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

———————— * ————————

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

ỨNG DỤNG HỌC SÂU TRONG BÀI TOÁN TÁCH TỪ TIẾNG VIỆT

Sinh viên thực hiện : **Lương Tuấn Dũng**
Lớp CNTT-TT 2.03 - K59
Giáo viên hướng dẫn: **TS. Nguyễn Kiêm Hiếu**

HÀ NỘI
Ngày 28 tháng 12 năm 2018

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên
   Họ và tên sinh viên: Lương Tuấn Dũng.
   Điện thoại liên lạc: 0772382213 Email: tuanluong04011996@gmail.com
   Lớp: CNTT-TT 2.03 - K59 Hệ đào tạo: Đại học chính quy.
   Đồ án tốt nghiệp được thực hiện tại: Đại học Bách Khoa Hà Nội.
   Thời gian làm ĐATN: Từ ngày 03/09/2018 đến 28/12/2018.

2. Mục đích nội dung của ĐATN
   Thử nghiệm áp dụng các mô hình học sâu để giải quyết bài toán tách từ trong tiếng Việt.

3. Các nhiệm vụ cụ thể của ĐATN

   - Tìm hiểu về bài toán tách từ và một phần lý thuyết về ngôn ngữ tiếng Việt
   - Tìm hiểu về các mô hình học máy và học sâu, đặc biệt là các mô hình được áp dụng vào bài toán tách từ
   - Đề xuất thử nghiệm những mô hình đã có và đề xuất một vài kiến trúc mô hình mới nhằm giải quyết bài toán tách từ trên bộ dữ liệu tiếng Việt.
   - Thử nghiệm các mô hình trên dữ liệu VLSP và đánh giá kết quả

4. Lời cam đoan của sinh viên:
   Tôi - *Lương Tuấn Dũng* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Kiêm Hiếu*. Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

   Hà Nội, ngày 28 tháng 12 năm 2018
   Tác giả ĐATN

   Lương Tuấn Dũng

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

   Hà Nội, ngày 28 tháng 12 năm 2018
   Giáo viên hướng dẫn

   TS. Nguyễn Kiêm Hiếu

LỜI CẢM ƠN

Quãng đời sinh viên trôi nhanh như một cơn gió, để rồi khi nhìn lại, chợt thấy bản thân chẳng có thành tích gì quá nổi bật. Hy vọng đồ án tốt nghiệp này sẽ là một thành công nho nhỏ, một điểm sáng đáng nhớ nhất của đời sinh viên.

Những lời đầu tiên của đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành tới các thầy cô của trường đại học Bách Khoa Hà Nội và đặc biệt là các thầy cô trong viện công nghệ thông tin, những con người tràn đầy nhiệt huyết, không chỉ truyền dạy cho em những kiến thức quý báu, mà còn truyền cho em ngọn lửa nhiệt huyết, tình yêu với khoa học công nghệ. Những lời dạy của thầy cô sẽ là hành trang quý báu để em có thể vững tin hơn trên những bước đường tương lai. Em chúc thầy cô luôn khỏe mạnh, thành công trong công việc và tiếp tục dìu dắt những thế hệ sinh viên xuất sắc trong tương lai.

Em xin gửi lời cảm ơn tới TS. Nguyễn Kiêm Hiếu, người thầy đã hướng dẫn và giúp đỡ em rất nhiều trong đồ án tốt nghiệp này. Những đánh giá, những lời khuyên, những kiến thức mà thầy đã chỉ bảo cho em trong suốt quá trình làm đồ án là cơ sở cốt lõi để em có thể hoàn thành được đồ án này. Con cũng gửi lời cảm ơn đến gia đình, bố, mẹ và chị gái đã sát cánh và dõi theo từng bước con đi. Cảm ơn những người bạn đã ở bên mình, chia sẻ từ niềm vui đến những giây phút khó khăn nhất, cùng nhiệt huyết với những dự định trong tương lai. Em xin cảm ơn tới anh em trong đội Game Research and Develop - công ty cổ phần VNG đã luôn giúp đỡ và tạo điều kiện để em có thể sử dụng các máy tính hiệu năng cao để phục vụ cho đồ án.

Tuy đã rất nỗ lực để có một đồ án thực sự đáng nhớ, nhưng vẫn không thể tránh khỏi những thiếu sót. Em rất mong nhận được những nhận xét quý báu của thầy cô để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Sinh viên

Lương Tuấn Dũng

TÓM TẮT NỘI DUNG

Với đề tài *ứng dụng học sâu trong bài toán tách từ tiếng Việt*, nội dung của đồ án sẽ đi sâu vào bài toán tách từ, học sâu và áp dụng học sâu vào bài toán tách từ. Chi tiết bao gồm các đầu mục như sau:

- Chương 1: **Giới thiệu đề tài**, đặt vấn đề, mục tiêu phạm vi của đề tài và định hướng giải pháp.
- Chương 2: **Cơ sở lý thuyết**, giới thiệu về bài toán tách từ và đặc điểm về từ trong tiếng Việt. Hệ thống lý thuyết về học máy, học sâu sẽ được sử dụng trong khuôn khổ đồ án.
- Chương 3: **Các phương pháp giải quyết bài toán**, đề xuất phương pháp giải quyết bài toán và giới thiệu một số phương pháp đã được áp dụng.
- Chương 4: **Thử nghiệm và đánh giá**, trình bày các kết quả thử nghiệm và đánh giá kết quả thử nghiệm.
- Chương 5: **Kết luận**, trình bày các kết quả đạt được và hướng phát triển trong tương lai.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục từ viết tắt và các thuật ngữ

LSTM	Long Short Term Memory
BiLSTM	Bi-directional Long Short Term Memory
CRF	Conditional Random Field
MTL	Multi-Tasks Learning
CBOW	continuous bag of word
PMI	Pointwise Mutual Information
CRFs	Conditional Random Fields
LM	Language Modeling

Chương 1

Giới thiệu đề tài

1.1 Đặt vấn đề

Trong lĩnh vực xử lý ngôn ngữ tự nhiên, tách từ là một bài toán cơ sở, đã xuất hiện từ rất sớm, được nhiều chuyên gia tiếp cận và tìm hướng giải quyết. Cụ thể yêu cầu của bài toán tách từ là chia nhỏ một đoạn văn bản thành các từ thành phần. Thoạt nghe vấn đề có vẻ đơn giản, vì đối với một số ngôn ngữ sử dụng bảng chữ cái la tinh làm gốc (điển hình là tiếng Anh), các từ, về mặt ngữ âm có thể là đơn hoặc đa âm tiết, tuy nhiên khi viết đều được viết dưới một mẫu tự chung là một chuỗi của các chữ cái và sử dụng phiên âm để thể hiện lên cách phát âm của chúng. Song vẫn có những từ được viết không theo chuẩn này, ví dụ: *hip hop*, *ice box...* Những từ này là những từ mới được thêm vào từ điển, trung bình một năm có khoảng 1000 từ mới được thêm vào từ điển và những từ loại này chỉ chiếm một số lượng rất nhỏ trong số đó. Do đó rõ ràng việc sử dụng khoảng trắng hay các dấu phân tách từ là đủ để có thể giải quyết bài toán trong những ngôn ngữ thuộc loại này. Tuy nhiên, với nhiều ngôn ngữ, ví dụ như tiếng Nhật Bản, tiếng Trung Quốc với mẫu tự tượng hình có sự phân tách được thể hiện giữa các câu, với tiếng Thái và tiếng Lào, sự phân tách được thể hiện giữa các cụm từ và câu, và đối với tiếng Việt, sự phân tách đoạn văn bản được thể hiện dưới dạng các âm tiết. Chính vì vậy, đối với các ngôn ngữ này, bài toán tách từ lại trở nên vô cùng khó và thách thức.

Rất nhiều phương pháp tiếp cận khác nhau để giải quyết bài toán này đã được áp dụng, từ tính toán thống kê cho đến các phương pháp học máy, học sâu. Tại thời điểm hiện tại, kết quả cho bài toán tách từ ở nhiều ngôn ngữ đều đã đạt được những kết quả rất khả quan: độ chính xác với tiếng Trung hay tiếng Nhật đều đã đạt 97-98%, và đối với tiếng Việt độ chính xác cao nhất theo báo cáo hiện đang là trên 98%.

Kết quả hiện tại cho bài toán tách từ là cao, tuy nhiên vẫn chưa thể nói là đã đạt đến trần của bài toán này. Hơn nữa, đây lại là một bài toán căn bản, là cơ sở để xử lý các bài toán mức cao hơn trong xử lý ngôn ngữ tự nhiên. Việc tăng độ chính xác trong bài toán tách từ sẽ đồng thời làm tăng độ chính xác trong các bài toán bậc cao đó. Với việc lựa chọn đề tài này, em mong muốn được tìm hiểu sâu hơn về bài toán, đồng thời đề xuất những thử nghiệm, những hướng tiếp cận mới đối với bài toán tách từ tiếng Việt.

1.2 Mục tiêu phạm vi của đề tài

Mục tiêu của đề tài là thử nghiệm áp dụng các mô hình học sâu để giải quyết bài toán tách từ trong tiếng Việt. Khai thác được sức mạnh của các mô hình học sâu, đề xuất được các mô hình mới áp dụng cho bài toán, đồng thời đưa ra được mô hình dựa trên học sâu mà có kết quả cạnh tranh được với các mô hình tách từ hiện tại.

Phạm vi của đề tài chỉ tiếp cận bài toán tách từ cho tiếng Việt, đồng thời chỉ tập chung vào phương pháp sử dụng các mô hình học sâu để giải quyết bài toán. Các phương pháp tiếp cận khác sẽ được đề cập nhưng chỉ giới thiệu và không chi tiết, Các kết quả đánh giá mô hình sẽ được so sánh với các phương pháp khác trên cùng tập dữ liệu sẽ được mô tả chi tiết trong chương 4.

1.3 Định hướng giải pháp

Sử dụng các mô hình học sâu, với cốt lõi mô hình là mạng nơ-ron hồi quy để xử lý bài toán tách từ như là một bài toán gán nhãn chuỗi. Thử nghiệm với các biến thể của mô hình, đồng thời thử nghiệm nhiều cách biểu diễn khác nhau của văn bản đầu vào để đánh giá mức độ ảnh hưởng tới mô hình cũng như độ chính xác đạt được với các thử nghiệm này.

Chương 2

Cơ sở lý thuyết

2.1 Bài toán tách từ trong tiếng Việt

2.1.1 Tổng quan về từ trong tiếng Việt

Dựa trên cơ sở về cấu trúc và chức năng, tiếng Việt được xếp vào loại hình ngôn ngữ đơn lập. Về mặt từ vựng, từ trong tiếng Việt không thay đổi hình thái. Hình thái của từ không thể hiện bất cứ điều gì về mặt quan hệ hay chức năng về ngữ pháp. Từ dù đặt trong câu hay đứng một mình đều hoàn toàn không thay đổi về hình thái của nó. Thay vào đó, quan hệ hay chức năng về ngữ pháp được thể hiện thông qua hư từ và trật tự của các từ trong câu.

ví dụ:

đã học → *đang học* → *sẽ học*
em đi bơi → *em bơi đi*

Trong ví dụ trên hư từ *đã, đang, sẽ* được sử dụng để làm thay đổi ý nghĩa về mặt thời gian của từ. Hay khi trật tự các từ trong câu thay đổi thì ngữ pháp câu cũng như ý nghĩa câu cũng thay đổi theo.

Từ trong tiếng Việt được được cấu thành bởi các đơn âm tiết. Một từ có thể được tạo bởi một hoặc nhiều âm tiết ghép thành. Dựa trên cách kết hợp các âm tiết được mô tả trong báo cáo hướng dẫn tách từ của VLSP 2013 [1], các từ trong tiếng Việt được chia thành các loại sau:

Từ đơn những từ được cấu thành bởi một âm tiết. Trong từ đơn lại được chia ra ba loại cụ thể:

- ***Thực từ*** là từ đơn, có ý nghĩa độc lập, có chức năng định danh (gọi tên các sự vật, hiện tượng, thuộc tính, phẩm chất, quan hệ). Hầu hết các từ này đều là các từ vựng cơ bản trong tiếng Việt, đã có từ lâu đời: *cha, mẹ, con, cỏ, cây, hoa, lá,...* Hay các từ mượn Ấn-Âu nhưng đã được Việt hóa: *tim, gan, buồng, phòng, cồn, xăng, săm, lốp,...* Hoặc những từ Hán-Việt được dùng độc lập: *tuyết, bút, học, cao, thấp,...*
- ***Hư từ*** hư từ là từ đơn không có ý nghĩa độc lập, không có chức danh định danh. Bao gồm các phụ từ, liên từ, giới từ: *đã, đang, sẽ, là, của, vì, bởi, tuy, nên, nếu, của, bằng,...*
- ***Tình thái từ*** là từ đơn đã mất ý nghĩa từ vựng và ý nghĩa ngữ pháp cụ thể, có chức năng như một phương tiện biểu thị tình thái. Bao gồm

thán từ và trợ từ: *à, ư, nhỉ, nhé, ạ, nào, đâu, vậy,...*

Từ ghép đẳng lập là từ tạo bởi hai thành tố có ý nghĩa thực kết hợp với nhau theo quan hệ bình đẳng về nghĩa. Hai thành tố bao giờ cũng có cùng phạm trù ngữ nghĩa hoặc có quan hệ logic với nhau. Trật tự giữa các thành tố nói chung có thể thay đổi được. Ví dụ: *quần áo - áo quần, đỏ đen - đen đỏ, ốm đau - đau ốm, chung riêng - riêng chung.*. Từ ghép đẳng lập còn được chia nhỏ hơn thành hai loại:

- ***Từ ghép đẳng lập gốc Việt*** là từ ghép trong đó hai thành tố đều là từ gốc Việt. Ví dụ: *đất nước, trời đất, đỏ đen, may rủi,...* Trong đó *đất nước, trời đất* gồm hai thành tố có sự gần nhau về nghĩa, *đỏ đen, may rủi* gồm hai thành tố có sự trái ngược nhau về nghĩa.
- ***Từ ghép đẳng lập gốc Hán*** là từ ghép trong đó hai thành tố đều là từ gốc Hán. Ví dụ: *giang sơn, mĩ lệ, học tập, ân nghĩa,...*
- ***Từ ghép đẳng lập gồm một thành tố gốc Việt một thành tố gốc Hán*** Ví dụ: ***binh*** *lính, bụng* ***dạ****, gan* ***dạ****, nuôi* ***dưỡng***. (trong đó thành phần in đậm là thành tố gốc Hán)

Từ ghép chính phụ do hai thành tố (A và B) kết hợp với nhau theo quan hệ không bình đẳng, một thành tố chính có ý nghĩa khái quát và một thành tố phụ có ý nghĩa hạn định. Ý nghĩa về mặt từ vựng do thành tố chính quyết định, thành tố phụ có vai trò bổ sung, phân loại, chuyên biệt hóa, sắc thái hóa cho thành tố chính. Thành tố chính có thể dùng thành từ, còn thành tố phụ thì không có khả năng này. Tương tự như từ ghép đẳng lập, từ ghép chính phụ cũng được chia thành hai loại:

- ***Từ ghép chính phụ gốc Việt*** vị trí của hai thành tố A và B trong cấu tạo từ ghép chính phụ gốc Việt là chính trước - phụ sau. Trong đó, từ ghép chính phụ gốc Việt còn được chia thành hai loại là từ ghép chính phụ bậc một và từ ghép chính phụ bậc hai. Từ ghép chính phụ bậc một được cấu thành bởi thành tố A là từ đơn và thành tố B là một từ đơn, hoặc một từ ghép, hoặc một tổ hợp từ, ví dụ: *cá* mè, *cá* trê, *cá* nhà táng, *xe* đạp, *xe* tăng, *xe* cứu thương,... Từ ghép chính phụ bậc hai được cấu thành bởi thành tố A là từ ghép và thành tố B là một từ đơn hoặc một từ ghép hoặc một tổ hợp từ, ví dụ: *cá trắm* đen, *cá trắm* cỏ, *máy bay* trực thăng, *máy bay* tiêm kích,...
- ***Từ ghép chính phụ gốc Hán*** trường hợp thông thường, hai thành tố A và B trong từ ghép chính phụ gốc Hán được sắp đặt theo trật tự phụ trước – chính sau. Trong đó, thành tố A là từ đơn được dùng độc lập hoặc không độc lập và thành tố B là một từ đơn, hoặc một từ ghép, ví dụ: dân *ca*, đồng *ca*, khải hoàn *ca*, bác *học*, văn *học*, kinh tế *học*,... ngoài ra còn có trường hợp thành tố B là từ gốc Anh: ampe *kế*, logic *học*,... Bên cạnh đó, có trường hợp thành tố A và B trong từ ghép chính phụ gốc Hán được sắp đặt theo trật tự chính trước – phụ sau; trường hợp này A là động từ và B là từ đơn gốc Hán được dùng độc lập hoặc không độc lập, ví dụ: *thuyết* giảng, *thuyết minh*, *thuyết* phục, *đả* đảo, *đả* động, *đả* kích,...

Từ láy Từ láy phổ biến là từ gồm hai tiếng (song tiết, hai âm tiết), trong đó một tiếng có hình thức lặp lại âm của tiếng kia. Các tiếng kết hợp với nhau vừa có sự hài hoà về ngữ âm, vừa có giá trị biểu cảm, gợi tả. Thường chỉ có một tiếng có nghĩa và một tiếng mờ nghĩa: *chậm chạp* (*chậm* có nghĩa), *long lanh* (*long* có nghĩa), *lúng túng* (*túng* có nghĩa), *long tong* (*tong* có nghĩa); hoặc cả hai tiếng đều mờ nghĩa: *khấp khểnh, lênh đênh, lênh khênh, lêu nghêu, lung linh,...* Xét trên cấu tạo từ, từ láy được chia thành các loại sau:

- ***Kiểu AA'*** (A là tiếng gốc, tiếng chính; A' là tiếng láy của A) *chậm chạp, nhanh nhảu, lành lặn, văn vẻ,...*
- ***Kiểu A'A*** (A là tiếng gốc; A' là tiếng láy của A) *đèm đẹp, lành lạnh, nho nhỏ, đo đỏ,...*
- ***Kiểu AA*** lặp lại hoàn toàn âm của tiếng gốc. Phần lớn là các từ tượng thanh: *ào ào, ầm ầm, ha ha, độp độp, khặc khặc,...* Ngoài ra còn có các từ mà lặp hoàn toàn âm của tiếng gốc một cách đơn điệu (ý nghĩa thay đổi rất ít hoặc không thay đổi so với tiếng gốc): *đen đen, đêm đêm, xanh xanh, run run, quen quen,...*
- ***Kiểu ABB*** (B là thành tố của từ ghép chính phụ AB) *đen sì sì, đỏ lòm lòm, nông choèn choèn, tối om om, xanh lè lè,...*
- ***Kiểu AB'B*** (B' là tiếng láy của B, AB là từ ghép chính phụ) *đen trùi trũi, đỏ hoen hoét, đỏ hon hỏn, cao lêu nghêu, dài đuồn đuỗn,...*
- ***Kiểu ABC*** *dửng dừng dưng, sạch sành sanh, loét loèn loẹt...*
- ***Kiểu AA'AB*** (A là tiếng đầu của từ ghép AB; A' là tiếng láy của A; A' có cấu tạo dạng *xa*, trong đó *x* là phụ âm đầu của A, *a* là phần vần có giá trị hoà phối ngữ âm cho cả khối) *ấm a ấm ức, đủng đa đủng đỉnh, long la long lanh, nhí nha nhí nhảnh,...*

Dạng lặp

- ***Kiểu AA*** (lặp hoàn toàn tiếng gốc để chỉ số lượng nhiều, hoặc chỉ mức độ cao; cả hai thành tố đều là danh từ) *ai ai, đâu đâu, đêm đêm, lớp lớp, ngày ngày, người người, nhà nhà, sáng sáng, tháng tháng, tối tối,...*
- ***Kiểu AAA*** (Thường là từ tượng thanh) *ầm ầm ầm, ha ha ha,...*
- ***Kiểu AABB*** (AB là từ ghép đẳng lập, trong đó A ngược nghĩa với B) *đi đi lại lại, hư hư thực thực, lên lên xuống xuống, quần quần áo áo, ra ra vào vào,...*
- ***Kiểu ABAC*** (B và C thường tạo thành từ ghép đẳng lập, trong đó B ngược nghĩa với C, nhưng đôi khi cũng có thể B đồng nghĩa với C; A là yếu tố chen vào đầu và giữa tổ hợp BC) *chạy ngược chạy xuôi, chẳng nói chẳng rằng, dặn đi dặn lại, đá đi đá lại, đảo đi đảo lại, khất quanh khất quẩn, khoắng lấy khoắng để, khua đi khua lại, người này người nọ, trông trước trông sau, về lâu về dài,...*

Năm loại từ trên là cách phân loại từ thường thấy trong tiếng Việt. Tuy nhiên đối với bài toán tách từ, khi các văn bản thuộc đa dạng các văn phạm khác nhau, ta cần bổn sung thêm một số từ loại cụ thể hơn, thông dụng trong bài toán này:

Từ ghép phụ gia là các từ được tạo bằng cách ghép các yếu tố có khả năng cấu tạo từ cao vào trước hay sau một từ đơn hoặc từ ghép khác. Danh sách các yếu tố và các ví dụ bao gồm

- ***Bán + N = N*** *bán nguyên âm, bán thành phẩm*
- ***Bán + A = A*** *bán tự động, bán vũ trang*
- ***Bất + A = A*** *bất bình đẳng, bất hợp lý, bất khả thi*
- ***Bất + V = V*** *bất hợp tác, bất tuân lệnh*
- ***Bất + N = N*** *bất đẳng thức, bất phương trình*
- ***Cố + N = N*** *cố thủ tướng, cố bộ trưởng, cố nhà văn*
- ***Cựu + N = N*** *cựu thủ tướng, cựu bộ trưởng, cựu giám đốc*
- ***Đa + N = N*** *đa phương tiện, đa tác vụ, đa chính phủ*
- ***Đại + N = N*** *đại ban doanh, đại cử tri, đại bộ phận*
- ***Hữu + N = A*** *hữu hạn, hữu hình*
- ***Hữu + V = A*** *hữu sinh, hữu dụng*
- ***Liên + N = N*** *liên bang, liên ngành, liên hiệp*
- ***Nguyên + N = N*** *nguyên thủ tướng, nguyên bộ trưởng*
- ***Nhà + V = N*** *nhà ngoại giao, nhà phê bình*
- ***Phi + N = N*** *phi lợi nhuận, phi chính phủ, phi nông nghiệp*
- ***Phó + N = N*** *phó giáo sư, phó chủ nhiệm, phó phòng*
- ***Siêu + N = N*** *siêu cầu thủ, siêu lợi nhuận, siêu giai cấp*
- ***Siêu + V = V*** *siêu dẫn, siêu thoát*
- ***Siêu + A = A*** *siêu thực, siêu trọng*
- ***Tái + V = V*** *tái cơ cấu, tái đầu tư, tái chỉ đạo*
- ***Tiểu + N = N*** *tiểu vương quốc, tiểu hòa thượng*
- ***Trưởng + N = N*** *trưởng bộ môn, trưởng phòng, trưởng thôn*
- ***Tối + A = A*** *tối kiên cường, tối thông minh*
- ***Vô + N = A*** *vô đạo đức, vô kỷ luật, vô gia cư*
- ***Vô + V = A*** *vô học, vô địch, vô can*
- ***Vô + V = P*** *vô kể, vô luận*
- ***N + hóa = V*** *công nghiệp hóa, tri thức hóa*
- ***A + hóa = V*** *hiện đại hóa, hợp pháp hóa*
- ***N + kiều = N*** *Hoa kiều, Việt kiều*
- ***N + trưởng = N*** *đại đội trưởng, tiểu đoàn trưởng*
- ***V + viên = N*** *cộng tác viên, lập trình viên, quan sát viên*
- ***N + viên = N*** *Đảng viên, công an viên*

Thành ngữ là tổ hợp từ có tính hoàn chỉnh cao về hình thức và ý nghĩa. Ý nghĩa của thành ngữ được hiểu trên cả một thành ngữ hoàn chỉnh, không phải trên từng tiếng thành phần. Ví dụ: *đục nước béo cò* thể hiện việc trục lợi nhân tình thế lộn xộn rối ren.

Quán ngữ là các tổ hợp từ được dùng đi dùng lại, lâu ngày trở nên ổn định về hình thức và ý nghĩa. Khác với thành ngữ, ý nghĩa của quán ngữ liên quan mật thiết tới ý nghĩa của các tiếng thành phần. Một số lượng lớn quán ngữ được dùng trong các văn bản liên kết, nhập đề, nhấn mạnh. Ví dụ: *có thể nói rằng, suy cho cùng, nói tóm lại, bên cạnh đó, đẹp như tiên, của đáng tội, lễ với nghĩa...*

Tên riêng tên người, tên địa danh, tên tổ chức cũng được coi là một đơn vị từ vựng. Trên phương diện bài toán tách từ, các từ thuộc loại này được tách theo quy định tách từ thông thường, riêng với danh từ riêng thì gộp làm một từ.

Ngày - tháng - năm giữ nguyên cả khối và coi là một từ với các dạng: *dd-mm-yy, dd/mm/yy, dd/mm*, ví dụ: *4-1-1996, (ngày) 2-9,...*, và tách thành từng đơn vị số, dấu, chữ như thông thường, ví dụ: *tháng 11 năm 1996, năm 73,...*

Số - chữ số - ký hiệu giữ nguyên cả khối và coi như là một từ với các dạng: công thức hóa học hoặc biểu thức toán học ($H + O_2 = H_2O$, $1 + 2 = 3$), biểu diễn liên tục một con số chính xác bằng số hoặc bằng chữ (*1.500, 1 500, 1500, một nghìn năm trăm*), biểu diễn cả số và ký hiệu một cách liên tục (*9g25, 1h30p*). Tách với dạng ký hiệu đơn vị đứng trước hoặc sau, không chen vào giữa các thành phần số (*20ha, 20kg, 30$*), biểu diễn hỗn hợp cả số và chữ (*70 phần trăm, hai tỉ rưỡi*).

Dấu câu mỗi dấu câu đều được coi là một từ.

Từ nước ngoài đối với các từ, thuật ngữ, khái niệm thì mỗi khối ký tự viết liền được coi là một từ. Tên riêng xử lý như đã đề cập. Duy với tên người và tên đệm viết tắt thì vẫn coi cả khối là một từ.

Chữ viết tắt mỗi khối viết liền là một từ (*CHXH, CSGT,...*). Đối với chữ viết tắt là một bộ phận của tên riêng thì cả khối tên riêng đó sẽ là một từ (*ĐHBK Hà Nội, Cty CP VNG,..*).

2.1.2 Bài toán tách từ

Tách từ là một bài toán kinh điển trong lĩnh vực xử lý ngôn ngữ. Yêu cầu của bài toán là với mỗi văn bản đầu vào, ta cần tách văn bản thành các bộ phận nhỏ, và mỗi bộ phận là một từ. Trong tiếng Việt, việc tách từ phụ thuộc vào văn phạm, ý nghĩa câu văn và cấu tạo từ (như đã đề cập trong mục 2.1.1). Với việc số lượng loại từ đa dạng, cùng với cấu tạo phức tạp của từng loại từ này, dễ dẫn đến nhập nhằng trong việc xác định biên của từ. Đây cũng là vấn đề khó nhất cần giải quyết trong bài toán.

2.1.3 Tính ứng dụng của bài toán

Từ là đơn vị mang ý nghĩa hoàn chỉnh nhỏ nhất trong ngôn ngữ. Chính vì thế tách từ là bài toán quan trọng bậc nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên, là tiền đề để hình thành và giải quyết các bài toán khác trong lĩnh vực. Một số bài toán hoàn toàn thực hiện trên mức từ như: gán nhãn từ loại, phân tích cú pháp,... Ngoài ra các bài toán cấp cao hơn như: dịch máy, hệ thống hội thoại, phát hiện đạo văn, tóm tắt văn bản,... đều rất khó để tiếp cận nếu chỉ sử dụng câu là thành phần mang ý nghĩa hoàn chỉnh nhỏ nhất, hầu hết các cách tiếp cận hiện thời đều cần sử dụng từ làm đơn vị mang ý nghĩa hoàn chỉnh nhỏ nhất. Việc tách từ sai sẽ gây lỗi cho mô hình xử lý của bài toán cấp cao hơn, dẫn đến kết quả cho bài toán đó sẽ bị giảm sút. Do đó, việc xây dựng một bộ tách từ có độ chính xác cao là điều tối quan trọng.

2.2 Học máy

2.2.1 Giới thiệu về học máy

Học máy là một nhánh con của lĩnh vực trí tuệ nhân tạo. Mục tiêu của học máy là hiểu được dữ liệu, đồng thời xây dựng một mô hình có thể mô phỏng được dữ liệu, và sử dụng mô hình đó trong các tác vụ thực tế. Tuy là một lĩnh vực con của khoa học máy tính, nhưng học máy lại tiếp cận bài toán theo một cách hoàn toàn khác so với tính toán truyền thống. Thông thường, thuật toán giải quyết các bài toán sẽ được lập trình bằng một loạt các tính toán rõ ràng, để từ đó đưa đến kết quả. Tuy nhiên với học máy, máy tính sẽ được lập trình để huấn luyện một mô hình với dữ liệu đầu vào và sử dụng phân tích thống kê để đưa ra giá trị trong một khoảng xác định. Học máy xây dựng mô hình để tự động hóa quá trình ra quyết định dựa trên những tri thức đã biết là dữ liệu đầu vào.

Học máy được chia thành ba loại chính: học tập có giám sát (supervised learning), học tập không giám sát (unsupervised learning) và học tập tăng cường.

Học tập có giám sát là phương pháp xây dựng mô hình trên tập dữ liệu đã được gán nhãn đầy đủ (có đầu vào và đầu ra tương ứng). Phân lớp (classification) và hồi quy (regression) là hai dạng bài toán chính trong phương pháp này.

Học tập không giám sát là phương pháp xây dựng mô hình trên tập dữ liệu chưa được gán nhãn (chỉ có đầu vào). Liên kết (association) và phân cụm (clustering) là hai dạng bài toán chĩnh trong phương pháp này.

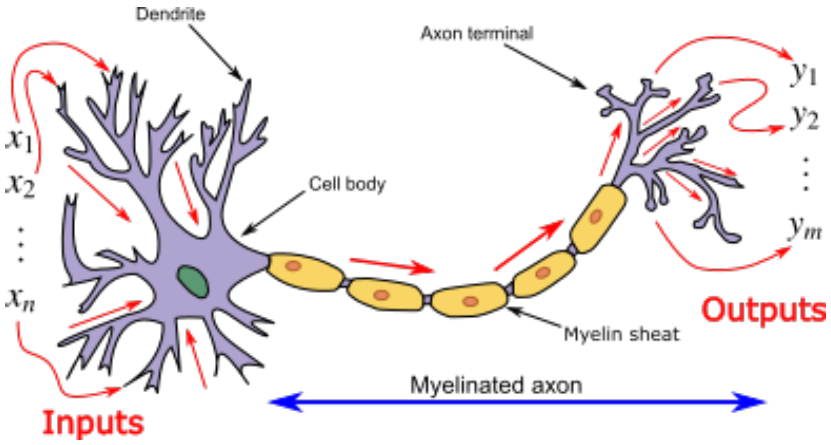
Học tập tăng cường là phương pháp xây dựng mô hình dựa trên những tương tác với môi trường động. Mọi quyết định của mô hình đều tác động đến môi trường. Phản hồi của môi trường là cơ sở để cập nhật mô hình.

Trong đồ án này, các mô hình giải quyết bài toán tách từ đều thuộc phương pháp học tập có giám sát. Dữ liệu sử dụng đều đã được gán nhãn đầy đủ.

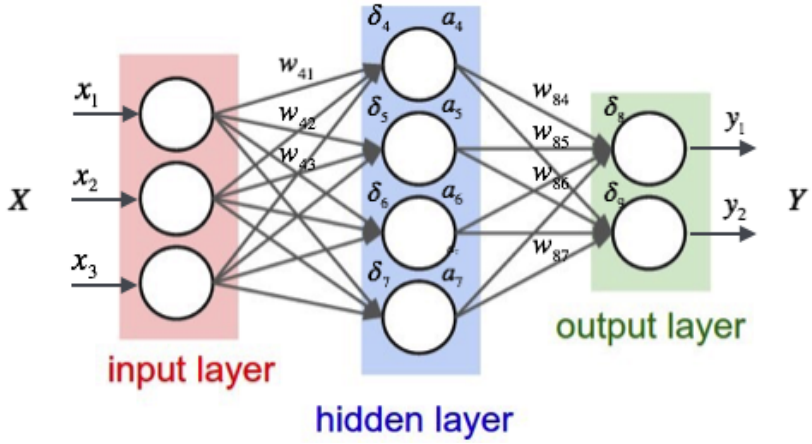
2.2.2 Mạng nơ-ron nhân tạo (Artifitial Neural Network)

Kiến trúc

Mạng nơ-ron nhân tạo là một phương pháp học máy được ra đời từ năm 1943, tuy nhiên lại phát triển một cách chậm chạp. Cho tới những năm đầu thế kỉ 21, với sự phát triển mạnh mẽ của phần cứng máy tính, mô hình này đã trở lại, được nghiên cứu phát triển cũng như ứng dụng mạnh mẽ vào nhiều lĩnh vực. Mạng nơ-ron nhân tạo lấy ý tưởng từ việc mô phỏng lại bộ não con người, với cấu tạo bao gồm nốt mô phỏng nơ-ron, cạnh nối giữa các nốt mô phỏng sợi trục nối giữa hai nơ-ron, hàm kích hoạt mô phỏng việc thay đổi trạng thái của tín hiệu trong quá trình lan truyền tín hiệu giữa hai nơ-ron. Mô hình mạng nơ-ron nhân tạo xếp các nơ-ron thành các tầng liên tiếp, tín hiệu được truyền giữa nơ-ron của các tầng với với nhau.



(a) Nơ-ron và kết nối sợi trục



(b) mô hình mạng nơ-ron đơn giản

Hình 2.1: Nơ-ron thực tế và mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo hiện đang phát triển vô cùng mạnh mẽ với đa dạng các kiến trúc mạng khác nhau, tuy nhiên vẫn có những đặc điểm chung của mạng nơ-ron nhân tạo đời đầu. Do đó, phần này chúng ta sẽ đề cập chi tiết tới mạng nơ-ron đơn giản, đó là mạng feed forward. Hình 2.1b mô tả một mạng feed forward với ba tầng: 1 tầng đầu vào, 1 tầng ẩn, và 1 tầng đầu ra. Mỗi nơ-ron của tầng đầu vào nhận giá trị đầu vào x như là một tín hiệu đầu vào, gọi n là số lượng nơ-ron của tầng đầu vào, ta có $X = \{x_1, ..., x_n\}$ là tập các giá trị đầu

vào. Các tín hiệu này được lan truyền cho các nơ-ron ở lớp kế tiếp thông qua các cạnh nối (mô phỏng kết nối sợi trục). Mỗi cạnh nối này mang một trọng số w, gọi $w_{i,j}^k$ là trọng số của cạnh nối từ nơ-ron i của tầng trước tới nơ-ron j của tầng thứ k. Giá trị tín hiệu tại mỗi nơ-ron j thuộc tầng thứ k trừ nơ-ron thuộc tầng đầu vào được tính dựa trên các tín hiệu lan truyền tới và trọng số tại mỗi cạnh mà tín hiệu đó đi qua

$$a_j^k = f(z_j^k) = f(\Sigma_i w_{i,j}^k \times a_i^{k-1} + b^k)$$

Trong đó f được gọi là hàm kích hoạt, giá trị b được gọi là bias. Một tầng được mô tả như trên, bao gồm giá trị trọng số, giá trị bias, và hàm kích hoạt được gọi là một tầng *dense*. Mục đích của hàm kích hoạt dùng để phi tuyến hóa tín hiệu vào, giúp cho việc sử dụng nhiều tầng ẩn có ý nghĩa trong quá trình học. Gọi W^k là ma trận gồm tất cả trọng số của các cạnh nối tới tầng k, việc lan truyền tín hiệu trên toàn bộ mạng khi không sử dụng hàm kích hoạt đươc mô tả bởi công thức

$$a^{out} = W^{out} \times ... \times W^2 \times W^1 \times X = W \times X$$

Việc không sử dụng hàm kích hoạt khiến cho nhiều tầng ẩn cũng chỉ như một tầng ẩn, không hề làm cho mô hình phức tạp hơn cũng như có khả năng học được nhiều đặc trưng hơn. Các hàm kích hoạt thông dụng bao gồm: sigmoid, tanh, ReLU [2],... Quá trình lan truyền tín hiệu được tiếp diễn qua từng tầng, từng tầng cho đến tầng đầu ra.

Tối ưu hóa tham số

Mô hình mạng nơ-ron được sử dụng để mô phỏng một hàm số thông qua việc thay đổi các giá trị trọng số của mô hình. Gọi hàm cần mô phỏng là $F(x)$, hàm mà mạng đã mô phỏng được là $F'(x)$. Ta cập nhật mạng sao cho hàm F' càng tiệm cận với hàm F càng tốt. Việc cập nhật này được thực hiện qua hai bước: tính toán sự khác biệt giữa hai hàm số và cập nhật tham số cho mô hình.

Sự khác biệt giữa hai hàm số được tính toán dựa trên hàm mất mát (loss function), do đó cũng được gọi là giá trị mất mát. Hàm mất mát chỉ đơn giản là một hàm có khả năng đo lường được sự khác nhau giữa hai hàm số, đồng thời nó cũng phải liên tục và khả vi để phù hợp phương pháp cập nhật tham số của mô hình. Hàm mất mát đóng vai trò quan trọng trong kiến trúc của mạng, do đó ngày càng có nhiều hàm mất mát ra đời để phục vụ cho yêu cầu của nhiều bài toán khác nhau và tăng độ chính xác của mô hình. Một số hàm mất mát có thể kể tên như: hàm khoảng cách Euclid, hàm Cross-Entropy, Hàm Triplet-loss[3], hàm CTC[4],...

Quá trình cập nhật tham số cho mô hình được thực hiện nhằm cực tiểu hóa giá trị mất mát, được thực hiện thông qua phương pháp Gradient Descent. Giá trị đầu ra được tính dựa trên giá trị các nơ-ron trên toàn mạng, do đó lỗi đầu ra được gây ra bởi toàn bộ các giá trị tại các nơ-ron trong mạng, mà các giá trị này lại được tính thông qua trọng số W, do đó ta thực hiện cập nhật trọng số W để thay đổi hàm mà mô hình đang mô phỏng. Dựa trên giá trị mất mát, các tín hiệu mất mát tương ứng với từng giá trị nơ-ron được tính dựa trên chính lỗi mà nơ-ron đó gây ra cho giá trị đầu ra. Tín hiệu mất mát được lan truyền về cho nơ-ron tại từng lớp trước, rồi thực hiện cập nhật giá trị trọng số w trên

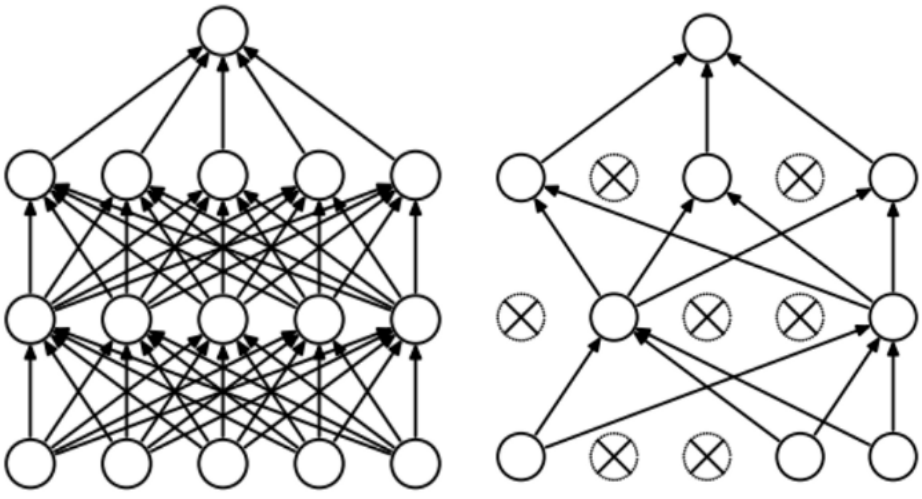
đường mà tín hiệu lỗi được lan truyền ngược về. Quá trình này được gọi là lan truyền ngược (back propagation)

Hai quá trình lan truyền tiến và lan truyền ngược được lặp đi lặp lại trong quá trình huấn luyện mô hình nhằm cập nhật tham số, để đạt được mô hình mô phỏng tốt nhất hàm số F.

Overfit và các phương pháp xử lý

Đối với các bài toán thực tế, ta cần phải xây dựng một mô hình có khả năng nhận vào mọi giá trị đầu vào có thể và đưa ra một đầu ra chính xác. Tuy nhiên, rất khó để có được thông tin về toàn bộ tập dữ liệu bao gồm các giá trị đầu vào và đầu ra tương ứng. Do đó, nhiệm vụ của học máy là xây dựng các mô hình có khả năng mô phỏng chính xác nhất yêu cầu trên với đầu vào chỉ là tập dữ liệu nhỏ. Điều đó dẫn đến việc mô hinh cần phải học được các đặc trưng chung từ tập dữ liệu nhỏ này, và bỏ qua những đặc trưng riêng mà tập dữ liệu nhỏ này có. Tưởng tượng ta có một bài toán xác định một vật có phải là cái bàn hay không, tập dữ liệu ta có toàn là bàn chữ nhật, nhưng làm sao mô hình vẫn phải học được các đặc trưng chung của chiếc bàn (mặt bàn, chân bàn,...), để với một chiếc bàn tròn hay hình dáng khác mô hình vẫn có thể dự đoán đúng. Chính điều này đã dẫn đến một khái niệm trong học máy được gọi là overfit. Đây là hiện tượng mô hình học quá khớp với tập dữ liệu, do đó không học được các đặc trưng tổng quát, để rồi với các đầu vào mới trong thực tế lại đưa ra kết quả không chính xác.

Trong mô hình mạng nơ-ron, một phương pháp thường hay sử dụng để tránh overfit là dropout[5]. Dropout là việc tắt một cách ngẫu nhiên một vài nơ-ron trong mỗi bước huấn luyện mô hình. Nơ-ron bị tắt sẽ không nhận cũng như truyền tín hiệu trong cả quá trình lan truyền tiến và lan truyền ngược.



(a) mạng không dropout (b) mạng có dropout

Hình 2.2: Droput trong mạng nơ-ron

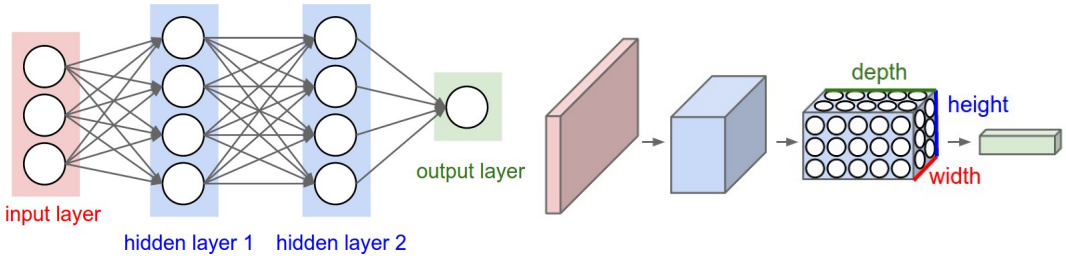
Cụ thể hơn, dropout trong mạng nơ-ron được áp dụng lên từng tầng. Tại mỗi bước huấn luyện, mỗi nơ-ron tại mỗi tầng có xác suất được giữ lại là p, $1-p$ khả năng là sẽ bị tắt đi. Trong quá trình test, toàn bộ nơ-ron đều được bật, tuy nhiên giá trị tại mỗi nơ-ron sẽ giảm đi một lượng tỉ lệ với p. Việc sử dụng

dropout tỏ ra rất hiệu quả trong việc giảm overfit của mô hình. Việc sử dụng toàn bộ nơ-ron trong quá trình học dễ khiến cho mô hình học nhiều hơn các mối quan hệ giữa các nơ-ron với nhau, làm hạn chế đặc trưng chung mà mỗi nơ-ron có thể học được, do đó, dropout giúp cho mô hình tránh bị overfit.

2.2.3 Mạng nơ-ron tích chập (convolutional neural network)

Kiến trúc

Mạng nơ-ron tích chập[6] là một mô hình cải tiến của mô hình mạng nơ-ron thông thường, nó cũng được xây dựng lên bởi các nơ-ron, các cạnh nối và các trọng số là thứ cần tối ưu trong quá trình huấn luyện mô hình. Tuy nhiên kiến trúc của mạng nơ-ron tích chập được xây dựng một cách phù hợp hơn đối với bài toán xử lý ảnh.



Hình 2.3: Mạng feed forward 3 tầng và mạng tích chập

Đối với mạng nơ-ron thông thường, mỗi nơ-ron là một giá trị, mỗi tầng sẽ là một vector chứa giá trị của các nơ-ron, các nơ-ron tầng sau được kết nối đầy đủ với tất cả các nơ-ron thuộc tầng trước. Tuy nhiên trong mạng tích chập, đầu vào được coi là ảnh kích thước ba chiều: chiều dài, chiều rộng, chiều sâu, đồng thời mỗi tầng cũng đều là các khối ba chiều. Mỗi nơ-ron ở từng tầng không kết nối đầy đủ mà chỉ kết nối tới một vùng nhỏ các nơ-ron ở lớp trước đó.

Thông thường một mạng nơ-ron tích chập được xây dựng lên bởi các tầng: *tầng đầu vào, tầng convolution (tầng tích chập), tầng ReLU, tầng pooling, và tầng fully conected (tầng kết nối đầy đủ).*

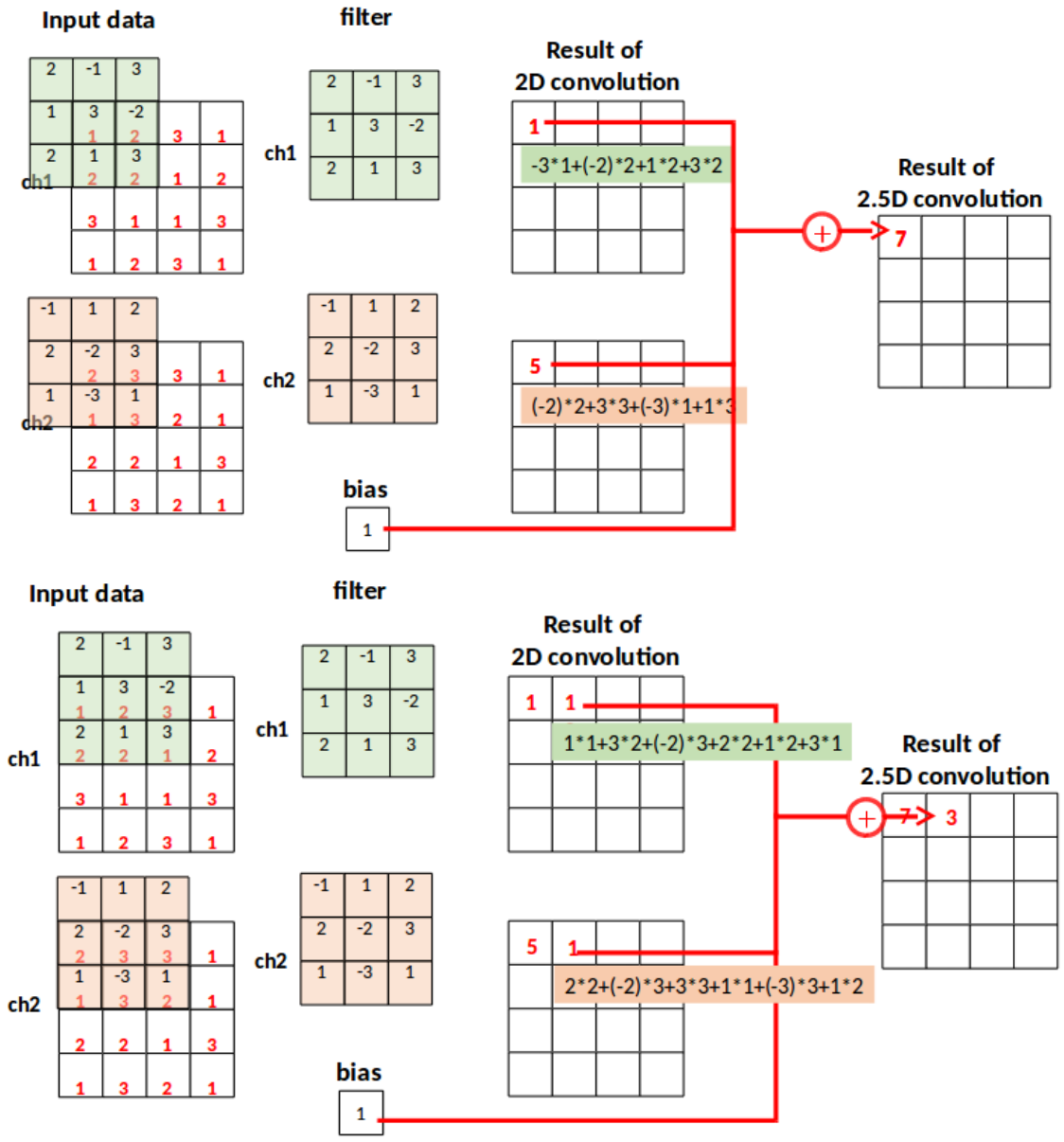
Tầng đầu vào

Do được xây dựng để phù hợp với bài toán xử lý ảnh nên đầu vào của mạng nơ-ron tích chập là một ma trận ba chiều $w \times h \times d$ (tương tự như với ảnh là chiều rộng, chiều dài và chiều sâu của ảnh). Độ sâu của ảnh có thể là 1 với ảnh đen trắng và có thể là 3 đối với ảnh màu (tương ứng với ba kênh R, B, G).

Tầng tích chập

Đặc trưng của tầng tích chập là các filter (bộ lọc) với kích thước là các khối ba chiều nhỏ, ví dụ filter thường hay sử dụng ở tầng tích chập đầu tiên có kích thước $5 \times 5 \times 3$ (trong đó 3 là chiều sâu của đầu vào). Trong quá trình huấn luyện, filter được dịch chuyển trên ảnh đầu vào theo chiều rộng và chiều dài của ảnh, rồi tính tích chập giữa filter và từng vùng ảnh mà nó đi qua để tổng hợp thành feature map. Để thực hiện quá trình dịch chuyển filter này, ta cần phải

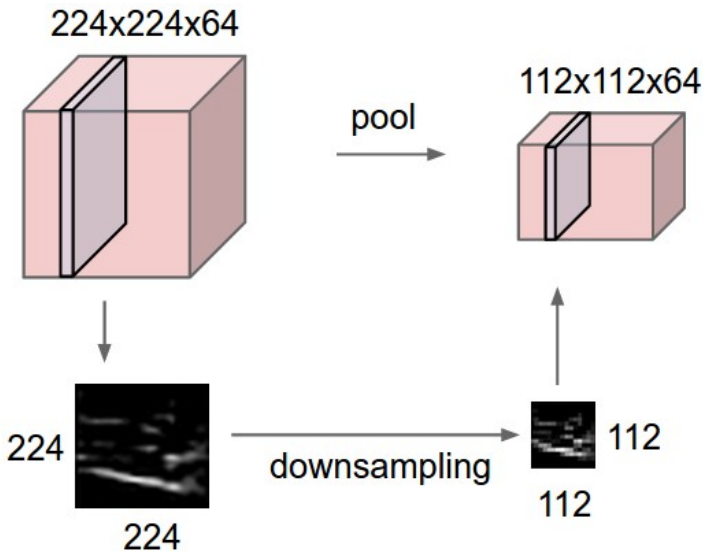
xác định một tham số là stride (bước nhảy), là giá trị quy định mỗi lần filter được dịch chuyển với khoảng cách bao nhiêu. Đồng thời padding cũng được thực hiện trên ảnh đầu vào để đảm bảo filter có thể di chuyển qua hết toàn bộ ảnh đầu vào.



Hình 2.4: Phép tích chập trong tầng tích chập

Hình 2.4 mô tả quá trình dịch chuyển filter và phép tích chập được thực hiện: kích thước ảnh đầu vào $4 \times 4 \times 2$, kích thước filter $3 \times 3 \times 2$, padding 1 xung quanh ảnh, và stride 1. Giá trị filter và bias là các trọng số cần học của mô hình. Kích thước của feature map phụ thuộc vào kích thước và số lượng filter, padding, stride, và kích thước của ảnh đầu vào. Ví dụ (tương tự hình 2.4):

- Đầu vào là ảnh kích thước $W_1 \times H_1 \times D_1 = 4 \times 4 \times 2$
- Mỗi filter kích thước $F \times F \times D_1 = 3 \times 3 \times 2$, số lượng filter $K = 10$
- Padding $P = 1$



Hình 2.5: Thay đổi kích thước trên tầng pooling

- Stride $S = 1$
- Feature map có kích thước $W_2 \times H_2 \times D_2 = 4 \times 4 \times 10$. Trong đó $W_2 = (W_1 - F + 2P)/S + 1$, $H_2 = (H_1 - F + 2P)/S + 1$, $D_2 = K$.

Tầng ReLU

Tầng Relu thực chất là tầng áp dụng một hàm kích hoạt lên từng phần tử của feature map thu được từ tầng tích chập. Hàm Relu có công thức: $ReLU(x) = max(0, x)$. Tầng ReLU không làm thay đổi kích thước của khối đầu vào.

Tầng Pooling

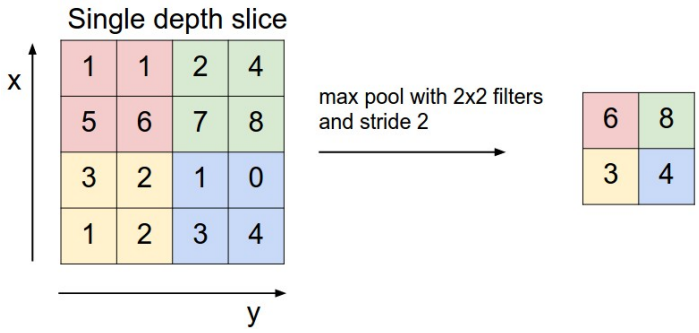
Tầng pooling có chức năng làm giảm chiều của khối đầu vào, nhờ đó mà giảm số lượng tham số và chi phí tính toán của mạng. Quá trình pooling được thực hiện trên từng lát cắt theo chiều sâu của khối đầu vào. Ví dụ trên hình 2.5, kích thước khối đầu vào giảm một nửa trên chiều rộng và chiều cao.

Hình thức phổ biến của tầng pooling là sử dụng các filter với kích thước 2×2, stride 2, dịch chuyển theo chiều rộng và chiều dài trên mỗi lát cắt của khối đầu vào. Sử dụng trên đó các hàm như: $max, min, average$ để giảm 75% kích thước của mỗi lát, tương ứng với các hàm sẽ là: *max pooling, min pooling* và *average pooling*.

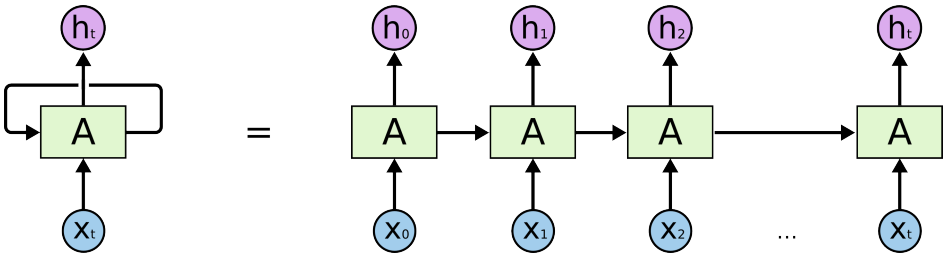
Tầng Fully Connected

Tầng fully connected là tầng cuối cùng của mô hình mạng nơ-ron tích chập, với các nơ-ron được kết nối đầy đủ với nhau tương tự như với mạng feed forward thông thường.

Trong một mạng nơ-ron tích chập, 3 tầng tích chập, ReLU, pooling thường lặp lại nhiều lần trong mạng. Toàn bộ mạng cho tới trước tầng fully connected được goi như một bộ trích chọn đặc trưng, và tầng fully connected được coi như một bộ phân loại, do đó có thể thay tầng này bằng bất cứ bộ phân loại nào, ví dụ như: SVM, softmax,... Mỗi filter được cho là học ra các đặc trưng trên ảnh,



Hình 2.6: Max pooling



Hình 2.7: Mô hình mạng nơ-ron hồi quy

filter ở lớp càng sâu thì học được đặc trưng càng phức tạp: chẳng hạn filter ở tầng tích chập đầu chỉ học được các nét thẳng, trong khi filter của các tầng tích chập sâu hơn có thể học được hình vuông, hình tam giác,... Chính vì thế mà mô hình mạng nơ-ron tích chập thể hiện được sức mạnh của mình trong việc học các đặc trưng theo vùng của dữ liệu đầu vào, do đó mạng nơ-ron tích chập còn được áp dụng vào nhiều bài toán khác không chỉ các bài toán với ảnh.

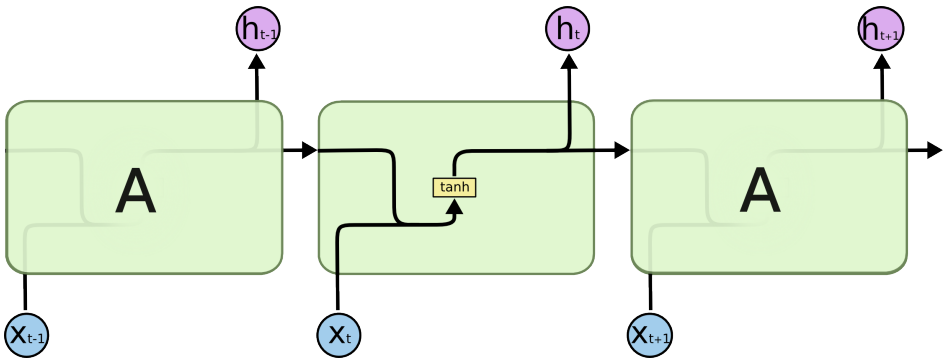
Tầng tích chập mô tả phía trên được gọi là 2D convolution (tích chập 2 chiều), với việc sử dụng filter hai chiều và dịch chuyển trên hai chiều của khối đầu vào. Một loại tầng tích chập khác là 1D convolution (tích chập 1 chiều), với việc sử dụng filter có 1 chiều cùng kích thước với 1 chiều của đầu vào, và chỉ di chuyển trên 1 chiều còn lại. Kiểu tầng tích chập 1 chiều này được sử dụng nhiều hơn trong các bài toán xử lý ngôn ngữ tự nhiên.

2.2.4 Mạng nơ-ron hồi quy (recurrent neural network)

Mạng nơ-ron hồi quy

Với mô hình mạng nơ-ron thông thường, đầu vào thường sẽ là một hoặc một vài thông tin đơn độc. Điều này càng được thể hiện rõ hơn ở việc các nơ-ron trong cùng một tầng không có kết nối với nhau. Tuy nhiên trong nhiều trường hợp, để có thể dự đoán được một điều gì đó, ta cần phải xâu chuỗi nhiều sự việc xảy ra liên tiếp. Đơn cử như việc ta chỉ có thể biết điều gì thực sự đang xảy ra trong bộ phim khi ta xem từ đầu bộ phim, chứ không thể chỉ dựa vào một frame hiện tại để có thể đoán được điều đó. Mạng nơ-ron thông thường không thể sử dụng được chuỗi thông tin như vậy, và mạng nơ-ron hồi quy ra đời để giải quyết điều đó.

Mạng nơ-ron hồi quy được mô tả dưới dạng một mô hình lặp. Đầu vào của mạng là một chuỗi thời gian các giá trị $X = x_0, x_1, ..., x_T$. Tại mỗi bước thời gian,



Hình 2.8: Nhân *Tanh* trong mạng nơ-ron hồi quy

giá trị đầu vào là x_t kết hợp với kết quả đầu ra tại bước thời gian trước đó là h_{t-1} để dự đoán ra kết quả h_t. Hình 2.7 mô tả việc trải phẳng mô hình lặp ra, và nhìn vào đó ta thấy được mạng hồi quy nhìn giống như nhiều mạng thông thường kết hợp liên tiếp lại với nhau, và các mạng này đều chia sẻ một kiến trúc và trọng số (khối A).

Khối A được gọi là nhân (cell) của mạng. Kiến trúc ban đầu của mạng nơ-ron hồi quy sử dụng nhân *Tanh* (hình 2.8). Đầu vào tại mỗi bước thời gian kết hợp với đầu ra tại bước thời gian trước qua hàm *Tanh* để dự đoán ra kết quả tại bước thời gian hiện tại.

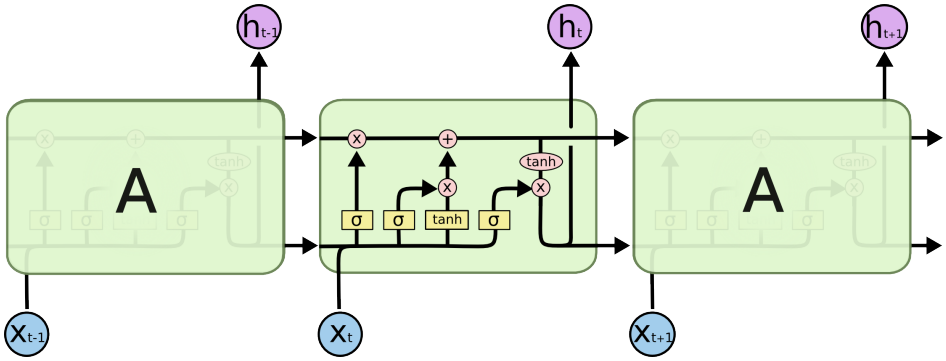
Trên lý thuyết, mô hình mạng nơ-ron hồi quy với việc sử dụng nhân *Tanh* hoàn toàn có thể ghi nhớ được thông tin theo thời gian, tuy nhiên thực nghiệm lại cho thấy rằng việc sử dụng nhân này chỉ cho phép ghi nhớ được thông tin gần với bước thời gian hiện tại mà không giữ được thông tin tại bước thời gian xa hơn. Điều này không tốt trong nhiều bài toán và không đạt được yêu cầu ban đầu đặt ra khi sử dụng mạng này.

Mạng Long Short Term Memory

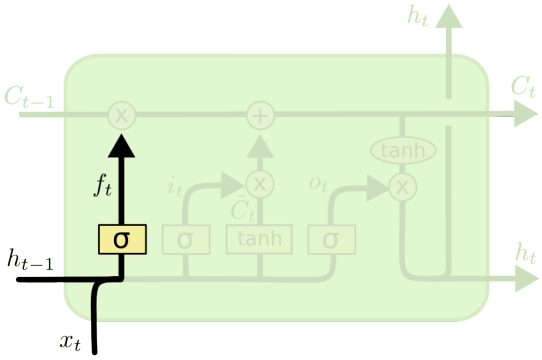
Do hạn chế trong việc ghi nhớ thông tin dài theo thời gian của nhân *Tanh*, nhân *LSTM* (*Long Short Term Memory* [7]) đã ra đời để giải quyết vấn đề này. Điểm khác biệt mấu chốt của nhân *LSTM* là bổ sung thêm giá trị cell state (hoặc memory state), nhằm ghi nhớ thông tin của toàn bộ chuỗi thời gian. Giá trị cell state chạy xuyên suốt các bước thời gian, chỉ bị tác động bởi một vài phép biến đổi tuyến tính, khiến cho thông tin mà nó lưu trữ ít bị biến đổi. Thông tin lưu trữ trong cell state được thêm vào hay bớt đi dựa vào cơ chế cổng. LSTM có 3 cổng nhằm kiểm soát cell state bao gồm: *tầng cổng đầu vào (input gate), tầng cổng quên (forget gate), tầng cổng đầu ra (output gate).*

Bước đầu tiên trong *LSTM* là quyết định xem với đầu vào tại trạng thái hiện tại, trạng thái tổng nên bỏ đi những thông tin nào. Việc này được thực hiện thông qua một tấng sigmoid có tên tầng cổng quên : nhận đầu vào là x_t và h_{t-1} (đầu vào tại thời điểm t và đầu ra tại bước thời gian $t-1$), đầu ra f_t là bộ giá trị trong khoảng $0-1$, thể hiện mỗi chiều của cell state sẽ mất đi bao nhiêu phần (1 là giữ lại toàn bộ và 0 quên toàn bộ), công thức chi tiết được thể hiện trên hình 2.10.

Bước tiếp theo ta cần quyết định xem sẽ lưu lại thông tin gì từ đầu vào mới vào cell state. Việc này được thực hiện thông qua hai bước: Bước 1, tầng sigmoid

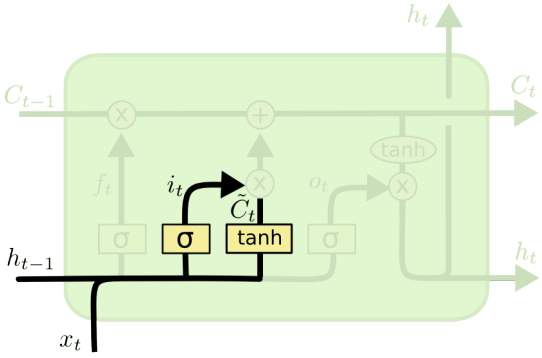


Hình 2.9: Nhân Long Short Term Memory



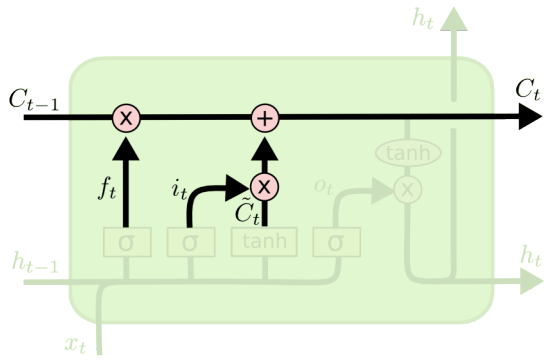
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hình 2.10: Forget gate trong LSTM



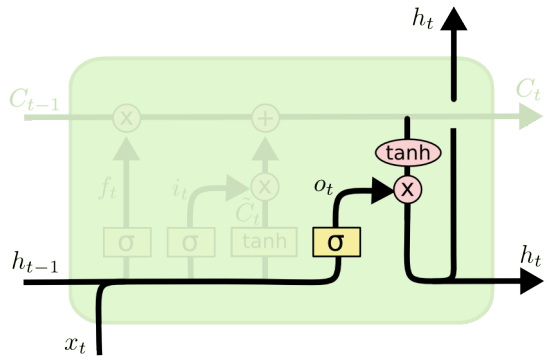
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 2.11: Input gate trong LSTM



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Hình 2.12: Cập nhật cell state trong LSTM



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Hình 2.13: Output gate trong LSTM

có tên là tầng cổng đầu vào với đầu ra là i_t quyết định xem những thông tin đầu vào sẽ được cập nhật bao nhiêu vào cell state; bước 2, tầng tanh tạo một ứng cử viên mới cho cell state $\tilde{C}_t$ dựa trên những thông tin mới từ đầu vào. Sau khi có được tất cả thông tin cần thiết, ta thực hiện cập nhật cell state cho bước thời gian t. Ta thực hiện nhân C_{t-1} với f_t để quên đi những thông tin không cần thiết trong cell state và cập nhật thông tin mới bằng cách cộng với $i_t \times \tilde{C}_t$.

Cuối cùng dựa vào đầu vào của bước thời gian hiện tại cùng với cell state vừa tính được, ta cần tính toán đầu ra tại bước thời gian hiện tại. Đầu tiên, ta chạy một tầng sigmoid có tên là tầng cổng đầu ra, đầu ra o_t quyết định phần nào của cell state là đầu ra của bước thời gian này. Sau đó, đầu ra của bước thời gian hiện tại được tính bằng cách đưa cell state qua một hàm *tanh* và nhân với o_t.

Việc tách biệt thông tin theo thời gian dài và thông tin tại từng bước thời gian khiến cho LSTM có khả năng ghi nhớ được thông tin dài hơn. Ngoài LSTM còn có nhiều biến thể khác, với việc sử dụng nhân khác như nhân GRU(Gated Recurrent Unit) nhằm giảm khối lượng tính toán trong nhân, hay sử dụng Bidirectional LSTM nhằm sử dụng thông tin theo cả hai chiều thời gian $0..T$ và $T..0$. Mạng nơ-ron hồi quy sử dụng nhân LSTM được gọi là mạng LSTM, tương tự với mạng GRU hay Bi-LSTM.

2.2.5 Conditional Random Fields

Mô hình CRFs, cụ thể hơn mô hình sẽ được sử dụng trong các thử nghiệm là mô hình CRFs tuyến tính (Linear CRF), là mô hình đồ thị xác suất vô hướng, mô hình hóa xác suất có điều kiện của chuỗi nhãn khi biết chuỗi đầu vào. Thay

vì đặc trưng chỉ được học trên dữ liệu đầu vào, mô hình CRFs còn học được các quan hệ, tính tương quan giữa các nhãn với nhau trong chuỗi nhãn. Mô hình này được sử dụng và cho kết quả rất tốt trên các bài toán gán nhãn chuỗi, đồng thời trong thời gian gần đây, CRFs còn được sử dụng như một tầng phân loại đầu ra trong các mô hình mạng học sâu.

Với mỗi chuỗi âm tiết đầu vào $X = x_1, x_2, ..., x_n$ ta thu được một chuỗi đặc trưng quan sát $Z = z_1, z_2, ..., z_n$, mỗi thành phần z_i đại diện cho véc-tơ đặc trưng cho âm tiết thứ i. Gọi $Y = y_1, y_2, ..., y_n$ là chuỗi nhãn tương ứng với các giá trị trong Z, CRFs xác định xác suất có điều kiện $p(y|z; W; b)$ như sau:

$$p(y|z; W; b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in Y(z)} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, z)}$$

Trong đó

$$\psi(y', y, z) = e^{W_{y',y}^T \times z_i + b_{y',y}}$$

Được gọi là các hàm tiềm năng (potential function), $W_{y',y}^T$ và $b_{y',y}$ lần lượt là trọng số và bias tương ứng cho cặp nhãn y' và y. Mô hình CRFs được huấn luyện sử dụng phương pháp MLE (*Maximum Likelihood Estimation*), cực đại hóa hàm *Log-Likelihood* với tập dữ liệu huấn luyện $\{(z_i, y_i)\}$ như sau:

$$L(W, b) = \sum_i log\ p(y|z, W, b)$$

Sau khi huấn luyện với mỗi chuỗi đầu vào, ta tìm chuỗi nhãn đầu ra có xác suất điều kiện lớn nhất

$$Y^* = argmax_{y \in Y(z)} p(y|z, W, b)$$

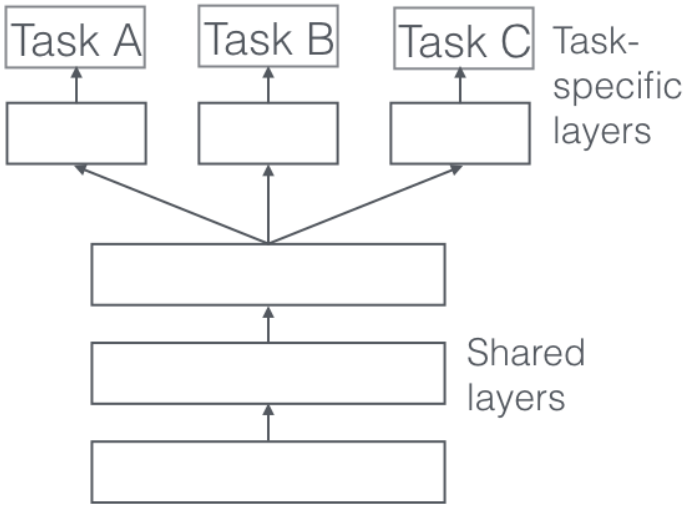
2.2.6 Multi task learning

Trong học máy, thông thường ta đi huấn luyện một mô hình cụ thể hoặc kết hợp kết quả của nhiều mô hình với nhau (ensamble models) để giải quyết một tác vụ (task) cụ thể. Tuy nhiên, khi huấn luyện các mô hình theo kiểu hướng đích như vậy (chỉ chú trọng duy nhất tới tác vụ đích), ta có thể sẽ bỏ qua những thông tin làm cải thiện chất lượng mô hình. Những thông tin thêm này là những thông tin có được từ những tác vụ liên quan tới tác vụ đích. Bằng việc chia sẻ biểu diễn giữa nhiều tác vụ có liên quan tới nhau, ta có thể cải thiện chất lượng mô hình trên tác vụ đích.

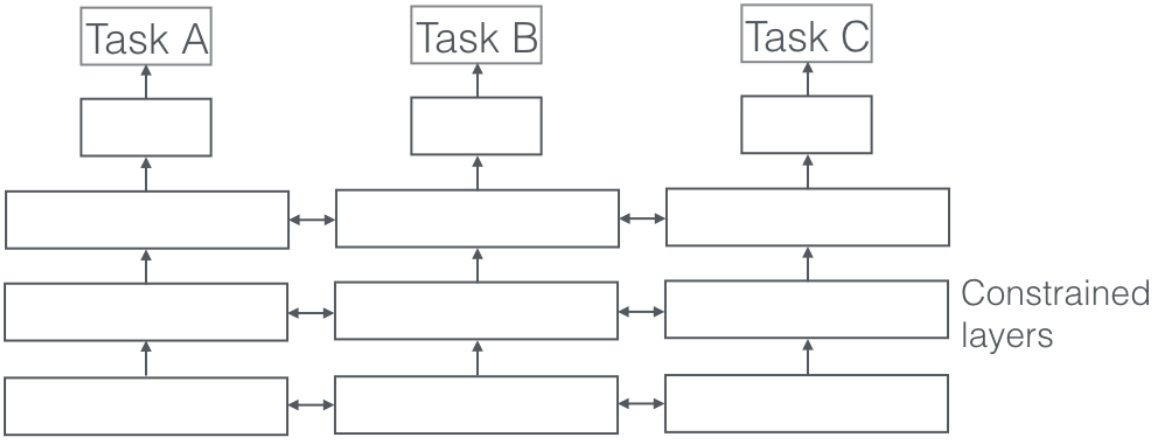
MTL dần được sử dụng rộng rãi trong học máy, từ xử lý ngôn ngữ tự nhiên, xử lý giọng nói hay thị giác máy tính. Một trong những ứng dụng đầu tiên của MTL trong xử lý ngôn ngữ tự nhiên có thể kể đến là xử lý đồng thời hai bài toán gán nhãn từ loại và phân tích cú pháp.

Hai mô hình MTL trong học sâu

Thông thường khi nhắc tới MTL, ta thường nghĩ ngay tới việc kết hợp giải quyết nhiều tác vụ trên cơ sở xây dựng mô hình học sâu để giải quyết điều này. Học sâu là khái niệm để chỉ việc sử dụng các mô hình mạng nơ-ron nhưng với kiến trúc và kích thước lớn, thể hiện ở việc mạng có rất nhiều tầng (sâu).



Hình 2.14: MTL chia sẻ trọng số cứng



Hình 2.15: MTL chia sẻ trọng số mềm

MTL trong học sâu được chia thành hai loại chính: chia sẻ tham số cứng (hard parameter sharing) và chia sẻ tham số mềm (soft parameter sharing) giữa các tầng ẩn.

Mô hình chia sẻ trọng số cứng được sủ dụng phổ biến hơn trong học sâu. Kiến trúc chung là các tầng ẩn được chia sẻ với toàn bộ các tác vụ và chỉ có một vài tầng cuối trong mạng là để học các đặc trưng riêng. Việc chia sẻ tầng ẩn này giúp cho mô hình học các đặc trưng chung phù hợp với tất cả các tác vụ và tránh đi được overfit trên tác vụ đích.

Với mô hình chia sẻ trọng số mềm thì ngược lại, mỗi tác vụ đều có một mô hình riêng, với bộ tham số riêng. Tuy nhiên các tham số này sẽ có ràng buộc với nhau, thông thường được chuẩn hóa cho tượng tự nhau.

Điểm mạnh của MTL

Điểm mạnh của MTL có thể kể đến:

Implicit data augmentation Sử dụng MTL giúp tăng kích thước tập dữ liệu sử dụng cho bài toán (sử dụng bộ dữ liệu của nhiều tác vụ trong huấn luyện mô hình). Mỗi bộ dữ liệu thông thường đều có nhiễu trong đó. Do

đó việc học riêng từng tác vụ rất dễ khiến mô hình bị overfit do học ra các đặc trưng lỗi này từ dữ liệu. Tuy nhiên việc sử dụng MTL sẽ giúp mô hình trú trọng hơn tới các đặc trưng quan trọng chung của các tập dữ liệu mà bỏ đi các đặc trưng lỗi riêng trong từng tập.

Attention focusing Đối với các tác vụ nhiều nhiễu hoặc với các dữ liệu với số chiều lớn, rất khó để mô hình có thể xác định được đặc trưng nào là quan trọng đối với mô hình. Khi sử dụng MTL, mô hình sẽ tập trung hơn tới các đặc trưng mà ảnh hưởng nhiều tới nhiều tác vụ khác nhau, đó cũng đồng thời là các đặc trưng quan trọng với mô hình.

Eavesdropping Một vài đặc trưng có thể dễ dàng học được với tác vụ A, tuy nhiên lại rất khó có thể tổng hợp được trong tác vụ B. Nguyên nhân là do B tương tác với những đặc trưng này theo cách phức tạp hơn, hoặc việc học các đặc trưng khác làm cản trở mô hình học được các đặc trưng này. Khi sử dụng MTL, mô hình có thể học được những đặc trưng này thông qua tác vụ A.

Representation bias MTL khiến cho các đặc trưng mà mô hình biểu diễn phải phù hợp với nhiều tác vụ khác nhau. Điều này giúp cho mô hình khái quát hóa các tác vụ mới trong tương lai. Mô hình đạt kết quả tốt trên một lượng lớn các tác vụ, cũng sẽ học tốt được các tác vụ khác, chỉ cần các tác vụ này cùng chung một miền bài toán.

Regularization MTL cũng có một phần chức năng bình thường hóa (regularization), khi mà trong số trong mỗi lần học được làm nhiễu bởi các tác vụ khác nhau, chính điều này cũng giúp mô hình tránh được overfit.

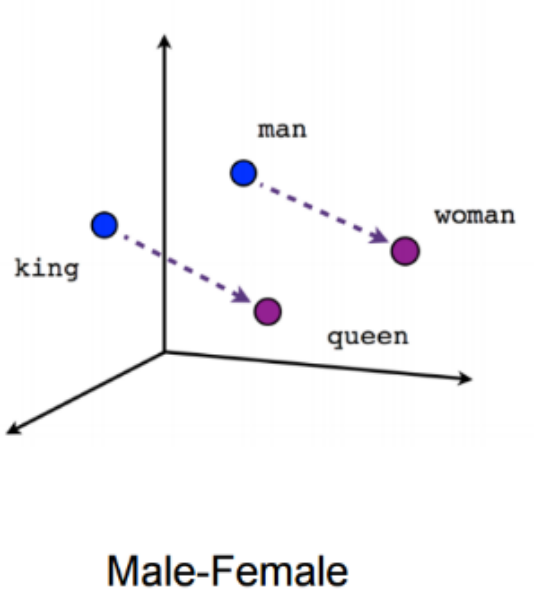
2.3 Vấn đề biểu diễn mức từ và mức âm tiết

2.3.1 Biểu diễn mức từ

Trong xây dựng mô hình học máy cho các bài toán xử lý ngôn ngữ tự nhiên, ta cần biểu diễn một cách phân biệt các từ đầu vào. Một cách biểu diễn đơn giản nhất là coi mỗi từ là một véc-tơ one hot (kích thước bằng kích thước từ điển từ và giá trị tại mọi chiều bằng 0, và giá trị tại chiều index của từ đó trong từ điển). Việc biểu diễn mỗi từ dưới dạng một one hot véc-tơ hoàn toàn giúp phân biệt được các từ với nhau, tuy nhiên các véc-tơ biểu diễn này lại rời rạc, thưa và có số chiều lớn. Việc sử dụng các véc-tơ này khiến cho mô hình trở nên nặng nề, đồng thời đặc trưng mà các véc-tơ này biểu diễn lại không mang ý nghĩa gì khác ngoài tính phân biệt.

Vector Space Models

Vector Space Models (VSMs) là mô hình nhằm biểu diễn từ dưới dạng một vector có giá trị liên tục, đồng thời các từ có tính tương đồng về mặt ngữ nghĩa sẽ gần nhau hơn trong không gian véc-tơ này. Tất cả các phương pháp trong VSMs đều dựa trên phân phối ngữ nghĩa (distributional semantic), với giả thiết rằng các từ xuất hiện trong cùng một văn phạm sẽ chia sẻ ngữ nghĩa với nhau.



Hình 2.16: Biểu diễn véc-tơ word2vec trong không gian

Xây dựng mô hình VSMs chia thành hai kiểu khác nhau: phương pháp dựa trên đếm số lượng (count-based method) và phương pháp dự đoán (predictive method) [10].

Phương pháp dựa trên đếm số lượng thống kê độ thường xuyên mà mỗi từ xuất hiện cùng với các từ lân cận nó trên một tập dữ liệu văn bản lớn, và mỗi từ sẽ được biểu diễn dưới dạng một véc-tơ nhiều chiều dựa trên thống kê này. Sau đó, thực hiện giảm chiều ít mất mát dữ liệu (sử dụng các phương pháp giảm chiều như Singular Vector Decomposition, Principal component analysis,...) để đưa véc-tơ thưa, nhiều chiều về thành một véc-tơ dày, ít chiều biểu diễn cho mỗi từ.

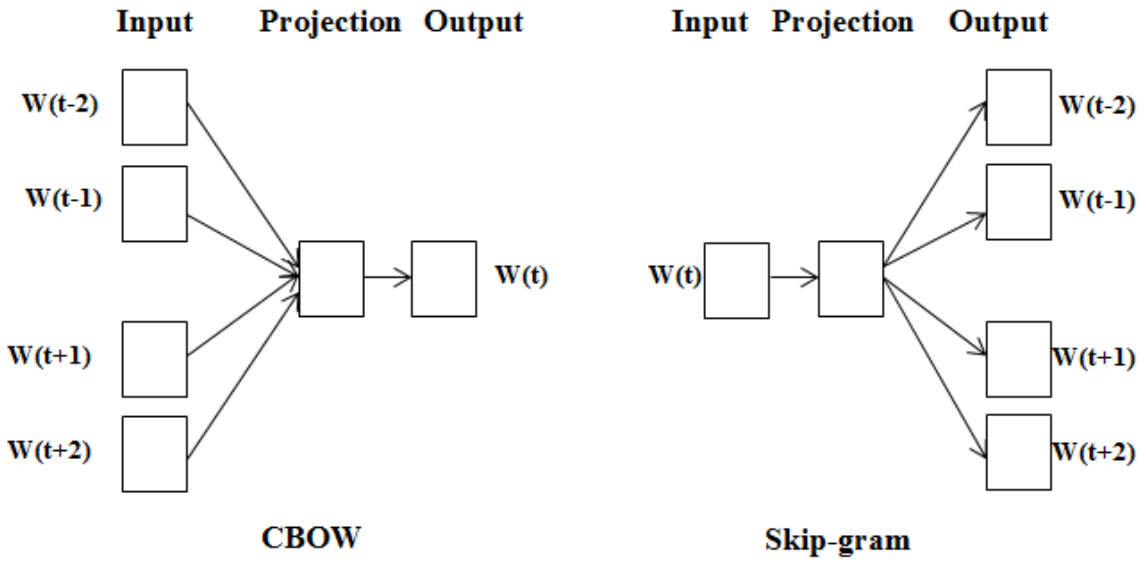
Với phương pháp dự đoán, mô hình thực hiện dự đoán các từ bên cạnh để học ra một véc-tơ viểu diễn dày và có số chiều nhỏ cho mỗi từ.

Word2vec

Word2vec là một mô hình cực kì hiệu quả trong số các phương pháp dự đoán. Véc-tơ biểu diễn bởi word2vec vừa có số chiều nhỏ vừa chứa các thông tin về mặt ngữ nghĩa như: từ đồng nghĩa (synonyms), từ trái nghĩa (antonyms) và từ tương tự (analogies). Word2vec được chia thành hai loại dựa trên hai phương pháp học biểu diễn từ khác nhau: *continuous bag-of-word (CBOW)* và *skip-gram*

Mô hình CBOW sử dụng một cửa sổ để xác định các từ xung quanh của một từ trung tâm, và thực hiện dự đoán từ trung tâm dựa vào các từ xung quanh trong cửa sổ đó. Một mô hình đơn giản được xây dựng như sau: mỗi từ trong từ điển sẽ được biểu diễn dưới dạng một véc-tơ one hot; sử dụng tập các véc-tơ biểu diễn của các từ xung quanh từ trung tâm làm đầu vào cho một mạng nơ-ron một tầng ẩn, sử dụng bộ phân loại softmax để dự đoán ra từ trung tâm; Sau khi huấn luyện mô hình, bộ trọng số của mô hình được sử dụng như các véc-tơ đặc trưng cho các từ.

Skip-gram là mô hình ngược lại của CBOW, thay vì việc sử dụng các từ



Hình 2.17: Mô hình CBOW và Skip-gram

trong cửa sổ từ để đoán từ trung tâm, ta sử dụng từ trung tâm để dự đoán xung quanh nó. Mô hình Skip-gram được đánh giá là chậm hơn so với CBOW nhưng lại cho độ chính xác cao hơn với các từ ít xuất hiện trong tập dữ liệu.

Fasttext[11] là mô hình học biểu diễn véc-tơ từ được công bố bởi Facebook AI Research, là mô hình cải tiến từ mô hình Skip-gram. Với cách biểu diễn véc-tơ đầu vào cho từng từ, mô hình Skip-gram hoàn toàn không quan tâm tới cấu trúc bên trong của từ. Để giải quyết vấn đề này, fasttext biểu diễn mỗi từ dưới dạng một túi từ n-gram kèm theo một phần tử đặc biệt là chính từ đó, ví dụ $where \rightarrow \{wh, whe, her, ere, re, where\}$. Mô hình huấn luyện tương tự như với skip-gram. Véc-tơ biểu diễn từ là tổng của các véc-tơ thành phần ($\{wh, whe, her, ere, re, where\}$)

2.3.2 Biểu diễn mức âm tiết

Biểu diễn mức từ đem lại nhiều lợi ích cho mô hình, khi nó mang nhiều ý nghĩa về mặt quan hệ giữa các từ với nhau. Tuy nhiên với bài toán tách từ, đầu vào là các âm tiết, việc biểu diễn mức âm tiết là rất khó khăn. Mỗi âm tiết đơn thuần mang rất nhiều ý nghĩa khác nhau, mà chỉ khi đặt vào từ cụ thể nó mới mang một ý nghĩa cụ thể. Một vài thử nghiệm trong biểu diễn mức âm tiết sẽ được đề cập trong chương 3.

Chương 3

Các phương pháp giải quyết bài toán

Bài toán tách từ nhân đầu vào là một câu và đầu ra yêu cầu là câu đã được phân tách các từ riêng biệt. Phương pháp giải quyết bài toán là coi câu đầu vào như một chuỗi âm tiết và ta sẽ đi giải quyết bài toán gán nhãn chuỗi. Có hai kiểu gán nhãn chuỗi thường được sử dụng cho tách từ: gán nhãn cho từng khoảng trắng trong câu (nhãn 1 thể hiện khoảng trắng là điểm phân tách từ và nhãn 0 thể hiện khoảng trắng không phải điểm phân tách) và gán nhãn cho các âm tiết trong câu (nhãn của mỗi âm tiết thể hiện đó là âm tiết đầu, giữa, hay cuối của từ). Trong khuôn khổ đồ án này, em sử dụng hướng tiếp cận là gán nhãn cho âm tiết. Mô tả cụ thể bài toán với cách tiếp cận này như sau: Nhận đầu vào là một chuỗi các âm tiết tương ứng với câu đầu vào, đầu ra yêu cầu là các nhãn tương ứng cho các âm tiết đó (B, I hoặc B, I, E tương ứng với đầu, giữa hay cuối từ).

Phương pháp giải quyết bài toán là xây dựng mô hình học sâu lấy trọng tâm là mô hình mạng Bi-LSTM. Xây dựng các kiến trúc mạng khác nhau nhằm thử nghiệm và tìm ra kiến trúc mạng phù hợp nhất cho bài toán. Các kiến trúc sự dụng sẽ được mô tả chi tiết trong chương này, các tham số thử nghiệm cho mỗi kiến trúc sẽ được đề cập trong chương 4.

3.1 Mô hình Bi-LSTM

Với cách tiếp cận coi bài toán tách từ như một bài toán gán nhãn chuỗi, mô hình Bi-LSTM là phù hợp để lựa chọn giải quyết bài toán. Một số vấn đề cần giải quyết trong xây dựng mô hình bao gồm: xác định biểu diễn đầu vào, lựa chọn bộ phân loại và thiết lập tham số (riêng thiết lập tham số sẽ được đề cập trong chương 4).

3.1.1 Biểu diễn mức âm tiết

Như đã đề cập, mỗi đầu vào của bài toán là một danh sách các âm tiết tương ứng với câu đầu vào cần thực hiện tách. Do đó ta cần biểu diễn mỗi âm tiết này dưới dạng một véc-tơ để làm đầu vào cho mô hình mạng nơ-ron phía sau. Thực hiện thử nghiệm với một vài phương pháp biểu diễn

Fasttext

Fasttext được sử dụng để xây dựng mô hình biểu diễn mức từ, tuy nhiên ta sẽ thử nghiệm huấn luyện mô hình fasttext trên một tập dữ liệu lớn để biểu diễn âm tiết.

Language model

Việc sử dụng mô hình fasttext, dự đoán các từ xung quanh dựa trên từ trung tâm có vẻ không phù hợp cho biểu diễn mức âm tiết. Một âm tiết có thể cấu thành nên rất nhiều từ, việc biểu diễn từ dựa trên nhiều từ xung quanh không mang lại nhiều ý nghĩa về ngữ nghĩa đối với biểu diễn của các âm tiết với nhau. Thay vào đó, ta sẽ đi huấn luyện một mô hình ngôn ngữ (language model), dự đoán từ tiếp theo dựa trên một vài từ phía trước. Mô hình ngôn ngữ được xây dựng là một mạng nơ-ron feed forward một tầng ẩn như trong mô hình word2vec. Mô hình được huấn luyện trên một tập dữ liệu lớn, trọng số của mô hình được lựa chọn làm biểu diễn của các âm tiết. Việc sử dụng mô hình này làm giảm thời gian huấn luyện, bỏ đi các quan hệ phức tạp không cần thiết với các âm tiết xung quanh và tập trung vào quan hệ duy nhất là các âm tiết này có thường xuyên đi liên tiếp nhau hay không. Hai âm tiết thường xuyên xuất hiện liên tiếp nhau xác suất rất cao có thể là một từ.

3.1.2 Biểu diễn mức ký tự

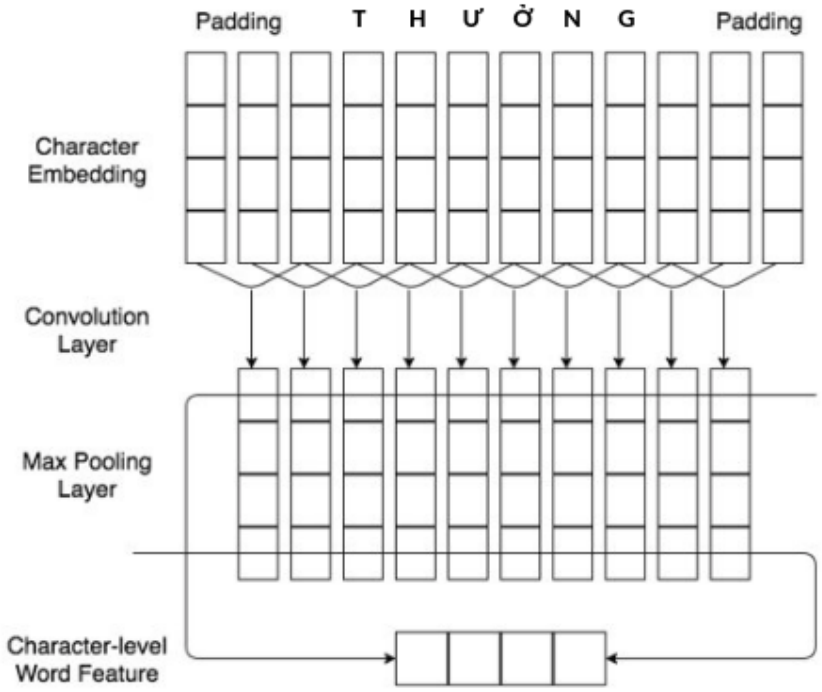
Thực hiện trích rút thông tin mức ký tự thông qua một mạng tích chập 1D (1D convoluional neural network). Đầu vào của mô hình là một ma trận được ghép bởi các véc-tơ biểu diễn mức ký tự. Ma trận này được coi như một ảnh và đưa qua một mạng 1D CNN với kiến trúc 1 tầng tích chập, 1 tầng ReLU và một tầng max pooling.

Mô hình CNN mạnh trong việc học đặc trưng theo vùng và phù hợp trong việc học đặc trưng giữa các ký tự (nhóm ký tự liên tiếp thành từng vùng) trong một âm tiết. Một đặc trưng dễ nhìn thấy nhất trong mức ký tự là ký tự viết hoa hoặc viết thường. Đặc trưng này đóng vai trò quan trọng trong việc gán nhãn cho âm tiết thuộc từ loại tên riêng và tiếng nước ngoài. Các véc-tơ biểu diễn mức ký tự sẽ không được huấn luyện trước trên tập dữ liệu lớn như đối với âm tiết mà sẽ học cùng với mô hình chính. lý do là số lượng ký tự là rất nhỏ, đồng thời thứ tự các ký tự trong trong âm tiết cũng là có quy tắc, do đó không cần thiết phải huấn luyện mô hình trước. Đầu ra của mạng CNN sẽ là vector biểu diễn mức ký tự.

3.1.3 Làm giàu đặc trưng đầu vào

Véc-tơ từ điển

Ngoài thông tin mức âm tiết và mức ký tự đều được trích xuất từ các mô hình học máy, ta sử dụng thêm chi thức bên ngoài thông qua việc sử dụng bộ từ điển. Từ điển thông thường được sử dụng để chuẩn hóa lại nhãn đầu ra (post process), tuy nhiên ta không muốn dùng cách này do việc chuẩn hóa đầu ra yêu cầu nhiều tinh chỉnh bằng tay hơn trước khi đưa ra nhãn cuối cùng, thay vào



Hình 3.1: Kiến trúc mạng 1d cnn biểu diễn mức ký tự

đó ta biểu diễn tri thức từ từ điển thành một đặc trưng đầu vào cho mô hình học, gọi là véc-tơ từ điển.

Véc-tơ từ điển của một âm tiết là một véc-tơ binary năm chiều mang giá trị 0 hoặc 1. Gọi s_i là âm tiết tại vị trí đang xét, khi đó mỗi chiều của véc-tơ từ điển thể hiện thể hiện:

- $s_{i-2}s_{i-1}s_i$ có xuất hiện trong từ điển hay không.
- $si-1s_i$ có xuất hiện trong từ điển hay không.
- $s_{i-1}s_is_{i+1}$ có xuất hiện trong từ điển hay không.
- s_is_{i+1} có xuất hiện trong từ điển hay không.
- $s_is_{i+1}s_{i+2}$ có xuất hiện trong từ điển hay không.

Nhóm âm tiết sử dụng so khớp có xuất hiện trong từ điển hay không đều được chuẩn hóa viết thường, đồng bộ với từ điển sử dụng.

Véc-tơ point-wise mutual information (PMI)

Việc sử dụng véc-tơ từ điển cần chi thức chuyên gia để có thể soạn được bộ từ điển phù hợp, đồng thời việc tạo từ điển có thể gọi là tạo đặc trưng bằng tay. Việc sử dụng một đặc trưng được trích rút bằng tay (handcraft features) như véc-tơ từ điển trong một mô hình gồm hoàn toàn các thành phần là các mô hình học tự động, khiến cho mô hình trở nên không đẹp mắt. Do đó, ta thực hiện xây dựng mô hình mô phỏng lại véc-tơ từ điển này dựa trên độ đo PMI. Giá trị PMI giữa âm tiết x, y thể hiện độ liên kết giữa hai âm tiết này với nhau.

$$pmi(x,y) = log\frac{p(x,y)}{p(x)p(y)}$$

$$pmi(x,y,z) = log\frac{p(x,y)p(x,z)p(y,z)}{p(x,y,z)p(x)p(y)p(z)}$$

Trong đó $p(x,y)$ là xác suất âm tiết x và y đi liên tiếp nhau trên tập dữ liệu, $p(x)$ và $p(y)$ là xác suất xuất hiện âm tiết x và y trong tập dữ liệu. Các giá trị *pmi* sẽ được tính toán trên một bộ dữ liệu lớn. Véc-tơ *pmi* của một âm tiết giống như có ý nghĩa giống như véc-tơ từ điển tuy nhiên mỗi chiều thay vì là có xuất hiện hay không xuất hiện trong từ điển sẽ thay bằng giá trị *pmi* của bộ các âm tiết đó.

3.1.4 Mô hình chính và bộ phân loại

Phần lõi chính của mô hình là mạng Bi-LSTM, học chuỗi đầu vào theo cả hai chiều (cùng chiều câu đầu vào và ngược chiều câu đầu vào). Đầu vào mô hình sẽ là kết hợp của các véc-tơ đặc trưng đã trích xuất ở các bước trước: đặc trưng mức âm tiết, đặc trưng mức ký tự, véc-tơ từ điển, véc-tơ pmi. Đầu ra mô hình được đưa qua một bộ phân loại để dự đoán nhãn cho từng âm tiết. Hai bộ phân loại được sử dụng gồm:

Softmax hay còn gọi là hàm trung bình mũ, thực hiện chuẩn hóa đầu ra về miền 0..1. Công thức của hàm softmax như sau (công thức áp dụng lên mạng nơ-ron):

$$a_i = \frac{e^{z_i}}{\sum j{=}1^{10} z_j} \ \forall i \in 1, 2, ...C$$

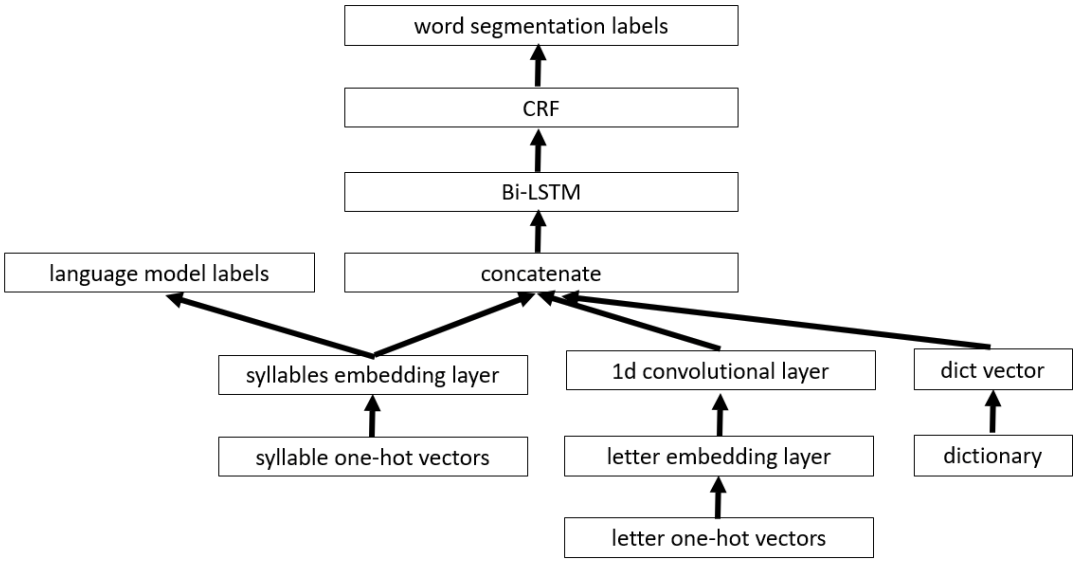
Trong đó z_i là giá trị tại nơ-ron và a_i là giá trị của nơ-ron sau khi kích hoạt (coi softmax như một hàm kích hoạt). Một tầng dense sử dụng hàm kích hoạt là *softmax* được gọi là tầng *softmax*

Tầng *CRF* sử dụng mô hình CRF như một bộ phân loại cho mô hình mạng nơ-ron chính. Đầu vào (X) của mô hình *CRF* là chuỗi véc-tơ trạng thái ẩn đầu ra của mạng *BiLSTM*. Mô hình *CRF* ước lượng tham số nhằm xác định xác suất có điều kiện của các nhãn khi biết đầu vào và các nhãn xung quanh, nhằm xác định nhãn phù hợp nhất với mỗi đầu vào. Do đó, mô hình *CRF* được sử dụng và gọi là tầng *CRF*, giúp bổ sung thông tin về quan hệ giữa các nhãn cho mô hình.

3.2 Mô hình Multi-tasks

Mô hình thử nghiệm cuối cùng là mô hình học đồng thời nhiều tác vụ, và cụ thể ở đây là hai tác vụ: mô hình ngôn ngữ và tách từ. Việc huấn luyện mô hình ngôn ngữ trong việc sinh biểu diễn mức âm tiết có tác dụng tốt tới mô hình. Do đó, ta thử nghiệm xây dựng một mô hình huấn luyện đồng thời hai tác vụ này với nhau thay vì huấn luyện hai mô hình riêng biệt cho hai tác vụ.

Có thể thấy trong hình 3.3, mô hình MTL sử dụng là mô hình chia sẻ trọng số cứng. Phần tham số chia sẻ giữa hai tác vụ là tầng nhúng mức âm tiết với ma trận biểu diễn mức âm tiết đồng thời là trọng số thuộc tầng ẩn trong bài toán xây dựng mô hình ngôn ngữ, đồng thời lại là tầng nhúng mức âm tiết trong bài toán tách từ. Điều này dễ dàng có thể nhận ra được khi mà ta tách biệt hai mô



Hình 3.2: Kiến trúc mô hình Multi-tasks

hình của hai tác vụ này. Với mỗi âm tiết, đầu ra của bài toán xây dựng mô hình ngôn ngữ là âm tiết tiếp theo trong câu, trong khi đầu ra của bài toán tách từ sẽ là nhãn âm tiết bắt đầu, giữa, hay cuối của từ.

3.3 Các phương pháp tiếp cận nổi trội đã được công bố

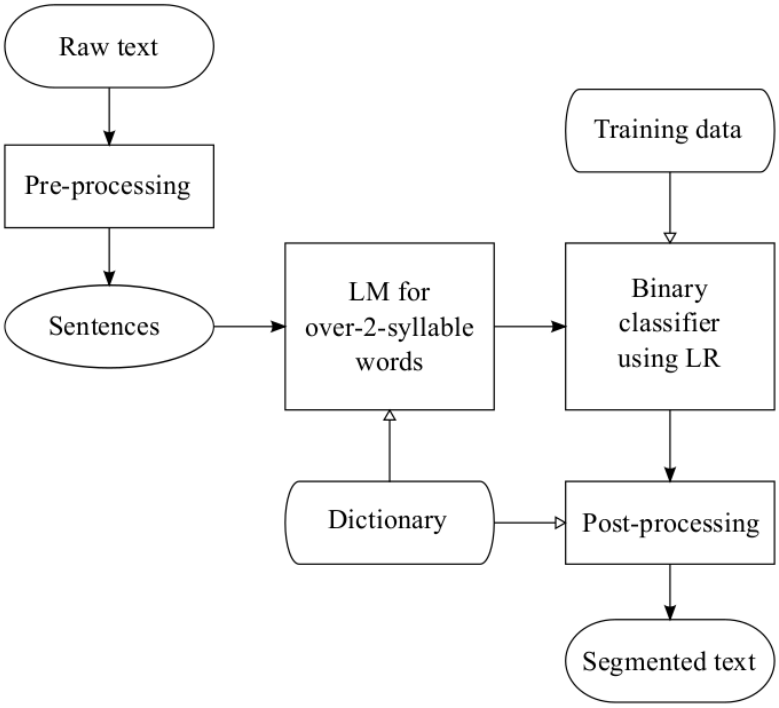
3.3.1 UETSegmenter

UETSegmenter[12] là bộ tách từ được công bố năm 2016, và tới thời điểm hiện tại, UETSegmenter là một trong hai bộ tách từ có độ chính xác cao nhất cho bài toán tách từ cho tiếng Việt. [1] Bộ tách từ này tiếp cận bài toán theo cách gán nhãn cho khoảng trắng. Kiến trúc tổng thể của hệ thống được mô tả trong hình 3.4.

Dữ liệu văn bản đầu vào được chuẩn hóa và tách thành các câu làm dữ liệu đầu vào cho mô hình. Do trong tiếng Việt, số lượng từ có từ 3 âm tiết trở nên không nhiều nên các từ này sẽ được gán nhãn trước sử dụng phương pháp so khớp dài nhất (*longest matching*) trên một bộ từ điển được chuẩn bị trước. Dữ liệu sau đó được đưa qua một bộ phân loại, thực hiện gán nhãn cho từng khoảng trắng. Bộ phân loại mà mô hình lựa chọn là *logistic regression* chọn 0.5 là ngưỡng để quyết định phân loại. Câu sau khi được gán nhãn được đưa qua bộ chuẩn hóa đầu ra. Quá trình chuẩn hóa đầu ra thực hiện chuẩn hóa lại nhãn cho nhãn có độ tự tin nhỏ hơn một giá trị ngưỡng cho trước. Quá trình chuẩn hóa dựa trên bốn luật áp dụng lên các từ xung quanh khoảng trắng đang xét:

- Nếu từ $s_{i-1}s_i$ xuất hiện trong từ điển nhưng từ s_is_{i+1} không xuất hiện

[1] RDRSegmenter đạt state-of-the-art trên hai độ đo *recall* và $f1$, trong khi đó UETSegmenter vẫn đạt state-of-the-art trên độ đo *precision*



Hình 3.3: Kiến trúc mô hình của UETSegmenter

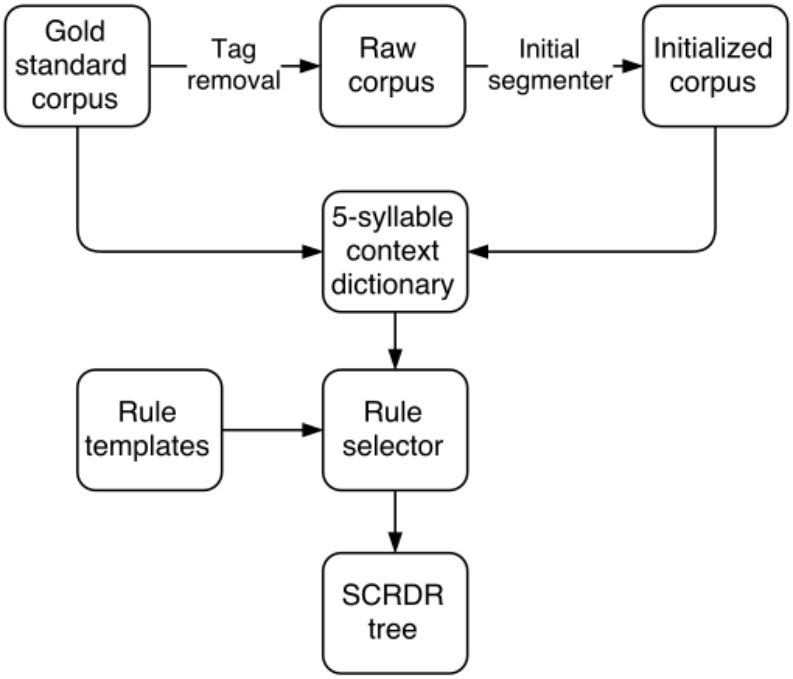
trong từ điển thì không thay đổi nhãn

- Nếu từ $s_i s_{i+1}$ xuất hiện trong từ điển nhưng từ $s_{i-1} s_i$ không xuất hiện trong từ điển thì không thay đổi nhãn
- Nếu cả hai từ đều không có trong từ điển thì gán lại nhãn
- Nếu cả hai từ đều có trong từ điển thì giữ lại khoảng trắng có độ tin cậy cao hơn

3.3.2 RDRSegmenter

RDRSegmenter là bộ tách từ mới nhất cho văn bản tiếng Việt [13]. RDRSegmenter tự động xây dựng cây SCRDR (*Single Classification Ripple Down Rules tree*) nhằm chuẩn hóa lại các kết quả tách từ sai từ quá trình xử lý so khớp dài nhất trước đó.

Tập huấn luyện được chia thành hai loại: tập huấn luyện gán nhãn chuẩn (gold standard corpus) và tập huấn luyện không nhãn (raw corpus). Quá trình huấn luyện bắt đầu bằng việc sinh ra tập các biểu diễn tách từ khởi tạo (initialized corpus) thông qua một bộ tách từ khởi tạo (initialized segmenter), trong đó bộ tách từ khởi tạo được xây dựng dựa trên phương pháp so khớp dài nhất trên một tập từ điển được xây dựng trước. Tiếp theo, một cửa sổ kích thước 5 âm tiết di chuyển từ trái sang phải câu, nhằm lấy ra các bộ 5 âm tiết liên tiếp kèm theo nhãn khởi tạo. Các bộ 5 âm tiết này được đưa qua cây SCRDR nhằm xây dựng cây dựa trên việc chỉnh lại các nhãn khởi tạo dựa trên tập huấn luyện gán nhãn chuẩn. Với cây SCRDR đã được huấn luyện, biểu diễn tách từ khởi tạo được tách thành các bộ 5 âm tiết liên tiếp làm đầu vào cho cây SCRDR để đưa



Hình 3.4: Kiến trúc mô hình của RDRSegmenter

ra nhãn cuối cùng cho từng bộ năm âm tiết này, sau đó kết hợp lại để làm kết quả cuối cùng của cả câu đầu vào.

Chương 4

Thử nghiệm và đánh giá

4.1 Dữ liệu và phương pháp đánh giá

4.1.1 Tập dữ liệu

Các thử nghiệm được huấn luyện và đánh giá trên bộ dữ liệu tách từ được công bố năm 2013 bởi dự án quốc gia VLSP trong lĩnh vực xử lý ngôn ngữ và giọng nói tiếng Việt. Bộ dữ liệu gốc gồm 75000 câu (độ dài trung bình mỗi câu là 23 từ). Tuy nhiên sử dụng để huấn luyện và đánh giá mô hình trong đề tài này, ta sử dụng hai tập dữ liệu được chia như sau:

Tập dữ liệu 48k gồm gần 48000 câu (chính xác là 47936 câu) được lấy từ bộ dữ liệu VLSP gốc sau khi đã loại bỏ đi các phần header và chỉ giữ lại phần nội dung của mỗi văn bản. Tập dữ liệu này được chia thành 3 tập train, dev, test với kích thước tương ứng là $33060, 7367, 7509$ câu.

Tập dữ liệu thứ hai sử dụng toàn bộ tập VLSP gồm 75000 câu làm tập train. Tập test gồm 2120 câu (độ dài trung bình là 31 từ một câu) được lấy từ 10 files *800001.seg* tới *800010.seg* từ tập test cho bài toán gán nhãn từ loại được công bố trong hội thảo VLSP năm 2013.

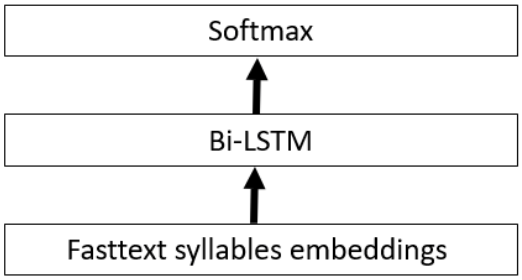
Tập dữ liệu sử dụng cho huấn luyện biểu diễn mức âm tiết được thu thập trên tất cả các chủ đề từ nguồn báo *zing news*. Tập dữ liệu gồm khoảng 8,7 triệu câu, mỗi câu có độ dài trung bình là 34 âm tiết.

Từ điển sử dụng trong làm giàu đặc trưng đầu vào được tổng hợp chính từ hai bộ từ điển *vietnamese lexicon* [14] và *vietnamese wordlist* [15]. Đồng thời từ điển cũng được bổ sung thêm các tên riêng bao gồm tên quốc gia, tên công ty và tên địa danh. Tổng kích thước từ điển gồm 75827 từ.

4.1.2 Phương pháp đánh giá

Sử dụng độ đo *precision*, *recall* và *f1 score* để đánh giá mô hình. Trong đó, các độ đo được tính như sau:

- Precision được tính bằng số lượng từ gán nhãn đúng trên số lượng từ mô hình dự đoán ra
- Recall được tính bằng số lượng từ gán nhãn đúng trên số lượng từ thực tế có trong câu



Hình 4.1: Kiến trúc mô hình thử nghiệm 1

- F1 score được tính theo công thức:

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

4.2 Mô hình thử nghiệm và thiết lập tham số

4.2.1 Thử nghiệm 1 (BiLSTM + softmax)

Thử nghiệm đầu tiên đươc thiết lập sử dụng duy nhất *fasttext* véc-tơ làm đặc trưng đầu vào. Sử dụng thư viện *gensim* để huấn luyện véc-tơ *fasttext* với tham số thiết lập như sau:

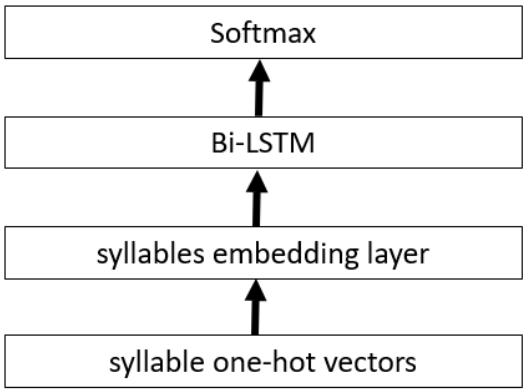
Tần suất xuất hiện tối thiểu	5
Kích thước véc-tơ fasttext	100
Kích thước cửa sổ	10
Số bước lặp	20

Bảng 4.1: Tham số huấn luyện fasttext

Mô hình chính sử dụng là *Bi-LSTM* kèm theo một bộ phân loại *softmax* tại tầng cuối cùng. *Dropout* được sử dụng trên toàn bộ mô hình nhằm tránh hiện tượng *overfit*. Thực hiện thay đổi các tham số mô hình: kích thước véc-tơ trạng thái ẩn (*hidden state size*), số lượng tầng ẩn và tỉ lệ dropout (*dropout rate*) trong quá trình huấn luyện. Tham số mô hình cố định bao gồm: số lượng âm tiết tối đa trong câu (nếu câu đầu vào có độ dài lớn hơn giá trị tối đa, thực hiện cắt đi phần dư thừa), giá trị khởi tạo cho các trọng số (khởi tạo ngẫu nhiên theo phân phối chuẩn). Tham số huấn luyện bao gồm *learning rate*, hàm mất mát *cross entropy* và tối ưu tham số *Adam*. Tất cả thử nghiệm được đề cập trong chương này đều sử dụng thư viện *tensorflow* để xây dựng kiến trúc mạng.

Huấn luyện và đánh giá mô hình trên tập 48k với 2 nhãn B-I thu được kết quả tốt nhất với thiết lập tham số như sau:

Mô hình:



Hình 4.2: Kiến trúc mô hình thử nghiệm 2

Kích thước véc-tơ trạng thái ẩn	200
Số lượng âm tiết tối đa trong câu	250
Số lượng tầng BiLSTM	2
Learning rate	0.01
tỉ lệ dropout	0.5

Bảng 4.2: Tham số mô hình thử nghiệm 1

Kết quả:

precision	recall	f1 score
96.07	96.88	96.48

Bảng 4.3: Kết quả thử nghiệm 1

Thử nghiệm 1 đạt kết quả *f1 score* là 96.48, kết quả khá cao chỉ với mô hình thiết lập đơn giản.

4.2.2 Thử nghiệm 2 (Syllable Embedding Layer + BiLSTM + softmax)

Hướng tiếp cận của mô hình tương tự như mô hình trong mục 4.2.1, tuy nhiên véc-tơ biểu diễn mức âm tiết sẽ được huấn luyện cùng mô hình thông qua một tầng nhúng. Việc học véc-tơ biểu diễn mức âm tiết cùng với quá trình huấn luyện mô hình giúp véc-tơ được biểu diễn sẽ phù hợp hơn với bài toán đích, tuy nhiên lại dễ trở nên overfit với dữ liệu. Tuy nhiên do véc-tơ biểu diễn mức âm tiết thông qua fasttext không biểu diễn được đầy đủ ý nghĩa của âm tiết như đã phân tích, do đó thử nghiệm này vẫn có thể thu được một kết quả tốt. Đầu vào của mô hình là véc-tơ one hot có kích thước bằng kích thước từ điển từ âm tiết. Từ điển âm tiết được tổng hợp trên tập huấn luyện, gồm các âm tiết có số lần xuất hiện lớn hơn hoặc bằng một ngưỡng (min occurences), các từ có số lần xuất hiện ít hơn ngưỡng hoặc không xuất hiện đều được gán là âm tiết <oov> (out-of-vocabulary).

Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với hai nhãn B-I thu được kết quả tốt nhất với thiết lập tham số như sau:

Mô hình:

Kích thước từ điển âm tiết	6280
Ngưỡng xuất hiện ít nhất của âm tiết	4
Kích thước véc-tơ biểu diễn mức âm tiết	100
Kích thước trạng thái ẩn	200
Số lượng âm tiết tối đa trong câu	250
Số tầng BiLSTM	2
Learning rate	0.01
tỉ lệ dropout	0.8

Bảng 4.4: Tham số mô hình thử nghiệm 2

Kết quả:

precision	recall	f1 score
96.54	96.83	96.68

Bảng 4.5: Kết quả thử nghiệm 2

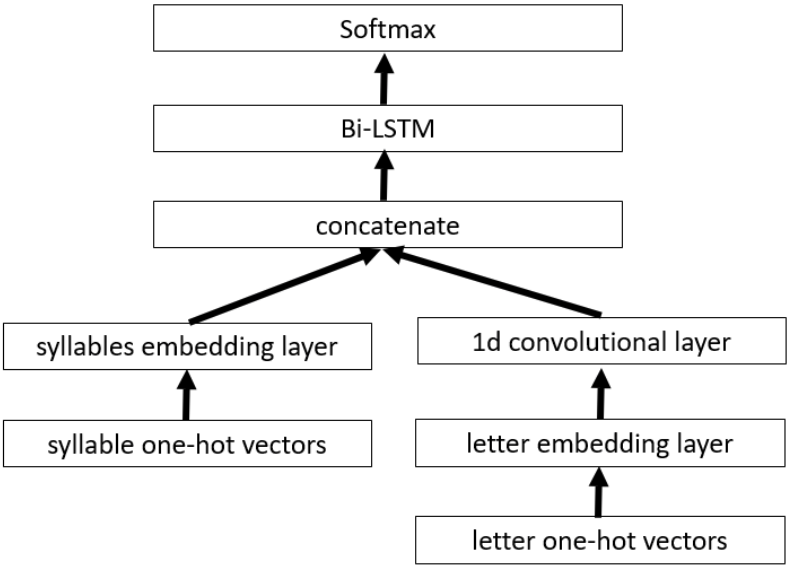
Thử nghiệm 2 đạt kết quả *f1 score* là 96.68 cao hơn so với thử nghiệm 1 là 96.48. Việc học biểu diễn mức âm tiết cùng với quá trình huấn luyện mô hình mang lại kết quả tốt hơn so với huấn luyện trước sử dụng *fasttext*.

4.2.3 Thử nghiệm 3 (Syllable Embedding Layer + Character Embedding Layer + CharCNN + BiLSTM + softmax)

Nhận thấy ký tự đóng vai trò quan trọng trong việc xác định nhãn trong bài toán tách từ, đặc biệt là vấn đề một âm tiết là viết hoa hay viết thường sẽ quyết định rất nhiều đến nhãn của âm tiết đó. Do đó ta cải tiến mô hình trong thử nghiệm 2 bằng cách bổ sung đặc trưng mức ký tự vào mô hình. Véc-tơ biểu diễn mức ký tự được học thông qua một tầng nhúng trước khi đưa vào một mạng *CNN 1D* để học ra véc-tơ đặc trưng mức ký tự. Véc-tơ này sau đó được ghép nối với véc-tơ biểu diễn mức âm tiết làm đầu vào cho mô hình *BiLSTM*.

Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với hai nhãn B-I thu được kết quả tốt nhất với thiết lập tham số như sau:

Mô hình:



Hình 4.3: Kiến trúc mô hình thử nghiệm 3

Kích thước từ điển âm tiết	6280
Ngưỡng xuất hiện ít nhất của âm tiết	4
Kích thước véc-tơ biểu diễn mức âm tiết	100
Kích thước từ điển ký tự	199
Kích thước véc-tơ biểu diễn mức ký tự	100
Kích thước trạng thái ẩn	200
Số lượng âm tiết tối đa trong câu	250
Số lượng ký tự tối đa trong từ	10
Số tầng BiLSTM	2
Kích thước các filters	1, 2, 3
Số lượng các filters	30, 30, 40
Learning rate	0.01
tỉ lệ dropout	0.5

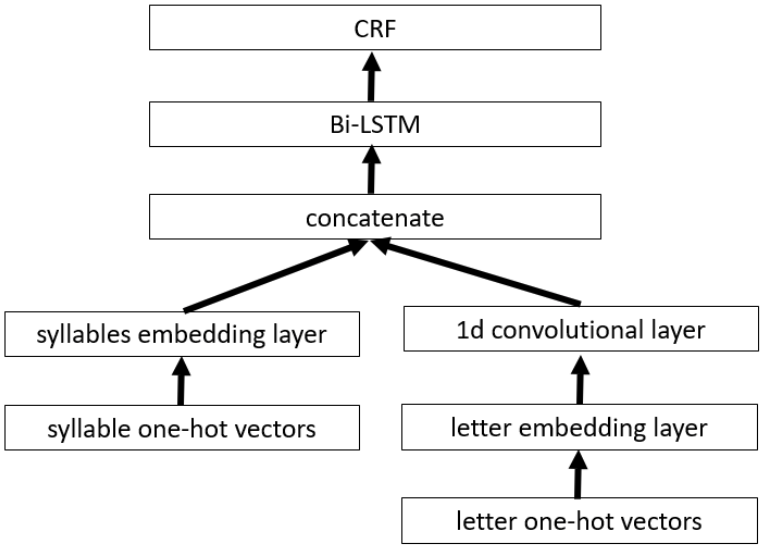
Bảng 4.6: Tham số mô hình thử nghiệm 3

Kết quả:

precision	recall	f1 score
96.95	97.32	97.13

Bảng 4.7: Kết quả thử nghiệm 3

Sau khi đánh giá kết quả trên tập test, nhận thấy mô hình vẫn có nhiều nhầm lẫn (1401 từ gán nhãn sai thuộc loại này trong tổng số 4514 từ gán nhãn sai) đối với các từ đã xuất hiện trong tập train nhưng trong tập test lại được viết dưới dạng khác (viết hoa hoặc viết thường), do đó, thực hiện thử nghiệm với việc đưa toàn bộ dữ liệu về cùng một dạng (viết thường) nhưng trên một



Hình 4.4: Kiến trúc mô hình thử nghiệm 4

nửa dữ liệu đầu vào (trên đầu vào ký tự hoặc đầu vào âm tiết). Mục đích nhằm giúp mô hình biết 2 cách viết khác nhau này vẫn là của một từ, nhưng một phần của mô hình vẫn học được phần khác nhau của hai từ này. Thử nghiệm với bộ tham số tương tự như trên, tuy nhiên kết quả thu được không khả quan:

	precision	recall	f1 score
lower character	96.57	97.20	96.89
lower word	96.74	97.42	97.08

Bảng 4.8: Kết quả thử nghiệm 3 với thay đổi viết hoa viết thường

Kết quả trong thử nghiệm này đạt *f1 score* là 97.13 tiếp tục tăng so với hai thử nghiệm trước đó. Việc đưa thông tin mức âm tiết vào mô hình cho kết quả khả quan. Hai thử nghiệm đơn giản nhằm giải quyết lỗi nhập nhằng viết hoa viết thường lại chưa đem lại kết quả cao như mong đợi khi kết quả thu được thấp hơn (97.08) trong khi lỗi sai do nhập nhằng viết hoa viết thường gần như không giảm (1397 từ gán nhãn sai)

4.2.4 Thử nghiệm 4 (Syllable Embedding Layer + Character Embedding Layer + CharCNN + BiLSTM + CRF)

Cải tiến mô hình trong thử nghiệm 3 bằng cách thay bộ phân loại *softmax* bằng *CRF*, mục tiêu nhằm giúp mô hình học được các đặc trưng quan hệ trên nhãn trong quá trình huấn luyện.

Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với hai nhãn B-I với thiết lập tham số tương tự như trong thử nghiệm 3, thu được kết quả như sau:

precision	recall	f1 score
96.84	97.26	97.05

Bảng 4.9: Kết quả thử nghiệm 4 trên nhãn B-I

Kết quả thu được không tốt hơn so với thử nghiệm 2. Đánh giá ban đầu là do việc sử dụng tập dữ liệu với hai nhãn B-I thể hiện rất ít các quan hệ trên nhãn mà tầng *CRF* có thể học được (ngoài nhãn I không thể ở đầu câu thì mọi sắp xếp của một chuỗi nhãn B-I đều có thể xảy ra trong một câu đầu vào). Để sử dụng được nhiều quan hệ trên nhãn hơn, ta sửa lại dữ liệu thành ba nhãn B-I-E (đầu, giữa, cuối từ), với những từ có hai âm tiết nhãn sẽ là (B-E). Thống kê trên tập dữ liệu huấn luyện, các từ có nhiều hơn 2 âm tiết là 24058 từ (chiếm 2.24%), do đó việc biểu diễn quan hệ trên nhãn được kỳ vọng sẽ mang lại hiệu quả nhỏ cho mô hình. Thử nghiệm với mô hình có thiết lập tham số tương tự như trong thử nghiệm 3, trên tập dữ liệu 48k với 3 nhãn B-I-E, ta thu được kết quả:

	precision	recall	f1 score
Nhãn B-I	96.84	97.26	97.05
Nhãn B-I-E	97.11	97.47	97.29

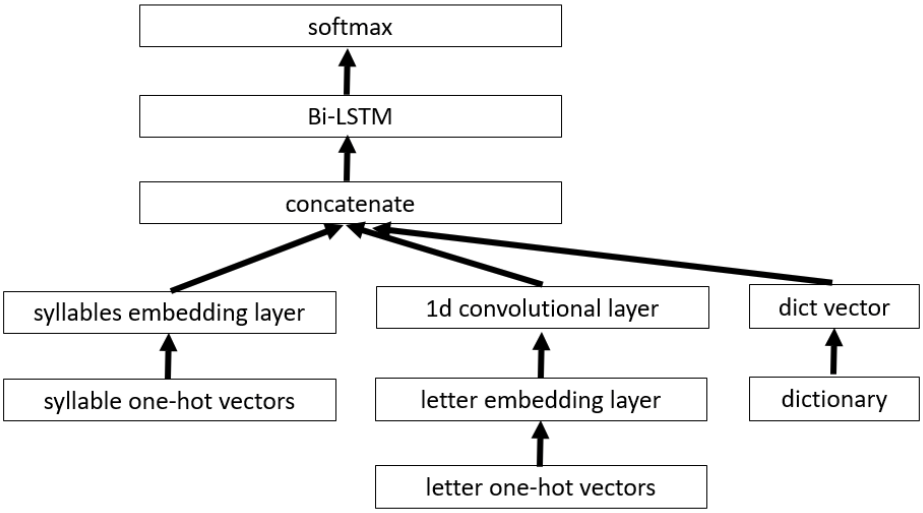
Bảng 4.10: Kết quả thử nghiệm 4 trên nhãn B-I-E

Kết quả thử nghiệm đạt *f1 score* là 97.29 tăng nhẹ so với thử nghiệm trước đó là 97.13. Việc sử dụng *CRF* trên dữ liệu 3 nhãn B-I-E đem lại hiệu quả đúng như mong đợi.

4.2.5 Thử nghiệm 5 (Syllable Embedding Layer + Character Embedding Layer + CharCNN + Dict-vector + BiLSTM + softmax)

Việc sử dụng hoàn toàn mô hình deep learning làm mô hình cho bài toán bước đầu mang lại những kết quả khả quan, tuy nhiên độ chính xác của mô hình lại chưa đủ cạnh tranh so với các mô hình có sử dụng từ điển hay các các đặc trưng được trích rút bằng tay khác. Việc đưa các thông tin bên ngoài vào mô hình thường được làm theo hai cách: biểu diễn thành đặc trưng làm đầu vào cho mô hình hoặc tinh chỉnh lại nhãn đầu ra của mô hình (*post processing*). Thử nghiệm tiếp theo đưa thông tin từ điển vào mô hình thông qua việc biểu diễn thành một véc-tơ đầu vào cho mô hình chính (véc-tơ từ điển - Dict-vector). Thử nghiệm này cải tiến từ mô hình 3 bằng cách sử dụng thêm véc-tơ từ điển làm đầu vào cho mô hình chính.

Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với hai nhãn B-I với thiết lập tham số tương tự như trong thử nghiệm 4, thu được kết quả như sau:



Hình 4.5: Kiến trúc mô hình thử nghiệm 5

precision	recall	f1 score
97.93	98.39	98.16

Bảng 4.11: Kết quả thử nghiệm 5

Kết quả thử nghiệm đạt 98.16 tăng rất nhiều so với những thử nghiệm trước đó (tốt nhất trước đó là 97.29). Việc sử dụng véc-tơ từ điển đem lại kết quả ngoài mong đợi khi chỉ sử dụng một véc-tơ binary đơn giản lại đem lại hiệu quả cao.

4.2.6 Thử nghiệm 6 (Syllable Embedding Layer + Character Embedding Layer + CharCNN + Dict-vector + BiLSTM + CRF)

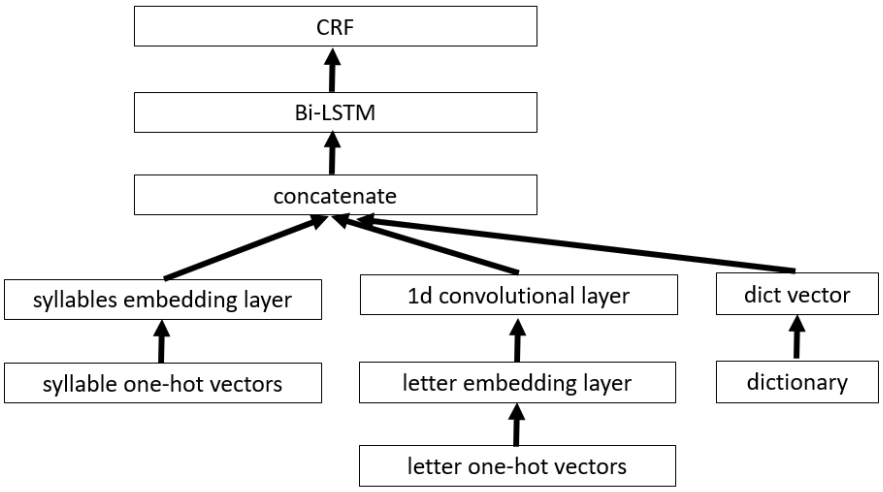
Tiếp tục cải tiến mô hình trong thử nghiệm 5 bằng cách thay bộ phân loại *softmax* bằng *CRF*.

Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với ba nhãn B-I-E với thiết lập tham số tương tự như trong thử nghiệm 5, thu được kết quả cao hơn như sau:

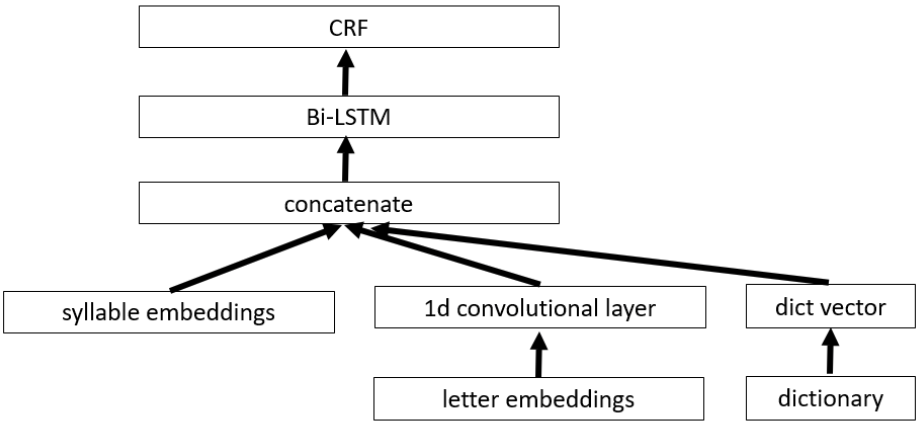
precision	recall	f1 score
98.07	98.40	98.23

Bảng 4.12: Kết quả thử nghiệm 6

Kết quả thử nghiệm đạt 98.23 tăng nhẹ so với kết quả trước đó là 98.16. Việc sử dụng *CRF* giúp tăng nhẹ kết quả tương tự như với kết quả thay đổi trong thử nghiệm 3 và 4.



Hình 4.6: Kiến trúc mô hình thử nghiệm 6



Hình 4.7: Kiến trúc mô hình thử nghiệm 7

4.2.7 Thử nghiệm 7 (Character Embedding Layer + Char-CNN + fasttext/LM vector + Dict-vector + BiLSTM + CRF)

Đánh giá việc sử dụng véc-tơ biểu diễn mức âm tiết thông qua fasttext mang lại nhiều thông tin không cần thiết và phù hợp với biểu diễn mức âm tiết, do đó ta sử dụng một dạng biểu diễn đơn giản hơn thông qua mô hình ngôn ngữ (3.1.1). Véc-tơ biểu diễn mức âm tiết thông qua mô hình ngôn ngữ (véc-tơ LM) được huấn luyện trên bộ dữ liệu *zing news*. Trong đó tham số của huấn luyện mô hình ngôn ngữ sử dụng như sau:

Tần suất xuất hiện tối thiểu	10
Kích thước véc-tơ lm	3
Kích thước cửa sổ	10
Số bước lặp	10

Bảng 4.13: Tham số mô huấn luyện mô hình ngôn ngữ

Đồng thời, ta cũng huấn luyện một mô hình sử dụng véc-tơ biểu diễn mức âm tiết *fasttext* để so sánh mức độ hiệu quả của véc-tơ LM mới được đưa vào mô hình. Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với ba nhãn B-I-E với thiết lập tham số tương tự như trong thử nghiệm 6, thu được kết quả như sau:

	precision	recall	f1 score
pretrain fasttext	98.10	98.52	98.31
pretrain language model	98.25	98.56	98.41

Bảng 4.14: Kết quả thử nghiệm 7

Việc sử dụng véc-tơ biểu diễn âm tiết thông qua *fasttext* và *mô hình ngôn ngữ* đều cho kết quả tốt hơn khi kết hợp với véc-tơ từ điển. Đồng thời sử dụng véc-tơ LM cho kết quả cao hơn 0.1% so với sử dụng véc-tơ fasttext (98.41 so với 98.31).

4.2.8 Thử nghiệm 8 (Character Embedding Layer + Char-CNN + LM vector + pmi-vector + BiLSTM + CRF)

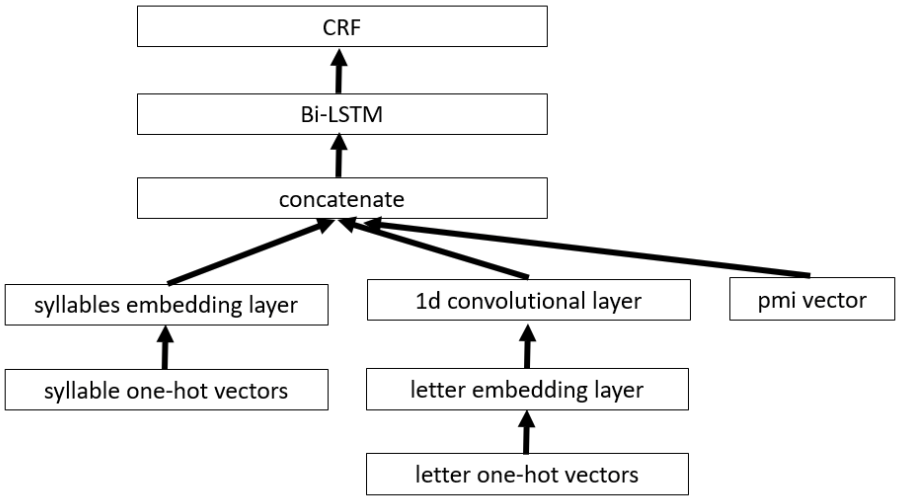
Các thử nghiệm đã thể hiện việc sử dụng véc-tơ từ điển cho kết quả tốt hơn rất nhiều. Tuy nhiên việc sử dụng véc-tơ từ điển cần phải xây dựng trước một bộ từ điển và trích rút đặc trưng bằng tay đơn thuần (handcraft features). Do đó ta nghĩ đến việc mô phỏng lại véc-tơ từ điển bằng một véc-tơ tương tự và đây là véc-tơ pmi. Các giá trị pmi giữa các âm tiết được xây dựng bằng quá trình thống kê trên tập *zing news*. Véc-tơ pmi thay thế cho véc-tơ từ điển, được ghép nối với véc-tơ biểu diễn mức âm tiết và mức ký tự làm đầu vào cho mô hình Bi-LSTM.

Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với ba nhãn B-I-E với thiết lập tham số tương tự như trong thử nghiệm 7, thu được kết quả như sau:

precision	recall	f1 score
95.66	96.90	96.28

Bảng 4.15: Kết quả thử nghiệm 8

Kết quả thu được thấp hơn nhiều so với các kết quả thử nghiệm trước đó. Giá trị pmi học được từ tập dữ liệu lớn không có khả năng phân biệt rõ ràng kết hợp âm tiết nào sẽ tạo ra từ. Các từ xuất hiện nhiều trong tập dữ liệu huấn luyện và ít nhập nhằng như *Sài Gòn*, *niềm vui*, *háo hức*,... đều có pmi cao (> 7) và ít nhập nhằng với các âm tiết xung quanh. Tuy nhiên do đặc trưng của tập



Hình 4.8: Kiến trúc mô hình thử nghiệm 8

huấn luyện, các kết hợp có pmi nhỏ hơn không giúp mô hình phân biệt đâu là kết hợp tạo từ đâu không phải, đồng thời gây nhập nhằng cho mô hình chính. Ví dụ: *pmi(cắm cờ)=6.21*, *pmi(mang tên)=4.69*, trong khi các từ trong từ điển tiếng Việt như *pmi(bộ đội)=2.11*, *pmi(nước mắt)=3.57*,... lại có giá trị pmi thấp hơn. Thử nghiệm với véc-tơ pmi không mang lại kết quả tốt, cần thiết phải sử dụng bộ dữ liệu huấn luyện véc-tơ pmi đa dạng hơn.

4.2.9 Thử nghiệm 9 (Mô hình multi-tasks)

Với việc mô hình sử dụng biểu diễn mức âm tiết bằng mô hình ngôn ngữ cho kết quả tốt nhất trong các thử nghiệm, cùng với các lợi điểm của mô hình MTL đã đề cập trong chương 3, và kết quả cải thiện khi học biểu véc-tơ biểu diễn mức âm tiết cùng với huấn luyện mô hình (thử nghiệm 1 và 2) ta nghĩ đến việc đưa mô hình ngôn ngữ vào học cùng với mô hình tách từ chính để đạt được kết quả tách từ tốt hơn. Huấn luyện và đánh giá mô hình trên tập dữ liệu 48k với ba nhãn B-I-E với thiết lập tham số tương tự như trong thử nghiệm 7, thu được kết quả như sau:

precision	recall	f1 score
98.17	98.53	98.34

Bảng 4.16: Kết quả thử nghiệm 9

Việc sử dụng mô hình multi tasks theo kiến trúc này không mang lại hiệu quả, khi kết quả *f1 score* thu được thấp hơn so với thử nghiệm 7 trước đó $(98.34 < 98.41)$

4.3 So sánh đánh giá các phương pháp

4.3.1 So sánh đánh giá trên tập 48k

Toàn bộ các thử nghiệm đều được đánh giá trên tập dữ liệu 48k, với kết quả thu được như sau:

Approach	precision	recall	f1 score
BiLSTM + softmax	96.07	96.88	96.48
Syllable Embedding Layer + BiLSTM + softmax	96.54	96.83	96.68
Syllable Embedding Layer + Character Embedding Layer + CharCNN + BiLSTM + softmax	96.95	97.32	97.13
Syllable Embedding Layer + Character Embedding Layer + CharCNN + BiLSTM + CRF	97.11	97.47	97.29
Syllable Embedding Layer + Character Embedding Layer + CharCNN + Dict-vector + BiLSTM + softmax	97.93	98.39	98.16
Syllable Embedding Layer + Character Embedding Layer + CharCNN + Dict-vector + BiLSTM + CRF	98.07	98.40	98.23
Character Embedding Layer + CharCNN + fasttext vector + Dict-vector + BiLSTM + CRF	98.10	98.52	98.31
Character Embedding Layer + CharCNN + LM vector + Dict-vector + BiLSTM + CRF	**98.25**	**98.56**	**98.41**
Character Embedding Layer + CharCNN + LM vector + pmi-vector + BiLSTM + CRF	95.66	96.90	96.28
Multi tasks	98.17	98.53	98.34

Bảng 4.17: Bảng so sánh kết quả thử nghiệm

Việc sử dụng học véc-tơ biểu diễn mức âm tiết đem lại hiệu quả tốt hơn so với việc sử dụng mô hình thông thường dùng học biểu diễn mức từ cho bài toán biểu diễn mức âm tiết này. Các kết quả thử nghiệm khi chưa dùng véc-tơ từ điển cho kết quả không cao, chỉ đạt mức $\approx 97\%$, tuy nhiên sau khi đưa véc-tơ từ điển vào mô hình, kết quả tăng lên rất cao đạt $> 98\%$. Véc-tơ từ điển thể hiện là một đặc trưng quan trọng trong quá trình huấn luyện mô hình, do đó các thử nghiệm phía sau đều sử dụng véc-tơ từ điển làm đặc trưng đầu vào. Việc sử dụng bộ phân loại *CRF* luôn cho kết quả cao hơn sử dụng bộ phân loại *softmax*

thông thường nhưng chỉ trên bộ nhãn B-I-E, do đó việc học các quan hệ trên nhãn là hiệu quả với bài toán tách từ này.

Những thử nghiệm với mô hình end-to-end tỏ ra khả quan khi kết quả cạnh tranh được với các mô hình huấn luyện trước tương ứng. Tuy nhiên do kết quả thu được khi sử dụng véc-tơ từ điển là rất tốt, nên các thử nghiệm tập trung nhiều hơn vào các mô hình huấn luyện trước.

Mô hình multi tasks đã thử nghiệm chưa cho kết quả cao như mong muốn, tuy nhiên kết quả là khá cao trong khi hoàn toàn không sử dụng thông tin trên một tập dữ liệu lớn (tập *zingnews*) và không cần thiết phải huấn luyện trước mô bất cứ mô hình nào.

Thống kê kết quả mà mô hình dự đoán trên tập kiểm thử, có 2479 từ được gán nhãn sai. Trong đó ta có thể chia thành 2 loại lỗi chính: từ chưa xuất hiện trong tập huấn luyện (1858 từ) và từ đã gặp trong tập huấn luyện (621 từ). Với những từ chưa xuất hiện trong tập huấn luyện, có 633 từ có xuất hiện trong tập huấn luyện nhưng dưới dạng khác (viết hoa hoặc viết thường). Việc sử dụng véc-tơ từ điển đồng thời cũng giúp sửa một phần lỗi này (từ 1401 giảm còn 633 lỗi).

So sánh với bộ tách từ UETSegmenter[1] trên tập dữ liệu 48K, kết quả thu được như sau:

	precision	recall	f1 score
UETSegmenter	97.90	98.33	98.11
Mô hình tốt nhất	**98.25**	**98.56**	**98.41**

Bảng 4.18: Kết quả so sánh với bộ UETSegmenter trên tập 48k

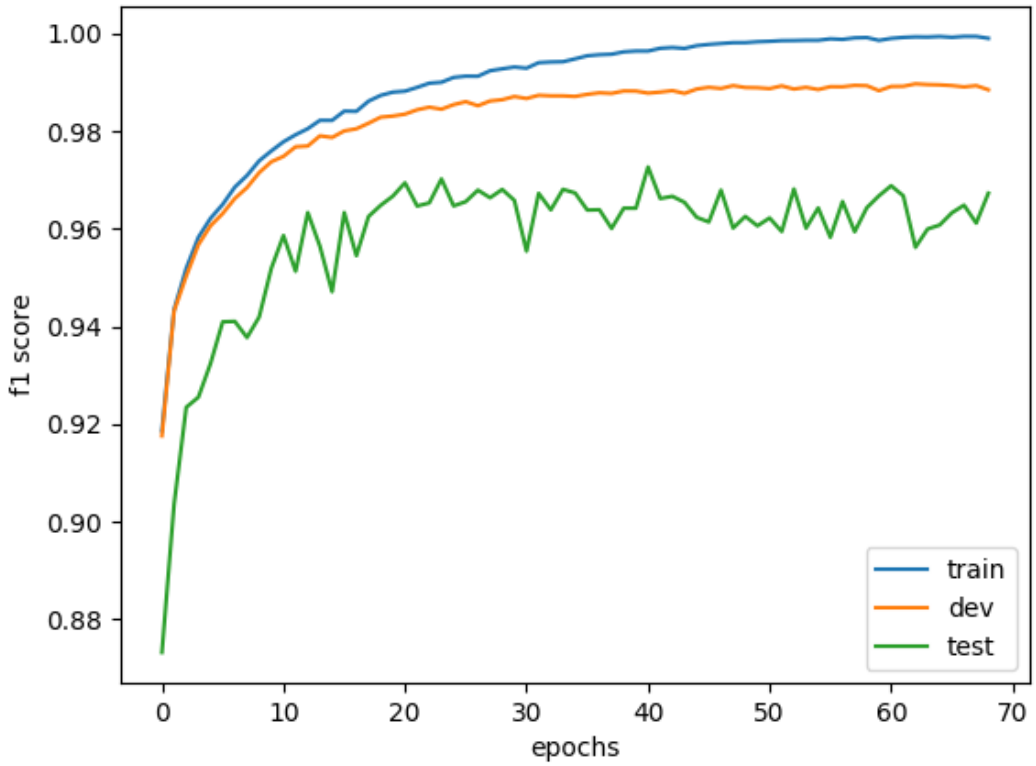
Mô hình sử dụng *CNN* để trích rút đặc trưng mức ký tự, học biểu diễn mức âm tiết thông qua quá trình học mô hình ngôn ngữ, sử dụng véc-tơ từ điển, với mô hình chính là *BiLSTM* và bộ phân loại *CRF* cho kết quả tốt hơn bộ tách từ *UETSegmenter* trên tập dữ liệu 48k.

4.3.2 So sánh đánh giá trên tập 75k

Với mô hình đạt được kết quả tốt trong các thử nghiệm trước đó, ta cần so sánh kết quả với các mô hình khác, đặc biệt là bộ tách từ tốt nhất hiện tại là RDRSegmenter. Tuy nhiên bộ tách từ này lại không cho phép huấn luyện mô hình với source code mà nhóm nghiên cứu công bố, đồng thời có một tham số trong mô hình được tùy chỉnh bằng tay trong quá trình huấn luyện, do đó ta không thể có được kết quả của bộ RDRSegmenter trên bộ dữ liệu 48k mà ta đã đánh giá. Do đó, ta lựa chọn sử dụng cùng bộ dữ liệu và phương pháp đánh giá trong bài báo của RDRSegmenter[**[13]**] công bố (bộ 75k).

Thực hiện chia ngẫu nhiên theo tỉ lệ 9-1 tập huấn luyện 75k thành tập train và tập dev. Huấn luyện mô hình tốt nhất đã thử nghiệm (mô hình Character Embedding Layer + CharCNN + LM vector + Dict-vector + BiLSTM + CRF) với cùng thiết lập tham số trên tập dữ liệu 75k sử dụng tập train, dev và test như đã đề cập, thu được kết quả như sau:

[1]Với bộ dữ liệu 48k, ta chỉ đi so sánh với bộ tách từ UETSegmenter do bộ tách từ này cho phép huấn luyện lại mô hinh với dữ liệu mới



Hình 4.9: Độ chính xác trên các tập trong huấn luyện

approach	precision	recall	f1 score
UETSegmneter	**97.51**	98.23	97.87
RDRSegmenter	97.46	**98.35**	**97.90**
My approach	96.69	97.84	97.26

Bảng 4.19: Kết quả so sánh trên tập 75k

Kết quả thu được trên bộ dữ liệu 75k không đem lại kết quả cao như mong đợi khi so sánh với 2 mô hình tốt nhất hiện tại là UETSegmenter và RDRSegmenter, trong khi với tập dữ liệu nhỏ 48k, kết quả của mô hình học sâu đề xuất cao hơn kết quả của bộ UETSegmenter, điểm khác biệt duy nhất nằm ở dữ liệu sử dụng. Với bộ dữ liệu 48k, ta hoàn toàn sử dụng dữ liệu trong tập vlsp2013 được công bố cho bài toán tách từ năm đó, với các văn bản thuộc 2 lĩnh vực (domains) là bài báo và truyện ngắn. Do đó, kết quả của mô hình học sâu đề xuất đạt kết quả rất tốt khi dữ liệu huấn luyện và kiểm tra nằm cùng trong hai lĩnh vực xác định. Đối với tập 75k, tập huấn luyện vẫn là dữ liệu thuộc 2 lĩnh vực bài báo và truyện ngắn, tuy nhiên dữ liệu kiểm tra lại thuộc lĩnh vực văn bản pháp luật.

Hình 4.9 cho thấy độ chính xác trên tập train và dev tăng đều và đồng thời trong quá trình huấn luyện, do dữ liệu thuộc hai tập này thuộc cùng lĩnh vực với nhau. Trong khi đó, kết quả trên tập test lại biến thiên mạnh và có xu hướng giảm khi mô hình fit dần với dữ liệu huấn luyện. Kết quả tập test tốt nhất đạt tại epoch 40, trong khi độ chính xác trên tập dev tiếp tục tăng và early stop tại epoch 68. Với kết quả thu được, mô hình học sâu đề xuất kém hơn khi đánh giá

trên dữ liệu thuộc lĩnh vực khác, tuy nhiên khi trong cùng lĩnh vực với dữ liệu huấn luyện, mô hình học sâu đề xuất lại cho kết quả tốt hơn.

Chương 5

Kết luận

5.1 Các kết quả đã đạt được

Trong quá trình làm đồ án này, em đã thu được các kết quả như sau:

- Tìm hiểu về bài toán tách từ và một phần lý thuyết về ngôn ngữ tiếng Việt
- Tìm hiểu về các mô hình học máy và học sâu, đặc biệt là các mô hình được áp dụng vào bài toán tách từ
- Đề xuất thử nghiệm những mô hình đã có và đề xuất một vài kiến trúc mô hình mới nhằm giải quyết bài toán tách từ trên bộ dữ liệu tiếng Việt.
- Kết quả thu được cao hơn bộ tách từ tốt thứ 2 hiện tại UETSegmenter trên tập dữ liệu sử dụng

Ngoài việc tìm hiểu và làm chủ được các kiến thức về tách từ, học máy, học sâu để áp dụng vào bài toán, thông qua đồ án này, em còn hiểu thêm nhiều về ngôn ngữ tiếng việt, cùng với đó là các phương pháp xử lý dữ liệu, đánh giá mô hình và hướng phát triển mô hình mới nhằm cải tiến độ chính xác của mô hình đã có.

5.2 Hướng phát triển trong tương lai

Trong quá trình thực hiện đồ án, nhiều ý tưởng mô hình đã được đề xuất và thử nhiệm, tuy nhiên vẫn còn những ý tưởng chưa được thực thi cũng như một số hướng phát triển tiếp đồ án trong tương lai nhằm cải thiện chất lượng cũng như hiệu năng của mô hình:

- Tìm cách mô phỏng véc-tơ từ điển thông qua mô hình học không giám sát trên dữ liệu lớn hoặc thay đổi mô hình để có khả năng biểu diễn được thông tin này
- Sử dụng thêm các phương pháp biểu diễn thông tin mức âm tiết hoặc dưới âm tiết
- Sử dụng đặc trưng bằng cách fine-tune các mô hình biểu diễn ngôn ngữ mới và đang rất nổi vào thời điểm hiện tại (BERT[16], ELMo [17])

- Sử dụng thêm post processing kết quả đầu ra của mô hình nhằm tăng độ chính xác
- Xây dựng mô hình đạt kết quả tốt trên bộ dữ liệu 75k

Tài liệu tham khảo

[1] Nguyễn Thị Minh Huyền, Hoàng Thị Tuyền Linh, Vũ Xuân Lương, *Hướng dẫn nhận biết đơn vị từ trong văn bản tiếng Việt*, Vietnamese Language and Speech Processing workshop, 2013.

[2] Prajit Ramachandran, Barret Zoph, Quoc V. Le, *Searching for Activation Functions*, arXiv preprint arXiv:1710.05941, 16 Oct 2017.

[3] Florian Schroff, Dmitry Kalenichenko, James Philbin, *FaceNet: A Unified Embedding for Face Recognition and Clustering*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015.

[4] Baidu Research – Silicon Valley AI Lab, *Deep Speech 2: End-to-End Speech Recognition in English and Mandarin*, arXiv preprint arXiv:1512.02595, 8 Dec 2015.

[5] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, in Journal of Machine Learning Research volume 15(Jun): 1929-1958, 2014.

[6] LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE. 86 (11): 2278–2324.

[7] Sepp Hochreiter, Jürgen Schmidhuber, *Long Short-Term Memory*, in Neural Computation Journal, Volume 9 Issue 8: 1735-1780 November 15, 1997.

[8] Tobias Glasmachers, *Limits of End-to-End Learning*, arXiv preprint arXiv:1704.08305v1, 26 Apr 2017.

[9] Sebastian Ruder, *An Overview of Multi-Task Learning in Deep Neural Networks*, arXiv preprint arXiv:arXiv:1706.05098v1, 15 Jun 2017.

[10] Marco Baroni, Georgiana Dinu, German Kruszewski, *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), June 2014.

[11] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov, *FastText.zip: Compressing text classification models*, arXiv preprint arXiv:1612.03651, 12 Dec 2016.

[12] T. P. Nguyen and A. C. Le, *A hybrid approach to Vietnamese word segmentation*, in IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), pages 114-119, 2016.

[13] Dat Quoc Nguyen and Dai Quoc Nguyen and Thanh Vu and Mark Dras and Mark Johnson, *A Fast and Accurate Vietnamese Word Segmenter*, Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), pages 2582-2587.

[14] Le, H. P., Nguyen, T. M. H., Roussanaly, A., and Ho, T. V., *A hybrid approach to word segmentation of Vietnamese texts*, In Proceedings of the 2nd International Conference on Language and Automata Theory and Applications, pages 240–249.

[15] Van-Duyet Le, *vietnamese-wordlist*, https://vietnamese-wordlist.duyet.net.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv:1810.04805, 11 Oct 2018.

[17] Peters, Matthew E. and Neumann, Mark and Iyyer, Mohit and Gardner, Matt and Clark, Christopher and Lee, Kenton and Zettlemoyer, Luke, *Deep contextualized word representations*, NAACL 2018.

[18] Vũ Hữu Tiệp, *Multi-layer Perceptron và Backpropagation*, https://machinelearningcoban.com/2017/02/24/mlp/.

[19] Michael Nielsen, *Neural Networks and Deep Learning*, http://neuralnetworksanddeeplearning.com/.

[20] Christopher Olah, *Understanding LSTM Networks*, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[21] *CS231n: Convolutional Neural Networks for Visual Recognition*, http://cs231n.github.io/convolutional-networks/.