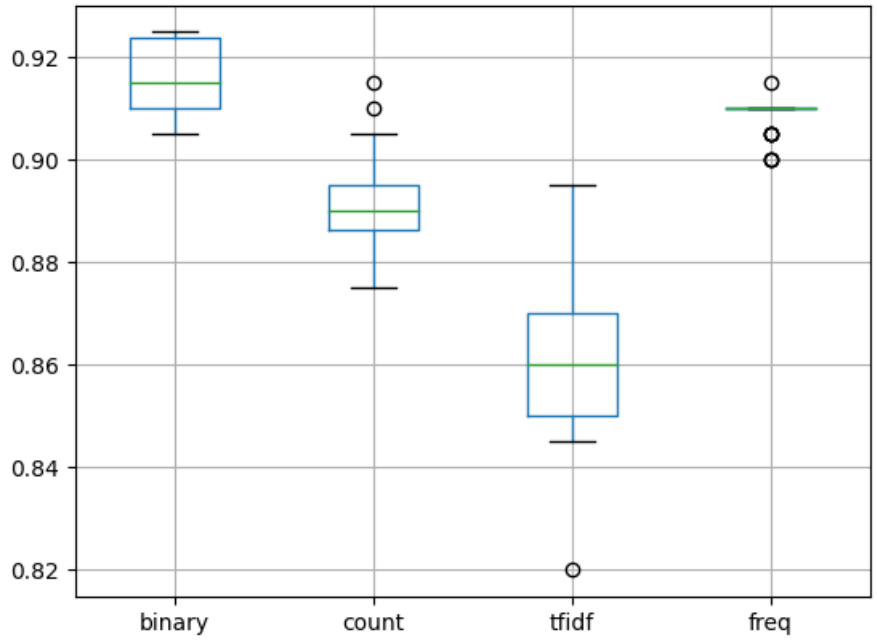
```
modes = ['binary', 'count', 'tfidf', 'freq']
results = DataFrame()
for mode in modes:
    # prepare data for mode
    Xtrain, Xtest = prepare_data(train_docs, test_docs, mode)
    # evaluate model on data for mode
    results[mode] = evaluate_mode(Xtrain, ytrain, Xtest, ytest)
# summarize results
print(results.describe())
# plot results
results.boxplot()
pyplot.show()
```

Khi kết thúc chương trình chạy, số liệu thống kê tóm tắt sự phân bố điểm cho từng phương pháp tính điểm từ trên 30 lần chạy cho mỗi phương pháp.

```
          binary      count      tfidf       freq
count  30.00000  30.000000  30.000000  30.000000
mean    0.91600   0.891833   0.861333   0.908500
std     0.00712   0.008855   0.015082   0.003511
min     0.90500   0.875000   0.820000   0.900000
25%     0.91000   0.886250   0.850000   0.910000
50%     0.91500   0.890000   0.860000   0.910000
75%     0.92375   0.895000   0.870000   0.910000
max     0.92500   0.915000   0.895000   0.915000
```

Ta có thể thấy rằng điểm trung bình của 2 phương pháp "freq" và "binary" trông có vẻ tốt hơn so với "count" và "tfidf".

Để tóm tăt phân phối độ chính xác cho mỗi phương pháp ta đưa ra biểu đồ hộp cho các kết quả trên.



Ta có thể thấy sự phân phối cho "freq" rất chặt chẽ. Ngoài ra, ta cũng thấy "binary" đạt được kết quả tốt nhất với độ chênh lệch không đáng kể và có thể nó là cách tiếp cận phù hợp với bộ dữ liệu này.

5.3 Đưa ra đánh giá cho những bài đánh giá mới

Cuối cùng, ta có thể sử dụng mô hình cuối để đưa ra dự đoán cho các bài đánh giá mới.

```
# classify a review as negative (0) or positive (1)
def predict_sentiment(review, vocab, tokenizer, model):
    # clean
    tokens = clean_doc(review)
    # filter by vocab
    tokens = [w for w in tokens if w in vocab]
    # convert to line
    line = ' '.join(tokens)
    # encode
    encoded = tokenizer.texts_to_matrix([line], mode='freq')
    # prediction
    yhat = model.predict(encoded, verbose=0)
    return round(yhat[0,0])
```

Dưới đây là một ví dụ với cả đánh giá tích cực và tiêu cực bằng cách sử dụng MLP đơn giản được phát triển ở trên với chế độ chấm điểm "freq".

```
# test positive text
text = 'Best movie ever!'
print(predict_sentiment(text, vocab, tokenizer, model))
# test negative text
text = 'This is a bad movie.'
print(predict_sentiment(text, vocab, tokenizer, model))
```

Sau khi chạy ví dụ ta sẽ phân loại được các đánh giá này.

```
1
0
```

Kết luận

Kết quả đạt được

Phát triển được một mô hình bag-of-words để dự đoán cảm xúc của một bài đánh giá phim là tích cực hay tiêu cực.

Kỹ năng đạt được

- Cách chuẩn bị dữ liệu văn bản để lập mô hình với vốn từ vựng hạn chế.
- Cách sử dụng mô hình bag-of-words để chuẩn bị dữ liệu huấn luyện và kiểm định.
- Cách phát triển một mô hình bag-of-words MLP (Multilayer Perceptron) và sử dụng nó để đưa ra dự đoán trên những bài đánh giá mới.

Tài liệu tham khảo

[1] <*https://machinelearningmastery.com/deep-learning-bag-of-words-model-sentiment-analysis/*>

[2] <*https://aws.amazon.com/vi/what-is/sentiment-analysis/*>