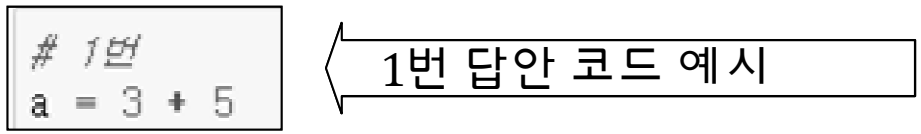
정보처리및자연어처리

2차 과제 지침

홍정하

과제 작성 및 제출 시 주의사항

- 제출기한: 5월19일(수) 23시59분(지각제출 불허)
- 배점: 총 10점(총 4 문제), 1번 5점, 2번, 4번 각 2점, 3번 1점
- 과제를 해결한 코드를 포함하는 jupyter notebook 파일(확장자 ipynb) 또는 txt 파일 또는 py 파일을 과제 게시판에 제출
  - 각 문제를 해결한 코드를 notebook 파일 또는 txt/py 파일 하나에 작성
  - notebook 파일 또는 txt/py 파일에 다음과 같이 input cell 첫 줄에 "# 문제 번호"를 쓰고 input cell 둘째 라인 이후에 해당 번호의 답안 코드 작성



- 강의에서 다루지 않은 코드/기능 사용 가능(단, 모듈은 re, nltk, pickle 3개의 모듈만 사용 가능)
- 각 문제의 코드 라인 제한수(3번, 4번은 1줄 코드 작성), 주의사항을 유의하여 코드 작성
- 각 문제마다 활용하는 변수명과 생성하는 변수명이 다르니, 문제별 변수명에 주의할 것.
- 문제 3번, 4번 제한 코드 라인수 준수
  - 한 줄 코드로 입력 가능하나 가독성을 위해 jupyter notebook에서 여러 라인으로 분할하여 제시하는 경우 한 줄 코드로 인정
  - ;을 이용하여 하나의 라인에 두 개 이상의 명령코드 사용 금지
  - 단, import 라인 및 변수 값 출력결과 확인을 위한 라인은 제한 코드 라인수에 포함되지 않음

- 채점기준
  - 각 문제에서 제한 코드 라인수를 초과하거나 (3번, 4번)금지하는 모듈/기능/코드를 사용하면 해당 문제는 오답 처리
  - 오류/미처리 기능 하나당 -1점(2-4번) 또는 -2점(1번)
  - 코딩수업에서 인정할 만한 코드로 작성할 것.(출력결과를 목록으로 만들어 출력하거나 거의 수작업에 가까운 코드 등)
- 과제 작성 기간 동안 과제 의도 또는 제시된 출력결과의 적절성에 대한 질문은 가능하나, 과제해결 또는 힌트를 위한 어떤 질문도 불가.
- 부정행위 가담자는 모두 F 처리

첨부파일 안내

- 자연어처리_2차과제지침.pdf ⇐ 지침 및 문제 소개
- data.txt ⇐ 1번 문제 입력 파일
- answer1.pkl ⇐ 1번 문제 출력 결과의 적절성 비교 파일

data.txt: 1번 문제 입력 파일

```
<distributor>국립국어원</distributor>
<idno>"BTE0075.txt, 원본:BRE0075.txt"</idno>
<availability>
 <p>배포 불가</p>
</availability>
</publicationStmt>
<notesStmt>
 <note>
  <p>21세기 세종계획 4차년도(2001년) 원시말뭉치에서 선정</p>
 </note>
</notesStmt>
<sourceDesc>
 <bibl>
  <author>이승우</author>
  <title>식물들의 사생활</title>
  <pubPlace></pubPlace>
  <publisher>문학동네</publisher>
  <date>2000</date>
```

- 문자코드: utf-8
- 세종형태분석 코퍼스
- header 부분: 서지/구축 정보

```
<body>
<head>
9BTEO0075-0000010	식물들의	식물/NNG + 들/XSN + 의/JKG
9BTEO0075-0000020	사생활	사생활/NNG
</head>
<head>
9BTEO0075-0000030	이승우	이승우/NNP
9BTEO0075-0000040	장편소설	장편/NN
</head>
<p>
9BTEO0075-0000050	왜	왜/MAG
9BTEO0075-0000060	웃어요?	웃/VV + 어/EC + 요/JX + ?/SF
9BTEO0075-0000070	하고,	하/VV + 고/EC + ,/SP
9BTEO0075-0000080	은색의	은색/NNG + 의/JKG
9BTEO0075-0000090	루즈를	루즈/NNG + 를/JKO
9BTEO0075-0000100	입술에	입술/NNG + 에/JKB
9BTEO0075-0000110	바른	바르/VV + ㄴ/ETM
9BTEO0075-0000120	거리의	거리/NNG + 의/JKG
9BTEO0075-0000130	여자가	여자/NNG + 가/JKS
9BTEO0075-0000140	눈을	눈/NNG + 을/JKO
```

- <body> ... </body>: 본문 영역
- <head> ... </head>: 제목 영역
- ... : 문단 단위 영역
- '어절번호\t어절\t형태분석'

1번: 배점 5점

다음을 수행하여 변수 tagged_paragraphs 를 만드시오.

1. data.txt 불러오기
2. 본문 영역(<body> … </body>) 내 문단 단위(…)로 구분된 어절과 형태분석 쌍의 튜플을 원소로 하는 리스트 구성
   - tagged_paragraphs = [[문단1] , [문단2], [문단3], ….]
   - 각 문단 = [(어절1, 형태분석1), (어절2, 형태분석2), …]

```
print(tagged_paragraphs[2:3])
```

```
[[('내', '나/NP + 의/JKG'), ('머릿속으로', '머릿속/NNG + 으로/JKB'), ('어떤', '어떤/
MM'), ('장면이', '장면/NNG + 이/JKS'), ('떠오른', '떠오르/VV + ㄴ/ETM'), ('것은',
'것/NNB + 은/JX'), ('그', '그/MM'), ('어느', '어느/MM'), ('순간이었다.', '순간/NNG +
이/VCP + 었/EP + 다/EF + ./SF'), ('그것은', '그것/NP + 은/JX'), ('확실히', '확실히/M
AG'), ('좀', '좀/MAG'), ('엉뚱한', '엉뚱/XR + 하/X                    으므로',
'연상/NNG + 이/VCP + 었/EP + 으므로/EC'), ('나는',                    , '나/NP
+ 도/JX'), ('모르게', '모르/VV + 게/EC'), ('피식',                    '웃/VV +
었/EP + 다/EF + ./SF'), ('내', '나/NP + 의/JKG'),                    JX'),
('채', '채/MAG'), ('형태를', '형태/NNG + 를/JKO'),                    EC'),
('못하', '못하/VX + ㄴ/ETM'), ('채', '채/NNB'), ('                    + 었/EP
+ 다/EF + ./SF')]]
```

```
len(tagged_paragraphs)
```

```
561
```

```
type(tagged_paragraphs)
```

```
list
```

- 1번 문제를 적절하게 해결했다면 첨부 파일 answer1.pkl 파일과 변수 tagged_paragraphs를 비교하면 True 출력

```
import pickle
answer1 = pickle.load(open('answer1.pkl', 'rb'))
tagged_paragraphs == answer1
```

```
True
```

2번: 배점 2점

다음을 수행하는 일련의 코드를 제시하시오.

1. 문제 1번의 변수 tagged_paragraphs로부터 전체 문단 중 90%(앞부분) 문단을 변수 train으로(training data), 나머지 10%(뒷부분) 문단을 변수 test로(test data) 만들기
2. 변수 train(training data)을 이용하여 unigram tagger, bigram tagger, trigram tagger로 구성된 combining tagger 만들고 변수는 combining_tagger로 설정(default tagger는 포함하지 않음)
3. 변수 test (test data)을 이용하여 combining tagger 변수 combining_tagger의 정확도를 측정하시오.

```
0.6167754897036665
```

combining tagger정확도

3번: 배점 1점

문제 2번의 변수 test로부터 어절만 순서대로 추출한 변수 test_words를 만드시오.(반드시 네모칸에 들어갈 한 줄 코드로 작성하시오)

```
test_words = [                                                    ]
print(test_words[:20])
```

```
['내', '계산은', '들어맞았다.', '나는', '우리', '가족들이', '마음으로는', '다들',
'원하면서도', '선뜻', '행동으로', '옮기지', '못하고', '있는', '일이', '무엇인지를',
'확신했고,', '그', '일을', '할']
```

```
print(test_words[-20:])
```

```
['것이', '문학일', '테니까요.', '연재', '기회를', '준', '「작가」와', '이', '불성실
한', '작가에게', '지속적인', '애정과', '신뢰를', '보내준', '문학동네에', '고마움을',
'전합니다.', '2000년', '가을', '이승우']
```

```
len(test_words)
```

```
3982
```

4번: 배점 2점

위쪽 그림으로부터 아래 그림을 도출할 수 있는 함수 defaultNNG 네모칸에 들어갈 한 줄 코드로 작성하시오. (None ⇒ '어휘/NNG')

```
print(combining_tagger.tag(test_words[:10]))
```

```
[('내', '나/NP + 의/JKG'), ('계산은', '계산/NNG + 은/JX'), ('들어맞았다.', None), ('나는', '나/NP + 는/JX'), ('우리', '우리/NP'), ('가족들이', '가족/NNG + 들/XSN + 이/JKS'), ('마음으로는', None), ('다들', None), ('원하면서도', None), ('선뜻', None)]
```

```
def defaultNNG(x):
    return([                                                    ])
```

```
print(defaultNNG(combining_tagger.tag(test_words[:10])))
```

```
[('내', '나/NP + 의/JKG'), ('계산은', '계산/NNG + 은/JX'), ('들어맞았다.', '들어맞았다./NNG'), ('나는', '나/NP + 는/JX'), ('우리', '우리/NP'), ('가족들이', '가족/NNG + 들/XSN + 이/JKS'), ('마음으로는', '마음으로는/NNG'), ('다들', '다들/NNG'), ('원하면서도', '원하면서도/NNG'), ('선뜻', '선뜻/NNG')]
```