

# 인공지능기초응용 II

## 5 주차 과제

인공지능응용

K2025029 금동환

## 목차

1. 파이썬 형태소 분석기(KoNLTK와 KoNLPy) 실습
2. "[실습] KoNLTKtest.zip"의 파이썬 코드 3개
3. [과제1]에서 준비한 텍스트 파일에 대해 파이썬/윈도용 형태소 분석기 실습

## 1. 파이썬 형태소 분석기(KoNLTK와 KoNLPy) 실습

```

1 from konlp.kma.klt2023 import klt2023
2 from konlp.tag import Okt
3 from konlp.tag import Kkma
4 from konlp.tag import Komoran
5
6 text = "안녕하세요. 국민대학교 소프트웨어융합대학원 인공지능응용 K2025029 김동환입니다."
7
8 print('----- klt2023')
9 klt2023 = klt2023()
10 print(klt2023.pos(text))
11 print(klt2023.morphs(text))
12 print(klt2023.nouns(text))
13 print('----- Okt')
14 okt = Okt()
15 print(okt.pos(text))
16 print(okt.morphs(text))
17 print(okt.nouns(text))
18 print('----- Kkma')
19 kkma = Kkma()
20 print(kkma.pos(text))
21 print(kkma.morphs(text))
22 print(kkma.nouns(text))
23 print('----- Komoran')
24 komoran = Komoran()
25 print(komoran.pos(text))
26 print(komoran.morphs(text))
27 print(komoran.nouns(text))
28

```

Run console output:

```

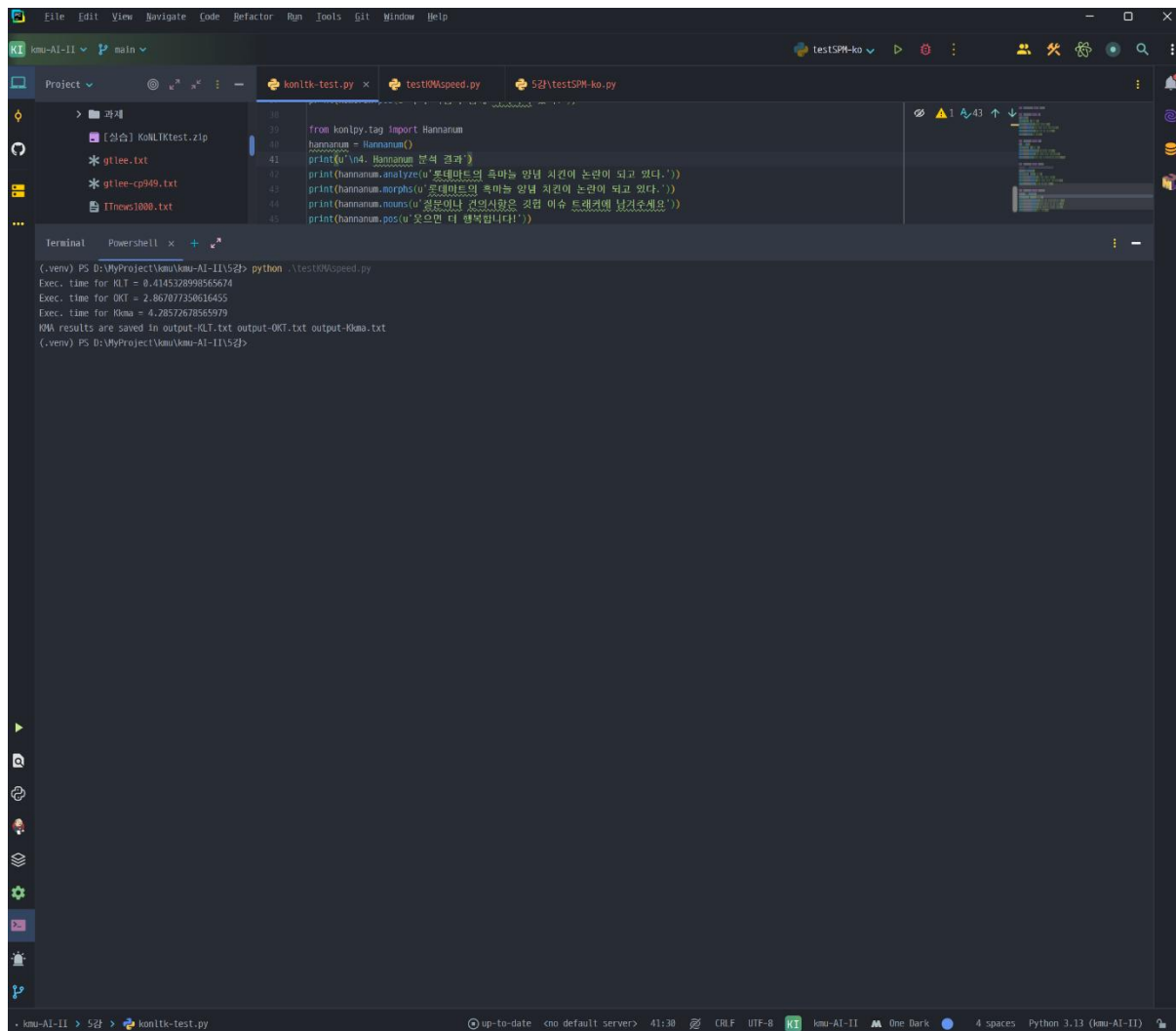
D:\MyProject\kmu\kmu-AI-11\venv\Scripts\python.exe D:\MyProject\kmu\kmu-AI-11\5강\과제\1.py
----- klt2023
['안녕/AF', '국민대학교/C', '소프트웨어융합대학원/K', '인공지능응용/C', 'K2025029/K', '김동환/K']
['안녕', '국민대학교', '소프트웨어융합대학원', '인공지능응용', 'K2025029', '김동환']
['안녕', '국민대학교', '소프트웨어융합대학원', '인공지능응용', '김동환']
----- Okt
[('안녕하세요', 'Adjective'), ('.', 'Punctuation'), ('국민', 'Noun'), ('대학교', 'Noun'), ('소프트웨어', 'Noun'), ('융합', 'Noun'), ('대학원', 'Noun'), ('인공', 'Noun'), ('지능', 'Noun'), ('응용', 'Noun'), ('K', 'Alph
[('안녕하세요', '.'), ('국민', '대학교', '소프트웨어', '융합', '대학원', '인공', '지능', '응용', 'K', '2025029', '금', '동환', '이', '입니다', '.')]
[('국민', '대학교', '소프트웨어', '융합', '대학원', '인공', '지능', '응용', '금동', '환', '이', '.')]
----- Kkma
[('안녕', 'NG'), ('하', 'XSV'), ('세요', 'EFN'), ('.', 'SF'), ('국민대학교', 'NG'), ('소프트웨어', 'NG'), ('융합', 'NG'), ('대학원', 'NG'), ('인공지능', 'NG'), ('응용', 'NG'), ('K', 'OL'), ('2025029', 'NR'), ('
[('안녕', '하', '세요', '.'), ('국민대학교', '소프트웨어', '융합', '대학원', '인공지능', '응용', 'K', '2025029', '금', '동환', '이', '입니다', '.')]
[('국민', '대학교', '소프트웨어', '융합', '대학원', '인공', '지능', '응용', '2025029', '금', '김동환', '동환')]
----- Komoran
[('안녕하세요', 'NMP'), ('.', 'SF'), ('국민대학교', 'NMP'), ('소프트웨어', 'NG'), ('융합', 'NG'), ('대학원', 'NMP'), ('인공지능', 'NMP'), ('응용', 'NMP'), ('K', 'SL'), ('2025029', 'SN'), ('금', 'NG'), ('동환', 'NMP
[('안녕하세요', '.'), ('국민대학교', '소프트웨어', '융합', '대학원', '인공지능', '응용', 'K', '2025029', '금', '동환', '이', '입니다', '.')]
[('안녕하세요', '국민대학교', '소프트웨어', '융합', '대학원', '인공지능', '응용', '금', '동환')]
Process finished with exit code 0

```



## 2. "[실습] KoNLTKtest.zip"의 파이썬 코드 3개

### A. testKMAAspeed.py



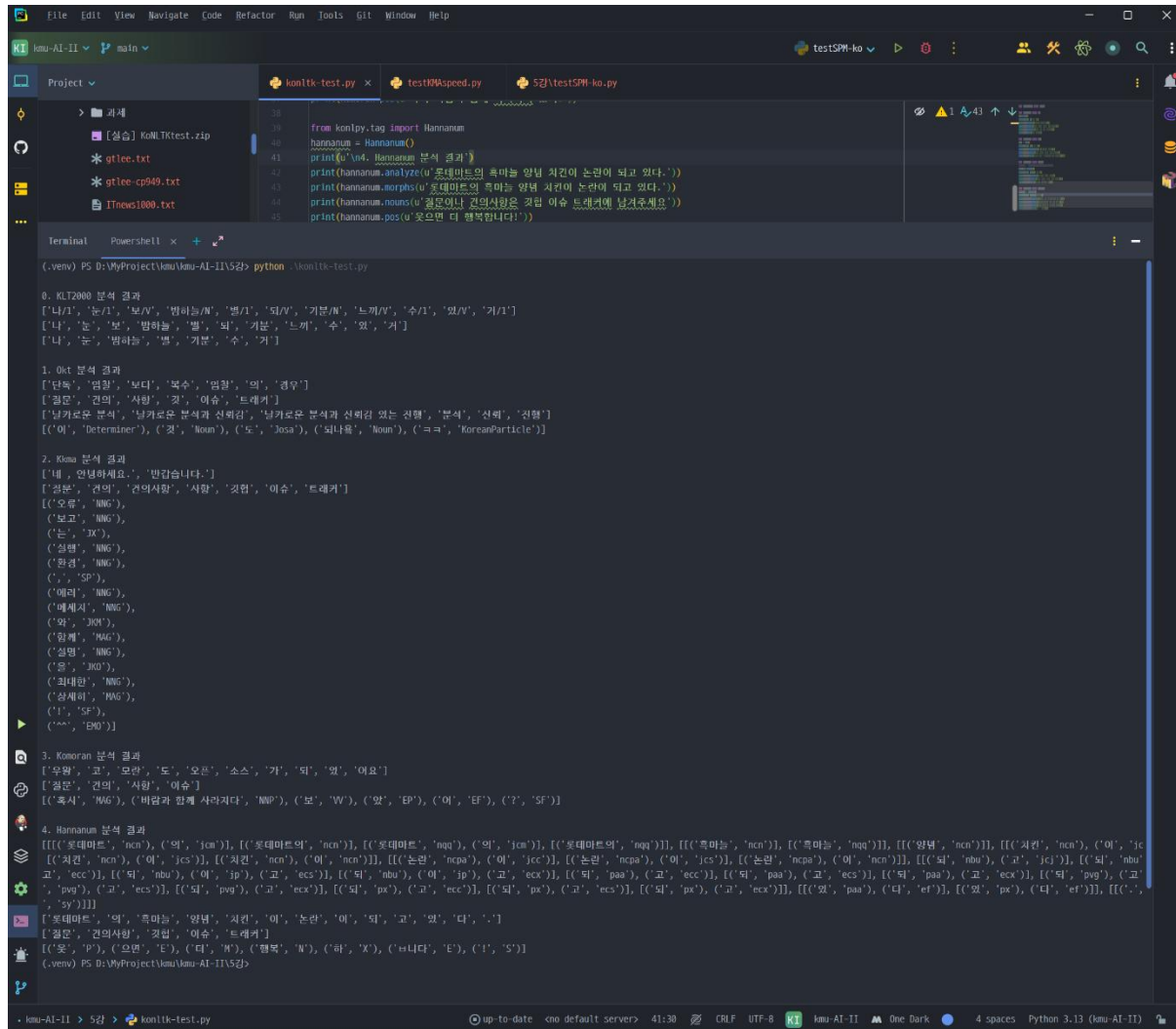
The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'kmu-AI-II' with a sub-project 'main'. Inside 'main', there is a folder '[실습] KoNLTKtest.zip' containing files 'gliee.txt', 'gliee-cp949.txt', and 'ITnews1000.txt'.
- Code Editor:** Displays the contents of 'testKMAAspeed.py'. The code imports 'Hannanum' from 'konlpy.tag', creates an instance, and performs morphological analysis on a sentence about a scandal involving a company's chairman.
- Terminal:** Shows the execution output of the script. It reports execution times for KLT, OKT, and KMA, and confirms that the results are saved in 'output-KLT.txt' and 'output-KMA.txt'.

```
from konlpy.tag import Hannanum
hannanum = Hannanum()
print(u'\n4. Hannanum 분석 결과')
print(hannanum.analyze(u'롯데마트의 촉마늘 양념 치킨이 논란이 되고 있다.'))
print(hannanum.morphs(u'롯데마트의 촉마늘 양념 치킨이 논란이 되고 있다.'))
print(hannanum.nouns(u'장문이나 긴문서장은 컷집 이후 트랙커에 넣겨주세요'))
print(hannanum.pos(u'웃으면 더 행복합니다!'))
```

```
(.venv) PS D:\MyProject\kmu\kmu-AI-II\52> python .\testKMAAspeed.py
Exec. time for KLT = 0.4145128998565674
Exec. time for OKT = 2.867077358616455
Exec. time for KMA = 4.28572678565979
KMA results are saved in output-KLT.txt output-OKT.txt output-KMA.txt
(.venv) PS D:\MyProject\kmu\kmu-AI-II\52>
```

B. konltk-test.py



## C. testSPM-ko.py

The screenshot shows a VS Code editor with the file `testSPM-ko.py` open. The code defines a `SentencePieceProcessor` and a `SentencePieceTrainer` to train a model. The terminal output shows the command being run and the training process details.

```

1 import sentencepiece as spm
2 sp = spm.SentencePieceProcessor()
3
4 spm.SentencePieceTrainer.Train('--input=Inews1000.txt --model_prefix=Inews --vocab_size=8000')
5 #spm.SentencePieceTrainer.Train('--input=gt100.txt --model_prefix=gt100 --vocab_size=16000')
6 #spm.SentencePieceTrainer.Train('--input=ko_wiki_text.txt --model_prefix=koiki --vocab_size=64000')
7
8 sp.load('Inews.model') Unresolved attribute reference 'load' for class 'SentencePieceProcessor'

```

```

(.venv) PS D:\MyProject\kmu-AI-IT\52J> python .\testSPM-ko.py
sentencepiece.trainer.cc(178) LOG(INFO) Running command: --input=Inews1000.txt --model_prefix=Inews --vocab_size=8000
sentencepiece.trainer.cc(78) LOG(INFO) Starts training with :
trainer_spec {
  input: Inews1000.txt
  input_format:
  model_prefix: Inews
  model_type: UNIGRAM
  vocab_size: 8000
  self_test_sample_size: 0
  character_coverage: 0.9995
  input_sentence_size: 0
  shuffle_input_sentence: 1
  seed_sentencepiece_size: 1000000
  shrinking_factor: 0.75
  max_sentence_length: 4192
  num_threads: 16
  num_sub_iterations: 2
  max_sentencepiece_length: 16
  split_by_unicode_script: 1
  split_by_numbers: 1
  split_by_whitespace: 1
  split_digits: 0
  pretokenization_delimiter:
  treat_whitespace_as_suffix: 0
  allow_whitespace_only_pieces: 0
  required_chars:
  byte_fallback: 0
  vocabulary_output_piece_score: 1
  train_extremely_large_corpus: 0
  seed_sentencepieces_file:
  hard_vocab_limit: 1
  use_all_vocab: 0
  unk_id: 0
  bos_id: 1
  eos_id: 2
  pad_id: -1
  unk_piece: <unk>
  bos_piece: <S>
  eos_piece: </S>
  pad_piece: <pad>
  unk_surface: ??
  enable_differential_privacy: 0
  differential_privacy_noise_level: 0
  differential_privacy_clipping_threshold: 0
}
normalizer_spec {
  name: nmt_nfkc
  add_dummy_prefix: 1
  remove_extra_whitespace: 1
}

```

```

1 import sentencepiece as spm
2 sp = spm.SentencePieceProcessor()
3
4 spm.SentencePieceTrainer.Train('--input=Inews1000.txt --model_prefix=Inews --vocab_size=8000')
5 #spm.SentencePieceTrainer.Train('--input=gt100.txt --model_prefix=gt100 --vocab_size=16000')
6 #spm.SentencePieceTrainer.Train('--input=ko_wiki_text.txt --model_prefix=kwiki --vocab_size=64000')
7
8 sp.load('Inews.model')

```

```

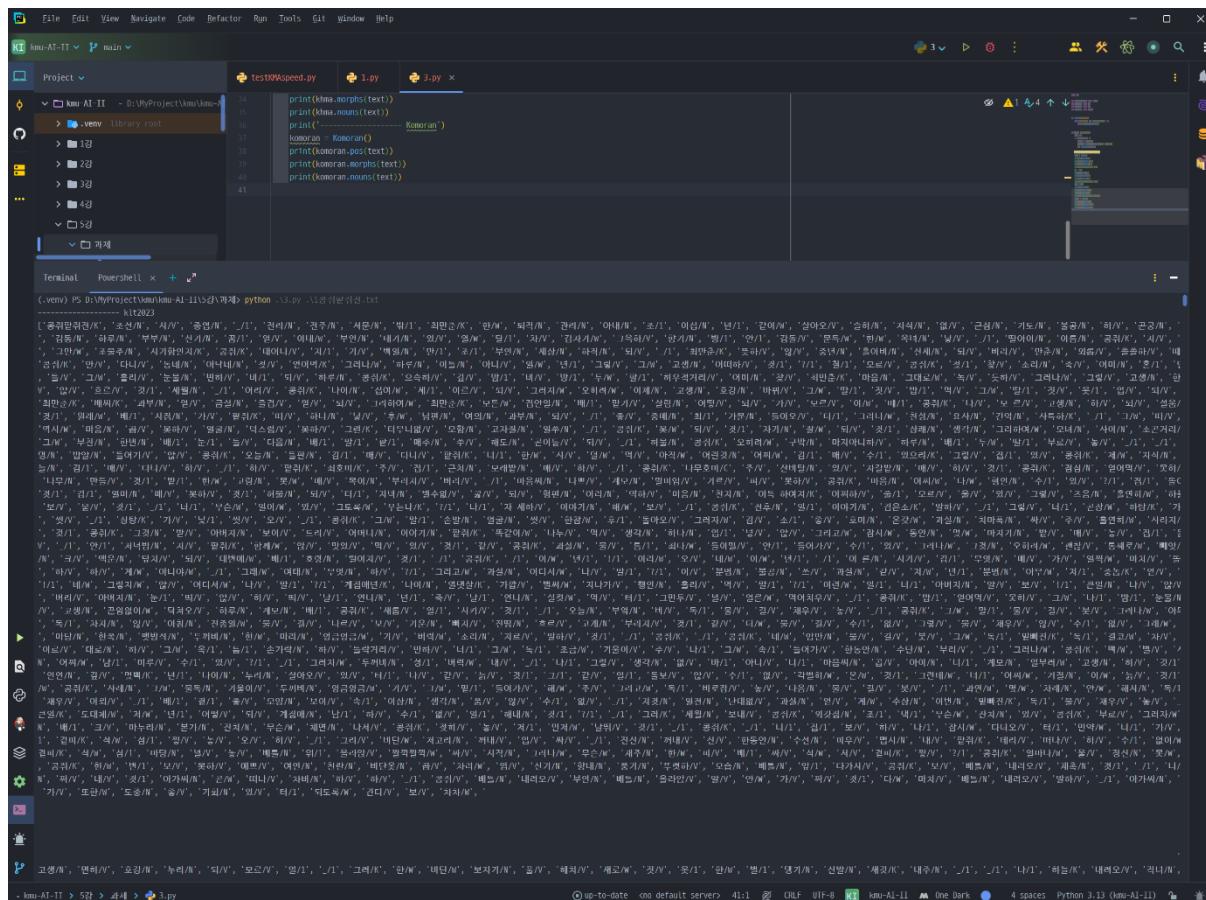
}
normalizer_spec {
  name: nmt_ofnc
  add_dummy_prefix: 1
  remove_extra_whitespace: 1
  escape_whitespace: 1
  normalization_rule_tsv:
}
denormalizer_spec {
  trainer_interface.cc(353) LOG(INFO) SentenceIterator is not specified. Using MultiFileSentenceIterator.
  trainer_interface.cc(185) LOG(INFO) Loading corpus: Inews1000.txt
  trainer_interface.cc(409) LOG(INFO) Loaded all 13708 sentences
  trainer_interface.cc(425) LOG(INFO) Adding meta_piece: <unk>
  trainer_interface.cc(425) LOG(INFO) Adding meta_piece: <eos>
  trainer_interface.cc(425) LOG(INFO) Adding meta_piece: </s>
  trainer_interface.cc(430) LOG(INFO) Normalizing Sentences...
  trainer_interface.cc(539) LOG(INFO) all chars count=1240970
  trainer_interface.cc(550) LOG(INFO) Done: 99.55% characters are covered.
  trainer_interface.cc(560) LOG(INFO) Alphabet size=1400
  trainer_interface.cc(561) LOG(INFO) Final character coverage=0.9995
  trainer_interface.cc(592) LOG(INFO) Done! preprocessed 11990 sentences.
  unigram_model_trainer.cc(265) LOG(INFO) Making suffix array...
  unigram_model_trainer.cc(269) LOG(INFO) Extracting frequent sub strings... node_num=478614
  unigram_model_trainer.cc(312) LOG(INFO) Initialized 70556 seed sentencepieces
  trainer_interface.cc(598) LOG(INFO) Tokenizing input sentences with whitespace: 11990
  trainer_interface.cc(609) LOG(INFO) Done! 90198
  unigram_model_trainer.cc(602) LOG(INFO) Using 90198 sentences for EM training
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=0 size=37902 obj=16.3766 num_tokens=221299 num_tokens/piece=5.83872
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=1 size=35037 obj=14.9972 num_tokens=232472 num_tokens/piece=6.34963
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=0 size=26274 obj=15.2302 num_tokens=232880 num_tokens/piece=8.86352
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=1 size=26262 obj=15.1597 num_tokens=232976 num_tokens/piece=8.87122
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=0 size=19696 obj=15.5268 num_tokens=246868 num_tokens/piece=12.5339
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=1 size=19694 obj=15.4518 num_tokens=246874 num_tokens/piece=12.5355
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=0 size=14769 obj=15.8639 num_tokens=261438 num_tokens/piece=17.7018
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=1 size=14769 obj=15.7748 num_tokens=261463 num_tokens/piece=17.7035
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=0 size=11076 obj=16.2608 num_tokens=277148 num_tokens/piece=25.0224
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=1 size=11076 obj=16.1676 num_tokens=277147 num_tokens/piece=25.0223
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=0 size=8000 obj=16.6585 num_tokens=289807 num_tokens/piece=32.9326
  unigram_model_trainer.cc(618) LOG(INFO) EM sub_iter=1 size=8000 obj=16.5011 num_tokens=289806 num_tokens/piece=32.9439
  trainer_interface.cc(607) LOG(INFO) Saving models: Inews.model
  trainer_interface.cc(609) LOG(INFO) Saving vocabs: Inews.vocab
  [
    '한글', '인', '권', '부', '는', '오른', '20', '열부터', '의', '배스', '의', '권', '월', '등', '내', '중', '보통', '과', '의', '의', '마브', '등', '내원', '사실', '연', '의', '개방', '원', '의', '국', '해석', '마', '스',
    '크', '의', '작', '물', '의무', '를', '해', '체', '한다고', '보였', '습니다', '']
  [797, 228, 130, 124, 7, 453, 442, 1015, 3, 3274, 24, 130, 697, 25, 257, 171, 3305, 23, 3, 4594, 25, 715, 2405, 428, 6, 2184, 189, 490, 328, 32, 570, 54, 407, 3, 1989, 106, 3790, 8, 513, 147, 1619, 4296, 2714, 4]
  [
    '관계개요', '의', '것', '보', '물', '의', '원인', '생성', '형', '연결', 'C', 'A', 'I', '의', '것', 'G', 'P', 'T', '개방', '사', '인', '오픈', 'A', 'I', '가', '14', '원', 'C', '원리', '시간',
    '의', '다육', '감각', '해', '신', '체로', '운', '연결', '물', '인', 'G', 'P', 'T', '의', '4', '물', '개', '했다', '의', 'A', 'I', '개', '을', '의', '물', '고', '구로', '물', '한', '경', '에', '트',
    '한', '마이크로소프트', 'C', 'MS', '의', '는', '국', '사', '관', '에', 'G', 'P', 'T', '의', '4', '물', '개', '했다', '의', 'A', 'I', '개', '을', '의', '물', '고', '구로', '물', '한', '경', '에', '트',
    [4227, 30, 0, 2500, 4902, 33, 5, 3, 7024, 3081, 189, 4863, 17, 406, 491, 18, 3, 0, 650, 461, 621, 92, 62, 21, 1596, 406, 491, 11, 894, 36, 17, 5040, 720, 18, 557, 892, 40, 320, 216, 89, 4863, 2791, 21, 2256, 461, 621, 206,
    108, 8, 459, 44, 4, 1827, 491, 120, 5, 3, 7971, 14, 2051, 359, 15, 434, 10, 1879, 15, 468, 17, 503, 18, 7, 4104, 591, 376, 10, 2256, 461, 621, 206, 108, 1700, 8, 3, 2783, 13, 2051, 6, 2906, 2631, 53, 41, 10, 1866, 4]
  ]
  (.venv) PS D:\MyProject\kmu-AI-II>

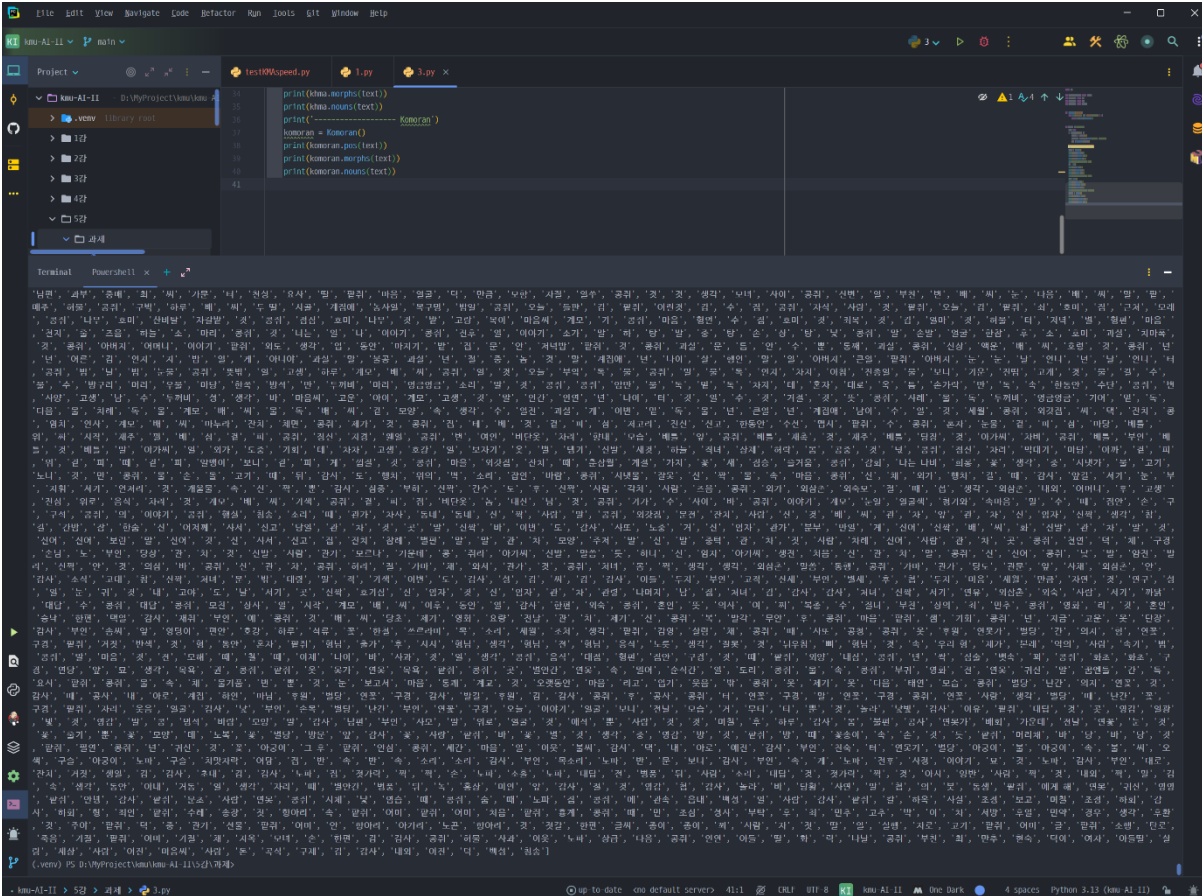
```

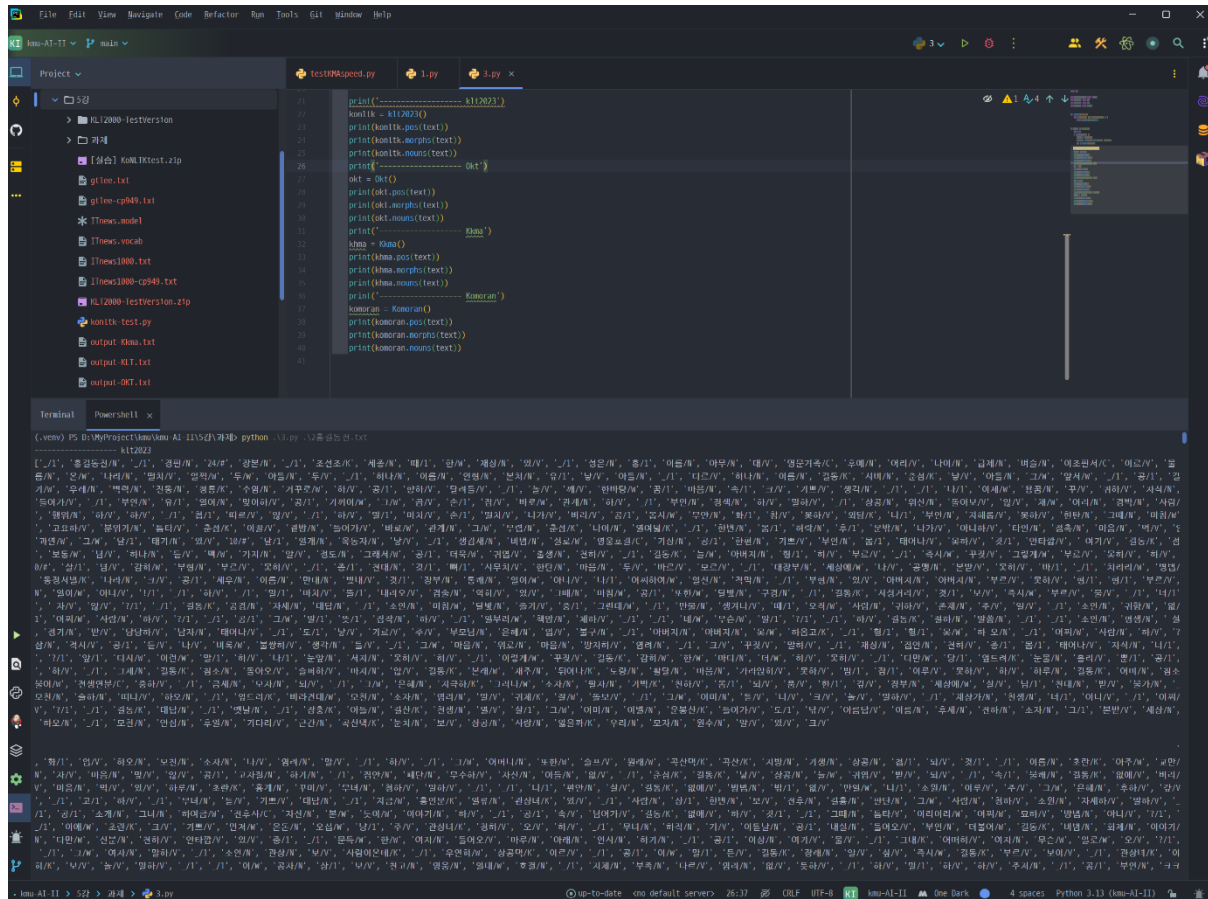


3. [과제1]에서 준비한 텍스트 파일에 대해 파이썬/윈도용 형태소 분석기 실습

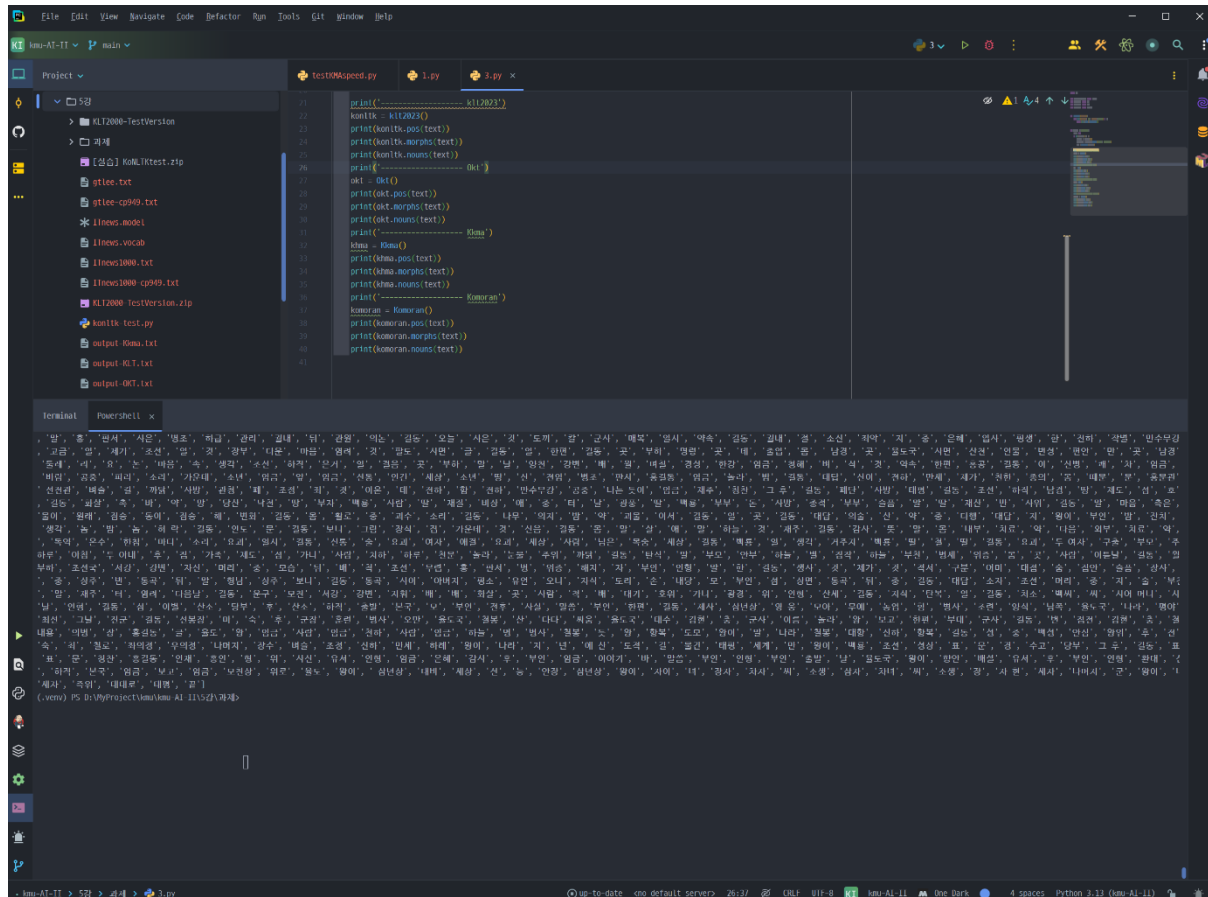
A. 파이썬 형태소 분석기(KoNLPy, KoNLTK) 실습 내용 및 실습화면 스샷







인공지능응용 K2025029 금동환



B. index2018.exe를 이용하여 윈도 cmd창에서 실습 내용 및 실습화면 스샷

The screenshot shows a Visual Studio Code editor with a Python script named `test1090speed.py` and its output in a terminal window. The script uses the `konlpy` library to process a text string. The terminal output shows the results of the processing, including a list of words and their frequencies.

```

1 text = "언론의세음: 국민대리교 소프트웨어융합대학원 인공지능응용 K2025029 김동환입니다."
2
3 print(----- h[12021])
4
5 konlpy.kl[2021]
6 print(konlpy.pos(text))
7 print(konlpy.morphs(text))
8 print(konlpy.nouns(text))
9 print(----- (kt))
10
11 okt = Okt()
12 print(oct.pos(text))
13 print(oct.morphs(text))

```

The terminal output shows the results of the processing, including a list of words and their frequencies:

```

(.venv) PS D:\Project\kmu\kmu-AI-11\5\J\VL12000-test\versionLXk> .\index2018.exe -i c:\p09.txt
-----
No: Freq Score Term
-----
1: 73 1000 공위
2: 23 415 발위
3: 66 394 고
4: 28 278 간사
5: 9 161 그라코
6: 15 150 그라나
7: 7 142 감감사
8: 10 179 그라사
9: 21 126 한
10: 10 123 관자
11: 8 122 세만준
12: 7 104 그연테
13: 17 163 사명
14: 13 65 왕연
15: 8 92 완리
16: 20 90 생리
17: 8 85 게모
18: 22 83 미
19: 6 75 회준
20: 10 75 미문
21: 9 70 노사
22: 8 67 선택
23: 7 66 예미
24: 7 66 미대거
25: 6 65 미열계
26: 8 65 소려
27: 8 63 연보
28: 6 61 남고
29: 6 59 연어
30: 3 50 만남미
31: 9 56 고연
32: 7 51 배두
33: 3 50 함아려
34: 5 49 목삼준
35: 7 48 연남
36: 3 48 연장염
37: 6 46 자거
38: 5 46 연호
39: 4 44 미금역
40: 3 44 연모기
41: 5 44 연남
42: 6 44 관성
43: 6 42 관자

```

```

1 text = "인공지능 응용, 국어재미교 소프트웨어융합대학원 인공지능융합 K2025029 김동환입니다."
2
3 print('----- k112023')
4
5 nltk.tokenize()
6
7 print(nltk.tokenize_pos(text))
8
9 print(nltk.tokenize_pos(text))
10
11 print(nltk.tokenize_pos(text))
12
13 print('----- k11')
14
15 okt = Okt()
16
17 print(okt.tokenize(text))
18
19 print(okt.tokenize(text))

```

No	Freq	Score	Term	Loc1	Loc2	Loc3	Loc4	Loc5
1:	84	1000	인공					
2:	102	547	*					
3:	22	441	응용					
4:	58	265	교					
5:	37	325	사					
6:	36	274	어					
7:	29	241	학					
8:	34	202	원					
9:	14	201	대					
10:	37	184	학					
11:	20	182	도					
12:	18	179	교					
13:	16	179	어					
14:	10	172	재					
15:	17	162	인					
16:	19	161	공					
17:	13	159	융					
18:	14	147	합					
19:	14	146	학					
20:	19	134	학					
21:	12	134	학					
22:	12	133	학					
23:	9	129	학					
24:	13	119	학					
25:	12	116	학					
26:	7	112	학					
27:	10	111	학					
28:	10	110	학					
29:	9	104	학					
30:	9	104	학					
31:	14	103	학					
32:	11	102	학					
33:	10	96	학					
34:	10	96	학					
35:	10	94	학					
36:	7	93	학					
37:	13	91	학					
38:	6	88	학					
39:	100	88	*					
40:	7	87	학					
41:	7	87	학					
42:	6	86	학					
43:	6	86	학					