Topic-detection
Nhận biết chủ đề

Nhóm 5

1. **Phát biểu bài toán:**
   a. **Input:**
   - Dữ liệu của tập huấn luyện là văn bản, có dạng:
     __label__<label> <text>
     Trong đó, __label__ là tiền tố nhận diện cho nhãn
     <label> là một trong 23 nhãn
     <text> là văn bản tương ứng với nhãn <label>
     Cụm __label__<label> và <text> cách nhau một khoảng trắng (whitespace)
     Mỗi cặp __label_<label> <text> được trình bày trên 1 dòng
   - Tập dữ liệu huấn luyện có 16000 ví dụ tương ứng với 23 nhãn
     Phân bố của 23 nhãn trong tập huấn luyện như sau:

STT	Nhãn	Phân bố		STT	Nhãn	Phân bố	
		Số ví dụ	%			Số ví dụ	%
1	Nha_dat	2542	15.89	13	Lam_dep_va_the_hinh	283	1.77
2	Do_an_va_do_uong	2356	14.725	14	Nha_va_vuon	262	1.64
3	Kinh_doanh_va_Cong_nghiep	2356	14.725	15	Giai_tri	211	1.32
4	Tai_chinh	1379	8.62	16	May_tinh_va_thiet_bi_dien_tu	202	1.26
5	Mua_sam	1169	7.31	17	Suc_khoe_va_benh_tat	194	1.21
6	Du_lich	881	5.51	18	Khoa_hoc	160	1
7	Chinh_tri	796	4.975	19	Thoi_quen_va_so_thich	160	1
8	Giao_duc	697	4.36	20	Phap_luat	131	0.82
9	Nghe_thuat	609	3.81	21	Giao_thong	108	0.675
10	Mang_internet_va_vien_thong	593	3.71	22	The_thao	76	0.475
11	Sach	415	2.59	23	Cong_nghe_moi	33	0.21
12	Con_nguoi_va_xa_hoi	387	2.42		**Tổng số**	**16000**	**100**

   - Tập dữ liệu kiểm tra có 10017 ví dụ dạng văn bản. Mỗi ví dụ trên 1 dòng.

   b. **Output:**
   - Với mỗi dữ liệu văn bản dạng <text> trong tập kiểm tra đưa vào mô hình cần cho ra một nhãn tương ứng có dạng __label__<label>
   - Tập kiểm tra bao gồm 10017 ví dụ cần được gán nhãn

Vậy mô hình cần cho ra một file chứa 10017 nhãn có dạng __label__<label>, mỗi nhãn trên một dòng và tương ứng với ví dụ trong tập kiểm tra.

   c. **Mục tiêu cố gắng đạt được:**
   - Gán nhãn với độ chính xác càng cao càng tốt

   d. **Các cản trở:**
   - Phân bố dữ liệu giữa các nhãn không đồng đều
   - Có các nhãn có nội dung gần nhau, như Kinh_doanh_va_Cong_nghiep với Tai_chinh, Lam_dep_va_the_hinh với Suc_khoe_va_benh_tat
   - Có nhiều ví dụ có nội dung không rõ ràng, như Bản tin Tài chính Kinh doanh (không biết nên chia vào nhãn Tai_chinh hay Kinh_doanh_va_Cong_nghiep) hoặc tường thuật một trận đấu e-sport (nên xếp vào mục Giai_tri nhưng lại có các viết giống The_thao, gây nhầm lẫn cho mô hình)

2. Đặc trưng của dữ liệu:

Dữ liệu thuộc dạng văn bản, được viết theo phong cách tự nhiên, không trang trọng, không cấu trúc. Cách lựa chọn các đặc trưng của dữ liệu như sau:

- **Với từ:**
  - Part-of-speech tagging: Gán thẻ cho từ theo từ loại hoặc nhiệm vụ trong câu, ví dụ: chủ ngữ danh từ, chủ ngữ động từ, động từ vị ngữ chính, tính từ vị ngữ chính,… tuy nhiên việc gán thẻ này cho tiếng Việt chưa phổ biến và cho kết quả với độ chính xác không cao, và về đánh giá tổng quan chung thì không ảnh hưởng đến chủ đề của văn bản
    => không chọn
  - Word embedding: Thể hiện về mặt quan hệ của từ với từ, có thể tự huấn luyện (BoW, TF-IDF) hoặc dùng bộ embedding có sẵn (word2vec, fasttext,…), có ảnh hưởng trực tiếp đến chủ đề của văn bản
    => chọn
  - Keyword: Mỗi chủ đề sẽ có một số keyword, nếu trong đoạn văn bản xuất hiện keyword của một label thì xác suất phân vào label đó cao hơn, tuy nhiên cần biết trước keyword
    => không chọn
- **Với câu:**
  - Tách câu: Nếu đoạn văn bản có nhiều câu thì tách câu ra và xử lý từng câu một rồi hợp kết quả của các câu lại tạo thành kết quả cho văn bản, tuy nhiên các xử lý này khá vụn và có xác suất lỗi cao do câu chủ đề có thể có nội dung khác các câu còn lại trong văn bản và không xác định được câu chủ đề, ví dụ: câu chủ đề: Phap_luat và các câu còn lại: Con_nguoi_va_xa_hoi
    => không chọn
  - Tách từ: Tách một câu, hoặc văn bản thành các từ ảnh hưởng trực tiếp đến chủ đề của văn bản, tuy nhiên tách từ tiếng Việt còn chưa hoàn thiện và kết quả chính xác chưa cao
    => không chọn
  - Parsing: Cây phân tích phụ thuộc về cú pháp và ngữ nghĩa của câu, không có ảnh hưởng nhiều đến chủ đề của văn bản
    => không chọn

3. Tiền xử lý dữ liệu:

Dữ liệu được tách ra làm hai phần: phần nhãn có dạng __label__<label>, phần văn bản có dạng <text>. Với phần văn bản <text>, tách văn bản thành các token bằng khoảng trắng (whitespace). Với mỗi token thực hiện như sau:

a. Loại bỏ tag HTML và đường dẫn:

Loại bỏ tất cả những tag HTML như tag <s> </s> hoặc các đường dẫn như http://www.dulich.com/. Mặc dù 'dulich' có thể hiểu là 'Du lịch', có liên quan đến chủ đề của văn bản nhưng cách chuyển đường dẫn thành những từ bình thường khó và không có độ chính xác cao nên lựa chọn loại bỏ.

b. Loại bỏ dấu:

Dấu câu không ảnh hưởng đến chủ đề của văn bản và không cần thiết để tách câu vì nhóm quyết định không chọn đặc trưng về câu nên có thể loại bỏ.

c. Loại bỏ số:

Số không ảnh hưởng đến chủ đề của văn bản nên có thể loại bỏ.

d. Loại bỏ kí tự đặc biệt:

Trong một số văn bản có sử dụng kí tự đặc biệt như kí tự bông hoa, mặt cười,… hoặc các kí tự của nước ngoài như tiếng Hàn, tiếng Nhật, tiếng Trung Quốc,… Các kí tự này có thể hoặc không liên quan đến chủ đề của văn bản nhưng việc chuyển đổi nghĩa của kí tự đặc biệt sang chữ rất khó và không có độ chính xác cao nên lựa chọn loại bỏ.

4. Phân lớp dữ liệu theo nhãn:

Để tiện cho việc đánh giá độ tốt của mô hình, tập huấn luyện gồm 16000 ví dụ được chia lại thành 3 tập: train (80%), dev (10%), test (10%). Việc chia 3 tập này là ngẫu nhiên nhưng vẫn đảm bảo có đủ 23 nhãn trong mỗi tập.

a. Solution 1:

- Chuyển văn bản thành chuỗi token
- Chuyển các token thành vector bằng Bag of Words
- Đồng thời nhóm các kí tự trong token thành các char-N-gram và chuyển chúng thành vector
  Vector thể hiện token là tổng hợp của vector BoW và char-N-gram.
- Nhóm các token thành các N-gram
- Chuyển các N-gram thành vector bằng Bag of N-grams (giống như Bag of Words nhưng cho N-gram thay vì word)
  Vector thể hiện N-gram là tổng hợp của vector thể hiện token và BoN.
- Truyền vector trên vào mô hình phân lớp bằng neural network gồm một tầng ẩn và một tầng softmax.
  Mô hình được huấn luyện với tập train. Các siêu tham số của mô hình sử dụng tune tự động để tìm giá trị macro-f1 cao nhất trong tập dev trong thời gian 3600 giây, và kiểm tra độ tốt lại với tập test. Các siêu tham số được giới hạn trong khoảng:
  - Epoch: từ 1 đến 100
  - Learning rate: từ 0.01 đến 5.00
  - Chiều của vector truyền vào mô hình: từ 1 đến 1000
  - Cỡ của N-gram: từ 1 đến 5
  - Cỡ của bucket (trong bước chuyển token thành vector): từ 10000 đến 10000000
  - Cỡ của char-N-gram: min: từ 1 đến 3; max: từ 1 đến min+3
  - Chiều của vector char-N-gram: 1 đến 4
- Sau khi có mô hình thì truyền dữ liệu của tập kiểm tra vào để dự đoán nhãn.

b. Solution 2:

Giống solution 1 nhưng thay vì tạo embedding cho token bằng BoW thì sử dụng embedding đã được huấn luyện sẵn (FastText: Word vectors for 157 languages: Vietnamese)

c. Solution 3:

- Lặp lại các bước tương tự solution 1 và 2 nhưng với các tập train, dev, test khác và 3 loại embedding khác (sonvx/word2vecVN)
- Sau khi có các mô hình thì truyền dữ liệu của tập kiểm tra để dự đoán nhãn
- Các dự đoán này được lưu lại và thống kê nhãn nào xuất hiện nhiều nhất với mỗi ví dụ thì lấy nhãn đó, loại bỏ các nhãn còn lại.
  Nếu xảy ra trường hợp có 2 nhãn có số lần xuất hiện bằng nhau thì lấy nhãn được lưu vào trước.
  Thứ tự chạy và lưu nhãn của các mô hình (tương đương độ ưu tiên nếu có 2 nhãn bằng nhau) là: FastText embedding 1, FastText embedding 2, BoW embedding 1, BoW embedding 2, word2vecVN embedding dim400, word2vec embedding dim300.