BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

NIÊN LUẬN
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

Text Classification – CNN (sử dụng cho Tiếng Việt)

Sinh viên: Nguyễn Phước Thành
Mã số: B1610669
Khóa: 42

Cần Thơ, 05/2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN

NIÊN LUẬN
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

Text Classification – CNN (sử dụng cho Tiếng Việt)

Người hướng dẫn
TS Lâm Nhựt Khang

Sinh viên thực hiện
Nguyễn Phước Thành
Mã số: B1610669
Khóa: 42

Cần Thơ, 05/2020

Lời cảm ơn

Cảm ơn TS. Lâm Nhựt Khang, Bộ môn Công nghệ Thông tin, Khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ đã tích cực hướng dẫn, giúp đỡ nghiên cứu đề tài này.

Mục lục

Danh mục đồ thị biểu bảng và hình ảnh

Tóm lượt

Hiện nay, các tiến bộ khoa học kỹ thuật dần thay thế các hoạt động thủ công truyền thống của con người, cùng với đó là sự tiến bộ không ngừng của công nghệ thông tin trong thời đại ngày nay đòi hỏi chiếc máy tính của chúng ta phải làm được nhiều hơn nữa, cuộc cách mạng công nghệ 4.0 đã thúc đẩy các nhà khoa học trong việc biến ước mơ của con người về trí tuệ nhân tạo thành hiện thực, những ứng dụng của trí tuệ nhân tạo ngày càng đa dạng, những thành tựu mà trí tuệ nhân tạo đem lại khiến chúng một phần không thể thiếu trong một xã hội phát triển nhanh chóng như ngày hôm nay. Hoạt động của trí tuệ nhân tạo được dựa trên tri thức và hành vi của con người, bên cạnh đó là bề dày về số lượng thuật toán cũng như giải thuật được thiết kế một cách tỉ mỉ. Trí tuệ nhân tạo bao gồm nhiều thành phần: xử lý ảnh, xử lý ngôn ngữ tự nhiên, học máy, học sâu, ...

Phần giới thiệu

Hiện nay phần lớn các doanh nghiệp đang đối mặt với "cơn lũ" dữ liệu về mọi mặt: feedback của khách hàng, thông tin đối thủ cạnh tranh, emails của khách hàng, các văn bản về sản phẩm và kỹ thuật. Việc khai thác các dữ liệu này là điểm mấu chốt để doanh nghiệp có thể nhanh chóng đưa ra các kế hoạch hoặc hành động kịp thời so với các đối thủ cạnh tranh.

Vấn đề ở đây là có quá nhiều thông tin cần xử lý cùng lúc, và kích thước dữ liệu ngày càng tăng. Điều này sẽ là bất khả thi nếu chỉ dựa vào sức người trong một giới hạn về số lượng và thời gian nhất định.

Tiếng nói và chữ viết là hai phạm trù cơ bản của ngôn ngữ. Việc nhận biết cũng như phân biệt ý nghĩa của một câu nói xem ra khá đơn giản đối với con người chúng ta, nhưng làm sao để máy tính có thể phân tích và đưa ra những nhận định gần giống con người đối với dữ liệu đầu vào là văn bản khi máy tính chỉ có thể hiểu ở những con số 0, 1 liên tiếp nhau.

Mục tiêu của đề tài này là tìm hiểu về việc phân loại văn bản (Text Classification) bằng việc sử dụng mô hình CNN (Convolutional Neural Network) để áp dụng cho văn bản Tiếng Việt.

Bố cục của bản báo cáo gồm 3 phần: phần giới thiệu, phần nội dung và phần kết luận. Trong đó phần nội dung gồm có 4 chương:
Chương 1 - Đặc tả yêu cầu: Mô tả phương pháp phân loại văn bản.
Chương 2 - Thiết kế giải pháp: Trình bày các lý thuyết có liên quan được sử dụng trong đề tài.
Chương 3 - Cài đặt giải pháp: Mô tả hoạt động của phương pháp phân loại văn bản sử dụng CNN (Convolutional Neural Network) thông qua ví dụ minh họa
Chương 4 - Đánh giá kiểm thử: Đánh giá và hướng phát triển

Phần nội dung

Chương 1 – Đặc tả yêu cầu

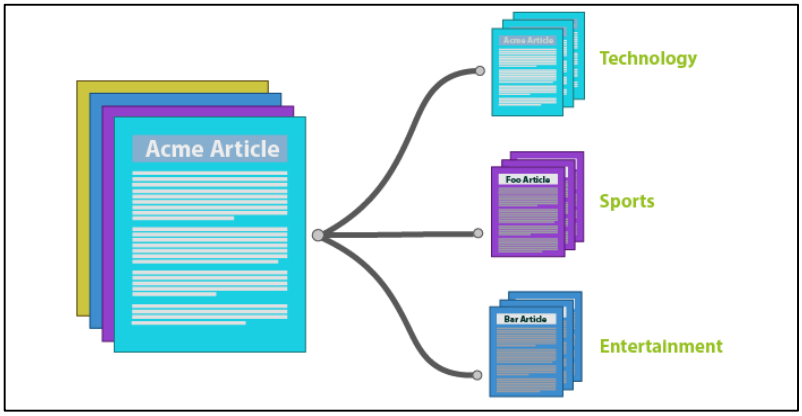
Xử lý ngôn ngữ là một phạm trù trong xử lý thông tin với dữ liệu đầu vào là ngôn ngữ, có thể được xem như văn bản hay tiếng nói. Đây là các dạng văn bản chính được lưu trữ dưới dạng tài liệu, đặc điểm chung của chúng là không có cấu trúc (non-structured) hoặc nửa cấu trúc (semi-structured) và không thể lưu lại dưới dạng bảng biểu. Vì vậy chúng ta cần phải xử lý chúng để máy tính có thể hiểu được.
Một trong những ứng dụng rộng rãi của xử lý ngôn ngữ tự nhiên (NLP) và máy học có giám sát đó là "phân loại văn bản". Mục tiêu là nhằm phân loại một cách tự động một văn bản (câu / từ) thuộc vào một hoặc nhiều nhóm đã được định nghĩa từ trước.

Phương pháp hiện tại là sử dụng các kỹ thuật có sẵn cũng như các thuật toán điển hình để xử lý theo từng giai đoạn:
Giai đoạn 1: Tiếp cận và tiền xử lý dữ liệu trước khi thực hiện chuyển các văn bản thành các vector đưa vào ma trận.
Giai đoạn 2: Sử dụng mô hình CNN cho việc huấn luyện tập dữ liệu và đưa ra mô hình phân loại văn bản.

Chương 2 – Thiết kế giải pháp

Việc tiếp cận một bài toán liên quan đến xử lý ngôn ngữ tự nhiên, thông thường sẽ trải qua các mức phân tích:

- Phân tích hình thái: cách từ được xây dựng, các tiền tố và hậu tố của từ.
- Phân tích cú pháp: mối liên hệ về cấu trúc và ngữ pháp giữa các từ và ngữ.
- Phân tích ngữ nghĩa: nghĩa của từ, cụm từ và cách diễn đạt.
- Tích hợp văn bản: quan hệ giữa các ý hoặc các câu.
- Phân tích thực nghĩa: mục đích phát ngôn, cách sử dụng ngôn ngữ trong giao tiếp.



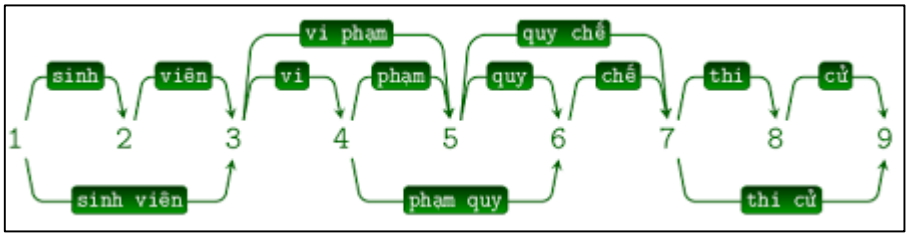
Hình 1: Phân loại bài viết

Mô hình được mong đợi sẽ có khả năng phân loại được bài viết dựa theo nội dung được dự đoán bởi mô hình phân loại.

2.1. Phương pháp tách từ Tiếng Việt - Tokenization and Word Segmentation

Việc tách từ là một trong cách phương pháp đầu tiên trong việc xử lý dữ liệu đối với các bài toán xử lý ngôn ngữ tự nhiên sau khi dữ liệu đã được làm sạch, mục đích của phương pháp này nhằm:

- Các khoảng trắng giữa các từ. (Điều này chỉ được phép đối với các ngôn ngữ như tiếng Việt, trong đó khoảng trắng được sử dụng để đánh dấu các ranh giới âm tiết thay vì ranh giới từ.)
- Loại bỏ các từ viết tắt, gây thừa thãi trong câu.
- Lọc các từ khóa chính làm nổi bật nội dung của cả câu.



Hình 2: Các đặc trưng trong câu

2.2. Các phương pháp biểu diễn ngôn ngữ

Định nghĩa ***word embeddings*** (tập tự nhúng):

"Tập nhúng từ *là tên chung cho một tập hợp các mô hình ngôn ngữ và các phương pháp học đặc trưng trong xử lý ngôn ngữ tự nhiên (NLP), nơi các từ hoặc cụm từ từ vựng được ánh xạ tới vectơ số thực. Về mặt khái niệm, nó liên quan đến việc nhúng toán học từ một không gian với một chiều cho mỗi từ vào một không gian vectơ liên tục với kích thước thấp hơn nhiều."*

Tóm lại **word embedding** là phương pháp ánh xạ mỗi từ vào một không gian số thực nhiều chiều có kích thước nhỏ hơn nhiều so với kích thước từ điển.
Trước khi word embedding ra đời đã có rất nhiều phương pháp được áp dụng và nghiên cứu, từ mã hóa **one-hot encoding** cho đến **biểu diễn theo ma trận đồng nhất**,... nhưng những giải pháp này vẫn gặp phải vấn đề về chi phí tính toán. Cho đến năm 2013, nhóm Mikolov giúp giới NLP thở phào với giải pháp mới mang tên word2Vec. Từ thời điểm này hàng loạt bài toán NLP được giải quyết với độ chính xác cao hơn nhiều so với trước.

2.3. Tổng quát phương pháp

Ý tưởng chính của **word2vec** là:

- Thay vì lưu thông tin xuất hiện của các từ bằng cách đếm trực tiếp như ma trận đồng xuất hiện, word2vec học để đoán từ lân cận của tất cả các từ.
- Các giải pháp sau đó như Glove cũng tương tự word2vec được đề xuất bởi nhóm Pennington năm 2014.
- Tính toán nhanh hơn và dễ dàng thêm dữ liệu mới vào trong mô hình.

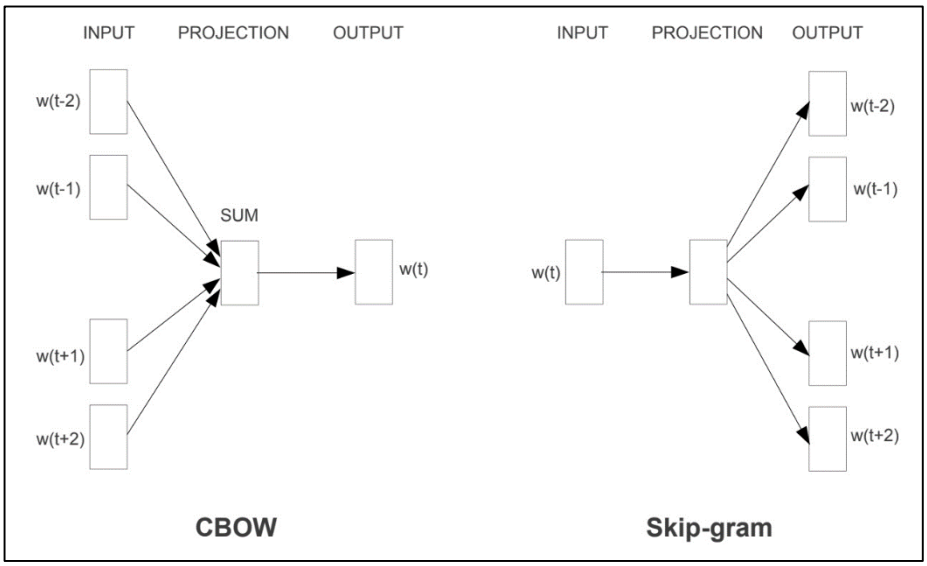
Phương pháp:

Đoán các từ lân cận trong cửa sổ m của mỗi từ 1.1. Với mỗi từ t = 1 . . . T1.2. Đoán các từ trong cửa sổ bán kính m của tất cả các từ.

Hàm mục tiêu(object function): tối ưu hợp lý hóa cực đại của bất kì từ ngữ cảnh (context word) đối với một từ đang xét hiện tại (center word).

$$J(\theta) = -\frac{1}{T}\prod_{t=1}^{T}\prod_{-m\leq j\leq m, j\neq 0} p(w_{t+j}|w_t;\theta)$$

Các kiến trúc khác nhau: (1) cho ngữ cảnh đoán từ hiện tại (CBoW), và (2) cho từ hiện tại đoán ngữ cảnh (Skip-gram).



Hình 3: Các loại mô hình trong word2vec

CBoW:

- Cho các từ ngữ cảnh
- Đoán xác suất của một từ đích

Skip-gram:

- Cho từ đích
- Đoán xác suất của các từ ngữ cảnh

Tốc độ học mô hình:

Model	*Vector Dimensionality*	*Training Words*	*Training Time*	*Accuracy [%]*
Collobert NNLM	50	660M	2 months	11
Turian NNLM	200	37M	few weeks	2
Mnih NNLM	100	37M	7 days	9
Mikolov RNNLM	640	320M	weeks	25
Huang NNLM	50	990M	weeks	13
Skip-gram (hier.s.)	1000	6B	hours	66
CBOW (negative)	300	1.5B	**minutes**	**72**

Hình 4: So sánh hiệu năng của các mô hình

Hình trên[1] so sánh hiệu năng của các mô hình khác nhau. Mô hình CBOW cho kết quả ấn tượng với thời gian học bằng phút thay vì bằng giờ hay tháng như các mô hình trước đây. Tất nhiên đây không phải là những so sánh tuyệt đối công bằng do các mô hình được đánh giá trên các tập dữ liệu khác nhau trên những máy tính có tốc độ xử lý khác nhau. Nhưng phần nào cũng thể hiện được ưu điểm mà word2vec mang lại.

Chương 3 - Cài đặt giải pháp

3.1. Khái quát CNN (Convolutional Neural Network)

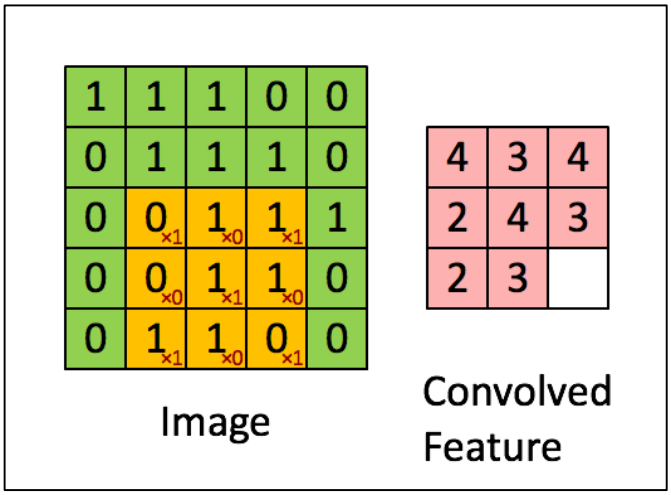
Khi nghe đến CNN chúng ta thường nghĩ đến Computer Vision. CNN hầu hết đảm nhận các công việc liên quan đến phân loại ảnh (Image Classification) và xuất hiện

[1] Theo Mikolov và cộng sự tại #mostly.ai/summit

trong đa số các hệ thống thị giác máy tính hiện nay từ tự động gán nhãn trên Facebook cho đến xe tự tự vận hành.

3.1.1. Tích chập (Convolution) là gì ?

Một trong những cách hiểu đơn giản về tích chập đó là cửa sổ trượt (ma trận) trên một ma trận.



Hình 5: Tích chập

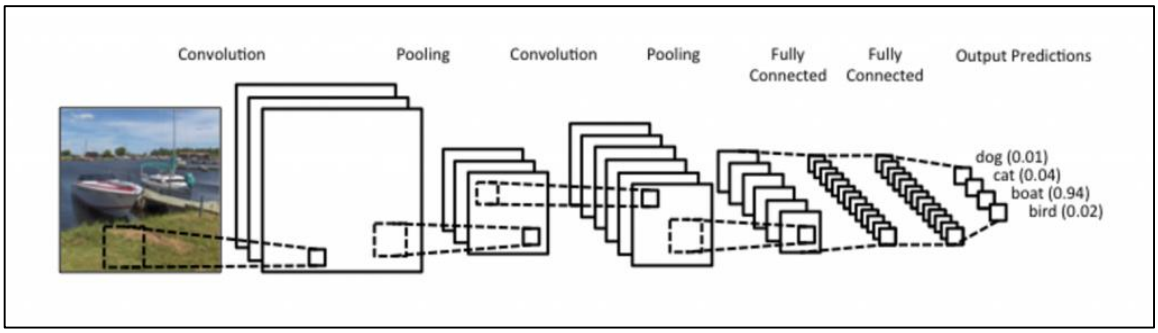
Bộ lọc sẽ biến đổi ma trận f và g (thường được gọi là ma trận đầu vào) để tạo thành một ma trận mới. Đối với ảnh đầu vào thì bộ lọc sẽ được gia giảm các tham số trên ma trận lọc để có thể dễ dàng lấy được các đặc trưng của ảnh.

3.1.2. Mạng nơ-ron tích chập (Convolutional Neural Network) là gì ?

Mạng nơ-ron tích chập (**CNN**) đơn thuần chỉ gồm một vài lớp (layer) tích chập (convolutional) kết hợp với hàm truyền phi tuyến (non-linear activation function) như **Relu** hay **tanh** để tạo ra thông tin trừu tượng hơn cho các lớp (layer) tiếp theo.

Các lớp liên kết với nhau thông qua cơ chế tích chập. Lớp tiếp theo là kết quả của lớp trước, nhờ vậy ta có các kết nối cục bộ. Nghĩa là mọi nơ-ron ở lớp tiếp theo sinh ra từ ma trận lọc (**filter**) áp đặt lên một vùng cục bộ nơron lớp trước đó.

Trong bài toán phân loại ảnh, CNN sẽ học để phát hiện các đặc trưng của ảnh thông qua các cạnh trong lớp (**layer**) đầu tiên, sau đó sử dụng các cạnh này để phát hiện các đặc trưng phức tạp hơn trong các lớp tiếp theo. Lớp cuối cùng là lớp phân loại.

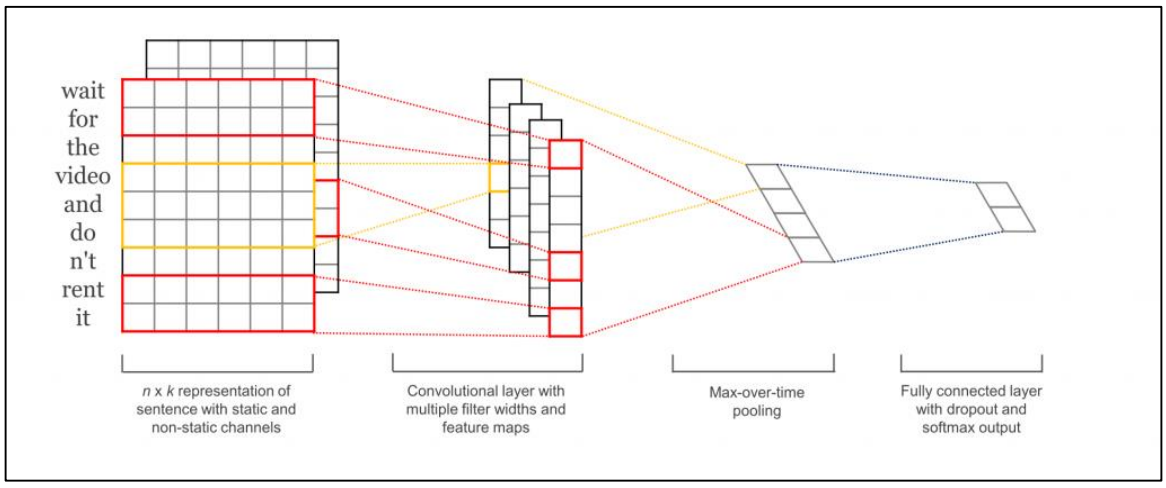


Hình 6: Hoạt động của CNN

3.2. Áp dụng CNN vào NLP

Thay vì các pixel của ảnh, dữ liệu đầu vào của các bài toán NLP sẽ là các câu, các đoạn văn bản được trình bày dưới dạng ma trận. Mỗi hàng trong ma trận sẽ đại diện cho một từ. Thông thường các từ sẽ được vectơ hóa thông qua các kĩ thuật như word2vec hay Glove.

Trong CNN bộ lọc (ma trận) sẽ có chiều rộng bằng với chiều rộng của ma trận đầu vào:



Hình 7: Văn bản dưới góc nhìn theo CNN

Ta có thể thấy các lớp đầu tiên được sử dụng để giảm chiều của các vectơ đầu vào. Lớp tiếp theo thực hiện các phép tích chập sử dụng nhiều bộ lọc có kích thước khác nhau. Ví dụ: trượt hơn 3, 4 hoặc 5 từ cùng một lúc. Tiếp theo, gộp tối đa kết quả của lớp tích chập thành một vectơ đặc trưng dài, thêm các lớp pooling nhằm giữ lại các đặc trưng kèm theo bỏ các phần thừa và phân loại kết quả bằng cách sử dụng lớp softmax.

Chương 4 – Đánh giá kiểm thử

Phân loại văn bản là một bài toán học có giám sát, để giải quyết một bài toán liên quan đến phân loại văn bản ta có thể thực hiện các bước sau:

- Chuẩn bị dữ liệu (Dataset Preparing)
- Xử lý thuộc tính dữ liệu (Feature Engineering)
- Xây dựng mô hình (Build Model)
- Tinh chỉnh mô hình và cải thiện hiệu năng (Improve Performance)

Bài toán: *Phân loại một đoạn bài viết thuộc 1 trong 10 thể loại*:

- Chính trị xã hội
- Đời sống
- Khoa học
- Kinh doanh
- Pháp luật
- Sức khỏe
- Thế giới
- Thể thao
- Văn hóa
- Vi Tính

4.1. Tiền xử lý dữ liệu (Processing Data)

Bộ dữ liệu sẽ sử dụng được download tại: https://github.com/duyvuleo/VNTC
Đây là tập dữ liệu bao gồm 2 phần: **33759** file cho phần train và **50373** cho phần test. Các dữ liệu được lấy tự động từ các trang báo như *vnexpress.vn, tuoitre.vn, thanhnien.vn, nld.com.vn.*

Giả sử ta có một đoạn văn bản sau: "*Thủ tướng Đức nhận lời tham dự lễ kỷ niệm D-Day Thủ tướng Gerhard Schroeder sẽ trở thành nguyên thủ Đức đầu tiên tham dự lễ kỷ niệm ngày quân đồng minh đổ bộ lên bãi biển Normandy trong Thế chiến II (mang mật danh D-Day) vào tháng 6 tới. Ông đã chấp nhận lời mời tham gia lễ kỷ niệm 60 năm ngày D-Day của Tổng thống Pháp Jacque Chirac. Phát ngôn viên của Berlin cho biết: "Tổng thống Chirac đã mời Thủ tướng Schroeder từ trước lễ Giáng sinh và ông đã nhận lời ngay. Thủ tướng cảm thấy rất vui khi được mời". Năm 1994, cố tổng thống Pháp Francois Mitterrand đã không mời cựu thủ tướng Đức Helmut Kohl đến dự lễ kỷ niệm 50 năm sự kiện D-Day...*"

Phía trên là văn bản thuộc thể loại **Thế giới.**

4.2. Chuẩn bị dữ liệu

Trước hết dữ liệu phải được loại bỏ các ký tự đặc biệt trong văn bản như dấu chấm, dấu phẩy, dấu đóng ngoặc, mở ngoặc,... bằng cách sử dụng thư viện **gensim**. Sau đó chúng ta sẽ sử dụng thư viện **PyVi** để tách từ Tiếng Việt. Điểm khác biệt trong Tiếng Việt là một từ được kết hợp bởi một hay nhiều từ khác nhau, ví dụ như: bắt_đầu, liên_minh,... khác với các ngôn ngữ khác như Tiếng Anh, các từ được phân cách với nhau thông qua khoảng trắng. Việc tách từ đảm bảo ý nghĩa của từ được toàn vẹn.

Chúng ta sẽ đưa mỗi bài báo về một cặp **(x, y)**. Trong đó **x** là văn bản đã được xử lý, **y** là nhãn của thể loại bài báo đó.

Kết quả thu được của 1 văn bản đã xử lý có dạng như sau: "*thủ_tướng đức nhận_lời tham_dự lễ kỷ_niệm day thủ_tướng gerhard schroeder sẽ trở_thành nguyên_thủ đức đầu_tiên tham_dự lễ kỷ_niệm ngày quân đồng_minh đổ_bộ lên bãi biển normandy trong thế_chiến ii mang mật_danh day vào tháng tới ông đã chấp_nhận lời mời tham_gia lễ kỷ_niệm năm ngày day của tổng_thống pháp jacque chirac phát_ngôn_viên của berlin cho biết tổng_thống chirac đã mời thủ_tướng schroeder từ trước lễ giáng_sinh và ông đã nhận_lời ngay thủ_tướng cảm_thấy rất vui khi được mời năm cố tổng_thống pháp francois mitterrand đã không mời cựu_thủ_tướng đức helmut kohl đến dự lễ kỷ_niệm năm sự_kiện day...*"

4.3. Feature Engineering

Ở bước này dữ liệu ở dạng văn bản sẽ được xử lý về dạng vector thuộc tính có dạng số học. Phương pháp **TF-IDF** (Term Frequency - Inverse Document Frequency), là một phương pháp phổ biến trong xử lý văn bản. Được tính theo công thức sau đây:

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

TF(t, d) - Tần suất xuất hiện của một từ trong văn bản = (Số lần xuất hiện của một từ trong văn bản **d**) / (Tổng số từ trong văn bản)

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

IDF(t, d, D) - Tần số nghịch của một từ trong một tập văn bản = **log**(Tổng số từ của tập văn bản / Số lượng văn bản chứa từ **t** bên trong văn bản **d** thuộc tập **D**)

Giá trị **TF-IDF**: *tfidf(t, d, D) = tf(t, d) x idf(t, d, D)*

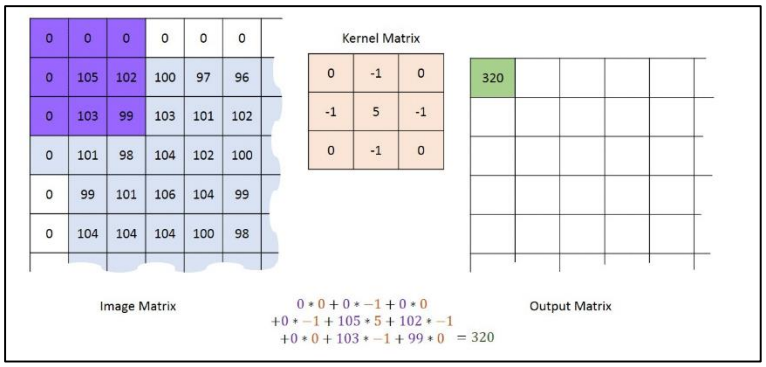
Sau khi thực hiện TF-IDF, chúng ta dễ dàng nhận thấy, ma trận mà chúng ta thu được có kích thước rất lớn, đòi hỏi việc tính toán làm tiêu tốn thời gian. Ví dụ, chúng ta có 100.000 văn bản và bộ từ điển bao gồm 50000 từ, khi đó ma trận mà chúng ta thu được sẽ có kích thước là *100000 * 50000*. Giả sử mỗi phần tử được lưu dưới dạng float32 - 4 byte, bộ nhớ mà chúng ta cần sử dụng là: *100000×50000×4=20000000000* byte, tức là tiêu tốn gần **18.63G** bộ nhớ, khó có thể lưu hết vào RAM được. Để giải quyết vấn đề này chúng ta sẽ sử dụng thuật toán **SVD** (Singular Value Decomposition) nhằm mục đích giảm chiều ma trận mà chúng ta thu được, mà vẫn giữ nguyên được các thuộc tính của ma trận gốc ban đầu. Chúng ta sẽ đưa vector ban đầu về thành vector **300** chiều.

4.4. Xây dựng mô hình

Trước khi bước CNN, chúng ta sẽ tiến hành thử nghiệm trên tập dữ liệu. Khi dữ liệu được đưa vào tuần tự, chúng ta sẽ sử dụng các lớp **Dense** cho việc đảm bảo đầu ra của các lớp cho trước, cũng như các lớp ẩn. Được minh họa bằng mô hình Deep Neural Network trong ví dụ cho kết quả sau **20** lần huấn luyện, độ chính xác ước tính khoảng **90.02%**.

CNN thông thường được sử dụng cho các tác vụ liên quan đến xử lý ảnh. Ở đây mô hình CNN được chọn bởi bằng việc sử dụng các bộ lọc xoắn để nắm bắt các mối quan hệ địa phương, nhờ đó CNN có khả năng lọc được các ngữ cảnh gần giữa các từ trong câu.

Đối với một từ được biểu diễn dưới dạng vector **300** chiều, một đoạn văn bản có độ dài trung bình khoảng **nx300** với n là độ dài của đoạn văn bản.



Hình 8: Bộ lọc theo chiều dọc với stride bằng 1

Trong hình trên ta có thể thấy, đối với bộ lọc có kích thước **3x3**, bộ lọc sẽ di chuyển với bước (**strides**) bằng 1 và tính tổng kết quả nhân ma trận, ghi nhận lại ở ma trận đầu ra. Nếu chúng ta tưởng tượng mỗi hàng dữ liệu là một từ trong câu, thì nó sẽ không học hiệu quả vì bộ lọc chỉ nhìn vào một phần của vectơ từ tại một thời điểm. CNN ở trên được gọi là mạng thần kinh chuyển đổi 2D do bộ lọc đang di chuyển trong không gian 2 chiều.

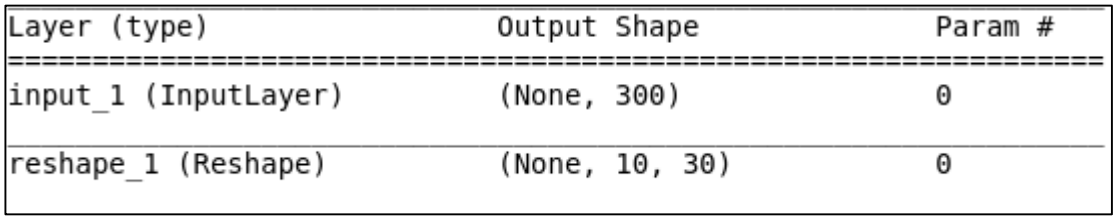
Do đó việc sử dụng bộ lọc một chiều (**1D)** sẽ hữu hiệu hơn. Nếu chiều rộng của cột lọc bộ lọc giống như chiều rộng cột dữ liệu, thì nó không có chỗ để trượt(**strides**) theo chiều ngang và chỉ sải bước theo chiều dọc. Ví dụ, nếu ta có đoạn văn bản với kích thước 45x300, thì chiều rộng cột của bộ lọc cũng sẽ có 300 cột và chiều dài của hàng (chiều cao) sẽ tương tự như khái niệm n-gram.

Nếu một bộ lọc có kích thước **2x300** được áp dụng vào 1 trong **45x300** ma trận, chúng ta sẽ có được 1 ma trận đầu ra với kích thước là **44x1**. Trong trường hợp của bộ lọc **1D**, chiều cao của ma trận đầu ra có thể được tính với công thức:

$$OutputHeight = \frac{H - F_h}{S} + 1$$

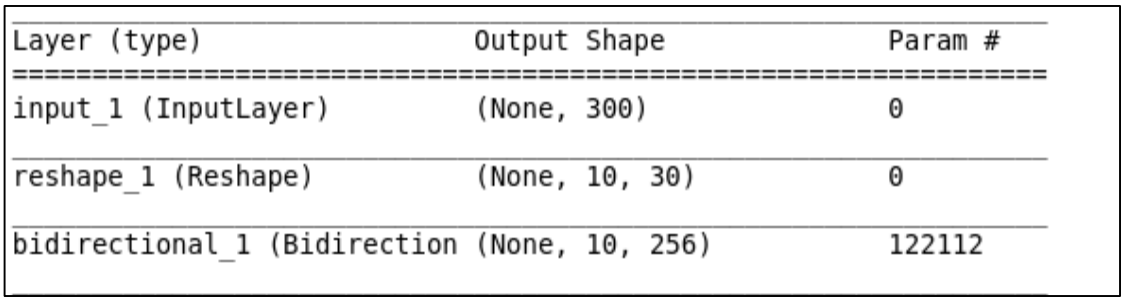
Trong đó:
H: Chiều cao ma trận đầu vào
Fh: Chiều cao bộ lọc
S: Số bước trượt

Mô hình được xây dựng sẽ nhận tham số đầu vào là một ma trận đầu vào với kích thước ban đầu là **nx300**.



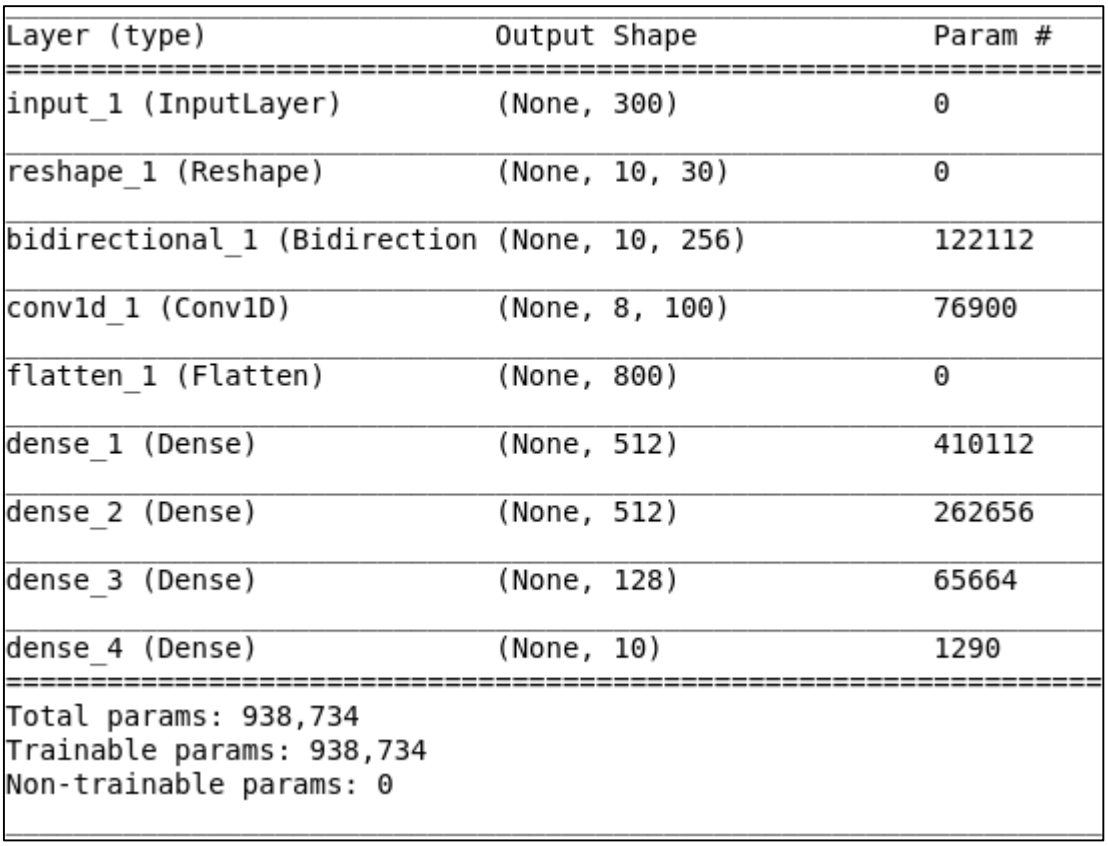
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 300)	0
reshape_1 (Reshape)	(None, 10, 30)	0

Do kích thước ban đầu của đầu vào rất lớn, nên ta chúng sẽ sử dụng hàm **Reshape**, để làm giảm kích thước mục đích của việc này là đến **layer** tiếp theo, thay vì việc sử dụng **Embedding** cho **layer** đầu tiên với chiều dài **input** cố định sẽ làm ảnh hưởng đến hiệu suất của mô hình, chúng ta sẽ sử dụng layer có tên **Bidirectional** dùng để cung cấp chuỗi đầu vào giống như đầu vào của lớp đầu tiên và cung cấp một bản sao đảo ngược của chuỗi đầu sang lớp tiếp theo với tham số là **GRU**.



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 300)	0
reshape_1 (Reshape)	(None, 10, 30)	0
bidirectional_1 (Bidirection	(None, 10, 256)	122112

Đến lúc này ta có thể áp dụng một **100** bộ lọc với chiều cao là **3**, điều này làm giảm số chiều của **layer** trước đó một cách tự nhiên. Tiếp sau đó là các lớp **Flatten, Dense** để cho ra các kết quả cuối cùng.



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 300)	0
reshape_1 (Reshape)	(None, 10, 30)	0
bidirectional_1 (Bidirection	(None, 10, 256)	122112
conv1d_1 (Conv1D)	(None, 8, 100)	76900
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 512)	410112
dense_2 (Dense)	(None, 512)	262656
dense_3 (Dense)	(None, 128)	65664
dense_4 (Dense)	(None, 10)	1290

Total params: 938,734
Trainable params: 938,734
Non-trainable params: 0

Hình 9: Mô hình tổng thể

Kết quả sau 20 lần huấn luyện, mô hình cho ra tỷ lệ với độ chính xác vào khoảng **90,14%**.

Phần kết luận

Việc sử dụng mạng nơ-ron tích chập (CNN) hay các mô hình deep learning trong bài toán xử lý ngôn ngữ tự nhiên đem lại kết quả tương đối khả quan, nhưng đối với từng loại văn bản khác nhau đòi hỏi các kỹ thuật xử lý khác nhau, thêm vào đó việc lựa chọn các tham số thích hợp cho mô hình cũng đòi hỏi việc mất thời gian.

Mô hình có thể cho kết quả tốt hơn nếu sử dụng bộ pre-trained word2vec cho ra số lượng vector được biểu diễn nhiều hơn, kết hợp với một mô hình CNN có thêm độ sâu và số các hidden layer tăng lên.

Tài liệu tham khảo

[1] Y. Kim, Convolutional Neural Networks for Sentence Classification, 2015.

[2] N. T. Duyen, N. X. Bach, and T. M. Phuong, An empirical study on sentiment analysis for Vietnamese, International Conference on Advanced Technologies for Communications (ATC 2014), pp. 309–314, 2014.

[3] Hồ Tú Bảo, Lương Chi Mai, Về xử lý tiếng Việt trong công nghệ thông tin, 2015.

[4] Deshpande, The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3), 2018.

[5] Ronan Collobert (NEC Labs America, Princeton NJ), Jason Weston (Google, New York, NY), L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa (NEC Labs America, Princeton NJ), Natural Language Processing (almost) from Scratch.