# **TÓM TẮT**

Tên đề tài: Phân loại tin giả cho Tiếng Việt trong lĩnh vực chính trị

Sinh viên thực hiện: Trần Thế Dâng

Số thẻ SV: 102180158 Lớp: 18TCLC\_DT2

Theo Cục An ninh mạng và phòng chống tội phạm sử dụng công nghệ cao – Bộ Công an, từ năm 2020 đến năm 2021, công an cả nước đã triệu tập, đấu tranh với hơn 1.800 người, khởi tố xử lý hình sự 21 người, xử phạt vi phạm hành chính 466 trường hợp với số tiền hơn 5 tỷ đồng vì phát tán tin giả.

Vì vậy để có thể góp phần vào công cuộc phát hiện, ngăn chặn tin giả, em đã nghiên cứu về học máy và sâu hơn là học sâu để áp dụng vào việc phân loại tin giả, song song với việc tìm ra được giải pháp tối ưu nhất trong các giải pháp được nghiên cứu.

Đề tài này nghiên cứu về hai phương pháp biểu diễn văn bản Bag-of-words và TF-IDF và hai phương pháp học máy là Naive Bayes và Recurrent Neural Network, cụ thể là tính hiệu quả khi kết hợp hai phương pháp biểu diễn với hai phương pháp học máy trong việc phát hiện tin giả.

**NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN**

………………………………………………………………………….………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

*Đà Nẵng, ngày … tháng … năm 20…*

Giảng viên hướng dẫn

**TS. Ninh Khánh Duy**

**NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN**

………………………………………………………………………….………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

………………………………………………………………………….

|  |  |
| --- | --- |
| ĐẠI HỌC ĐÀ NẴNG  **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  KHOA CÔNG NGHỆ THÔNG TIN | **CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM**  Độc lập - Tự do - Hạnh phúc |

# **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Trần Thế Dâng Số thẻ sinh viên: 102180158

Lớp: 18TCLC\_DT2 Khoa: Công nghệ thông tin Ngành: Công nghệ thông tin

1. *Tên đề tài đồ án:*Phân loại tin giả cho Tiếng Việt trong lĩnh vực chính trị
2. *Đề tài thuộc diện:*  *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:*

*……………………………………..……………………………………………..……......……………………………………………………………………………………………..………………………………….…..………………………..………………………*

1. *Nội dung các phần thuyết minh và tính toán:*

* Chương 1: Tổng quan: trình bày tổng quan về nội dung đề tài, xử lý ngôn ngữ tự nhiên và bài toán phân loại văn bản.
* Chương 2: Giải pháp phát hiện tin giả dùng học sâu: trình bày về khái niệm tin giả, học máy, học sâu và giải pháp phát hiện tin giả khi sử dụng học máy và học sâu.
* Chương 3: Triển khai và đánh giá kết quả: trình bày cách cài đặt môi trường, triển khai mã nguồn và các hình ảnh kết quả của chương trình.
* Kết luận: trình bày kết quả đạt được và hướng phát triển trong tương lai.

1. *Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Họ tên người hướng dẫn:* Ninh Khánh Duy
2. *Ngày giao nhiệm vụ đồ án:*  *…/…/2022*
3. *Ngày hoàn thành đồ án: …/…/2022*

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày tháng năm 20* |
| **Trưởng Bộ môn** …………………….. | **Người hướng dẫn** |

# **LỜI NÓI ĐẦU**

Đề tài “Phân loại tin giả cho Tiếng Việt trong lĩnh vực chính trị” đã được hoàn thành. Trong quá trình nghiên cứu và hoàn thiện, em đã nhận được sự hướng dẫn và giúp đỡ nhiệt tình từ các thầy cô, đặc biệt là T.S. Ninh Khánh Duy.

Em xin gửi lời cảm ơn chân thành tới nhà trường đã tận tình chỉ bảo, góp ý và tạo điều kiện cho em hoàn thành đề tài nghiên cứu một cách tốt nhất.

Em xin cảm ơn T.S. Ninh Khánh Duy, thầy đã ân cần, tận tâm, hết mình hỗ trợ, hướng dẫn, thúc đẩy em trong quá trình thực hiện đề tài.

Trong quá trình thực hiện đề tài tốt nghiệp, em đã cố gắng nỗ lực hết mình, tuy nhiên không tránh khỏi sai sót. Em mong nhận được sự góp ý của thầy cô giáo để đề tài của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

# **CAM ĐOAN**

Tôi xin cam đoan đồ án trên là công trình nghiên cứu của riêng bản thân mình dưới sự hướng dẫn của T.S. Ninh Khánh Duy. Những nhận định được nêu ra trong đồ án cũng là kết quả từ sự nghiên cứu trực tiếp, nghiêm túc, độc lập của bản thân tác giả dựa và các cơ sở tìm kiếm, hiểu biết và nghiên cứu tài liệu khoa học hay bản dịch khác đã được công bố. Đồ án vẫn sẽ giúp đảm bảo được tính khách quan, trung thực và khoa học.

Các số liệu và kết quả nghiên cứu được đưa ra trong đồ án là trung thực và không sao chép hay sử dụng kết quả của bất kỳ đề tài nghiên cứu nào tương tự. Nếu như phát hiện rằng có sự sao chép kết quả nghiên cứu đề những đề tài khác bản thân tôi xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày 14 tháng 12 năm 2022

Sinh viên thực hiện

Trần Thế Dâng

# **MỤC LỤC**

[**TÓM TẮT** i](#_Toc122000302)

[**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** iv](#_Toc122000303)

[**LỜI NÓI ĐẦU** i](#_Toc122000304)

[**CAM ĐOAN** ii](#_Toc122000305)

[**MỤC LỤC** iii](#_Toc122000306)

[**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT** vi](#_Toc122000307)

[**DANH SÁCH BẢNG VÀ HÌNH VẼ** vii](#_Toc122000308)

[**MỞ ĐẦU** 1](#_Toc122000309)

[**Chương 1: Tổng quan** 3](#_Toc122000310)

[**1.1. Nội dung của đề tài** 3](#_Toc122000311)

[**1.2. Tổng quan về xử lý ngôn ngữ tự nhiên** 4](#_Toc122000312)

[*1.2.1. Khái niệm*…………………………………………………………………………………………………………………….4](#_Toc122000313)

[*1.2.2. Các bài toán cơ bản của xử lý ngôn ngữ tự nhiên* 5](#_Toc122000314)

[*1.2.3. Các bước để xử lý ngôn ngữ tự nhiên* 6](#_Toc122000315)

[**1.3. Bài toán phân loại văn bản** 7](#_Toc122000316)

[*1.3.1. Văn bản*……………………………………………………………………………………………………………………….7](#_Toc122000317)

[*1.3.2. Biểu diễn văn bản bằng vector đặc trưng* 8](#_Toc122000318)

[*1.3.3. Phân loại văn bản* 9](#_Toc122000319)

[*1.3.4. Các bước phân loại* 11](#_Toc122000320)

[**Chương 2: Giải pháp phát hiện tin giả dùng học sâu** 12](#_Toc122000321)

[**2.1. Tin giả và phân loại tin giả** 12](#_Toc122000322)

[*2.1.1. Tin sai (mis-information)* 13](#_Toc122000323)

[*2.1.2 Tin xuyên tạc, tin dắt mũi (dis-information)* 13](#_Toc122000324)

[*2.1.3 Tin nguy hại (mal-information)* 14](#_Toc122000325)

[**2.2. Học máy nói chung và học sâu nói riêng** 15](#_Toc122000326)

[*2.2.1. Học máy*……………..……………………………………………………………………………………………………..15](#_Toc122000327)

[*2.2.2. Mạng nơ-ron* …………………………………………………………………………………………………………….16](#_Toc122000328)

[*2.2.3. Học sâu*………………………………………………………………………………………………………………………17](#_Toc122000329)

[*2.2.4. Phân loại học sâu* 18](#_Toc122000330)

[**2.3. Giải pháp phân loại tin giả dùng học máy và học sâu** 19](#_Toc122000331)

[*2.3.1. Tổng quan vấn đề và giải pháp* 19](#_Toc122000332)

[*2.3.2. Mô hình tổng quát* 20](#_Toc122000333)

[*2.3.3. Biểu diễn văn bản bằng BoW* 21](#_Toc122000334)

[*2.3.4. Biểu diễn văn bản bằng TF-IDF* 22](#_Toc122000335)

[*2.3.5. Phân loại văn bản bằng mô hình Naive Bayes* 25](#_Toc122000336)

[*2.3.6. Phân loại văn bản bằng mô hình RNN* 26](#_Toc122000337)

[**Chương 3: Thực nghiệm và đánh giá** 31](#_Toc122000338)

[**3.1. Cài đặt môi trường** 31](#_Toc122000339)

[**3.2. Mô tả dữ liệu** 33](#_Toc122000340)

[*3.2.1. Nguồn gốc, đặc tính của dữ liệu* 33](#_Toc122000341)

[*3.2.2. Tiền xử lý dữ liệu* 35](#_Toc122000342)

[**3.3. Bag-of-words và TF-IDF kết hợp với Naive Bayes** 36](#_Toc122000343)

[*3.3.1. Biểu diễn văn bản bằng Bag-of-words* 36](#_Toc122000344)

[*3.3.2. Biểu diễn văn bản bằng TF-IDF* 37](#_Toc122000345)

[*3.3.3. Đánh giá*…………………………………………………………………………………………………………………….38](#_Toc122000346)

[*3.3.4. Phân loại bằng mô hình Multinomial Naive Bayes* 38](#_Toc122000347)

[**3.4. Mô hình RNN** 41](#_Toc122000348)

[*3.4.1. Vấn đề gặp phải khi biểu diễn văn bản và cách giải quyết* 41](#_Toc122000349)

[*3.4.2. Xây dựng mô hình RNN* 43](#_Toc122000350)

[**KẾT LUẬN** 48](#_Toc122000351)

[**TÀI LIỆU THAM KHẢO** 49](#_Toc122000352)

# **DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| RNN | Recurrent Neural Network |
| CNN | Convolutional Neural Network |
| NLP | Natural Language Processing |
| NB | Naive Bayes |
| BoW | Bag-of-words |
| TF-IDF | Term frequency–Inverse document frequency |
| VAFC | Vietnam Anti Fake-news Center |
| MLP | Multi Layer Perceptron |
| LM | Language Modelling |
| NP | Non-deterministic Polynomial-time |
| AI | Artificial Intelligence |
| CFG | Context-free Grammar |
| CCG | Combinatory Categorial Grammar |
| DG | Dependency Grammar |
| LSTM | Long-short Term Memory |

**DANH SÁCH BẢNG VÀ HÌNH VẼ**

**Danh sách hình vẽ**

[Hình 1.1 Ví dụ về các mô hình và ứng dụng của Xử lý ngôn ngữ tự nhiên 5](#_Toc121863425)

[Hình 1.2. Ví dụ về tách từ và gán nhãn từ loại 6](#_Toc121863426)

[Hình 1.3. Mô hình tổng quát của Phân loại văn bản 11](#_Toc121863427)

[Hình 2.1. Phân biệt các loại tin giả 12](#_Toc121995141)

[Hình 2.2. Mô hình Mạng nơ-ron 17](#_Toc121995142)

[Hình 2.3. Mối quan hệ giữa Mạng nơ-ron và Học sâu 17](#_Toc121995143)

[Hình 2.4. Mô hình tổng quát 21](#_Toc121995144)

[Hình 2.5. Bag-of-words 21](#_Toc121995145)

[Hình 2.6. Cách xây dựng nên vector nhị phân của Bag-of-words 22](#_Toc121995146)

[Hình 2.7. Ví dụ về Naive Bayes 25](#_Toc121995147)

[Hình 2.8. Mạng nơ-ron thông thường 27](#_Toc121995148)

[Hình 2.9. Mô hình RNN Many to One 28](#_Toc121995149)

[Hình 2.10. Mô hình RNN Many to Many 28](#_Toc121995150)

[Hình 2.11. Mô hình One to Many 29](#_Toc121995151)

[Hình 2.13. Một khung RNN đơn giản 30](#_Toc121995152)

[Hình 2.14. Ví dụ một mô hình RNN dự đoán từ 30](#_Toc121995153)

[Hình 3.1. Các hàm cần import 32](#_Toc121948742)

[Hình 3.2. Cái nhìn tổng quan về dataset 33](#_Toc121948743)

[Hình 3.3. Số lượng của tin thật và tin giả trong dataset 34](#_Toc121948744)

[Hình 3.4. Word Cloud của tin thật 34](#_Toc121948745)

[Hình 3.5. Word Cloud của tin giả 35](#_Toc121948746)

[Hình 3.6. Ví dụ 1 đoạn văn bản sau khi được xử lý 36](#_Toc121948747)

[Hình 3.7. Chạy hàm tiền xử lý 1023 bài báo 36](#_Toc121948748)

[Hình 3.8. Kết quả khi chạy Bag-of-words 37](#_Toc121948749)

[Hình 3.9. Kết quả khi chạy TF-IDF 38](#_Toc121948750)

[Hình 3.10. Kết quả mô hình NB kết hợp BoW 39](#_Toc121948751)

[Hình 3.11. Kết quả mô hình NB kết hợp TF-IDF 40](#_Toc121948752)

[Hình 3.12. Vector một bài báo mẫu 42](#_Toc121948753)

[Hình 3.13. Trích xuất đặc trưng phù hợp với mô hình 43](#_Toc121948754)

[Hình 3.14. Thiết lập mô hình RNN 44](#_Toc121948755)

[Hình 3.15. Train mô hình với epochs = 5 44](#_Toc121948756)

[Hình 3.16. Độ chính xác của RNN 45](#_Toc121948757)

[Hình 3.17. Ma trận nhầm lẫn của mô hình RNN 46](#_Toc121948758)

**Danh sách bảng**

[Bảng 3.1. Bảng thông tin của 1 số thư viện được sử dụng 32](#_Toc121948840)

[Bảng 3.2 .Kết quả sau nhiều lần chạy thử NB kết hợp BoW 39](#_Toc121948841)

[Bảng 3.3. Kết quả sau nhiều lần chạy thử NB kết hợp TF-IDF 40](#_Toc121948842)

# **MỞ ĐẦU**

Vì tin giả như một loại virus độc hại, nó xâm nhập, gây rối dư luận, gây rối lòng tin, thậm chí làm khủng hoảng niềm tin. Các thế lực phản động, thù địch và cơ hội chính trị lợi dụng điều này để xuyên tạc, kích động chống phá hòng gây mất ổn định đất nước ta. Vì thế, phải đấu tranh mạnh mẽ để loại bỏ mối hiểm họa thật sự này.

Thông qua đề tài này em muốn nghiên cứu ứng dụng kỹ thuật học máy và sâu hơn là học sâu vào phân loại văn bản tiếng Việt nói chung và tin giả nói riêng. Đưa ra được kết quả thực nghiệm khi so sánh các phương pháp phân loại nhằm chọn ra phương pháp tối ưu nhất. Đồng thời củng cố kho dữ liệu và các công cụ để phục vụ phân loại văn bản Tiếng Việt.

* + 1. **Mục đích**
* Nghiên cứu mở rộng kiến thức về lập trình Python, kiến thức về trí tuệ nhân tạo, học máy và học sâu.
* Tìm hiểu và thử nghiệm các phương pháp học máy và học sâu vào phân loại văn bản, cụ thể hơn là phân loại tin giả cho Tiếng Việt
  + 1. **Ý nghĩa**
* Hỗ trợ việc phân loại tin giả cho Tiếng Việt trong lĩnh vực chính trị, qua đó thúc đẩy sự phát triển của đất nước và làm sạch mạng lưới thông tin.
  + 1. **Phương pháp thực hiện**
* Phương pháp thu thập dữ liệu từ các nguồn tài liệu của các nghiên cứu khoa học, các blog học thuật, các bài viết của các chuyên gia,…
* Phương pháp nghiên cứu toán học để diễn dãi cách hoạt động của từng mô hình.
* Phương pháp nghiên cứu định lượng, tổng hợp kết quả để đưa ra sự so sánh giữa các mô hình.
  + 1. **Phạm vi đề tài**

Phạm vi đề tài chỉ bao gồm cho tin tức Tiếng Việt trong lĩnh vực chính trị. Phạm vi xoay quanh các phương pháp biểu diễn văn bản Bag-of-words, TF-IDF, Encoding từ và các phương pháp phân loại văn bản Naive Bayes, Recurrent Neural Network.

* + 1. **Nhiệm vụ của đề tài**
* Tìm hiểu về học máy và học sâu.
* Ứng dụng được học máy và học sâu vào phát hiện tin giả trong phạm vi đề tài.
* Tìm ra phương pháp tối ưu hơn trong các phương pháp được nêu ra trong phạm vi đề tài.
  + 1. **Bố cục đồ án**

Đồ án bao gồm các nội dung sau:

*Mở đầu*

*Chương 1: Tổng quan*

*Chương 2: Giải pháp phát hiện tin giả dùng học máy và học sâu*

*Chương 3: Thực nghiệm và đánh giá*

*Kết luận và hướng phát triển.*

# **Chương 1: Tổng quan**

## **1.1. Nội dung của đề tài**

Ngày nay, cùng với sự phát triển vượt bậc của khoa học kỹ thuật và công nghệ thông tin đã đem đến cho con người khả năng tiếp cận với tri thức khoa học một cách nhanh chóng, cụ thể như: thư viện điện tử, cổng thông tin điện tử, báo mạng, các ứng dụng tìm kiếm,… Giúp con người thuận tiện hơn trong việc trao đổi, cập nhật thông tin trên toàn cầu thông qua mạng Internet.

Tuy nhiên, ngoài những thông tin hữu ích, chúng ta cũng đã phải gặp những thông tin, bài báo sai lệch, giả dối. Vấn nạn tin giả vì vậy đã trở thành một vấn đề nhức nhối chưa có cách giải quyết triệt để.

Tin giả như một loại virus độc hại, nó xâm nhập, gây rối dư luận, gây rối lòng tin, thậm chí làm khủng hoảng niềm tin. Các thế lực phản động, thù địch và cơ hội chính trị lợi dụng điều này để xuyên tạc, kích động chống phá hòng gây mất ổn định đất nước ta. Vì thế, phải đấu tranh mạnh mẽ để loại bỏ mối hiểm họa thật sự này.

Việc phân loại tin giả hiện nay thường được làm bằng thủ công, tức là chính bằng kiểm duyệt trực tiếp của con người, hay các chuyên gia. Bộ Thông tin và Truyền thông đã mở một cổng trao đổi với tên miền [www.tingia.gov.vn](http://www.tingia.gov.vn), vận hành bởi Trung tâm Xử lý Tin giả Việt Nam (VAFC) dưới sự quản lý của Cục Phát thanh, Truyền hình và Thông tin điện tử. Triết lý hoạt động của trung tâm là “Tin giả do con người tạo ra nên chỉ có duy nhất con người mới có thể nhận biết và xử lý được tin giả.”

Tuy nhiên để tiết kiệm nguồn lực cũng như tận dụng trí tuệ nhân tạo, em đã tìm hiểu về đề tài này. Phương pháp em sử dụng dựa vào phân loại văn bản bằng học sâu. Mục đích là huấn luyện một mô hình để có thể phân loại văn bản tự động. Và qua đó tạo được một kho dữ liệu và các công cụ phục vụ phân loại văn bản Tiếng Việt.

Sử dụng học máy các mô hình liên quan để tăng hiệu quả, tỷ lệ chính xác của bộ phân lớp nói riêng, của đề tài nói chung.

Đề tài của em gồm 3 phần: Phần đầu tiên là tổng quan về vấn đề xử lý ngôn ngữ tự nhiên và bài toán phân loại văn bản. Phần thứ hai là chi tiết, lý thuyết và phương trình của phương pháp học sâu được áp dụng trong đề tài. Phần thứ ba là thực nghiệm, kết quả và đánh giá.

## **1.2. Tổng quan về xử lý ngôn ngữ tự nhiên**

*1.2.1. Khái niệm*

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên ra đời từ những năm 40 của thế kỷ 20, trải qua các giai đoạn phát triển với nhiều phương pháp và mô hình xử lý khác nhau. Có thể kể tới các phương pháp sử dụng ô-tô-mát và mô hình xác suất (những năm 50), các phương pháp dựa trên ký hiệu, các phương pháp ngẫu nhiên (những năm 70), các phương pháp sử dụng học máy truyền thống (những năm đầu thế kỷ 21), và đặc biệt là sự bùng nổ của học sâu trong thập kỷ vừa qua.

Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, không hoàn toàn độc lập, bao gồm xử lý tiếng nói (speech processing) và xử lý văn bản (text processing). Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói (dữ liệu âm thanh). Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản tự động, hay kiểm lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu liên quan tới các bài toán phân tích văn bản thì sinh liên quan tới nhiệm vụ tạo ra văn bản mới như trong các ứng dụng về dịch máy hoặc tóm tắt văn bản tự động.

Diagram

Description automatically generated

Hình 1.1 Ví dụ về các mô hình và ứng dụng của Xử lý ngôn ngữ tự nhiên

*1.2.2. Các bài toán cơ bản của xử lý ngôn ngữ tự nhiên*

* Mô hình hóa ngôn ngữ (Language modelling): gán một xác suất cho bất kỳ chuỗi từ nào. Về cơ bản, trongbài toán này, ta cần dự đoán từ tiếp theo xuất hiện theo trình tự, dựa trên lịch sử của các từ đã xuất hiện trước đó. LM rất quan trọng trong các ứng dụng khác nhau của NLP, và là lý do tại sao máy móc có thể hiểu được thông tin định tính. Một số ứng dụng của Mô hình hóa ngôn ngữ bao gồm: nhận dạng giọng nói, nhận dạng ký tự quang học, nhận dạng chữ viết tay, dịch máy và sửa lỗi chính tả.
* Phân loại văn bản (Text classification): gán các danh mục được xác định trước cho văn bản dựa trên nội dung của nó. Cho đến nay, phân loại văn bản là ứng dụng phổ biến nhất của NLP, được sử dụng để xây dựng các công cụ khác nhau như trình phát hiện thư rác và chương trình phân tích cảm xúc.
* Trích xuất thông tin (Information extraction): là tự động trích xuất thông tin có liên quan từ các tài liệu văn bản không có cấu trúc và / hoặc bán cấu trúc. Ví dụ về các loại tài liệu này bao gồm lịch sự kiện từ email hoặc tên của những người được đề cập trong một bài đăng trên mạng xã hội.
* Truy xuất thông tin (Information retrieval): làm nhiệm vụ tìm kiếm các tài liệu có liên quan từ một bộ dữ liệu lớn các tài liệu liên quan đến truy vấn do người dùng thực hiện.
* Tác tử phần mềm hội thoại (Conversational agent): thuộc AI hội thoại, liên quan đến việc xây dựng các hệ thống đối thoại mô phỏng các tương tác của con người. Các ví dụ phổ biến về AI hội thoại bao gồm Alexa, Siri, Google Home, Cortana, hay trợ lý ảo ViVi. Các công nghệ như chatbot cũng được hỗ trợ bởi tác tử phần mềm hội thoại và ngày càng phổ biến trong các doanh nghiệp.
* Tóm tắt văn bản (Text summarization): là quá trình rút ngắn một tập hợp dữ liệu để tạo một tập hợp con đại diện cho thông tin quan trọng nhất hoặc có liên quan trong nội dung gốc
* Hỏi đáp (Question answering): là bài toán xây dựng các hệ thống có thể tự động trả lời cho các câu hỏi do con người đặt ra bằng ngôn ngữ tự nhiên.
* Dịch máy (Machine translation): là một nhánh con của ngôn ngữ học tính toán liên quan đến việc chuyển đổi một đoạn văn bản từ ngôn ngữ này sang ngôn ngữ khác. Một ứng dụng phổ biến của loại này là Google Dịch.
* Mô hình hóa chủ đề (Topic modelling): là một kỹ thuật Học máy không giám sát giúp khám phá cấu trúc chủ đề của một bộ tài liệu lớn. Ứng dụng NLP này là một công cụ khá phổ biến, được sử dụng trên nhiều lĩnh vực khác nhau – như Văn học, và Tin sinh học.

*1.2.3. Các bước để xử lý ngôn ngữ tự nhiên*

* Phân tích hình vị: là sự nhận biết, phân tích, và miêu tả cấu trúc của hình vị trong một ngôn ngữ cho trước và các đơn vị ngôn ngữ khác, như từ gốc, biên từ, phụ tố, từ loại, v.v. Trong xử lý tiếng Việt, hai bài toán điển hình trong phần này là tách từ (word segmentation) và gán nhãn từ loại (part-of-speech tagging), ví dụ ở Hình 1.2.

Graphical user interface, text, application

Description automatically generated

Hình 1.2. Ví dụ về tách từ và gán nhãn từ loại

* Phân tích cú pháp: là quy trình phân tích một chuỗi các biểu tượng, ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ máy tính, tuân theo văn phạm hình thức. Văn phạm hình thức thường dùng trong phân tích cú pháp của ngôn ngữ tự nhiên bao gồm Văn phạm phi ngữ cảnh (Context-free grammar – CFG), Văn phạm danh mục kết nối (Combinatory categorial grammar – CCG), và Văn phạm phụ thuộc (Dependency grammar – DG). Đầu vào của quá trình phân tích là một câu gồm một chuỗi từ và nhãn từ loại của chúng, và đầu ra là một cây phân tích thể hiện cấu trúc cú pháp của câu đó.
* Phân tích ngữ nghĩa: là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ. Phân tích ngữ nghĩa bao gồm hai mức độ: Ngữ nghĩa từ vựng biểu hiện các ý nghĩa của những từ thành phần, và phân biệt nghĩa của từ; Ngữ nghĩa thành phần liên quan đến cách thức các từ liên kết để hình thành những nghĩa rộng hơn.
* Phân tích diễn ngôn: là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng (context-of-use). Phân tích diễn ngôn, do đó, được thực hiện ở mức độ đoạn văn hoặc toàn bộ văn bản thay vì chỉ phân tích riêng ở mức câu.

## **1.3. Bài toán phân loại văn bản**

*1.3.1. Văn bản*

Văn bản được hiểu theo nghĩa rộng là một thực thể mang thông tin được ghi bằng ký hiệu ngôn ngữ của con người. Văn bản dùng để lưu trữ, ghi nhận và truyền đạt thông tin từ người này đến người khác. Có nhiều hình thức thể hiện văn bản. Thể hiện được dùng rộng rãi nhất là thể hiện trên giấy như báo giấy, tác phẩm văn học, khoa học kỹ thuật, công văn, khẩu hiệu,… Ngoài ra còn có các thể hiện bằng âm thanh như băng ghi âm, đĩa nghe và thể hiện bằng bản vẽ,… Hiện nay, với sự phát triển của khoa học máy tính, việc lưu trữ hay truyền tải thông tin còn có thể trên các tập tin như “.txt”, “.pdf”, hay “.doc”,”.docx”. Vì vậy những tập tin này cũng có thể được gọi là văn bản. Vì tất cả mọi thông tin, dữ liệu trên máy tính đều được lưu trữ dưới dạng hệ cơ số nhị phân, nên nghiên cứu này định nghĩa những văn bản thể hiện trên máy tính là “Văn bản số”. Cụ thể hơn, khi các văn bản số được viết bởi ngôn ngữ tiếng Việt thì gọi là “Văn bản số tiếng Việt”

Văn bản có thể được biểu diễn dưới các dạng:

* **Vector**: Là dạng dữ liệu cơ bản nhất. Nó thể hiện đặt tính của một sự vật, sự việc trong một môi trường cụ thể.
* **Danh sách**: Là danh sách dữ liệu hoặc đặc tính được liệt kê của sự vật, sự việc.
* **Tập hợp**: là một tập hợp các dữ liệu.
* **Ma trận**: thường như là một bảng dữ liệu 2 chiều trong đó dữ liệu có thể được xác định khi và chỉ khi biết chính xác số hàng và số cột của dữ liệu đó.
* **Hình ảnh**: hình ảnh được hiểu như một mảng hai chiều, trong đó dữ liệu là các con số như cường độ ánh sáng, màu sắc, điểm ảnh (pixel) của ảnh được số hóa.
* **Video**: là một danh sách các hình ảnh chúng được biểu diễn bởi một mảng 3 chiều để thuận lợi trong việc tính toán, xử lý.
* **Cây hoặc đồ thị**: thể hiện các mối quan hệ giữa các dữ liệu với nhau thông qua các nút của cây hoặc các đỉnh của đồ thị.
* **Xâu lý tự**: là một chuỗi các ký tự.
* **Cấu trúc sưu tập**: là cấu trúc có thể hỗn hợp của nhiều kiểu dữ liệu khác nhau khi thể hiện một đối tượng nào đó.

*1.3.2. Biểu diễn văn bản bằng vector đặc trưng*

Bước đầu tiên trong qui trình phân lớp văn bản là thao tác chuyển văn bản đang được mô tả dưới dạng chuỗi các từ thành một mô hình khác, sao cho phù hợp với các thuật toán phân lớp. Thông thường người ta biểu diễn văn bản dưới dạng một véc tơ đặc trưng, cụ thể là véc tơ có trọng số.

Ý tưởng của mô hình này là xem mỗi văn bản Di = di,i, trong đó:

* di là vector đặc trưng của văn bản và di = {wi1,wi2,…,win} trong đó n là số lượng đặc trưng, wij là trọng số (số lần xuất hiện) của đặc trưng thứ j
* i là chỉ số để nhận diện văn bản này

Vấn đề ta cần quan tâm là lựa chọn đặc trưng và số chiều cho không gian vector, chọn bao nhiêu từ, chọn các từ nào, phương pháp chọn ra sao.

Việc lựa chọn phương pháp biểu diễn văn bản để áp dụng vào bài toán phân lớp tuỳ thuộc vào độ thích hợp, phù hợp, độ đo đánh giá mô hình phân lớp của phương pháp đó sử dụng so với bài toán mà chúng ta đang xem xét giải quyết. Ví dụ nếu văn bản là một trang Web thì sẽ có phương pháp để lựa chọn đặc trưng khác so với các loại văn bản khác.

Khi biểu diễn văn bản dưới dạng véc tơ, ta thấy chúng có các đặc điểm sau:

- Số chiều không gian đặc trưng thường rất lớn. Các văn bản càng dài, lượng thông tin nó đề cập đến nhiều vấn đề thì không gian đặc trưng càng lớn.

- Các đặc trưng độc lập khác nhau, sự kết hợp các đặc trưng này thường không có ý nghĩa trong phân loại.

- Các đặc trưng có tính rời rạc. Véc tơ đặc trưng di có thể có nhiều thành phần mang giá trị 0, do đó có nhiều đặc trưng không xuất hiện trong văn bản di (nếu chúng ta tiếp cận theo cách sử dụng giá trị nhị phân 0,1 để biểu diễn cho việc có xuất hiện hay không một đặc trưng nào đó trong văn bản đang được biểu diễu thành véc tơ). Tuy nhiên, nếu đơn thuần cách tiếp cận sử dụng giá trị nhị phân 0,1 này thì kết quả phân loại phần nào hạn chế là do có thể đặc trưng đó không có trong văn bản đang xét, nhưng trong văn bản đang xét lại có từ khóa khác với từ đặc trưng nhưng có ngữ nghĩa giống với từ đặc trưng này, do đó một cách tiếp cận khác là không sử dụng số nhị phân 0,1 mà sử dụng giá trị số thực để phần nào giảm bớt sự rời rạc trong véc tơ văn bản.

Hầu hết các văn bản có thể được phân chia một cách tuyến tính bằng các hàm tuyến tính. Như vậy, độ dài của véc tơ là số các từ khoá xuất hiện trong ít nhất một mẫu dữ liệu huấn luyện. Trước khi đánh trọng số cho các từ khoá cần tiến hành loại bỏ các từ dừng. Từ dừng là những từ thường xuất hiện nhưng không có ích trong việc đánh chỉ mục, nó không có ý nghĩa gì trong việc phân loại văn bản. Có thể nêu một số từ dừng trong tiếng Việt như “và”, “là”, “thì”, “như vậy”, …, trong tiếng Anh như “and”, “or”, “the”,…Thông thường từ dừng là các trạng từ, liên từ, giới từ.

*1.3.3. Phân loại văn bản*

Phân loại văn bản là quá trình phân tích và gán một văn bản vào một hay nhiều lớp cho trước nhờ một mô hình phân loại. Mô hình phân loại này được xây dựng dựa trên một tập hợp các văn bản đã gán nhãn từ trước (đã xác định tên chủ đề trước) gọi là tập dữ liệu huấn luyện. Tập dữ liệu huấn luyện là tập các trang văn bản đã gán nhãn lớp tương ứng từng chủ đề. Quán trình xây dựng tập dữ liệu huấn luyện này thường được thực hiện bằng con người. Sau đó, mô hình được sử dụng để phân loại các trang văn bản chưa gán nhãn.

Bộ phân lớp có thể được xây dựng bằng tay dựa vào các kỹ thuật ứng dụng tri thức (thường là xây dựng một tập các tri thức) hoặc có thể được xây dựng một cách tự động bằng các kỹ thuật học máy thông qua một tập các dữ liệu huấn luyện được định nghĩa trước phân lớp tương ứng. Trong hướng tiếp cận học máy, ta chú ý đến các vấn đề sau:

• Biểu diễn văn bản: Một văn bản thông thường được biểu diễn bằng một véc tơ trọng số, độ dài của véc tơ là số các từ khóa xuất hiện trong ít nhất một mẫu dữ liệu huấn luyện. Biểu diễn trọng số có thể là nhị phân (từ khóa đó có hay không xuất hiện trong văn bản tương ứng) hoặc không nhị phân (từ khóa đó đóng góp tỷ trọng bao nhiêu cho ngữ nghĩa văn bản).

• Loại bỏ các từ dừng và lấy từ gốc: Trước khi đánh trọng số cho các từ khoá cần tiến hành loại bỏ các từ dừng (stop-word). Từ điển Wikipedia định nghĩa: “Từ dừng là những từ xuất hiện thường xuyên nhưng lại không có ích trong đánh chỉ mục cũng như sử dụng trong các máy tìm kiếm hoặc các chỉ mục tìm kiếm khác”.

Thông thường, các trạng từ, giới từ, liên từ là các từ dừng. Tuy nhiên, có thể liệt kê danh sách các từ dừng cho tiếng Việt mặc dù có thể là không đầy đủ. Việc lấy từ gốc và lưu lại các từ phát sinh từ mỗi từ gốc để nâng cao khả năng tìm kiếm được áp dụng cho các ngôn ngữ tự nhiên có chia từ.

• Tiêu chuẩn đánh giá: Phân loại văn bản được coi là không mang tính khách quan theo nghĩa dù con người hay bộ phân loại tự động thực hiện việc phân loại thì đều có thể xảy ra sai sót. Tính đa nghĩa của ngôn ngữ tự nhiên, sự phức tạp của bài toán phân loại được coi là những nguyên nhân điển hình nhất của sai sót phân loại. Hiệu quả của bộ phân loại thường được đánh giá qua so sánh quyết định của bộ phân loại đó với quyết định của con người khi tiến hành trên một tập kiểm thử (test set) các văn bản đã gán nhãn lớp trước.

Có nhiều bài toán phân loại văn bản như: phân lớp nhị phân (chỉ cần xác định một văn bản có thuộc một lớp cho trước hay không), phân lớp đa lớp (một văn bản thuộc một lớp nào đó trong danh sách các lớp cho trước), phân lớp đa trị (một văn bản có thể thuộc nhiều hơn một lớp trong danh sách các lớp cho trước).

Diagram

Description automatically generated

Hình 1.3. Mô hình tổng quát của Phân loại văn bản

*1.3.4. Các bước phân loại*

Để tiến hành phân loại văn bản nói chung, ta thực hiện các bước sau:

Bước 1: Xây dựng bộ dữ liệu chủ quan dựa vào tài liệu văn bản đã được phân loại sẵn. tiến hành học cho bộ dữ liệu, xử lý và thu thập được dữ liệu của quá trình học là các đặc trưng riêng biệt cho từng chủ đề.

Bước 2: Dữ liệu cần phân loại được xử lý, rút ra đặc trưng kết hợp với đặc trưng được học trước đó để phân loại và rút ra kết quả.

Dữ liệu đầu vào cho quá trình học máy hay dữ liệu đầu vào để phân loại đều là dạng văn bản đã qua công đoạn tiền xử lí. Công đoạn tiền xử lí này rất quan trọng và cần thiết, nó làm tối ưu hóa dữ liệu trong việc lưu trữ và xử lí. Các công đoạn trong quá trình tiền xử lí văn bản bao gồm: tách từ tiếng Việt, loại bỏ các từ dừng, từ tầm thường. Sau đó, rút trích đặc trưng và biểu diễn văn bản.

# **Chương 2: Giải pháp phát hiện tin giả dùng học sâu**

## **2.1. Tin giả và phân loại tin giả**

Các ví dụ điển hình của tin giả bao gồm quảng cáo lừa dối (trong kinh doanh và chính trị), tuyên truyền của chính phủ, hình ảnh gốc được chỉnh sửa hoặc sử dụng sai mục đích, tài liệu giả mạo, bản đồ giả mạo, gian lận trên Internet, trang web giả mạo và mục nhập Wikipedia không đúng sự thật, v.v. Tin giả có thể gây ra thiệt hại đáng kể nếu nhiều người tin vào nó. Để giải quyết mối đe dọa này đối với chất lượng thông tin, trước tiên chúng ta cần hiểu chính xác các loại tin giả.

Có nhiều nghiên cứu về phân loại tin giả và tin giả, một trong những báo cáo được tham khảo và trích dẫn nhiều nhất về phân loại tin giả là của Claire Wardle và Hossein Derakhshan. Do đó, họ đưa ra một khung khái niệm mới để kiểm tra tình trạng rối loạn thông tin, xác định ba loại khác nhau: Mis-information, Dis-information và Mal-information. Sử dụng các khía cạnh của tác hại và sai lệch, có thể mô tả sự khác biệt giữa ba loại thông tin sau:

* Mis-information là khi thông tin sai lệch được chia sẻ nhưng không gây hại.
* Dis-information là khi thông tin sai lệch được cố ý chia sẻ để gây hại.
* Mal-information là khi thông tin xác thực được chia sẻ để gây hại, thường là phát tán thông tin một cách thầm lặng nhưng cho nhiều người.

Diagram, venn diagram

Description automatically generated

Hình 2.1. Phân biệt các loại tin giả

### *2.1.1. Tin sai (mis-information)*

Những thông tin sai lệch một cách tự nhiên, không chủ đích của người viết, thường đến từ việc hiểu sai vấn đề, không rà soát lỗi chính tả, không kiểm tra lại thông tin.

Trên truyền thông, ta có thể bắt gặp tin sai ở các dạng lỗi sai phạm, tắc trách hay định kiến vô thức của người đưa tin. Đó có thể là lỗi viết sai tên đối tượng của một bài báo, lỗi viết nhầm ngày, địa điểm trong một văn kiện, hay việc chia sẻ một tin sai sự thật với mục đích giúp đỡ nhưng không kiểm chứng nội dung được chia sẻ.

### *2.1.2 Tin xuyên tạc, tin dắt mũi (dis-information)*

Tin xuyên tạc, hay dắt mũi là các loại thông tin cố ý đưa sai sự thật với nhiều kỹ thuật chỉnh sửa hình ảnh, nội dung, số liệu, ngữ cảnh, văn phong… nhằm đạt được mục đích lợi ích cụ thể và để lại những hậu quả nghiêm trọng.

Theo bài đăng trên tạp chí Tia Sáng, trong thực tế, tin xuyên tạc/dắt mũi còn có các phân loại nhỏ hơn. Các phân loại đó là:

* Lời nói dối gây hại (malicious lies): Đây là thông tin sai lệch được tạo ra với mục đích hãm hại một đối tượng cụ thể, nhằm mang lại lợi ích thương mại, cá nhân, hoặc chính trị cho đối tượng lan truyền thông tin. Ví dụ: Lời đàm tiếu vô căn cứ, tin đồn thất thiệt, đả kích các đối tượng chính trị mà không có căn cứ…
* Thông tin, hình ảnh đánh lạc hướng (visual dis-information): Hình ảnh gây hiểu lầm với chức năng tạo dựng câu chuyện sai lệch. Ví dụ: Hình ảnh bị chỉnh sửa, bản đồ giả, hoạ đồ, đồ thị thiếu chuẩn khoa học nhằm tạo nên đánh giá thông tin sai…
* Tin dắt mũi/xuyên tạc đúng (true dis-information): Tin dắt mũi mang tính chất đúng, hoặc nhấn mạnh một phần sự thật, nhưng lại gây hiểu lầm hoặc hiểu sai. Ví dụ: Các bài báo ‘giật tít’ nhằm thu hút chú ý của người xem tin với tiêu đề gây hiểu lầm, các hình thức PR quan hệ công chúng mang tính ‘nhào nặn’ sự thật theo hướng có lợi cho cơ quan và doanh nghiệp.
* Tin dắt mũi/xuyên tạc gây hiệu ứng phụ (side-effect dis-information): Thông tin gây hiểu sai, dù người đưa tin không cố ý lừa lọc người nhận thông tin. Một ví dụ thực tế là trong một nghiên cứu nổi tiếng vào năm 2004, G.S. Halavais của Đại Học Arizona State University thử nghiệm chèn thông tin sai lệch vào các bài viết trên Wikipedia và đo lượng thời gian các thông tin này được phát hiện ra bởi các biên tập viên trên trang này. Thông tin sai lệch đó có thể được xem là tin dắt mũi/xuyên tạc gây hiệu ứng phụ vì G.S. Halvais không cố ý đưa tin giả nhằm lừa lọc người đọc trên Wikipedia.
* Tin dắt mũi/xuyên tạc mang tính thích nghi (Adaptive dis-information): Thông tin sai lệch mang lại lợi ích có hệ thống cho người đưa tin hoặc một đối tượng chính trị xã hội nào đó. Ở các quốc gia dân chủ, truyền thông đại chúng phân chia theo truyền thông và truyền thông cánh hữu; kênh Fox News ở Mỹ thường đưa tin sai lệch theo cách có lợi cho đảng Cộng Hoà (cánh hữu) ở Mỹ. Thông tin sai lệch đó là ví dụ cho tin dắt mũi/xuyên tạc mang tính thích nghi.
* Tin dắt mũi/xuyên tạc vị tha (altruistic dis-information): Thông tin sai lệch với mục đích mang lại lợi ích cho đối tượng tiếp nhận thông tin. Ví dụ: Bác sĩ nói dối bệnh nhân về tình trạng bệnh nhằm giúp bệnh nhân lạc quan hơn, chính phủ lượt bỏ chi tiết về khủng hoảng chính trị, kinh tế, xã hội, nhằm ngăn ngừa hoảng loạn đám đông.
* Tin dắt mũi/xuyên tạc gây bất lợi (detrimental disinformation): Thông tin sai lệch được đưa ra với mục đích cứu vãn một tình thế khác. Ví dụ: Bệnh nhân ngại đưa ra thông tin đúng về sức khỏe của mình (chế độ dinh dưỡng, chế độ tập thể thao, loại thuốc họ đang sử dụng) cho bác sĩ, cung cấp thông tin sai lệch và có thể gây hại cho bản thân.

### *2.1.3 Tin nguy hại (mal-information)*

Đây là thông tin dựa trên hiệu thực nhưng được dùng để gây hại cho một cá nhân, tổ chức hay quốc gia.

Một số ví dụ cho tin nguy hại là tin tiết lộ thiên hướng tính dục của một cá nhân với mục đích phỉ báng, tin tiết lộ đời tư cá nhân của người nổi tiếng không có sự cho phép của người đó, hay tin công kích một đối tượng vì phẫu thuật chuyển giới.

Ta cần phân biệt các thông điệp đúng sự thật với thông điệp không đúng sự thật, những cũng cần phân biệt thông tin đúng sự thật (hoặc chứa một phần sự thật) nhưng lại được sáng tạo, sản xuất hay phân phối bởi “những tác nhân” có ý đồ hủy hoại hơn là phục vụ lợi ích công. Những tin nguy hại như thế - như thông tin có thật nhưng lại xâm hại đến sự riêng tư của một cá nhân mà không mang lại lợi ích công gì - đi ngược lại chuẩn mực và đạo đức của báo chí.

## **2.2. Học máy nói chung và học sâu nói riêng**

### *2.2.1. Học máy*

Trước khi đến với học sâu chúng ta hãy nhìn sơ lại về học máy. Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc phát triển các kỹ thuật cho phép các máy tính có thể "học". Học máy là lĩnh vực liên quan nhiều đến thống kê do cả hai lĩnh vực đều tập trung vào việc nghiên cứu để phân tích dữ liệu. Tuy nhiên, học máy có sự khác biệt với thống kê, học máy tập trung vào nghiên cứu sự phức tạp của các giải thuật trong quá trình tính toán, xử lý dữ liệu. Trên thực tế, có nhiều bài toán suy luận được xếp loại là bài toán NP- khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ để có thể xử lý được lớp các bài toán nhị phân một cách tổng quát nhất.

Trên cơ sở đó, người ta phân loại học máy theo hai dạng:

* Học máy dựa trên quy nạp: Máy học phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều, sẵn có.
* Học máy dựa trên suy diễn: Máy học phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ học máy

Học máy có nhiều phương pháp khác nhau, như:

* Học có giám sát
* Học không giám sát
* Học bán giám sát
* Học tăng cường
* Học sâu

### *2.2.2. Mạng nơ-ron*

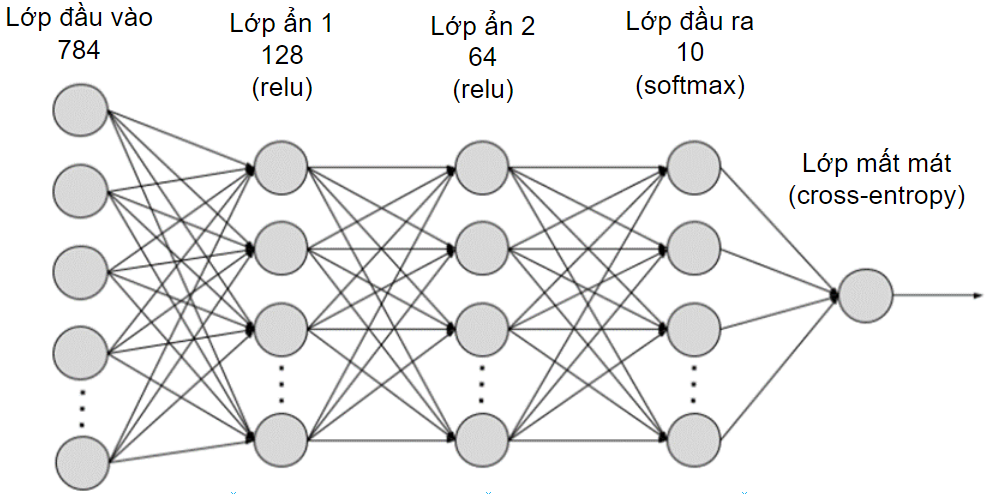
Mạng nơ-ron (Neural Networks) – hay cụ thể hơn là mạng nơ-ron nhân tạo (Artificial Neural Networks ANNs) – bắt chước bộ não con người thông qua một tập hợp các thuật toán. Ở cấp độ cơ bản, mạng nơ-ron có bốn thành phần chính: đầu vào (inputs), weights, bias hoặc threshold và đầu ra (outputs). Tương tự như hồi quy tuyến tính (linear regression), công thức đại số sẽ giống như sau:

Một mạng nơ-ron cơ bản bao gồm các nơ-ron nhân tạo liên kết theo 3 lớp:

* Lớp đầu vào: Thông tin từ thế giới bên ngoài đi vào mạng nơ-ron nhân tạo qua lớp đầu vào. Các nút đầu vào xử lý dữ liệu, phân tích hoặc phân loại và sau đó chuyển dữ liệu sang lớp tiếp theo.
* Lớp ẩn: Dữ liệu đi vào lớp ẩn đến từ lớp đầu vào hoặc các lớp ẩn khác. Mạng nơ-ron nhân tạo có thể có một số lượng lớn lớp ẩn. Mỗi lớp ẩn phân tích dữ liệu đầu ra từ lớp trước, xử lý dữ liệu đó sâu hơn và rồi chuyển dữ liệu sang lớp tiếp theo.
* Lớp đầu ra: Lớp đầu ra cho ra kết quả cuối cùng của tất cả dữ liệu được xử lý bởi mạng nơ-ron nhân tạo. Lớp này có thể có một hoặc nhiều nút. Ví dụ: giả sử chúng ta gặp phải một vấn đề phân loại nhị phân (có/không), lớp đầu ra sẽ có một nút đầu ra, nút này sẽ cho kết quả 1 hoặc 0. Tuy nhiên, nếu chúng ta gặp phải vấn đề phân loại nhiều lớp, lớp đầu ra sẽ có thể bao gồm nhiều hơn một nút đầu ra.

Mạng nơ-ron chuyên sâu, hoặc mạng deep learning, có nhiều lớp ẩn với hàng triệu nơ-ron nhân tạo liên kết với nhau. Một con số, có tên gọi là trọng số, đại diện cho các kết nối giữa hai nút. Trọng số sẽ dương nếu một nút kích thích nút còn lại, hoặc âm nếu một nút ngăn cản nút còn lại. Các nút với trọng số cao hơn sẽ có ảnh hưởng lớn hơn lên các nút khác.

Về mặt lý thuyết, mạng nơ-ron chuyên sâu có thể ánh xạ bất kỳ loại dữ liệu đầu vào với bất kỳ loại dữ liệu đầu ra nào. Tuy nhiên, chúng cũng cần được đào tạo hơn rất nhiều so với các phương pháp máy học khác. Chúng cần hàng triệu ví dụ về dữ liệu đào tạo thay vì hàng trăm hoặc hàng nghìn ví dụ mà một mạng đơn giản hơn thường cần.



Hình 2.2. Mô hình Mạng nơ-ron

### *2.2.3. Học sâu*

Học sâu (Deep learning) được bắt nguồn từ thuật toán Neural network vốn xuất phát chỉ là một ngành nhỏ của học máy (Machine learning). Học sâu là một chi của ngành máy học dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 15 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn, phim, ảnh, âm nhạc.

Chart

Description automatically generated with medium confidence

Hình 2.3. Mối quan hệ giữa Mạng nơ-ron và Học sâu

Deep learning gồm rất nhiều thuật toán và mỗi thuật toán có ứng dụng riêng tùy vào bài toán:

* Linear Regression
* Logistic Regresstion
* Decision Tree and Random Forest
* Naive Bayes
* Support Vector Machines
* K-Nearest Neighbors
* Principal component analysis (PCA)
* Neural network
* .....

Deep Learning cho phép chúng ta huấn luyện một AI có thể dự đoán được các đầu ra dựa vào một tập các đầu vào. Cả hai phương pháp **có giám sá**t và **không giám sát** đều có thể sử dụng để huấn luyện.

Khi kết thúc huấn luyện, một hệ thống Deep Learning sẽ có thể đưa ra dự đoán gần như chính xác khi được cung cấp đủ dữ liệu.

*2.2.4. Phân loại học sâu*

1. Học sâu dành cho phương pháp học không giám sát (unsupervised) hoặc mang tính tổng quát (generative learning) nhằm nắm bắt mối tương quan bậc cao của dữ liệu quan sát hoặc nhìn thấy được cho các mục đích phân tích hoặc tổng hợp mẫu khi không có thông tin về nhãn lớp đích. Đề cập đến tính năng chưa được truyền bá hoặc việc học đại diện trong tài liệu cho danh mục này của các mạng sâu. Khi được sử dụng trong chế độ chi tiết chung, cũng có thể nhằm đặc trưng cho thống kê chung phân phối dữ liệu hiển thị và các lớp liên quan của chúng khi có sẵn và được coi là một phần của dữ liệu hiển thị. Trong trường hợp thứ hai, việc sử dụng quy tắc Bayes có thể biến loại mạng thành một mạng phân biệt để học. Một vài phương pháp có thể kể đến như DBN (Deep belief network), BM (Boltzmann machine), RBM (Restricted Boltzmann machine), DNN (Deep neural network),…
2. Học sâu dành cho phương pháp học có giám sát (supervised) nhằm mục đích cung cấp khả năng phân biệt cho các mục đích phân loại mẫu, thường bằng cách mô tả đặc điểm của các phân phối sau của các lớp được điều kiện dựa trên dữ liệu hiển thị. Dữ liệu nhãn mục tiêu là luôn có sẵn dưới các hình thức trực tiếp hoặc gián tiếp để được giám sát như vậy học tập. Chúng còn được gọi là mạng sâu phân biệt đối xử. Một vài phương pháp có thể kể đến như RNN (Recurrent neural network), CNN (Convolutional neural network), HMM (Hidden Markov Model), CRF (Conditional random fields), TDNN (Time delay neural network),…
3. Học sâu hơn trong đó mục tiêu là phân biệt đối xử được đẩy lên, theo một cách đáng kể, với kết quả của các mạng lưới sâu tổng quát hoặc không được giám sát. Điều này có thể được thực hiện bằng tối ưu hóa tốt hơn hoặc/và chính quy hóa các mạng sâu trong học sau có giám sát (2). Mục tiêu cũng có thể đạt được khi các tiêu chí không phân biệt đối xử đối với việc học có giám sát được sử dụng để ước tính các thông số trong bất kỳ tầng sâu tạo ra hoặc không được giám sát mạng trong học sâu tổng quát (1) ở trên. Các phương pháp của loại này thường là của hai loại trước (1) và (2) nhưng được sửa đổi để phù hợp với khái niệm trên. Ví dụ như DBN, một mạng sâu của học không giám sát được chuyển đổi và sử dụng làm mô hình ban đầu của DNN để học có giám sát với cùng cấu trúc mạng, được đào tạo riêng biệt hơn nữa hoặc được tinh chỉnh bằng các nhãn được cung cấp. Khi DBN được sử dụng theo cách này được gọi là mô hình DBN-DNN.

## **2.3. Giải pháp phân loại tin giả dùng học máy và học sâu**

### *2.3.1. Tổng quan vấn đề và giải pháp*

Việc phát hiện tin tức giả một cách thủ công thường bao gồm tất cả các kỹ thuật và quy trình mà con người có thể sử dụng để xác minh tin tức. Tuy nhiên, lượng dữ liệu trực tuyến được tạo ra hàng ngày hiện nay là quá nhiều để có thể phân loại bằng phương pháp thủ công. Hơn nữa, thông tin lan truyền trực tuyến quá nhanh khiến việc kiểm tra thủ công nhanh chóng trở nên không hiệu quả và không thực tế. Kiểm tra thủ công gặp khó khăn lớn nhất khi mở rộng quy mô xác minh do khối lượng lớn dữ liệu được tạo ra và nhanh chóng. Vì vậy, nhiệm vụ tự động phát hiện tin giả là một nhu cầu cấp thiết và quan trọng.

Hệ thống phát hiện tin tức giả mạo tự động sẽ giúp xác minh một tin tức là giả hay thật mà không cần sự can thiệp trực tiếp của con người. Có nhiều kỹ thuật và cách tiếp cận khác nhau được sử dụng trong nghiên cứu phát hiện tin tức giả. Các kỹ thuật và cách tiếp cận này phụ thuộc vào quan điểm và mục tiêu theo dõi của nhà phát triển.

Trong phạm vi của đề tài này,em trình bày giải pháp đề xuất để phát hiện tin giả trên các bài báo viết bằng tiếng Việt và dựa trên kỹ thuật phân loại văn bản bằng học sâu, cụ thể là mô hình RNN.

### *2.3.2. Mô hình tổng quát*

Mô hình chung của cách tiếp cận này như sau:

* Bước đầu tiên trong mô hình này là giai đoạn thu thập dữ liệu để xây dựng bộ dữ liệu đào tạo. Bộ dữ liệu này bao gồm những tin tức đã được dán nhãn là giả hoặc thật. Trong trường hợp học có giám sát, tất cả dữ liệu được sử dụng cho đào tạo phải được gắn nhãn, trong trường hợp học bán giám sát, cả dữ liệu được gắn nhãn và không được gắn nhãn.
* Giai đoạn tiền xử lý cho phép các kỹ thuật xử lý ngôn ngữ tự nhiên được sử dụng để làm sạch dữ liệu, loại bỏ thông tin không hữu ích và đại diện cho dữ liệu.
* Giai đoạn đào tạo cho phép trích xuất các đặc điểm ngôn ngữ cần thiết để tạo ra các mô hình phân loại và nhận dạng nội dung. Trên cơ sở các đặc trưng được trích xuất, thực hiện huấn luyện theo các thuật toán lựa chọn để xây dựng mô hình đặc trưng. Mô hình này sẽ được sử dụng để dự đoán xem một bản tin là giả hay thật.
* Giai đoạn dự đoán có chức năng so sánh các đặc điểm của tin cần xác minh với mô hình đặc trưng được tạo trong giai đoạn huấn luyện để quyết định tin là giả hay thật.

Diagram

Description automatically generated

Hình 2.4. Mô hình tổng quát

### *2.3.3. Biểu diễn văn bản bằng BoW*

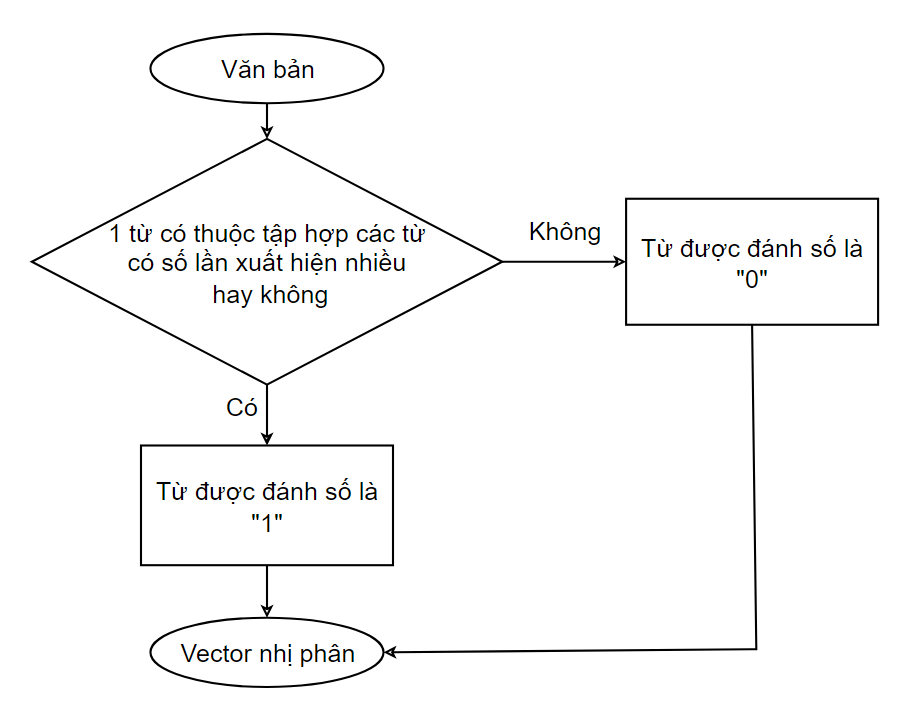
Mô hình BoW (Bag-of-words) là một biểu diễn đơn giản hóa được sử dụng trong xử lý ngôn ngữ tự nhiên và truy xuất thông tin. Trong mô hình này, một tài liệu văn bản được biểu diễn như thể nó là một túi các từ của nó, không tính đến ngữ pháp và trật tự từ mà chỉ giữ tần suất xuất hiện của mỗi từ trong tài liệu. Mô hình Bag-of-words thường được sử dụng trong các phương pháp phân loại tài liệu, trong đó sự xuất hiện của mỗi từ được sử dụng như một đặc điểm để đào tạo một trình phân loại.

Diagram, shape

Description automatically generated with medium confidence

Hình 2.5. Bag-of-words

Khi mô hình này được áp dụng để biểu diễn trong văn bản, mỗi từ được biểu thị một số nhị phân phụ thuộc vào việc từ này có thuộc tập hợp các từ có tần số cao hay không. Kết quả là, văn bản đầu vào được biểu diễn bằng vectơ nhị phân. Thuật toán xác định đặc trưng nhị phân của văn bản được trình bày trong Hình 2.6.



Hình 2.6. Cách xây dựng nên vector nhị phân của Bag-of-words

Hãy xem xét một ví dụ với một tài liệu ngắn: “Strategy Analytics nhận định trong khi Apple có màn hình tương phản nhạt về việc tung ra iPhone X, thì Samsung đã thành công vì đa dạng hóa danh mục điện thoại thông minh của mình, bao gồm dòng Galaxy S hàng đầu và điện thoại thông minh Galaxy A tầm trung. ”. Với giả thiết rằng tập các từ có tần số cao trong dữ liệu bao gồm {điện, điện thoại, thông, minh, Samsung, Apple, Galaxy, iPhone}, thì đặc trưng nhị phân của tài liệu là [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0 0]

### *2.3.4. Biểu diễn văn bản bằng TF-IDF*

Trong phần trước em đã nói qua về Bag-of-words để tạo các vectơ đặc trưng mã hóa có hay không một từ từ tập từ vựng - tập hợp các từ có tần suất cao. Các vectơ đặc trưng này không mã hóa ngữ pháp, thứ tự từ hoặc tần số của từ. Trực quan là tần suất mà một từ xuất hiện trong tài liệu có thể cho biết mức độ liên quan của tài liệu với từ đó. Một tài liệu dài chứa một lần xuất hiện của một từ có thể thảo luận về một chủ đề hoàn toàn khác với một tài liệu chứa nhiều lần xuất hiện của cùng một từ. Trong phần này, em sẽ tạo các vectơ đặc trưng mã hóa tần số của các từ bằng trọng số TF-IDF thay vì sử dụng giá trị nhị phân cho mỗi phần tử trong vector đặc trưng như phần trước.

Trong truy xuất thông tin, TF-IDF (Term Frequency-Inverse Document Frequency), là một thống kê số nhằm phản ánh tầm quan trọng của một từ đối với tài liệu trong một bộ sưu tập hoặc kho ngữ liệu. Nó thường được sử dụng như một yếu tố trọng số trong các tìm kiếm về truy xuất thông tin, khai thác văn bản và mô hình hóa người dùng. Giá trị TF-IDF tăng tỷ lệ thuận với số lần một từ xuất hiện trong tài liệu, nhưng thường được bù đắp bởi tần suất xuất hiện của từ đó trong kho ngữ liệu, điều này giúp điều chỉnh thực tế là một số từ nói chung xuất hiện thường xuyên hơn.

TF – Term Frequency

TF đơn giản chỉ là số lần xuất hiện của từ trong tài liệu

tf(t,d) = sự xuất hiện của từ *t* trong tài liệu *d.*

IDF – Inverse Document Frequency

IDF thì phức tạp hơn một chút, dùng để đo độ phổ biến hoặc hiếm của một từ trong tập tài liệu qua đó đo được độ quan trọng của từ đó trong tập tài liệu. Khi tính toán TF, tất cả các thuận ngữ đều quan trọng như nhau, nhưng ta biết có các từ chẳng hạn như các giới từ “có”, “của”,… có thể xuất hiện nhiều lần ở nhiều tài liệu khác nhau, nhưng chúng ít quan trọng trong việc biểu diễn nội dung của tài liệu. IDF được tính như sau:

idf(t,D) = ln

Với:

D: tổng số tài liệu trong tập tài liệu

df(d,t): là số tài liệu mà từ t xuất hiện

Tuy nhiên nếu từ t không xuất hiện trong tập tài liệu, sẽ dẫn đến phép chia 0, dẫn đến kết quả của TF-IDF sẽ bằng 0 trong mọi trường hợp. Để ngăn chặn điều này, ta sẽ tính IDF như sau:

idf(t,D) = ln

TFIDF - Term Frequency-Inverse Document Frequency

tf-idf(t,d,D) = tf(t,d)\*idf(t,D)

Sau đây sẽ là một ví dụ nhỏ cách tính TF-IDF:

Ta có tài liệu

d1: Có một con **mèo** trắng đang nằm trên một con **mèo**, chúng nó thật dễ thương.

d2: Có một con **mèo** trắng đang nằm trên một con chó, chúng nó thật dễ thương.

TF(“mèo”,d1) = 2 / 17 TF(“mèo”,d2) = 1 / 17

IDF(“mèo”,D) = ln = 1

Vậy nên

TF-IDF(“mèo”,d1,D) = 2/17\*1 = 2/17 TF-IDF(“mèo”,d2,D) = 1/17\*1 = 1/17

Còn đây là một ví dụ khác:

Hãy xem xét một tài liệu chứa 1000 từ, trong đó từ ‘cat’ và ‘the’ xuất hiện lần lượt 100 lần và 500 lần. Tần suất thuật ngữ cho ‘cat’ và ‘the’ khi đó lần lượt là 100 và 500.

Bây giờ, giả sử chúng ta có 1 triệu tài liệu và từ "cat" xuất hiện ở 1 nghìn trong số này trong khi từ "the" xuất hiện ở 900 nghìn trong số này. IDF được tính như sau:

IDF (‘cat’, D) = ln = 7,91

IDF (‘the’, D) = ln = 1.105

Trọng số TF-IDF là tích của các đại lượng này:

TF-IDF (‘cat’, d, D) = 100 \* 7.91 = 791

TF-IDF (‘the’, d, D) = 500 \* 1.105 = 552,5

Kết quả là TF-IDF (‘cat’, d, D) lớn hơn TF-IDF (‘the’, d, D). Từ ‘cat’ quan trọng hơn từ ‘the’ mặc dù từ ‘the’ xuất hiện thường xuyên.

Các vectơ TF-IDF kết quả sau đó được chuẩn hóa theo chuẩn Euclide:

vnorm = =

### *2.3.5. Phân loại văn bản bằng mô hình Naive Bayes*

Naive Bayes là một thuật toán phân lớp được mô hình hoá dựa trên định lý Bayes trong xác suất thống kê:

* *P(y|X)* gọi là posterior probability: xác suất của mục tiêu *y* với điều kiện có đặc trưng *X*.
* P*(X|y)* gọi là likelihood: xác suất của đặc trưng *X* khi đã biết mục tiêu *y*
* *P(y)* gọi là prior probability: xác suất xảy ra của mục tiêu *y*.
* *P(X)* gọi là prior probability: xác suất xảy ra của đặc trưng *X*.

Ta có thể thay X bằng các vector đặc trưng, được viết dưới dạng:

Khi đó đẳng thức Bayes có thể được viết dưới dạng:

Để hiểu hơn về Naive Bayes, em có một ví dụ một mô hình Naive Bayes đơn giản như sau:

Table

Description automatically generated

Hình 2.7. Dữ liệu ví dụ Naive Bayes

Xác suất xuất hiện của nhãn so1 là

Xác suất xuất hiện của nhãn so1 là

Xác suất xuất hiện từ w trong nhãn so1 là

Trong văn bản 5 ta có 3 từ xuất hiện, nên ta có 3 xác suất của 3 từ trong mỗi loại nhãn là:

Từ đó để xác định nhãn cho văn bản 5, ta sẽ nhân tất cả lại với nhau:

Vậy văn bản số 5 có nhiều khả năng nằm ở loại so1 hơn.

Trong mô hình Naive Bayes, có hai giả thiết được đặt ra:

* + Các đặc trưng đưa vào mô hình là độc lập với nhau. Tức là sự thay đổi giá trị của một đặc trưng không ảnh hưởng đến các đặc trưng còn lại.
  + Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau đối với đầu ra mục tiêu.

Khi đó, kết quả mục tiêu y để P(y|X) đạt cực đại trở thành:

Chính vì hai giả thiết gần như không tồn tại trong thực tế trên, mô hình này mới được gọi là naive (ngây thơ). Tuy nhiên, chính sự đơn giản của nó với việc dự đoán rất nhanh kết quả đầu ra khiến nó được sử dụng rất nhiều trong thực tế trên những bộ dữ liệu lớn, đem lại kết quả khả quan. Một vài ứng dụng của Naive Bayes có thể kể đến như: lọc thư rác, phân loại văn bản, dự đoán sắc thái văn bản, ...

### *2.3.6. Phân loại văn bản bằng mô hình RNN*

Trong phân loại học sâu cho phương pháp học giám sát, có hai thuật toán phổ biến là CNN và RNN, trong khi CNN được sử dụng rất hiệu quả trong việc xử lý ảnh, thì RNN được áp dụng rất nhiều vào các bài toán NLP vì sự hiệu quả trong từng loại mô hình.

Ý tưởng chính của RNN là sử dụng chuỗi thông tin. Trong mạng nơ-ron truyền thống, tất cả các đầu vào và đầu ra đều độc lập với nhau, nghĩa là chúng không bị mắc xích với nhau. Mô hình này không phù hợp với nhiều vấn đề. Ví dụ: nếu chúng ta muốn đoán từ tiếp theo có thể xuất hiện trong một câu, chúng ta cần biết các từ trước đó xuất hiện lần lượt như thế nào (lặp lại) vì chúng thực hiện cùng một nhiệm vụ cho tất cả các phần tử của một chuỗi có đầu ra phụ thuộc vào các tính toán trước đó, nói cách khác, RNN có khả năng ghi nhớ thông tin đã tính toán trước đó. Về lý thuyết, RNN có thể sử dụng thông tin của một văn bản rất dài, nhưng trên thực tế nó chỉ có thể nhớ được một số bước trước đó.

Để hiểu rõ hơn về RNN, ta cùng nhìn lại mô hình nơ-ron ở Hình 2.8:

Diagram

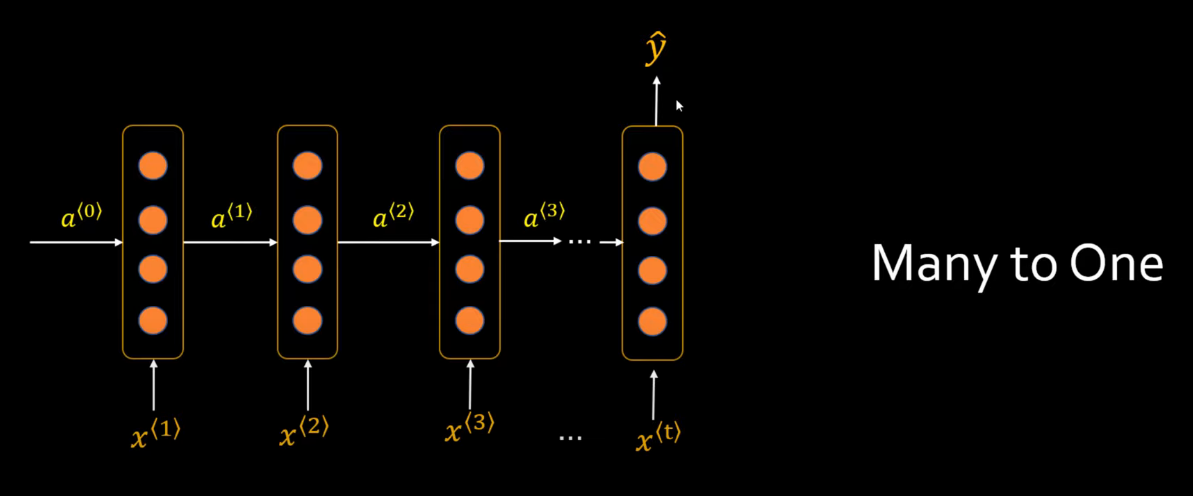
Description automatically generated

Hình 2.8. Mạng nơ-ron thông thường

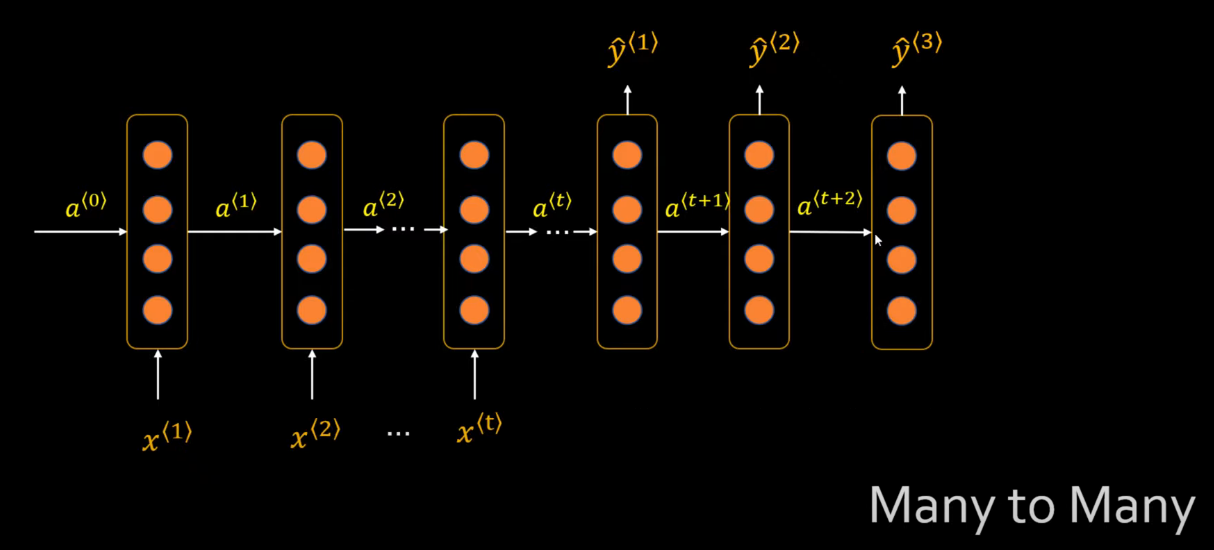
Mạng nơron bao gồm 3 phần chính: Lớp đầu vào, lớp ẩn và lớp đầu ra. Chúng ta có thể thấy rằng đầu vào và đầu ra của mạng nơ-ron này là độc lập với nhau. Vì vậy, mô hình này không phù hợp với các bài toán về chuỗi như mô tả, hoàn thành câu, v.v., bởi vì các dự đoán tiếp theo như từ tiếp theo phụ thuộc vào vị trí của nó trong câu và các từ trước nó.

Vì vậy, RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu trữ thông tin từ các bước tính toán trước đó để dựa vào đó đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại.

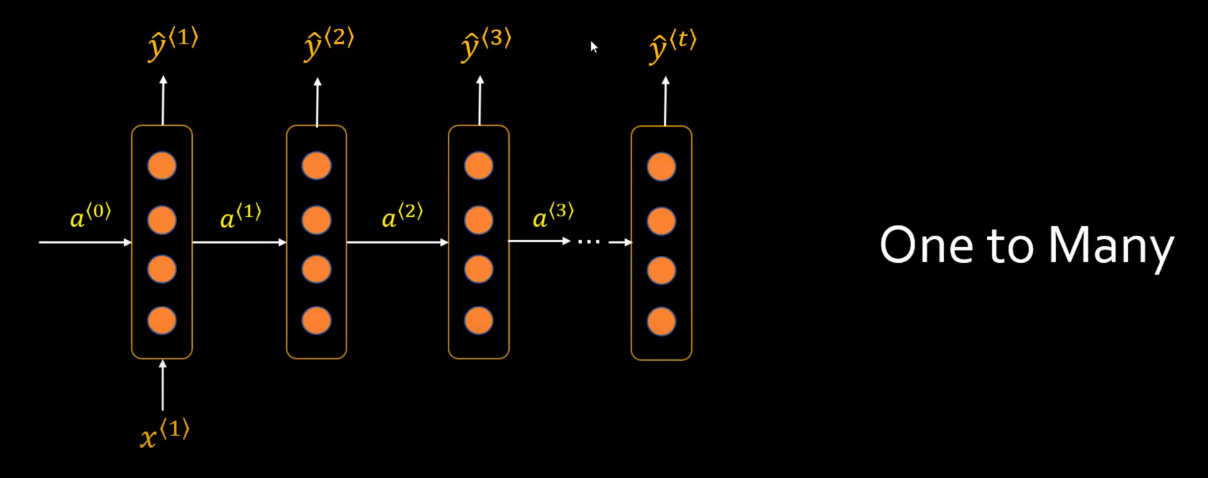
Có 3 dạng mô hình RNN, được mô tả ở các hình Hình 2.9, Hình 2.10, Hình 2.11 dưới đây:



Hình 2.9. Mô hình RNN Many to One



Hình 2.10. Mô hình RNN Many to Many



Hình 2.11. Mô hình One to Many

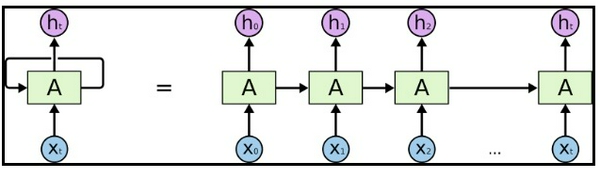
Nếu mạng thần kinh chỉ là lớp đầu vào *x* đi qua lớp ẩn *h* và kết quả là lớp đầu ra *y* với kết nối đầy đủ giữa các lớp. Trong RNN, đầu vào *xt* sẽ được kết hợp với *ht-1* lớp ẩn bởi hàm *fw* để tính toán lớp ẩn hiện tại *ht* và đầu ra *yt* sẽ được tính từ *ht*. *W* là tập hợp các trọng số và nó thu được trong tất cả các cụm, *L1, L2,..., Lt* là các hàm mất mát. Như vậy, kết quả từ các lần tính toán trước đã được “ghi nhớ” bằng cách thêm *ht-1* để tính toán ht nhằm tăng độ chính xác của các dự đoán hiện tại. Cụ thể, quy trình tính toán được viết dưới dạng toán học như sau:

ht = tanh(Whhht-1 + Wxhxt)

yt = Whyht

Lúc này, 3 thứ mới xuất hiện: *Wxh, Whh, Why*. Đối với NN chỉ sử dụng một ma trận trọng số *W*, đối với RNN nó sử dụng 3 ma trận trọng số cho 2 phép tính: Whh kết hợp với "bộ nhớ trước" *ht-1* và Wxh kết hợp với *xt* để tính "bộ nhớ của bước hiện tại" *ht* từ đó kết hợp với *Why* để tính toán *yt*.

Như vậy, RNN là một phương pháp dựa trên nơ-ron chuyên biệt có hiệu quả trong việc xử lý thông tin tuần tự. RNN áp dụng một cách đệ quy một phép tính cho mọi trường hợp của chuỗi đầu vào có điều kiện dựa trên các kết quả đã tính toán trước đó. Các chuỗi này thường được biểu diễn bằng một vectơ có kích thước cố định được cung cấp tuần tự (từng cái một) cho một đơn vị tuần hoàn. Hình 2.12 minh họa một khung RNN đơn giản bên dưới.



Hình 2.12. Một khung RNN đơn giản

Còn Hình 2.13 là một ví dụ về cách một mô hình RNN dự đoán từ tiếp theo trong câu “Việt Nam đẹp lắm ta …”

Diagram

Description automatically generated

Hình 2.13. Ví dụ một mô hình RNN dự đoán từ

Ưu điểm chính của RNN là khả năng ghi nhớ kết quả của các phép tính trước đó và sử dụng thông tin đó trong tính toán hiện tại. Điều này làm cho các mô hình RNN phù hợp với mô hình phụ thuộc ngữ cảnh trong các đầu vào có độ dài tùy ý để tạo ra một bố cục thích hợp của các đầu vào. RNN đã được sử dụng để nghiên cứu các tác vụ NLP khác nhau như dịch máy, chú thích hình ảnh và mô hình ngôn ngữ, trong số những tác vụ khác.

# **Chương 3: Thực nghiệm và đánh giá**

# **3.1. Cài đặt môi trường**

|  |  |
| --- | --- |
| Sklearn | Scikit-learn là một thư viện máy học phần mềm miễn phí cho ngôn ngữ lập trình Python |
| Numpy | Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó |
| Pandas | Pandas là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu |
| Seaborn | Seaborn là một trong những thư viện Python được đánh giá cao nhất thế giới được xây dựng nhằm mục đích tạo ra các hình ảnh trực quan đẹp mắt. Nó có thể được coi là một phần mở rộng của Matplotlib vì nó được xây dựng trên đó. |
| Matplotlib | Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở. |
| Underthesea | Underthesea là một toolkit hỗ trợ cho việc nghiên cứu và phát triển xử lý ngôn ngữ tự nhiên tiếng Việt. Underthesea ra đời vào tháng 3 năm 2017, trong bối cảnh ở Việt Nam đã có một số toolkit khá tốt như vn.vitk, pyvi, nhưng vẫn thiếu một toolkit hoàn chỉnh, mã nguồn mở, dễ dàng cài đặt và sử dụng như các sản phẩm tương đương đối với tiếng Anh như nltk, polyglot, spacy. |
| Tensorflow | TensorFlow chính là thư viện mã nguồn mở cho Machine Learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong Machine Learning và Deep Learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều. |
| WordCloud | Một thư viện dùng để trực quan hóa thông tin |
| BeautifulSoup | BeautifulSoup là thư viện dùng để trích xuất, xử lý thông tin từ file HTML, XML hoặc LXML,… rất hữu ích cho việc tìm kiếm trên web, thường được xử dụng để crawl hoặc tiền xử lý dữ liệu. |

Bảng 3.1. Bảng thông tin của 1 số thư viện được sử dụng

Chương trình được chạy trên ngôn ngữ python3…

Để cài đặt cái thư viện cần thiết. Tất cả đã được tổng hợp lại trong file requirement.txt. Ta có thể bật terminal lên và gõ “*pip install -r requirement.txt –user*”

Và sau đó tiếp tục import các hàm của các thư viện

Text

Description automatically generated

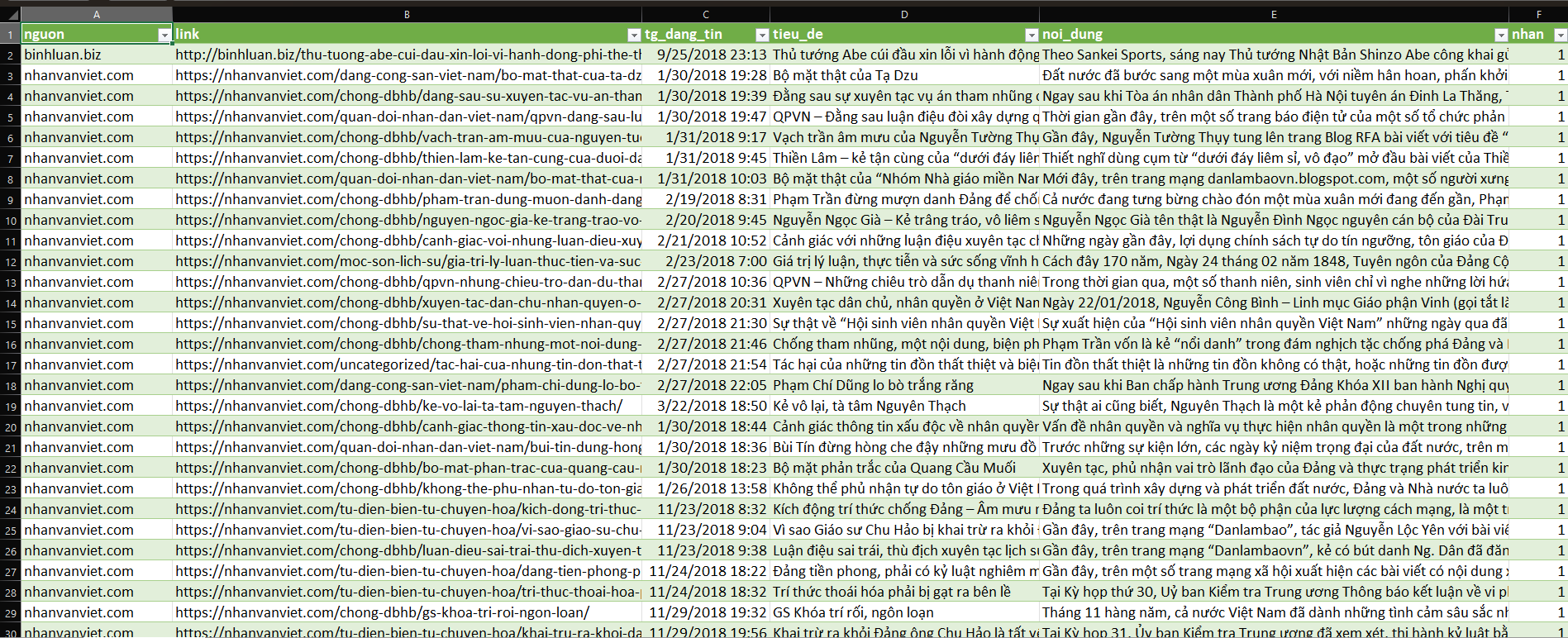
Hình 3.1. Các hàm cần import

# **3.2. Mô tả dữ liệu**

## *3.2.1. Nguồn gốc, đặc tính của dữ liệu*

Dữ liệu là các bài báo chủ đề chính trị được lấy từ các trang báo Tiếng Việt ở các trang uy tín như [thanhnien.vn](https://thanhnien.vn/), [dantri.com.vn](https://dantri.com.vn/), [vnexpress.net](https://vnexpress.net/), [tuoitre.vn](https://tuoitre.vn/), v,v,… và từ các trang không uy tín như viettan.org, thoibao.de, v,v,… Dữ liệu thu thập được có các thông tin như nguồn, link bài báo, thời gian đăng tin, tiêu đề, nội dung và nhãn.

Việc gán nhãn các bài báo được thực hiện thủ công, nhưng hầu hết các bài báo đến từ các trang mạng uy tín đều được gán nhãn 1 (thật), còn đối với các bài báo từ các trang không uy tín thì sẽ được kiểm tra thủ công và đưa ra kết luận. Bởi vì tin giả thì ngay cả con người cũng khó mà phân biệt, nên việc gán nhãn thủ công tốn rất nhiều thời gian và công sức. Vì vậy em chỉ có thể tổng hợp được hơn một nghìn bài báo, cụ thể là 1015 bài báo, trong đó có 515 bài báo được gán nhãn 1 (thật) và 500 bài báo được gán nhãn 0 (giả).



Hình 3.2. Cái nhìn tổng quan về dataset

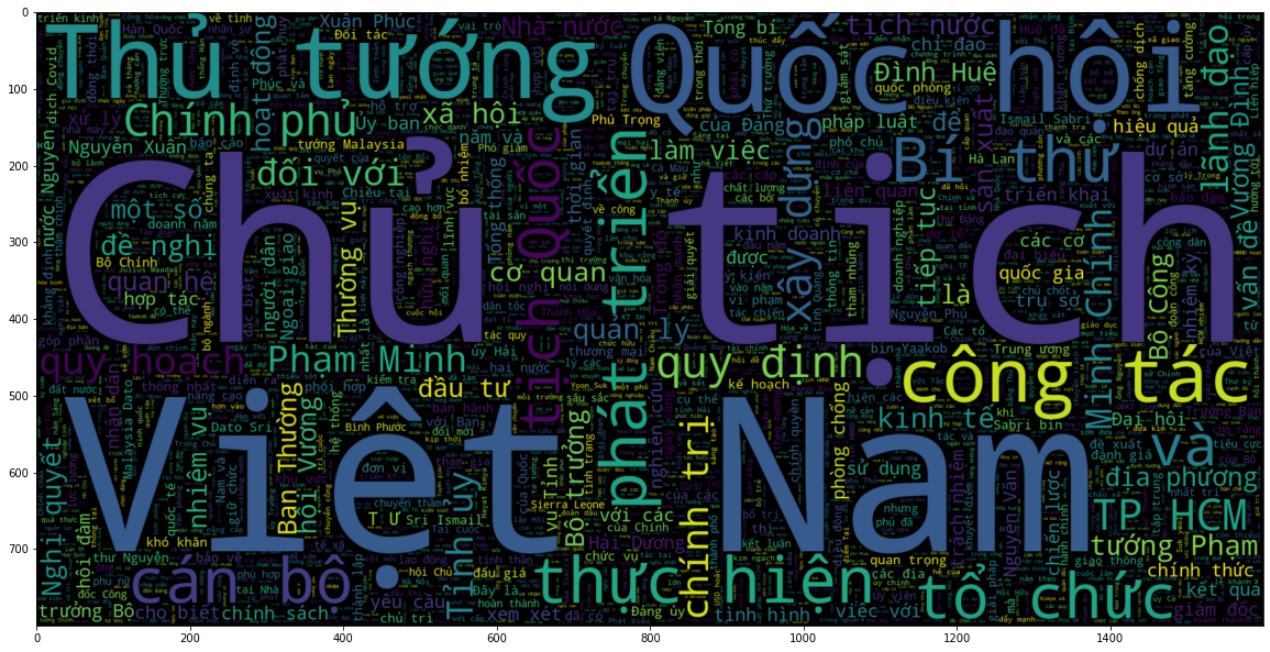
Tỷ lệ tin giả và tin thật trong kho dữ liệu đã thu thập, xấp xỉ 50%

Chart, bar chart, treemap chart

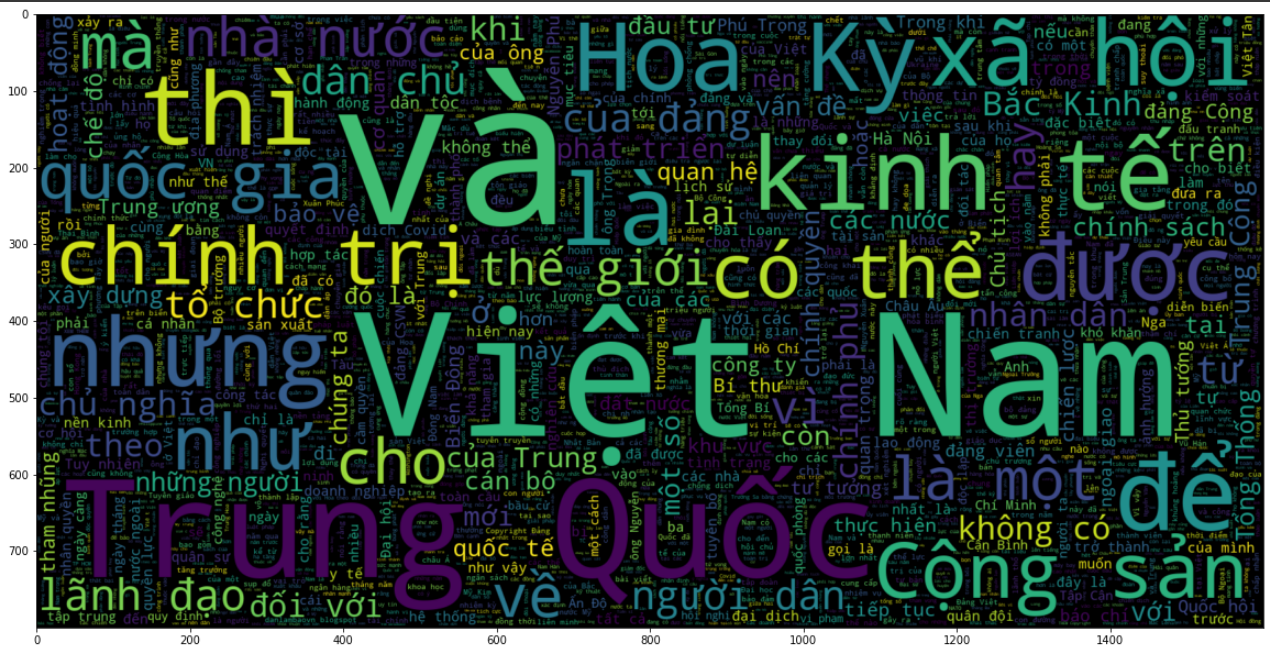
Description automatically generated

Hình 3.3. Số lượng của tin thật và tin giả trong dataset

Các từ xuất hiện nhiều lần trong kho dữ liệu thật và giả, ta có thể thấy có 1 sự tương đồng nhẹ vì các bài báo này đều nói về chung 1 chủ đề đó là chính trị. Tuy nhiên ở các bài báo thật, các từ về các vị trí cao trong nhà nước như “Thủ tướng”, “Quốc hội”, “Chủ tịch”,… lại chiếm ưu thế. Còn ở các bài báo giả thì là các từ về các nước, các đảng khác nhau như “Hoa kì”, “Trung Quốc”, “Cộng sản”,…



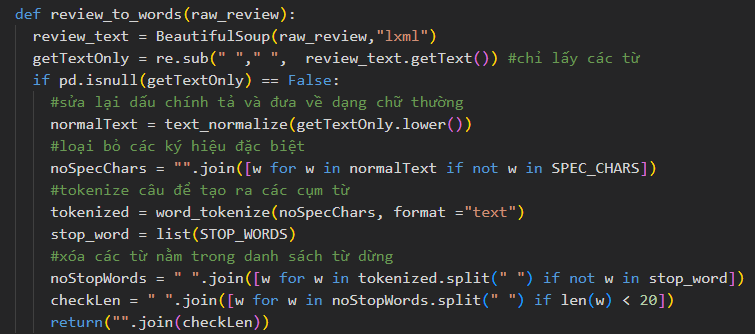
Hình 3.4. Word Cloud của tin thật



Hình 3.5. Word Cloud của tin giả

## *3.2.2. Tiền xử lý dữ liệu*

Đầu tiên ta sẽ loại bỏ các bài báo có giá trị noi\_dung null, ta sẽ còn 1015 bài báo để xử lý.



Hình 3.6. Hàm tiền xử lý dữ liệu

Sau đó sát nhập 2 cột là tiêu đề và nội dung lại, em sử dụng thư viện BeautifulSoup để chuyển dữ liệu từ dạng lxml sang dạng text.

Sau đó sử dụng hàm text\_normalize kết hợp với hàm lower() của str để sửa lại dấu chính tả và đưa tất cả từ về dạng chữ thường (ví dụ **oà** úy thành **òa** úy)

Sau đó tiếp tục em xét duyệt từng kí tự, nếu kí tự nào không thuộc bảng kí tự đặc biệt (SPEC\_CHARS) thì sẽ được thêm vào mảng tạm.

Sử dụng hàm word\_tokenize để tạo ra các cụm từ, từ giữa các cụm nối nhau bằng dấu gạch dưới (\_), còn các từ/cụm từ phân biệt nhau bằng khoảng trắng

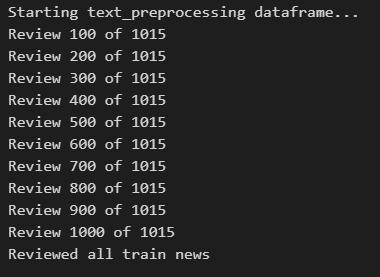
Tiếp tục em sẽ xử lý về các từ dừng, sau khi đã tokenize thành các cụm từ, sử dụng bộ từ dừng đã có sẵn, em xét duyệt từng từ/cụm từ và qua đó loại bỏ được các từ dừng

Nhận thấy có 1 vài bài báo có dính 1 đường link dài, sau khi bỏ kí tự đặc biệt thì sẽ là 1 chuỗi kí tự liền nhau không có nghĩa, em loại bỏ bằng cách chỉ lấy chiều dài tối đa của mỗi từ là 20. Như vậy sau khi tiền xử lý, 1 bài báo có thể sẽ được rút gọn đi 10-20% độ dài, qua đó tăng chất lượng cũng như hiệu quả của các phương pháp phân loại về sau.



Hình 3.7. Ví dụ 1 đoạn văn bản sau khi được xử lý

Sau đó chạy lặp chương trình để tiền xử lý tất cả 1015 bài báo

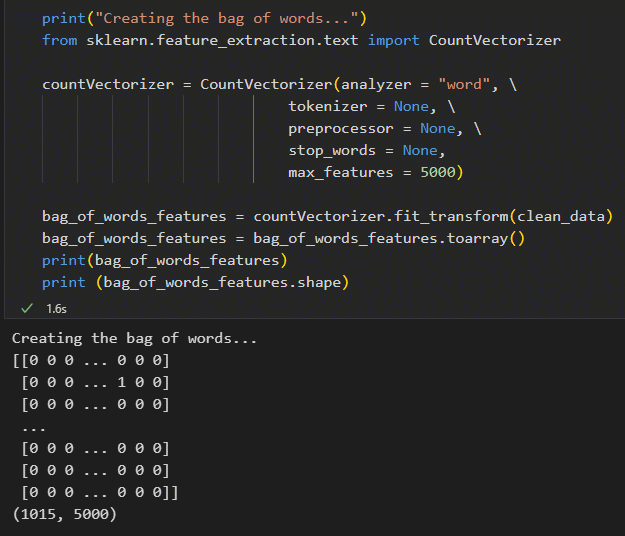


Hình 3.8. Chạy hàm tiền xử lý 1015 bài báo

# **3.3. Bag-of-words và TF-IDF kết hợp với Naive Bayes**

## *3.3.1. Biểu diễn văn bản bằng Bag-of-words*

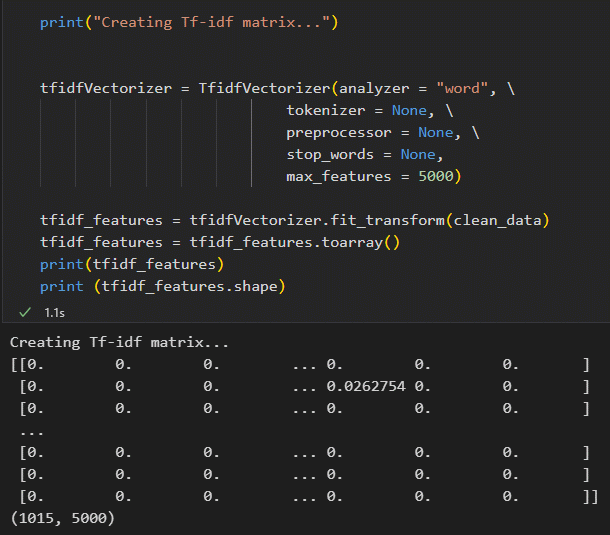
Nhờ vào bước tiền xử lý trước đó, em đã thu được 1 list gồm 1015 bài báo đã được xử lý. Đưa dữ liệu này vào chương trình tạo Bag-of-words của thư viện Sklearn. Ta thu được 1 list 1015 array có 5000 chiều tương ứng với 5000 từ có số lần xuất hiện cao nhất trong tập dữ liệu.



Hình 3.9. Kết quả khi chạy Bag-of-words

## *3.3.2. Biểu diễn văn bản bằng TF-IDF*

Tương tự với Bag-of-words, TF-IDF cũng có cùng số chiều nhưng thay vì hiển thị số lần xuất hiện của từ i trong văn bản D thì TF-IDF sẽ đưa ra 1 số thập phân cho biết được sự quan trọng của từ đó trong tập văn bản D, số càng lớn thì từ đó càng quan trọng.



Hình 3.10. Kết quả khi chạy TF-IDF

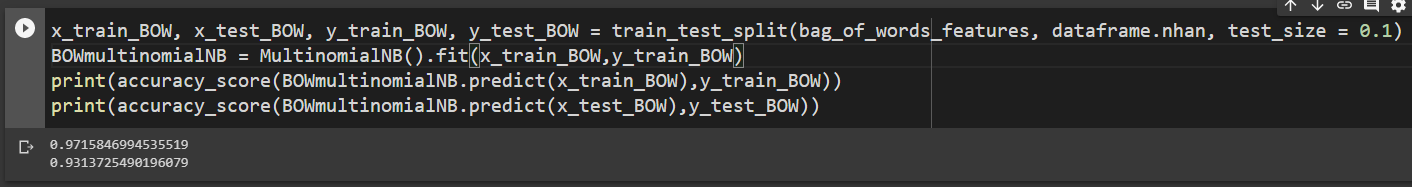
## *3.3.3. Đánh giá*

Sau khi biểu diễn văn bản bằng 2 phương pháp Bag-of-words và TF-IDF. Em nhận thấy việc sử dụng 2 phương pháp này để kết hợp với mô hình RNN là không hiệu quả, vì cả Bag-of-words và TF-IDF đều không quan tâm đến trật tự, thứ tự của các từ xuất hiện trong câu, trong khi mạng hồi qui nơ-ron lại rất quan tâm đến trật tự từ. Đều đó tạo nên sự bất hợp lý trong lúc xây dựng mô hình. Vì lý do đó em sẽ sử dụng Bag-of-words và TF-IDF chung oviws 1 mô hình học máy và sau đó so sánh kết quả với mô hình RNN. Mô hình học máy mà em lựa chọn để kết hợp với Bag-of-words và TF-IDF là Multinomial Naive Bayes.

## *3.3.4. Phân loại bằng mô hình Multinomial Naive Bayes*

Đầu tiên em sẽ chia tập dữ liệu ra làm 2 phần bằng phương pháp random, 1 phần sử dụng để train model, 1 phần để test model với tỷ lệ 9:1 bằng cách sử dụng hàm train\_test\_split của thư viện nltk.

Mô hình Naive Bayes kết hợp với Bag-of-words:



Hình 3.11. Kết quả mô hình NB kết hợp BoW

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lần 1 | Lần 2 | Lần 3 | Lần 4 | Lần 5 |
| Độ chính xác trên tập train | 96,17% | 95,62% | 96,06% | 96,39% | 95,95% |
| Độ chính xác trên tập test | 95,09% | 97,05% | 97,05% | 94,11% | 96,07% |

Bảng 3.2 .Kết quả sau nhiều lần chạy thử NB kết hợp BoW

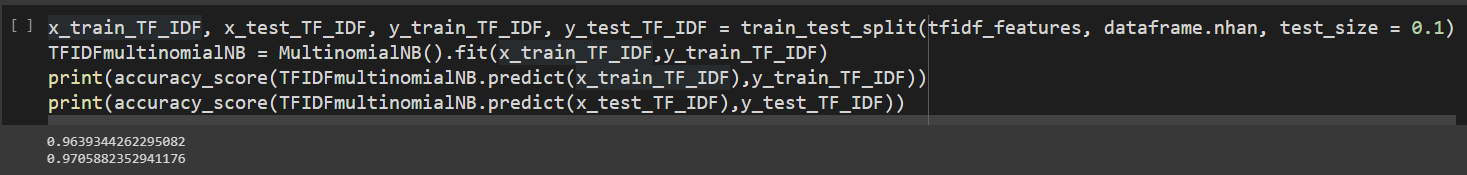
Tỷ lệ chính xác của mô hình Bag-of-words kết hợp Naive Bayes xấp xỉ 96,01%

~~Graphical user interface, application

Description automatically generated~~

Hình 3. 12. Ma trận nhẫm lẫn của mô hình BoW kết hợp với NB

Mô hình Naive Bayes kết hợp với TF-IDF:

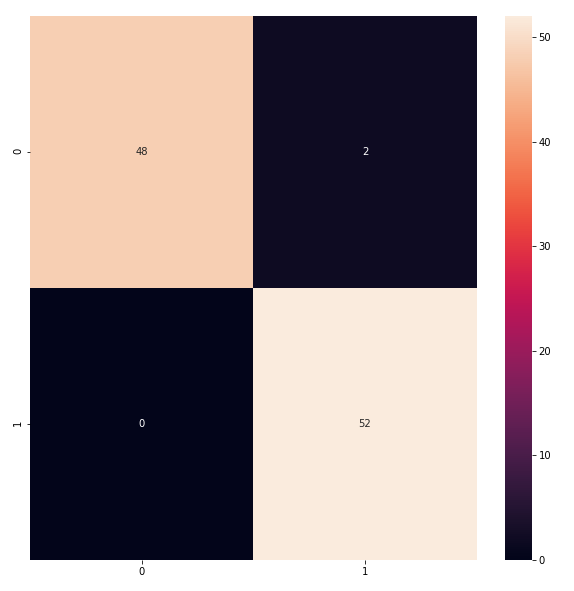


Hình 3.13. Kết quả mô hình NB kết hợp TF-IDF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lần 1 | Lần 2 | Lần 3 | Lần 4 | Lần 5 |
| Độ chính xác trên tập train | 96,03% | 96,17% | 96,05% | 96,28% | 96,17% |
| Độ chính xác trên tập test | 97,05% | 97,03% | 97,05% | 95,09% | 98,03% |

Bảng 3.3. Kết quả sau nhiều lần chạy thử NB kết hợp TF-IDF

Tỷ lệ chính xác của mô hình TF-IDF kết hợp Naive Bayes xấp xỉ 96,21%

~~~~

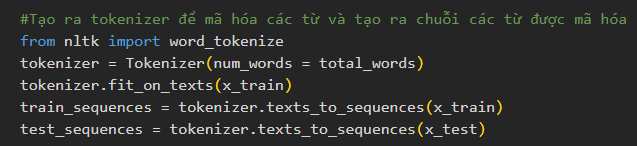
Hình 3.14. Ma trận nhầm lẫn của mô hình TF-IDF kết hợp với NB

Qua đó cho thấy sự tương đồng của TF-IDF và Bag-of-words khi kết hợp với mô hình Naive Bayes.

# **3.4. Mô hình RNN**

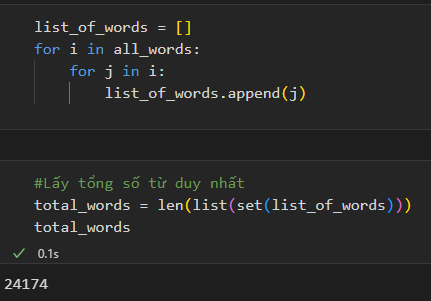
## *3.4.1. Vấn đề gặp phải khi biểu diễn văn bản và cách giải quyết*

Vì mô hình RNN rất quan tâm đến thứ tự xuất hiện của các từ trong văn bản còn Bag-of-words hay TF-IDF đều không thể hiện được điều đó nên thay vì sử dụng Bag-of-words hay TF-IDF để biểu diễn văn bản, em sẽ mã hóa dữ liệu bằng hàm Tokenizer của thư viện Tensorflow.



Hình 3.15. Mã hóa từ bằng Tokenizer

Hàm này hoạt động bằng cách lưu các từ trong tập dữ liệu vào 1 cuốn từ điển của nó, sau đó đánh số thứ tự (hay còn gọi là indexing) các từ đó. Số từ trong tập từ điển là do ta quy định, nhưng theo quy tắc những từ có nhiều lần xuất hiện nhất, những từ càng gần 0 thì xuất hiện càng nhiều. Ở đây em đã cho số lượng từ trong tập từ điển bằng với số lượng từ mà ta đã tìm ra trong tập dữ liệu.

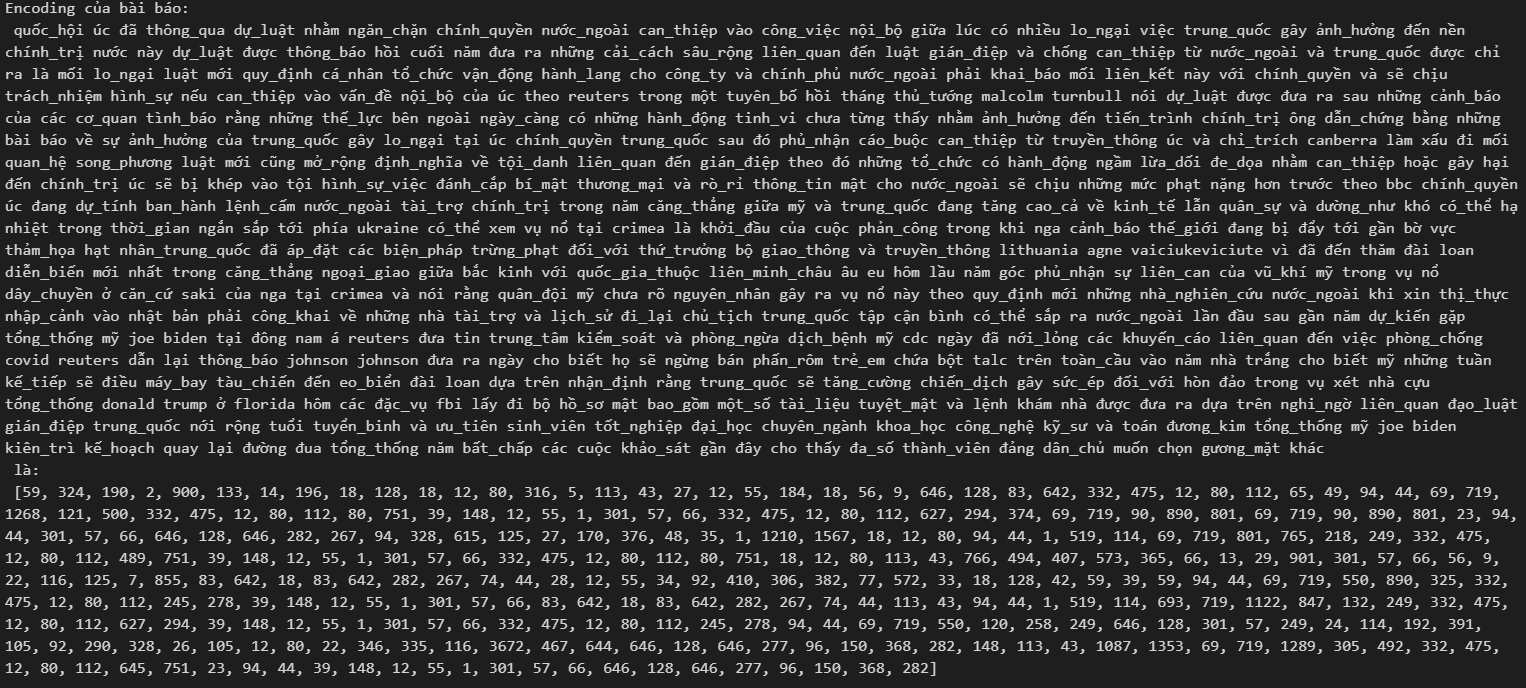


Hình 3. 16. Số từ duy nhất trong tập tài liệu

Sau đó sử dụng hàm fit\_on\_texts để cập nhật số từ đó vào tập từ điển.

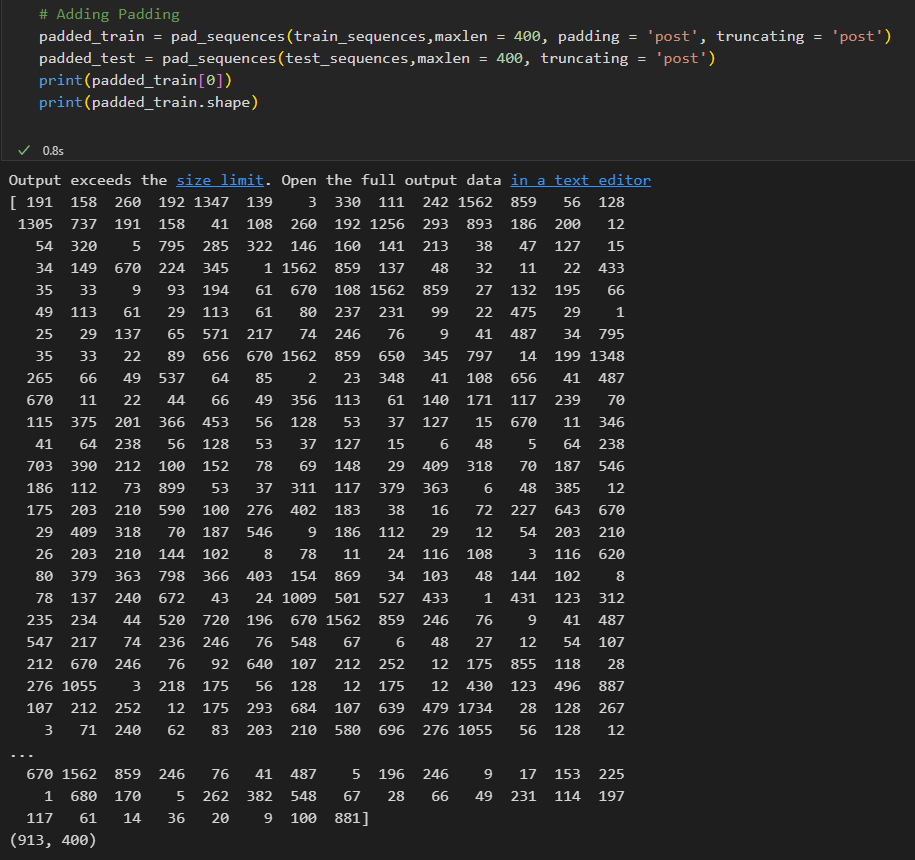
Và bước cuối cùng là hàm texts\_to\_sequences để chuyển tập dữ liệu chữ sang dữ liệu số với quy tắc đánh số là các index của tập từ điển được tạo ra ở bước trước đó, điều này cho phép việc biểu diễn văn bản vẫn giữ nguyên thứ tự xuất hiện của các từ trong câu.

Ta có thể nhìn thấy 1 ví dụ ở hình



Hình 3.17. Vector một bài báo mẫu

Và tiếp theo là hạ số chiều của văn bản xuống để tất cả các văn bản trong dữ liệu đều cùng 1 chiều, thuận tiện cho việc tính toán sau này. Vậy nên em đã tìm độ dài tối thiểu trong tập dữ liệu và chọn đó làm maxlen, tức số chiều của mỗi vector biểu diễn.

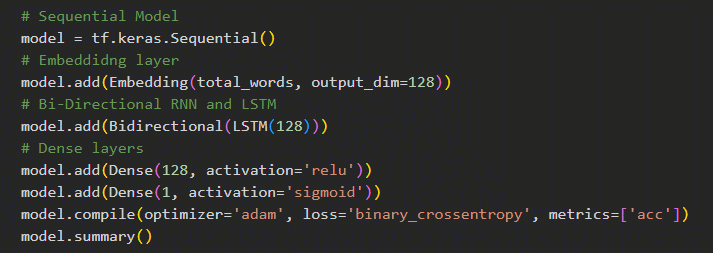


Hình 3.18. Trích xuất đặc trưng phù hợp với mô hình

## *3.4.2. Xây dựng mô hình RNN*

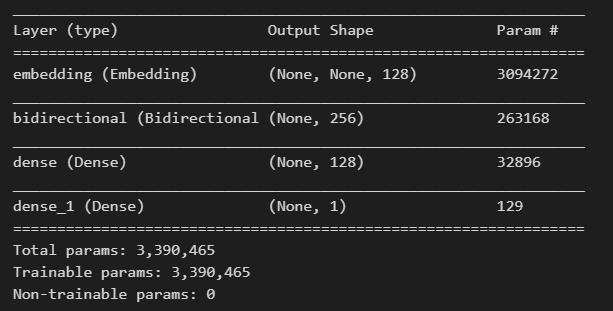
Xây dựng mô hình RNN bằng Tensorflow rất đơn giản. Vì thứ ta cần quan tâm là chọn số chiều và số lớp ẩn cho phù hợp với chương trình, phần tính toán đã có thư viện này xử lý.

* Bước Embedding: Chuyển số chiều của các vector thành 128 chiều.
* Thêm số lớp ẩn: thêm 1 lớp ẩn LSTM chạy dưới dạng Bidirectional. Bidirectional có nghĩa là hàm sẽ chạy theo 2 chiều, chiều từ trái sang phải và từ phải sang trái, đồng nghĩa với số chiều sẽ được nhân 2.
* Làm đặc xuống còn 128 chiều bằng hàm relu.
* Sau đó làm đặc còn 1 chiều bằng hàm sigmoid.
* Tổng hợp kết quả và so sánh bằng bước compile.



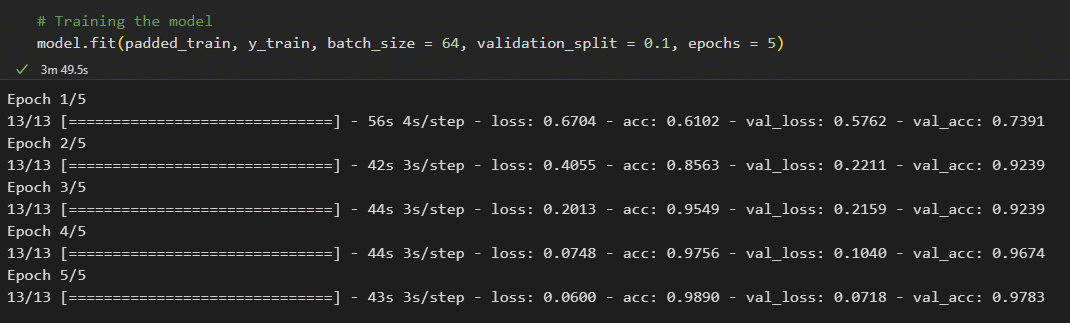
Hình 3.19. Thiết lập mô hình RNN

Bảng tổng quát thông số của mô hình RNN ở Hình 3.20



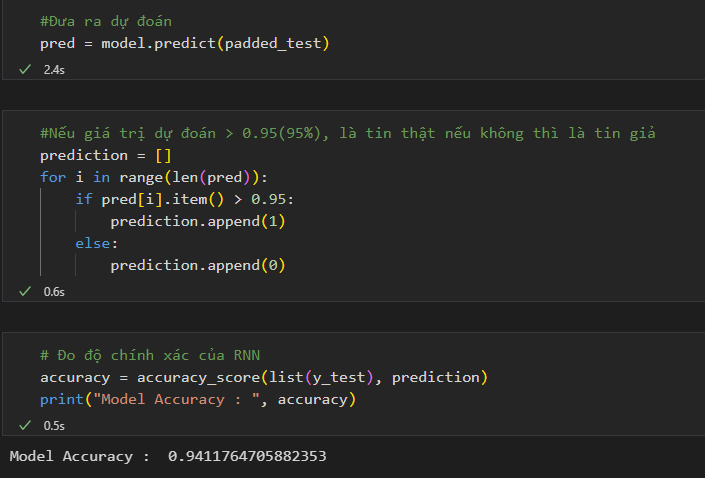
Hình 3.20. Thông số của mô hình RNN

Huấn luyện mô hình với dữ liệu đã được xử lý và nhãn, batch\_size = 64, validation\_split = 0.1 và sẽ huấn luyện 5 lần.



Hình 3.21. Train mô hình với epochs = 5

Sau đó ta đo độ chính xác của mô hình với accuracy\_score



Hình 3.22. Độ chính xác của RNN

Tạo nên 1 ma trận nhầm lẫn để trực quan hóa kết quả của mô hình RNN với đầu vào là nhãn test và kết quả dự đoán. Ta có thể quan sát ở Hình 3.23.

Application

Description automatically generated

Hình 3.23. Ma trận nhầm lẫn của mô hình RNN

Đánh giá: mô hình RNN có ưu điểm là lưu giữ lại các thông tin trong quá khứ để dựa vào đó xử lý thông tin mới nhập vào. Tuy nhiên việc này lại là 1 thử thách nếu mô hình phải xử lý một dữ liệu lớn đồng nghĩa với bước thời gian quá dài, điều này bắt buộc mô hình RNN phải có nhiều lớp ẩn sâu để xử lý. Nhưng khi mô hình có quá nhiều lớp ẩn sâu, mô hình sẽ trở nên không thể train được, điều này gọi là Vanishing gradient hay sự biến mất của đạo hàm.

Khi em thử tìm cách xử lý vấn đề này, em nhận thấy rằng nếu như gradient là 1 hằng số, thì mô hình sẽ không tự cải thiện được sau mỗi lần train vì mô hình học dựa vào sự thay đổi của gradient trong nó. Tương tự như nếu em hạ gradient xuống quá thấp thì mô hình cũng bị ảnh hưởng bởi Vanishing gradient vì gradient sẽ bị triệt tiêu chỉ sau vài bước. Vậy nên sau khi tham khảo nhiều nguồn trên mạng em nhận ra có 2 cách để giảm thiểu Vanishing gradient:

Cách thứ nhất, thay vì sử dụng activation function là tanh và sigmoid, ta thay bằng ReLu (hoặc các biến thể như Leaky ReLu). Đạo hàm của ReLu hoặc là 0 hoặc là 1, nên ta có thể kiểm soát phần nào vấn đề mất mát đạo hàm.

Cách thứ hai, ta thấy RNN thuần không hề có thiết kế nào để lọc đi những thông tin không cần thiết. Ta cần thiết kế một kiến trúc có thể nhớ dài hạn hơn, chẳng hạn như LSTM (Long-short Term Memory).

# **KẾT LUẬN**

**Thành tựu:**

* Trong quá trình tìm hiểu, nghiên cứu, phân tích và triển khai đề tài này, em đã áp dụng được các kiến thức cần thiết trong chuyên ngành công nghệ phần mềm, từ các kiến thức cơ bản về lập trình, cho đến các quy tắc và quy trình vận hành. Học hỏi, hiểu sâu hơn về các công nghệ và áp dụng vào tính toán, qua đó rút ra được những kinh nghiệm quý báu cho các dự án sau này.
* Hơn nữa, qua quá trình triển khai đồ án, đã giúp bản thân em biết cách quản lý thời gian cá nhân và quản lý công việc.
* Áp dụng khoa học công nghệ vào thực tiễn, góp phần phòng chống tin giả qua đó làm sạch mạng lưới thông tin.
* So sánh được sự khác nhau giữa các mô hình phân loại, giữa các mô hình học máy học sâu.

**Hướng phát triển:**

* Nghiên cứu thêm về các mô hình học sâu khác từ đó đưa ra được một bảng so sánh tổng quát về độ hiểu quả của từng mô hình trong việc phân loại tin giả.
* Xây dựng giao diện người dùng để có thể sử dụng các phương pháp trong đề tài một cách rộng rãi và hiệu quả hơn.

# **TÀI LIỆU THAM KHẢO**

* Luận án Tiến Sĩ kỹ thuật - Nghiên Cứu Ứng Dụng Kỹ Thuật Học Bán Giám Sát Vào Lĩnh Vực Phân Loại Văn Bản Tiếng Việt – Võ Duy Thanh – Đà Nẵng 2017
* [DeepLearningBook\_RefsByLastFirstNames.pdf (microsoft.com)](https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DeepLearningBook_RefsByLastFirstNames.pdf)
* [Deep Learning Algorithms - Javatpoint](https://www.javatpoint.com/deep-learning-algorithms)
* Tạp chí khoa học - Phân Loại Văn Bản Với Máy Học Vector Hỗ Trợ Và Cây Quyết Định - Trần Cao Đệ và Phạm Nguyên Khang – Trường Đại Học Cần Thơ.
* Research And Application Of Deep Learning Techniques In Automated Fake News Detection On Vietnamese News - Vo Trung Hung, Phan Thi Le Thuyen, Ninh Khanh Chi.