자연어 처리 – AI 전문가 양성(기본) Day 3

KLUE Dataset – unsupervised multi-classification + textrank

INGLab

실습 소개

- **실습 목표**
  - 저번 실습에 이어 비지도학습방식으로 다중 분류 해보기
  - TextRank를 이용하여 요약 후 다중 분류 해보기

- **실습 내용**
  - KLUE 데이터셋 중 context, title 전처리
  - K-means를 이용해 다중 분류 및 평가
  - K-means 결과 시각화
  - TextRank를 사용하여 context 요약 후 k-means 적용

KLUE 한국어 데이터셋

- https://github.com/KLUE-benchmark/KLUE
- https://huggingface.co/datasets/klue

KLUE MRC dataset description

Jihyung Moon edited this page on 27 May · 1 revision

Name	Description
title	title of the context
source	document source of the context
news_category	category if the context is news
paragraphs	a list of contexts and question-and-answer pairs
context	target passage text
qas	a list of question-and-answer pairs
guid	a primary key of each question and answer pair
question	question text
answers	a list of answers
text	answer text
answer_start	start position of the answer
question_type	type of question
is_impossible	flag for unanswerable
plausible_answers	fake answers for type 3 question

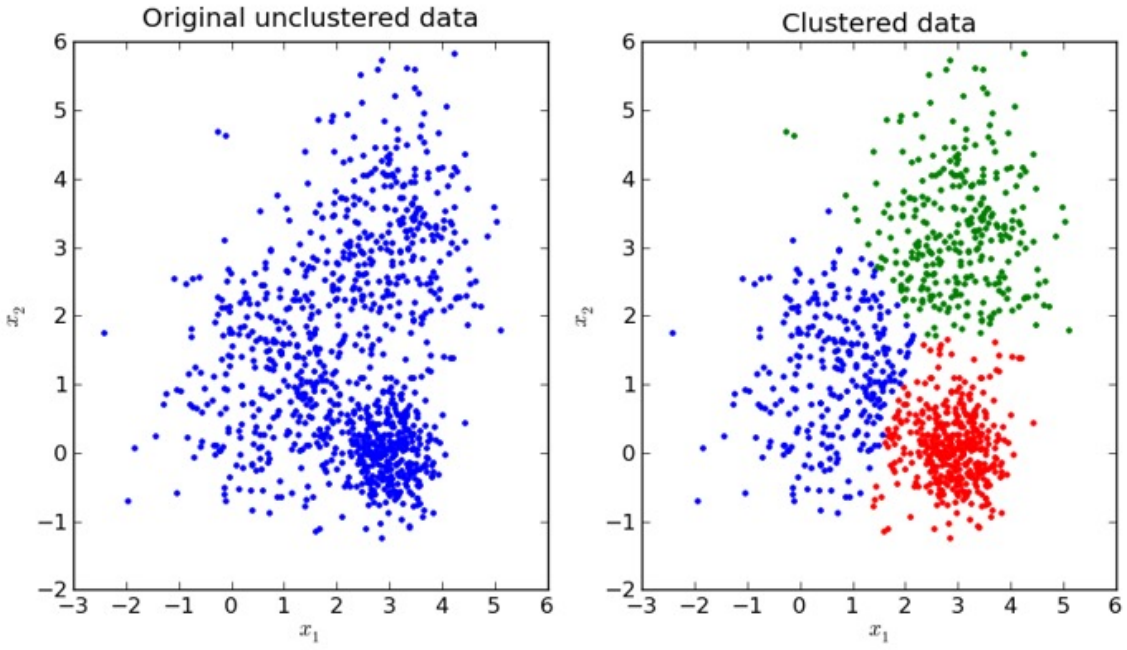
mrc

An example of 'train' looks as follows.

```
{'answers': {'answer_start': [478, 478], 'text': ['한 달가량', '한 달']},
 'context': '올여름 장마가 17일 제주도에서 시작됐다. 서울 등 중부지방은 예년보다
 'guid': 'klue-mrc-v1_train_12759',
 'is_impossible': False,
 'news_category': '종합',
 'question': '북태평양 기단과 오호츠크해 기단이 만나 국내에 머무르는 기간은?',
 'question_type': 1,
 'source': 'hankyung',
 'title': '제주도 장마 시작 … 중부는 이달 말부터'}
```

K-means

- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

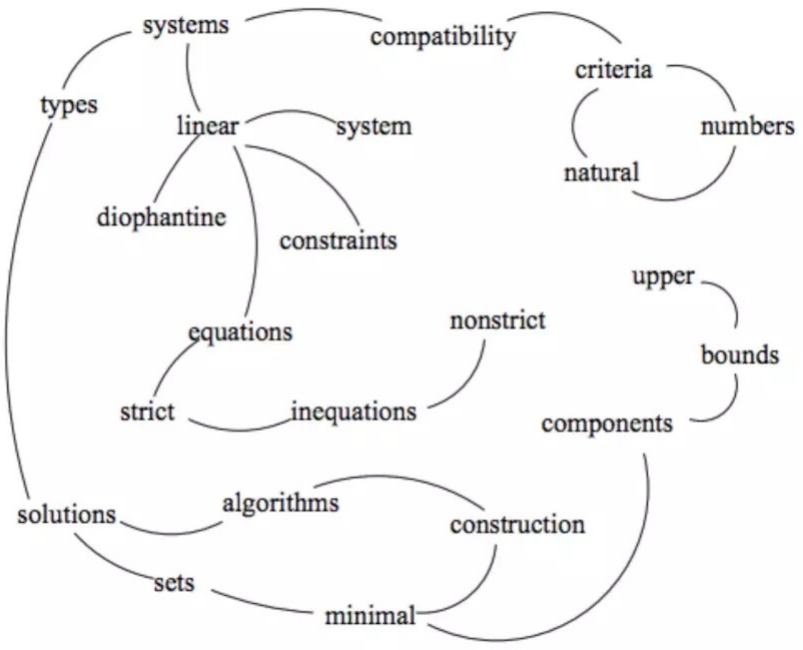


n_clusters	k-means로 나눌 클러스터의 개수 (기본값 = 8)
n_init	k-means의 시행 횟수 k-means를 여러 번 학습한 후 그 중 최선을 선택함 (기본값 = “10")
max_iter	k-means 의 최대 반복 횟수 (기본값 = 300)
⋮	⋮

TextRank

- https://github.com/lovit/textrank

- Text의 연결 관계를 같은 문서/문장에 가까이 등장하는지 여부로 중요도를 정의
- TextRank로 뽑히는 단어들은 문서/문장 내 중요도가 높다고 볼 수 있음 = 요약 단어
- 문서/문장 내의 rank가 높은 순으로 단어들을 구해낸 뒤 다른 문서와 비교해 유사도 측정도 할 수 있음



tokenize	토크나이저 함수 문장/문서가 들어왔을 때 단어로 분리할 수 있는 형태의 함수
min_count	모든 문서에서 최소 등장 횟수가 min_count 이상인 단어들만 사용 (기본값 = 2)
window	문서에 함께 있음 (연결됨) 을 나타내는 거리 -1 : 거리와 상관없이 같은 문서에 있으면 연결되었다 간주 (기본값 = -1)
min_cooccurrence	연결이 min_cooccurence 이상인 단어들만 연결된 것으로 취급 자잘한 연결들로 인해 연결 그래프 행렬이 필요 이상으로 dense해지는 것을
vocab_to_idx	따로 외부에서 단어 → 인덱스 매핑을 주고 싶을 때 전달하는 매개변수
⋮	⋮

COLAB - practice

- KLUE 데이터셋 ~ 전처리
- K-MEANS

```
import collections
from sklearn.cluster import KMeans
```

```
n_clusters = 5
```

```
kmeans = KMeans(n_clusters=n_clusters).fit(title_x)
```

```
title_preds = kmeans.predict(title_x)
```

K-means의 cluster와 뉴스 카테고리 매칭

```
def calc_cluster_counts(y_data, preds):
    clusters = {i: [] for i in range(n_clusters)}
    for i, (p, y) in enumerate(zip(preds, y_data)):
        clusters[p].append(y)
    cluster_counts = {k: collections.Counter(v) for k, v in clusters.items()}
    return cluster_counts
```

```
cluster_counts = calc_cluster_counts(title_y, title_preds)
```

```
cluster_mapping = {k: sorted(list(v.items()), key=lambda x: x[1])[-1][0] for k, v in cluster_counts.items()}
```

```
title_final_preds = [cluster_mapping[x] for x in title_preds]
```

COLAB - practice

- ## Visualization

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.decomposition import PCA
```

```
reduced_data = PCA(n_components=2).fit_transform(context_train_x.todense())
kmeans = KMeans(n_clusters=5).fit(reduced_data)
```

```
h = .002
x_min, x_max = reduced_data[:, 0].min(), reduced_data[:, 0].max()
y_min, y_max = reduced_data[:, 1].min(), reduced_data[:, 1].max()
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
```

```
preds = kmeans.predict(np.stack([xx.ravel(), yy.ravel()], axis=1))
preds = preds.reshape(xx.shape)
```

```
color_table = {x: f"C{i}" for i, x in enumerate(using_categories)}
colors = [color_table[x] for x in context_train_y]
```

```
plt.figure(num=1, figsize=(20, 10))
plt.clf()
plt.imshow(preds, interpolation="nearest", extent=(xx.min(), xx.max(), yy.min(), yy.max()), aspect="auto")
plt.scatter(reduced_data[:, 0], reduced_data[:, 1], c=colors)
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```

COLAB - practice

- **Textrank**

```
!pip install git+https://github.com/lovit/textrank.git
```

```
from textrank import KeywordSummarizer
from tqdm import tqdm_notebook
```

추가 전처리

```
tokenizer = Komoran()
def tokenize(text):
  words = tokenizer.pos(text)
  words = [w for w in words if w[1] in {'NNG', 'NNP', 'NNB', 'XR', 'VA', 'VV'}]
  assert len(words) > 0
  return words
```

```
summarizer = KeywordSummarizer(tokenize=tokenize, min_count=2, min_cooccurrence=1)
keywords = [summarizer.summarize([x], topk=20) for x in tqdm_notebook(context_processed)]
```