ขั้นตอนวิธีการตัดคำ
สำหรับการตรวจสอบตัวสะกดภาษาไทย

A word segmentation algorithm
for Thai spell checking

ทิวา เจริญสวัสดิพงศ์
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

NAC2005
NSTDA Annual Conference
S&T in Thailand : Towards the Molecular Economy

ปัญหา

- ปัญหาของการตรวจสอบตัวสะกดภาษาไทย คือ เขียนคำติดกัน
- ไม่ทราบขอบเขตของคำ
- ต้องนำข้อความมาผ่านการตัดคำก่อน
- นำคำที่ได้ไปเทียบกับคำในพจนานุกรม

เสนอ

- อัลกอริธึมการตัดคำเพื่อการตรวจสอบตัวสะกดภาษาไทย
- ประสิทธิภาพการตรวจสอบตัวสะกดดีกว่าที่มีอยู่ในปัจจุบัน

โปรแกรมประมวลผลคำทั่วไป

- ไมโครซอฟท์เวิร์ด 2002, ปลาดาวออฟฟิศ 2.0
- ขีดเส้นใต้เพื่อระบุการสะกดผิด ไม่ตรงกับจุดที่ผิดจริง ๆ ซึ่งมีผลทำให้ไม่สามารถแก้ไขคำนั้นได้ถูกต้อง

คำที่สะกดผิด	สะกดผิดจาก	ตำแหน่งที่ผิด	บริเวณที่ขีดเส้นใต้ระบุการสะกดผิด	
			ไมโครซอฟท์เวิร์ด 2002	ปลาดาวออฟฟิศ 2.0
อิเล็กทรตอนิกส์	อิเล็กทรอนิกส์	ต (พิมพ์เกิน)	อิเล็กทร	อิเล็กทร
วิชาากร	วิชาการ	าก (พิมพ์สลับ)	วิชาา	วิชาา
ไฝยากรณ์	ไวยากรณ์	ฝ (พิมพ์ผิด)	กรณ์	กรณ์
อุปกรณ	อุปกรณ์	ณ (พิมพ์ตก)	อุปก	ไม่มี

อิเล็กทรตอนิกส์

วิชาากร

ไฝยากรณ์

อุปกรณ

โปรแกรมประมวลผลคำทั่วไป (2)

- หากคำผิดสามารถแตกย่อยเป็นคำที่ถูกได้ จะไม่ระบุว่าเป็นคำผิด

คำที่สะกดผิด	พิมพ์ผิดจาก	คำย่อย
รัฐนมตรี	รัฐมนตรี	รัฐ นม ตรี
รัฐปะหาร	รัฐประหาร	รัฐ ปะ หาร
สามทารถ	สามารถ	สาม ทา รถ

โปรแกรมตัดคำทั่วไป

- โปรแกรมที่นำมาทดสอบ ได้แก่ Swath, ICU
- ตัดคำผิดพลาดเมื่อได้รับข้อความที่มีคำสะกดผิด

คำที่สะกดผิด	สะกดผิดจาก	ผลการตัดคำจากโปรแกรมตัดคำ	
		Swath	ICU
เทครโนโลยี	เทคโนโลยี	เท|คร|โน|โล|ยี	เท|คร โน โลยี|
เจตนรมณ์	เจตนารมณ์	เจ|ตน|รมณ์|	เจ|ตนรมณ์|

- คำยาวสะกดผิด แต่สามารถตัดเป็นคำถูกได้ -> คำถูกย่อย ๆ

คำที่สะกดผิด	สะกดผิดจาก	ผลการตัดคำจากโปรแกรมตัดคำ	
		Swath	ICU
รัฐนมตรี	รัฐมนตรี	รัฐ|นม|ตรี|	รัฐ|นม|ตรี|
กระทรงวง	กระทรวง	กระ|ทรง|วง|	กระ|ทรง|วง|

การสะกดผิด

- คำที่พิมพ์ผิด เกิดขึ้นจากความผิดพลาด 4 ชนิด [Damerau, 1964]
  - พิมพ์เกิน เช่น วิทยาลัย พิมพ์เป็น วิทยยาลัย
  - พิมพ์ตก เช่น อาหาร พิมพ์เป็น อหาร
  - พิมพ์ผิด เช่น นักเรียน พิมพ์เป็น นักเรือน
  - พิมพ์สลับ เช่น รัฐบาล พิมพ์เป็น รัฐาบล

พฤติกรรมการพิมพ์ผิดพลาดของมนุษย์

- ผู้ที่พิมพ์ชำนาญมักผิดพลาดจากการพิมพ์เกินใน 2 แป้นที่อยู่ติดกัน
- ผู้ที่พิมพ์ไม่ชำนาญมักผิดพลาดจากการพิมพ์ผิด และการพิมพ์ผิดนั้นมักเกิดจากแป้นที่อยู่ติดกัน [Grudin, 1983]
- มีเพียงส่วนน้อยมากของความผิดพลาดที่เกิดจากอักขระตัวแรกของคำ [Kukich, 1992]

วิธีการตัดคำแบบเหมือนมากที่สุด (Maximal Matching) [วิรัช, 2536]

- หาวิธีในการตัดคำที่สามารถจะเป็นไปได้ทั้งหมด
- เลือกแบบที่จำนวนคำน้อยที่สุด
- เช่น เมื่อมีข้อความว่า "ไปหามเหสี" ก็จะตัดคำได้ 2 เส้นทาง คือ
  - 1. ไป หาม เห สี มีจำนวนคำเท่ากับ 4
  - 2. ไป หา มเหสี มีจำนวนคำเท่ากับ 3
- เลือกเส้นทางที่ 2 เนื่องจากมีจำนวนคำน้อยที่สุด

ระยะแก้ไข (Edit Distance)

- ระยะแก้ไข ระหว่าง 2 สตริง คือจำนวนครั้งที่น้อยที่สุดที่ต้องการในการแก้ไข (เพิ่ม,ลบ,เปลี่ยน) สตริงหนึ่งให้เท่ากับอีกสตริงหนึ่ง
- วิทยาลัย กับ วิทยยาลัย ระยะแก้ไขเท่ากับ 1
- อาการ กับ อาหารร ระยะแก้ไขเท่ากับ 2

การหาระยะแก้ไข

- Damerau-Levenshtein Dynamic Programming [Levenshtein, 1965]
  - Edit distance between string s and t
  - cost = (s[i] == t[j])? 0 : 1
  - d[i, j] = Min (d[i-1, j]+1 ,d[i, j-1]+1, d[i-1, j-1]+cost)

		อ	า	ห	า	ร
	0	1	2	3	4	5
อ	1	0	1	2	3	4
ห	2	1	1	1	2	3
า	3	2	1	2	1	2
ร	4	3	2	2	2	1

การจับคู่แบบประมาณ (Approximate Matching)

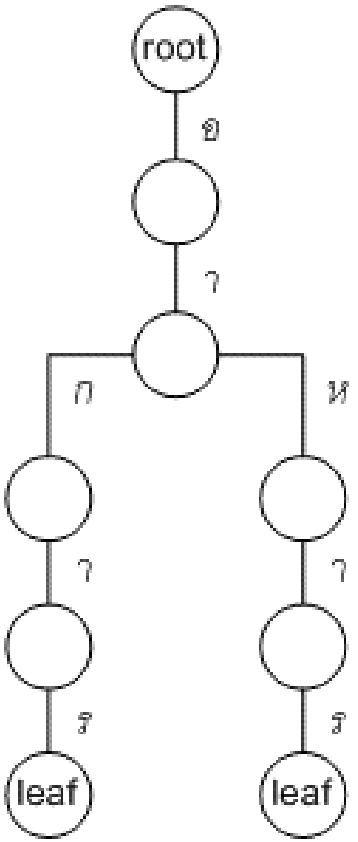
- การจับคู่ระหว่างคำซึ่งมีระยะแก้ไขไม่เกินค่าที่กำหนด
- ใช้ในการหาคำที่ต้องการจากเซตของคำที่เก็บไว้ (พจนานุกรม) โดยอนุญาตให้มีความผิดพลาดเกิดขึ้นได้ตามแต่จะกำหนด (Approximate Searching) เช่น
  - พจนานุกรมมีคำว่า “อาหาร” “อากร” “อาการ” “อาหร”
  - ค้นหาคำว่า “อาหร” โดยความผิดพลาดไม่เกิน 1 จะได้คำว่า “อาหร” “อาหาร” และ “อากร” เป็นคำตอบ

ทรัย (Trie)

- เป็นโครงสร้างข้อมูลแบบทรี (Tree) ซึ่งโหนดมีค่าว่าง (Empty)
- ข้อมูลจะอยู่บนเส้นทางการเดินระหว่างโหนดราก (Root Node) ไปยังโหนดใบ (Leaf Node)
- สามารถนำทรัยมาใช้ในการเก็บคำในพจนานุกรมเพื่อค้นหาคำที่อยู่ในพจนานุกรม โดยแต่ละกิ่ง (Branch) ระหว่างโหนด จะมีอักขระกำกับอยู่

ตัวอย่างทรัยที่เก็บคำในพจนานุกรม

- เก็บคำว่า “อาการ” และ “อาหาร”



ข้อดีของการใช้ทรัยสำหรับเก็บคำในพจนานุกรม

- ใช้ส่วนหน้าของคำ (prefix) ร่วมกัน
- ประหยัดเนื้อที่ ง่ายต่อการเพิ่มคำ
- เวลาที่ใช้ไม่ขึ้นกับจำนวนคำในพจนานุกรม แต่ขึ้นกับความยาวของสตริงที่ต้องการค้นหา

การค้นหาโดยการจับคู่แบบประมาณบนทรัย

[H.Shang, T.H.Merrett, 1996]

- ค้นหาโดยเดินไปตามทรัยแบบ Depth First Search, Recursive
- ขณะค้นหาเมื่ออยู่ที่ทรัยย่อยใด ๆ พบความผิดพลาด หากพบ Ukkonen's Cutoff จะหยุดการค้นหาในทรัยย่อยนั้น (ไม่จำเป็นต้องเดินให้ถึง leaf node) และย้อนกลับไปค้นหาในทรัยย่อยถัดไป

Ukkonen's Cutoff [Ukkonen, 1985]

- ลดการคำนวณตาราง ในเซลล์ที่ไม่จำเป็นต้องคำนวณ เนื่องจากค่าที่นำมาคำนวณเกินความผิดพลาดที่ยอมรับได้แล้ว

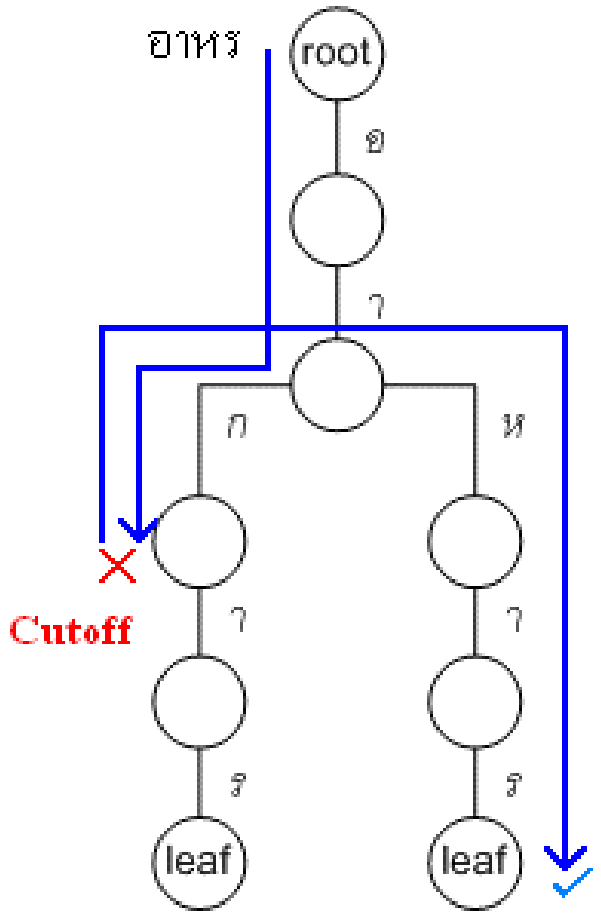
		อ	า	ก	า	ร
	0	1	2	3	4	5
อ	1	0	1	2		
ห	2	1	1	2		
า	3	2	1	2		
ร	4	3	2	2		

ตัวอย่างการค้นหาโดยการจับคู่แบบประมาณบนทรัย

- ในพจนานุกรมมีคำว่า

“ อาการ “ และ “ อาหาร “

- คำที่นำค้นหาแบบประมาณ “อาหร”



แนวคิดการออกแบบอัลกอริธึมการตัดคำ

- จากวิธีการตัดคำแบบ Maximal Matching
- นอกจากพิจารณาจำนวนคำถูก ควรพิจารณาถึงโอกาสของคำที่จะเป็นคำผิด 1 อักขระด้วย
  - เดิม # [คำถูก + คำถูก + คำถูก + …] > # [คำถูก] ดังนั้น เลือกด้านขวา
  - ใหม่ # [คำถูก + คำถูก + คำถูก + …] > # [คำผิด 1อักขระ] ควรเลือกด้านขวา
- เช่น
  - # [รัฐ + มน + ตรี] > # [รัฐมนตรี] ดังนั้นตัดคำเป็น รัฐมนตรี
  - # [รัฐ + นม + ตรี] > # [รัฐนมตรี] ควรจะตัดเป็น รัฐนมตรี
  - # [กระ + ทรง + วง] > # [กระทรงวง] ควรจะตัดเป็น กระทรงวง

แนวคิดการออกแบบอัลกอริธึมการตัดคำ(2)

- การพิจารณาจากจำนวนคำเพียงอย่างเดียว อาจเกิดความผิดพลาดใน 2 กรณี

  1. คำถูก + คำถูก ไม่ควรตัดคำเป็น คำผิด
     - ว่า การ ไม่ควรตัดคำเป็น ว่าการ (วฏาการ)
     - จะ ต้อง ไม่ควรตัดคำเป็น จะต้อง (มะต้อง)
  2. คำถูก + คำถูก + คำถูก (ถูก 3) ไม่ควรตัดคำเป็น คำผิด + คำผิด (ผิด 2)
     - ทดลอง พบ ว่า -> ทดลองพ(ทดลอง) + บว่า(บ่า)

แนวคิดการออกแบบอัลกอริธึมการตัดคำ (3)

- ดังนั้นควรกำหนดค่า cost ของคำถูกและคำผิด แล้วเลือกด้านที่ cost น้อยกว่า
- จากวิธีการเลือกตัดคำที่ผ่านมา ทำให้กำหนดความสัมพันธ์ของ costได้ดังนี้

  กำหนดให้ R=cost of right word, W=cost of misspelled word

  - cost [คำถูก + คำถูก + …] > cost [คำถูก] (mR > nR; m > n) (Maximal Matching)
  - cost [คำถูก + คำถูก + คำถูก + …] > cost [คำผิด] (nR > W; n > 2) => 3R > W
  - cost [คำถูก] < cost [คำผิด]
  - cost [คำถูก + คำถูก] < cost [คำผิด] (nR < W; n <=2) => 2R < W
  - cost [คำถูก + คำถูก + คำถูก] < cost [คำผิด + คำผิด] (3R < 2W)
- สรุปได้ว่า 2R < W <3R
- ต้องการทราบค่าของ W ที่เหมาะสม

การทดสอบเพื่อหาค่าต้นทุนของคำผิดที่เหมาะสม

- Orchid Corpus ของ NECTEC
- สุ่มเลือกมา 1250 บรรทัด
- 250 บรรทัด สร้างความผิดพลาดบรรทัดละ 1 อักขระ ทั้ง 4 แบบ
- ถูกต้อง 1000 บรรทัด ผิดพลาด 1000 บรรทัด
- การวัดประสิทธิภาพ
  - Recall = จำนวนคำผิดที่พบ / จำนวนคำผิดทั้งหมด
  - Precision = จำนวนคำผิดที่พบและผิดจริง / จำนวนคำผิดที่พบทั้งหมด

การทดสอบเพื่อหาค่าต้นทุนของคำผิดที่เหมาะสม(2)

ต้นทุนของคำผิด (W)	Recall	Precision
1.9R	957 / 1000 = 95.7%	957 / 1449 = 66.05%
2.0R	839 / 1000 = 83.9%	839 / 839 =100%
2.5R	839 / 1000 = 83.9%	839 / 839 =100%
2.9R	839 / 1000 = 83.9%	839 / 839 =100%
3.0R	806 / 1000 = 80.6%	839 / 839 =100%

อัลกอริธึมการตัดคำสำหรับการตรวจสอบตัวสะกด

- ค้นหาคำที่เป็นไปได้ทั้งหมดในสตริง ดังนี้
  - คำที่พบในพจนานุกรม => คำถูก
  - คำที่ไม่พบในพจนานุกรม ผิดไม่เกิน 1 อักขระ ในพจนานุกรม => คำผิด
- เลือกเส้นทางตัดคำที่ต้นทุนน้อยที่สุด โดยพิจารณาดังนี้
  - คำถูก มีต้นทุนเท่ากับ 1
  - คำผิด มีต้นทุนเท่ากับ 2.5
  - **ต้นทุน = (จำนวนคำถูก * 1) + (จำนวนคำผิด * 2.5)**
  - หากต้นทุนเท่ากัน ใช้หลัก Longest Matching ในการเลือกเส้นทาง

การทดสอบประสิทธิภาพ

	ขั้นตอนวิธีที่นำเสนอ	ไมโครซอฟท์เวิร์ด 2002	ปลาดาวออฟฟิศ 2.0
จำนวนบรรทัดทั้งหมด	2000	2000	2000
จำนวนบรรทัดที่ถูก (AR)	1000	1000	1000
จำนวนบรรทัดที่ผิด (AW)	1000	1000	1000
คำที่ถูกระบุว่าผิด (IW)	839	757	783
คำผิดที่ถูกพบ (DW)	839	757	783
Recall (DW / AW)	**83.90%**	**75.70%**	**78.30%**
Precision (DW / IW)	100%	100%	100%
พบบนตำแหน่งที่ผิด (AP)	782	664	633
ความแม่นยำในการตรวจพบบนตำแหน่งที่ผิด (AP / DW)	**93.21%**	**87.71%**	**80.84%**
แนะนำคำที่ถูกต้อง (AS)	713	320	258
ความสามารถในการแก้ไขเมื่อตรวจสอบพบ (AS / DW)	**84.98%**	**42.27%**	**32.95%**

วิเคราะห์ผลการทดสอบประสิทธิภาพ

- สามารถตรวจสอบคำผิดดีกว่าอีก 2 โปรแกรมเล็กน้อย
- สามารถในการแก้ไขเมื่อตรวจสอบพบได้ดีกว่ามาก
- เนื่องจากการตัดคำเมื่อข้อความมีความผิดพลาดทำได้ดีกว่าเมื่อเทียบกับอีก 2 โปรแกรม จึงทำให้สามารถระบุคำผิดได้ถูกต้อง และแนะนำคำที่ถูกได้

ผลการทดสอบประสิทธิภาพ แบ่งตามประเภทของความผิดพลาด

ประเภทความผิดพลาด	ทั้งหมด	ตรวจสอบพบ	ความสามารถในการตรวจพบ
พิมพ์เกิน	250	221	88.4%
พิมพ์ผิด	250	207	82.8%
พิมพ์ตก	250	177	70.8%
พิมพ์สลับ	250	234	93.6%
รวม	1000	839	83.9%

วิเคราะห์ผลการทดสอบประสิทธิภาพ (2)

- พิมพ์ตกจะสามารถตรวจสอบพบได้น้อยที่สุด เนื่องจากคำในพจนานุกรมส่วนใหญ่จะเป็นคำเล็ก การพิมพ์ตกทำให้อักขระลดน้อยลง จึงมีโอกาสที่จะเป็นคำถูกแม้ว่าจะพิมพ์ตก จึงตรวจสอบพบได้ยากกว่าแบบอื่น ๆ
- พิมพ์สลับสามารถตรวจสอบพบได้มากที่สุด เพราะอาจมองเป็นการพิมพ์ผิดถึง 2 ที่ จึงมีโอกาสน้อยที่จะยังคงเป็นคำถูกอยู่หลังจากการสลับที่ จึงตรวจสอบพบได้ง่ายกว่าแบบอื่น ๆ

การทดสอบความเร็วของขั้นตอนวิธี

รหัสกรณีทดสอบ	กรณีทดสอบ	จำนวนอักขระ	เวลาที่ใช้ (วินาที)
TDS6-001	การประชุมทางวิชาการโครงการวิจัยและพัฒนาปีงบประมาณศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ	148	8.922
TDS6-002	รัฐมนตรีว่าการกระทรวงวิทยาศาสตร์ประเทศไทยได้มีการปรับเปลี่ยนโครงสร้างในการพัฒนาเศรษฐกิจของประเทศจากประเทศเกตรกรรมไปสู่ความเป็นประเทศอุตสาหกรรม	141	8.625
TDS6-003	มนุษย์ส่วนใหญ่ในวังคมจะต้องประสบกับปัญหาสังคมไม่แกระทบกระเทือนไม่ระทบกระเทือนเพื่อให้ปัญหานั้นบรรทาเบาบางลงการจราจรคิดขัดความจริงไม่ได้เป็นหลักประกนว่าจะแก้ปัญหาได้	164	9.890
TDS6-004	ความล้มเหลวของกฎเกณฑ์ที่สังคมกำหดแต่เป็นเพียงสภาวการณ์ที่ไม่สอดล้องกับคุณค่าที่กลุ่มยึดถือแต่เป็นเพียงสภาวการณ์ที่ไม่สอดคลองกับคุณค่าที่กลุ่มยึดถือในระบบปรมวลกฎหมาย	164	9.312

ข้อจำกัด

- ไม่สามารถตรวจสอบคำผิดที่สามารถแตกย่อยเป็นคำถูก 2 คำ
  - สอดคล้อง สะกดผิดเป็น สอดคลอง จะตัดคำได้ สอด|คลอง
    (ที่ถูกต้องควรตัดเป็น สอด|คลอง)

- การสลับอักขระระหว่างคำ หรือระหว่างประโยค ซึ่งทำให้ส่วนที่เหลือของคำใดคำหนึ่งถูก
  - สูงมาก สะกดผิดเป็น สูมงาก จะตัดคำได้ สู|มงาก
    (ที่ถูกต้องควรตัดเป็น สูม|งาก)
    (“สู” เป็นคำถูก ทำให้ “มงาก” ถูกมองเป็นการพิมพ์เกิน)

ข้อจำกัด(2)

- คำที่มี 2 อักขระ หากพิมพ์ตกไป 1 อักขระ อักขระที่เหลือจะถูกมองเป็นการพิมพ์เกินของคำข้างเคียง
  - ต่อไป สะกดผิดเป็น ต่อป จะตัดคำได้ ต่อป
    (ที่ถูกต้องควรตัดเป็น ต่อ|ป)

- กรณีที่ต้นทุนเท่ากัน แล้วใช้วิธี Longest Matching ในการเลือกตัดคำ อาจ ได้ผลลัพธ์ที่ไม่ถูกต้อง
  - วิธีแสดง สะกดผิดเป็น วิธีสแดง จะตัดคำได้ วิธีส|แดง
    (ที่ถูกต้องควรตัดเป็น วิธี|สแดง)

ข้อเสนอแนะ

- ขั้นตอนวิธีจับคู่ประมาณสำหรับการค้นหาคำผิดในสตริงหนึ่ง ๆ ควรได้รับการปรับปรุงให้เร็วขึ้น
- การตรวจสอบตัวสะกดในกรณีที่เป็นคำถูกอยู่ติดกัน 2 คำ ซึ่งเป็นข้อจำกัดของอัลกอริธึมนี้ สามารถใช้ข้อมูลทางสถิติ หรือทางไวยากรณ์บ่งบอกคำผิดได้

จบการนำเสนอ