Một trong những chén thánh của xử lý ngôn ngữ tự nhiên là hệ thống hỏi trả lời. Facebook Babi tasks là một bộ dữ liệu gồm 20 tasks được phát hành bởi Facebook AI Research team điều này giúp các hệ thống mong muốn làm điều này.

Một ví dụ từ task thứ 2. Hai hỗ trợ sự thật (QA2), là bên dưới:

1 John moved to the bedroom.  
2 **Mary grabbed the football there.**  
3 Sandra journeyed to the bedroom.  
4 Sandra went back to the hallway.  
5 Mary moved to the garden.  
6 **Mary journeyed to the office.**  
7 Where is the **football**? office 2 6

Mặc dù điều này có vẻ tầm thường đối với bạn, nó có thể đại diện cho một thách thức lớn cho mô hình học máy cao cấp. Các task bAbi bao gồm nhiều hơn tầm hiểu biết tuy nhiên chúng được cho là đại diện cho một điều kiện tiên quyết đối với QA AI-Complete. Mỗi điều kiện nhắm đến một khía cạnh độc đáo của văn bản và lý do, kiểm tra sự khác biệt cho phép cho mô hình học. để trả lời câu hỏi chính xác, mô hình phải có khả năng trình bày thực hiện khởi tạo, khấu trừ, kết nối thực tế, và nhiều hơn nữa.

Trong khi thực hiện tốt nhiệm vụ này đòi hỏi công cụ tiên tiến, chúng ta có thể thực hiện một giải pháp cơ sở chỉ trong một vài dòng bằng cách sử dụng thư viện học máy Keras. Kết quả có thể so sánh được ( và đôi khi vươt trội) đối với các tiêu chuẩn được cung cấp bởi Weston et al’s [Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks](http://arxiv.org/abs/1502.05698) cho chỉ 100 mẫu và không có bất kì hyperparameter điều chỉnh.

# Try it yourself!

Code QA là một bài báo liên quan tới một phần riêng của Keras: babi\_rnn.py trong thư mục ví dụ. khi bạn chay ví dụ, Keras sẽ tự đồng tải về dataset và bắt đầu huấn luyện

Tốt nhất cho tất cả, trái ngược với hầu hết các tasks deep learning, đào tạo các mô hình này sẽ chỉ mất vài phút mỗi!

## Why a synthetic dataset? Tại sao một tập dữ liệu tổng hợp

Vì bAbi là một tập dữ liệu tổng hợp, bạn có thể hỏi tại sao chúng ta hứng thú trong việc làm tốt nó, hoặc tại sao chúng tôi thậm chí tạo ra tất cả nó.

Dữ liệu thật trên thế giới khá tạp, hiếm khi nó được cung cấp một cách rõ ràng và đơn giản câu trả lời cho huấn luyện của bạn. thêm vào đó, mặc dù dữ liệu từ thế giới thật khá tạp, phức tập và lỗi.

Thay vì sự dụng dữ liệu thật, chúng ta thay vì thử mô hình học máy sử dụng mô phỏng hồi tưởng lại các trò chơi phiêu lưu văn bản cổ điển. Tasks được tạo ra bằng cách mô phỏng hồi tưởng của các văn bản truyền thống. bằng cách sử dụng thế giới nhân tạo chúng ta biết chính xác trạng thái của thế giới và quy luật nó vận hành. Biết ơn điều đó, tạo ra huấn luyện và kiểm tra dữ liệu là không đán kể.

Trái ngược với tài liệu thế giới thực, dữ liệu cũng được giám sát tốt. từ vựng ( một bộ từ) là hạn chế, câu thì luôn luôn được cấu trúc tốt (dữ liệu tạp chỉ được sử dụng khi ta muốn kiểm tra mô hình), và việc thực hiện các nhiệm vụ cụ thể có thể được kiểm tra mà không có những công việc khác can thiệp. như chúng ta biết chính xác trạng thới của thế giới và sao có đucợ điều đó, chúng ta cũng cung cấp thêm những thông tin hữu ích, như là điểm chẳng hạn như chỉ ra chính xác cách thức có thể đạt được câu trả lời (các sự kiện hỗ trợ được in đậm ở trên).

Với bộ dữ liệu, tất cả các kiến thức thông thưởng và nguyên nhân được yêu cầu để kiểm tra nên được chứa trong tập huấn luyện. bằng cách đó, nếu một mô hình học máy thất bại trong việc giải task, chúng ta biết thách thức là bản thân mô hình, không phải bởi dữ liệu.

## How do we approach the problem?

Một trong những cách dễ nhất để tiệp cận một vấn đề là code phương pháp baseline . phương pháp Baseline là cung cấp tốt “bang for the buck” – ít việc phải lắm cho kết quả tốt nhất có thể. Trong tình huống này, một neural network hồi quy (RNN) là baseline chúng ta có thể sử dụng.

RNN cũng như LSTM và Gated Recurrent Unit (GRU) là neural network nó có thể xử lý một câu đầu vào, cập nhật network bên trong như đọc thêm dữ liệu. Điều này cho phép nó để tìm hiểu phụ thuộc lâu dài như phù hợp với bracket matching. Như chúng ta đã mã hóa vào việc trình bày vector, chúng ta có thể xem xét một câu như là một chuỗi của từ, cho nó vào RNN cùng một lúc.

Thay vì hiện thực mô hình của chúng ta, thay vào đó chúng ta có thể xử dụng mô hình dự sẵn.

## Keras - a deep learning library for Python

Keras là một nhỏ gọn, cao mô-đun thư viện mạng nơron trong tinh thần của Torch, viết bằng Python, sử dụng Theano cho thao tác tensor tối ưu trên GPU và CPU.

Là một dự án nguồn mở, nó có tài liệu mạnh mẽ, một cộng đồng tích cực, và một nhà lãnh đạo giỏi. Chỉ ba giờ sau khi gửi pull request cho code ví dụ này, François Chollet (fchollet) được merged code. Sự phản hồi lại nhanh chóng của dự án và các ví dụ mạnh mẽ làm cho nó trở thành một thư viện tốt để bắt đầu với deep learning. Keras cũng thúc đẩy thư viện Theano, một thư viện Python xác định, tối ưu hóa, và đánh giá các biểu thức toán học có liên quan đến các mảng đa chiều một cách hiệu quả.

## The challenge at hand

Ý tưởng của chúng tôi là như sau: mỗi tác vụ đều có thành phần câu chuyện và một thành phần truy vấn. Chúng tôi sẽ chạy một RNN trên cả hai thành phần này chuyển đổi chuỗi các từ dài thành một biểu diễn vector cố định. Hình đại diện vector cố định này nên hy vọng đóng gói tất cả các đầu vào có liên quan. Cuối cùng, chúng ta nuôi hai hình đại diện vectơ cố định này thành một mạng nơ ron truyền thống dày đặc, nơi nó có thể nhìn vào các truy vấn được mã hóa, sau đó tại câu chuyện mã hóa, và hy vọng trả lời câu hỏi một cách chính xác.

### Word vectors

Một thành phần bổ sung là biểu diễn word vectors. Đây là nơi hy vọng để chuyển một từ thành một vector cố định biểu diễn đóng một kiến thức lớn về nó. Word vectors hy vọng để thu thập nghĩa đằng sau từ, kích hoạt các từ liên quan để xem xét tính tương tự và du vậy hành động trong tính tương tự.

Điều này có thể quan trọng nếu cho ví dụ, chúng ta có 2 câu:

**John put down the apple.**  
**John dropped the apple.**

2 câu chỉ thú vị khi trả lời cho câu hỏi "**Does John have the apple?**",nơi mà các sắc thái giữa việc đặt một cái gì đó xuống và thả nó là không quan trọng.

Trong khi chúng ta có thể học được các biểu diễn vector từ tốt cho một nhóm từ nhỏ trong các nhiệm vụ này, nó sẽ không có kiến ​​thức rộng hơn. Đối với các task thực sự, nơi mà kiến thức chính xác có thể hữu dụng (như là frog ~=toad), chúng ta có thể sự dụng những từ đã được huấn luyện trên 1 tỉ từ, như là Stanford’s GloVe.

### The code

May mắn thay, code cho các tính năng tuyệt vời trở nên đơn giản nhờ Keras. Bạn có thể nhìn thấy code đầy đủ ở file babi\_rnn.py như code RNN nhanh chóng và tối thiểu dưới đây.

## Final results

Trong phần này, tôi so sánh kết quả cuối cùng cho hệ thống trả lời câu hỏi dựa trên Keras với đường cơ sở LSTM được cung cấp bởi giấy Facebook.

Kết quả có thể so sánh được (và đôi khi vượt trội) đối với các tiêu chuẩn cho đường cơ sở LSTM được cung cấp ở Weston et al.. [Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks](http://arxiv.org/abs/1502.05698) chỉ có 1000 mẫu và không có bất kỳ điều chỉnh siêu đối xứng nào. Mô hình tương tự cũng được sử dụng trên tất cả các nhiệm vụ.

Thật không may, cơ sở là chỉ có vậy. Sử dụng các mạng thần kinh tái phát truyền thống, chẳng hạn như LSTM hoặc GRU, sẽ không cho bạn hiệu năng tốt hơn đáng kể ngay cả khi bạn mở rộng mạng lưới lên rất nhiều. Để có kết quả tốt hơn, các cấu hình mạng thần kinh mới đã được đề xuất và sử dụng, chẳng hạn như Facebook's Memory Network (tiếp tục cải tiến trong bài báo trình bày bộ dữ liệu bAbi), Máy Turing Neural của Google và Hệ thống Dynamic Memory Networks của MetaMind.

Tất cả các mô hình này có thể tận dụng sự hiểu biết nơi mà các sự kiện hỗ trợ, học tập nơi để tập trung sự chú ý vào đầu vào, và thực hiện nhiều "tra cứu" để theo dõi các thông tin có liên quan. Tôi hy vọng sẽ thực hiện một phiên bản đơn giản của một trong những mô hình này trong tương lai gần.

Tuy nhiên, bây giờ, tôi hài lòng với đường cơ sở đơn giản của tôi.

…

## Note: Dataset issues - duplication in Positional Reasoning (QA17) and Size Reasoning (QA18)

Các kết quả trên cho thấy một sự khác biệt hiệu suất lớn giữa đường cơ sở Facebook LSTM và hệ thống Keras QA về QA18 - nhảy từ 52 lên 91. Trong khi điều tra, tôi nhận thấy rằng có rất nhiều câu lệnh và câu hỏi trùng lặp trong tập dữ liệu huấn luyện và thử nghiệm của QA18. Đây cũng là một vấn đề trong QA17 và có thể là các vấn đề khác. Cho rằng chỉ có 1000 điểm dữ liệu thử nghiệm và xe lửa (bạn có thể xác nhận bằng cách chạy …), sự lặp lại có thể gây ra các vấn đề nghiêm trọng.

Tôi sẽ gửi email cho người quản lý bộ dữ liệu khi tôi thực hiện phân tích đầy đủ với hy vọng điều này sẽ được sửa cho phiên bản 1.3 của dữ liệu.

### Update - Full duplicate analysis

Sau khi hoàn thành phân tích trùng lặp, có những vấn đề trong bộ dữ liệu cần được khắc phục. may mắn thay dataset đã được phát hành trong một phiên bản, mặc dù báo cáo không công bố dataset họ sử dụng và lịch sử phiên bản ko có sẵn. Phân tích trùng lặp được thực hiện bằng cách chỉ tìm ra bộ đôi duy nhất (câu chuyện, truy vấn, câu trả lời) trong tập huấn luyện và tập kiểm tra, sau đó tìm ra có bất kỳ sự giao thoa nào giữa các bộ đôi duy nhất đó.

Vấn đề cực đoan nhất là một trong số các nhiệm vụ, QA4, có khoảng 13% số mẫu duy nhất có trong đào tạo và thử nghiệm.

…

Một vấn đề khác là trùng lặp trong các bộ đào tạo và thử nghiệm, đặc biệt là vấn đề trong QA15, QA17 và QA18. Điều này đặc biệt quan trọng vì các thuật toán này chỉ được huấn luyện trên 1000 mẫu.

…

Những vấn đề này thậm chí trở nên cực đoan hơn khi các nhiệm vụ bAbi chứa 10.000 mẫu được sử dụng.