CHULA ENGINEERING
Foundation toward Innovation
COMPUTER

Tokenization

2110594: Natural Language Processing (NLP)

Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University

Based on Aj.Ekapol's slide in 2017

Outlines

The need for segmentation

Thai Tokenization

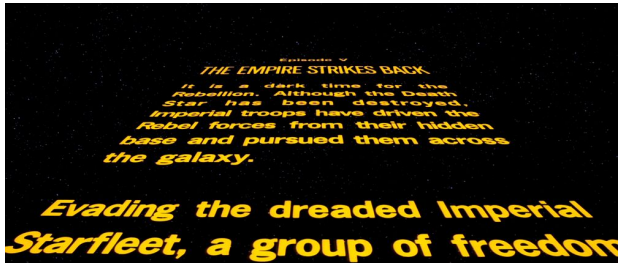
1) Longest Matching

2) Maximal Matching

- Dynamic Programming
- LexTo

The need for segmentation

- Text as a stream of characters



- We need a way to understand the meaning of text
  - Break into words (assign meaning to word) ← Tokenization
  - Break into sentences (put word meanings back to sentence meaning)

Tokenization

- A token should
  - 1. Linguistically significant
  - 2. Methodologically useful

Tokenization - Thai

- Thai has no space between words
- Thai has no clear sentence boundaries

- เริ่มจากชนวนของสงครามแรกสุด สมาพันธ์การค้า ทำการบีบล้อม-รุกรานดาวนาบู ส่งผลต่อมาขยายเป็นสงครามโคลนอันมีฝ่ายแบ่งแยกดินแดนเป็นผู้ชักใยสงคราม หมายโค่นล้มฝ่ายสาธารณรัฐ ขนานไปกับเรื่องราวพัฒนาการของของเด็กน้อยคนหนึ่งผู้มีพลังสถิตแรงมาก ชาวดาวทะเลทรายทาทูอีนนาม "อนาคิน สกายวอล์คเกอร์" ผู้ถูกคาดการณ์ว่าคือผู้ถูกเลือกในตำนานของเจได หลังจากสงครามยุทธการดาวนาบู อนาคิน ก็ได้รับฝึกฝนในวิถีเจได โดย อ.เจได "โอบีวัน" แต่ เมื่ออนาคินโตเป็นหนุ่มก็ดันแหกกฎเจไดโดยแอบมีความสัมพันธ์ชู้สาวลับๆกับ ราชินี "อมิดาลา" แห่งดาวนาบู จนเธอตั้งครรภ์ลูกแฝด ... และอนาคินก็ค่อยๆถูกกลืนเข้าสู่ด้านมืดของพลังจนกลายเป็นซิธลอร์ด ใต้ฉายา "ดาร์ธ เวเดอร์" ภายใต้การโน้มน้าวชี้นำของ ซิธลอร์ดลึกลับนาม "ดาร์ธ ซีเดียส" ซึ่งเผยในตอนท้ายว่า ซีเดียส ไม่ใช่ใครที่ไหน แต่คือท่าน "พัลพาทีน" สมุหนายกผู้นำสูงสุดของฝ่ายสาธารณรัฐเสียเอง และแท้จริงก็ยังเป็นผู้นำลับชักใยฝ่ายแบ่งแยกดินแดนด้วยอีกต่างหาก ... สงครามจบลงที่ฝ่ายสาธารณรัฐพ่ายแพ้ล่มสลาย อมิดาลาก็ตายหลังคลอดลูกแฝด ทั้งเหล่าอัศวินเจไดก็ถูกฆ่ากวาดล้างสิ้นแบบไม่ทันตั้งตัว เหลือแต่ อ.โอบีวัน กับ อ.โยดา ต้องลี้ภัยหลบหนีซ่อนตัว โดยลูกแฝดของอนาคินได้ถูกส่งไปสู่ที่หลบซ่อนลับเช่นกัน ... และแล้ว พัลพาทีน ก็กินรวบทั้งกระดาน เปลี่ยนการปกครองจาก ระบอบสาธารณรัฐเดิมไปเป็น ระบอบเผด็จการจักรวรรดิซิธแทน ตั้งตนเป็นจักรพรรดิปกครองแกแลคซีทั้งปวง โดยมี ดาร์ธ เวเดอร์ เป็นขุนพลซิธลอร์ดเคียงข้างนับแต่นั้นมา

http://www.starwars.siligon.com/swcong.html

Tokenization - Thai

Social media text

#สตอรี่ของโม 😊🐙 #Days23ofMobile 🌈 ...23 วันแล้วนะเจ้าโม พี่กำลังคิดถึงหนูอยู่เลย แหนะ เมื่อวานหายเลยนะ 🙈 พี่ก็ว่าหายไปไหนแอบหนีไปเล่นลองบอร์ดนี่เอง เล่นระวังๆนะพี่เป็นห่วง เดวล้มแถมไม่มีตะหลามคอยเล่นเป็นเพื่อนอีก 😊 พี่จะบอกว่า "หนูอย่าลืมลงชื่อเลือกตั้งนะ" เดี๋ยวพวกพี่ขำกันไม่ออกนะ 5555 ฝากไว้กันลืม ยิ่งเด๋อๆอยู่อีกอย่างๆหนูต้องกลับมานะพวกเรารอหนูอยู่ 😊 พี่นี่เตรียมรอโหวตให้หนูเต็มที่เล้ย 😊 ไปเรียนเป็นไงบ้าง ยังเหงาอยู่มั้ย แต่พี่ว่าคงไม่แล้วหละมั้ง วันนี้ขึ้นสเตจอีกแล้วสินะ เหนื่อยมั้ยคะ แต่คงสนุกมากกว่าอยู่แล้วเนอะ 😁 แค่ได้เห็นรอยยิ้มของหนูพี่ก็สบายใจและ แต่เอ๊ะ เมื่อวานใครบ่นอยากกลับบ้านอีกแล้วนะ อดดู the toy เลยอ่ะดิ้ 😊 หว่ายๆๆๆ น่าสงสาร 555 โอกาสหน้ายังมีนะ ตั้งใจทำงานนะคะ เจ้าหลามแฟนหนูก็อด ไม่ได้อดคนเดียวซะหน่อยนะ 555 🙈 ดูมีความสุขจังน้าเจ้าตัวเล็ก 😊💗 คิดถึงนะ เดี๋ยวก็ได้ 2shot กันแล้ว พี่โคตรตื่นเต้นเลย ตื่นเต้นกว่าไปจับมือเยอะ 🙈 คิดท่าไม่ออกเล้ย มีท่าไหนแนะนำมั้ย ช่วยพี่หน่อยยยยย 😊... #Mobilebnk48 #ตู้เพลงโมบิล #ชาวเหรียญหยอดตู้ #MOTA09

Tokenization - Thai

- Many word boundaries depends on the context (meaning)

  ตา กลม vs ตาก ลม **คณะกรรมการตรวจสอบถนน**

- Even amongst Thais the definition of word boundary is unclear
  - Needs a consensus when designing a corpus
  - Sometimes depends on the application
    - Linguist vs machine learning concerns

Dictionary-based vs Machine-learning-based

- Dictionary-based
  - Longest matching
  - Maximal matching
- Machine-learning-based

Dictionary-based word segmentation

- Perform by scanning a string and match each substring against words from a **dictionary**. (No dataset needed, just prepare a dictionary!)
- However, there is ambiguity in matching.(There are many many ways to match)

ป้ายกลับรถ

- So, **matching methods** are developed:
  1. Longest matching
  2. Maximal matching

Meknavin, Surapant, Paisarn Charoenpornsawat, and Boonserm Kijsirikul. "Feature-based Thai word segmentation." Proceedings of Natural Language Processing Pacific Rim Symposium. Vol. 97. 1997.

1) Longest Matching

- Scan a sentence from left to right
- Keep finding a word from the starting point, until no word matched, then move to the next point
  - Backtrack if current segmentation leads to an un-segmentable chunk

- ป้ายกลับรถ Start scanning with “ป” as the starting point
- ป้ายกลับรถ Keep scanning …
- ป้าย/กลบรถ No more words start with “ป้าย”, move to the next point
- …
- ป้าย/กลับ/รถ

2) Maximal Matching

- Generate all possible segmentations
- Select the segmentations with the **fewest** words

Haruechaiyasak, Choochart, Sarawoot Kongyoung, and Matthew Dailey. "A comparative study on thai word segmentation approaches." Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on. Vol. 1. IEEE, 2008.

What if?

- What if there are more than one segmentation with the fewest words?
- Other heuristics are applied, for example
  - Language model score (next lecture!)

> Longest Matching คุณ | อากร | กช

> Language Modeling คุณ | อา | กรกช

Maximal matching

- Maximal matching can be done using dynamic programming.
- Let d(i,j) be the function which returns number of the fewest word possible with the last word starts with i^{th} character (row) and ends with j^{th} character (column). It can be defined as:

$$d(i,j) = \begin{cases} 1 & \text{if i=1 and c[1, j] is in the dictionary.} \\ 1 + min_{k=1\ldots i-1} d(k, i-1) & \text{if c[i..j] is in the dictionary.} \\ \infty, & \text{otherwise.} \end{cases}$$

when c[i..j] is a string of word in the sentence (assume it is started at index 1) and the base case is d(1,1) = 1.

Maximal matching: Initialization (1st row)

- Maximal matching can be done using dynamic programming.
- Let d(i,j) be the function which returns number of the fewest word possible with the last word starts with i^{th} character (row) and ends with j^{th} character (column). It can be defined as:

1st row and it is a word

$$d(i,j) = \begin{cases} 1 & \text{if i=1 and c[1, j] is in the dictionary.} \\ 1 + min_{k=1\ldots i-1} d(k, i-1) & \text{if c[i..j] is in the dictionary.} \\ \infty, & \text{otherwise.} \end{cases}$$

when c[i..j] is a string of word in the sentence (assume it is started at index 1) and the base case is d(1,1) = 1.

Maximal matching: Find a word in dictionary

- Maximal matching can be done using dynamic programming.
- Let d(i,j) be the function which returns number of the fewest word possible with the last word starts with i^{th} character (row) and ends with j^{th} character (column). It can be defined as:

$$d(i,j) = \begin{cases} 1 & \text{if i=1 and c[1, j] is in the dictionary.} \\ 1 + min_{k=1\ldots i-1} d(k, i-1) & \text{if c[i..j] is in the dictionary.} \\ \infty, & \text{otherwise.} \end{cases}$$

If last word is a word

when c[i..j] is a string of word in the sentence (assume it is started at index 1) and the base case is d(1,1) = 1.

Maximal matching: Check all possible segmentations before the final word

- Maximal matching can be done using dynamic programming.
- Let d(i,j) be the function which returns number of the fewest word possible with the last word starts with i^{th} character (row) and ends with j^{th} character (column). It can be defined as:

$$d(i,j) = \begin{cases} 1 & \text{if i=1 and c[1, j] is in the dictionary.} \\ 1 + min_{k=1\ldots i-1} d(k, i-1) & \text{if c[i..j] is in the dictionary.} \\ \infty, & \text{otherwise} \end{cases}$$

Check all possible segmentations before the final word
Check the whole column in the previous row

when c[i..j] is a string of word in the sentence (assume it is started at index 1) and the base case is d(1,1) = 1.

Maximal matching: The final word

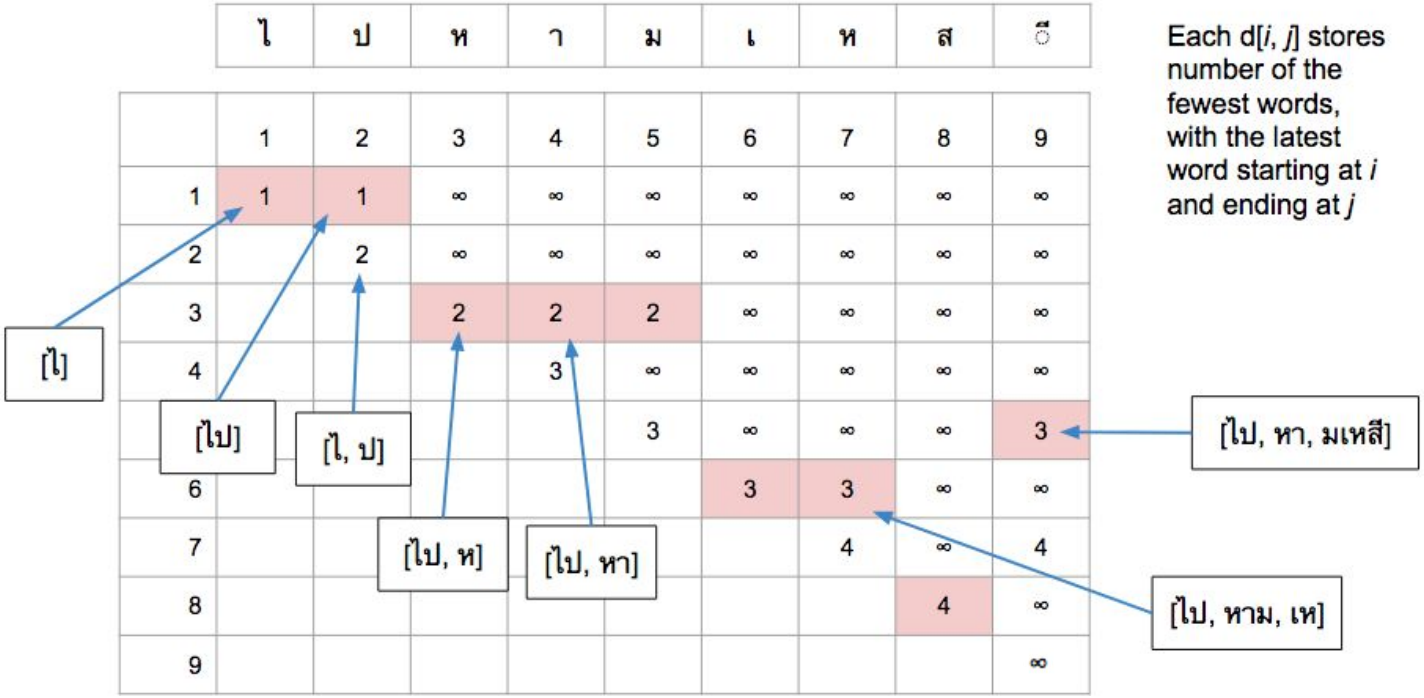
- Maximal matching can be done using dynamic programming.
- Let d(i,j) be the function which returns number of the fewest word possible with the last word starts with i^{th} character (row) and ends with j^{th} character (column). It can be defined as:

$$d(i,j) = \begin{cases} 1 & \text{if i=1 and c[1, j] is in the dictionary.} \\ 1 + min_{k=1\ldots i-1} d(k, i-1) & \text{if c[i..j] is in the dictionary.} \\ \infty, & \text{otherwise.} \end{cases}$$

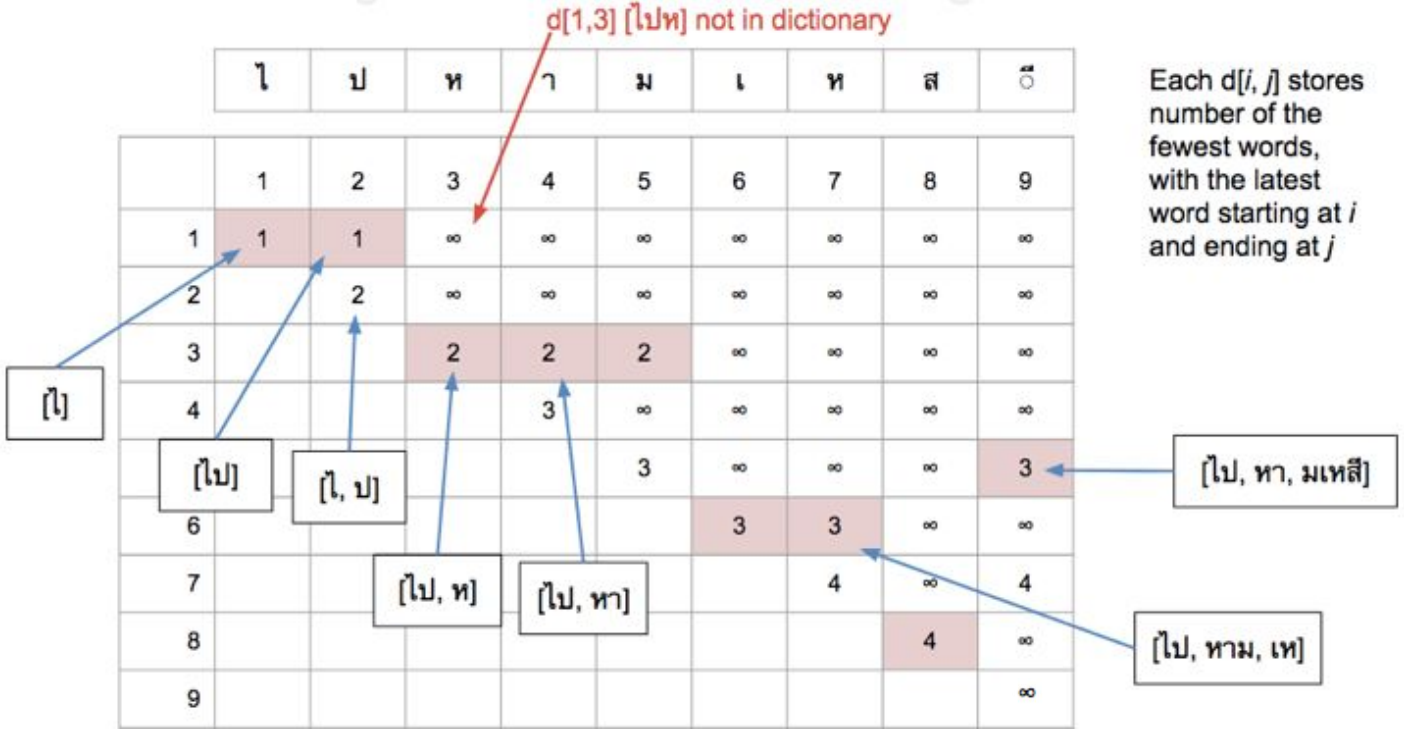
The final word

when c[i..j] is a string of word in the sentence (assume it is started at index 1) and the base case is d(1,1) = 1.

Dynamic programming example (1)

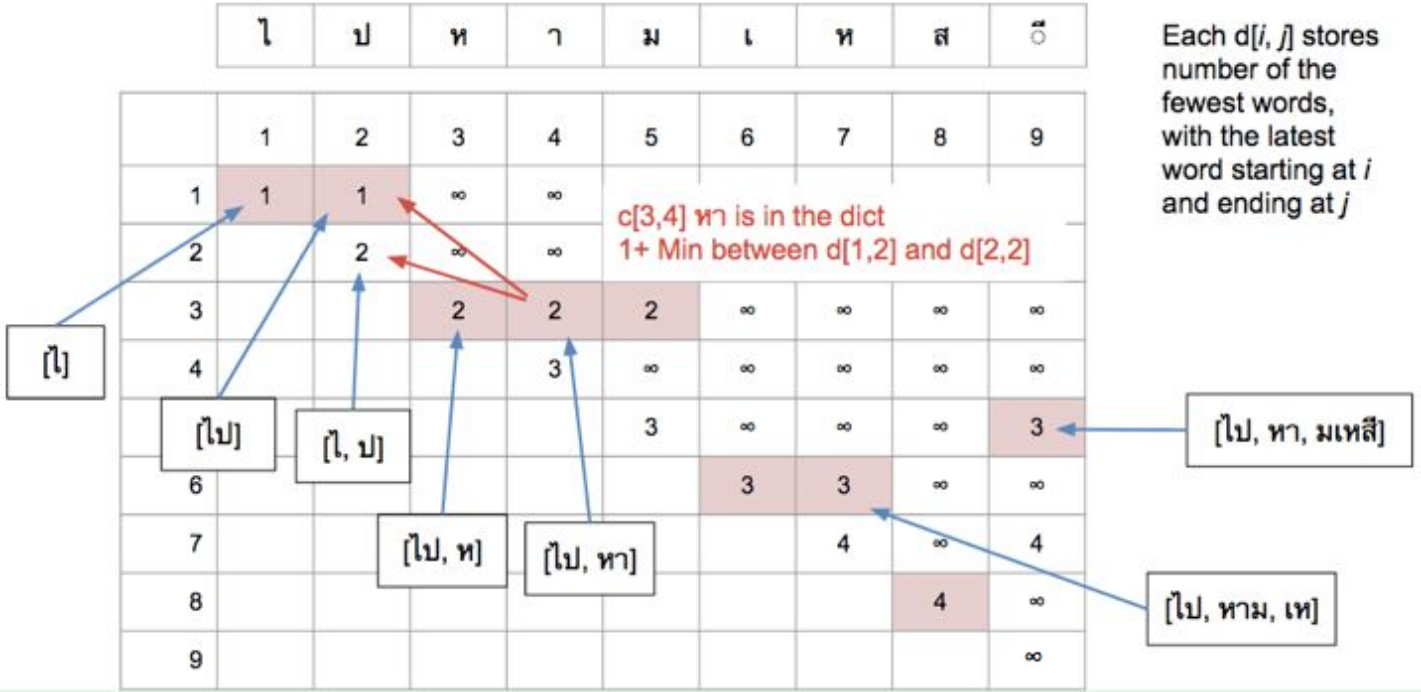


Dynamic programming example (2)



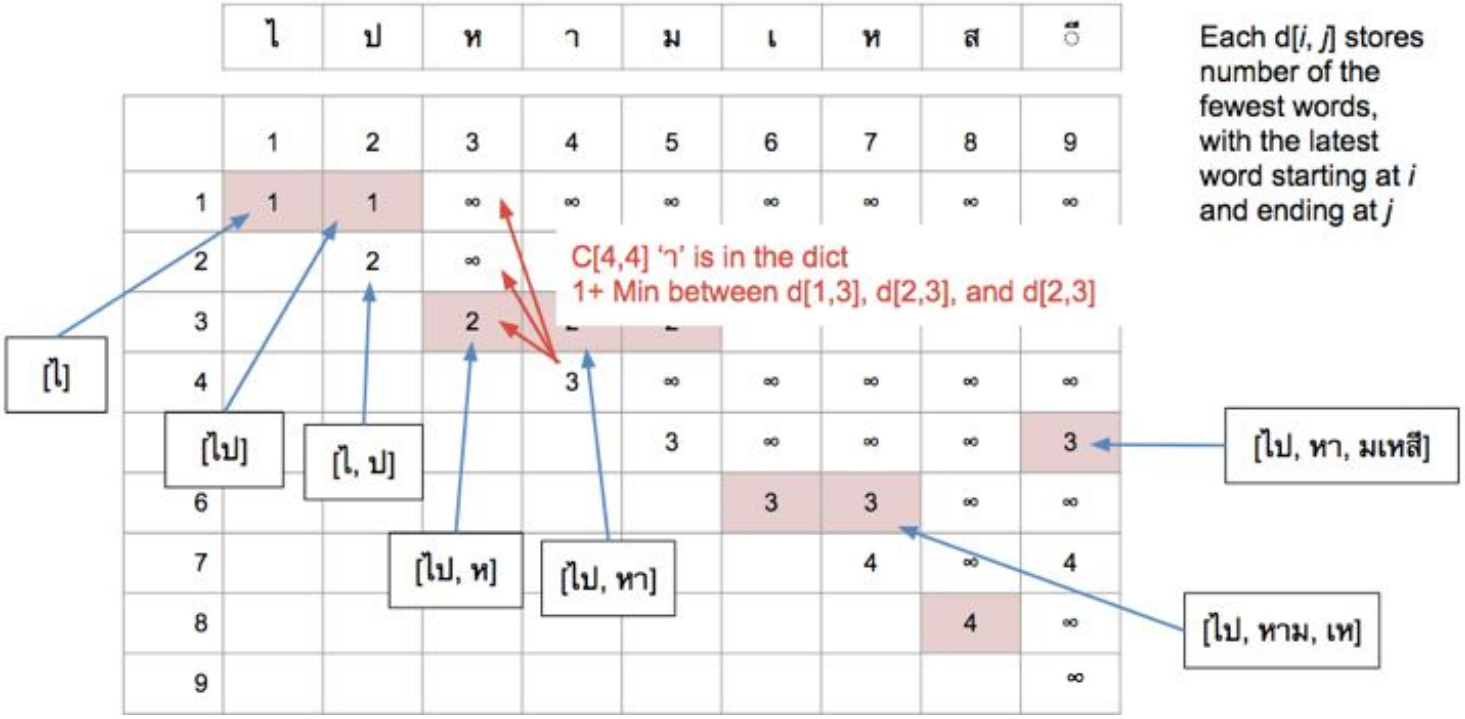
Dict: ไ, ป, ห, า, ม, เ, ส, ี, ไป, หา, หาม, เห, สี, มเหสี

Dynamic programming example (3)



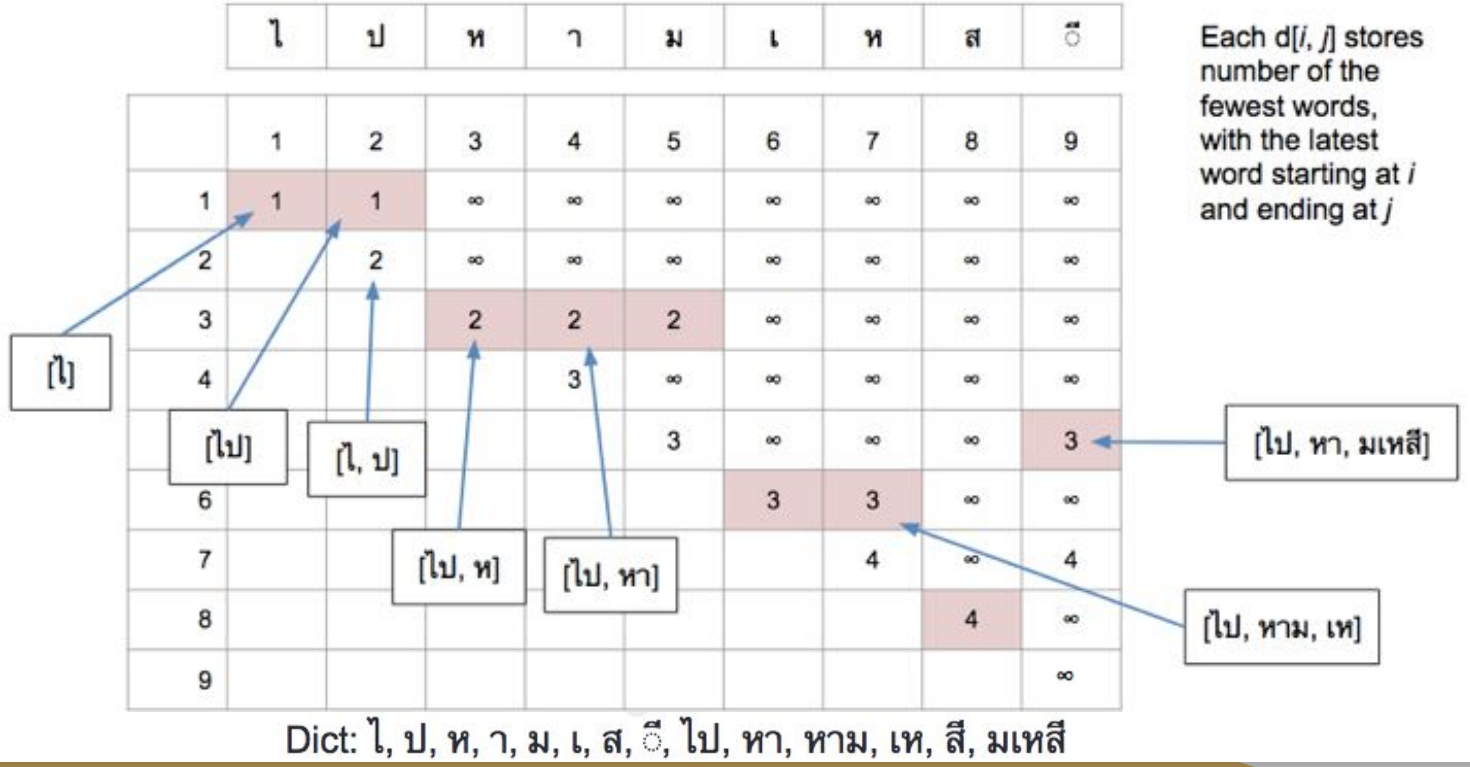
Dict: ไ, ป, ห, า, ม, เ, ส, ี, ไป, หา, หาม, เห, สี, มเหสี

Dynamic programming example (4)



Dict: ไ, ป, ห, า, ม, เ, ส, ี, ไป, หา, หาม, เห, สี, มเหสี

Dynamic programming example (5):
Backtracking



LexTo

- http://www.sansarn.com/lexto/
- Dictionary-based longest matching