คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

ข้อสอบกลางภาค ภาคการศึกษาปลาย ปีการศึกษา 2559

ข้อสอบวิชา 517 432 การประมวลผลภาษาธรรมชาติ

517 661 การประมวลผลภาษาธรรมชาติ

สอบวันอังคาร 14 มีนาคม 2560 เวลา 16.40-19.40 น. ห้อง 1641 ว.1

คำสั่ง

1. ข้อสอบมีทั้งหมด 8 หน้า 12 ข้อ 70 คะแนน+โบนัส 10 คะแนน (35 %)
2. ทำข้อสอบทุกข้อในกระดาษคำตอบ ถ้าที่ว่างในกระดาษคำตอบไม่พอ สามารถขอกระดาษเปล่าเพิ่มได้
3. ข้อที่มีเครื่องหมาย * ต่อท้ายเลขข้อ คือ ข้อที่สามารถนำไปเขียนโปรแกรมส่งมาทางอีเมล์เพื่อเพิ่มคะแนนได้หลังสอบเสร็จแล้ว ถ้าพบว่าลอกกันมาจะได้ 0 คะแนน
   3.1 ส่งอีเมล์มาที่ soonklang_t@silpakorn.edu
   3.2 กำหนด subject ชื่อ [517432Mid] รหัสนักศึกษา
   3.3 ตั้งชื่อไฟล์ รหัสนักศึกษา_mid.py โดยรวมทุกข้อไว้ในไฟล์เดียวกัน แต่ให้เขียน comment ระบุว่าเป็นข้อใดไว้บรรทัดก่อนหน้าโค้ดนั้น เช่น #exam1a
   3.4 กำหนดส่งภายใน 24.00 น. ของวันที่ 14 มีนาคม 2560
4. ข้อความที่กล่าวถึงในข้อสอบนี้หมายถึง ข้อความในภาษาอังกฤษ
5. อนุญาตให้นำเอกสารเข้าห้องสอบได้

1. อธิบายความหมายของคำต่อไปนี้โดยสังเขปและยกตัวอย่างประกอบ ไม่มี ต.ย.ไม่ได้คะแนน [10 คะแนน]

   a) WordNet

   b) Word type

   c) Lexical diversity

   d) Collocation

   e) Stopwords

   f) Part-of-speech (POS)

   g) Homonyms

   h) Antonyms

   i) Lemma

   j) Trigram

2. * text3 ในโมดูล nltk.book เป็นข้อความจากหนังสือ The Book of Genesis ซึ่งกล่าวถึงจุดเริ่มต้นของโลก มนุษย์ และอิสราเอล ซึ่งเป็นตัวแทนของชนชาติที่พระเจ้าได้เลือกไว้ ถือเป็นหนังสือเล่มแรกของคัมภีร์ไบเบิล เป็นเรื่องเล่าที่เรียงลำดับการเกิดขึ้นของสิ่งต่างๆ บนโลก [6 คะแนน]

   2.1 โครงสร้างของคลังข้อความนี้จัดอยู่ในรูปแบบใด .....

   2.2 ระบุคำสั่งสร้างกราฟการกระจายตัวของชื่อคนในลิสต์ต่อไปนี้

   names =['Abel', 'Abraham', 'Adam', 'Cain', 'Esau', 'Eve', 'Hagar', 'Ishmael', 'Jacob', 'Joseph', 'Noah', 'Sarah', 'Seth']

   2.3 จากกราฟที่ได้ วิเคราะห์ใครเกิดในยุคเดียวกันบ้าง และมีใครที่ไม่เกิดร่วมยุคกับคนอื่นหรือไม่

   2.4 ถ้าต้องแบ่งคนตาม generation คิดว่าจะแบ่งได้กี่ generation ใช้หลักเกณฑ์ใดในการแบ่ง

3. * เขียนโปรแกรมเพื่อหาว่ามีจำนวน token จำนวนคำศัพท์ (vocabulary) และค่า lexical diversity ของทุกไฟล์ใน webtext corpus ในกรณีของคำศัพท์ ถ้าเขียนด้วยตัวใหญ่หรือตัวเล็กก็ถือเป็นคำเดียวกัน ไฟล์ที่มีความหลากหลายของการใช้คำศัพท์มากที่สุดคือ..... [13 คะแนน]

Filename	Number of token	Number of vocabulary	Lexical Diversity

4. * สร้างฟังก์ชันชื่อ vocab(text) ที่รับพารามิเตอร์เป็น list ของคำ และทำ return ค่าเป็นคำศัพท์ทั้งหมด โดยไม่นับรวม stopwords และเครื่องหมายวรรคตอน ตัวเลขหรือสัญลักษณ์พิเศษต่างๆ จากนั้นทดสอบโดยเรียกใช้กับ inaugural corpus ว่ามีจำนวนคำศัพท์ทั้งหมดเท่าใด [4 คะแนน]

5. * นำคำศัพท์ที่ได้ในข้อ 4 มา plot กราฟหาคำที่มีความถี่สะสมสูงสุด 50 คำแรก [3 คะแนน]
   5.1 คำสั่งที่ใช้.....
   5.2 คำที่มีความถี่สูงสุด 5 คำแรก ได้แก่ .....

6. เขียน regular expression เพื่อใช้ในการค้นหา string ที่มีลักษณะต่อไปนี้ [4 คะแนน]
   6.1 คำที่ขึ้นต้นด้วยตัวอักษร หรือเครื่องหมาย ‘_’ เท่านั้น ห้ามขึ้นด้วยตัวเลข ส่วนตัวที่ตามมาเป็นตัวอักษร ตัวเลขหรือเครื่องหมาย ‘_’ ก็ได้ ความยาวกี่ตัวก็ได้ [2 คะแนน]

.....

   6.2 เลขจำนวนจริงที่เป็นค่าบวกหรือลบก็ได้ เช่น -2.5, 8.0, 0.4565, +9.46, .74 [2 คะแนน]

.....

7. อธิบายว่า string ลักษณะใดที่จะตรงกับ regular expression ในข้อต่อไปนี้ พร้อมยกตัวอย่าง string ประกอบ ข้อละ 2 ตัวอย่าง [4 คะแนน ข้อละ 1 คะแนน]
   7.1 [A-Za-z0-9]+

.....

   7.2 [A-Z][0-9]*

.....

   7.3 c[aeiouy]{2,4}d

.....

   7.4 [A-Za-z]+[aeiou]{2,}\w+

.....

8. * เปลี่ยนคำสั่งบรรทัดที่ 1-4 เปลี่ยนเป็น list comprehension [3 คะแนน]

```
1  sent = 'The quick brown fox jumps over the lazy dog'.split()
2  result = [ ]
3  for w in sent:
4      w_len = (w.lower(), len(w))
5      if len(w) > 4:
6          result.append(w_len)
7  print result
8  # output is [(‘quick', 5), ('brown', 5), ('jumps', 5)]
```

9. * เขียนฟังก์ชัน my_collocation(word, text) เพื่อหา bigram ที่มีความถี่มากที่สุด โดยส่งคำแรกของ bigram เข้ามาเป็นพารามิเตอร์ จากนั้นให้ทดลองเรียกใช้ฟังก์ชันโดยส่งคำทั้งหมดใน inaugural corpus ไปเป็นพารามิเตอร์ [7 คะแนน]

<u>ตัวอย่าง</u> ต้องการหาว่าคำใดที่เขียนตามหลังคำว่า United ที่มีความถี่มากสุด
เรียกใช้ฟังก์ชันโดย my_collocation(‘United’, inaugural.words())
จะได้คำตอบเป็น United **States 153**
นั่นคือ คำว่า United States มีความถี่ในการพบ 153 ครั้ง

<u>คำแนะนำ</u> คำสั่ง nltk.bigrams(text) เป็นคำสั่งที่ใช้สร้าง bigram โดย text เป็นลิสต์ของคำ
ข้อนี้ต้องใช้การนับความถี่แบบมีเงื่อนไข

ระบุคำตามหลังคำต่อไปนี้ ที่มีความถี่สูงสุดพร้อมความถี่

1. American .....
2. never .....
3. political .....

หา collocation ของ inaugural corpus ระบุอย่างน้อย 3 คำที่เป็นคำนามวลี

1. .....
2. .....
3. .....

10. * จงเขียนโปรแกรมหาข้อมูลเกี่ยวกับคำว่า “cookbook” จาก wordnet [8 คะแนน]

กำหนดให้ syn = wordnet.synset('cookbook.n.01')

10.1 คำที่เป็น synonyms (คำที่มีคำศัพท์ level เดียวกัน) ของ syn มีทั้งหมด.....คำ ได้แก่

10.2 เปรียบเทียบความคล้ายคลึงกับคำว่า “magazine” , “textbook” และ “novel” (ทุกคำเป็น noun และเลือกความหมายแรก)

คำที่มีความหมายใกล้เคียงมากที่สุด คือ .....

คำที่มีความหมายใกล้เคียงน้อยที่สุด คือ .....

ใช้คำสั่งอะไรในการตรวจสอบ และตีความหมายอย่างไร

10.3 คำระบุ path ของคำว่า cookbook โดยวาดเป็นผังลำดับชั้น (hierarchy) ตั้งแต่โหนดที่เป็น root โดยรวมคำในข้อ 9.1 และ 9.2 ไว้ในผังด้วย

11. *เขียนฟังก์ชัน syllable(word) เพื่อหาจำนวนพยางค์ของคำ โดยรับคำเข้ามาเป็นพารามิเตอร์ และเรียกใช้ cmudict แล้ว return ค่าเป็นลิสต์ของ tuple ที่ประกอบด้วยคำอ่านและจำนวนพยางค์ [9 คะแนน]

คำแนะนำ prondict = cmudict.dict() จะเก็บข้อมูลในรูปของ dictionary มี key เป็นคำ และ value คือ list ของคำอ่าน

ตัวอย่างเช่น prondict[‘fire’] = [['F', 'AY1', 'ER0'], ['F', 'AY1', 'R']]

หมายความว่า fire อ่านออกเสียงได้สองแบบ แบบแรก ['F', '**AY1**', '**ER0**'] มี phoneme สองตัวที่มีตัวเลขห้อยท้าย = 2 พยางค์ แบบที่สอง ['F', '**AY1**', 'R'] มี phoneme 1 ตัวที่มีตัวเลขห้อยท้าย = 1 พยางค์

ดังนั้น ถ้าเรียกใช้ syllable('fire')) จะได้คำตอบเป็น [(['F', 'AY1', 'ER0'], 2), (['F', 'AY1', 'R'], 1)]

12.* <u>เลือกทำข้อหนึ่งต่อไปนี้ เพียงข้อเดียว</u> โดยใช้ข้อมูลจาก brown corpus หมวด news (9 คะแนน)

a) เขียนโปรแกรมเพื่อหาว่าคำใดเป็นที่คำที่กำกวมมากสำหรับการ ติด POS tag โดยพิมพ์คำที่พบว่าติด tag มากกว่า 5 ประเภท (ไม่ต้องใส่ option tagset)

b) เขียนโปรแกรมเพื่อหาว่า tag ใดที่ปรากฏอยู่หน้า NOUN มากที่สุด 3 ลำดับแรก (ใช้ option tagset เป็น Universal)

c) เขียนโปรแกรมเพื่อหาคำ 2 คำที่มีรูปแบบคำและการ tag เป็น ‘to’ + VERB ใช้ option tagset เป็น Universal เช่น คำว่า to use, to make (เขียนคำตอบแค่ 3 ตัวอย่าง)

เลือกทำข้อ _____