# NER – Named Entity Recognition

1. Giới thiệu

* Nhận dạng thực thể, đây là một bài toán con của NLP (Natural Language Processing).
* Vai trò chính là nhận dạng các cụm danh từ trong văn bản và phân loại chúng vào trong các nhóm (entity) đã được định trước như tên người, tổ chức, địa điểm, thời gian, tiền tệ...
* Từ đó phục vụ cho các bài toán cụ thể như Chatbot, Search, Phân cụm...
* Ví dụ:

Ca sĩ Thủy Tiên đã kêu gọi được hơn 100 tỉ ủng hộ đồng bào miền Trung.

Trong ví dụ: Thủy Tiên – tên người

100 tỉ - tiền tệ

Miền Trung – địa điểm

1. Các phương pháp cơ bản

* Hầu hết các hệ thống NER tập trung vào việc áp dụng các phương pháp học máy và không quá đòi hỏi kiến thức chuyên sâu về mặt ngôn ngữ.
* Có thể chia làm 2 nhóm lớn.
* Nhóm đầu tiên áp dụng các kỹ thuật trích xuất đặc trưng bằng tay và kết hợp với thuật toán gán nhãn theo chuỗi như Conditional Random Field (CRF), Hidden Markov Model (HMM), hoặc Maximum Entropy Markov Model (MEMM). Các phương pháp này đã được nghiên cứu kỹ lưỡng trong thời gian dài và đặc biệt thích hợp với các ngôn ngữ có nguồn dữ liệu ít như Tiếng Việt. Tuy nhiên, điểm yếu của nhóm phương pháp này chính là độ chính xác phụ thuộc rất lớn vào việc lựa chọn chuẩn xác các đặc trưng bằng tay.
* Nhóm thứ hai là các mô hình học tự động Machine Learning / Deep Learning được để xuất để giải quyết bài toán gán nhãn chuỗi. Ưu điểm vượt trội của các kiến trúc Deep Neural Network là khả năng End-to-end Learning, tức là khả năng học được các quy luật gán nhãn chuỗi từ tập dữ liệu gán nhãn trước mà không cần có bất cứ sự can thiệp của con người. Điều này đã loại bỏ đi nhược điểm của nhóm phương pháp 1 phải dựa trên kiến thức ngôn ngữ để lựa chọn các đặc trưng. Tuy nhiên, nhóm phương pháp 2 cũng tồn tại nhược điểm liên quan đến kích cỡ tập dữ liệu huấn luyện. Thông thường, các kiến trúc Deep Neural Network yêu cầu tập dữ liệu khá lớn để có thể đạt được độ chính xác cao. Đối với ngôn ngữ có ít dữ liệu có nhãn như Tiếng Việt, ta cần sử dụng ma trận Word Embedding tốt đã được huấn luyện từ một tập dữ liệu không nhãn lớn (tin tức online, diễn đàn, …nguồn dữ liệu do các tổ chức uy tín xây dựng).
* Named entity recognition có nghĩa là tìm các vùng văn bản có chưa tên riêng, và sau đó phân loại các thực thể. Việc nhận dạng khó khăn một phần vì sự mơ hồ của các phân khúc, chúng ta cần phải quyết định cái gì là thực thể và cái gì không phải, và đâu là ranh giới để phân định (đây cũng là chỗ áp dụng các thuật toán CRF, HMM, ME).
* Chuyển về bài toán Sequence Labeling. Bài toán được định nghĩa như sau: cho trước tập các chuỗi quan sát ký hiệu Thông thường được biểu diễn dưới dạng vector. Ta mong muốn gán nhãn dựa vào dữ kiện từ các trước đó. Để gán nhãn, ta thường dùng BIO notation. Với mỗi entity kiểu T, ta có hai nhãn B-T và I-T. B-T là begin type T, I-T là inside type T. Ngoài ra, ta còn có nhãn O cho biết outside name entity.
* Ví dụ:

Đại học Khoa học Tự Nhiên là trường có học phí rất tốt.

B-Org I-Org I I I I O O O O O O O

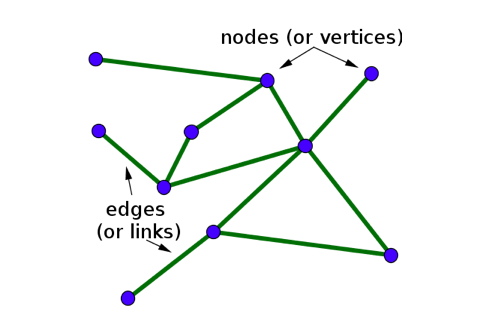
1. Thử nghiệm và Đánh giá

* Do mới dừng lại ở mức độ tìm hiểu và hạn chế về khả năng tìm được bộ dữ liệu thử nghiệm ().

# SIGMA JS

1. Khoa học mạng

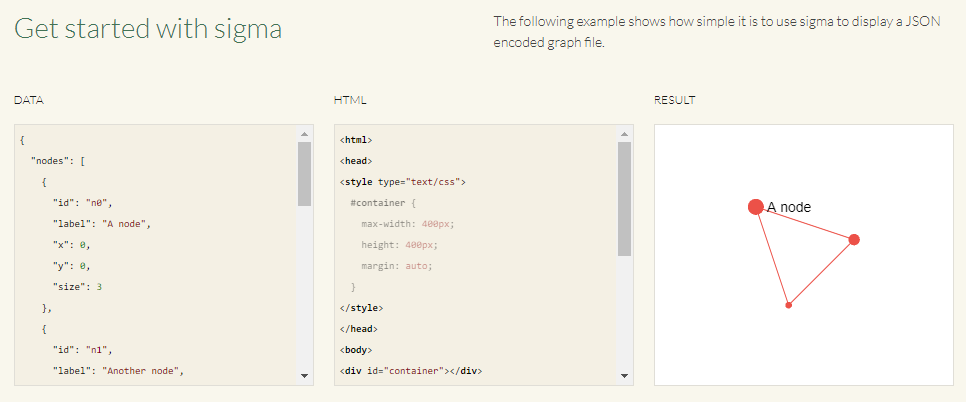
* Khoa học mạng là ngành nghiên cứu các mạng phức tạp như mạng viễn thông, mạng máy tính, mạng sinh học và thậm chí cả mạng ngữ nghĩa / xã hội.
* Chúng thường bao gồm các phần tử riêng biệt được đại diện bởi các nút và kết nối giữa các nút này thông qua các liên kết được gọi là các cạnh.
* Biểu diễn trực quan của mạng:



1. Giới thiệu Sigma JS

* Là một thư viện JavaScript dành riêng cho việc vẽ đồ thị
* Nó giúp dễ dàng trích xuất mạng trên các trang Web và cho phép các nhà phát triển tích hợp tính năng khám phá mạng trong các ứng dụng Web phong phú.
* Sigma JS được thiết kế như một công cụ mà bạn có thể tùy chỉnh và sử dụng để phát triển các ứng dụng Web có tính tương tác cao để hiển thị các hình ảnh đồ thị. Một số tính năng nổi bật:
  + Kết xuất tùy chỉnh: Trình kết xuất tích hợp [Canvas](http://en.wikipedia.org/wiki/Canvas_element) hoặc [WebGL](http://en.wikipedia.org/wiki/WebGL) sẽ giúp bạn hiển thị đồ họa 2D và 3D tương tác trong bất kỳ trình duyệt web tương thích nào mà không cần sử dụng trình cắm thêm. Và các trình kết xuất tích hợp cũng cung cấp rất nhiều cách để tùy chỉnh kết xuất.
  + Định hướng tương tác: Bạn có thể biết khi nào người dùng nhấp hoặc cuộn chuột qua một nút, biết khi nào người dùng kéo biểu đồ hoặc phóng to và luôn biết vị trí của biểu đồ tương đối với màn hình...
  + Mô hình đồ thị mạnh mẽ: Sigma chỉ là một công cụ kết xuất, nhưng bạn có thể muốn làm nhiều hơn thế, như chạy các thuật toán đồ thị của riêng bạn. Vì vậy, mô hình biểu đồ của sigma có thể tùy chỉnh và bạn có thể thêm các chỉ mục tùy chỉnh của riêng mình trên dữ liệu.
  + Có thể mở rộng: Có thể dễ dàng phát triển các plugin hoặc đoạn mã đơn giản để tăng cường các tính năng của sigma. Một số đã có sẵn trong kho lưu trữ chính để đọc một số định dạng tệp đồ thị phổ biến hoặc để chạy các thuật toán bố cục phức tạp.
  + Khả năng tương thích: Sigma chạy trên tất cả các trình duyệt hiện đại hỗ trợ [Canvas](http://caniuse.com/canvas) và hoạt động nhanh hơn trên trình duyệt có [hỗ trợ WebGL](http://caniuse.com/webgl).

1. Ví dụ đơn giản

* Từ sigmajs.org

Cơ bản gồm 3 phần:

* + Data (thể hiện nodes và edges)
  + Mã HTML
  + Đồ thị