토픽모델링

- 목적: 문서 내에서 논의된 주요 주제를 자동으로 식별합니다.
- 방법: LDA(Latent Dirichlet Allocation)와 같은 주제 모델링 기법을 사용하여 텍스트에서 발견되는 주요 주제를 분석합니다. 이를 통해 회사와 관련된 다양한 논의의 축을 파악할 수 있습니다.

In [4]:
```
!pip install pyLDAvis
```

```
Requirement already satisfied: pyLDAvis in /usr/local/lib/python3.10/dist-packages (3.4.1)
Requirement already satisfied: numpy>=1.24.2 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.13.1)
Requirement already satisfied: pandas>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (2.1.4)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (3.1.4)
Requirement already satisfied: numexpr in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (2.10.1)
Requirement already satisfied: funcy in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (2.0)
Requirement already satisfied: scikit-learn>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (1.3.2)
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (4.3.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from pyLDAvis) (71.0.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.0.0->pyLDAvis) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.0.0->pyLDAvis) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.0.0->pyLDAvis) (2024.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->pyLDAvis) (3.5.0)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim->pyLDAvis) (7.0.4)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->pyLDAvis) (2.1.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=2.0.0->pyLDAvis) (1.16.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from smart-open>=1.8.1->gensim->pyLDAvis) (1.16.0)
```

In [7]:
```
## 처음 한번 다운로드 필요
import nltk
nltk.download('punkt')
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
ionWarning: `should_run_async` will not call `transform_cell` automatically
in the future. Please pass the result to `transformed_cell` argument and an
y exception that happen during thetransform in `preprocessing_exc_tuple` in
IPython 7.17 and above.
  and should_run_async(code)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

Out[7]:
```
True
```

In [8]:

```
import os
import re
from gensim import corpora
from gensim.models.ldamodel import LdaModel
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import pyLDAvis.gensim as gensimvis  # 수정된 부분
import pyLDAvis
import matplotlib.pyplot as plt

# 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'
}

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

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

# 파일 내용을 하나로 결합
combined_text = ""

for file_path in file_paths:
    with open(file_path, 'r', encoding='utf-8') as file:
        combined_text += file.read() + "₩n"

# 텍스트 전처리 및 토큰화
def preprocess(text):
    # 소문자 변환, 특수 문자 제거, 토큰화
    text = re.sub(r'[^₩w₩s]', '', text.lower())
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in korean_stopwords and
    return tokens

# 전처리된 문서 리스트 생성
documents = preprocess(combined_text)

# 단어 사전 생성
dictionary = corpora.Dictionary([documents])

# 코퍼스 생성 (문서를 BOW(Bag of Words)로 변환)
corpus = [dictionary.doc2bow(documents)]

# LDA 모델 생성
lda_model = LdaModel(corpus, num_topics=3, id2word=dictionary, passes=15)

# pyLDAvis를 이용한 시각화
vis_data = gensimvis.prepare(lda_model, corpus, dictionary)
pyLDAvis.display(vis_data)

# 필요 시, HTML 파일로 저장
pyLDAvis.save_html(vis_data, 'lda_visualization.html')
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

In [ ]: