텍스트 빈도분석

- 목적: 문서에서 자주 언급되는 키워드나 주요 주제를 파악합니다.
- 방법: 문서 내에서 특정 키워드의 "강점", "약점")의 빈도를 계산하고, 가장 많이 등장하는 단어들을 시각화(예: 워드 클라우드)합니다.

In [ ]:
```
# nltk stopwords 다운로드 (최초 한 번만 필요)
# import nltk
# nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Out[1]:
```
True
```

한글 표시

In [ ]:
```
import matplotlib as mpl              # 기본 설정 만지는 용도
import matplotlib.pyplot as plt       # 그래프 그리는 용도
import matplotlib.font_manager as fm # 폰트 관련 용도
import numpy as np
```

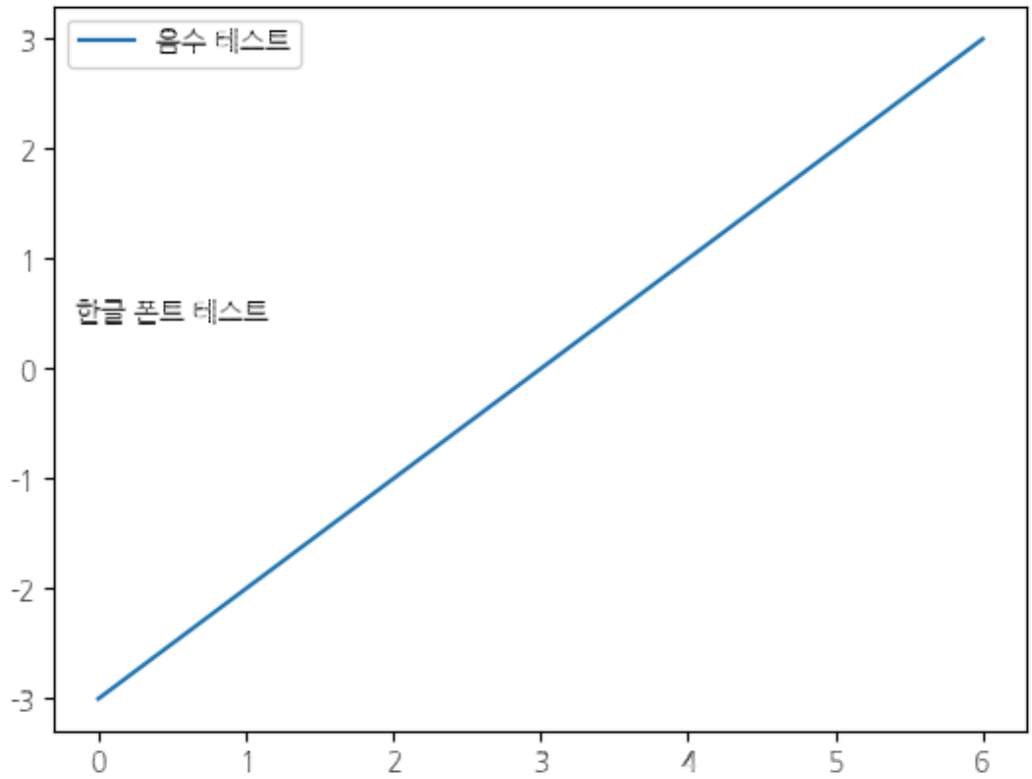
In [ ]:
```
### 나눔 고딕 설치
!apt-get update -qq   # 설치를 업데이트   -qq  : 로그를 최소한으로
!apt-get install fonts-nanum* -qq # 설치한다. fonts-nanum*   =>  ttf-nanum,
```

```
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to pro
vide it (sources.list entry misspelt?)
Selecting previously unselected package fonts-nanum.
(Reading database ... 123594 files and directories currently installed.)
Preparing to unpack .../fonts-nanum_20200506-1_all.deb ...
Unpacking fonts-nanum (20200506-1) ...
Selecting previously unselected package fonts-nanum-coding.
Preparing to unpack .../fonts-nanum-coding_2.5-3_all.deb ...
Unpacking fonts-nanum-coding (2.5-3) ...
Selecting previously unselected package fonts-nanum-eco.
Preparing to unpack .../fonts-nanum-eco_1.000-7_all.deb ...
Unpacking fonts-nanum-eco (1.000-7) ...
Selecting previously unselected package fonts-nanum-extra.
Preparing to unpack .../fonts-nanum-extra_20200506-1_all.deb ...
Unpacking fonts-nanum-extra (20200506-1) ...
Setting up fonts-nanum-extra (20200506-1) ...
Setting up fonts-nanum (20200506-1) ...
Setting up fonts-nanum-coding (2.5-3) ...
Setting up fonts-nanum-eco (1.000-7) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
```

In [ ]:

```
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
import matplotlib as mpl

# 폰트 파일 경로
path = '/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf'

# 폰트 프로퍼티 생성
font_prop = fm.FontProperties(fname=path, size=10)
font_name = font_prop.get_name()
print(font_name)  # NanumGothic Eco

# 폰트 매니저에 폰트 추가
fm.fontManager.addfont(path)

# matplotlib에 폰트 설정
plt.rc('font', family=font_name)

# 음수 표시되도록 설정
mpl.rcParams['axes.unicode_minus'] = False

# 예제 플롯
plt.figure()
plt.plot([-3, -2, -1, 0, 1, 2, 3], label='음수 테스트')
plt.text(0.5, 0.5, '한글 폰트 테스트', ha='center', va='center', fontpropert
plt.legend()
plt.show()
```

```
NanumGothic Eco
```



In [ ]:

```
### 빈도분석
```

In [ ]:

```
import os
from collections import Counter
import re
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
```

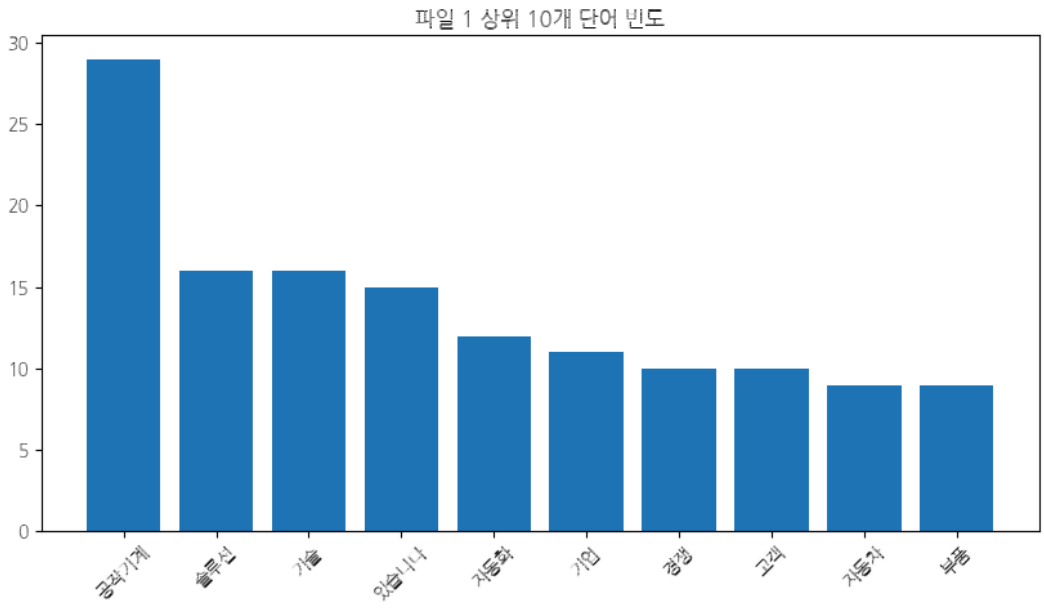
In [ ]:

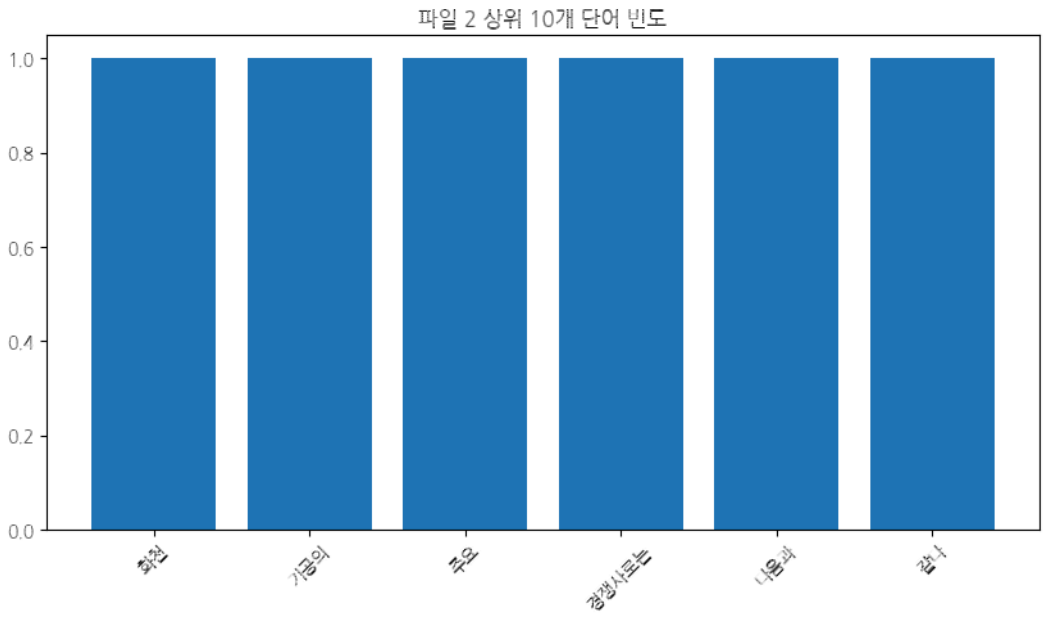
```
# 텍스트 파일 경로
file_paths = [
    "01_다른경쟁사와간단비교.txt",
    "02_기업리서치관련정리.txt",
    "03_생성AI분석.txt"
]

# 한국어 불용어 추가
# 수동으로 정의한 한국어 불용어 리스트
korean_stopwords = {
    '의', '가', '이', '은', '들', '는', '좀', '잘', '걍', '과', '도', '를',
    '자', '에', '와', '한', '하다', '에서', '것', '및', '위해', '그', '되다'
}

additional_stopwords = {'강점', '약점', '경쟁사'}  # 분석에 불필요한 단어 추
korean_stopwords.update(additional_stopwords)

# 파일별 텍스트 처리 및 단어 빈도 계산
def process_text(text):
    # 텍스트 전처리: 소문자화, 특수 문자 제거, 불용어 제거
    text = text.lower()
    text = re.sub(r'[^₩w₩s]', '', text)
    words = text.split()
    words = [word for word in words if word not in korean_stopwords and le
    return words

# 빈도 분석 결과 저장
word_frequencies = []

for file_path in file_paths:
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.read()
        words = process_text(text)
        word_freq = Counter(words)
        word_frequencies.append(word_freq)

# 파일별로 가장 자주 등장한 상위 10개 단어 출력
for i, freq in enumerate(word_frequencies):
    print(f"₩n파일 {i+1}의 상위 10개 단어:")
    print(freq.most_common(10))

# 시각화: 파일별 상위 10개 단어 빈도
for i, freq in enumerate(word_frequencies):
    common_words = freq.most_common(10)
    words, counts = zip(*common_words)

    plt.figure(figsize=(10, 5))
    plt.bar(words, counts)
    plt.title(f'파일 {i+1} 상위 10개 단어 빈도')
    plt.xticks(rotation=45)
    plt.show()
```

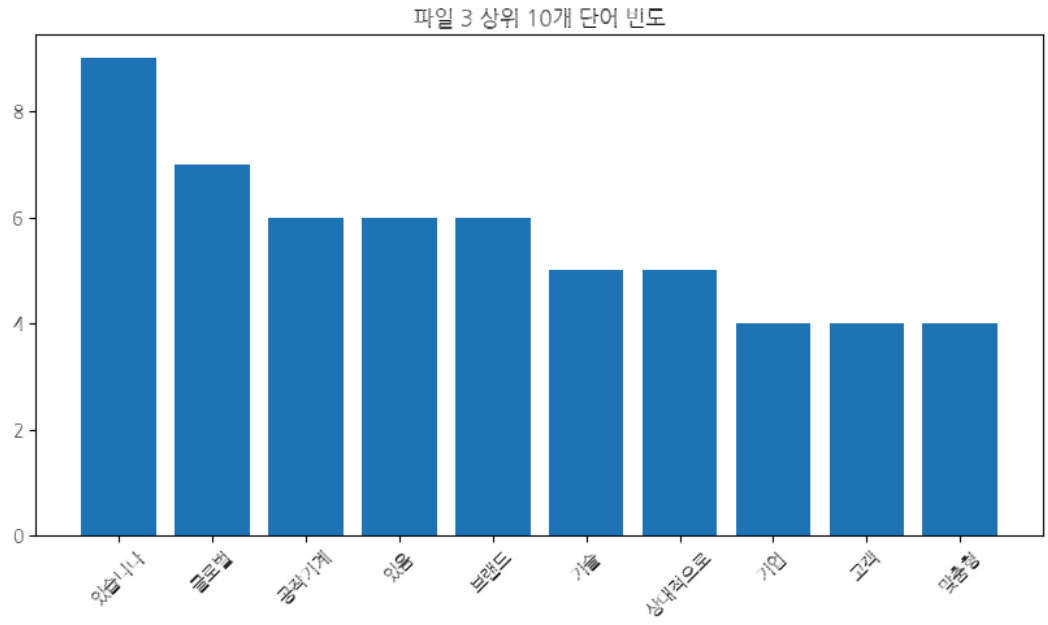
파일 1의 상위 10개 단어:
[('공작기계', 29), ('솔루션', 16), ('기술', 16), ('있습니다', 15), ('자동화', 12), ('기업', 11), ('경쟁', 10), ('고객', 10), ('자동차', 9), ('부품', 9)]

파일 2의 상위 10개 단어:
[('화천', 1), ('기공의', 1), ('주요', 1), ('경쟁사로는', 1), ('다음과', 1), ('같다', 1)]

파일 3의 상위 10개 단어:
[('있습니다', 9), ('글로벌', 7), ('공작기계', 6), ('있음', 6), ('브랜드', 6), ('기술', 5), ('상대적으로', 5), ('기업', 4), ('고객', 4), ('맞춤형', 4)]





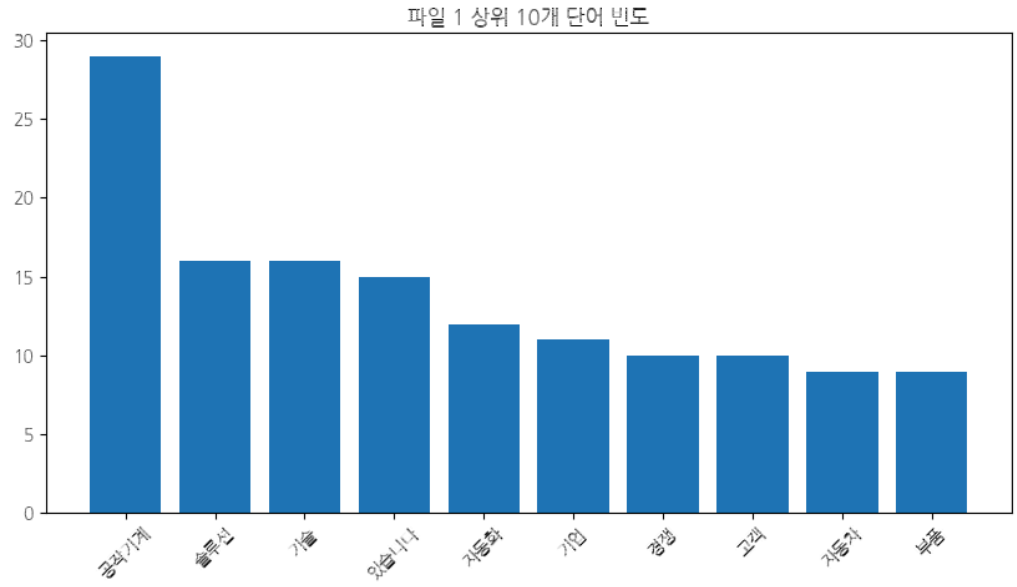


워드 클라우드

- 가장 많이 등장하는 단어들을 시각화

In [ ]:

```
# 시각화: 파일별 상위 10개 단어 빈도
for i, freq in enumerate(word_frequencies):
    common_words = freq.most_common(10)
    words, counts = zip(*common_words)

    plt.figure(figsize=(10, 5))
    plt.bar(words, counts)
    plt.title(f'파일 {i+1} 상위 10개 단어 빈도')
    plt.xticks(rotation=45)
    plt.show()

    # 워드 클라우드 생성
    wordcloud = WordCloud(
        width=800, height=400, background_color='white',
        font_path='/usr/share/fonts/truetype/nanum/NanumGothic.ttf'
    ).generate_from_frequencies(freq)

    # 워드 클라우드 시각화
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'파일 {i+1} 워드 클라우드')
    plt.show()
```



파일 1 워드 클라우드

In [ ]: