TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN

Web Mining

ĐỀ TÀI: SENTIMENT ANALYSIS: COMMENTS OR ARTICLES ABOUT PRODUCTS (VIETNAMESE)

SINH VIÊN THỰC HIỆN:

1. Nguyễn Tùng Cương 20150469

2. Vũ Công Luật 20142745

3. Lê Thị Dung 20140692

4. Nguyễn Văn Túc 20145070

GIẢNG VIÊN HƯỚNG DẪN: TS. Nguyễn Kiêm Hiếu

Hà Nội, tháng12 năm 2018

MỤC LỤC

Contents

Lời nói đầu

Ngày này dưới sự bùng nổ của Internet cùng với sự ra đời của các mạng xã hội, các trang thương mại điện tử có lượng người dùng khổng lồ. Giúp cho các công ty, tổ chức, cá nhân kinh doanh có thể đưa sản phẩm của mình tới gần hơn người tiêu dùng. Cùng với đó người tiêu dùng cũng có thể dễ dàng thể hiện quan điểm của mình về các sản phẩm, dịch vụ từ phía nhà cung cấp thông qua các trang mạng xã hội, các trang thương mại điện tử. Đây là nguồn thông tin vô cùng giá trị, nó có thể giúp cho các công ty, tổ chức, các nhà cung cấp dịch vụ có được những đánh giá khác quan về sản phẩm dịch vụ của mình để từ đó đưa ra những điều chỉnh phù hợp nhất với sản phẩm của mình nhằm cải thiện chất lượng sản phẩm, dịch vụ của mình.

Tuy nhiên lượng dữ liệu từ người dùng trên các trang mạng xã hội, các trang thương mại điện tử thường có số lượng rất lớn, việc sử dụng con người trức tiếp thu thập đánh giá thông tin là việc làm vô cùng tốn kém và tỏ ra không hiệu quả. Đã có rất nhiều hệ thống đánh giá quan điểm người dùng tự động đã ra đời, tuy nhiên các hệ thống này chỉ hỗ trợ đánh giá thông tin là tiếng anh. Điều đó thức đẩy nhóm xây dựng lên hệ thống đánh giá quan điểm người dùng bằng tiếng việt.

Với phạm vi bài tập lớn môn học, hạn chế về mặt dữ liệu, thời gian, công cụ hỗ trợ... Nhóm lựa chọn tiến hành triển khai hệ thống trên bộ dữ liệu SA VLSP 2016. Đây là bộ dữ liệu được hội đồng, ban tổ chức của hội nghị xây dựng nhằm mục đích để các doanh nghiệp, tổ chức, nhóm, cá nhân nghiên cứu sử dụng để phát triển bài toán phân tích quan điểm. Kết quả của các nhóm thực hiện được công bố
tại website của VLSP.

I, Mô tả bài toán:

1, Mục đích của bài toán:

- Mục đích của bài toán là để phân loại nhận xét, đánh giá của người dùng thu thập trên internet thành 2 nhóm: Positive (Tích cực), Negative (Tiêu cực). Đầu vào và đầu ra được mô tả như sau:

  (+) Input: Đoạn text đánh giá của người dùng về sản phẩm, dịch vụ.

  (+) Output: Quan điểm của người dùng về sản phẩm, dịch vụ đó.

2, Phương pháp giải quyết bài toán:

- Một hướng tiếp cận cho bài toán này là sử dụng các mô hình kiến trúc Deep learning. Khi đó, những vấn đề đặt ra cho bài toán lại có những sự thay đổi:

  (+) Word embedding

  (+) Các mạng học sâu, trích chọn đặc trưng tự động

  (+) Mạng neural cho phân loại

3, Kịch bản hệ thống:

- Kịch bản hoạt động của hệ thống được thực hiện như sau:

  (+) Người dùng nhập một đánh giá dạng text

  (+) Hệ thống tiến hành tiền xử lý dữ liệu

  (+) biểu diễn đoạn text về Word

  embedding

  (+) Tải lại model đã thu được ở bước huấn luyện hệ

  thống

  (+) Hệ thống tiến hành gán nhãn cho câu đánh giá.

4, Triển khai:

- Môi trường: Ubuntu 16.04, CPU core i3

- Ngôn ngữ: Python 2.7

- Thư viện: sklearn, numpy, pandas, tensorflow, gensim, pyvi

- Công cụ hỗ trợ: Git

II, Tiền xử lý dữ liệu:

1, Đặc điểm của dữ liệu:

- Bộ dữ liệu thô được lấy từ hội nghị: VLSP 2016 về Sentiment Analysis.
- Đây là một bộ dữ liệu được VLSP lấy từ: Tinhte.vn, Vnexpress.net và Facebook, bao gồm 5100 bình luận của người dùng được chia đều cho 3 nhãn: Tích cực, tiêu cực và trung tính.
- Qua phân tích thống kê, chúng em rút ra được đặc điểm của bộ dữ liệu này là:

  (+) Bộ dữ liệu nhỏ, chỉ phù hợp cho nghiên cứu.

  (+) Nhiều câu đánh giá có chứa đường dẫn tới trang web khác, đây có thể là một comment spam cần bị loại bỏ nhưng cũng có thể là sự so sánh hơn kém của người dùng. Những đánh giá này thường là tiêu cực.

  (+) Các đánh giá có độ dài ngắn khác nhau, tập trung nhiều từ 30-100 từ, nhiều đánh giá chỉ có 1 từ nhưng cũng có những đánh giá hơn 2400 từ.

  (+) Dữ liệu chứa nhiều thông số kỹ thuật.

  (+) Người dùng đề cập nhiều tới giá tiền trong các bình luận của mình.

  (+) Những đánh giá trung tính thường chứa nhiều câu nói cả về tích cực và tiêu cực, và có độ dài lớn hơn 2 nhãn còn lại.

  (+) Nhiều từ viết tắt, sai ngữ pháp, không theo chuẩn ngữ pháp.

  (+) Nhiều đánh giá là tiếng Việt không dấu.

2, Xử lý dữ liệu thô:

- Dựa vào các đặc điểm phân tích dữ liệu được nêu ở phía trên, ta đưa ra mô hình chuẩn cho việc tiền xử lý dữ liệu thô, bao gồm các công đoạn sau:

  (+) Thay thế các url trong dữ liệu bởi nhãn *link_spam*: Sử dụng biểu thức chính quy.

  (+) Chuyển hết dữ liệu thành chữ thường.

  (+) Thay tất cả các biểu thức giá tiền bằng nhãn *money*: Sử dụng biểu thức chính quy. Các đơn vị tiền tệ được xử lý bao gồm k, đ, ngàn, nghìn, usd, tr, củ, triệu, yên và $ ở trước và sau giá tiền.

  (+) Loại bỏ các dấu câu và các ký tự đặc biệt, bao gồm: '!"#$%&\'()*+, ./:;<=>?@[\\]^_`{|}~'

(+) Xóa bỏ các thông số kỹ thuật: Sử dụng biểu thức chính quy. Các thông số ở đây được hiểu như là các từ được cấu tạo bởi chữ và số kết hợp. Vd: J2prime, 30px, 50Gb,…

(+) Xử lý các trường hợp láy âm tiết: Sử dụng biểu thức chính quy. Ví dụ: Ngoooon, Niceeee, Điiiiii,...

(+) Thay thế các từ viết tắt bởi từ gốc của nó: Sử dụng biểu thức chính quy. Ví dụ: k, ko --> không, sr --> xin lỗi,…

(+) Loại bỏ số và các từ chỉ có 1 ký tự.

(+) Tách từ: Sử dụng bộ tách từ có sẵn Pyvi của thầy Trần Việt Trung. Ví dụ: trường đại học bách khoa hà nội. trường đại_học bách_khoa hà_nội .

- Sau khi thực hiện tuần tự và đầy đủ theo quy trình trên, ta thu được bộ dữ liệu sạch cho pha tiếp theo của mô hình. Chia dữ liệu theo tỉ lệ 80:20 để có được dữ liệu train và test hệ thống. Tỷ lệ này đều cho mỗi nhãn.

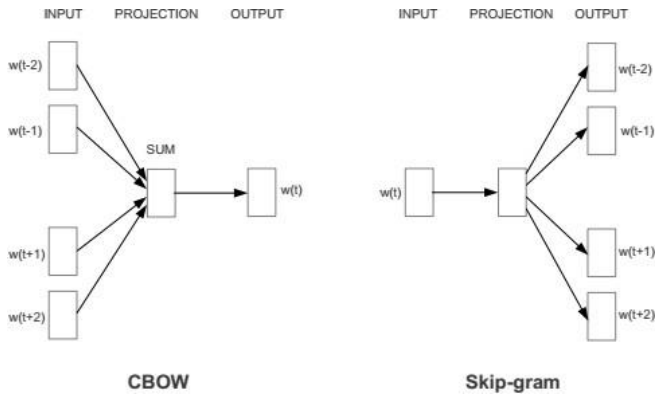
III, Deep Learning

1, Word embedding: Word2vec

- Word embedding là một nhóm các mô hình được xây dựng nhằm mục đích ánh xạ một từ sang một vector, thông qua đó, có thể thấy được ngữ nghĩa tiềm ẩn và mỗi quan hệ ngữ nghĩa giữa các từ trong không gian từ vựng.
- Word2vec được tạo ra bởi một nhóm các nhà nghiên cứu dẫn đầu bởi Tomas Mikolov tại Google với mục đích tái tạo lại ngữ cảnh của ngôn ngữ thông qua không gian vector.
- Word2vec là một mạng neural 2 lớp với duy nhất một tầng ẩn, lấy đầu vào là một corpus lớn và sinh ra không gian vector có số chiều khoảng vài trăm, với mỗi từ duy nhất trong corpus được gắn với một vector tương ứng trong không gian. Các word vectors được xác định trong không gian vector sao cho những từ có chung ngữ cảnh trong corpus được đặt gần nhau trong không gian.
- Có hai cách xây dựng word2vec:

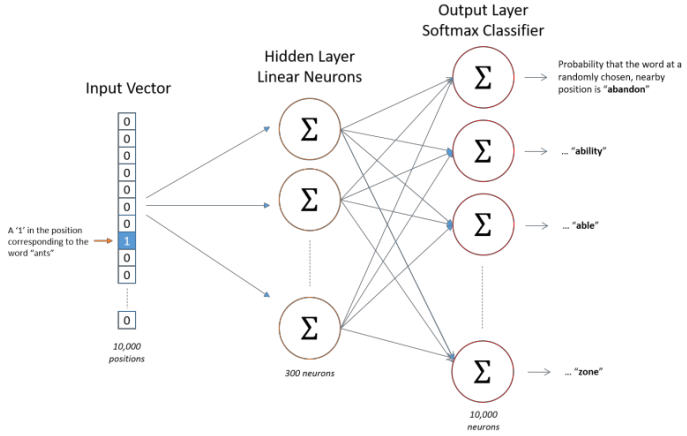
  (+) Sử dụng ngữ cảnh để dự đoán mục tiêu (CBOW)

  (+) Sử dụng một từ để dự đoán ngữ cảnh mục tiêu (Skip-gram).



Hình 1: Mô hình CBOW và Skip-gram (Nguồn: skip-gram model)

- Trong báo cáo này sẽ chỉ đề cập tới Skip-gram do khả năng của nó làm việc với bộ dữ liệu lớn tốt hơn người anh em của nó(CBOW) và Skip-gram được sử dụng để triển khai các đề xuất trong báo cáo này.



Hình 2: Mô hình Skip-gram chi tiết (Nguồn: Skip-gram)

- Khi cho một từ cụ thể ở giữa câu(input word), nhìn vào những từ ở gần và chọn ngẫu nhiên. Mạng neural sẽ cho chúng ta biết xác suất của mỗi từ trong từ vựng về việc trở thành từ gần đó mà ta vừa chọn.
- Mục tiêu của Skip-gram model là học ra trọng số các lớp ẩn, trọng số này chính là các word vectors.

2, Convolutional neural network – CNN:

(Nội dung được tham khảo và dịch ngữ theo blog Wildml, các hình ảnh không chú thích nguồn trong phần này đều thuộc sở hữu hoặc được trích dẫn nguồn thông qua Wildml).

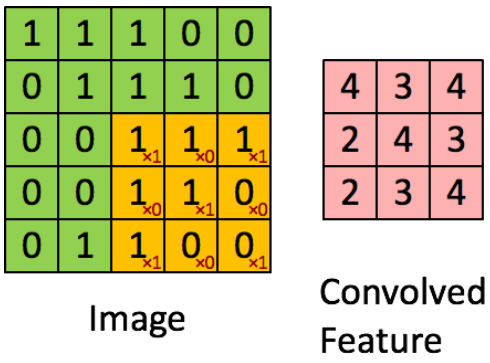
- Khi chúng ta nghe tới Convolutional neural network, chúng ta thường nghĩ ngay tới Computer Vision. Tuy nhiên, ở một khía cạnh khác, khi sử dụng cho các bài toán NLP, CNN lại cho ra những kết quả hết sức thú vị.
- Vậy, dữ liệu text và dữ liệu ảnh có điểm gì tương đồng mà ta có thể sử dụng

một mô hình xử lý ảnh cho dữ liệu text? Có cách nào để chúng ta nhìn nhận một đoạn text như một bức ảnh không?

- Như một cách để giúp chúng ta dễ tưởng tượng hơn, giả sử ta đã sử dụng word embedding như đã trình bày ở trên để biến mỗi từ trong một đoạn văn bản thành các vector có số chiều là **n**. Từ đó ta có thể coi môt câu văn(đoạn văn) như một ma trận **mxn**, trong đó m là kích thước hay số từ có trong văn bản đó. Ma trận này về mặt biểu diễn trông cũng có vẻ tương đồng với ma trận biểu diễn cho một bức ảnh đa mức xám(1 channel) với kích thước **mxn**.
- Vậy, về mặt hình thức, ta có thể dễ dàng thấy được việc sử dụng CNN cho bài toán NLP là có thể. Giờ hãy cùng đi sâu vào chi tiết cho Convolutional neural network để xem nó là gì và nó áp dụng cho bài toán của chúng ta như thế nào.

2.1, Convolutional là gì?

- Để đơn giản, chúng ta có thể hiểu nó như một cửa sổ trượt áp dụng cho ma trận được sử dụng như một bộ lọc tính năng.



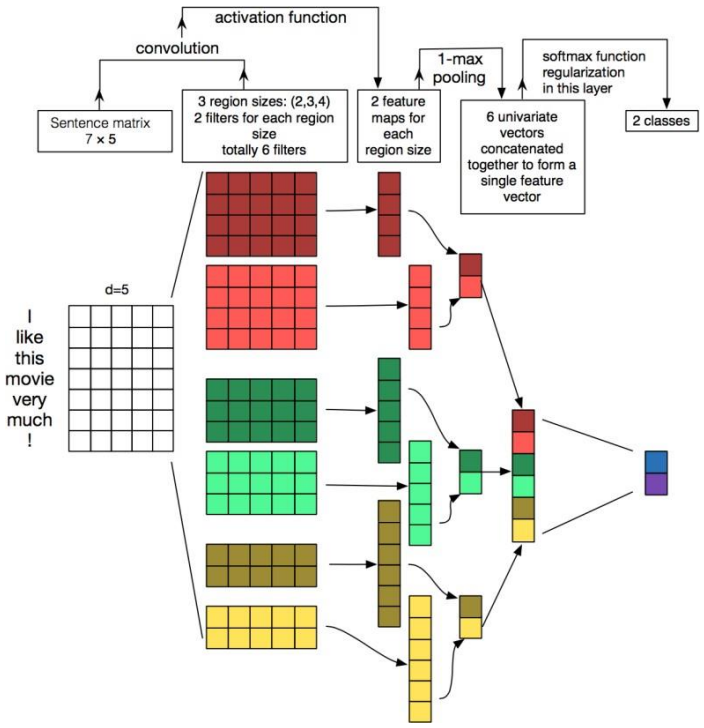
Hình 3: Ở đây, chúng ta có một bộ lọc với kích thước 3x3. Một cách khôn ngoan, ta trượt cửa sổ này trên hết ma trận gốc (5x5), nhân nó với ma trận gốc sau đó tổng hợp lại ta được một ma trận mới 3x3. Có thể hiểu đây là 1 cách lọc ra các đặc trưng của ma trận gốc thông qua các lân cận 3x3.

2.2, Convolutional Neural Network:

- Convolutional neural network cơ bản là một mạng gồm nhiều lớp convolution chồng lên nhau, sử dụng các hàm kích hoạt phi tuyến như tanh hoặc relu. Khác với mạng neural truyền thống là chúng ta kết nối mỗi nút từ tầng ẩn này tới mỗi nút ở tầng ẩn kế tiếp, đầy đủ từ tầng đầu vào đến lớp đầu ra, CNN sử dụng convolution cho mỗi tầng để có được đầu ra để cho vào tầng tiếp theo. Mỗi layer sẽ áp dụng các bộ lọc khác nhau. Sau đó chúng ta sẽ tổng hợp chúng lại để được một biểu diễn đặc trưng cho đầu vào của mạng.
- Ngoài ra, còn có 1 lớp đặc biệt gọi là pooling, được thiết kế để giảm số chiều của mỗi ouput của lớp convolution để làm input cho lớp sau.
- Trong giai đoạn học từ dữ liệu, CNN tự động học được giá trị của các bộ lọc, dựa trên nhiệm vụ mà chúng ta đưa ra cho nó.

2.3, Ứng dụng CNN vào bài toán NLP:

- Như đã đề cập ở trên, thay vì đầu vào là pixels của hình ảnh, ta có một ma trận **mxn** là biểu diễn cho một câu hay một đoạn văn bản. Mỗi hàng của ma trận là vector đại diện cho một từ. Các vector này là các vector được lấy ra từ word embedding mà thường là GloVe, Word2vec(CBOW, Skip-gram),.. hoặc cũng có thể chỉ đơn giản là một one-hot vector.
- Chúng ta cũng vừa đề cập tới khái niệm convolutional trong trường hợp tổng quát là một bộ lọc trượt đầy đủ trên ma trận. Đối với bài toán ứng dụng CNN cho lĩnh vực NLP, bộ lọc này có một điều đặc biệt. Đó là kích thước theo chiều ngang của bộ lọc thường chính bằng số cột của ma trận biểu diễn cho văn bản. Để khi trượt, ta có thể trượt được trên toàn bộ từ, hay nói cách khác là ta có thể nắm bắt được ngữ nghĩa của toàn bộ từ. Khái niệm "trượt" ở đây sẽ được hiểu đơn giản chỉ là trượt từ trên xuống dưới, cho tới khi hết văn bản. Kết hợp tất cả những gì đã đề cập ở trên, sắp xếp lại ta được một biểu diễn mạng neural như sau:



Hình 4: Kiến trúc mạng neural (CNN) để phân loại câu. Ở đây ta mô tả có 3 kích thước của bộ lọc tương ứng là 2, 3, 4. Mỗi loại bộ lọc có 2 bộ lọc thành phần. Mỗi bộ lọc thực hiện việc

trích rút tính năng từ ma trận gốc thành một bản đồ tính năng. Sau đó tổng hợp 1-max được thực hiện trên mỗi bản đồ này, nghĩa là số lớn nhất từ mỗi bản đồ được ghi lại. Đây chính là lớp pooling như đã đề cập. Vì vậy, một vector đặc trưng đơn biến được tạo ra từ 6 bộ lọc này, chúng được nối với nhau và tạo nên vector đặc trưng cho văn bản đầu vào. Lớp softmax cuối cùng nhận vector đặc trưng này làm đầu vào để thực hiện phân loại câu. Ở đây giả định là phân loại nhị phân.

2.4, Các tham số cấu hình cho ứng dụng mạng CNN:

- Narrow vs. Wide convolution:

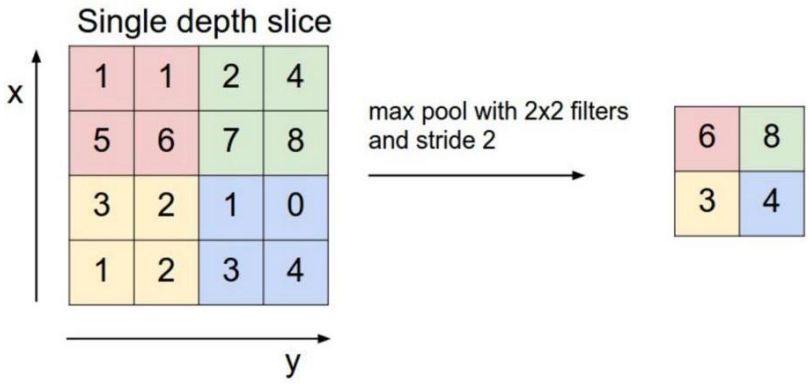
(+) Khi đề cập tới convolutional ở trên, chúng ta đã bỏ qua một chút về chi tiết cách mà các bộ lọc hoạt động. Khi thực hiện trượt bộ lọc trên một ma trận, các lân cận trung tâm của ma trận thì được thực hiện khá tốt, nhưng điều đó lại không xảy ra khi chúng ta trượt qua các cạnh. Hiểu đơn giản là khi chúng ta trượt bộ lọc trên một ma trận, các phần từ sẽ được trượt qua nhiều lần trong nhiều lần lọc, ngoại trừ các điểm ở cạnh,ví dụ như các lân cận ở trên cùng, bên trái. Để giải quyết vấn đề này, ta có thể sử dụng một khái niệm được gọi là zero-padding. Điều đó đơn giản chỉ là ta cho tất cả các phần tử bên ngoài ma trận bằng 0. Việc thêm zero-padding cũng được gọi là Wide convolution vì nó làm đầu ra của convolution rộng hơn hoặc bằng kích thước của ma trận đầu vào. Ngược lại, việc không sử dụng zero-padding được gọi là narrow convolution.

- Stride Size:

(+) Một tham số khác cho convolution là Stride size, hiểu một cách đơn là là độ rộng của một bước nhảy. Bước nhảy ở đây được hiểu như cách mà bộ lọc dịch chuyển trên ma trận. Thông thường người ta lấy giá trị của tham số này bằng 1, và bộ lọc lọc qua các phần của ma trận được chồng lấn lên nhau.

- Pooling layer:

(+) Một khía cạnh quan trọng của Convolutional neural network chính là các pooling layer, thường được sử dụng sau lớp convolution. Cách phổ biến nhất để pooling là áp dụng các phép xử lý tính **max** trên kết quả của convolution. Trong trường hợp tổng quát, chúng ta không nhất thiết phải đi qua phải pooling trên toàn bộ ma trận kết quả của lớp convolution trước đó mà có thể sử dụng khái niệm cửa sổ tương tự như các filter trong convolution.



(+) Tuy nhiên, khi áp dụng vào bài toán NLP, chúng ta lại thường bỏ qua khái niệm cửa sổ và thực hiện pooling trên toàn bộ ma trận để lấy được một đặc trưng nổi trội duy nhất..

(+) Việc pooling cung cấp cho chúng ta một đầu ra có kích thước cố định, rất tốt để giải quyết các vấn đề phân loại. Tức là việc bạn áp dụng bất kể kích thước bộ lọc là bao nhiêu thì bạn vẫn được cung cấp một đầu ra với định dạng duy nhất để dễ dàng đưa sang một trình phân loại khác.

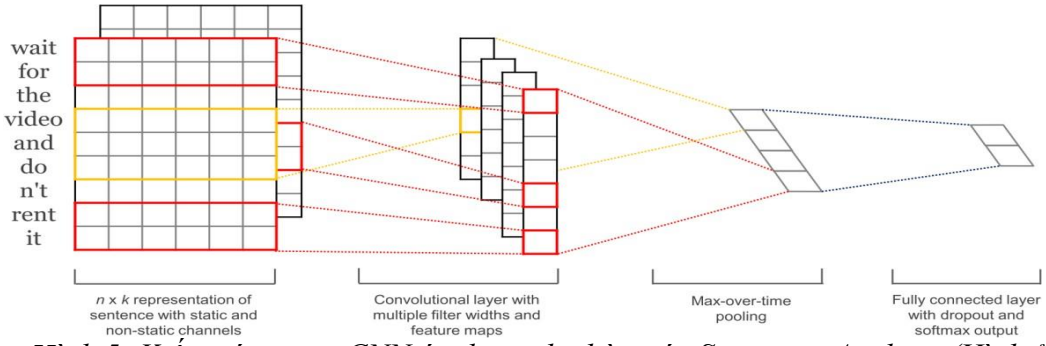
(+) Một ý nghĩa khác của pooling mà chúng ta đã đề cập ở ngay phần giới thiệu về CNN đó là khả năng giảm kích thước cho đầu ra bằng cách chỉ giữ lại những thông tin nổi trội nhất. Như tên gọi của nó, chúng ta có thể hiểu mỗi bộ lọc có khả năng phát hiện ra một tính năng, khía cạnh cụ thể nào đó trong câu, ví dụ như sự phủ định, sự cảm thán,... Việc pooling sẽ cho ta giữ lại những đặc điểm nổi trội của tính năng đó mà bỏ qua những khía cạnh xung quanh.

- Channel:

(+) Khi nói đến channel, ta thấy khái niệm này với dữ liệu ảnh phù hợp hơn (kênh màu). Tuy nhiên, một cách tổng quát, channel được hiểu như là các khía cạnh, góc nhìn của ta với dữ liệu. Ví dụ trong công việc xử lý ảnh, ta thường hay sử dụng 3 kênh màu RGB. Trong NLP, chúng ta có thể hiểu mỗi channel (kênh) ở đây được hiểu như là các cách biểu diễn dữ liệu khác nhau cho cùng một câu (GloVe, Word2vec,...) hay như là cùng một câu nhưng được biểu diễn qua các ngôn ngữ khác nhau (Tiếng Việt, Tiếng Anh,...).

2.5, Cấu hình mạng CNN cho bài toán Sentiment Analysis:

- Lớp đầu vào là một câu bao gồm các từ được vector hóa sử dụng mô hình word2vec Skip-gram. Các vector biểu diễn cho mỗi từ được ghép nối để trở thành ma trận biểu diễn cho câu. Tiếp theo là một convolution layer với nhiều 450 bộ lọc chia làm 3 loại [3, 5, 7], sau đó là một lớp max pool, và cuối cùng mà một mang neural liên kết đầy đủ được xây dựng nhằm mục đích phân loại. Mạng này chứa 1 tầng ẩn chứa 100 nút, sử dụng hàm kích hoạt ReLu. Đầu ra là một phân loại softmax.



Hình 5: Kiến trúc mạng CNN áp dụng cho bài toán Sentiment Analysis (Hình ảnh minh họa)

IV, Đánh giá mô hình

- Bảng accurancy với các epoch khác nhau trên kiến trúc:

Epoch	CNN
4	71.18
7	80.00
8	75.88
9	82.94
10	**84.12**
11	79.41

- Kết quả chi tiết về Precision,Recall,F1 score sau khi train với 10 Epoch:

	Positive	Negative
Precision	80.85	88.16
Recall	89.41	78.82
F1	84.92	83.23

V, Khó khăn gặp phải và cách giải quyết:

- Khó khăn lớn nhất của nhóm trong quá trình thực hiện đồ án là vấn đề thu thập dữ liệu. Như đã trình bày phía trên, trong bài toán này, nhóm chúng em đã tiến hành sử dụng bộ dữ liệu VLSP 2016. Đây là bộ dữ liệu nhỏ, lại chứa rất nhiều "nhiễu" như : các ký tự đặc biệt, các đường dẫn url, các thông số kỹ thuật, các từ viết tắt, sai chính tả, không dấu, không đúng ngữ pháp,… Chính điều này đã khiến cho công đoạn tiền xử lý và làm sạch dữ liệu của nhóm gặp rất nhiều khó khăn vì đa phần các thành viên trong nhóm chưa có nhiều kinh nghiệm xử lý dữ liệu, đặc biệt là những dữ liệu về ngôn ngữ tiếng việt.

VIII, Đề xuất, cải tiến, phát triển trong tương lai:

- Như đã trình bày phía trên, nhóm gặp hạn chế về tập dữ liệu. Hy vọng trong tương lai nhóm có thể tìm kiếm được những bộ dữ liệu lớn hơn để có thể cải thiện hệ thống cả về chất lượng lẫn tốc độ thực thi.

TÀI LIỆU THAM KHẢO

[1] Slide bài giảng môn học Khai phá web (TS. Nguyễn Kiêm Hiếu)
[2] Slide bài giảng môn Học máy (TS. Thân Quang Khoát)

[3] Blog http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/

[4] Yoon Kim. Convolutional neural networks for sentence classification

[5] Blog http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp

[6] Thư viện học máy scikit-learn: http://scikit-learn.org/stable

[7] Thư viện deep learning tensorflow: https://www.tensorflow.org