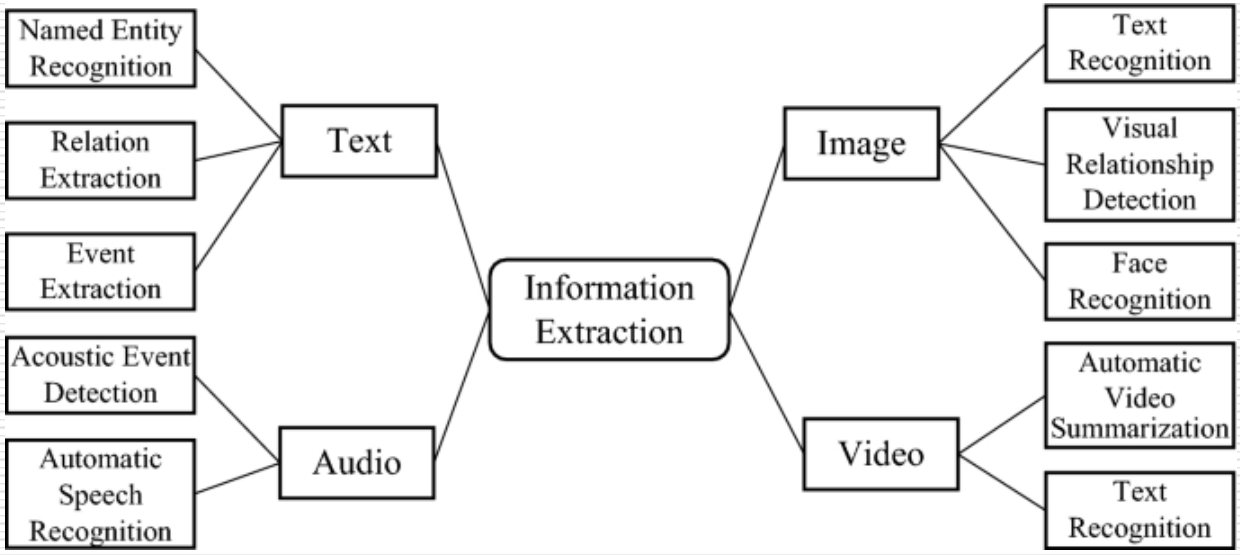

# RÚT TRÍCH THÔNG TIN

# Nội dung

- Kiến trúc hệ thống rút trích thông tin
- Nhận diện thực thể có tên (NER)
- Rút trích quan hệ không giám sát
- Giám sát từ xa
- Phân giải đồng tham chiếu

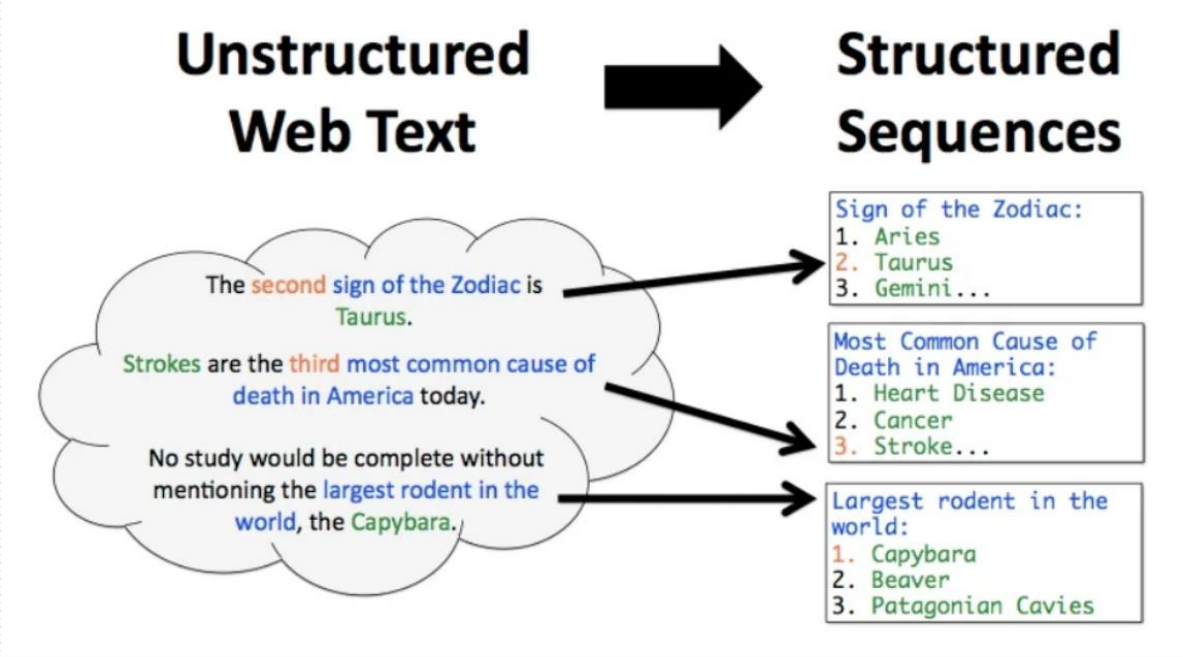
# 1. Kiến trúc của hệ thống RTTT

- Rút trích thông tin là quá trình tìm kiếm các thực thể và mối quan hệ giữa các thực thể này trong văn bản
- Rút trích thông tin phục vụ khai phá văn bản ở mức chính xác và cô đọng hơn các tác vụ như phân loại văn bản hay gán nhãn văn bản
- Các loại thực thể và quan hệ được định nghĩa trước



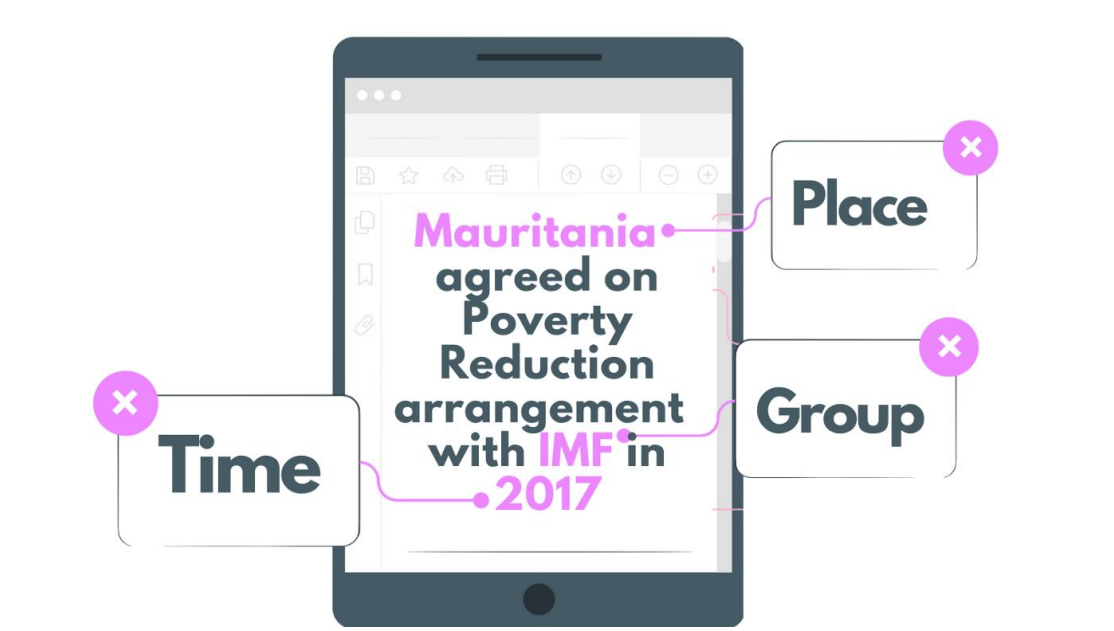
# 1. Kiến trúc của hệ thống RTTT

- Các giả thiết của rút trích thông tin
  - Thông tin được thể hiện một cách tường minh và không yêu cầu suy diễn
  - Một số lượng nhỏ khuôn mẫu có thể tóm tắt được nội dung của văn bản
  - Thông tin cần thiết xuất hiện cục bộ trong văn bản

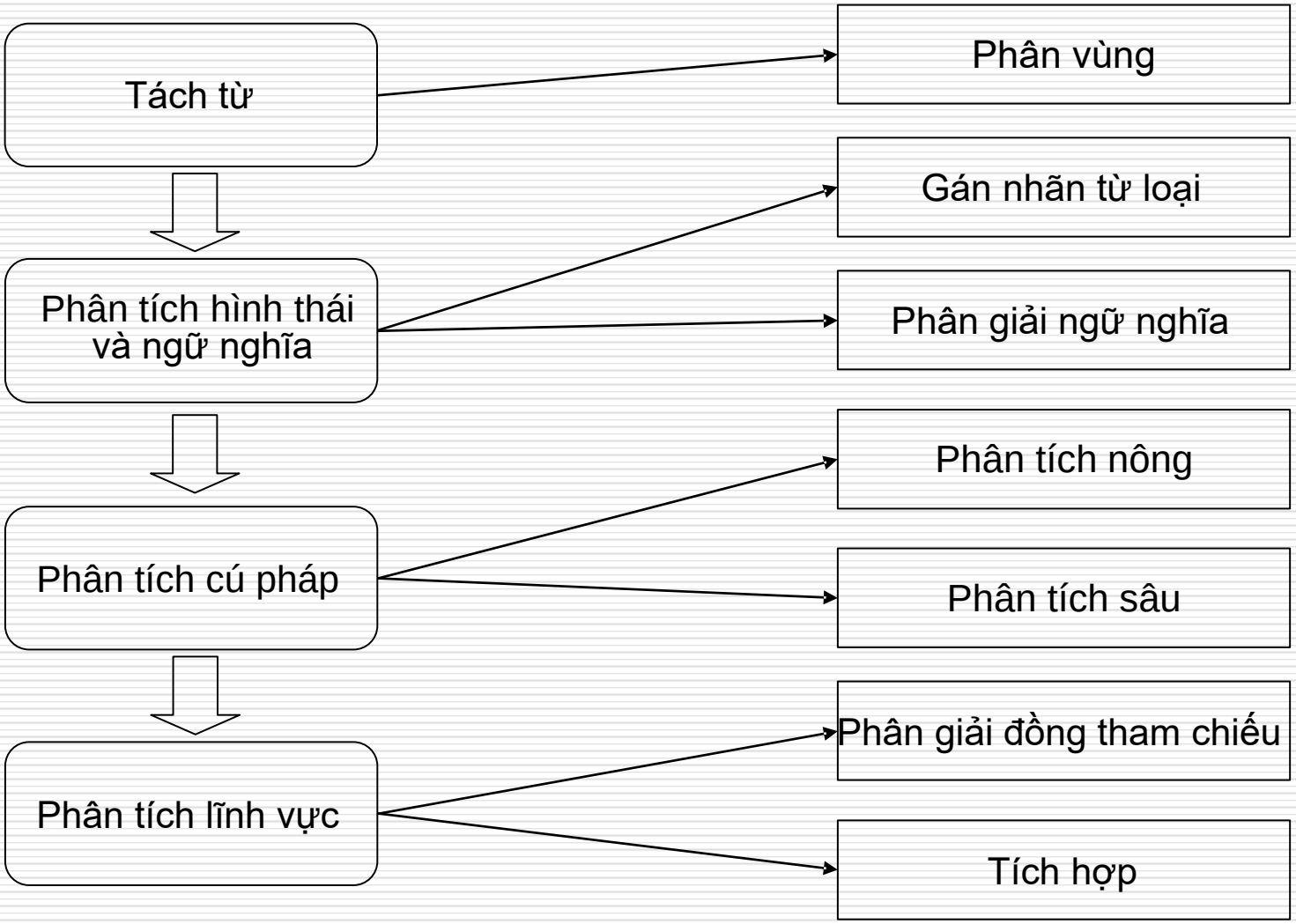


# 1. Kiến trúc của hệ thống RTTT

- Các loại thông tin được rút trích
  - **Thực thể:** Con người, tổ chức, địa điểm,…
  - **Thuộc tính (của thực thể):** Chức danh, tuổi, loại tổ chức…
  - **Thực tế:** quan hệ giữa nhân viên và công ty, quan hệ giữa virus và bệnh,…
  - **Sự kiện:** hai công ty sát nhập, động đất, khủng bố,...



# 1. Kiến trúc của hệ thống RTTT



# 2. Nhận diện thực thể có tên

- Phát hiện các thực tể có tên trong văn bản và phân loại vào các lớp được định nghĩa trước

**[Forbes]**ORG**: [Việt Nam]**LOC **có 4 tỷ phú**

- **Phân cụm:** Phát hiện các cụm từ trong câu

Trong đó , Việt Nam có 4 đại diện là Chủ tịch Vingroup Phạm Nhật Vượng, CEO VietJet Air Nguyễn Thị Phương Thảo, Chủ tịch Thaco Trần Bá Dương và Chủ tịch Techcombank Hồ Hùng Anh.

- **Rút trích quan hệ:** Rút trích các quan hệ giữa các thực thể (thuộc tính, thực tế, sự kiện)

Aikido là một môn võ thuật Nhật Bản hiện đại

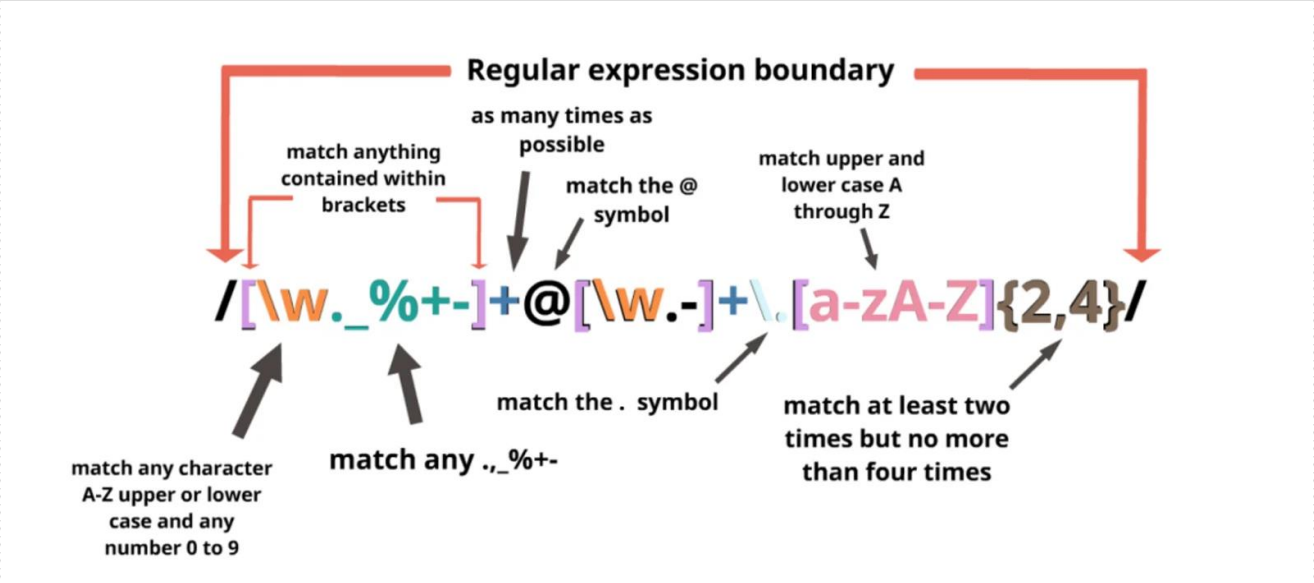
# 2. Nhận diện thực thể có tên

- **Phân giải đồng tham chiếu:** Phát hiện sự xuất hiện của cùng một thực thể dưới dạng các tham chiếu khác nhau

Aikido$_1$ là một môn võ thuật Nhật Bản hiện đại được phát triển bởi Ueshiba Morihei$_2$ như một sự tổng hợp các nghiên cứu võ học, triết học và tín ngưỡng tôn giáo của ông$_2$. Aikido$_1$ thường được dịch là "con đường hợp thông ( với ) năng lượng cuộc sống "hoặc" con đường của tinh thần hài hòa". Mục tiêu của Ueshiba$_2$ là tạo ra một nghệ thuật$_1$ mà các môn sinh$_3$ có thể sử dụng để tự bảo vệ mình$_3$ trong khi vẫn bảo vệ người tấn công$_4$ khỏi bị thương. Các kĩ thuật của Aikido$_1$ bao gồm: irimi (nhập thân), chuyển động xoay hướng (tenkan - chuyển hướng đà tấn công của đối phương$_4$), các loại động tác ném và khóa khớp khác nhau.

# 2. Nhận diện thực thể có tên

- **Dựa trên từ điển:**
  - Có thể phát hiện được các thực thể phổ biến
  - Yêu cầu xây dựng từ điển tên riêng
  - Không xử lý được nhập nhằng
- **Dựa trên biểu thức chính quy**
  - Sử dụng kiến thức chuyên gia
  - Có thể phát hiện được các mẫu phổ biến

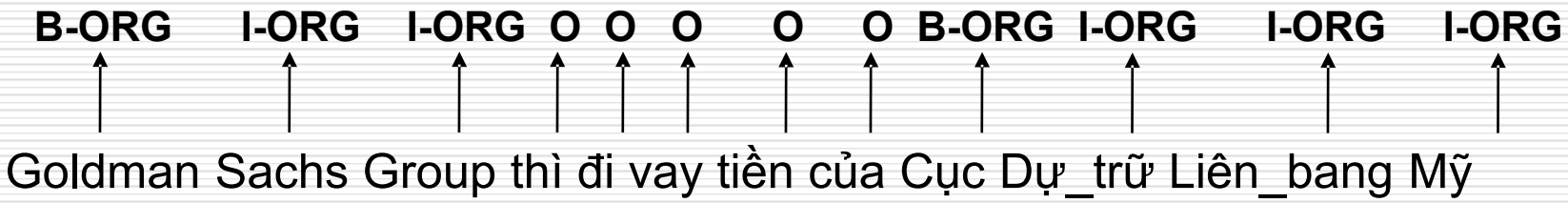


# 2. Nhận diện thực thể có tên

- **Dựa trên học máy:**
  - Yêu cầu dữ liệu huấn luyện
  - Độ chính xác không thay đổi nhiều giữa các lĩnh vực
  - Quy về bài toán gán nhãn chuỗi BIO
  - Đầu vào là một câu
  - Đầu ra là nhãn của mỗi từ trong câu

# 2. Nhận diện thực thể có tên

- **Gán nhãn chuỗi BIO:**
  - B: Begin
  - I: Inside
  - O: Outside



# 2. Nhận diện thực thể có tên

- **Tập đặc trưng**
  - Các từ trong cửa sổ [-k, k] (k= 2, 3)
  - Hình thái từ:
    - Viết hoa, viết thường
    - Chữ số
    - Dấu câu
  - Loại từ: Đầu ra của bài toán gán nhãn từ loại
  - Phạm vi từ: Đầu ra của bài toán phân cụm

# 2. Nhận diện thực thể có tên

- **Đánh giá kết quả**

**Table 4.** Accuracy of our NER system with default and generated PoS, chunking tags; and without PoS and chunking tags

| Setting | Precision | Recall | $F_1$ |
|---|---|---|---|
| Default PoS and chunking tags | 93.87 | 93.99 | 93.93 |
| PoS and chunking tags generated by NNVLP [7] | 90.21 | 86.72 | 88.43 |
| PoS and chunking tags generated by Underthesea | 90.28 | 88.35 | 89.3 |
| **Without PoS, chunking tags** | 89.91 | 90.15 | **90.03** |

**Table 5.** Proposed NER systems without chunking tag-based features. We compare default PoS with PoS generated by other tools.

| Setting | Precision | Recall | $F_1$ |
|---|---|---|---|
| Default PoS tags | 90.13 | 90.55 | 90.34 |
| PoS by NNVLP [7] | 90.05 | 85.65 | 88.31 |
| PoS by Underthesea | 90.27 | 88.58 | 89.42 |
| PoS by Pyvi | 90.16 | 88.72 | 89.43 |
| PoS by Vtik | 89.62 | 86.42 | 87.99 |
| PoS by VnMarMoT [19] | 90.51 | 89.15 | 89.83 |
| **Without PoS, chunking tags** | 89.91 | 90.15 | **90.03** |

# 2. Nhận diện thực thể có tên

- **Đánh giá kết quả**

**Table 6.** Accuracy of NER system with default and generated word segmentation. We did not use features based on PoS, chunking tags here.

| Setting | Precision | Recall | $F_1$ |
|---|---|---|---|
| Default Word segmentation | 89.91 | 90.15 | 90.03 |
| Word segmentation generated by UETSegmenter | 87.67 | 84.95 | 86.29 |
| Word segmentation generated by RDRsegmenter | 89.05 | 84.98 | **86.97** |

**Table 7.** Accuracy of NER system with syllable-based and word-based model. We do not use features based on PoS and chunking tags. "ws" stands for word segmentation

| Setting | Precision | Recall | $F_1$ |
|---|---|---|---|
| Syllable-based model | 88.78 | 82.94 | 85.76 |
| Word-based model with gold ws | 89.91 | 90.15 | 90.03 |
| Word-based model with ws generated by RDRsegmenter | 89.05 | 84.98 | **86.97** |

# 2. Nhận diện thực thể có tên

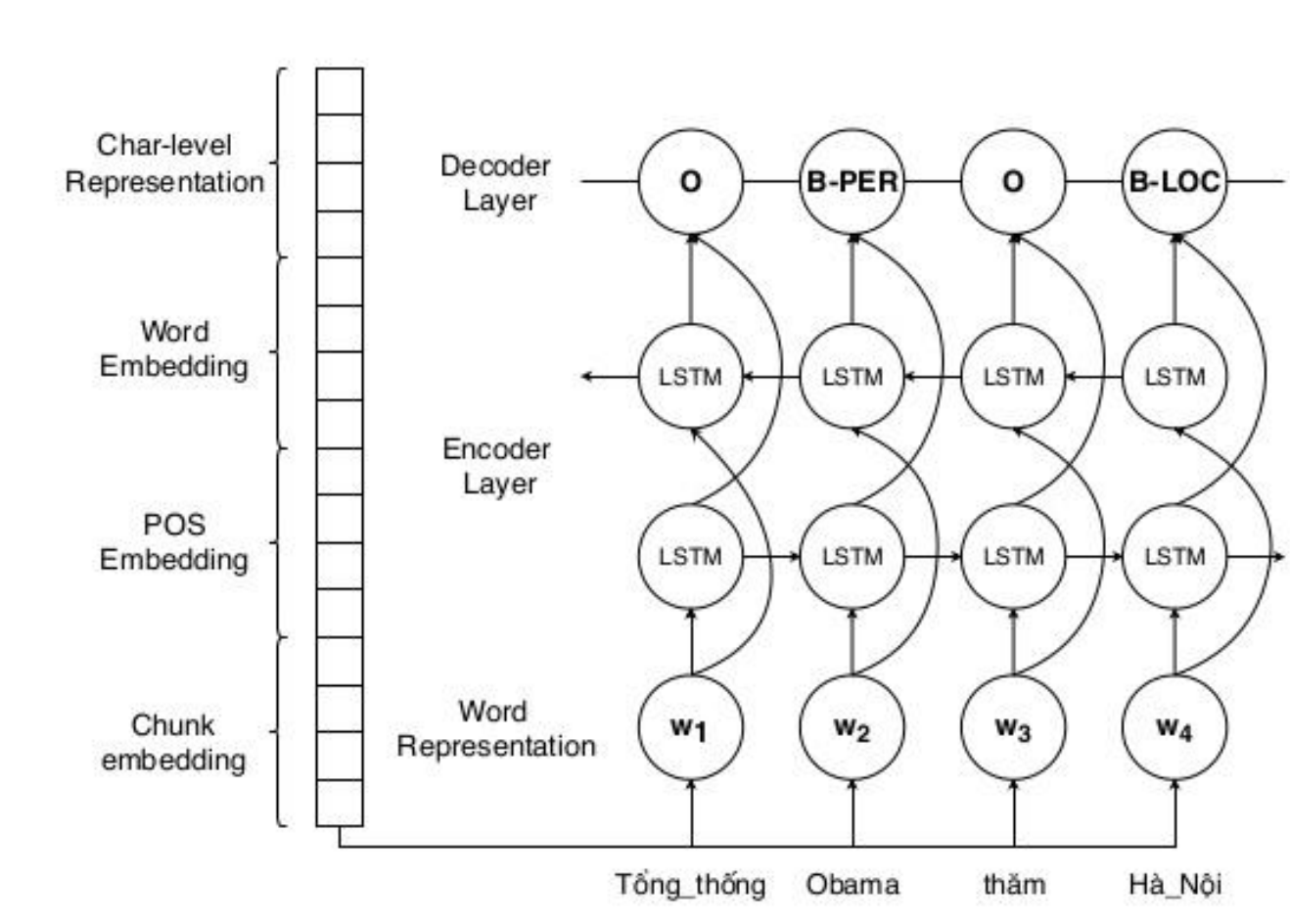
- **Đánh giá kết quả**
  - **Word**: Các từ trong cửa sổ
  - **Word shapes**: Hình thái từ
  - **w2v**: Biểu diễn từ nhúng
  - **Cluster**: Biểu diễn phân cụm Brown

**Table 8.** Impact of word representation-based features. w2v denotes features based on word embeddings. "cluster" denotes cluster-based features.

| Setting | Precision | Recall | $F_1$ |
|---|---|---|---|
| (1) = all features with default PoS, Chunk | 93.87 | 93.99 | 93.93 |
| (2) = (1) - cluster - w2v | 91.66 | 92.02 | 91.84 |
| (4) = word + word shapes + default PoS | 88.01 | 87.95 | 87.98 |
| (5) = word + word shapes + cluster + w2v | 89.91 | 90.15 | 90.03 |
| (6) = word + word-shapes | 88.17 | 88.08 | 88.13 |
| (7) = word + word-shapes + w2v | 88.69 | 88.72 | 88.70 |
| (8) = word + word-shapes + cluster | 88.96 | 89.99 | 89.97 |

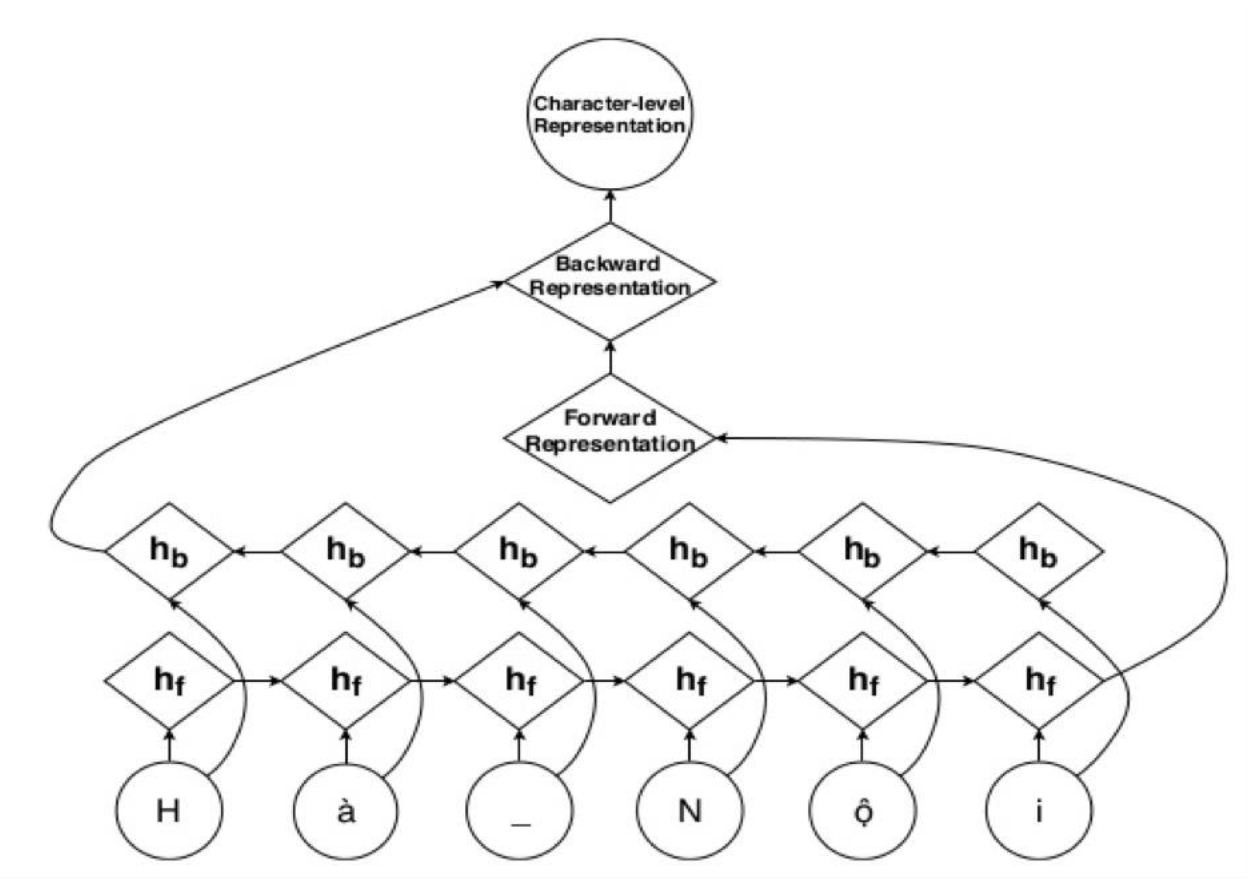
# 2. Nhận diện thực thể có tên

- **NER dựa trên RNN**



# 2. Nhận diện thực thể có tên

- **NER dựa trên RNN**
  - Học biểu diễn ký tự



# 2. Nhận diện thực thể có tên

- **NER dựa trên RNN**
  - Đánh giá kết quả

| Method | P | R | F1 | F1 (w.o char) |
|---|---|---|---|---|
| Feature-rich CRFs [25] | 93.87 | 93.99 | 93.93 | - |
| NNVLP [7] | 92.76 | 93.07 | 92.91 | - |
| BiLSTM-CRFs | 90.97 | 87.52 | 89.21 | 76.43 |
| BiLSTM-CRFs + POS | 90.90 | 90.39 | 90.64 | 86.06 |
| BiLSTM-CRFs + Chunk | 95.24 | 92.16 | 93.67 | 87.13 |
| BiLSTM-CRFs + POS + Chunk | **95.44** | **94.33** | **94.88** | 91.36 |

# 2. Nhận diện thực thể có tên

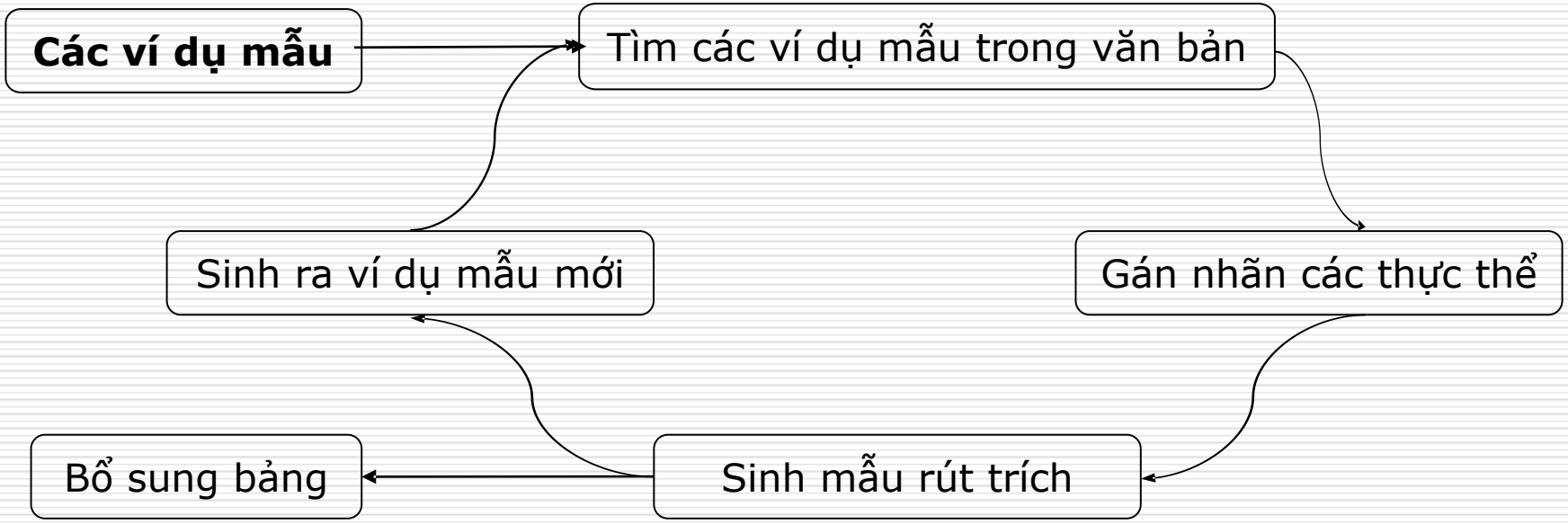
- **NER dựa trên RNN**
  - Đánh giá kết quả

| Method | P | R | F1 | F1 (w.o char) |
|---|---|---|---|---|
| Feature-rich CRFs [25] | 93.87 | 93.99 | 93.93 | - |
| NNVLP [7] | 92.76 | 93.07 | 92.91 | - |
| BiLSTM-CRFs | 90.97 | 87.52 | 89.21 | 76.43 |
| BiLSTM-CRFs + POS | 90.90 | 90.39 | 90.64 | 86.06 |
| BiLSTM-CRFs + Chunk | 95.24 | 92.16 | 93.67 | 87.13 |
| BiLSTM-CRFs + POS + Chunk | **95.44** | **94.33** | **94.88** | 91.36 |

# 3. Rút trích quan hệ không giám sát

- Học có giám sát có độ chính xác cao nhưng đòi hỏi DL huấn luyện
- Học không giám sát tận dụng được lượng DL lớn nhưng có độ chính xác thấp hơn
- Giám sát từ xa tận dụng được cơ sở tri thức và cải thiện độ chính xác so với học không giám sát

# 3. Rút trích quan hệ không giám sát




**Các ví dụ mẫu**

Do người dùng cung cấp, sau đó hệ thống tự động rút trích ra từ văn bản. Ví dụ: Quan hệ <tập đoàn, trụ sở>

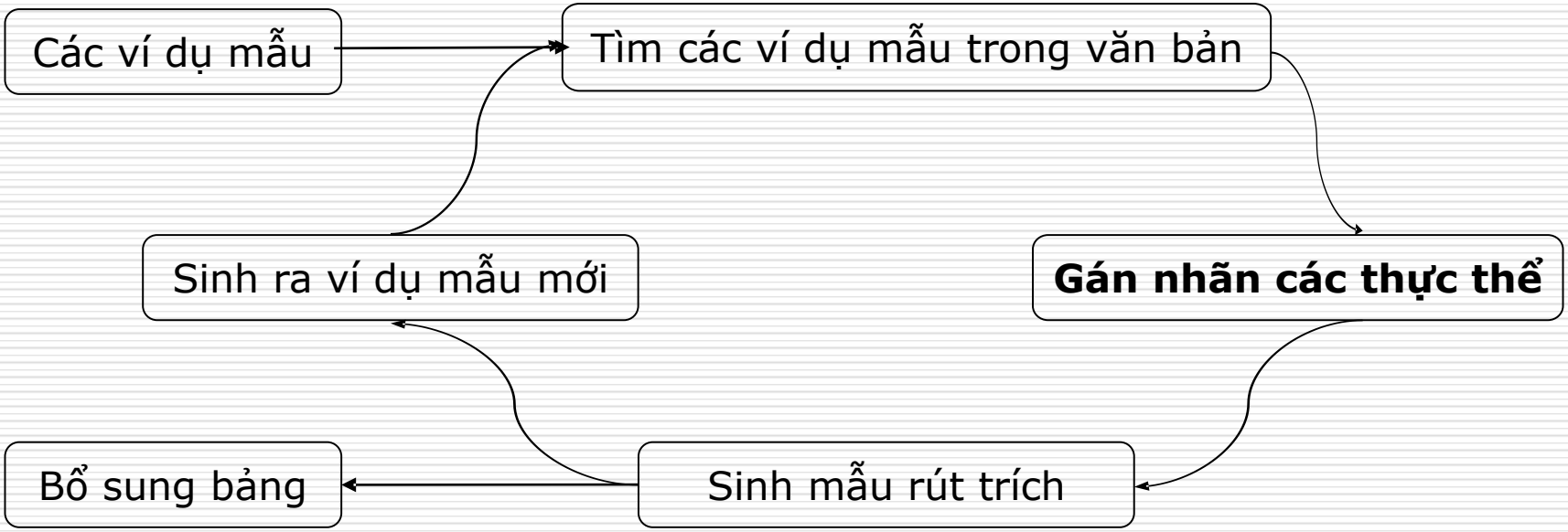
- <Microsoft, Redmond>
- <Exxon, Irving>
- <IBM, Armonk>

# 3. Rút trích quan hệ không giám sát


**Tìm các ví dụ mẫu trong văn bản**

- "Máy chủ của **Microsoft** nằm ở trụ sở chính **Redmon**"
- "Tin đồn rút nhân viên khỏi Iraq đến từ trụ sở chính của **Exxon**, **Irving**..."
- "... vừa nhận được email từ trụ sở chính của **Boeing** ở **Seattle**."

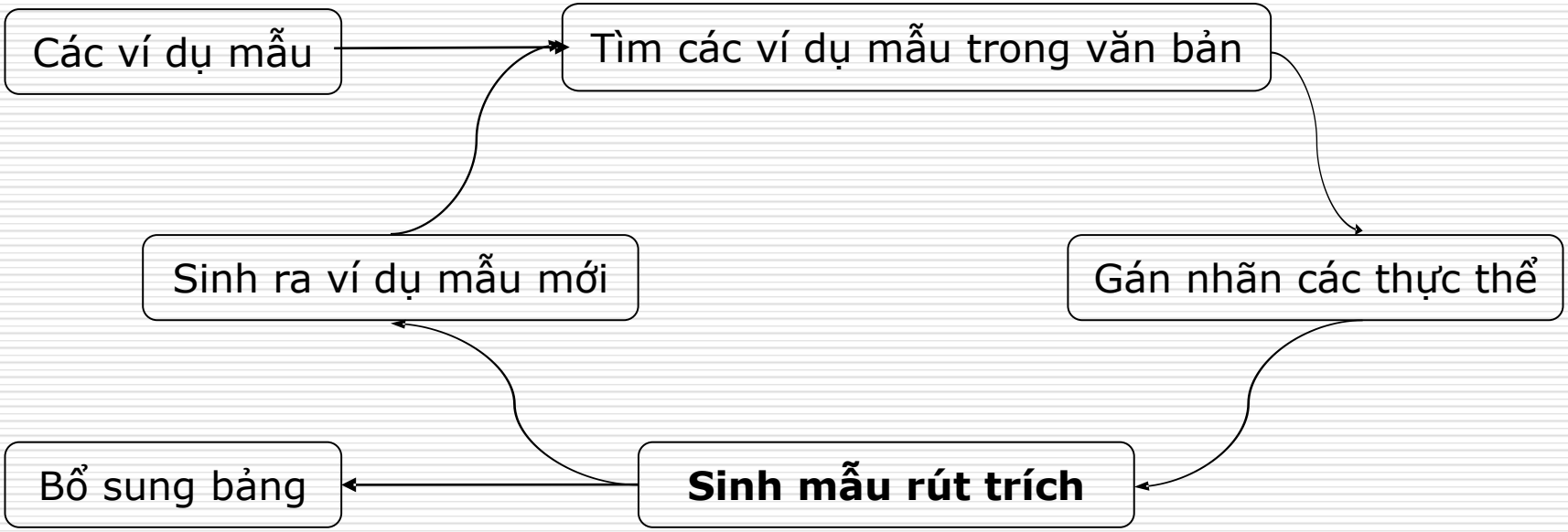
# 3. Rút trích quan hệ không giám sát




**Gán nhãn thực thể**

- "Máy chủ của <ORG> nằm ở trụ sở chính <LOC>"
- "Tin đồn rút nhân viên khỏi Iraq đến từ trụ sở chính của <ORG>, <LOC>..."
- "... vừa nhận được email từ trụ sở chính của <ORG> ở <LOC>."

# 3. Rút trích quan hệ không giám sát




**Sinh mẫu rút trích**

- **Mô hình 5-tuple:**

<trái, thẻ 1, giữa, thẻ 2, phải>

# 3. Rút trích quan hệ không giám sát

**Sinh mẫu rút trích:**

- Cho 2 5-tuple có cùng tag1và tag2:
  - t= {l, tag1, m, tag2, r}
  - t'= {l', tag1, m', tag2, r'}
- **Độ tương đồng:**

  match(t, t') = l*l'+ m*m' + r*r'
- Phân cụm các 5-tuple dựa trên độ tương đồng
- Lấy centroid của c làm mẫu rút trích:

  p = {lc, tag1, mc, tag2, rc}

# 3. Rút trích quan hệ không giám sát

**Algorithm** GenerateTuples(tập mẫu)

1. **foreach** đoạn ∈ tập văn bản **do**
2. {<$o$, $l$>, <$l_s$, $t_1$, $m_s$, $t_2$, $r_s$>} = CreateOccurrence(đoạn);
3. $T_C$ = <$o$, $l$>;
4. $Sim_{Best}$ = 0;
5. **foreach** $p$ ∈ tập mẫu
6. sim = Match(<$l_s$, $t_1$, $m_s$, $t_2$, $r_s$>, $p$);
7. **if** (sim ≥ $T_{sim}$) **then**
8. UpdatePatternSelectivity(p, $T_C$);
9. **if**(sim ≥ $Sim_{Best}$) **then**
10. $Sim_{Best}$= sim;
11. $P_{Best}$= p;
12. **endif**
13. **endif**
14. **endfor**
15. **if**($Sim_{Best}$≥ $T_{sim}$) **then**
16. CandidateTuples[$T_C$].Patterns[$P_{Best}$] = $Sim_{Best}$;
17. **endif**
18. **endfor**
19. **return** CandidateTuples;

# 3. Rút trích quan hệ không giám sát

**Đánh giá mẫu:**

- Với mỗi ví dụ <org, loc>, phân loại:
  - Positive nếu đã tồn tại ví dụ mẫu
  - Negative nếu tồn tại ví dụ mẫu <org, loc'>
  - Unknown nếu <org, *> chưa tồn tại
- Độ tin tưởng của mẫu P:

$$conf(P) = \frac{P.positive}{P.positive + P.negative}$$

  - P.positive: số ví dụ positive khớp với P
  - P.negative: số ví dụ negative khớp với P

# 3. Rút trích quan hệ không giám sát

**Đánh giá ví dụ:**

- Độ tin tưởng của ví dụ T = {org, loc}

$$Conf(T) = 1 - \prod_{i=0}^{|P|} (1 - (Conf(P_i) \cdot Match(C_i, P_i)))$$

  - P = {Pi} là tập các mẫu sinh ra ví dụ T
  - Ci là 5-tuple ứng với đoạn văn bản khớp với Pi với độ tương tự Match(Ci, Pi)
  - Tập ví dụ mẫu = {T| Conf(T) > $\tau$t}

# 3. Rút trích quan hệ không giám sát

**Ưu điểm:**

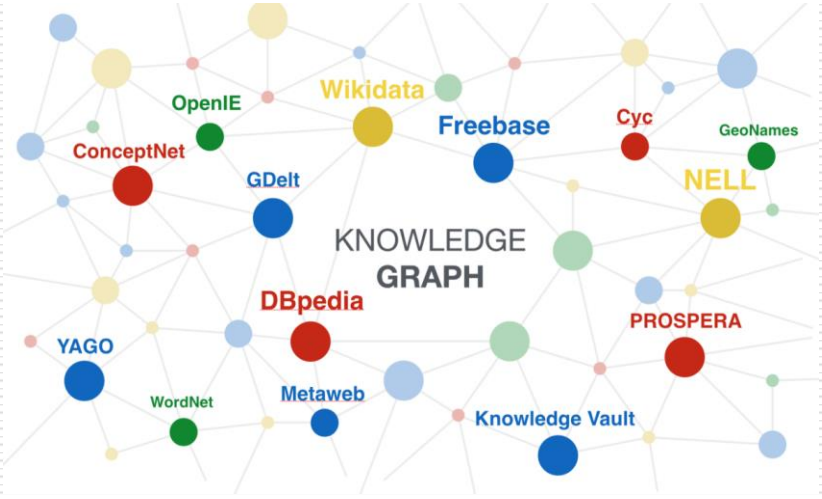
- Tận dụng được dữ liệu không có nhãn
- Chỉ cần một số ít ví dụ mẫu gốc

**Nhược điểm:**

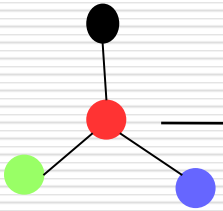
- Vẫn yêu cầu gán nhãn thủ công từ người dùng
- Quá trình lặp dẫn đến suy giảm chất lượng

# 4. Giám sát từ xa

- **Freebase** là cơ sở tri thức lớn và có chất lượng về quan hệ giữa các thực thể
- **Freebase** được xây dựng từ **Wikipedia**
- Giám sát từ xa:
  - Freebase giám sát quá trình rút trích từ văn bản
  - Freebase + corpus = dữ liệu có nhãn



# 4. Giám sát từ xa



Freebase

{<**entity**$_1$, relation$_i$, **entity**$_2$>}

match

corpus

Sentence 1: …. **entity**$_1$…. **entity**$_2$….
Sentence 2: …. **entity**$_2$…. **entity**$_1$….
…
Sentence *n*: …. **entity**$_2$…. **entity**$_1$….

{**features**}

({**features**}, relation$_i$)

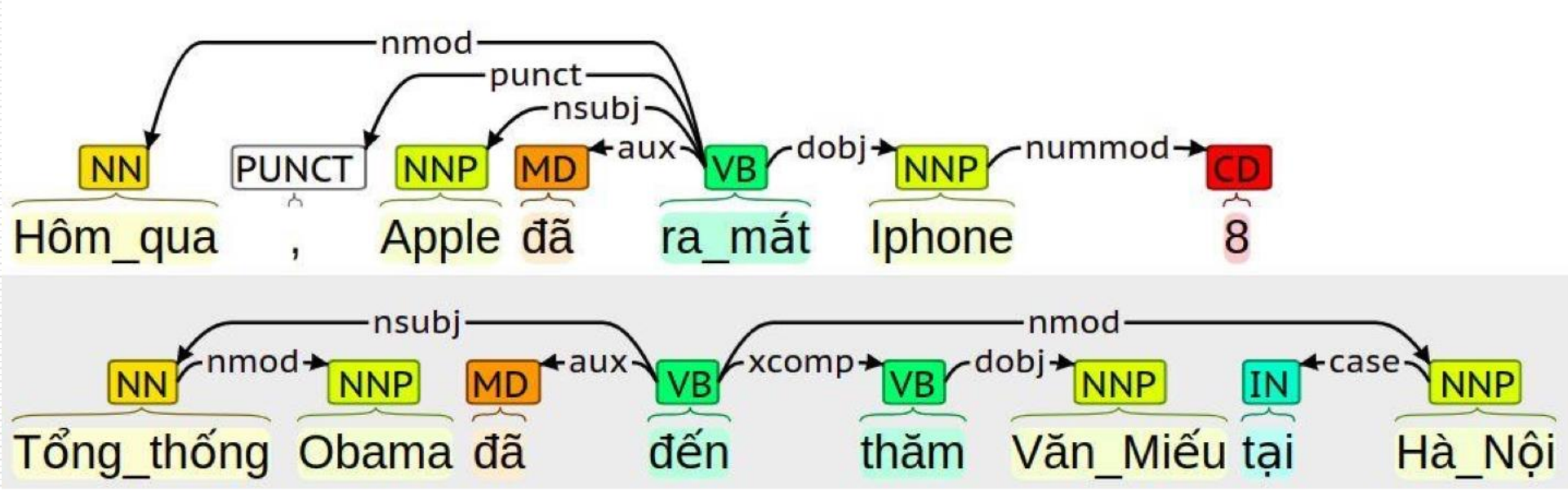
{<**entity**$_1$', relation$_i$, **entity**$_2$'>} → ({**features'**}, relation$_i$)

{<**entity**$_1$'', relation$_j$, **entity**$_2$''>} → ({**features''**}, relation$_j$)

**BỘ PHÂN LOẠI ĐA LỚP *f*: {relation$_1$, relation$_2$, …, relation$_m$}**

# 4. Giám sát từ xa

- Quan hệ phụ thuộc



# 5. Phân giải đồng tham chiếu

- Là quá trình phát hiện một cặp từ hay cụm từ trong văn bản chỉ đến cùng một thực thể
- Đồng tham chiếu là hiện tượng phổ biến trong ngôn ngữ
- Có vai trò quan trọng với các ứng dụng khai phá văn bản

# 5.1. Các loại đồng tham chiếu

- **Đại từ làm chủ ngữ:** "**Cô ta** đang học trực tuyến"
- **Đại từ làm tân ngữ:** "Hãy liên lạc với **anh ấy** ngay"
- **Đại từ sở hữu:** "Lịch trình của **chúng ta** đã được thống nhất"
- **Tên riêng:** "Thủ tướng **Nguyễn Xuân Phúc** tuyên bố giãn cách xã hội."
- **Apposition:** "**Phạm Nhật Vượng, Chủ tịch Vingroup** là một trong số các tỉ phú được Forbes nêu tên."
- **Động từ 'là' :** "Park Hang Seo **là** HLV trưởng đội tuyển bóng đá nam Việt Nam."

# 5.1. Các loại đồng tham chiếu

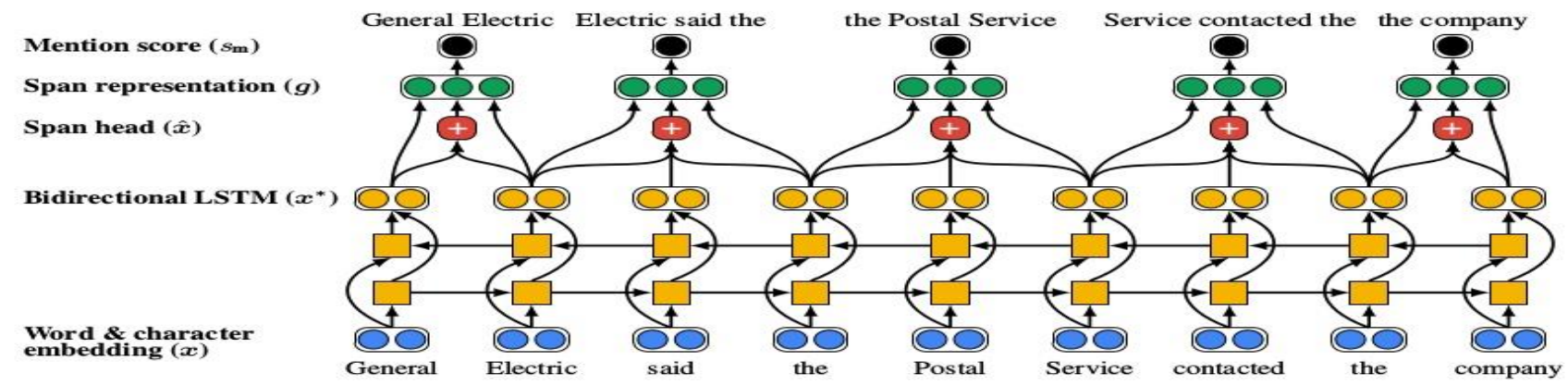
- **Nhóm người:** "**Mây Trắng** tuyên bố tái hợp. Nhóm dự định ra mắt album mới đầu năm sau."
- **Thuộc tính - giá trị:** "Giá cổ phiếu **VIC** là **94.800 VND**"
- **Thứ tự:** "IBM và Microsoft là những ứng cử viên **cuối cùng**, nhưng đại diện nhà đầu tư ưu tiên ứng cử viên **thứ hai**."
- **Bộ phận - toàn thể:** "Vinfast mới ra mắt dòng xe mới. **Bộ truyền động** sử dụng công nghệ CVT vô cấp tiên tiến."

# 5.2. Các phương pháp truyền thống

- Tập trung vào đại từ là loại xuất hiện phổ biến nhất
- Dùng thông tin ngôn ngữ để phát hiện các ứng cử viên phía trước
- Loại bỏ các ứng cử viên dựa trên tính chất như giới tính, số ít số nhiều, …
- Tính điểm các ứng cử viên
  - So khớp
  - Luật
  - Học máy

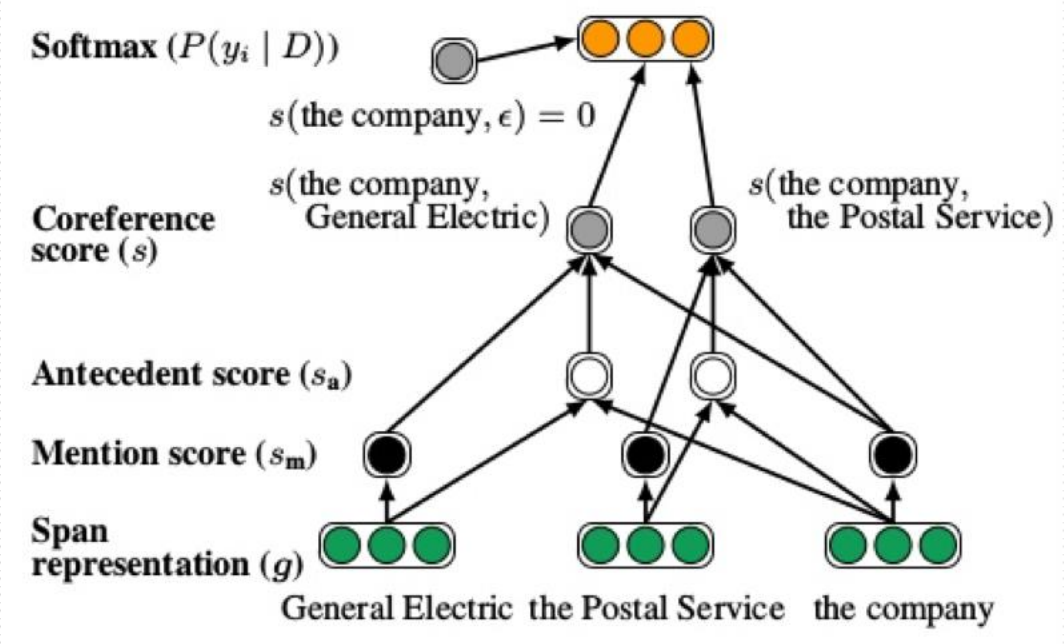
# 5.3. Phương pháp dựa trên neural network

- Hạn chế sử dụng các đặc trưng phức tạp
- Hạn chế sử dụng phân tích cú pháp
- Tận dụng biểu diễn học trước
- Thách thức:
  - Sử dụng thông tin thay thế cho thông tin cú pháp
  - Biểu diễn cụm từ, ngữ cảnh
  - Phân giải đồng tham chiếu bản chất là một bài toán phân cụm cứng trong phạm vi văn bản



# 5.3. Phương pháp dựa trên neural network

- **Kiến trúc mô hình**
- **Biểu diễn đoạn:** $g_i = [x^*_{START(i)}, x^*_{END(i)}, \hat{x}_i, \Phi(i)]$
- **Điểm mention:** $s_m(i) = w_m * FFNN_m(g_i)$
- **Điểm tương đồng:** $s_a(i, j) = w_a * FFNN_a([g_i, g_j, g_i \circ g_j, \Phi(i, j)])$



# 5.3. Phương pháp dựa trên neural network

- **Đánh giá kết quả**

| | Avg. F1 | Δ |
|---|---|---|
| Our model (ensemble) | 69.0 | +1.3 |
| Our model (single) | 67.7 | |
| – distance and width features | 63.9 | -3.8 |
| – GloVe embeddings | 65.3 | -2.4 |
| – speaker and genre metadata | 66.3 | -1.4 |
| – head-finding attention | 66.4 | -1.3 |
| – character CNN | 66.8 | -0.9 |
| – Turian embeddings | 66.9 | -0.8 |

# Tài liệu tham khảo


- Zdravko Markov and Daniel T. Larose. **Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage**, Wiley, 2007, ISBN: 978-0-471-66655-4.
- Web Mining, ĐHBK Hà Nội.

# Thank you!

# Q&A