텍스트 전처리 (Text Preprocessing)

- 텍스트를 자연어 처리를 위해 용도에 맞도록 사전에 표준화 하는 작업
- 텍스트 내 정보를 유지하고, 중복을 제거하여 분석 효율성을 높이기 위해 전처리를 수행

1. 토큰화(Tokenizing)

- 텍스트를 자연어 처리를 위해 분리 하는 것
- 토큰화는 단어별로 분리하는 "단어 토큰화(Word Tokenization)"와 문장별로 분리하는 "문장 토큰화(Sentence Tokenization)"로 구분
  (이후 실습에서는 단어 토큰화를 "토큰화"로 통일하여 칭하도록 한다)

```
[ ]  1 text = '''인생은 모두가 함께하는 여행이다. 매일매일 사는 동안 우리가 할 수 있는 건 최선을 다해 이 멋진 여행을 만끽하는 것이다.'''
```

```
[ ]  1 # 띄어쓰기로 토큰화
     2 print(text.split(' '))
```

```
['인생은', '모두가', '함께하는', '여행이다.', '매일매일', '사는', '동안', '우리가', '할', '수', '있는', '건', '최선을', '다해', '이', '멋진', '여행을', '만끽하는', '것이다.']
```

```
     1 # !pip install konlpy
```

```
[ ]  1 # 코모란
     2 from konlpy.tag import Komoran
     3 # 선언
     4 komoran= Komoran()
     5 # 토큰화 : morphs
     6 komoran_tokens = komoran.morphs(text)
     7 print(komoran_tokens)
```

```
['인생', '은', '모두', '가', '함께', '하', '는', '여행', '이', '다', '.', '매일', '매일', '살', '는', '동안', '우리', '가', '하', 'ㄹ', '수', '있', '는', '건', '최선', '을', '다', '하',
```

```
[ ]  1 # 한나눔
     2 from konlpy.tag import Hannanum
     3 hannanum= Hannanum()
     4 hannanum_tokens = hannanum.morphs(text)
     5 print(hannanum_tokens)
```

```
['인생', '은', '모두', '가', '함께하', '는', '여행', '이', '다', '.', '매일매일', '사', '는', '동안', '우리', '가', '하', 'ㄹ', '수', '있', '는', '거', '은', '최선', '을', '다하', '어', '이
```

```
[ ]  1 #Okt
     2 from konlpy.tag import Okt
     3 okt= Okt()
     4 okt_tokens = okt.morphs(text)
     5 print(okt_tokens)
```

```
['인생', '은', '모두', '가', '함께', '하는', '여행', '이다', '.', '매', '일', '매일', '사는', '동안', '우리', '가', '할', '수', '있는', '건', '최선', '을', '다해', '이', '멋진', '여행', '을
```

```
[ ]  1 # Kkma
     2 from konlpy.tag import Kkma
     3 kkma= Kkma()
     4 kkma_tokens = kkma.morphs(text)
     5 print(kkma_tokens)
```

```
['인생', '은', '모두', '가', '함께', '하', '는', '여행', '이', '다', '.', '매일', '매일', '살', '는', '동안', '우리', '가', '하', 'ㄹ', '수', '있', '는', '것', '은', '최선', '을', '다하',
```

2) 품사 부착(PoS Tagging)

- 각 토큰에 품사 정보를 추가
- 분석시에 불필요한 품사를 제거하거나 (예. 조사, 접속사 등) 필요한 품사를 필터링 하기 위해 사용

```
[ ]  1 # 코모란
     2 komoranTag = []
     3 for token in komoran_tokens:
     4     komoranTag += komoran.pos(token)
     5 print(komoranTag)
```

```
[('인생', 'NNG'), ('은', 'NNP'), ('모두', 'MAG'), ('가', 'VV'), ('아', 'EC'), ('함께', 'MAG'), ('하', 'NNG'), ('늘', 'VV'), ('ㄴ', 'ETM'), ('여행', 'NNG'), ('이', 'MM'), ('다',
```

```
[ ]  1 # 한나눔
     2 hannanumTag = []
     3 for token in hannanum_tokens:
     4     hannanumTag += hannanum.pos(token)
     5 print(hannanumTag)
```

```
[('인생', 'N'), ('은', 'N'), ('모두', 'M'), ('가', 'J'), ('함께하', 'P'), ('어', 'E'), ('늘', 'P'), ('ㄴ', 'E'), ('여행', 'N'), ('이', 'M'), ('다', 'M'), ('.', 'S'), ('매일매일',
```

```
[ ]  1 #Okt
     2 oktTag = []
     3 for token in okt_tokens:
     4     oktTag += okt.pos(token)
     5 print(oktTag)
```

```
[('인생', 'Noun'), ('은', 'Noun'), ('모두', 'Noun'), ('가', 'Verb'), ('함께', 'Adverb'), ('하는', 'Verb'), ('여행', 'Noun'), ('이다', 'Josa'), ('.', 'Punctuation'), ('매', 'No
```

```
     1 # Kkma
     2 kkmaTag = []
     3 for token in kkma_tokens:
     4     kkmaTag += kkma.pos(token)
     5 print(kkmaTag)
```

```
[('인생', 'NNG'), ('은', 'NNG'), ('모두', 'MAG'), ('가', 'NNG'), ('함께', 'MAG'), ('하', 'NNG'), ('늘', 'VA'), ('ㄴ', 'ETD'), ('여행', 'NNG'), ('이', 'NNG'), ('다', 'NNG'), ('
```

3) 불용어 처리 (Stopword)

- 자연어 처리를 위해 불필요한 요소를 제거하는 작업
- 불필요한 품사를 제거하는 작업과 불필요한 단어를 제거하는 작업으로 구성
- 불필요한 토큰을 제거함으로써 연산의 효율성을 높임

```
[ ]  1 #twitter
     2 # 최빈어 조회. 최빈어를 조회하여 불용어 제거 대상을 선정
     3 from collections import Counter
     4 Counter(oktTag).most_common()
```

```
[(('가', 'Verb'), 2),
 (('하는', 'Verb'), 2),
 (('여행', 'Noun'), 2),
 (('이다', 'Josa'), 2),
 (('.', 'Punctuation'), 2),
 (('을', 'Josa'), 2),
 (('인생', 'Noun'), 1),
 (('은', 'Noun'), 1),
 (('모두', 'Noun'), 1),
 (('함께', 'Adverb'), 1),
 (('매', 'Noun'), 1),
 (('일', 'Noun'), 1),
 (('매일', 'Noun'), 1),
 (('사는', 'Verb'), 1),
 (('동안', 'Noun'), 1),
 (('우리', 'Noun'), 1),
 (('할', 'Verb'), 1),
 (('수', 'Noun'), 1),
 (('있는', 'Adjective'), 1),
 (('건', 'Noun'), 1),
 (('최선', 'Noun'), 1),
 (('다해', 'Noun'), 1),
 (('이', 'Noun'), 1),
 (('멋진', 'Adjective'), 1),
 (('만끽', 'Noun'), 1),
 (('것', 'Noun'), 1)]
```

```
 1 #불용어 처리
 2 stopPos = ['Josa','Punctuation','Suffix','Foreign','Alpha','Number']
 3 stopWord = ['을','은','가']
 4
 5 word = []
 6 for tag in oktTag:
 7     if tag[1] not in stopPos:
 8         if tag[0] not in stopWord:
 9             word.append(tag[0])
10 print(word)
```

```
['인생', '모두', '함께', '하는', '여행', '매', '일', '매일', '사는', '동안', '우리', '할', '수', '있는', '건', '최선', '다해', '이', '멋진', '여행', '만끽', '하는', '것']
```

```
[ ]  1 oktTag
```

```
[('인생', 'Noun'),
 ('은', 'Noun'),
 ('모두', 'Noun'),
 ('가', 'Verb'),
 ('함께', 'Adverb'),
 ('하는', 'Verb'),
 ('여행', 'Noun'),
 ('이다', 'Josa'),
 ('.', 'Punctuation'),
 ('매', 'Noun'),
 ('일', 'Noun'),
 ('매일', 'Noun'),
 ('사는', 'Verb'),
 ('동안', 'Noun'),
 ('우리', 'Noun'),
 ('가', 'Verb'),
 ('할', 'Verb'),
 ('수', 'Noun'),
 ('있는', 'Adjective'),
 ('건', 'Noun'),
 ('최선', 'Noun'),
 ('을', 'Josa'),
 ('다해', 'Noun'),
 ('이', 'Noun'),
 ('멋진', 'Adjective'),
 ('여행', 'Noun'),
 ('을', 'Josa'),
 ('만끽', 'Noun'),
 ('하는', 'Verb'),
 ('것', 'Noun'),
 ('이다', 'Josa'),
 ('.', 'Punctuation')]
```