NỘI DUNG ÔN THI

1. **Ai and Medicine**

Trí tuệ nhân tạo (AI) đang trở thành một xu hướng quan trọng trong y tế, mang lại nhiều ứng dụng nhằm giảm tải công việc cho các bệnh viện, nâng cao chuyên môn và hiệu quả làm việc lâm sàng. Các ứng dụng AI hứa hẹn mang lại cải thiện đáng kể trong điều trị và chăm sóc sức khỏe, đặc biệt ở các khu vực khó tiếp cận như nông thôn hay vùng sâu vùng xa.

**AI trong Y Tế**

**Khái niệm**: AI trong Y Tế mô tả việc áp dụng các thuật toán học máy và công nghệ nhận thức để phân tích và hành động dựa trên dữ liệu Y Tế.

**Tầm quan trọng**:

* AI hỗ trợ bác sĩ, y tá và nhân viên y tế, nâng cao dịch vụ chăm sóc và chất lượng cuộc sống.
* AI giúp dự đoán và theo dõi sự lây lan của các bệnh truyền nhiễm, hỗ trợ công tác phòng chống dịch bệnh.

**Ứng dụng AI trong Y Tế**

1. **AI trong quyết định lâm sàng**

* Giúp bác sĩ xử lý lượng thông tin y khoa khổng lồ, cải thiện hiệu quả việc tra cứu và áp dụng thông tin y khoa vào điều trị

1. **AI trong chẩn đoán hình ảnh**

* Nâng cao chất lượng chẩn đoán bằng cách giảm thiểu sai sót
* Ví dụ: DrAid của VinBrain hỗ trợ chẩn đoán và điều trị ung thư gan và trực tràng, sử dụng các kỹ thuật tiên tiến để cải thiện độ chính xác

1. **AI trong hồ sơ sức khỏe điện tử**

* Giúp tăng cường tính tương tác, chất lượng và an toàn trong chăm sóc bệnh nhân bằng cách tối ưu hóa quản lý thông tin bệnh nhân

1. **AI trong robot y học**

* Hỗ trợ quá trình phẫu thuật với các công cụ và hướng dẫn từ AI
* AI giúp bác sĩ phẫu thuật trong việc đưa ra quyết định và thực hiện các nhiệm vụ đơn giản

1. **AI trong y học cá nhân hóa**

* Cải thiện độ chính xác của chẩn đoán và điều trị dựa trên cấu trúc di truyền, môi trường và lối sống của từng cá nhân
* Có tiềm năng cách mạng hóa chăm sóc sức khỏe bằng cách làm cho việc điều trị hiệu quả và giá cả phải chăng hơn

1. **AI trong quản lý khám, chữa bệnh**

* Hỗ trợ các tổ chức y tế dự đoán và chẩn đoán sớm các nguy cơ bệnh
* Tạo ra hệ sinh thái chăm sóc sức khỏe kết nối mạnh mẽ giữa bệnh nhân, bệnh viện, chuyên gia y tế và các tổ chức nghiên cứu

**Ưu điểm và nhược điểm**

**Ưu điểm**:

* Tăng cường chẩn đoán và điều trị
* Dự đoán kết quả và xu hướng
* Tối ưu hóa quản lý dữ liệu và tài nguyên

**Nhược điểm**:

* Không cung cấp chẩn đoán đa dạng , phù hợp với nhu cầu của mọi bệnh nhân
* Đạo đức và quyền riêng tư
* Khả năng thay thế con người

AI đã và đang tạo ra những thay đổi tích cực trong y tế, từ việc hỗ trợ quyết định lâm sang, chẩn đoán hình ảnh đến quản lý thông tin và phẫu thuật. Sự ứng dụng rộng rãi của AI hứa hẹn nâng cao chất lượng hiệu quả của hệ thống chăm sóc sức khỏe toàn cầu.

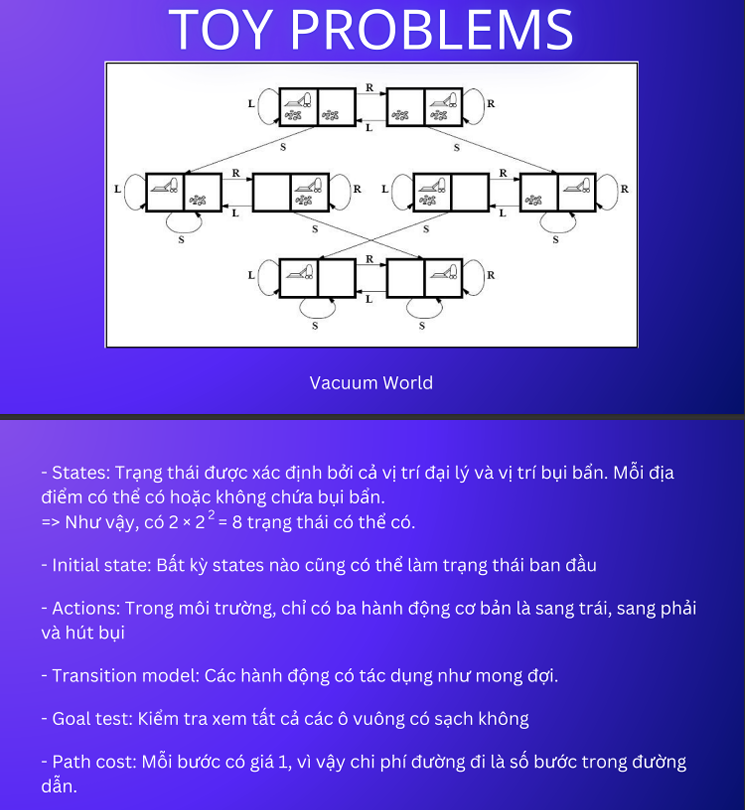
1. **Solving Problems by Search**

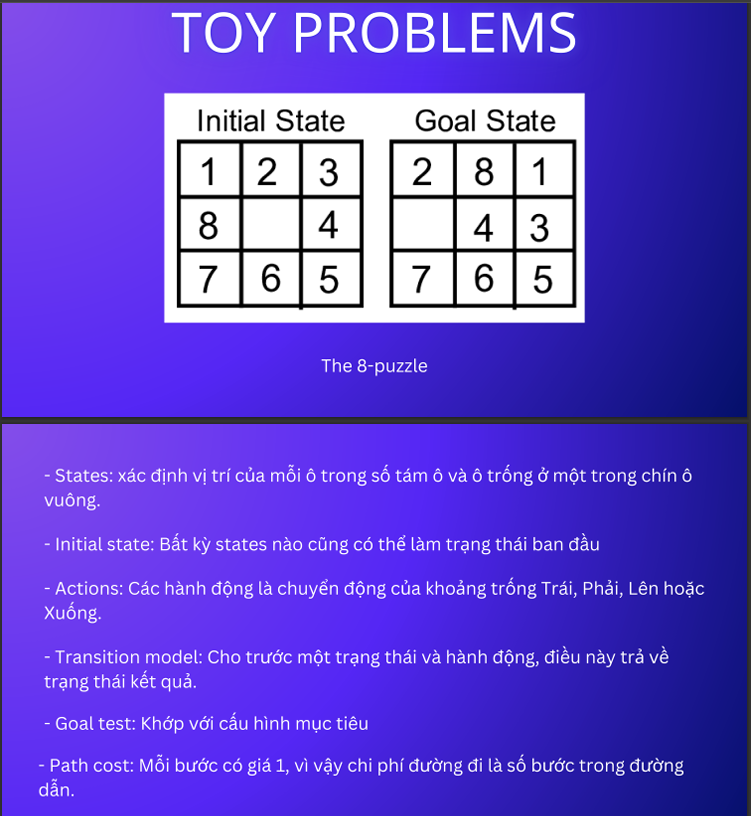
Solving Problems by Search là một phương pháp cơ bản trong trí tuệ nhân tạo (AI) để tìm kiếm giải pháp cho các bài toán phức tạp. Phương pháp này sử dụng các thuật toán tìm kiếm để khám phá không gian trạng thái của bài toán và tìm ra con đường từ trạng thái ban đầu đến trạng thái mục tiêu.

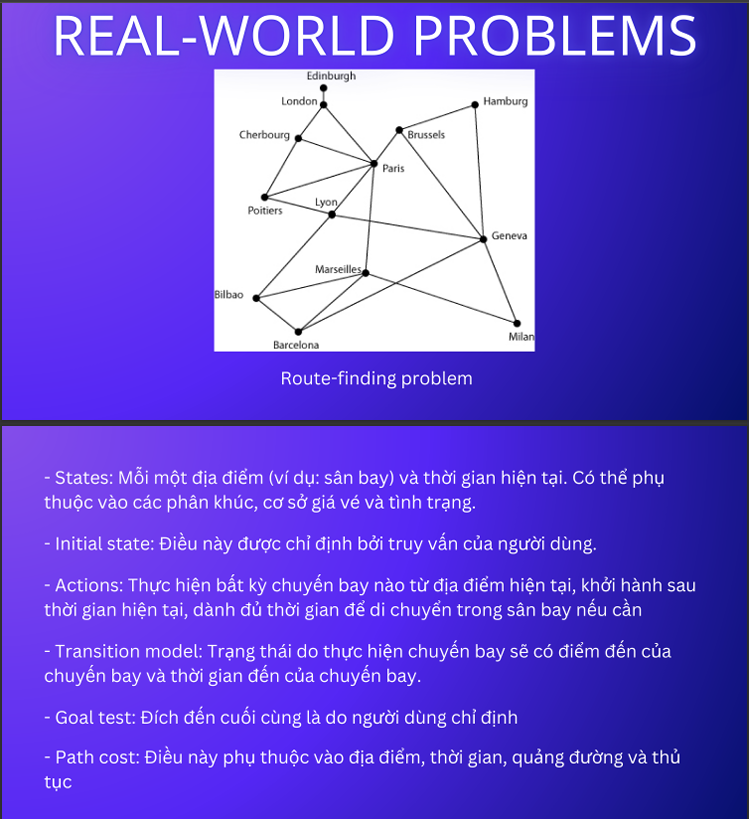
**Khái niệm cơ bản**

* Trạng thái (States): Mỗi tình huống hoặc cấu hình của bài toán tại một thời điểm cụ thể.
* Trạng thái ban đầu (Initial State): Trạng thái xuất phát của bài toán
* Trạng thái mục tiêu (Goal State): Trạng thái mà ta mong muốn đạt được
* Không gian trạng thái (State Space): Tập hợp tất cả các trạng thái có thể có từ trạng thái ban đầu thông qua các hành động
* Hành động (Action): Bước hoặc phép biến đổi có thể áp dụng để chuyển từ trạng thái này sang trạng thái khác
* Đường đi (Path): Chuỗi các trạng thái liên tiếp từ trạng thái ban đầu đến trạng thái mục tiêu

**Một số ví dụ**







**Searching for solutions**

* **Xác định vấn đề**: Đầu tiên, hãy hiểu rõ vấn đề bạn đang đối mặt. Xác định các yếu tố cụ thể, nguyên nhân gốc rễ và mục tiêu mong muốn
* **Nghiên cứu và thu thập thông tin**: Khi đã xác định được vấn đề, nghiên cứu các tài liệu, dữ liệu và thông tin liên quan. Thu thập dữ liệu và thông tin cần thiết để hiểu sâu hơn về vấn đề
* **Phân tích và đánh giá**: Phân tích thông tin đã thu thập để hiểu rõ hơn về các yếu tố ảnh hưởng và tương tác giữa chúng. Đánh giá các lựa chọn có sẵn và những hậu quả có thể xảy ra từ mỗi lựa chọn
* **Xác định phương pháp và chiến lược**: Dựa trên thông tin và phân tích, xác định phương pháp và chiến lược tiếp cận phù hợp nhất để giải quyết vấn đề. Cân nhắc các phương tiện và công cụ có thể sử dụng, cũng như các bước cụ thể cần thực hiện

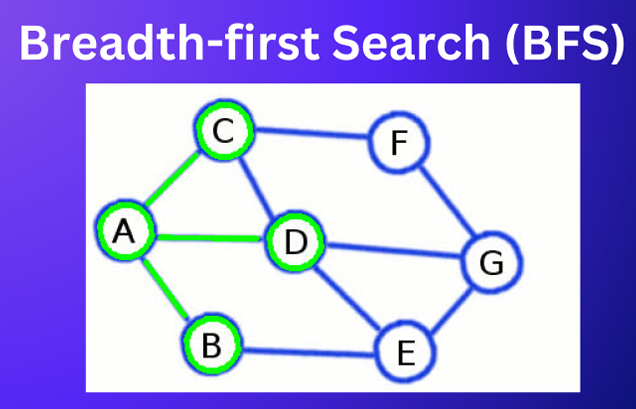
**Uniformed Search Strategies**

* **Thực hiện và kiểm soát**: Triển khai phương pháp và chiến lược đã chọn vào thực tế. Theo dõi quá trình thực hiện và kiểm soát để đảm bảo rằng nó đang tiến triển theo kế hoạch và đạt được kết quả mong muốn
* **Đánh giá và điều chỉnh**: Sau khi triển khai, đánh giá kết quả và hiệu suất của phương pháp. Dựa vào đánh giá này, điều chỉnh và cải tiến chiến lược nếu cần thiết để đạt được kết quả tốt nhất

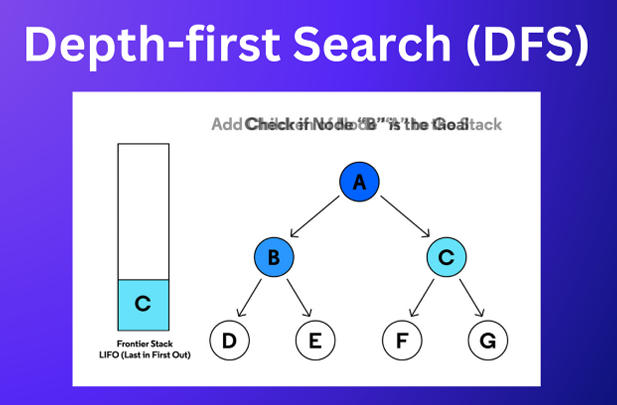
**What is uniformed search?**

Các thuật toán thực hiện tìm kiếm mà không có bất kỳ thông tin nào về mục tiêu ngoài cấu trúc của không gian tìm kiếm. Điều này có nghĩa là chúng không sử dụng bất kỳ dữ liệu nào có thể giúp chúng định hình hoặc ưu tiên các hướng tìm kiếm

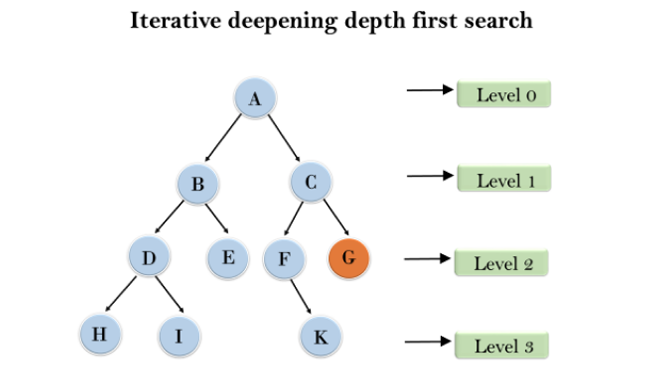
* **Tìm kiếm theo chiều rộng (Breadth-First Search – BFS):** Khám phá các trạng thái ở mức độ gần trạng thái ban đầu trước, sau đó mới đến các mức độ xa hơn. BFS đảm bảo tìm được đường đi ngắn nhất trong không gian trạng thái không trọng số



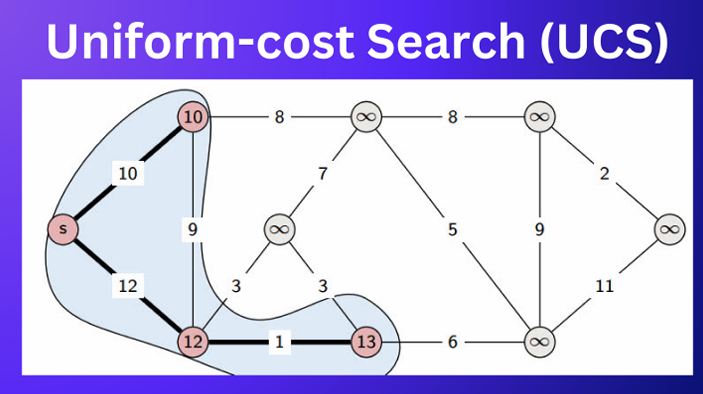
* + **Ưu điểm**
    - Chắc chắn tìm ra lời giải nếu có
    - Duyệt qua đầy đủ các đỉnh nếu số đỉnh là hữu hạn
    - Đường đi tìm được qua ít đỉnh nhất
  + **Nhược điểm**
    - Không phù hợp với bài toán có không gian tìm kiếm lớn
    - Tốn bộ nhớ
    - Tăng chi phí xử lý các nút không cần thiết
* **Tìm kiếm theo chiều sâu (Depth-First Search – DFS):** Khám phá sâu vào cây tìm kiếm, tiếp tục đi xa nhất có thể trước khi quay lại và khám phá các nhánh khác



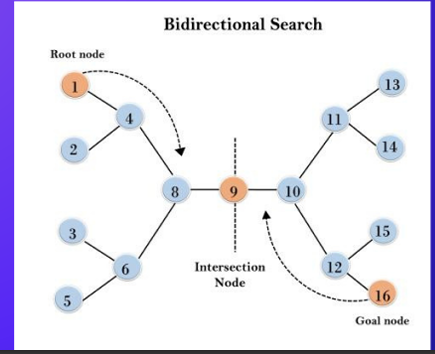
* + **Ưu điểm**
    - Tốn ít bộ nhớ hơn BFS
    - Trong trường hợp đi đúng đường, nhanh hơn BFS
    - Nếu lời giải ở rất sâu, thời gian tìm ra nhanh hơn BFS
  + **Nhược điểm**
    - Không phù hợp với bài toán có không gian tìm kiếm lớn
    - Dễ bị rơi vào vòng lặp vô hạn nếu đầu vào quá lớn
    - Trong trường hợp đầu vào quá lớn có thể không tìm được lời giải
* **Tìm kiếm theo chiều sâu có giới hạn (Depth-Limited Search):** Phiên bản cải tiến của DFS với giới hạn độ sâu để tránh duyệt vô tận
  + **Ưu điểm**
    - Tận dụng được các ưu điểm của DFS
    - Trong trường hợp đích gần đỉnh, chi phí sẽ được tối ưu => Hiệu quả bộ nhớ
    - Tránh được việc rơi vào vòng lặp vô hạn nếu đầu vào quá lớn
  + **Nhược điểm**
    - Khi mà các lời giải nằm ở độ sâu lớn hơn giới hạn độ sâu thì thuật toán thất bại
    - Không xét qua đầy đủ các đỉnh vì giới hạn, có khả năng không tìm ra lời giải dù lời giải tồn tại
    - Không thể tối ưu nếu bài toán có hơn 1 giải pháp
* **Tìm kiếm theo chiều sâu lặp (Iterative Deepening Search):** Kết hợp ưu điểm của BFS và DFS bằng cách lặp lại DFS với các giới hạn độ sâu tăng dần



* + **Ưu điểm**
    - Tận dụng ưu điểm của DFS và BFS
    - Tránh được việc rơi vào vòng lặp vô hạn nếu đầu vào quá lớn
    - Tối ưu chi phí nếu chi phí các cạnh của đồ thị là như nhau
  + **Nhược điểm**
    - Duyệt lặp lại các đỉnh đã lặp trước đó
    - Thời gian tìm kiếm có thể tăng đáng kể
* **Uniform-cost Search (UCS)**



* + **Ưu điểm**
    - Con đường tìm ra luôn tối ưu chi phí
    - Luôn tìm thấy lời giải nếu tồn tại
    - Khả năng tìm kiếm trong đồ thị có trọng số biến đổi
  + **Nhược điểm**
    - Không tối ưu nếu có trọng số (chi phí) < 0
    - Trường hợp các cạnh có trọng số như nhau, thuật toán biến thành BFS
* **Bidirectional Search (BiS)**

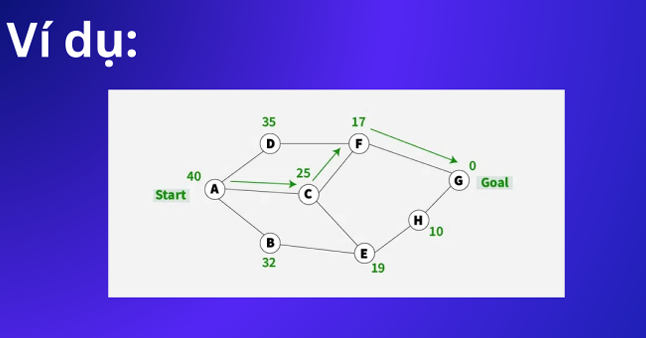


* + **Ưu điểm**
    - Hạn chế việc lặp vô tận
    - Tốn ít bộ nhớ hơn
    - Tốc độ nhanh hơn các phương pháp khác
  + **Nhược điểm**
    - Chỉ phù hợp với các bài toán đã xác định được điểm bắt đầu và đích
    - Trong trường hợp bài toán thay đổi, khó để kiểm soát việc đồng bộ giữa hai hướng tìm kiếm
    - Trong thực tế rất khó triển khai vì không biết được điểm giao giữa đích và điểm bắt đầu

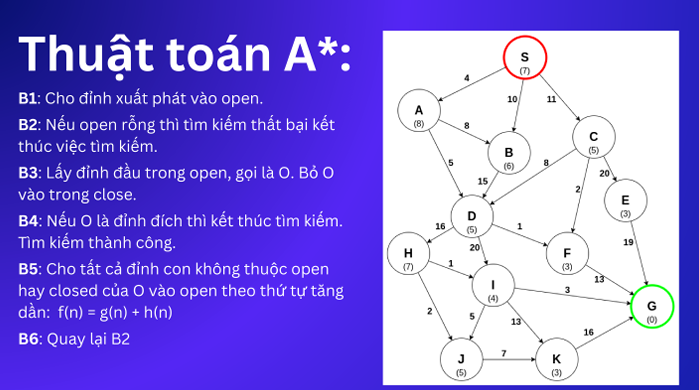
**Informed (Heuristic) search strategies**

Một chiến lược sử dụng kiến thức cụ thể về vấn đề vượt ra ngoài định nghĩa của vấn đề chính nó – có thể tìm ra các giải pháp hiệu quả hơn so với một chiến lược không thông tin

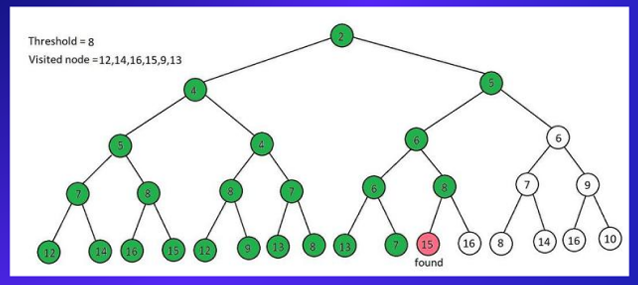
* **Tìm kiếm tham lam (Greedy Best-First Search):** Cố gắng mở rộng nút gần nhất với mục tiêu, với lý do là điều này có thể dẫn đến một giải pháp nhanh chóng, Do đó, nó đánh giá các nút bằng cách sử dụng chỉ hàm kinh nghiệm, nghĩa là f(n) = h(n)
  + **Ưu điểm**
    - Tính hiệu quả
    - Dễ triển khai
    - Dễ dàng tùy chỉnh
    - Hiệu quả trong các vấn đề tìm kiếm gần mục tiêu
  + **Nhược điểm**
    - Không đảm bảo tối ưu
    - Dễ rơi vào vòng lặp vô hạn
    - Dễ bị ảnh hưởng bởi lỗi của hàm heuristic



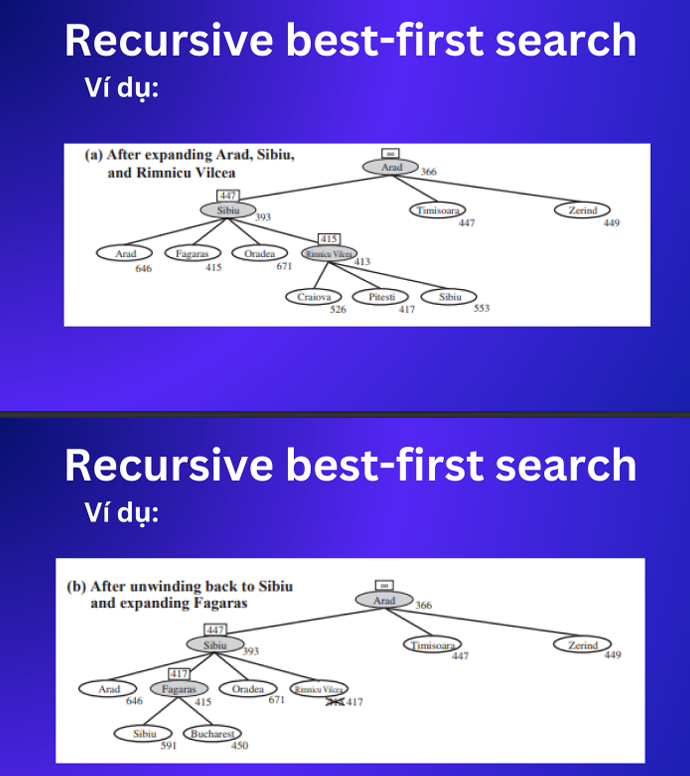
* **A\* search:** Thuật toán A\* là một trong những kỹ thuật tốt nhất và phổ biến nhất được sử dụng trong tìm đường và duyệt đồ thị. Sử dụng khoảng g(n) là khoảng cách từ đỉnh bắt đầu đến đỉnh đang xét. Sử dụng khoảng h(n) là khoảng các từ đỉnh đang xét đến đỉnh kết thúc

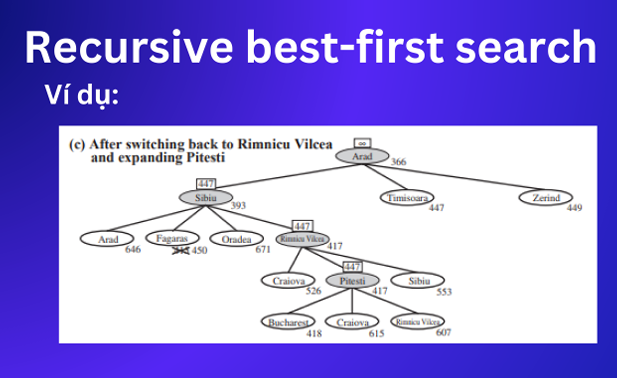


* + **Ưu điểm**
    - TÍnh tối ưu
    - Tính linh hoạt
    - Hiệu suất cao
    - Có thể được tối ưu hóa
  + **Nhược điểm**
    - Đòi hỏi thông tin chính xác từ hàm heuristic
    - Tính chậm chạp trong một số trường hợp
* **Memory-bounded heuristic search**: Một phương pháp tìm kiếm sử dụng hàm heuristic nhưng giới hạn bộ nhớ được sử dụng trong quá trình tìm kiếm. Phương pháp này thường được sử dụng trong các tình huống mà bộ nhớ là hạn chế, như trong các ứng dụng thời gian thực hoặc trên các thiết bị có dung lượng bộ nhớ hạn chế
  + Các chiến lược phổ biến bao gồm
    - Giảm bớt dữ liệu lưu trữ
    - Chọn lựa cẩn thận nút để duyệt
    - Tối ưu hóa quá trình mở rộng nút
    - Tái sử dụng thông tin
* **IDA\***

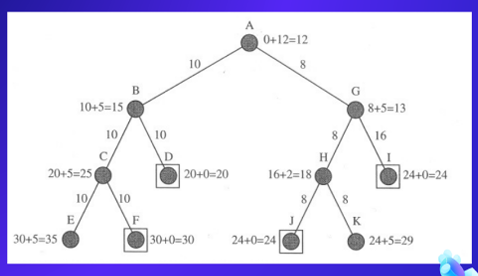


* + Sử dụng phương pháp “iterative deepening” giống như thuật toán tìm kiếm theo chiều sâu (DFS). Thay vì duyệt qua toàn bộ cây tìm kiếm trong một lần duyệt, IDA\* lặp lại quá trình tìm kiếm với các ngưỡng tăng dần
  + Khi duyệt mỗi lần lặp, IDA\* sử dụng một ngưỡng f-cost để quyết định xem một nút có nên được mở rộng hay không. Ngưỡng này được tăng dần qua mỗi lần lặp lại, giúp thuật toán tìm kiếm ưu tiên các nút có cho phí thấp hơn
* **Recursive best-first search**
  + RBFS là một thuật toán tìm kiếm được sử dụng để giải quyết các bài toán trên đồ thị hoặc không gian trạng thái
  + RBFS là một biến thể của thuật toán Best-First Search (BFS). Nó sử dụng phương pháp đệ quy để tìm kiếm trên cây tìm kiếm và chọn lựa chọn tốt nhất từ các nút mở rộng
  + Ý tưởng: Thuật toán này khám phá các nút trong cây tìm kiếm theo chiều sâu, đánh giá mỗi nút dựa trên giá trị f (hàm đánh giá) và giới hạn f, một biến để theo dõi giá trị f tốt nhất của con đường thay thế từ bất kỳ nút cha nào. Nếu giá trị f của một nút vượt quá giới hạn f, thuật toán quay lui trở lại nút cha và tiếp tục khám phá từ nó





* **SMA\*:** luôn loại bỏ nút lá tồi nhất – nút có giá trị f cao nhất. Giống như RBFS, SMA\* sau đó sao lưu giá trị của nút bị quên vào nút cha của nó. Điều này có nghĩa là cha của một cây con bị quên sẽ biết chất lượng của đường dẫn tốt nhất trong cây con đó. Với thông tin này, SMA\* tái tạo lại cây con chỉ khi tất cả các con đường khác đã được chứng minh là tệ hơn đường dẫn mà nó đã quên



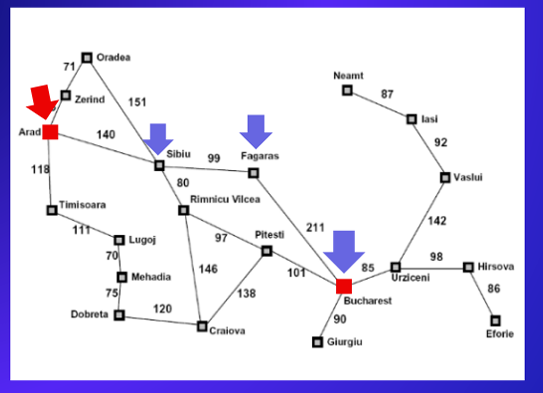
* + **Ưu điểm**
    - Quản lý bộ nhớ hiệu quả
    - Tìm kiếm thông minh
    - Tính động và linh hoạt
  + **Nhược điểm**
    - Khả năng tìm kiếm không đầy đủ
    - Hiệu suất phụ thuộc vào hàm đánh giá
    - Khả năng tái tạo cây tìm kiếm

**Heuristic Functions**

**Heuristic Function là gì?** Hàm heuristic (Thuật toán) hay đơn giản là một phương pháp để giải một bài toán mục tiêu là đưa ra giải pháp khả thi trong một khoảng thời gian khả thi

**Vai trò của hàm heuristic trong trí tuệ nhân tạo**

* **Guiding Search Algorithms**
  + Hướng dẫn các thuật toán tìm kiếm trong việc điều hướng không gian trạng thái của một bài toán. Sử dụng phương pháp phỏng đoán để ước tính mức độ mong muốn của các trạng thái khác nhau, giúp chúng đưa ta quyết định sang suốt hơn về việc nên khám phá đường dẫn nào trước tiên
  + Heuristics cung cấp một cách để ưu tiên và đánh giá các giải pháp tiềm năng, tập trung vào những giải pháp có nhiều khả năng dẫn đến trạng thái mục tiêu hơn. Hướng dẫn này làm giảm nhu cầu khám phá tất cả các tùy chọn có thể một cách thấu đáo, giúp tìm kiếm hiệu quả hơn
* **Speeding Up Problem Solving**
  + Tìm ra câu trả lời nhanh hơn hoặc giải pháp gần đúng, ngay cả khi nó không tối ưu
  + Trading accuracu for speed



* **Improving Decision-Making**
  + Sử dụng trong quá trình ra quyết định, chẳng hạn như trong AI trong trò chơi hoặc lập kế hoạch tuyến đường. Chúng giúp các tác nhân AI đánh giá mức độ mong muốn của các động thái hoặc hành động khác nhau bằng cách ước tính kết quả tiềm năng và giá trị mong đợi của những lựa chọn đó

**Ứng dụng thực tế của hàm Heuristic**

* **Lập kế hoạch lộ trình cho điều hướng GPS**
  + Ước tính khoảng cách di chuyển, xem xét các yếu tố như độ dài đường, điều kiện giao thông và dữ liệu thời gian thực khác
  + Hướng dẫn các thuật toán tìm kiếm các tuyến đường hiệu quả bằng cách đánh giá các tuyến đường tiềm năng dựa trên ước tính heuristic
    - * + Tính toán tuyến đường hiệu quả
        + Đề xuất tuyến đường nhanh hơn
        + Cung cấp các tuyến đường thay thế và định tuyến lại trong thời gian thực
* **AI trong trò chơi**
  + Tính toán các bước hành động kế tiếp để đưa ra các quyết định thông minh
  + Nâng cao hiệu suất, độ thông minh của AI nhằm tạo thách thức cho người chơi
* **Năng suất và hiệu quả**
  + Tăng tốc đáng kể trong quá trình ra quyết định
  + Khả năng xử lý các tình huống phức tạp và năng động một cách hiệu quả
  + Cải thiện trải nghiệm người dung và áp dụng AI rộng rãi hơn trong cuộc sống hàng ngày của chúng ta

1. **Xử lý ngôn ngữ tự nhiên**

**Xử lý ngôn ngữ tự nhiên (NLP) là gì?**

Natural Language Processing (NLP) hay còn gọi là xử lý ngôn ngữ tự nhiên là một lĩnh vực trong trí tuệ nhân tạo (AI) và khoa học máy tính. Chúng tập trung vào việc phát triển các phương pháp và công nghệ để máy tính có thể hiểu, phân tích và tạo ra ngôn ngữ tự nhiên của con người. Nói cách khác, NLP cho phép máy móc “hiểu” và “sử dụng” ngôn ngữ của chúng ta.

**Vai trò và nhiệm vụ**

**Vai trò**

* **Trợ lý ảo và chatbot:** NLP giúp tạo ra các trợ lý ảo và chatbot có khả năng hiểu và phản hồi lại ngôn ngữ tự nhiên của con người - VD: Chatbot AIDA phát triển cho trang web bán hàng, Trợ lý ảo Siri
* **Nhận dạng giọng nói:** NLP giúp chuyển đổi dữ liệu giọng nói thành dữ liệu cho phép máy tính hiểu và xử lý thông tin - VD: Loa thông minh Amazon Echo, Google Home
* **Tìm kiếm thông tin:** NLP giúp cải thiện khả năng tìm kiếm thông tin của máy tính, từ đó giúp người dung có thể tìm kiếm thông tin một cách nhanh chóng và chính xác - VD: Công cụ tìm kiếm Google và Edge
* **Dịch và dự đoán ngôn ngữ:** NLP có vai trò quan trọng trong các hệ thống dịch máy tự động - VD: Công cụ dịch thuật Google Translate và chức năng phát hiện ngôn ngữ
* **Phân tích cảm xúc:** NLP giúp phân tích cảm xúc từ văn bản, các nguồn dữ liệu từ mạng xã hội, đánh giá sản phẩm, từ đó giúp hiểu rõ hơn về ý kiến và cảm xúc của người dùng – VD: Hệ thống NLP tự động thu nhập các đánh giá từ diễn đàn

**Nhiệm vụ**

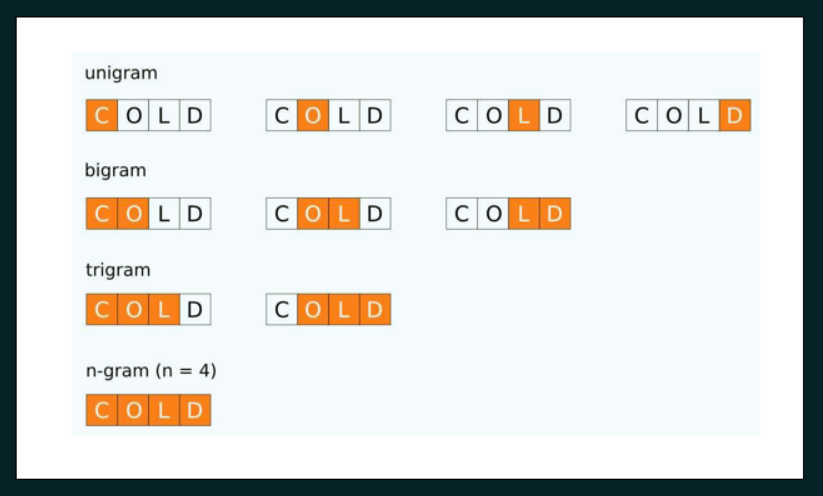
* **Mô hình hóa ngôn ngữ**
* **Phân loại văn bản**: NLP giúp máy tính hiểu ý nghĩa văn bản, bao gồm việc phân tích cú pháp (syntax), phân loại từ loại (POS tagging) và dự đoán ngữ cảnh (context prediction)
* **Trích xuất thông tin**: NLP giúp trích xuất thông tin từ văn bản, từ việc tìm kiếm thông tin đến phân tích dữ liệu trong các tài liệu lớn
* **Truy xuất thông tin**
* **Tóm tắt văn bản**
* **Hỏi đáp**
* **Dịch máy**: NLP giúp máy tính dịch từ ngôn ngữ này sang ngôn ngữ khác một cách tự động nhờ vào các mô hình dịch máy dựa trên học máy và deep learning
* **Mô hình hóa chủ đề**
* **Phân tích cảm xúc và ý kiến**: NLP có thể phân tích cảm xúc và ý kiến trong văn bản, giúp đánh giá sự phản hồi của người dung hoặc phản ứng của công chúng đối với các sự kiện hay sản phẩm
* **Tạo ra văn bản tự động:** NLP có thể được sử dụng để tạo ra văn bản tự đọng, từ việc viết bài báo đến tạo ra nội dung cho website hoặc ứng dụng
* **Tương tác người – máy:** NLP là cơ sở cho các ứng dụng tương tác người – máy. Có thể kể đến như chatbot, trợ lý ảo và hệ thống trả lời tự động

**Language Model**

* Ngôn ngữ tự nhiên, như tiếng Anh hoặc tiếng Tây Ban Nha, không thể được thể hiện như một tập hợp xác định các câu mà còn các cấu trúc ngữ pháp. Mọi người đều đồng ý rằng "Not to be invited is sad" là một câu tiếng Anh, nhưng mọi người không đồng ý về tính chính xác ngữ pháp của "To be not invited is sad." Do đó, việc định nghĩa một mô hình ngôn ngữ tự nhiên dưới dạng một phân phối xác suất trên các câu là có ý nghĩa hơn việc định nghĩa dưới dạng một tập hợp xác định. Đó là, thay vì hỏi xem một chuỗi từ có thuộc của tập hợp định nghĩa ngôn ngữ, chúng ta thay vào đó hỏi về xác suất của những từ đó.
* Ngôn ngữ tự nhiên cũng có tính mơ hồ (**Ambiguous**). "He saw her duck" có thể có nghĩa là anh ấy đã nhìn thấy một con vịt thuộc về cô ấy, hoặc là anh ấy đã nhìn thấy cô ấy di chuyển để tránh cái gì đó hoặc cũng có thể là anh ấy cưa con vịt của cô ấy. Do đó, một lần nữa, chúng ta không thể nói về một ý nghĩa duy nhất cho một câu, mà thay vào đó là về một phân phối xác suất trên các ý nghĩa có thể.
* Cuối cùng, ngôn ngữ tự nhiên khó xử lý vì chúng rất lớn và liên tục thay đổi. Vì vậy, mô hình ngôn ngữ của chúng tôi, tốt nhất là, chỉ là một xấp xỉ. Chúng tôi bắt đầu với những xấp xỉ đơn giản nhất có thể và tiến triển từ đó.
* Language Model (LM) là một mô hình **đại diện cho những kiến thức đã biết về một ngôn ngữ**, những kiến thức ấy có thể là những từ, chuỗi các từ có thể có hay mức độ thường xuyên mà chúng xuất hiện.

**Ứng dụng của Language Model**

* **N-gram character models:** 
  + Là một loại mô hình ngôn ngữ được sử dụng để xác định xác suất xuất hiện của các chuỗi ký tự liên tiếp trong một văn bản
  + Mỗi “n-gram” là một chuỗi gồm n ký tự, có thể là ký tự, số, khoảng trắng hoặc các ký tự đặc biệt
* Ứng dụng: Là để xác định ngôn ngữ bằng cách tính xác suất xuất hiện các n-gram trong các tập văn bản của ngôn ngữ. Từ đó xác định được văn bản đó thuộc ngôn ngữ nào với xác suất lớn nhất



* **N-gram word models:** Thay vì xác định tần suất xuất hiện của các kí tự thì xác định tần suất xuất hiện của các từ

Vấn đề

* + Với các ký tự thì tập hợp nhỏ chỉ khoảng 100 ký tự cho hầu hết các ngôn ngữ, nhưng với từ thì có đến hàng ngàn, hàng triệu từ
  + Out of vocabulary: với các ký tự thì chuyện người ta tự phát minh ra một ký tự mới là rất hiếm, nhưng với từ thì các từ được liên tục tạo ra và ta có thể giải quyết bằng cách gán các từ mới lần đầu xuất hiện bằng <UNK> (unknown word)



**Các phương pháp Smoothing n-gram models**

Với mô hình N-gram có một vấn đề là tập dữ liệu chỉ chữa một số lượng từ hữu hạn vậy nên với các từ thông dụng thì xác xuất trả về rất ổn định nhưng với những từ hiếm khi xuất hiện và chưa được thêm vào bộ dữ liệu thì khi tính xác suất trả về bằng 0. Nhưng việc các từ vì  chưa xuất hiện trong tập dữ liệu mà đánh giá là không tồn tại trong ngôn ngữ đó là không chính xác. Vì vậy người ta đưa ra phương pháp làm mịn để khắc phục vấn đề này. Có thể chia thành 3 loại chính:

**Discounting – Giảm xác suất**

* Giảm xác suất các cụm n-gram có xác suất lớn hơn 0 để bù cho các cụm n-gram chưa xuất hiện
* Discounting – Add one (Laplace): với một n-gram chưa từng xuất hiện trong một tập dữ liệu n phần tử thì thay vì cho xác suất của nó bằng 0 thì sẽ cho nó bằng 1/(n+2)

**Back-off – Truy hồi**

* Tính xác suất các n-gram chưa xuất hiện hoặc có xác suất rất nhỏ bằng cách “backing-off” (lùi lại) thành các cụm ngắn hơn và có xác suất lớn hơn 0

**Interpolation – Nội suy**

* Tính xác suất của tất cả các cụm n-gram bằng các cụm ngắn hơn

**Model evaluation**

Trong thực tế, ta cần áp dụng nhiều thuật toán Machine learning để chọn ra được mô hình phù hợp nhất cho bài toán của mình. Vấn đề đặt ra, làm thế nào để đánh giá và chọn ra các mô hình. Ngoài thuật toán học máy, sự thực thi của mô hình có thể phụ thuộc vào các yếu tố khác như sự phân bố của các lớp, chi phí phân loại sai, kích thước của tập huấn luyện và tập thử nghiệm, độ đo thực thi. Trong bài viết này, ta sẽ đánh giá thực thi: tập trung vào khả năng dự đoán của mô hình hơn là tốc độ phân loại hay xây dựng mô hình, khả năng co giãn.

**Text Classification – Phân loại văn bản**

**Khái niệm**

Phân loại văn bản là quá trình nhóm văn bản thành các danh mục hoặc lớp được xác định trước. Vd: Phân tích tâm trạng (phân loại 1 bài đánh giá phim hoặc sản phẩm là tích cực hoặc tiêu cực), phát hiện thư rác (phân loại một tin nhắn email là rác [spam] hoặc không phải rác[not spam - ham]).

**Spam detection**

* Chúng ta có thể coi phát hiện thư rác như 1 vấn đề trong học giám sát.
* Đặc điểm quan trọng:  rác có khả năng cao là tất cả chữ in hoa và có dấu câu nhúng trong các từ. Mô hình ký tự có thể phát hiện ra điều này. Chúng ta có thể tạo ra 1 bộ ký tự n-gram đầy đủ của SPAM-HAM, hoặc chúng ta có thể tạo ra các đặc điểm được tạo thủ công như “số lượng dấu câu được nhúng trong các từ”.

**Classification by data compression**

* Phân loại văn bản có thể được hiểu như là một vấn đề trong nén dữ liệu, trong đó các thuật toán nén dữ liệu hoạt động bằng cách tạo ra một mô hình ngôn ngữ để mô tả dữ liệu một cách hiệu quả.

Vd: Văn bản "0.142857142857142857" có thể được nén thành "0.[142857]\*3." Thuật toán nén hoạt động bằng cách xây dựng từ điển các phần từ của văn bản, và sau đó tham chiếu đến các mục trong từ điển, ở đây chỉ có một mục trong từ điển, "142857."

* Thuật toán nén dữ liệu như LZW tạo ra một mô hình ngôn ngữ trực tiếp, và việc phân loại bằng cách nén dữ liệu của các nhóm văn bản có thể được thực hiện bằng cách so sánh độ nén của mỗi nhóm để dự đoán lớp của một văn bản mới.
* Mặc dù các thuật toán nén có thể chạy chậm, nhưng chúng có độ chính xác tương đương với các thuật toán phân loại truyền thống, cho thấy tiềm năng của việc sử dụng các phương pháp nén dữ liệu trong phân loại văn bản.

**Các trường hợp sử dụng phân loại văn bản**

* Giám sát các trường hợp khẩn cấp
  + Quét các bài đăng và cuộc trò chuyện
  + Áp dụng các công cụ phân loại văn bản
  + Lọc tính khẩn cấp
  + Phát hiện các phản ứng tiêu cực
* Xác định các cách để quảng bá thương hiệu
  + Theo dõi các đánh giá, phản hồi của người dung
  + Xác định những người có ảnh hưởng, người quảng bá và người gièm pha
* Xử lý dữ liệu dễ dàng hơn
* Phân loại yêu cầu dịch vụ
  + Các doanh nghiệp quản lý rất nhiều yêu cầu dịch vụ mỗi ngày
  + Dễ dàng gắn thẻ công việc dựa trên danh mục, vị trí và yêu cầu
* Cải thiện trải nghiệm người dùng trang web
  + Phân tích nội dung và hình ảnh của sản phẩm
  + Gán nó và đúng danh mục để cải thiện trải nghiệm người dung khi mua sắm

**Quy trình phân loại văn bản**

* **Sơ chế (Pre-Processing)**

**Mã hóa:** Văn bản được chia thành các dạng văn bản nhỏ hơn và đơn giản hơn để dễ dàng phân loại.

* **Lựa chọn tính năng (Feature Selection)**

Lựa chọn tính năng là một bước cơ bản trong phân loại văn bản. Quá trình này nhằm mục đích đại diện cho các văn bản có tính năng phù hợp nhất. Lựa chọn tính năng giúp loại bỏ dữ liệu không liên quan và nâng cao độ chính xác.

* **Khai thác tính năng (Feature Extraction)**

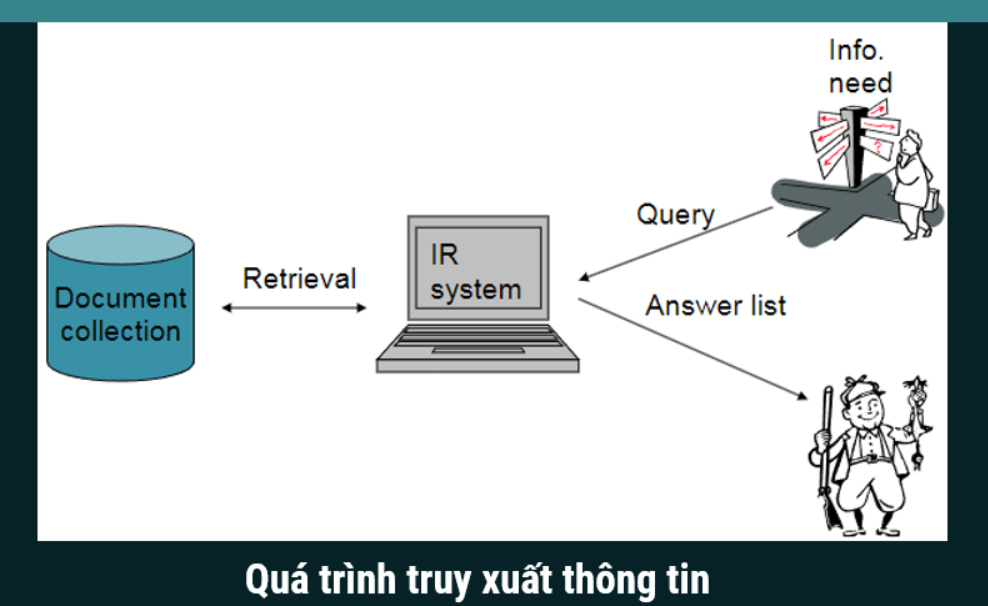
Là 1 bước trích xuất các tính năng chính bổ sung trong dữ liệu. Trích xuất đối tượng sử dụng một số kỹ thuật, chẳng hạn như ánh xạ, lọc và phân cụm. Lợi ích chính của việc sử dụng tính năng trích xuất là - nó giúp loại bỏ dữ liệu dư thừa và cải thiện tốc độ phát triển mô hình

* **Gắn thẻ dữ liệu vào danh mục được xác định trước (Data Categorization)**

Gắn thẻ văn bản vào các danh mục được xác định trước là bước cuối cùng trong phân loại văn bản. Nó có thể được thực hiện theo ba cách khác nhau,

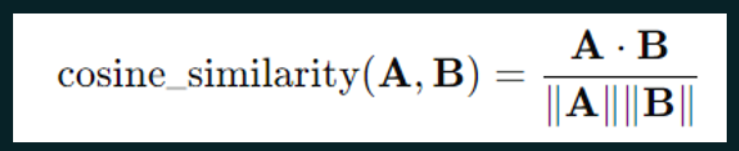
* + Gắn thẻ thủ công
  + Đối sánh dựa trên quy tắc
  + Thuật toán học tập - Các thuật toán học tập có thể được phân loại thành hai loại như gắn thẻ được giám sát và gắn thẻ không được giám sát.

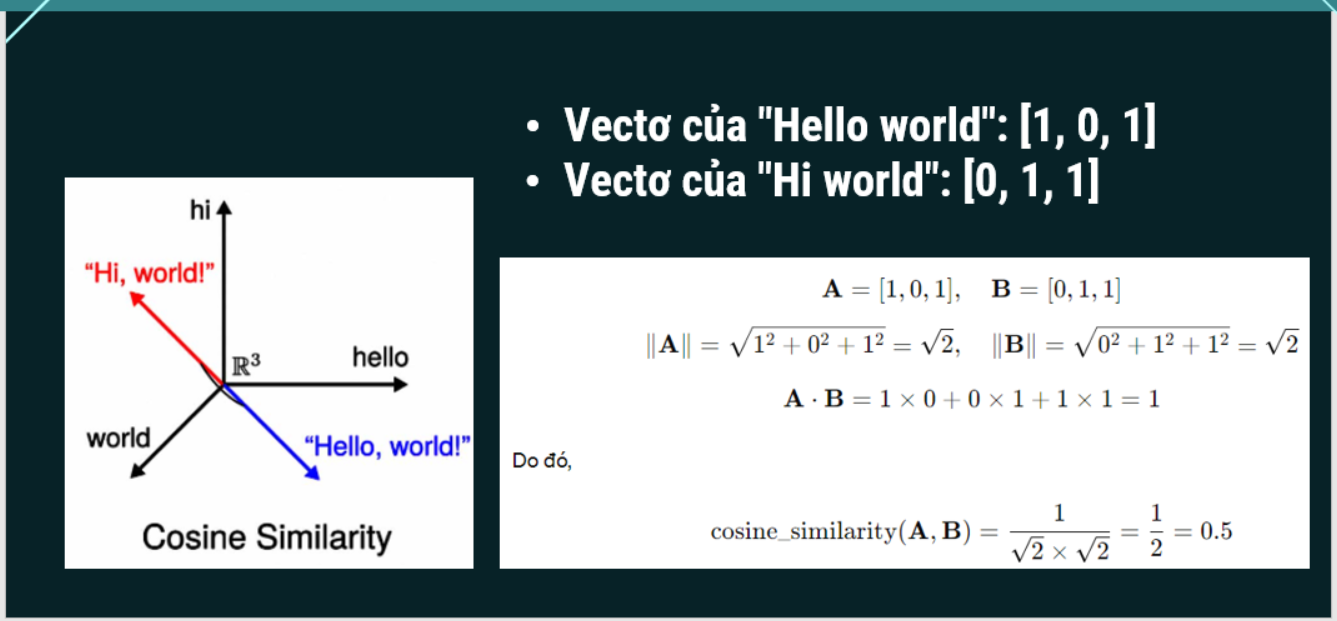
**Information Retrieval**

**Information Retrieval (IR)** là quá trình tìm kiếm, truy xuất và truy cập thông tin từ các nguồn dưới dạng không có cấu trúc (thường là văn bản). Mục tiêu của IR là tự động tìm kiếm và trả về các tài liệu phù hợp nhất với yêu cầu của người dùng

**IR scoring functions**

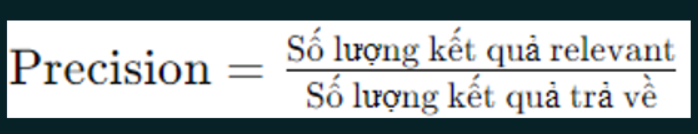
* **IR scoring functions** được sử dụng để đánh giá mức độ liên quan hoặc sự tương đồng giữa một truy vấn và các tài liệu trong một hệ thống tìm kiếm, sau đó sắp xếp các tài liệu theo thứ tự giảm dần của điểm số để trả về kết quả tìm kiếm cho người dùng.
  + Cosine Similarity
  + TF-IDF (Term Frequency-Inverse Document Frequency)
  + BM25 (Best Matching 25)
  + Okapi BM25
  + Language Models
* **Cosine Similarity** là một trong những phương pháp phổ biến nhất, cosine similarity đo độ tương tự giữa hai vectơ trong không gian vector bằng cách tính cosin của góc giữa chúng. Trong IR, các vectơ biểu diễn của truy vấn và tài liệu được sử dụng, và cosine của góc giữa chúng đánh giá mức độ tương đồng giữa chúng.



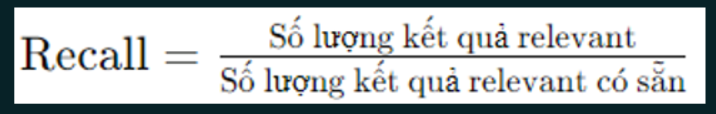


**IR system evaluation**

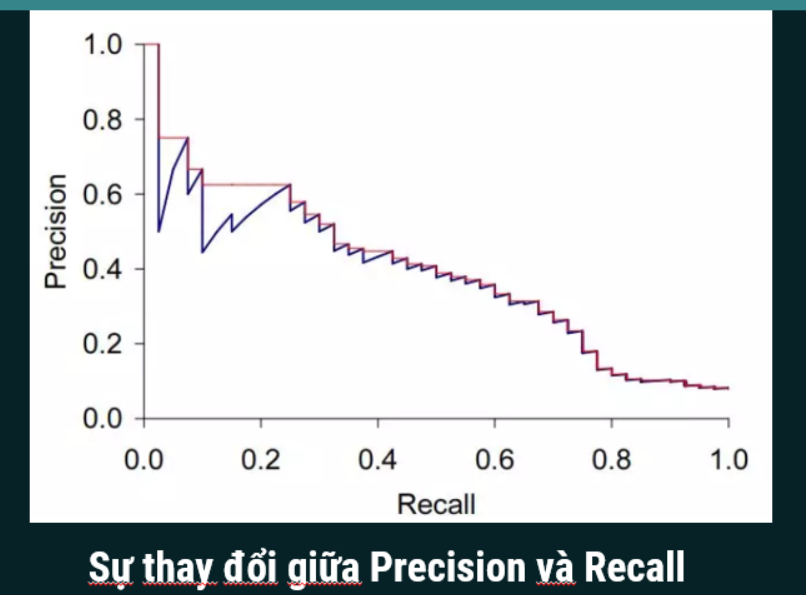
* **IR system evaluation** là quá trình đo lường hiệu suất và chất lượng của hệ thống tìm kiếm trong việc truy xuất thông tin từ một tập hợp các tài liệu dựa trên các truy vấn người dùng. Các phương pháp đánh giá IR được phát triển để đo lường khả năng của hệ thống IR trong việc đáp ứng các yêu cầu của người dùng.
* Hai độ đo cơ bản:
  + **Precision (Độ chính xác)** đo lường tỷ lệ của các kết quả có liên quan (relevant) trong số tất cả các kết quả được hệ thống trả về



* + **Recall (Độ phủ)** đo lường tỷ lệ của các kết quả có liên quan (relevant) trong số tất cả các kết quả có liên quan có sẵn trong cơ sở dữ liệu.



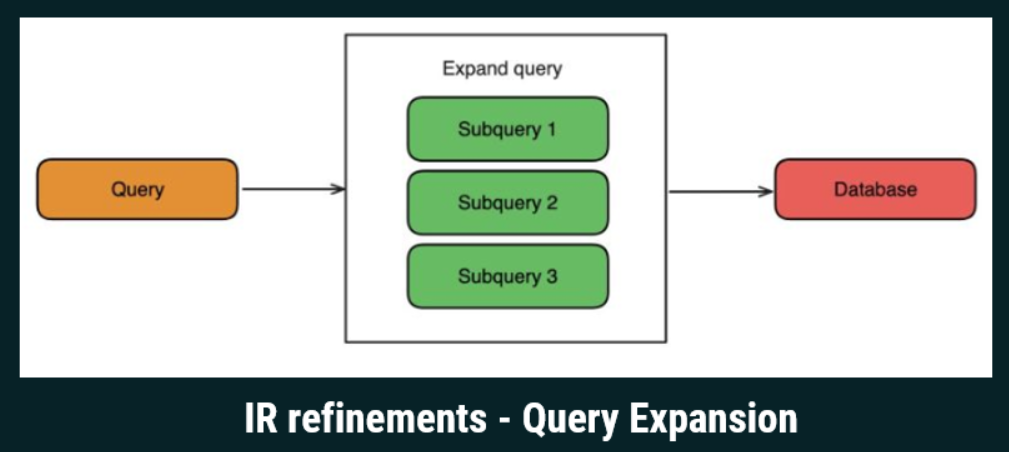
Trong quá trình đánh giá, Precision và Recall thường được sử dụng cùng nhau để cung cấp cái nhìn toàn diện về hiệu suất của hệ thống IR. Tuy nhiên, có một sự đánh đổi giữa Precision và Recall: khi một chỉ số tăng lên, chỉ số kia thường giảm xuống và ngược lại. Điều này đặc biệt quan trọng trong các tình huống mà việc tìm kiếm các kết quả chính xác là quan trọng hơn so với việc bao quát tất cả các kết quả đúng có sẵn.



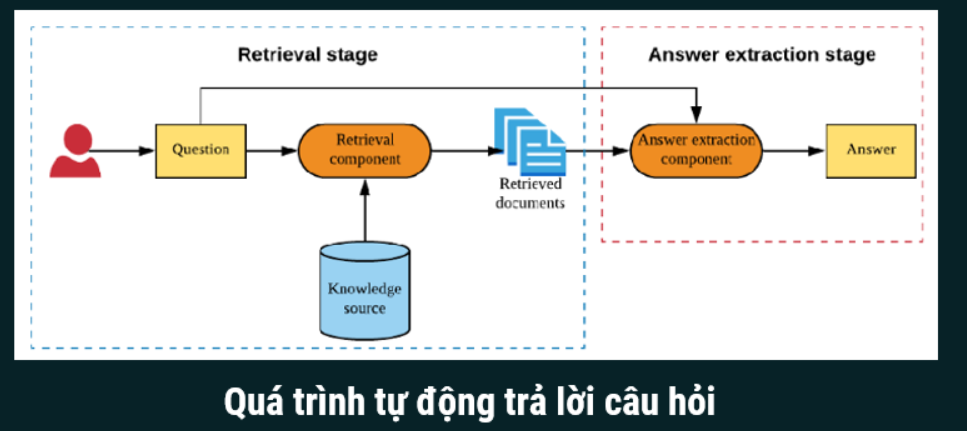
**IR refinements**

Là các cải tiến và điều chỉnh được áp dụng để cải thiện hiệu suất và chất lượng của hệ thống tìm kiếm thông tin

* **Query Expansion** được sử dụng để cải thiện hiệu suất của hệ thống tìm kiếm bằng cách tự động mở rộng hoặc bổ sung truy vấn ban đầu của người dùng với các từ khóa hoặc cụm từ liên quan. Mục tiêu của kỹ thuật này là tăng khả năng tìm kiếm và đảm bảo rằng các kết quả trả về là đầy đủ và phù hợp hơn với nhu cầu của người dùng.



* **Question answering** là quá trình tự động trả lời các câu hỏi của người dùng dựa trên thông tin có sẵn trong các nguồn dữ liệu văn bản. Trong QA, hệ thống được yêu cầu xử lý và hiểu câu hỏi của người dùng, tìm kiếm thông tin từ các nguồn dữ liệu khác nhau và trả về câu trả lời chính xác và có ý nghĩa.



**Information Extraction**

**Information extraction (Rút trích thông tin)** là quá trình thu thập thông tin bằng cách **đọc lướt đoạn văn và tìm kiếm sự xuất hiện** của một lớp đối tượng, những mối quan hệ giữa các đối tượng với nhau và **chuyển dữ liệu không cấu trúc thành dữ liệu có cấu trúc** và có thể chỉnh sửa.

**Finite-state automata for information extraction**

Finite-state automata (FSA), còn được gọi là máy trạng thái hữu hạn (FSM), là **mô hình tính toán** được sử dụng để **biểu diễn** và điều khiển các hệ thống có số trạng thái hữu hạn. Các máy này hoạt động bằng cách **chuyển đổi giữa các trạng thái dựa trên các ký tự đầu vào (biểu thức chính quy)**

* **Ưu điểm**: Phương pháp chỉ hoạt động tốt **trên một miền cụ thể**, trong đó có thể xác định trước được đối tượng cần rút trích
* **Nhược điểm**: khi **dữ liệu đầu vào đa dạng** thì phương pháp này **hoạt động kém**

**Probabilistic models for information extraction**

* Để khắc phục nhược điểm của FSA, thì ta có mô hình xác suất tuần tự với trạng thái ẩn là **mô hình ẩn Markov** (HMM là mô hình thống kê trong đó hệ thống được mô hình hóa như **một quá trình Markov** với các **tham số không biết trước**)
* Mô hình Markov là **một mô hình ngẫu nhiên được sử dụng để mô hình hóa các hệ thống thay đổi ngẫu nhiên giả.**
* **Ưu điểm**
  + HMM là **mô hình tuyến tính**, và chịu được sự nhiễu thông tin. Trong biểu thức chính quy, nếu 1 ký tự đơn mong muốn mất, thì thì biểu thức sẽ không khớp. Còn với HMMs, khi thiếu ký tự hoặc chữ thì sẽ nhận được **xác suất mức độ khớp**
  + HMMs có thể được **train từ dữ liệu**; chúng **không yêu cầu kỹ thuật tốn nhiều của mẫu,** như vậy chúng có thể **dễ dàng bắt kịp sự thay đổi mới nhất của văn bản** theo thời gian
* **Nhược điểm**: Có 1 vấn đề cho việc rút trích thông tin với HMMs là **chúng mô phỏng nhiều xác suất không cần thiết.**

**Conditional random fields for information extraction**

CRF **mô hình hóa 1  phân phối xác suất có điều kiện** của 1 tập các biến mục tiêu được đưa ra bởi 1 tập biến đã được quan sát.

**Ontology extraction from large corpora**

**Ontology extraction** từ các bộ dữ liệu lớn là **quá trình tự động nhận diện và xây dựng ontology** từ các nguồn dữ liệu lớn như văn bản, cơ sở dữ liệu, tập tin log, hoặc dữ liệu web. Mục tiêu của quá trình này là tạo ra **một bản mô tả tự động và có cấu trúc của kiến thức trong lĩnh vực cụ thể** từ các nguồn dữ liệu không cấu trúc hoặc bán cấu trúc.

* **Bước 1: Thu thập dữ liệu**:
  + Thu thập dữ liệu từ nhiều nguồn khác nhau như văn bản, trang web, cơ sở dữ liệu, tập tin log, v.v. Đối với mục đích ontology extraction, các nguồn dữ liệu này thường là các văn bản lớn trong lĩnh vực cụ thể như y học, kỹ thuật, hoặc kinh doanh.
* **Bước 2: Tiền xử lý dữ liệu:**
  + Tiền xử lý dữ liệu để loại bỏ dữ liệu không cần thiết, tiêu chuẩn hóa dữ liệu và chuyển đổi dữ liệu thành định dạng có thể xử lý. Điều này có thể bao gồm loại bỏ các từ không cần thiết, chấm câu, các ký tự đặc biệt, và việc chuyển đổi văn bản thành các đối tượng dữ liệu có cấu trúc hơn như từ điển, cây từ, v.v.
* **Bước 3: Phân tích dữ liệu:**
  + Sử dụng các **kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và học máy** để phân tích dữ liệu và **nhận diện các khái niệm, mối quan hệ và thuộc tính** từ dữ liệu.
  + Các kỹ thuật phổ biến bao gồm **đánh giá tần suất xuất hiện của từ,** phân tích **cú pháp**, phân loại **văn bản** và trích xuất thông tin cấu trúc.
* **Bước 4: Xây dựng ontology**:
  + Dựa trên các khái niệm, **mối quan hệ và thuộc tính** đã được phân tích, **xây dựng ontology bằng cách tự động tạo các mục nhập ontology và quan hệ giữa chúng.**
  + Các **mục nhập ontology có thể là các thực thể,** các **loại thực thể,** các mối **quan hệ giữa chúng**, và các thuộc tính của các thực thể.
* **Bước 5: Kiểm tra và cập nhật**:
  + Kiểm tra và cập nhật ontology để đảm bảo rằng nó đáp ứng được nhu cầu và đáng tin cậy trong quá trình sử dụng.
  + Điều này có thể bao gồm việc xác định và loại bỏ các mục không cần thiết, điều chỉnh quan hệ giữa các mục và cập nhật ontology với dữ liệu mới.

**Automated template construction**

* Automated template construction (xây dựng mẫu tự động) là quá trình tạo ra các mẫu hoặc mẫu cấu trúc tự động từ dữ liệu đầu vào. Mục tiêu chính của quá trình này là tạo ra các mẫu có thể được sử dụng để tổ chức thông tin hoặc tạo ra nội dung mới dựa trên dữ liệu có sẵn.
* Nhược điểm lớn của phương pháp này là nếu mẫu dữ liệu đầu vào không đúng thì các mẫu mà được tạo ra cũng không đúng

**Machine reading**

* **Machine reading** là quá trình mà máy tính được lập trình để hiểu và rút trích thông tin từ văn bản tự nhiên. Mục tiêu của machine reading là biến dữ liệu văn bản không cấu trúc thành dữ liệu có ý nghĩa và có thể sử dụng được cho các mục đích khác nhau như phân tích, tóm tắt, trả lời câu hỏi, và hơn thế nữa.
* **Quá trình:**
  + **Xác định mục tiêu:** Xác định các loại thông tin cụ thể mà bạn muốn máy tính trích xuất từ văn bản.
  + **Tiền xử lý dữ liệu**: Loại bỏ dữ liệu không cần thiết như ký tự đặc biệt, chấm câu, v.v. Tiêu chuẩn hóa dữ liệu để đảm bảo nhất quán trong quá trình xử lý.
  + **Phân tích ngôn ngữ tự nhiên (NLP):** Sử dụng các công cụ NLP để phân tích cú pháp và ngữ nghĩa của văn bản. Điều này bao gồm việc phân tích cú pháp để hiểu cấu trúc của câu và phân tích ngữ nghĩa để hiểu ý nghĩa của từ và cụm từ.
  + **Trích xuất thông tin:** Sử dụng các thuật toán trích xuất thông tin để rút trích thông tin có ý nghĩa từ văn bản. Các phương pháp có thể bao gồm trích xuất từ khoá, phân tích cú pháp để xác định các mối quan hệ giữa các thành phần của văn bản, và sử dụng mô hình học máy để dự đoán các thông tin cụ thể.
  + **Xây dựng thông tin cấu trúc:** Tổ chức thông tin được trích xuất thành các dạng có cấu trúc như bảng hoặc các dạng dữ liệu có thể sử dụng trong các hệ thống thông tin hoặc hệ thống quản lý tri thức. (ontology)

**Conversational agent (Chatbot)**

* 1 ứng dụng phổ biến được tạo nên từ sự kết hợp AI và xử lý ngôn ngữ tự nhiên là Chatbot
* **Chatbot là** một chương trình kết hợp với trí tuệ nhân tạo được thiết kế để tương tác với con người thông qua các cuộc trò chuyện
* **Cách hoạt động**
  + **Translator**: Thông tin yêu cầu của người dùng sau khi được nhập vào nền tảng chat trung gian sẽ được dịch lại theo ngôn ngữ lập trình để giúp máy tính có thể hiểu được các thông tin cần thực hiện.
  + **Processor**: Lúc này, công nghệ AI kết hợp với xử lý ngôn ngữ tự nhiên sẽ xử lý yêu cầu của người dùng.
  + **Respondent**: Máy tính sẽ nhận kết quả từ AI và trả về cho người dùng câu trả lời tương ứng thông qua nền tảng chat trung gian
* **Nội dung của câu trả lời có thể được lấy từ các nguồn:**
  + Bộ kịch bản đã được soạn sẵn
  + Thông tin theo ngữ cảnh dựa trên các dữ liệu thực tế được cung cấp
  + Database có sẵn trong kho lưu trữ từ trước của doanh nghiệp
* **Hệ thống chatbox Có thể chia 2 loại:**
  + **Task-Oriented chatbot**

mục tiêu chính của các chatbot này là hướng người dùng đến một mục đích hoặc mục tiêu cụ thể thông qua các cuộc trò chuyện hoặc tương tác. Các task-oriented chatbot thường được lập trình để hiểu và thực hiện các yêu cầu hoặc câu hỏi trong một lĩnh vực hoặc ngữ cảnh nhất định. Chẳng hạn, một task-oriented chatbot trong lĩnh vực y tế có thể được thiết kế để hỗ trợ bác sĩ trong việc đặt lịch hẹn, cung cấp thông tin về bệnh lý, hoặc theo dõi triệu chứng của bệnh nhân.

VD: Một số chatbot nổi tiếng thuộc loại task-oriented chatbot bao gồm:1. \*\*Ada (Ada Support):\*\* Ada là một chatbot hỗ trợ khách hàng tự động, giải quyết vấn đề và cung cấp thông tin về sản phẩm hoặc dịch vụ.3. \*\*Amelia (IPsoft):\*\* Amelia là một chatbot AI đa năng, hỗ trợ khách hàng đến quản lý tài nguyên nhân sự.4. \*\*IBM Watson Assistant:\*\* Watson Assistant của IBM là một 5. \*\*Salesforce Einstein Bot:\*\* Einstein Bot là một chatbot được tích hợp trong nền tảng Salesforce

* + **Non Task-Oriented chatbot:** Là những hệ thống được thiết kế để xử lý các đoạn hội thoại mở rộng, không có cấu trúc, tập trung chủ yếu vào bắt chước càng giống người càng tốt.Non-task oriented chatbot không được thiết kế để xử lý bất kỳ nhiệm vụ cụ thể nào. Thay vào đó, nó rất hữu ích khi trả lời các câu hỏi, truy vấn từ người dùng. Đó là những truy vấn “bất kỳ”, “ngẫu nhiên” như “thời tiết thế nào” sau đó là “tin thị trường tài chính mới nhất”. Không tồn tại bất kỳ một bối cảnh cụ thể nào ở đây. Chatbot non task-oriented phải có khả năng truy xuất dữ liệu từ các nguồn trực tuyến để giao tiếp hiệu quả.

VD: CHATGPT: ChatGPT có khả năng tạo ra các cuộc trò chuyện phong phú, đa dạng về nội dung và chủ đề, và thích ứng với các yêu cầu và câu hỏi của người dùng một cách linh hoạt.

ChatGPT là một loại non-task-oriented chatbot, với các phiên bản được phát triển bởi OpenAI. Được xây dựng trên nền tảng của mạng nơ-ron GPT (Generative Pre-trained Transformer), ChatGPT được huấn luyện để có khả năng tương tác tự nhiên với người dùng thông qua việc tạo ra các cuộc trò chuyện phong phú và đa dạng về nội dung.

ChatGPT không được thiết kế để thực hiện các nhiệm vụ cụ thể hoặc giải quyết các vấn đề nhất định. Thay vào đó, nó được tạo ra để cung cấp trải nghiệm gần giống như việc trò chuyện với một người đồng nghiệp thông qua các cuộc trò chuyện mở rộng và không có mục tiêu cụ thểMục tiêu của ChatGPT là cung cấp một trải nghiệm tương tác tự nhiên và linh hoạt cho người dùng, giúp họ tương tác với một hệ thống AI một cách thoải mái và hiệu quả.

VD: Trí tuệ nhân tạo ngày nay đã được tích hợp vào các thiết bị và ứng dụng. Một trong số đó là Mitsuku . Đó là một chatbot AI đã giành được Bài kiểm tra Turing của Giải thưởng Loebner. Đúng như quảng cáo, đây là chatbot trò chuyện tốt nhất thế giới.

Khi tôi thử trò chuyện với Mitsuki , phản hồi của cô ấy (bot) rất tốt như một con người. Trong khi trò chuyện, tôi có cảm giác như đang nói chuyện với một người bạn. Nó chọn những từ tương tự như phản ứng thông thường của con người thực sự.

* Từ 2 loại Chatbot kể trên mà người ta có khái niệm về **Open Domain** và **Closed Domain**.
  + **Open Domain**,người dùng có thể đưa cuộc hội thoại đến bất cứ đâu. Không có một chủ đề hay một mục đích cố định nào
  + **Closed Domain**, inputs và outputs được cho là có hạn bởi vì hệ thống chỉ chú tâm vào giải quyết một mục tiêu cụ thể.
* **Chatbox framework**
  + Chatbot sử dụng các frameworks như: Chatfuel, ManyChat, FlowXO, Octane, Recime
  + Có thể sử dụng các APIs, services được cung cấp bởi các công ty công nghệ hàng đầu
* **Chatbot models nhìn chung chia ra ít nhất làm 2 loại là Retrieval-based models và Generative models.**
  + **Retrieval-based model:**

Những chatbot này chọn phản hồi phù hợp nhất dựa trên dữ liệu học tập của chúng. Các chatbot dựa trên truy xuất sử dụng các phản hồi được xác định trước, xử lý đầu vào, nhận dạng ý định, lựa chọn phản hồi, quản lý bối cảnh và các câu trả lời dự phòng cùng nhiều thứ khác. Họ cũng tận dụng các hệ thống như Hiểu ngôn ngữ tự nhiên (NLU) để tạo ra phản hồi giống con người.

* + - **Ưu điểm:**
      * Không xảy ra các lỗi về ngữ pháp vì đáp án được lưu sẵn.
      * Ít rủi ro
      * Chi phí cố định
      * Có khả năng Tích hợp
    - **Nhược điểm**:
      * Tính linh hoạt hạn chế
      * Thời gian đào tạo lâu hơn
      * Dễ bị lặp lại
  + **Generative models: (Bot AI sáng tao):**

Các chatbot AI sáng tạo tạo ra phản hồi mới từ đầu dựa trên dữ liệu đào tạo của chúng.  Các bot tận dụng Mô hình ngôn ngữ lớn (LLM). LLM phổ biến nhất vào năm 2023 là ChatGPT

* + - **Ưu điểm:**
      * Linh hoạt hơn
      * Mang tính trò chuyện nhiều hơn
      * Đào tạo nhanh hơn
    - **Nhược điểm:**
      * Không dễ để implement và output thì có thế không chính xác (sai lỗi ngữ pháp, sai về mặt ngữ nghĩa)
      * Cần nhiều tài nguyên
      * Chi phí khá cao
      * Cần rất nhiều dữ liệu để huấn luyện
      * Không có khả năng tích hợp

=> Chatbot là một nhân tố quan trọng trong trong tương lai sắp tới, nơi mà máy móc trở nên thông minh và hiểu con người hơn. Cho dù các ứng dụng vẫn còn hạn chế trên một số lĩnh vực nhất định, tuy nhiên chắc chắn vai trò của bot sẽ ngày càng tăng cao, với nhiều ứng dụng hơn vào các bài toán trong đời sống.

1. **Deep Learning**

**Deep Learning là gì?**

* Deep Learning có thể được xem là một lĩnh vực con của Machine Learning - ở đó các máy tính sẽ học và cải thiện chính nó thông qua các thuật toán.
* Deep Learning được xây dụng dựa trên các khái niệm phức tạp hơn rất nhiều, chủ yếu hoạt động với các mạng nơ-ron nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người.
* Mạng nơ-ron nhân tạo là động lực chính để phát triển Deep Learning. Các mạng nơ-ron sâu (DNN) bao gồm nhiều lớp nơ-ron khác nhau, có khả năng thực hiện các tính toán có độ phức tạp rất cao.
* Deep Learning hiện đang phát triển rất nhanh và được xem là một trong những bước đột phá lớn nhất trong Machine Learning.

**Deep Learning hoạt động như thế nào?**

* **Xây dựng mạng nơ-ron**: Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, số lượng layer càng nhiều thì mạng sẽ càng “sâu”. Trong mỗi layer là các nút mạng (node) và được liên kết với những lớp liền kề khác. Mỗi kết nối giữa các node sẽ có một trọng số tương ứng, trọng số càng cao thì ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.
* **Huấn luyện mạng nơ-ron**: Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm ra bộ giá trị của trọng số sao cho phán đoán là tốt nhất.
* **Dự đoán và phân loại**: Các hệ thống Deep Learning yêu cầu phần cứng phải rất mạnh để có thể xử lý được lượng dữ liệu lớn và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất nhiều tuần, thậm chí nhiều tháng để triển khai trên những phần cứng tiên tiến nhất hiện nay.

**Ưu điểm và nhược điểm**

**Ưu điểm:**

* Tự động học cao:
  + Tự điều chỉnh và tự tối ưu
  + Giảm sự phụ thuộc vào con người
* Giải quyết vấn đề phức tạp
  + Với độ chính xác rất cao
  + Đem lại ứng dụng mạnh mẽ
* Xử lý nhiều loại dữ liệu
  + Tuần tự
  + Cấu trúc và phi cấu trúc
* Khả năng mở rộng
  + Đáp ứng được khối lượng dữ liệu ngày càng tăng
  + Mang lại sự linh hoạt và khả năng truy cập

**Nhược điểm:**

* Yêu cầu dữ liệu lớn
* Chi phí tính toán cao
* Dễ bị Overfitting
* Thiếu nền tảng lý thuyết

**Ứng dụng của Deep Learning trong đời sống**

* Nhận dạng hình ảnh
* Xe tự động
* Trợ lý ảo
* Nghiên cứu y học
* Mạng xã hội

**Khi nào nên sử dụng Deep Learning**

* Mục tiêu và độ phức tạp của dự án
* Nguồn dữ liệu
* Khối lượng dữ liệu lớn

**Có nên sử dụng Deep Learning thay cho Maching Learning**

* Độ phức tạp và mục tiêu của dự án
* Tài nguyên tính toán
* Số lượng dữ liệu
* Tính tương thích với bài toán

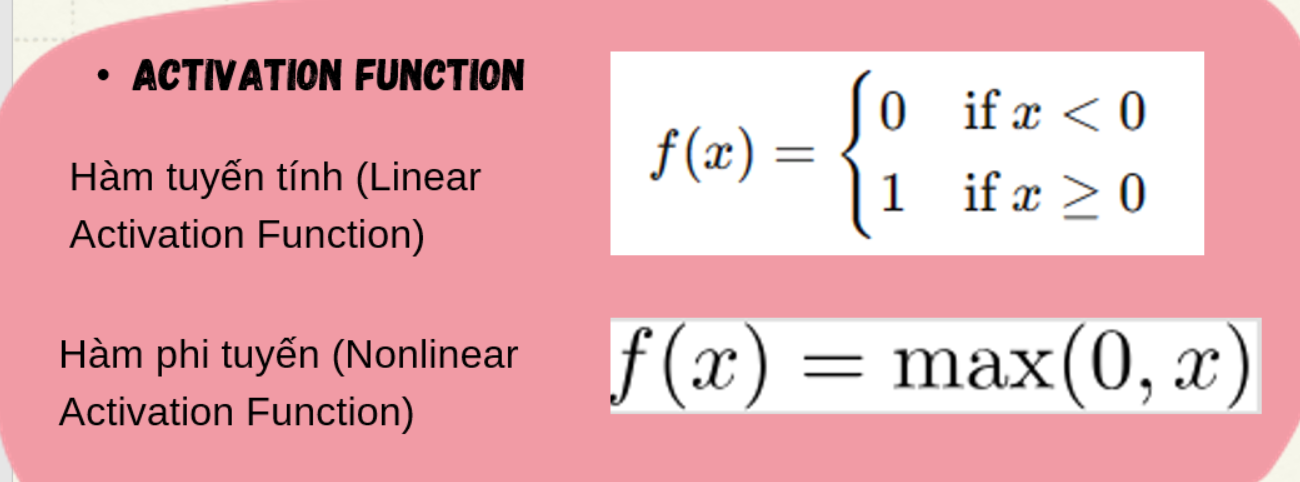
**Deep Learning Techniques**

1. **Classical Neural Network**

* Mạng nơ-ron cổ điển được thiết kế bởi Fran Rosenblatt vào năm 1957, chủ yếu được sử dụng cho các bài toán phân lớp nhị phân
* Kiến trúc cổ điển của mạng nơ-ron là mạng kết nối đầy đủ, thường được xác định bằng các perceptron đa lớp
* Perceptron
  + Input Binary
  + Output Binary
  + Weights
  + Activation Function
  + Threshold

1. **Convolutional Neural Network – CNN**

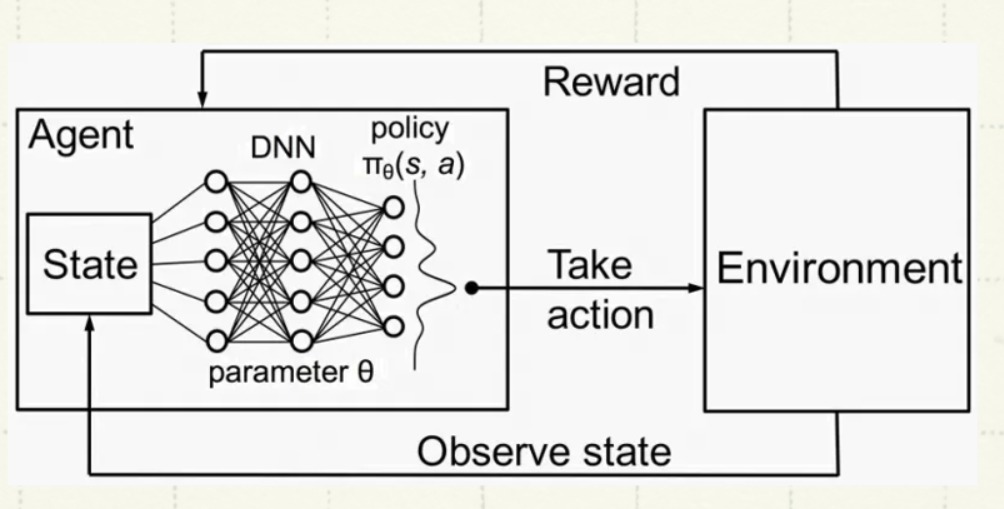
* Thuật toán CNN có thể giải quyết tốt với kiểu dữ liệu không có cấu trúc như hình ảnh, âm thanh, videos,…
* **Tại sao chúng ta lại cần CNN?**
  + Ảnh có thể rất lớn
  + Vị trí vật thể bất biến
  + Tập trung vào điểm quan trọng
* **Convolutional**
  + Bộ lọc di chuyển qua toàn bộ input theo bước nhảy. Tích chập giữ bộ lọc và đầu vào
  + Đầu ra là feature map biểu diễn các đặc trưng cụ thể trong hình ảnh
* **Kenrel (Filter)**
  + Mỗi Filter sẽ tạo ra một ảnh có đặc điểm khác nhau
  + Filter được huấn luyện khi kích hoạt một đặc điểm ảnh
* **Padding**
  + Tránh việc giảm kích thước output khi thực hiện tích chập
  + Thêm dòng hoặc cột xung quanh input
* **Strides**
  + Biểu diễn bước đi khi kernel thực hiện trượt trên input
  + Giảm số chiều để tìm những đặc trưng trừu tượng
* **Activation Function**



* **Pooling**
  + Gộp cực đại và gộp trung bình
  + Giảm số chiều về mặt không gian
  + Translation invariance

1. **Deep Reinforcement Learning**

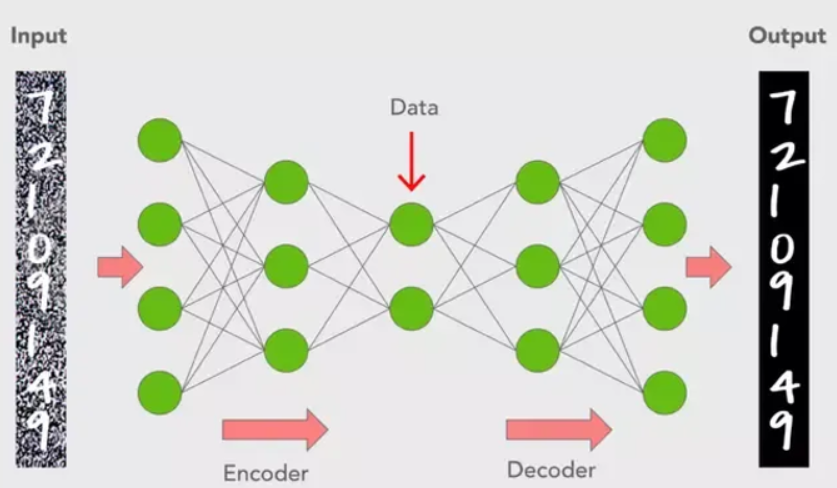
* Học tăng cường (Reinforcement Learning – RL) ra đời từ những năm 1980
* DRL là một nhánh của Học Máy (Machine Learning) kết hợp Học Tăng Cường (Reinforcement Learning) với Học Sâu (Deep Learning)
* DRL cho phép các tác nhân (agent) học cách tương tác với môi trường để đạt được mục tiêu cụ thể
  + Robot học
  + Trò chơi điện tử
  + Tài chính
  + Y tế
* Các thuật toán DRL phổ biến
  + Q-Learning
  + Deep Q-Network (DQN)
  + Policy Gradient
  + Actor-Critic



* **Ưu điểm và nhược điểm**
  + **Ưu điểm**
    - Khả năng học tập
    - Khả năng giải quyết các bài toán
    - Khả năng tự động điều chỉnh hành động
  + **Nhược điểm**
    - Lượng dữ liệu lớn
    - Tài nguyên tính toán mạnh mẽ
    - Giải tích và kiểm soát

1. **Autoencoder**

* Autoencoder là một trong những kỹ thuật Deep Learning phổ biến nhất hiện nay
* Một số loại autoencoder chính gồm có:
  + Sparse
  + Denoising
  + Contractive
  + Stacked



* **Ứng dụng và khó khăn** 
  + **Ứng dụng** 
    - Giảm nhiễu dữ liệu
    - Phát hiện dị thường
    - Hệ thống đề xuất
    - Tóm tắt hình ảnh
  + **Khó khăn**
    - Lựa chọn kiến trúc mạng nơ-ron phù hợp
    - Biểu diễn trung gian được học