HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

WEB MINING

LECTURE 07: INFORMATION EXTRACTION

ONE LOVE. ONE FUTURE.

Content

1. Information extraction system architecture
2. Named Entity Recognition
3. Unsupervised relation extraction
4. Distant supervision for relation extraction
5. Coreference resolution

1. Information extraction system architecture

- Information extraction is the process of finding entities and relationships between these entities in a text
- Extracting information for text mining is more precise and concise than tasks such as text classification or text labeling.
- Predefined entity types and relationships

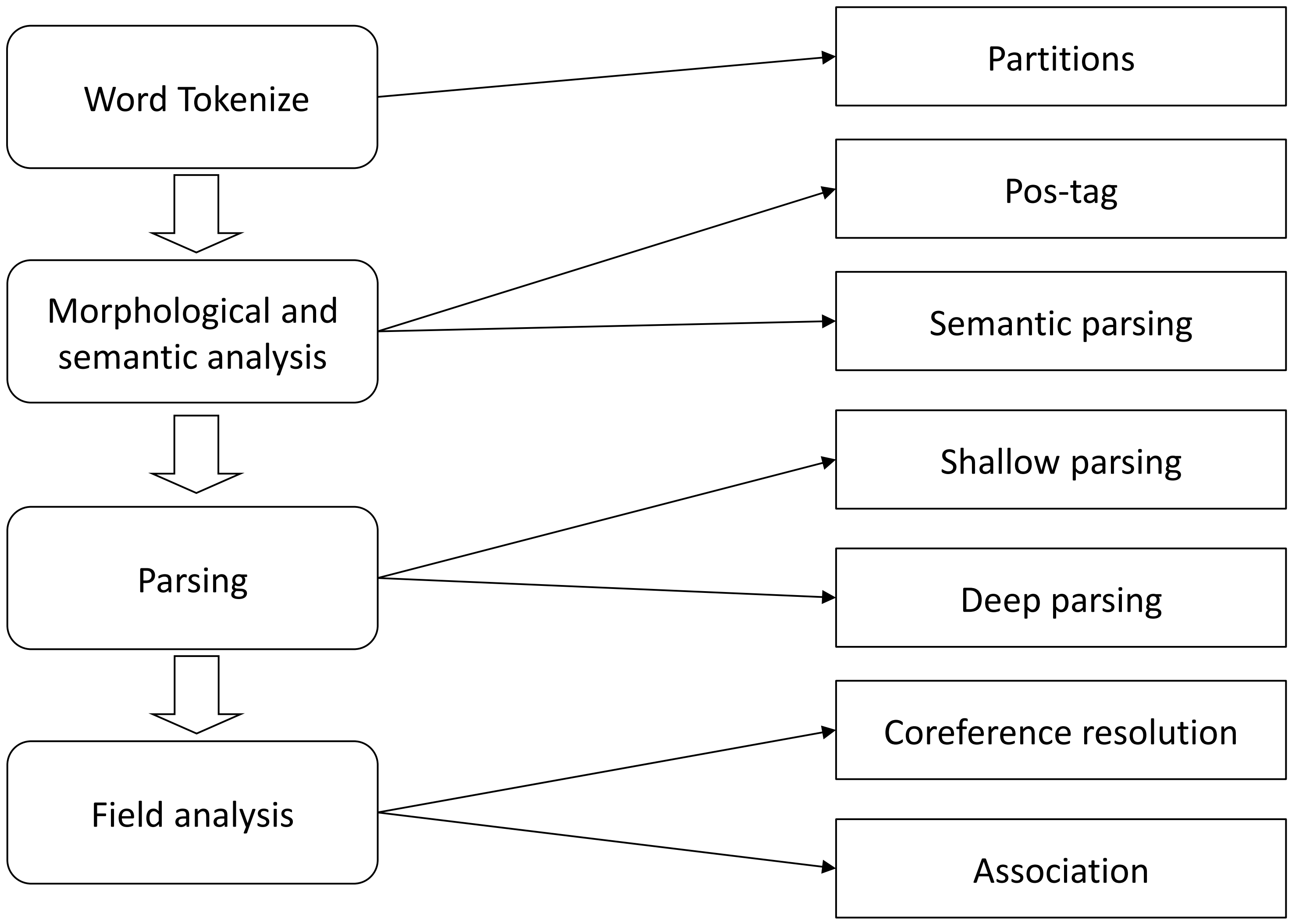
Assumptions of information extraction

- Information is presented explicitly and requires no inference
- A small number patterns can summarize the content of the text
- Necessary information appears locally in the text

Types of information extracted

- Entities: People, organizations, locations, etc.
- Attributes (of the entity): Title, age, type of organization...
- Fact: the relationship between employees and the company, the relationship between viruses and diseases, etc.
- Events: two companies merging, earthquake, terrorism,...

Information extraction system architecture



Named Entity Recognition

- Detects named entities in text and classifies into predefined classes

[Forbes]$_{\text{ORG}}$: [Việt Nam]$_{\text{LOC}}$ có 4 tỷ phú

Phrase chunking

- Detect noun and verb phrases in sentences

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

Relation Extraction

- Extract relationships between entities (attributes, events)

BORROW

Goldman Sachs Group thì đi vay tiền của Cục Dự trữ Liên bang Mỹ.

ORIGIN

Aikido là một môn võ thuật Nhật Bản hiện đại

Coreference resolution

- Detect occurrence of the same entity as different references

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. Named Entity Recognition

- Based on the dictionary:
  - Can detect common entities
  - Request to build a dictionary of own names
  - Can't handle ambiguity
- Based on regular expression
  - Using expert knowledge
  - Common patterns can be detected

Based on machine learning

- Request training data
- Accuracy does not vary much between fields
- Problem of labeling the string BIO
  - Input is a sentence
  - The output is the label of each word in the sentence

BIO scheme

- B: Begin
- I: Inside
- O: Outside

B-ORG	I-ORG	I-ORG	O	O	O	O	O	B-ORG	I-ORG	I-ORG	I-ORG
Goldman	Sachs	Group	thì	đi	vay	tiền	của	Cục	Dự_trữ	Liên_bang	Mỹ

Feature set

- Words in window [-k, k] (k = 2, 3)
- Word form:
  - Uppercase, lowercase
  - Number
  - Punctuation
- Word type: Output of the word-type labeling problem
- Word postion: Output of the chunking problem

NER based on CRF

- [1]: Using golden PoS and chunking
- [2, 3]: Automatic PoS and chunking by NNVLP engine and Underthesea
- [4]: No PoS and chunking

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**

from P. Q. N. Minh. *"Feature-rich CRFs for Vietnamese NER"*. CICLING 2018

Evaluation

- [1]: Using golden PoS
- [2-6]: Automatic PoS from tools
- [7]: No PoS and chunking

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**

Evaluation (cont.)

- [1]: Use golden word segmentation
- [2,3]: Automatic tokenizer using UETSegmenter and RDRSegmenter

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**

Evaluation (cont.)

- [1]: syllable-based model (no word tokenize)
- [2]: Use standard separator
- [3]: Automatic word tokenizer with RDR Segmenter tool

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**

Evaluation (cont)

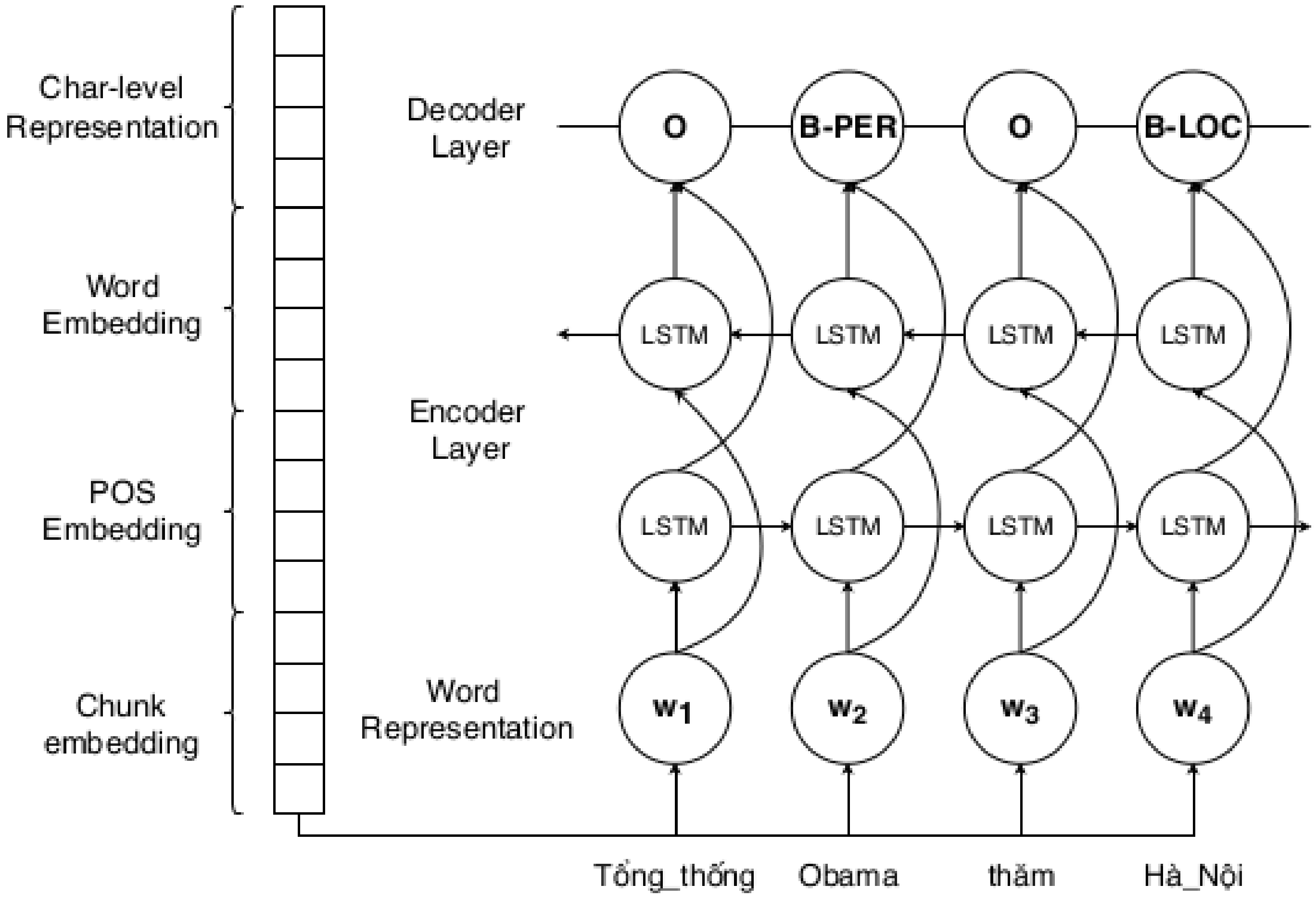
- Word: word in window
- Word shapes: word form
- w2v: word embedding
- Cluster: Brown clustering representation

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

RNN-based NER

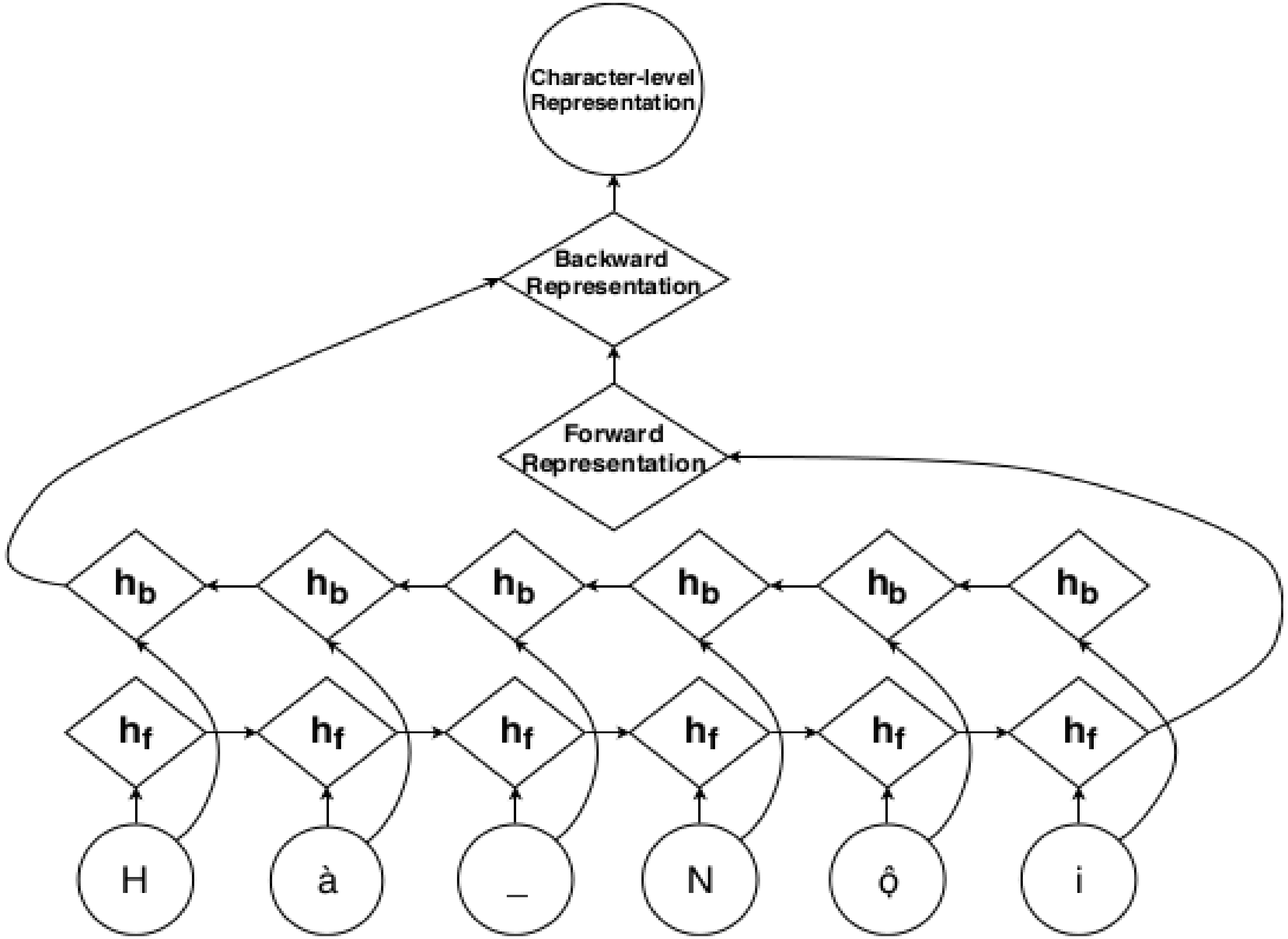
from Nguyen et al. “*Neural sequence labeling for Vietnamse POS tagging and NER*”. RIVF 2019



Input layer

- Combined Embedded Representation:
  - Word representation: Using word embedding pre-trained by word2vec on 2 million documents
  - Character representation: Using bidirectional LSTM network to learn character representation with random initialization
  - Word type: One-hot representation
  - Chunking: One-hot representation

Learn to represent characters



Bidirectional LSTM

- Using two LSTM networks in forward and reverse direction
  - Purpose: Words at the beginning of a sentence can use both the information at the end of the sentence to make predictions and vice versa
- Outputs are coupled to feed into output layer

Output layer

- Predict BIO labels for entity types
  - For example: With 3 entity types ORG, PER, LOC, the label set has 7 labels (B-ORG, I-ORG, B-PER, I-PER, B-LOC, I-LOC, O)
- The output layer can be fed into a model of CRFs to represent the relationship with the label at a previous point in time through transition probabilities.

Evaluation

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

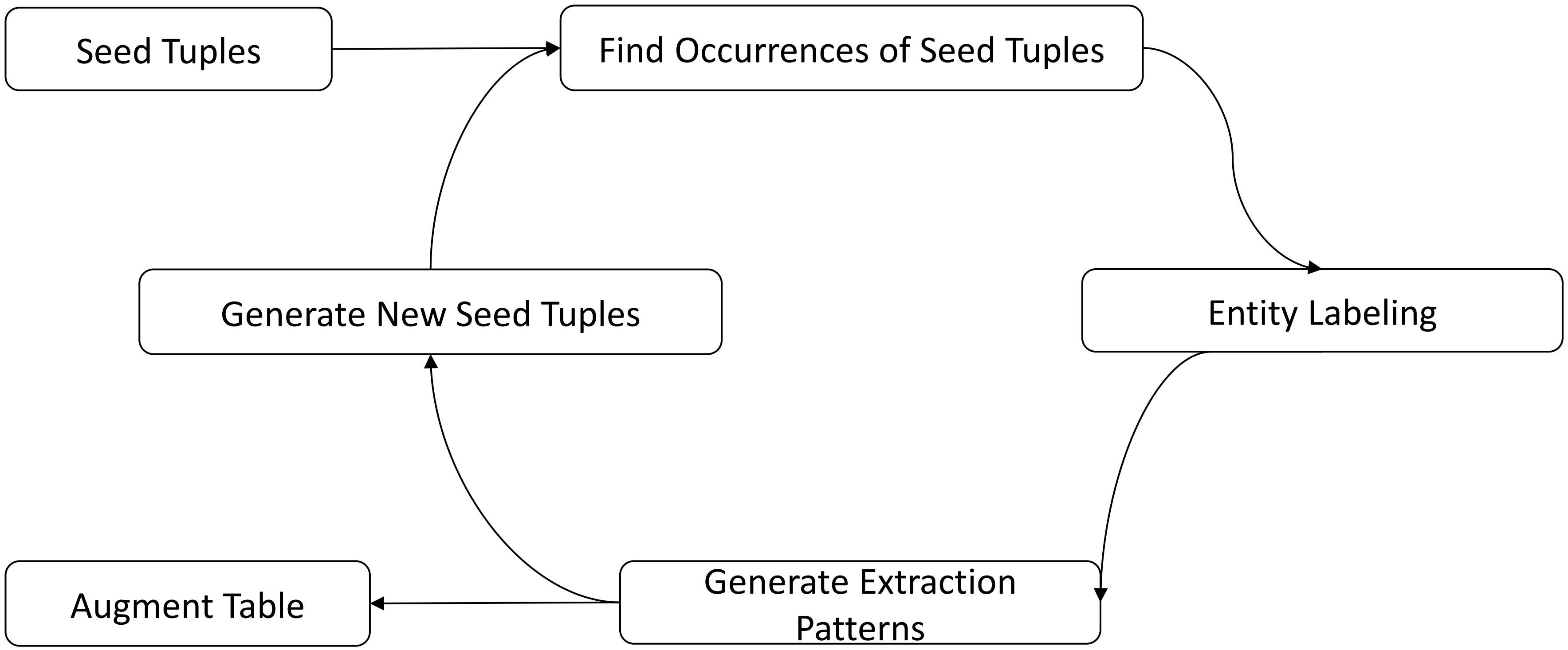
BiLSTM-CRFs use additional PoS and clustering information

BiLSTM-CRFs don't incorporate character level representation

3. Unsupervised Relation Extraction

- Supervised learning is highly accurate but requires training data
- Unsupervised learning takes advantage of large amounts of data but has less accuracy
- Distant supervision leverages the knowledge base and improves accuracy over unsupervised learning

Snowball


Seed Tuples
Find Occurrences of Seed Tuples
Generate New Seed Tuples
Entity Labeling
Augment Table
Generate Extraction Patterns

Seed Tuples

- User-provided
- Then the system automatically extracts from the text
- Ex: Relationship <tập đoàn, trụ sở>
  - <Microsoft, Redmond>
  - <Exxon, Irving>
  - <IBM, Armonk>

Search seed tuples

- “Hệ thống máy chủ của **Microsoft** nằm ở trụ sở chính **Redmon**”
- “**Exxon**, **Irving** đang dần trở thành tập đoàn dầu khí…”
- “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**.”

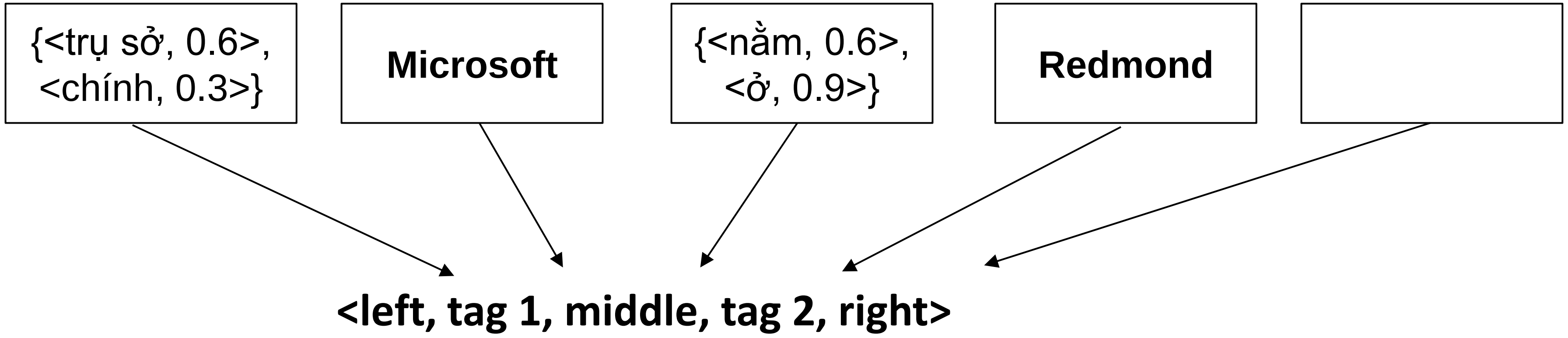
Entity Labeling

- “Hệ thống máy chủ của **<ORG>** nằm ở trụ sở chính **<LOC>**”
- “**<ORG>**, **<LOC>** đang dần trở thành tập đoàn dầu khí...”
- “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>**.”

Generate 5-tuple

- 5-tuple: <left, tag 1, middle, tag 2, right>
- Left: k words to the left along with the weight vector
- Tag 1: first entity
- Middle: words in the middle along with the weight vector
- Tag 2: second entity
- Right: k words to the right along with the weight vector

Generate 5-tuple (cont)



Generate 5-tuple (cont.)

{<trụ sở, 0.6>, <chính, 0.3>}	**ORG**	{<nằm, 0.6>, <ở, 0.9>}	**LOC**	
	ORG	{<’,’, 0.7>}	**LOC**	{<đang, 0.2>, <dần, 0.1>, <trở_thành, 0.15>}
{<trụ sở, 0.6>, <chính, 0.3>, <của, 0.5>}	**ORG**	{<’,’, 0.7>}	**LOC**	
{<trụ sở, 0.6>, <chính, 0.3>, <của, 0.5>}	**ORG**	{<ở, 0.95>}	**LOC**	

Generate Extraction Patterns

- Given 2 5-tuples with the same tag_1 and tag_2:
  - $t = \{l, tag_1, m, tag_2, r\}$
  - $t' = \{l', tag_1, m', tag_2, r'\}$
- Similarity: $match(t, t') = l \cdot l' + m \cdot m' + r \cdot r'$
- Clustering 5-tuples based on similarity
- For each cluster, take the centroid of c as extraction patterns

$p = \{l_c, tag_1, m_c, tag_2, r_c\}$

Generate New Seed Tuples

```
Algorithm GenerateTuples
    foreach paragraph ∈ corpus  do
        {<o, l>, <l_s, t_1, m_s, t_2, r_s>} = CreateOccurrence(paragraph);
        T_C = <o, l>;
        Sim_Best = 0;
        foreach p ∈ Patterns
            sim = Match(<l_s, t_1, m_s, t_2, r_s>, p);
            if (sim ≥ τ_sim) then
                UpdatePatternSelectivity(p, T_C);
                if (sim ≥ Sim_Best) then
                    Sim_Best = sim;
                    P_Best = p;
                endif
            endif
        endfor
        if (Sim_Best ≥ τ_sim) then
            CandidateTuples[T_C].Patterns[P_Best] = Sim_Best;
        endif
    endfor
    return CandidateTuples;
```

Patterns Evaluation

- for each example <org, loc>, classify:
  - Positive if an pattern already exists
  - Negative if exists pattern <org, loc’>
  - Unknown if <org, *> not exist yet
- Confidence of sample P:

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

- P.positive: number positive examples matching P
- P.negative: number negative examples matching P

Example Evaluation

- Example confidence T = {org, loc}

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

- $P = \{P_i\}$ set of patterns that generate for example T
- C_i is 5-tuple corresponding to the text matches P_i with similarity Match(C_i, P_i)
- Pattern example set= $\{T \mid \text{Conf}(T) > \tau_t\}$

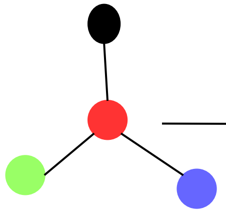
Pros, Cons

- Advantages:
  - Take advantage of unlabeled data
  - Just a handful of original pattern examples
- Defect:
  - Still requires manual labeling from users
  - Iterative process leads to quality degradation

4. Distant supervision

- Freebase is a large and quality knowledge base about relationships between entities
- Freebase is built from Wikipedia
- Distant supervision:
  - Freebase supervises the process of extracting relations from the text
  - Freebase + corpus = labeled data

Distant supervision



Freebase → $\{<\textbf{entity}_1, \text{relation}_i, \textbf{entity}_2>\}$

↕ match

corpus →

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

} {**features**}

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

$\{<\textbf{entity}_1', \text{relation}_i, \textbf{entity}_2'>\} \rightarrow (\{\textbf{features'}\}, \text{relation}_i)$

$\{<\textbf{entity}_1'', \text{relation}_j, \textbf{entity}_2''>\} \rightarrow (\{\textbf{features''}\}, \text{relation}_j)$

➡ **multiclass classifier *f*: {relation$_1$, relation$_2$, …, relation$_m$}**

Distant supervision

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

match

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

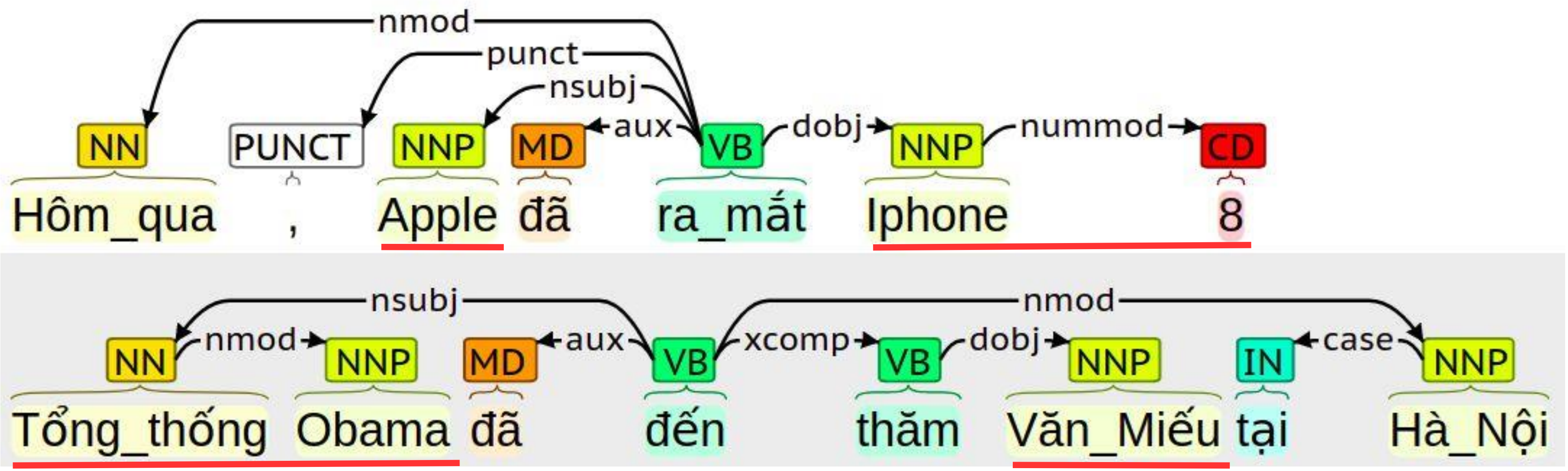
{**features**}

PREDICT: *f*({**features**})

Feature set

- Words and POS in between two entities and PoS
- Order of two entities
- Words and POS of k words on the left
- Words and POS of k words on the right
- Entity Type
- The path between two entities in the dependency tree

Dependency tree



Kiem-Hieu Nguyen. "*BKTreebank: Building a Vietnamese dependency treebank*". LREC 2018.
http://45.117.171.213/bknlptool/

5. Coreference resolution

- Coreferencing resolution is the process of detecting a pair of words or phrases in the text that refer to the same entity
- Coreferencing is a common phenomenon in languages
- Coreferencing resolution is important for information extraction

Types of coreferences

- Pronoun as subject: “**Cô ta** đang học trực tuyến”
- Pronoun as object: “Hãy liên lạc với **anh ấy** ngay”
- Possessive pronoun: “Lịch trình của **chúng ta** đã được thống nhất”
- “Anh ta tự làm khó **mình**”

Types of coreferences (cont.)

- First name: "**Thủ tướng Nguyễn Xuân Phúc** tuyên bố giãn cách xã hội. **Thủ tướng Phúc** cũng yêu cầu người dân tự giác thực hiện các quy định."
- Apposition: "**Phạm Nhật Vượng**, **Chủ tịch Vingroup** là một trong số các tỉ phú được Forbes nêu tên."
- Verb 'là' : "**Park Hang Seo** là **HLV trưởng đội tuyển bóng đá nam Việt Nam.**"

Types of coreferences (cont.)

- Group people: “**Mây Trắng** tuyên bố tái hợp. **Nhóm** dự định ra mắt album mới đầu năm sau.”
- Attribute - value: “**Giá cổ phiếu VIC** là **94.800 VND**”
- Order: “IBM và **Microsoft** là những ứng cử viên cuối cùng, nhưng đại diện nhà đầu tư ưu tiên **ứng cử viên thứ hai**.”
- Part - whole: “Vinfast mới ra mắt **dòng xe mới**. **Bộ truyền động** sử dụng công nghệ CVT vô cấp tiên tiến.”

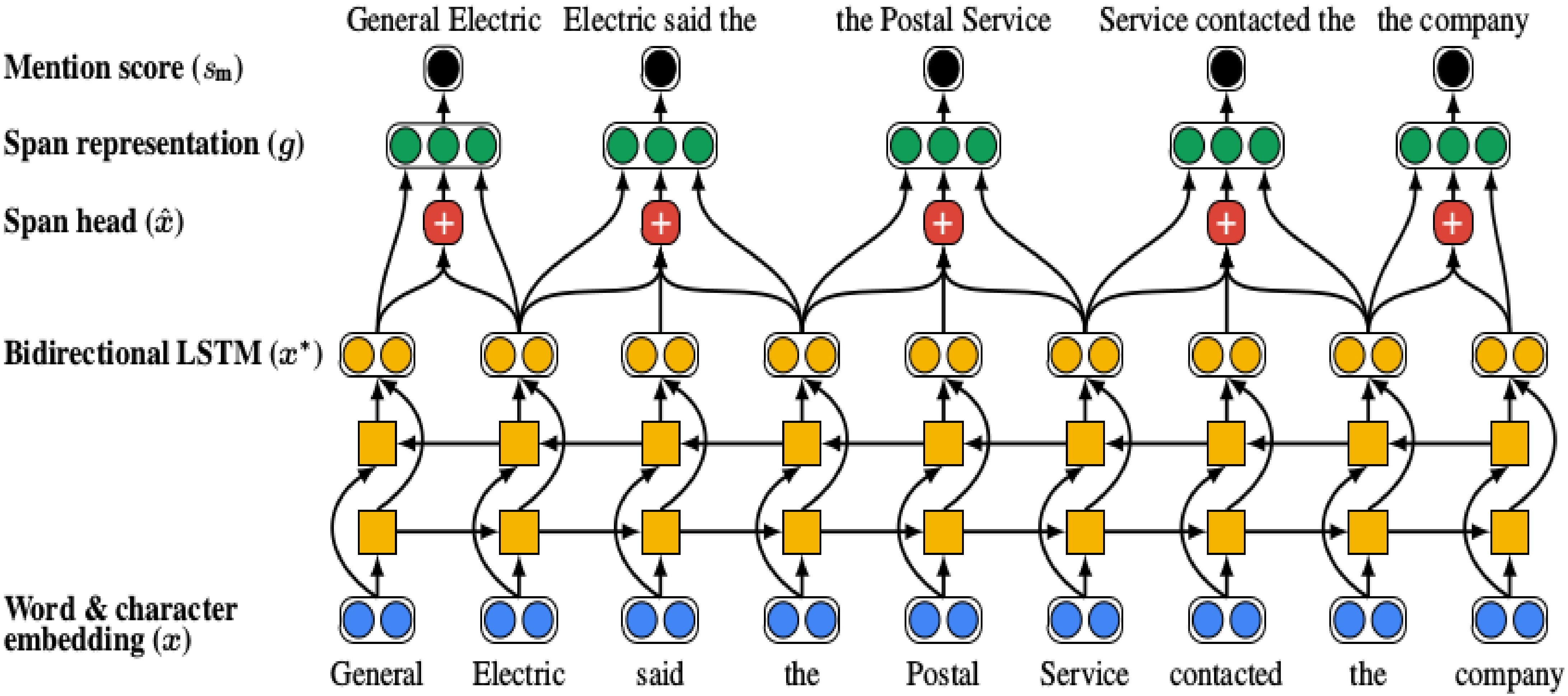
Traditional methods

- Focus on pronouns which are the most common occurrences
- Using linguistic information to spot frontrunners
- Eliminate candidates based on properties such as gender, singular plural, etc.
- Score candidates
  - Matching
  - Rule
  - Machine Learning

Neural network based method

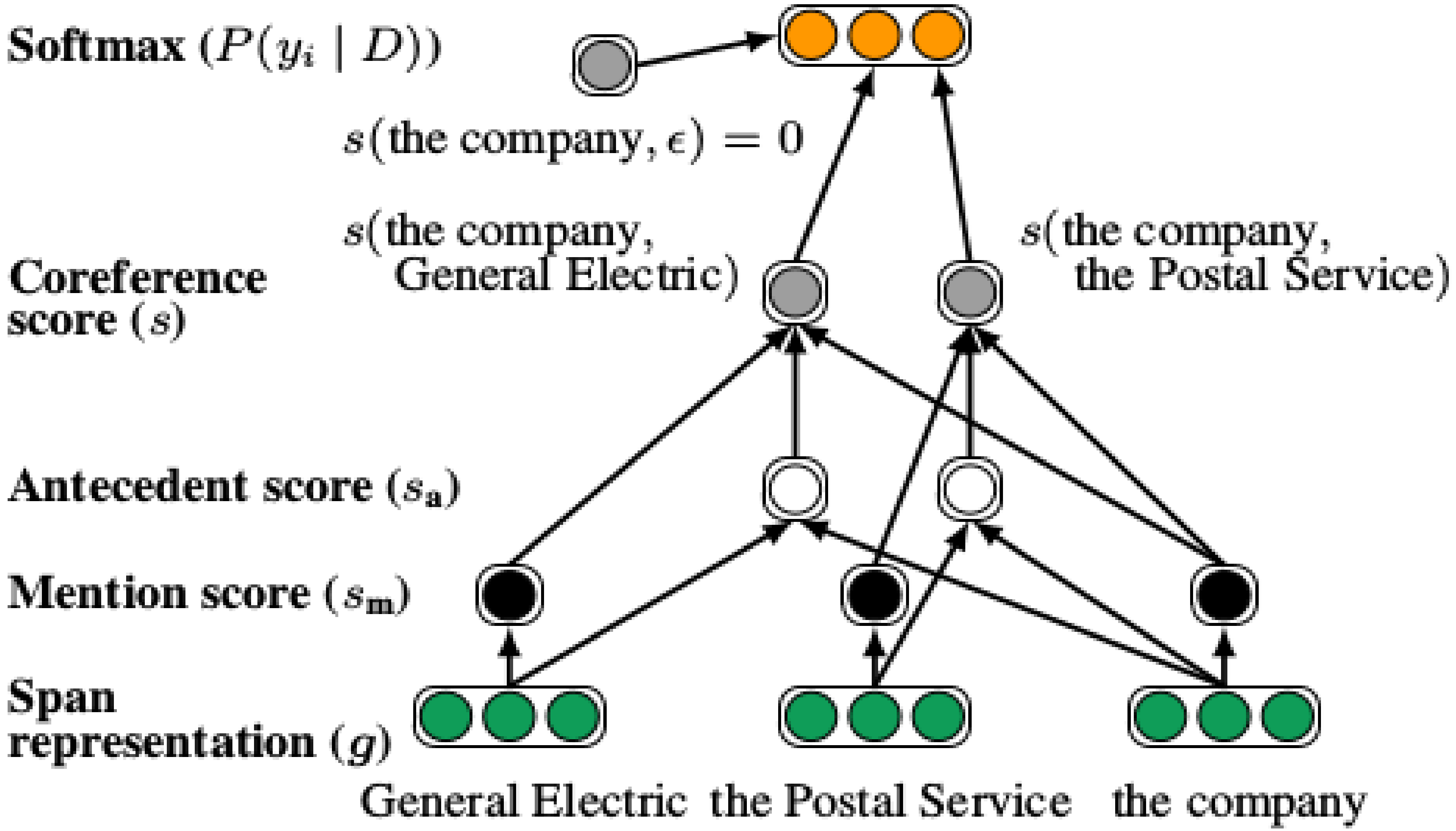
- Limit the use of complex features
- Limit the use of parsers
- Take advantage of pre-trained representation
- Challenge:
  - Use alternative information for syntax information
  - Expressing phrases, contexts
  - Coreferencing resolution is essentially a hard clustering problem within text

Model architecture



Lee et al. “*End-to-end Neural Coreference Resolution*”. EMNLP 2017.

Model architecture (cont.)



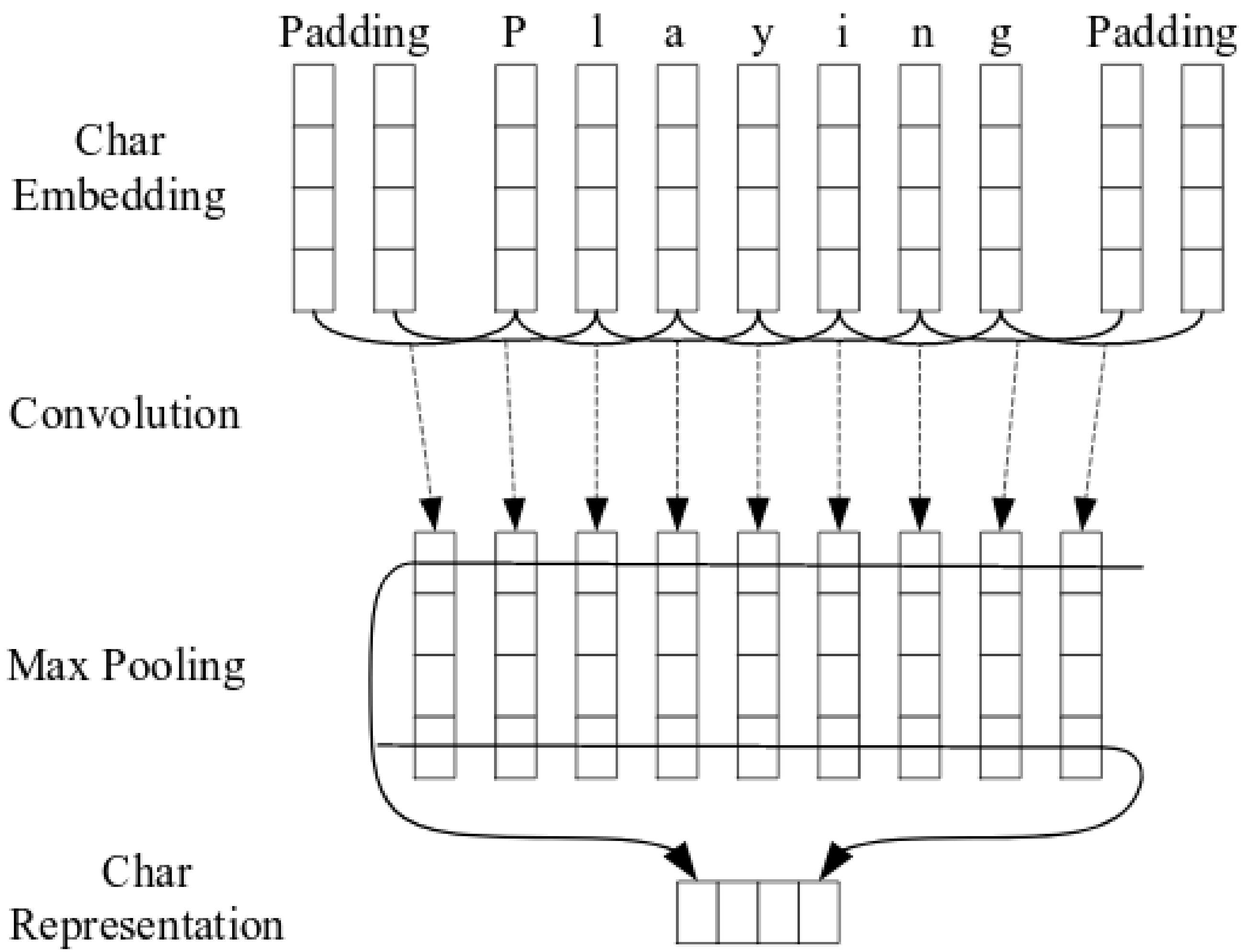
Problem statement

- Document *D* consists of a sequence of words w_1, w2,…, w_T
- D contains $N = T(T+1)/2$ paragraphs with length from 1 to T
- The paragraphs are sorted by the position of the starting word START(i); paragraphs with the same starting word are sorted by the position of the ending word END(i)
- With each paragraph *i*, find paragraph *j* preceding it representing an entity *i* refer to: $j = y_i$
  - if i not refer to any paragraph $y_i = \varepsilon$

Input layer

- Word Embedding:
  - Combined Glove 300 dim and Turian et al. (2010)
  - OOV: Vector 0
- CNN-based character representation:
  - Input character has 8 dimensions
  - Windows {3, 4, 5} character, each with 50 filters

Character representation based on CNN



from Xuezhe Ma and Eduard Hovy. “*End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF*”. ACL 2016

Contextual representation

- The word representation is fed into two LSTM
  - Forward LSTM: Shows dependence of current word on previous words in sentence
  - Backward LSTM: Shows dependence of current word on the following words in sentence
  - The final representation is concatenation of two representations

Span reprentation

- $\mathbf{g}_i = [\boldsymbol{x}^*_{\text{START}(i)}, \boldsymbol{x}^*_{\text{END}(i)}, \tilde{\boldsymbol{x}}_i, \Phi(i)]$
- $\boldsymbol{x}^*_{\text{START}(i)}$: First word representation
- $\boldsymbol{x}^*_{\text{END}(i)}$: Last word representation
- $\tilde{\boldsymbol{x}}_i$: "soft" representation of main word in the span is based on attention mechanism
- $\Phi(i)$: Represents length of i (number of words in i)

Soft representation of main word

$$\alpha_t = \boldsymbol{w}_\alpha \cdot \text{FFNN}_\alpha(\boldsymbol{x}_t^*)$$

$$a_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)}$$

$$\hat{\boldsymbol{x}}_i = \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot \boldsymbol{x}_t$$

- FNNN_α: feed forward neural network learn attention weights
- $\boldsymbol{w}_\alpha$: Link weights of FNNN_α
- α_t: Output of FNNN_α at time t

Scoring mention

- $s_m(i) = \boldsymbol{w}_m \cdot \mathrm{FFNN}_m(\boldsymbol{g}_i)$
- $\boldsymbol{g}_i$: Span *i* representation
- FNNN_m: feed forward neural network score mention
- $\boldsymbol{w}_m$: link weight of FNNN_m

Calculate similarity

- $s_a(i, j) = \boldsymbol{w}_a \cdot \text{FFNN}_a([\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \circ \boldsymbol{g}_j, \Phi(i, j)])$
- FNNN_a: feed forward neural network computes the similarity between two segments *i* and *j*
- $\boldsymbol{w}_a$: link weight of FNNN_a
- $\boldsymbol{g}_i \circ \boldsymbol{g}_j$: inner product
- $\Phi(i, j)$: Represents speaker information and gender, and distance between two spans *i* and *j*

Loss function

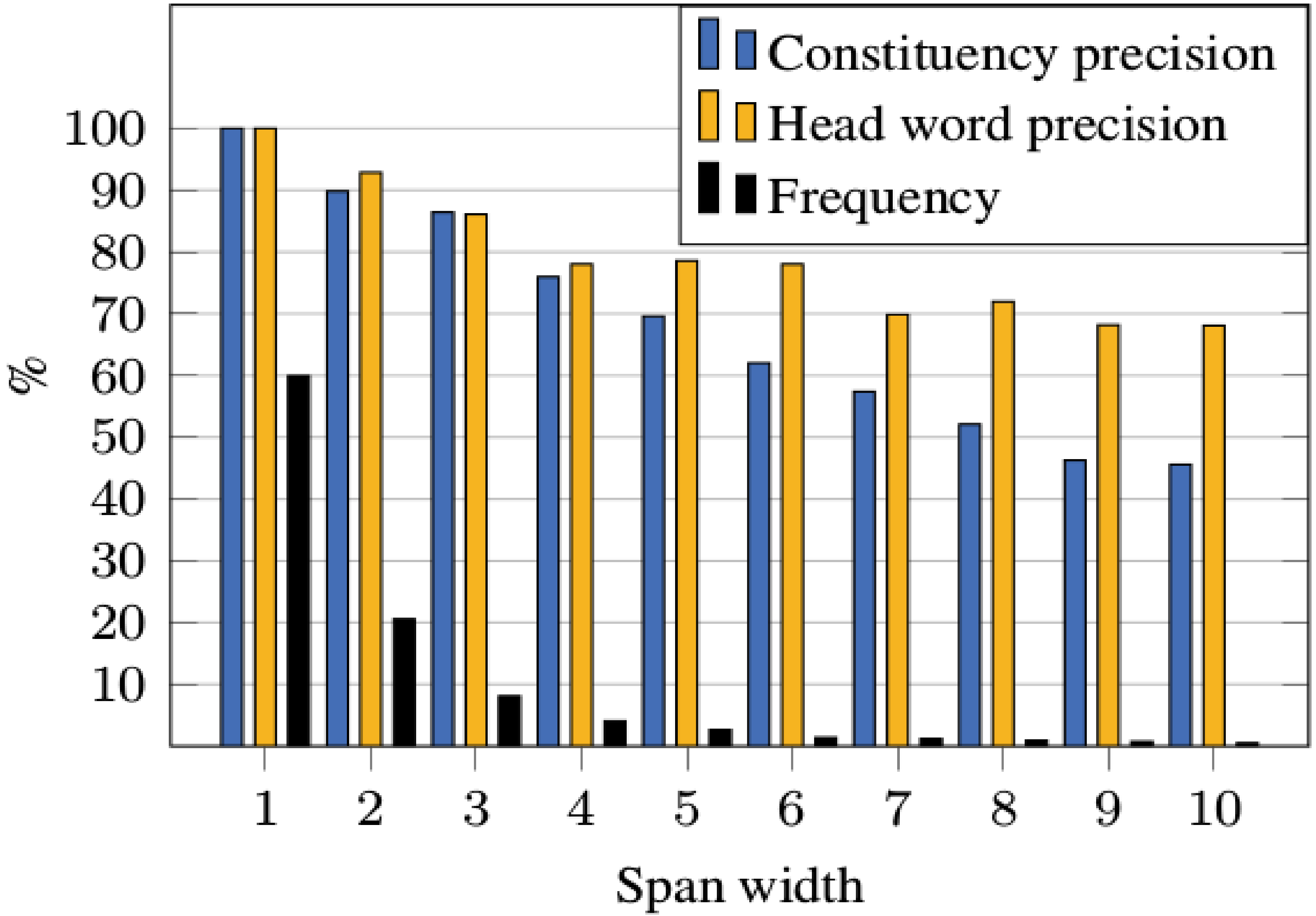
$$
\begin{aligned}
P(y_1, \ldots, y_N \mid D) &= \prod_{i=1}^{N} P(y_i \mid D) \\
&= \prod_{i=1}^{N} \frac{\exp(s(i, y_i))}{\sum_{y' \in \mathcal{Y}(i)} \exp(s(i, y'))}
\end{aligned}
$$

- Marginal probabilities of segments representing entities
- $s(i, y_i)$: Possibility i refer to y_i

Evaluation

	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

Evaluation (cont.)



Evaluation (cont.)

1	(**A fire in a Bangladeshi garment factory**) has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee (**the blaze**) in the four-story building.
	A fire in (**a Bangladeshi garment factory**) has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in (**the four-story building**).
2	We are looking for (**a region of central Italy bordering the Adriatic Sea**). (**The area**) is mostly mountainous and includes Mt. Corno, the highest peak of the Apennines. (**It**) also includes a lot of sheep, good clean-living, healthy sheep, and an Italian entrepreneur has an idea about how to make a little money of them.
3	(**The flight attendants**) have until 6:00 today to ratify labor concessions. (**The pilots'**) union and ground crew did so yesterday.
4	(**Prince Charles and his new wife Camilla**) have jumped across the pond and are touring the United States making (**their**) first stop today in New York. It's Charles' first opportunity to showcase his new wife, but few Americans seem to care. Here's Jeanie Mowth. What a difference two decades make. (**Charles and Diana**) visited a JC Penney's on the prince's last official US tour. Twenty years later here's the prince with his new wife.
5	Also such location devices, (**some ships**) have smoke floats (**they**) can toss out so the man overboard will be able to use smoke signals as a way of trying to, let the rescuer locate (**them**).

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !