



Natural Language Processing with Deep Learning

BÀI 6: SPELLING CORRECTION



https://csc.edu.vn/data-science-machine-learning/natural-language-processing-with-deep-learning_293



GRAMMAR CORRECTION



I. Tổng quan về Spelling Correction

II. Non-word Spelling Errors

III. Real word Spelling Errors

IV. Các vấn đề hay gặp



Tổng quan về Spelling Correction

Spelling Correction

Là quá trình xác định và sửa lỗi ngữ pháp, chính tả trong văn bản tự động bằng các thuật toán Machine Learning.

→ Cải thiện **độ chính xác** và **đồng nhất** của văn bản, giúp truyền đạt thông tin một cách rõ ràng.

New Message

Recipients

Hello from Chris!

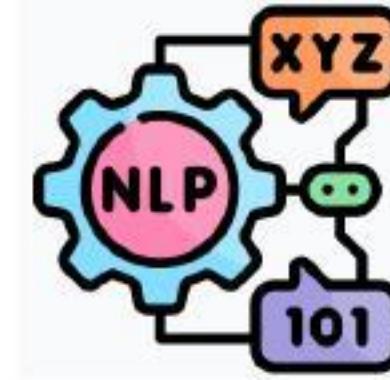
Hi Sam,

It's been a while! I hope you're doing well. I noticed that your company is looking for a new marketing manager, and I've decided to throw my hat in the ring. Would you be willing to put in a good word for me, or maybe

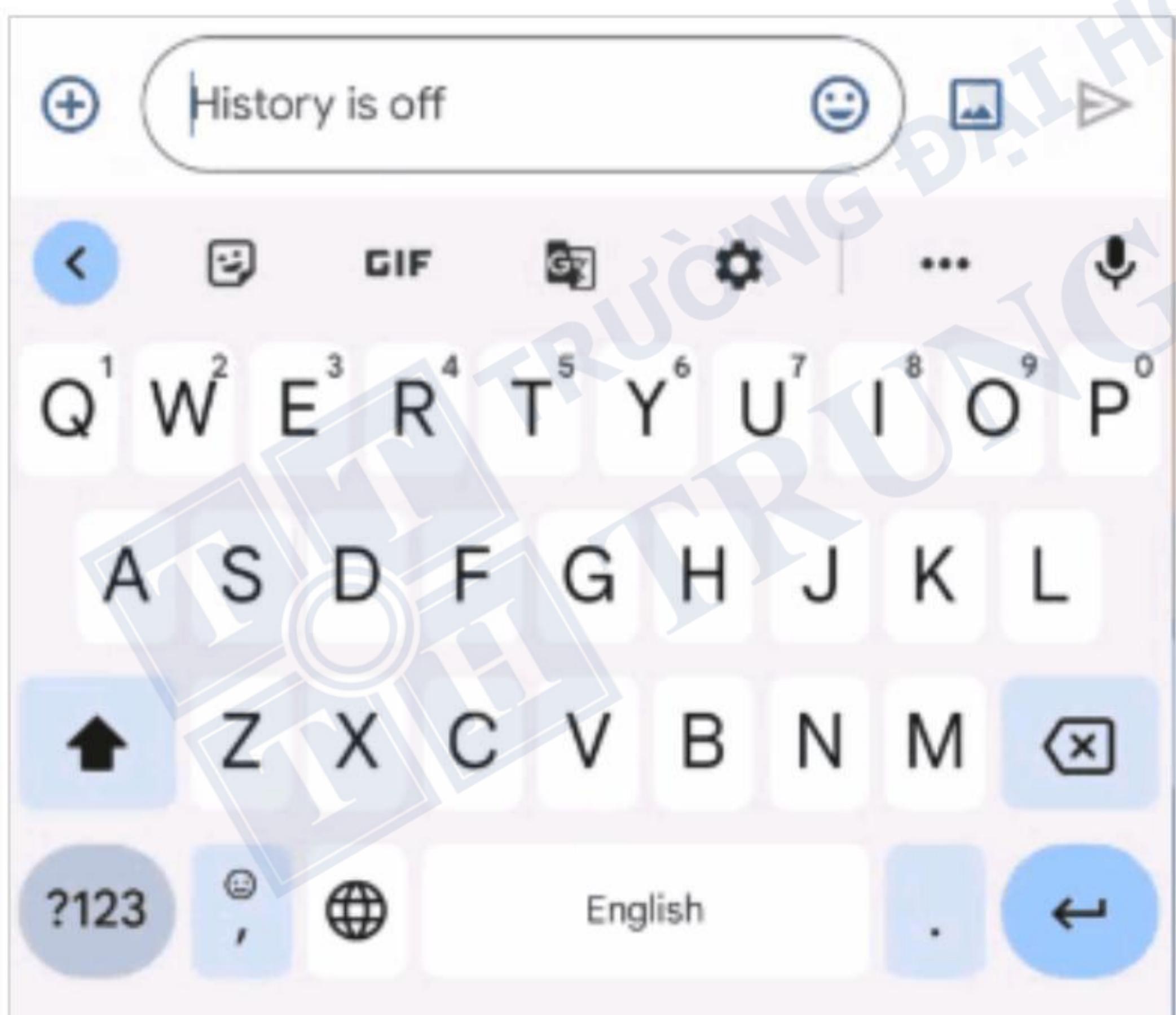
G

Send A U G S L : T :

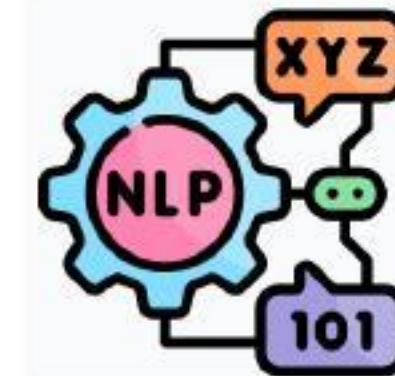
Ứng dụng của Spelling Correction



1. Xử lý từ ngữ
2. Web search



3. Các tác vụ trong thiết bị di động



Các dạng Spelling Errors

I. Non-word errors: Các từ không có trong từ điển

- *Graffe* → *giraffe*

II. Real-word errors: Các từ có trong từ điển

1. Typographical errors: là những lỗi chính tả hoặc sai sót trong việc gõ máy, ví dụ như viết sai chữ cái, đảo ngược thứ tự từ, hay bỏ sót chữ.

- *Three* → *there*

2. Cognitive errors (từ đồng âm): là những lỗi liên quan đến hiểu lầm từ có cách phát âm giống nhau nhưng có nghĩa khác nhau.

- *Piece* → *peace*
- *too* → *two*

SPELLING CORRECTION



I. Tổng quan về Spelling Correction

II. Non-word Spelling Errors

III. Real word Spelling Errors

IV. Các vấn đề hay gặp



Non-Word Spelling Errors

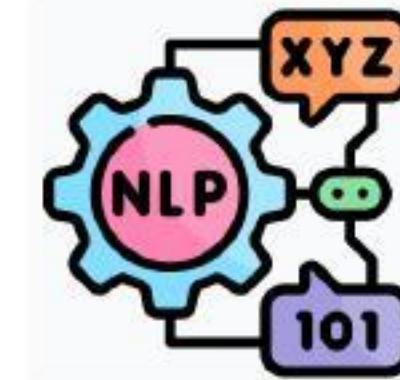
Non-word spelling error detection

- Bất cứ từ nào không có trong **dictionary** là error.
- Dictionary lớn hơn cho hiệu quả tốt hơn.

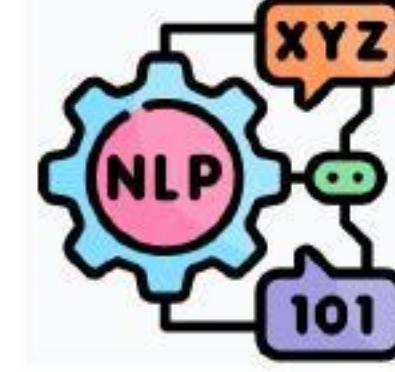
Non-word spelling error correction

- Từ thay thế - **Candidate generation**
Gợi ý các từ thực có cấu trúc tương tự với từ lỗi.
- Chọn từ thay thế phù hợp nhất bằng:
 - **Shortest weighted edit distance**
 - **Highest noisy channel probability**

Candidate Generation



- Các từ có spelling tương tự.
- Ví dụ: user gõ 7outub
 - ➔ Từ nào gần nhất: *youtube*, *yourtube*, *youthumb*?
 - ➔ Sử dụng **edit distance** để tính **d(*youtube*, 7outub)**



Minimum Edit Distance

- Tên khác là Damerau-Levenshtein Edit Distance
 - Là một phép đo **khoảng cách giữa hai chuỗi ký tự**.
 - Đếm số lượng các thao tác chỉnh sửa cần thiết để biến đổi một chuỗi thành chuỗi khác:
 - *Insertion*
 - *Deletion*
 - *Substitution*
 - *transposition*
- Đây là một kỹ thuật quan trọng trong xử lý ngôn ngữ tự nhiên, ứng dụng trong kiểm tra chính tả, gợi ý từ và nhận dạng giọng nói.



Cách tính Minimum Edit Distance

Có 2 cách:

1. Naïve search → Chi phí cao.
2. Dynamic Programming.

1. Cho 2 chuỗi X và Y:

- X có độ dài là n với $1 \leq i \leq n$
- Y có độ dài là m với $1 \leq j \leq m$

2. Tính khoảng cách $D(i,j)$:

- Edit distance giữa $X[1, \dots, i]$ và $Y[1, \dots, j]$

VD: *Edit distance giữa ký tự đầu tiên của X và Y ($i, j = 1$).*

- Edit distance giữa X và Y → $D(n,m)$



Cách tính Minimum Edit Distance

Tính bảng
 $D(n,m)$

- **Initialize:** tính $D(i,j)$ cho các giá trị i, j nhỏ.
- **Iterate:** tính các $D(i,j)$ lớn hơn dựa trên các giá trị nhỏ hơn đã tính.

→ Initialization:

$$D(3, 0) = 3$$

intention



execution

for all i

$$D(i, 0) = i$$

$$D(0, 5) = 5$$

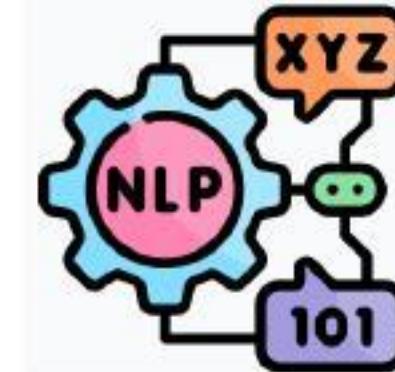
intention



execution

for all j

$$D(0, j) = j$$



Cách tính Minimum Edit Distance

→ Recurrence Relation (iteration):

```
for each i = 1...m  
    for each j = 1...n
```

Ta có:

$$D(i, j) = \min \left\{ \begin{array}{l} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } x(i) \neq y(j) \\ 0 & \text{if } x(i) = y(j) \end{cases} \end{array} \right.$$



Cách tính Minimum Edit Distance

→ Recurrence Relation (iteration):

- Giá trị nhỏ nhất của:

- $D(\text{int}, \text{exec}) = \text{del}[t] + D(\text{in}, \text{exec})$
- $D(\text{int}, \text{exec}) = D(\text{int}, \text{exe}) + \text{ins}[c]$
- $D(\text{int}, \text{exec}) = \text{substitute}[t, c] + D(\text{in}, \text{exe})$

Tính Recurrence relation:

```
for each i = 1...m  
    for each j = 1...n
```

$$D(i, j) = \min \left\{ \begin{array}{l} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } x(i) \neq y(j) \\ 0 & \text{if } x(i) = y(j) \end{cases} \end{array} \right.$$



Cách tính Minimum Edit Distance

→ Initialization:

for all i , $D(i, 0) = i$

for all j , $D(0, j) = j$

→ Recurrence Relation (iteration):

for each $i = 1 \dots m$

 for each $j = 1 \dots n$

$$D(i, j) = \min \left\{ \begin{array}{l} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } X(i) \neq Y(j) \\ 0 & \text{if } X(i) = Y(j) \end{cases} \end{array} \right.$$

→ Termination:

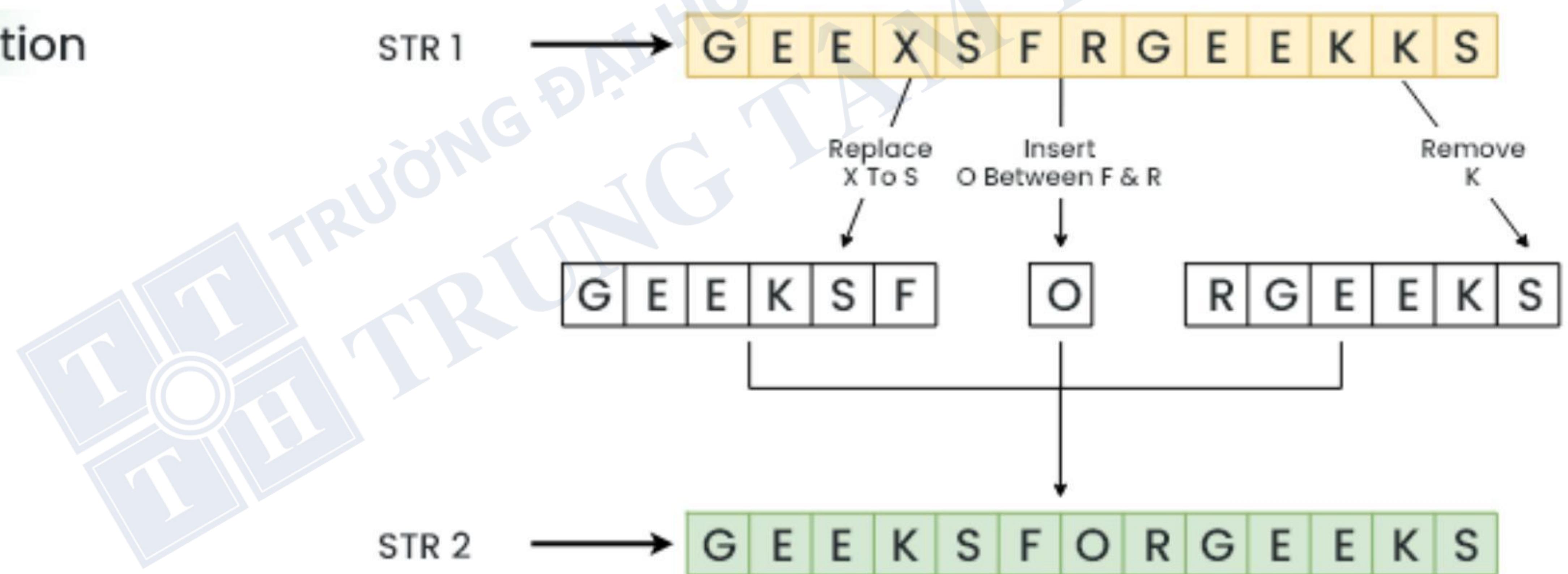
$D(n, m)$ là edit distance

Ví dụ về Minimum Edit Distance

Example



Solution



Minimum Number Of Edits To Convert Str1 To Str2 = 3



Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } X(i) \neq Y(j) \\ 0 & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1	2								
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } X(i) \neq Y(j) \\ 0 & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1	2								
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } X(i) \neq Y(j) \\ 0 & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

↗



Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1	2	3							
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } X(i) \neq Y(j) \\ 0 & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

Ví dụ Minimum Edit Distance

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2	3								
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

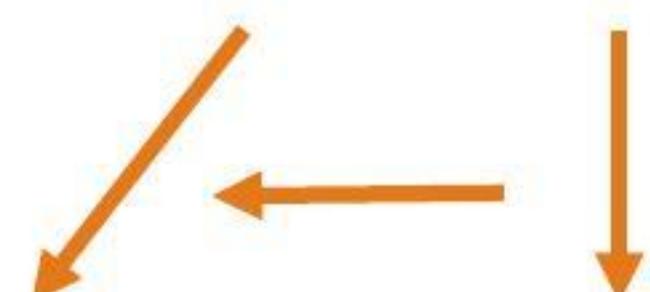
$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 & \text{if } X(i) \neq Y(j) \\ 0 & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

Ví dụ Minimum Edit Distance

N	9	8	9	10	11	12	11	10	9	8
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
T	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Ví dụ Minimum Edit Distance

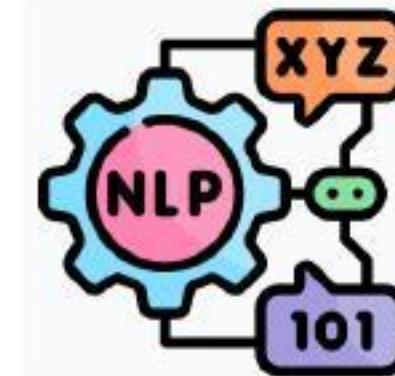
N	9	8	9	10	11	12	11	10	9	8
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
T	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N





Weighted Edit Distance

- Các thao tác chỉnh sửa (thêm, xóa, thay thế hoặc hoán đổi) được gán một trọng số w khác nhau.
- Thông thường, w được xác định bởi tần suất xuất hiện của error trong training set.
 - Một số từ có xác suất bị nhập sai cao hơn → **w nhỏ hơn.**
 - Để tính trọng số cho các loại thao tác chỉnh sửa, thống kê tần suất xuất hiện của các loại error.



Weighted Edit Distance

Cụ thể, ta có thể thu thập thông tin về:

Thao tác	Cách thống kê	Ví dụ
Substitution sub[x,y]	Đếm số lần một chữ cái được thay thế bằng một chữ cái khác	$\text{Cat} \rightarrow \text{Bat}$
deletion del[x,y]	Đếm số lần một chữ cái bị xóa khỏi chuỗi.	$\text{Cat} \rightarrow \text{Ct}$
Insertion ins[x,y]	Đếm số lần một chữ cái được thêm vào chuỗi.	$\text{Cat} \rightarrow \text{Cart}$
Transposition trans[x,y]	Đếm số lần hai chữ cái được hoán đổi vị trí cho nhau.	$\text{Cat} \rightarrow \text{Cta}$

- Chữ cái bị gõ sai nhiều hơn có w thấp hơn, các chữ cái ít bị sai sẽ có w cao hơn.



Cách tính Weighted Edit Distance

→ Initialization:

$$D(0,0) = 0$$

$$D(i,0) = D(i-1,0) + \text{del}[x(i)] \quad 1 < i \leq n$$

$$D(0,j) = D(0,j-1) + \text{ins}[y(j)] \quad 1 < j \leq m$$

→ Recurrence Relation (iteration):

$$D(i-1,j) + \text{del}[x(i)]$$

$$D(i,j) = \min D(i,j-1) + \text{ins}[y(j)]$$

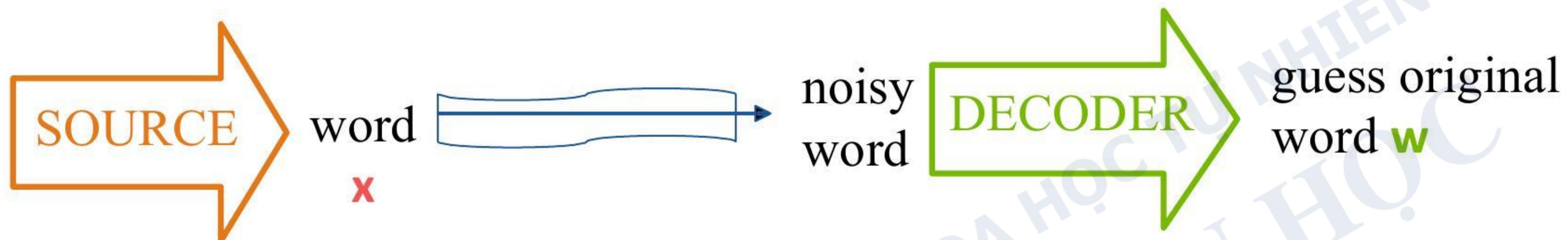
$$D(i-1,j-1) + \text{sub}[x(i), y(j)]$$

→ Termination:

$D(n,m)$ là edit distance



Noisy Channel Model



- Mô hình giả định rằng có noise giữa người gửi và người nhận thông điệp, dẫn đến sự biến đổi hoặc lỗi trong thông điệp gốc.
- Sử dụng **transformer decoder** tìm ra từ đúng qua quan sát từ sai, tạo ra các từ gần giống từ sai, so sánh và đánh giá tính hợp lý của từ đúng trong ngữ cảnh.



Noisy Channel Model



Có quan sát x là một dạng lỗi sai của w

→ Decode từ đúng (w):

$$\hat{w} = \operatorname{argmax}_{w \in V} P(w | x)$$

prob of w given
misspelled x

best guess

search over
vocabulary V



Noisy Channel Model

Công thức
được viết
lại như sau:

$$\hat{w} = \operatorname{argmax}_{w \in V} P(w | x)$$

$$= \operatorname{argmax}_{w \in V} \frac{P(x | w)P(w)}{P(x)}$$

$$= \operatorname{argmax}_{w \in V} P(x | w) \underbrace{P(w)}_{\text{noisy language channel model}} \underbrace{P(w)}_{\text{model}}$$



Language Model $P(w)$

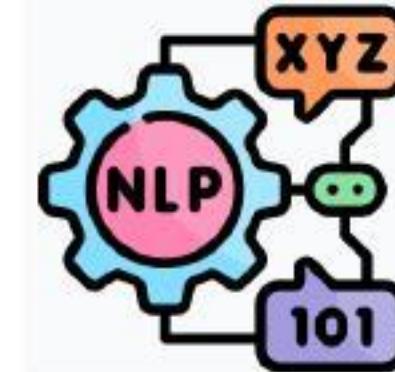
Sử dụng các thuật toán Language Modeling:

- Nếu w là 1 từ → **Unigram** model.
- Nếu w có ngũ cảnh (nhiều từ)
→ **Bigram** và **Trigram** models.

VD: Cho câu *Like ther books*

- Non-word: *ther*
- Cả *there* và *their* đều có edit distance với *ther* là 1.
- Cả $P(\text{their})$ và $P(\text{there})$ đều có giá trị lớn.
- Thêm ngũ cảnh:
 $P(\text{like there books})$ và $P(\text{like their books})$

- Dùng smoothing nếu cần:
 - Add-Delta, Good-Turing, ...



Noisy Channel Model $P(x|w)$

Còn được gọi là **Error Probability Model**.

- Giống như language model, được model từ **marked training data**.
- Cần có nội dung chứa **marked mistakes/ corrections**. Trong đó:
 - Misspelled word x có các ký tự x_1, x_2, \dots, x_m
 - Dictionary word w có các ký tự w_1, w_2, \dots, w_n
- Giả sử mỗi từ chỉ có 1 lỗi:
Tìm từ x qua w chỉ với 1 thao tác biến đổi từ.
 $\rightarrow P(x|w)$ là xác suất của lỗi này.



Noisy Channel Model $P(x|w)$

Giả sử có thể từ x qua w với 1 error:

$P(x|w) =$

$$\frac{\text{del}[w_{i-1}, w_1]}{\text{count}[w_{i-1}, w_1]}$$
$$\frac{\text{ins}[w_{i-1}, w_1]}{\text{count}[w_{i-1}, w_1]}$$
$$\frac{\text{sub}[w_{i-1}, w_1]}{\text{count}[w_{i-1}, w_1]}$$
$$\frac{\text{trans}[w_{i-1}, w_1]}{\text{count}[w_{i-1}, w_1]}$$

if deletion

if insertion

if substitution

if transposition



Ví dụ về Non-word Spelling Error

Các từ trong 1 edit distance của từ **ACRESS**

Error	Candidate Correction	Correct Letter	Error Letter	Type
acress	actress	t	-	Deletion
acress	cress	-	a	Insertion
acress	caress	ca	ac	Transposition
acress	access	c	r	Substitution
acress	across	o	e	Substitution
acress	acres	-	s	Insertion
acress	acres	-	s	Insertion



Ví dụ về Non-word Spelling Error

- 80% errors nằm trong edit distance 1
 - Hầu hết tất cả errors nằm trong edit distance 2
- Cần chèn dấu cách (space) và dấu gạch ngang (hyphen)
- Thisdish* → *This dish* *Inlaw* → *in-law*

Unigram Probability - ACRESS

Word	Frequency of word	P(word)
actress	9,321	.0000230573
cress	220	.0000005442
caress	686	.0000016969
access	37,038	.0000916207
across	120,844	.0002989314
acres	12,874	.0000318463



Ví dụ về Non-word Spelling Error

Noisy Channel Model - ACRESS

Candidate Correction	Correct Letter	Error Letter	$x w$	$P(x word)$
actress	t	-	c ct	.000117
cress	-	a	a #	.00000144
caress	ca	ac	ac ca	.00000164
access	c	r	r c	.000000209
across	o	e	e o	.0000093
acres	-	s	es e	.0000321
acres	-	s	ss s	.0000342



Ví dụ về Non-word Spelling Error

Noisy Channel Probability của ACRESS

Candidate Correction	Correct Letter	Error Letter	x w	P(x word)	P(word)	$10^9 * P(x w)P(w)$
actress	t	-	c ct	.000117	.0000231	2.7
cress	-	a	a #	.00000144	.000000544	.00078
caress	ca	ac	ac ca	.00000164	.00000170	.0028
access	c	r	r c	.000000209	.0000916	.019
across	o	e	e o	.0000093	.000299	2.8
acres	-	s	es e	.0000321	.0000318	1.0
acres	-	s	ss s	.0000342	.0000318	1.0

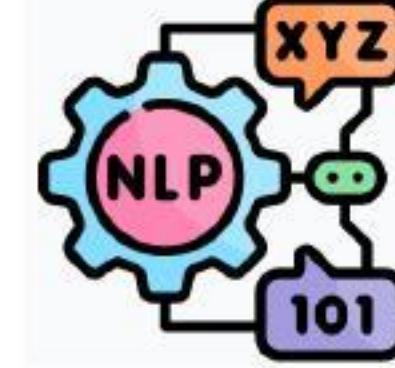


Ví dụ về Non-word Spelling Error

Kết hợp $P(w)$ và $P(x|w)$ - ACRESS

Candidate Correction	Correct Letter	Error Letter	x w	$P(x \text{word})$	$P(\text{word})$	$10^9 * P(x w)P(w)$
actress	t	-	c ct	.0000117	.0000231	2.7
cress	-	a	a #	.000000144	.000000544	.00078
caress	ca	ac	ac ca	.000000164	.00000170	.0028
access	c	r	r c	.0000000209	.0000916	.019
across	o	e	e o	.0000093	.000299	2.8
acres	-	s	es e	.0000321	.0000318	1.0
acres	-	s	ss s	.0000342	.0000318	1.0

→ Xác suất cao nhất là chữ across bằng substitution.



Ví dụ về Non-word Spelling Error

Thêm ngũ cành - ACRESS

"a stellar and versatile acress whose combination of sass and glamour"

→ Tăng hiệu quả:

- Sử dụng bigram model.
- Tìm $P(w)$ bằng CCAE với add-1 smoothing

$$P(\text{actress} \mid \text{versatile}) = .000021$$

$$P(\text{whose} \mid \text{actress}) = .0010$$

$$P(\text{versatile actress whose}) = .000021 * .0010 = 210 \times 10^{-10}$$

$$P(\text{across} \mid \text{versatile}) = .000021$$

$$P(\text{whose} \mid \text{across}) = .000006$$

$$P(\text{versatile across whose}) = .000021 * .000006 = 1 \times 10^{-10}$$



Ví dụ về Non-word Spelling Error

Bigram Language Model - ACRESS

- Tiếp theo, nhân với noisy channel probabilities

$$P(\text{acress} \mid \text{actress}) = .000117$$

$$P(\text{acress} \mid \text{across}) = .0000093$$

→ Tính $P(w).P(x|w)$:

$$\begin{aligned} P(\text{versatile actress whose})P(\text{acress} \mid \text{actress}) &= .000117 \times 210 \times 10^{-10} \\ &= .002457 \times 10^{-10} \end{aligned}$$

$$\begin{aligned} P(\text{versatile across whose})P(\text{acress} \mid \text{across}) &= .0000093 \times 1 \times 10^{-10} \\ &= .0000093 \times 10^{-10} \end{aligned}$$

SPELLING CORRECTION



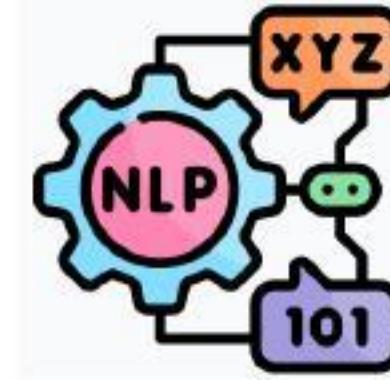
I. Tổng quan về Spelling Correction

II. Non-word Spelling Errors

III. Real word Spelling Errors

IV. Các vấn đề hay gặp phải

Real Word Spelling Errors



Following difficult trading in 2017 and 2018, Leaf decided to close eight of its poor performing stores and reduced its product line range by 20 percent

- ## ● Possible confused word

→ its

 Ignore

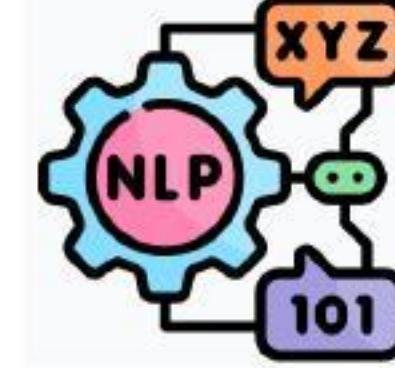
Having less stores

g its net earnings by



- Spelling errors là các từ có trong từ điển
- Cần sử dụng **ngữ cảnh**





Real Word Spelling Errors

1. Với mỗi từ w trong câu:

- Tạo ra candidate set gồm:
 - Từ w.
 - Tất cả single-letter edits là từ tiếng Anh.
 - Các từ đồng âm.

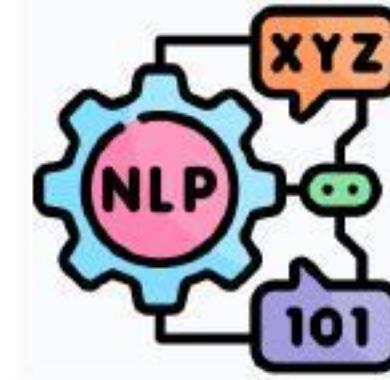
2. Chọn word combination candidates tốt nhất:

- *Noisy channel model*
- *Task-specific classifier*

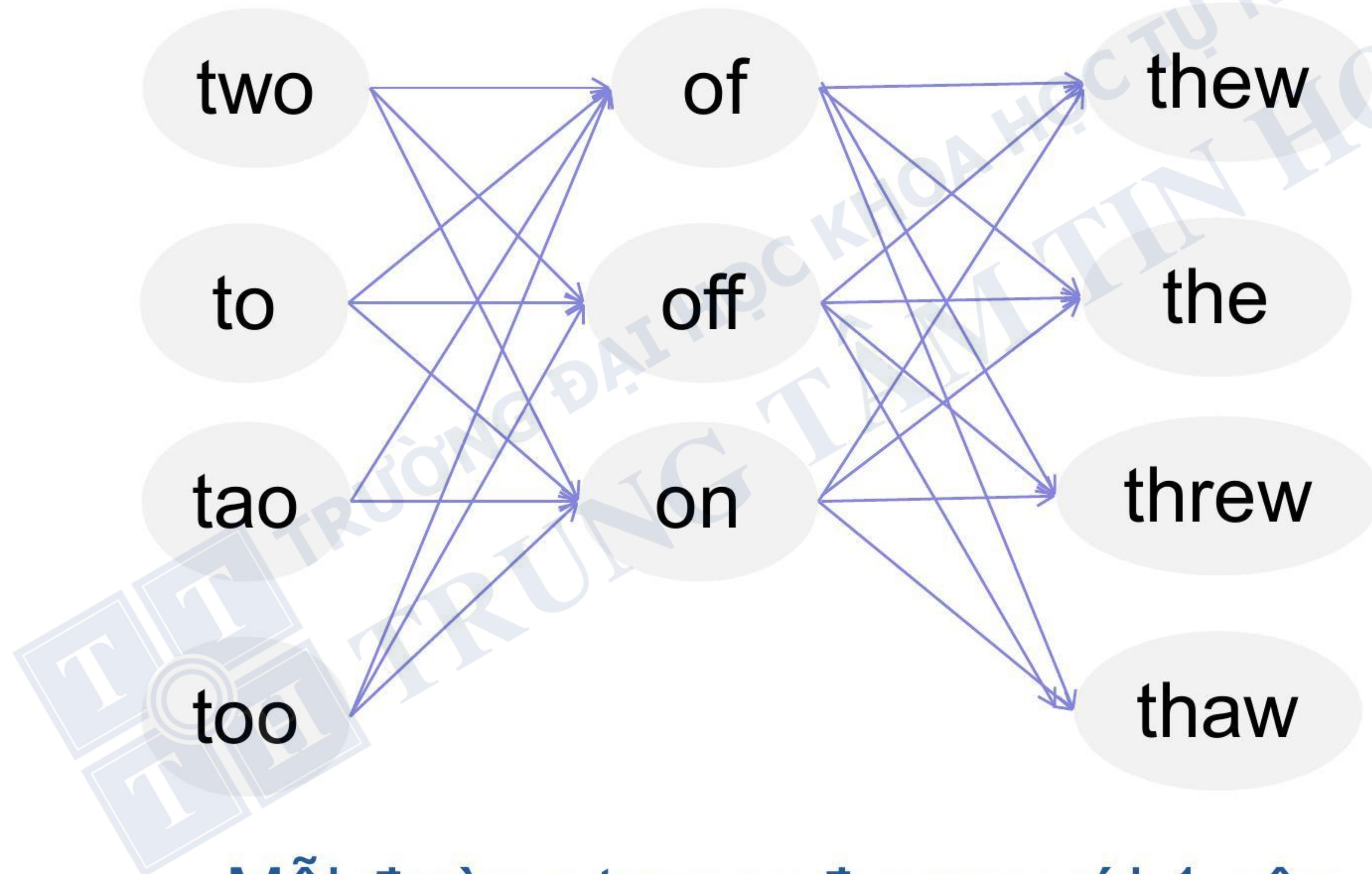
VD: Cho cụm từ “two of thew”

- Candidate (two) = {two, to, tao, too}
- Candidate (of) = {of, off, on}
- Candidate (thew) = {thew, the, threw, thaw}

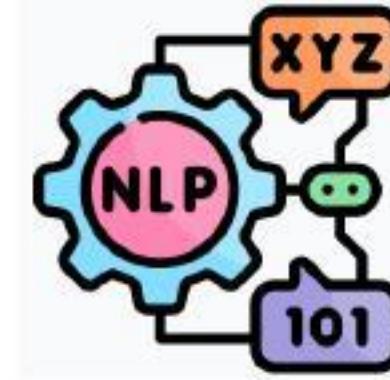
Real Word Spelling Errors



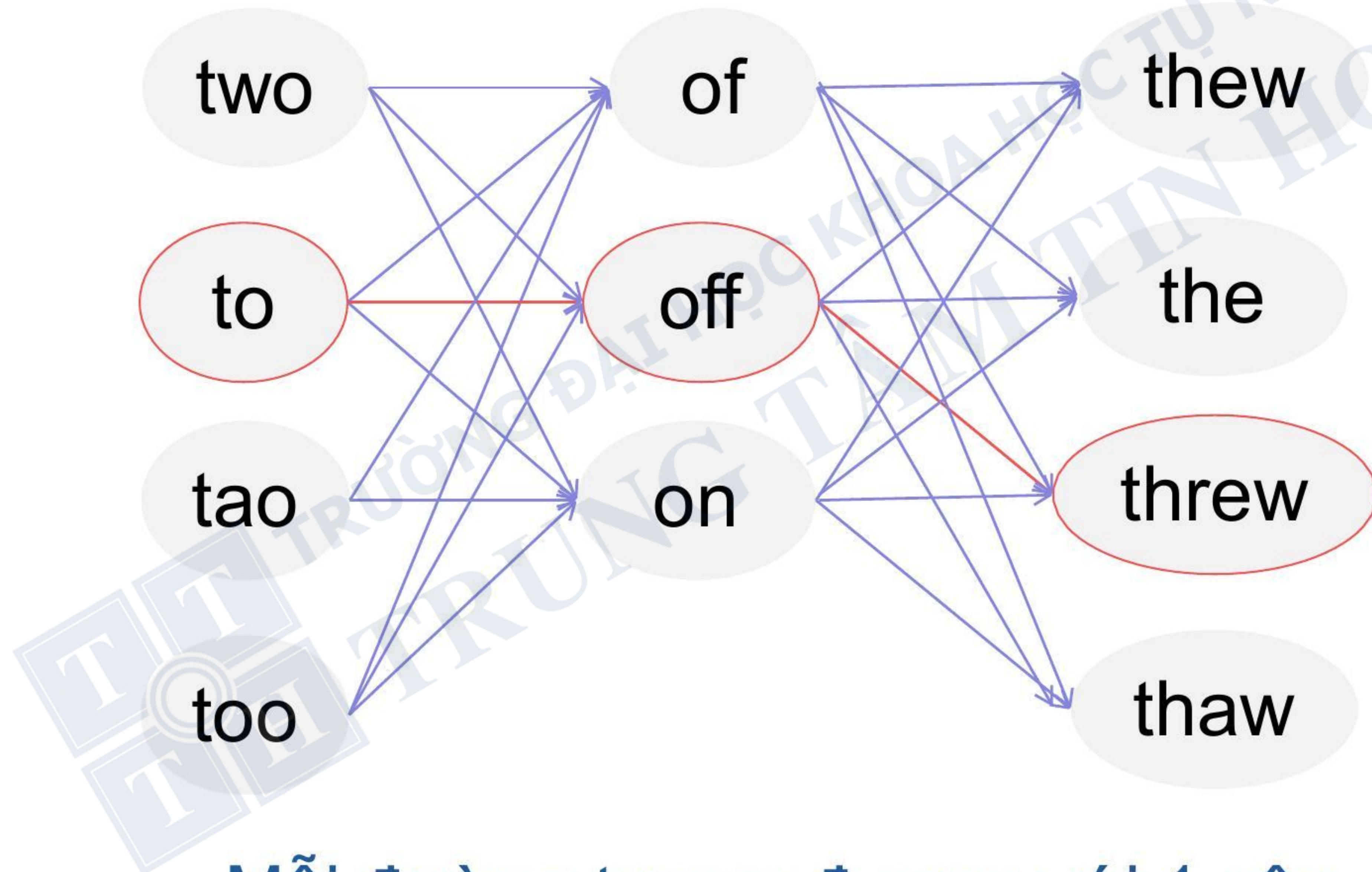
Possible Candidate Combinations



Real Word Spelling Errors

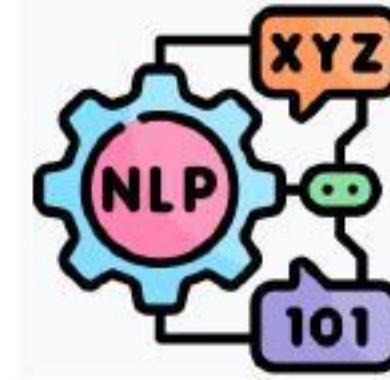


Possible Candidate Combinations

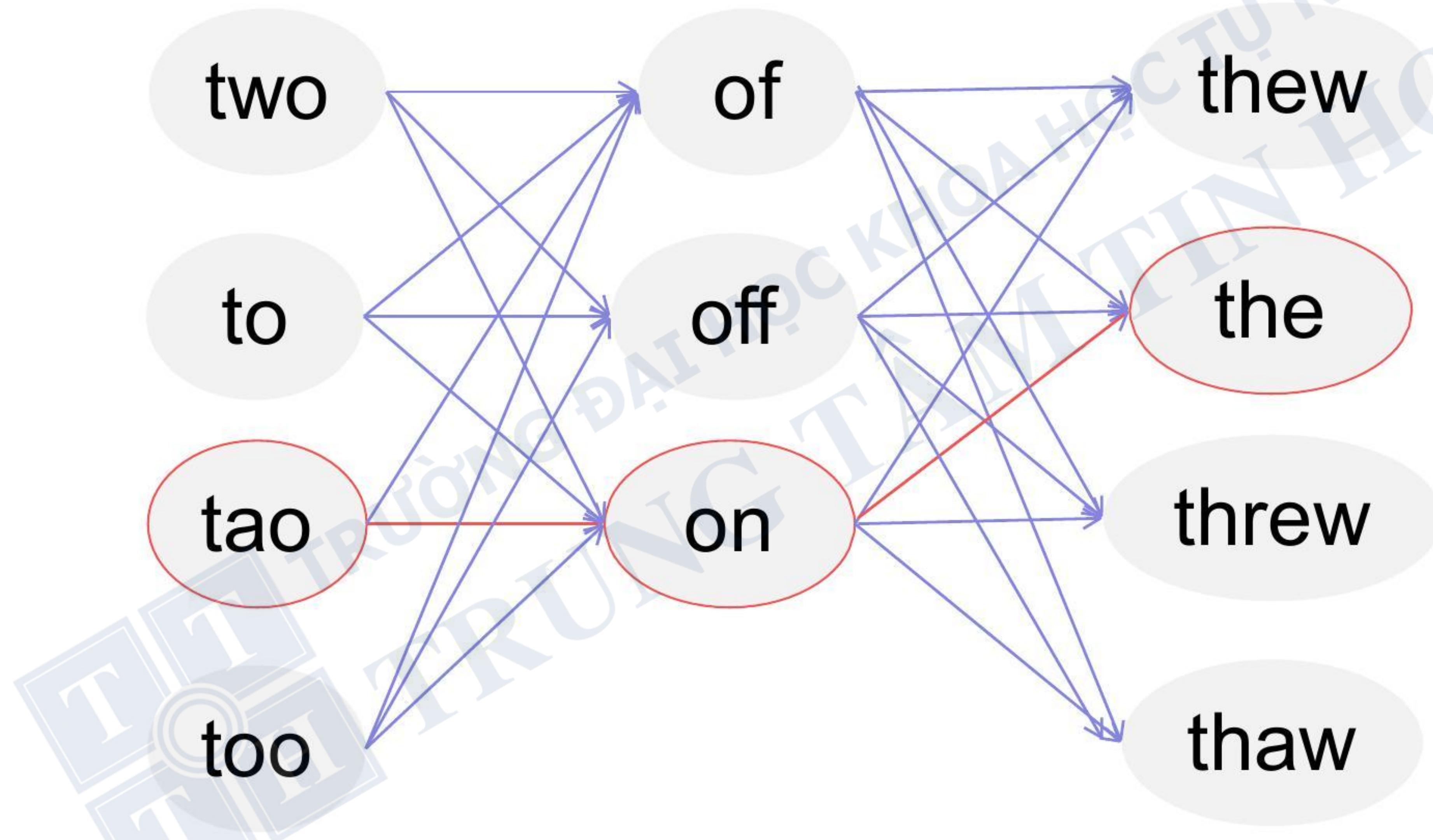


Mỗi đường tương đương với 1 câu.

Real Word Spelling Errors



Possible Candidate Combinations

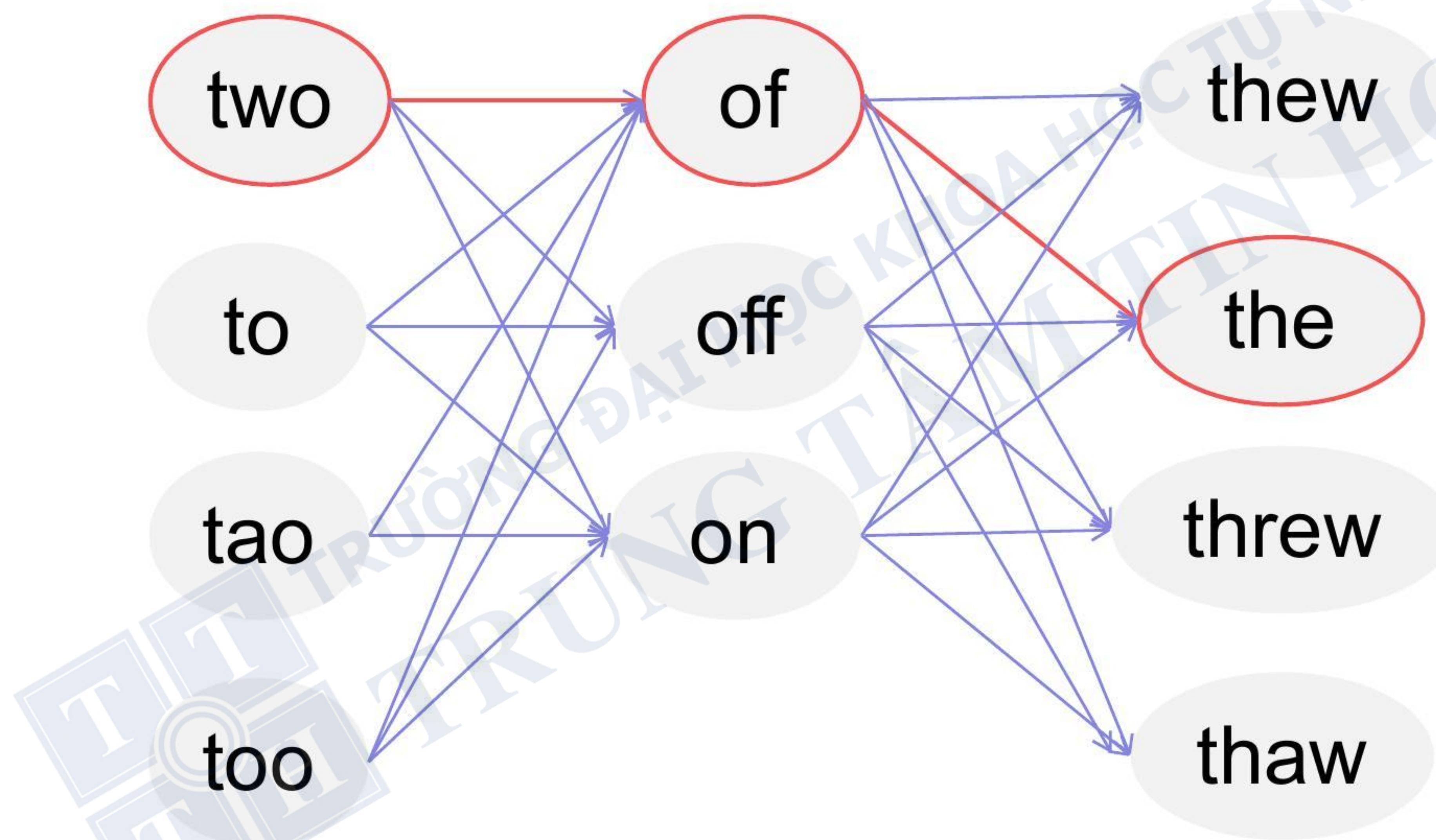


1. Tính xác suất của mỗi đường (câu).
2. Chọn câu có xác suất cao nhất.

Real Word Spelling Errors



Possible Candidate Combinations



1. Tính xác suất của mỗi đường (câu).
2. **Chọn câu có xác suất cao nhất.**



Các bước tính

→ Language Model $P(w)$:

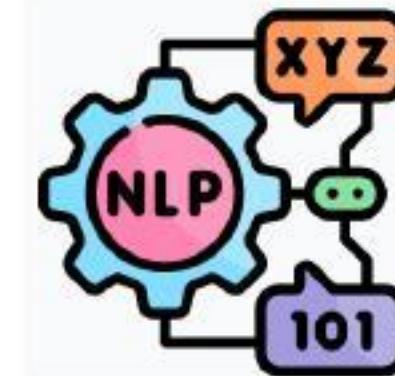
Bigram hoặc **Trigram** model, không dùng Unigram
(không có ngũ cảnh).

→ Noisy Channel Model $P(w|x)$: (như non-word)

$$\hat{w} = \operatorname{argmax}_{w \in V} P(w | x)$$

Chú ý: Có xét xác suất không có error $P(w|w)$

→ Chọn câu có $P(w|x)P(w)$ lớn nhất



Ví dụ về Real-word Spelling Error

“Two of thew”

x	w	x w	P(x w)	P(w)	$10^9 P(x w)P(w)$
thew	the	ew e	0.000007	0.02	144
thew	thew		0.95	0.00000009	90
thew	thaw	e a	0.001	0.0000007	0.7
thew	threw	h hr	0.000008	0.000004	0.03
thew	thwe	ew we	0.000003	0.00000004	0.0001

SPELLING CORRECTION



I. Tổng quan về Spelling Correction

II. Non-word Spelling Errors

III. Real word Spelling Errors

IV. Các vấn đề hay gặp



Human-Computer Interaction Issues

HCI issues là các vấn đề phát sinh khi người dùng tương tác với hệ thống máy tính có liên quan tới **spelling**.

1. **Autocorrection**: sửa từ đúng thành sai
2. **Gợi ý từ không rõ ràng**: gợi ý từ không phù hợp
3. **Thiếu tính cá nhân hóa**: từ chuyên ngành
4. **Nhầm từ đồng âm**
5. **Sử dụng đa ngôn ngữ**: xử lý nhiều ngôn ngữ một lúc



Vấn đề thực tế

1. Never just multiply the prior and the error model
2. Independence assumptions → probabilities not commensurate
3. Instead: add weights (thêm trọng số)

$$\hat{w} = \operatorname{argmax}_{w \in V} P(x | w) P(w)^\lambda$$

Học λ bằng một validation set

Real-word Spelling Correction Classifier



1. Thay vì chỉ dùng Noisy Channel Model và Language Model.
2. Sử dụng thêm các **feature** khác tạo thành 1 **classifier**.
3. Tạo classifier cho 1 cặp từ xác định.

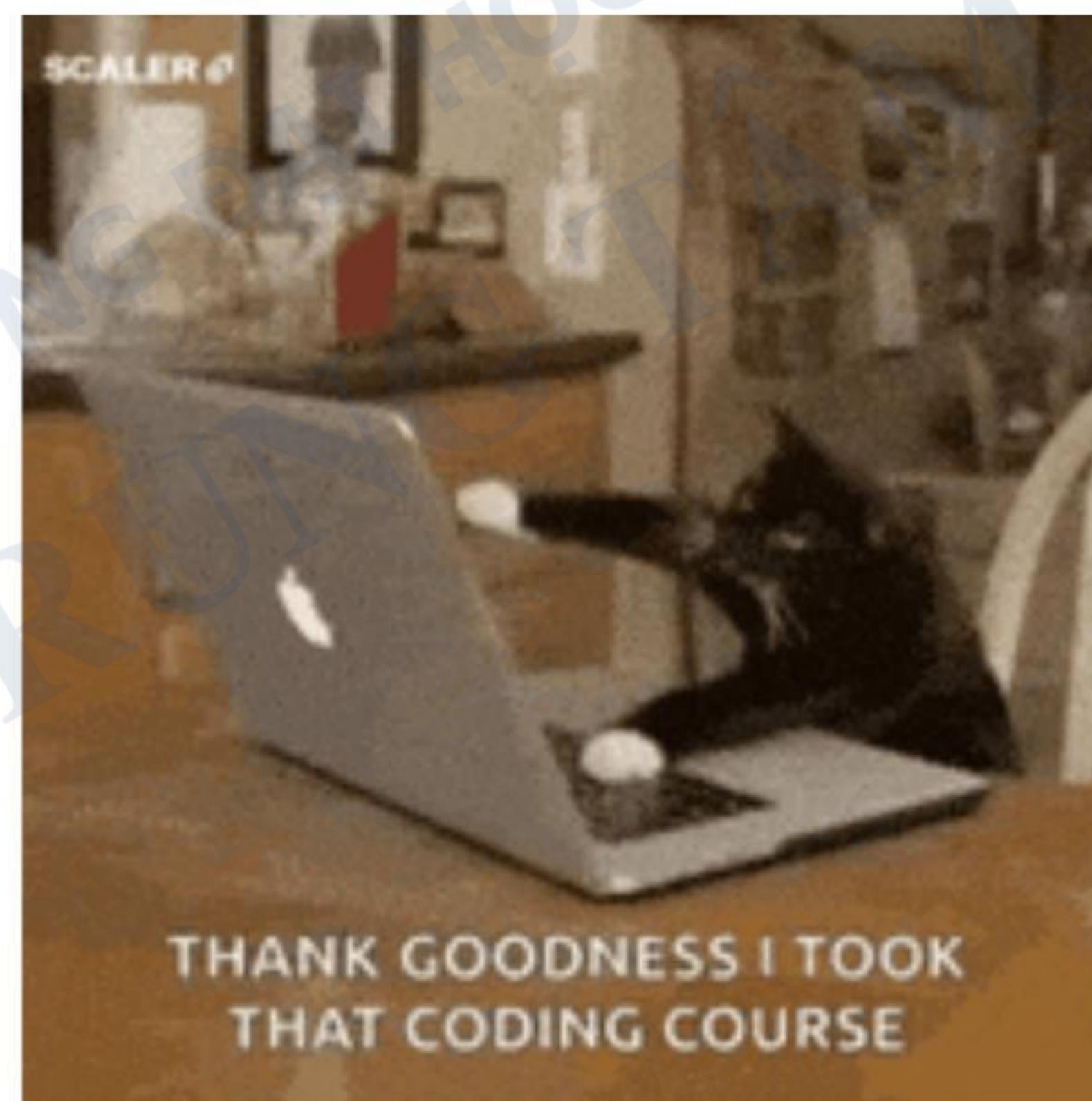
VD: Có cặp từ **whether / weather**

• Có từ “*cloudy*” trong phạm vi ± 10 từ.
_____ to V_o
_____ or not

Code Demo



DEMO



Q&A

