

# 1

## Giới thiệu về Xử lý ngôn ngữ tự nhiên

*Dave Bowman: Open the pod bay doors, HAL.*

*HAL: I'm sorry Dave, I'm afraid I can't do that.*

Stanley Kubrick và Arthur C. Clarke,  
cảnh trong phim *2001: A Space Odyssey*

Ý tưởng về việc máy tính có khả năng xử lý ngôn ngữ tự nhiên đã đến từ rất lâu ngay từ khi máy tính mới xuất hiện. Quyển sách này nói về các phương pháp và sự thực hiện ý tưởng thú vị đó. Chúng tôi giới thiệu về một lĩnh vực nghiên cứu sôi động, liên ngành với rất nhiều tên gọi khác nhau, như **xử lý ngôn ngữ và tiếng nói**, **công nghệ ngôn ngữ con người**, **xử lý ngôn ngữ tự nhiên**, **ngôn ngữ học tính toán**, và **tổng hợp và nhận dạng tiếng nói**. Mục tiêu của lĩnh vực này là giúp máy tính có thể thực hiện các nhiệm vụ liên quan đến ngôn ngữ của con người, những nhiệm vụ có sự tương tác giữa máy tính-con người, cải thiện giao tiếp con người-con người, hoặc đơn giản xử lý văn bản và tiếng nói.

Một ví dụ cho lĩnh vực này là các **máy giao tiếp**. Chiếc máy tính HAL 9000 trong phim *2001: A space Odyssey* của Stanley Kubrick là một trong những nhân vật đáng chú ý trong điện ảnh thế kỷ hai mươi. HAL là một chiếc máy thông minh có khả năng xử lý ngôn ngữ tiên tiến như hiểu và nói được tiếng Anh, và thậm chí còn đọc môi. Đến bây giờ, có thể thấy tác giả Clarke đã hơi lạc quan khi dự đoán rằng một chương trình thông minh như HAL có thể thành hiện thực. Nhưng trí tưởng tượng của ông đã đi xa như thế nào? Chúng ta gọi những chương trình như HAL là các **máy giao tiếp** hay các **hệ thống hội thoại**. Trong quyển sách này, chúng ta cùng tìm hiểu về các thành phần để làm nên một chương trình hội thoại hiện đại, gồm có xử lý ngôn ngữ đầu vào (**nhận dạng tiếng nói** hay **lĩnh vực hiểu ngôn ngữ**) và xử lý ngôn ngữ đầu ra (**sinh ngôn ngữ tự nhiên** hay **tổng hợp tiếng nói**).

Hãy chuyển qua một nhiệm vụ hữu ích khác, đã giúp cho rất nhiều người không

nói tiếng Anh có thể tiếp cận được với những thông tin khoa học khổng lồ trên các trang web tiếng Anh. Hay dịch cho những người nói tiếng Anh hàng trăm triệu trang web viết bằng các ngôn ngữ khác như tiếng Trung Quốc. Mục tiêu của **dịch máy** là tự động dịch một văn bản từ ngôn ngữ này sang ngôn ngữ khác. Chúng tôi sẽ giới thiệu các thuật toán và các công cụ toán học cần thiết để hiểu một chương trình dịch máy hiện đại hoạt động như thế nào. Dịch máy là một vấn đề còn gặp nhiều thách thức, chúng tôi sẽ giới thiệu các thuật toán được sử dụng trong lĩnh vực này, cũng như các thành phần chính.

Rất nhiều nhiệm vụ xử lý ngôn ngữ liên quan đến Web. Một trong đó là **hệ thống hỏi đáp dựa trên nền Web**. Ví dụ như web tìm kiếm, khi người dùng có thể gõ một vài từ khóa để tìm kiếm thông tin, hay có thể hỏi một câu trọn vẹn, từ đơn giản đến phức tạp:

- "Bất đồng" có nghĩa là gì?
- Abraham Lincoln sinh năm nào?
- Ở Mỹ có bao nhiêu bang trong năm đó?
- Người Trung Quốc xuất khẩu bao nhiêu lụa sang Anh cho đến hết thế kỷ 18?
- Các nhà khoa học nghĩ gì về vấn đề đạo đức trong nhân bản người?

Một vài câu hỏi ở trên là câu hỏi **rõ ràng**, hoặc đơn giản là câu hỏi **dữ liệu thực tế** như ngày tháng hay địa điểm, có thể trả lời bằng cách sử dụng các máy tìm kiếm. Nhưng trả lời các câu hỏi phức tạp hơn có thể cần phải trích xuất thông tin trong các bản, hay phải thực hiện **suy diễn** (đưa ra kết luận từ các sự kiện), hoặc cần tóm tắt và tổng hợp thông tin từ nhiều người. Trong quyển sách này, chúng ta nghiên cứu nhiều thành phần tạo nên các hệ thống hiểu ngôn ngữ hiện đại, như **trích rút thông tin, phân giải nhập nhằng nghĩa từ**.

Mặc dù các lĩnh vực con và các vấn đề chúng tôi vừa nêu còn rất xa mới giải quyết được triệt để, chúng là các lĩnh vực nghiên cứu sôi động và có nhiều kỹ thuật đã có các ứng dụng công nghiệp. Trong phần còn lại của chương này, chúng tôi trình bày tóm tắt những loại kiến thức cần cho các nhiệm vụ này (và những nhiệm vụ khác như **sửa lỗi chính tả, kiểm tra ngữ pháp**), cũng như các mô hình toán học sẽ được giới thiệu trong sách.

## 1.1. TRI THỨC TRONG XỬ LÝ TIẾNG NÓI VÀ NGÔN NGỮ

Sự khác biệt giữa những ứng dụng xử lý ngôn ngữ và những ứng dụng xử lý dữ liệu thông thường là việc sử dụng *tri thức về ngôn ngữ*. Như chương trình Unix `wc`,

được sử dụng để đếm số lượng ký tự, từ và dòng trong một file văn bản. Khi dùng để đếm ký từ và dòng, `wc` là một chương trình xử lý dữ liệu bình thường. Tuy nhiên, khi nó được dùng để đếm số từ, nó cần biết được *tri thức về thế nào là một từ*, từ đó trở thành một chương trình xử lý ngôn ngữ.

Tất nhiên, `wc` là một hệ thống rất đơn giản với sự hiểu biết rất hạn chế về tri thức của ngôn ngữ. Những hệ thống giao tiếp phức tạp như HAL, hay hệ thống thông dịch máy, hay các hệ thống hỏi đáp, yêu cầu tri thức về ngôn ngữ rộng và sâu hơn rất nhiều. Để lường tượng được phạm vi và loại tri thức cần thiết, hãy xem xét những gì mà HAL cần biết để tham gia vào cuộc hội thoại như ở ví dụ đầu trường, hoặc trong trường hợp hệ hỏi đáp, hệ thống có thể trả lời được một trong những câu hỏi kể trên.

HAL cần phải nhận ra các từ từ tín hiệu âm thanh, sau đó sinh ra âm thanh từ một chuỗi từ. Các nhiệm vụ như nhận dạng tiếng nói và tổng hợp tiếng nói yêu cầu hiểu biết về ngữ âm và âm vị học; các từ được phát âm thế nào, và các âm được nhận ra như thế nào.

Chú ý rằng không giống như Star Trek's Commander Data, HAL có khả năng nói các câu phủ định như *can't*. Để có thể nói và tiếp nhận những điều này hay những điều khác (như nhận ra *doors* là số nhiều) đòi hỏi hiểu biết về **ngôn ngữ hình thái học**, các từ được tách thành các thành phần mang nghĩa như *số ít* hay *số nhiều*.

Chuyển đến các từ độc lập, HAL cần sử dụng một cấu trúc thích hợp để đưa ra câu trả lời. Ví dụ, HAL cần phải hiểu chuỗi từ này là vô nghĩa đối với Dave, mặc dù nó vẫn bao gồm chính xác số từ như trong câu gốc.

(1.1) Dave Xin lỗi , tôi không thể rằng e tôi thể làm được

Tri thức cần thiết để sắp xếp và tổ chức các từ để trở thành câu được gọi là **cú pháp**. Giờ hãy xem xét một câu hỏi trong hệ thống hỏi đáp:

(1.2) Người Trung Quốc xuất khẩu bao nhiêu lụa sang Tây Âu cho đến hết thế kỷ 18?

Để trả lời câu hỏi này chúng ta cần biết về nghĩa từ, nghĩa của tất cả các từ (như xuất khẩu hay lụa) cũng như tổng hợp nghĩa (cụm từ Tây Âu so với Đông hoặc Nam Âu), từ *hết* có nghĩa là gì khi kết hợp với *thế kỷ 18*. Chúng ta cũng cần biết về liên hệ giữa từ với cấu trúc ngữ pháp. Ví dụ chúng ta cần biết cụm từ *cho đến hết thế kỷ 18* là một cụm từ chỉ thời gian, chứ không phải một đơn vị, như trong ví dụ dưới đây:

(1.3) Người Trung Quốc xuất khẩu bao nhiêu lụa sang Tây Âu thông qua qua các thương nhân miền nam?

HAL cũng cần phân biệt câu nói của Dave là một yêu cầu, thay vì là một câu

mô tả hay câu hỏi về cái kiểu, như các cách nói khác nhau dưới đây

YÊU CẦU: *HAL, open the pod bay door.*

MÔ TẢ: *HAL, the pod bay door is open.*

CÂU HỎI: *HAL, is the pod bay door open?*

Mặc dù HAL không nghe lời. Nhưng HAL hiểu đủ để trả lời một cách lịch sự với Dave. Nó có thể trả lời trống không, như *No* hay *No, I won't open the door.* Thay vào đó, đầu tiên nó nói xin lỗi *I'm sorry*, sau đó nó thêm vào cụm từ *I'm afraid* thay vì từ chối một cách trực tiếp *I can't*. Tri thức cần có cho loại câu này là tri thức về **ngữ dụng** hay **hội thoại**.

(1.4) Nước Mỹ có bao nhiêu bang trong năm đó?

*Năm đó* là năm nào? Để có thể hiểu được những từ như *năm đó*, hệ thống hỏi đáp cần xem xét các câu hỏi trước đó; trong trường hợp này, câu trước đó hỏi về năm Lincoln ra đời. Nhiệm vụ này gọi là **phân giải đồng tham chiếu**, sử dụng các từ như *đó* hay các đại từ như nó hoặc cô ấy được đề cập trong những câu trước đó.

Tổng kết lại, để có sử dụng thành thạo ngôn ngữ cần có các hiểu biết về các loại tri thức:

- Ngữ âm và âm vị học - tri thức về âm thanh trong tiếng nói
- Ngôn ngữ hình thái học - tri thức về ý nghĩa của các thành phần trong từ
- Ngữ pháp - tri thức về cấu trúc của các từ
- Ngữ nghĩa - tri thức về nghĩa
- Ngữ dụng - tri thức về sự liên hệ giữa mục tiêu và ý định của người nói
- Diễn ngôn - tri thức về các đơn vị ngôn ngữ nhiều hơn một câu

## 1.2. NHẬP NHẰNG

Một điều đáng chú ý về việc phân chia các tri thức về ngôn ngữ là phần lớn các nhiệm vụ trong xử lý ngôn ngữ và tiếng nói có thể xem như việc giải quyết **nhập nhằng** ở một cấp độ nhất định. Một đầu vào được gọi là **nhập nhằng** nếu có nhiều cấu trúc ngôn ngữ có thể chỉ đến nó. Xem xét câu nói *I made her duck*. Có năm cách hiểu câu trên (bạn có thể nghĩ thêm), mỗi một cách hiểu ví dụ cho sự nhập nhằng ở từng cấp độ:

(1.5) Tôi nấu món vịt cho cô ấy ăn.

(1.6) Tôi nấu con vịt của cô ấy

(1.7) Tôi tạo ra con vịt đồ chơi cho cô ấy.

(1.8) Tôi khiến cô ấy trở nên ngốc nghếch (như con vịt).

(1.9) Tôi biến cô ấy thành con vịt.

Các nghĩa khác nhau xảy ra bởi những sự nhập nhằng. Đầu tiên, từ *duck* và *her* là nhập nhằng về hình thái từ và cú pháp. Từ *duck* có thể là động từ hoặc một danh từ, trong khi *her* có thể là đại từ tân ngữ hoặc đại từ sở hữu. Thứ hai, từ *make* cũng có sự nhập nhằng ngữ nghĩa, có thể có nghĩa là *make* (tạo) hoặc *cook* (nấu ăn). Cuối cùng, động từ *make* cũng dẫn đến sự nhập nhằng về cú pháp. *Make* có thể là ngoại động từ một tân ngữ (1.6), hoặc có thể là ngoại động từ hai tân ngữ (1.9), trong đó tân ngữ thứ nhất (*her*) bị biến thành tân ngữ thứ hai (*duck*). Cuối cùng, *make* có thể nhận một tân ngữ trực tiếp và một động từ (1.8), tân ngữ thứ nhất (*her*) bị biến thành tân ngữ thứ hai (*duck*). Hơn nữa, trong một câu nói, thậm chí còn có một mức độ nhập nhằng sâu hơn; từ thứ nhất có thể là từ *eye* và từ thứ hai có thể là từ *maid*.

Chúng tôi thường xuyên giới thiệu các mô hình và giải thuật trong quyển sách này là cách để **khử** hay **phân giải** nhập nhằng. Ví dụ để quyết định *duck* là động từ hoặc danh từ có thể được giải quyết qua bài toán **gán nhãn từ loại**. Quyết định từ *make* có nghĩa là "create" (tạo) hay "cook" (nấu ăn) có thể giải quyết qua bài toán **khử nhập nhằng nghĩa từ**. Phân giải nhập nhằng từ loại và nghĩa từ là hai bài toán quan trọng trong **khử nhập nhằng từ vựng**. Rất nhiều các bài toán khác có thể cũng được coi là bài toán khử nhập nhằng từ vựng. Như một hệ thống tổng hợp tiếng nói đọc từ *lead* cần quyết định nó phát âm như trong *lead pipe* hay *lead me on*. Mặt khác, quyết định *her* và *duck* là một thành phần của một thực thể (như 1.5 hay 1.8) hay ở các thực thể riêng biệt (như 1.6) là một ví dụ của nhiệm vụ **khử nhập nhằng cú pháp** và có thể được giải quyết bằng phương pháp **probabilistic parsing** (phân tích câu dựa xác suất). Chúng ta cũng sẽ xem xét các sự nhập nhằng không xuất hiện trong ví dụ cụ thể này, như xác định một câu là câu trần thuật hay câu hỏi (có thể được giải quyết bằng việc **diễn giải hành động lời nói**).

## 1.3. CÁC MÔ HÌNH VÀ GIẢI THUẬT

Một trong những điểm nhấn trong 50 năm nghiên cứu về xử lý ngôn ngữ tự nhiên là rất nhiều kiến thức được trình bày ở phần trước có thể xử lý bằng một vài mô hình và giả thuyết. May mắn thay, các mô hình và giả thuyết này được xuất hiện như những công cụ cơ bản cho khoa học máy tính, toán học, ngôn ngữ học và có thể được áp dụng giống nhau trong các ngành này. Bên cạnh những mô hình quan trọng nhất như **state machines**, **rule systems**, **logic**, **probabilistic models** và **vector-space models**. Bản thân các mô hình này, lại dựa vào một vài giải thuật cơ bản, như các giải thuật tìm kiếm trong gian trạng thái như **dynamic programming**, hay các giải thuật về học máy như các phân lớp và **Expected-Maximization (EM)** và các giải thuật khác.

Ở dạng đơn giản nhất, state machines là các mô hình hình thức bao gồm trạng thái, chuyển giữa các trạng thái, và một biểu diễn đầu vào. Một vài biến thể của mô hình cơ bản này như **deterministic** và **non-deterministic finite-state automata** và **finite-state transducers**.

Liên quan chặt chẽ đến các mô hình này là các hệ thống formal rule. Chúng ta sẽ xem xét những mô hình quan trọng nhất như **regular grammars** và **regular relations**, **context-free grammars**, và **feature-augmented grammars**. Các hệ thống máy trạng thái và formal rule là các công cụ chính để xử lý các tri thức về âm vị học, ngôn ngữ hình thái học và cú pháp.

Lớp mô hình thứ ba đóng vai trò quan trọng trong việc mô hình hóa tri thức là logic. Chúng ta sẽ tìm hiểu về **first order logic**, hay còn gọi là **predicate calculus**, cũng như các mô hình hình thức liên quan như lambda-calculus, feature-structures hay semantic primitives. Các biểu diễn logic này thường được sử dụng để mô hình hóa ngữ nghĩa và ngữ dụng, mặc dù các kỹ thuật gần đây thường tập trung vào các phương pháp hiệu quả hơn từ các ngữ liệu phi logic.

Mô hình xác suất đóng vai trò quan trọng trong việc mô hình hóa tất cả các cấp độ về ngôn ngữ. Từng mô hình khác (state machines, formal rule systems hay logic) có thể tăng cường với xác suất. Ví dụ, state machine có thể tăng cường với xác suất để trở thành **weighted automaton** hay **Markov model**. Chúng ta cũng sẽ dành một thời lượng nhất định cho **hidden Markov models** hay **HMMs**, được sử dụng ở mọi nơi, trong gán nhãn từ loại, nhận dạng tiếng nói, hiểu hội thoại, tổng hợp tiếng nói và dịch máy. Một điểm quan trọng của mô hình xác suất là khả năng có thể giải quyết được nhiều loại nhập nhằng đã đề cập trước đó; hầu hết các bài toán về xử lý ngôn ngữ và tiếng nói có thể coi như "cho N lựa chọn từ một đầu vào nhập nhằng, chọn lựa chọn có khả năng xảy ra cao nhất".

Cuối cùng, vector-space models, dựa vào đại số tuyến tính, được sử dụng trong bài toán trích rút thông tin và nghĩa từ.

Xử lý ngôn ngữ sử dụng những mô hình này thường liên quan đến bài toán tìm kiếm trong một không gian trạng thái biểu diễn các giả định từ một đầu vào. Trong nhận dạng tiếng nói, chúng ta cần tìm kiếm trong một không gian các chuỗi âm cho một từ đầu ra. Trong bài toán phân tích cú pháp, chúng ta cần tìm kiếm trong không gian các cây biểu diễn cú pháp cho một câu đầu vào. Trong dịch máy, chúng ta cần tìm kiếm trong không gian các giả định đầu ra cho một ngôn ngữ cho một câu đầu vào ở ngôn ngữ khác. Trong các bài toán không dựa xác suất, thường liên quan đến các máy trạng thái, chúng ta sử dụng các thuật toán phổ biến về đồ thị như **depth-first search**. Đối với các bài toán dựa xác suất, chúng ta thường sử dụng nhiều giải thuật heuristic như **best-first** hay **A\* search**, và dựa vào các giải thuật quy hoạch động để giảm khối lượng tính toán.

Các công cụ học máy như **các bộ phân lớp** và **các mô hình chuỗi** đóng một vị trí quan trọng trong nhiều bài toán xử lý ngôn ngữ. Dựa vào tính chất của từng đối

tượng, một bộ phân lớp thường gán mỗi đối tượng cho một lớp trong khi mô hình chuỗi cố gắng chuỗi các chuỗi đối tượng thành chuỗi các lớp.

Ví dụ, trong bài toán xác định một từ có đúng chính tả hay không, các bộ phân lớp như **cây quyết định**, **support vector machines**, **Gaussian Mixture Models**, và **logistic regression** có thể sử dụng để làm các bộ phân lớp nhị phân (đúng hay không đúng) cho một từ ở một thời điểm. Các mô hình chuỗi như **hidden Markov models**, **maximum entropy Markov models**, và **conditional random fields** có thể gán nhãn đúng/không đúng cho tất cả các từ trong một câu cùng một lúc.

Cuối cùng, các nhà nghiên cứu xử lý ngôn ngữ sử dụng rất nhiều công cụ từ các nhà nghiên cứu trong lĩnh vực học máy - như sử dụng tập huấn luyện và tập test, các kỹ thuật thống kê như **cross-validation**, và việc đánh giá hệ thống đã huấn luyện một cách kỹ lưỡng.

## 1.4. NGÔN NGỮ, SUY NGHĨ VÀ SỰ HIỂU BIẾT

Với nhiều người, khả năng máy tính có thể xử lý được ngôn ngữ một cách thành thạo như con người là dấu hiệu của trí thông minh thực sự. Nguyên nhân của niềm tin này đến từ việc sử dụng ngôn ngữ hiệu quả gần lớn với các khả năng nhận thức của chúng ta. Một trong những người đầu tiên coi tác động của sự liên hệ đó là Alan Turing (1950). Trong bài báo nổi tiếng này, Turing giới thiệu một thử nghiệm mà sau này được biết với tên Turing test. Turing bắt đầu với một luận điểm, rằng câu hỏi "Liệu máy tính có thể nghĩ" là một câu hỏi không thể trả lời được vì sự thiếu chính xác vốn có trong cụm từ *máy tính* và *nghĩ*. Thay vào đó, ông đề xuất một phép thử thực tế, một trò chơi, trong đó việc sử dụng ngôn ngữ của máy tính cho thấy khả năng suy nghĩ của nó. Nếu máy tính có thể thắng trò chơi này, nó sẽ được đánh giá là thông minh.

Trong trò chơi của Turing, có ba bên tham gia: hai người và một máy tính. Một người đóng vai trò là người chơi và đóng vai trò là người hỏi. Để chiến thắng, người hỏi cần phải xác định xem hai thành viên còn lại ai là máy tính bằng cách hỏi một chuỗi câu hỏi thông qua việc gõ bàn phím. Nhiệm vụ của máy tính là lừa người hỏi tin rằng nó là con người bằng cách trả lời các câu hỏi. Nhiệm vụ của người chơi thứ hai là thuyết phục người hỏi, người chơi kia là máy tính, mình mới là con người.

Đoạn hội thoại dưới đây được lấy từ bài báo của Turing minh họa cho trò chơi. Rõ ràng, việc giả là người không cần yêu cầu kiến thức của tất cả các lĩnh vực

Q: Please write me a sonnet on the topic of Forth Bridge

A: Count me out on this one. I never could write poetry

Q: Add 34957 to 70764

A: (Pause about 30 seconds and then give answer as) 105621

Hãy viết một bài thơ về chủ đề cầu Forth

Bỏ qua cho tôi câu này. Tôi chưa bao giờ làm thơ

Cộng 34957 với 70764

(Dừng 30 giây sau đó trả lời) 105621

Turing dự đoán đến hết thế kỷ hai mươi, một chiếc máy tính với 10 gigabytes bộ nhớ có khoảng 30

Rất ngạc nhiên, nhiều vấn đề được đề xuất bởi Turing liên hệ nhiều đến chương trình **ELIZA** xuất hiện vào năm 1966 (Weizenbaum, 1966). ELIZA là hệ thống xử lý ngôn ngữ tự nhiên có khả năng thực hiện một số đoạn hội thoại với người dùng. Cùng xem xét đoạn hội thoại sau đây với một phiên bản của ELIZA

- User<sub>1</sub> : You are like my father in some ways.  
*Bạn rất giống bố tôi*
- ELIZA<sub>1</sub> : WHAT RESEMBLANCE DO YOU SEE  
*BẠN THẤY GIỐNG Ở ĐIỂM NÀO*
- User<sub>2</sub> : You are not very aggressive but I think you don't want me to notice that.  
*Bạn không quá hung hăng nhưng tôi không nghĩ bạn muốn tôi để ý điều đó*
- ELIZA<sub>2</sub> : WHAT MAKES YOU THINK I AM NOT AGGRESSIVE  
*ĐIỀU GÌ KHIẾN BẠN NGHĨ TÔI KHÔNG HUNG HĂNG*
- User<sub>3</sub> : You don't argue with me  
*Bạn không cãi nhau với tôi*
- ELIZA<sub>3</sub> : WHY DO YOU THINK I DON'T ARGUE WITH YOU  
*TẠI SAO BẠN NGHĨ TÔI KHÔNG CẢI NHAU VỚI BẠN*
- User<sub>4</sub> : You are afraid of me.  
*Bạn sợ tôi*
- ELIZA<sub>4</sub> : DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU  
*BẠN NGHĨ RẰNG TÔI SỢ BẠN À?*

Như chúng tôi sẽ thảo luận chi tiết trong chương 2, ELIZA là một chương trình rất đơn giản sử dụng pattern-matching để xử lý đầu ra và chuyển thành đầu ra tương ứng. Sự thành công của kỹ thuật đơn giản trong lĩnh vực này dựa vào việc ELIZA không thực sự cần biết điều gì để giả người. Như trong ghi chép của Weizenbaum, đây là một trong số ít các loại hội thoại trong đó người nghe có thể phản hồi mà không cần biết điều gì về thế giới.

ELIZA liên hệ nhiều đến ý tưởng của Turing khi có nhiều người tương tác với ELIZA đã tin nó thực sự hiểu họ và vấn đề của họ. Hơn nữa, Weizenbaum (1976) cho thấy nhiều người tiếp tục tin vào khả năng của ELIZA thậm chí cách thức hoạt động của nó được giải thích cho họ. Trong những năm sau, Thí nghiệm của Weizenbaum được lặp lại trong các môi trường khác nhau. Năm 1991, một sự kiện được biết đến như cuộc thi Loebner Prize được tổ chức, cố gắng đưa nhiều chương trình máy tính vào thí nghiệm Turing. Mặc dù những cuộc thi như vậy không thu hút nhiều giới khoa học, những kết quả qua từng năm cho thấy ngay cả các chương trình đơn giản nhất cũng có lúc có thể lừa các giám khảo (Shieber, 1994a). Không ngạc nhiên, các kết quả đó không thể dập tắt được cuộc tranh luận về tính thích hợp của phép thử Turing cho việc kiểm tra trí thông minh giữa các nhà triết học và các nhà nghiên cứu trí tuệ nhân tạo (Searle, 1980).

May thay, với mục đích của cuốn sách, các kết quả đó không liên quan đến việc khả năng của máy tính trong việc trở nên thông minh, hay hiểu ngôn ngữ tự nhiên.



Điều quan trọng hơn nữa là các kết quả nghiên cứu trong lĩnh vực xã hội học đã khẳng định một dự báo của Turing trong cùng bài báo đó.

Tôi tin rằng cuối thế kỷ này, việc sử dụng từ và các ý kiến được giáo dục sẽ thay đổi nhiều đến mức chúng ta có thể trò chuyện với các máy tính suy nghĩ mà không mong đợi được phản đối.

Bây giờ rõ ràng là bất kể mọi người tin hay biết gì về cách thức hoạt động của máy tính, mọi người thường nói về chúng và tương tác với chúng như là những thực thể trong xã hội. Mọi người thường nói chuyện với máy tính nếu họ nghĩ máy tính là người; họ sẽ lịch sự, đối xử như những thành viên, hi vọng rằng máy tính có thể hiểu nhu cầu của họ, và có thể tương tác với họ một cách tự nhiên. Ví dụ, Reeves và Nass (1966) phát hiện rằng khi một máy tính hỏi một người đánh giá máy tính như thế nào, người đó đưa ra nhiều phản hồi tính cực khi một máy tính khác hỏi các câu tương tự. Con người thường sợ trở nên bất lịch sự. Trong một thử nghiệm khác, Reeves và Nass phát hiện rằng con người thường cho điểm cao hơn nếu máy tính biết tăng bốc. Từ các thiên hướng này, các hệ thống dựa vào ngôn ngữ và tiếng nói cần đưa cho người dùng những giao diện tự nhiên nhất. Điều này dẫn đến sự tập trung dài hơn vào thiết kế cho các **máy giao tiếp** và hệ thống thông minh có thể đối thoại.

## 1.5. CÁC KẾT QUẢ TỐT NHẤT

*Chúng ta chỉ có thể nhìn thấy một quãng đường ngắn trước mắt,  
nhưng chúng ta có thể thấy rất nhiều việc cần làm.*

Alan Turing.

Hiện tại là thời điểm sôi động lĩnh vực xử lý ngôn ngữ và tiếng nói. Bắt đầu từ việc có nhiều tài nguyên tính toán hơn, việc phát triển của Web đưa đến một lượng lớn thông tin có thể thu thập được hỗ trợ việc phát triển mọi khía cạnh của các ứng dụng xử lý ngôn ngữ và tiếng nói. Các ví dụ dưới đây đưa ra các hệ thống đã được triển khai trong xu hướng này:

- Các công ty du lịch như Amtrak, United Airlines hay các nhà cung cấp du lịch khác đưa ra các chương trình hội thoại có thể hướng dẫn trong quá trình đặt vé và đưa ra các thông tin về các chuyến đi.
- Các công ty xe cao cấp như Mercedes-Benz đưa ra các hệ thống tổng hợp và nhận diện tiếng nói có thể cho người lái điều khiển các hệ thống môi trường, giải trí và định vị trong xe bằng giọng nói. Hệ thống thoại tương tự cũng được triển khai cho các phi hành gia ở International Space Station.

- Blinkx và các công ty trong lĩnh vực tìm kiếm video khác cung cấp các dịch vụ tìm kiếm cho hàng triệu giờ video trên Web sử dụng công nghệ nhận diện tiếng nói để xác định các từ trong video.
- Google cung cấp các dịch vụ dịch và trích rút thông tin đa ngôn ngữ khi người dùng có thể gõ câu hỏi tìm kiếm bằng ngôn ngữ mẹ đẻ trong các ngôn ngữ khác. Google sẽ tự động dịch câu hỏi, tìm kiếm những trang web thích hợp và sau đó tự động dịch chúng lại về tiếng bản ngữ của người dùng.
- Các tổ chức xuất bản lớn như Pearson, hay các dịch vụ kiểm tra như ETS, sử dụng các hệ thống tự động để phân tích, chấm điểm và đánh giá hàng ngàn bài luận của sinh viên.
- Các hệ thống hướng dẫn tự động, dựa vào các nhân vật hoạt hình, dạy trẻ em đọc (Wise et al., 2017).
- Các công ty phân tích văn bản như Nielsen, Buzzmetrics, Umbria và Collective Intellect cung cấp các dịch vụ tiếp thị thông minh dựa và việc tự động giá ý định, sở thích, tính cách và thái độ của người dùng từ các trang web và forum.

## 1.6. MỘT VÀI LỊCH SỬ NGẮN GỌN

Trong lịch sử,