Coleridge Initiative - Show US the Data

최연석, 오서영

Overview

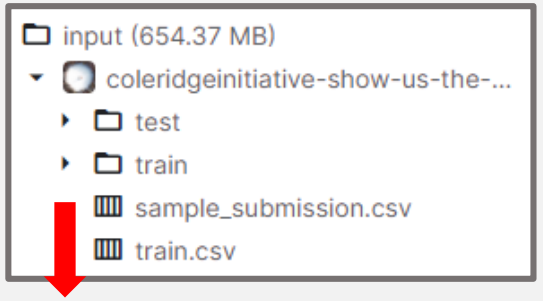
Introduction

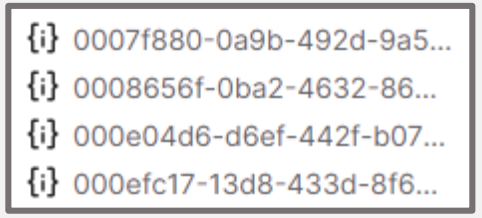
과학과 사회에 필요한 데이터에 대한 많은 정보는 출판물에 잠겨 있음
-> 기계 학습을 통해 연구 기사에 사용 된 단어, 기사에서 참조 된 데이터 사이의 연결 고리를 찾기
-> 자연어 처리 (NLP)를 사용하여 출판물에서 과학 데이터가 참조되는 방식을 자동으로 발견하기

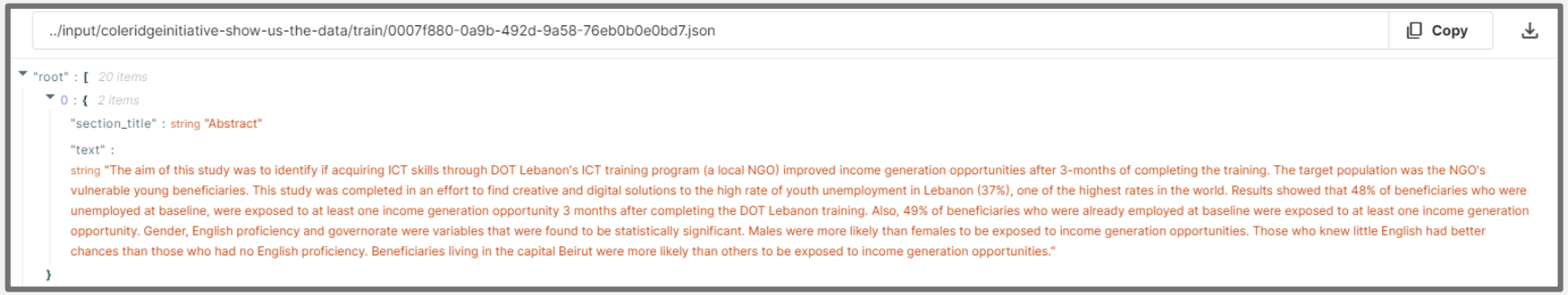
Goal

과학 출판물 내에서 데이터 세트에 대한 언급을 식별하는 것
-> 출판물에서 발췌 한 짧은 내용

1. Data Exploration and Visualization







Load Datasets

```
dataset_path = Path('../input/coleridgeinitiative-show-us-the-data')
```

```
train_df = pd.read_csv(dataset_path/'train.csv')
train_df.head()
```

	Id	pub_title	dataset_title	dataset_label	cleaned_label
0	d0fa7568-7d8e-4db9-870f-f9c6f668c17b	The Impact of Dual Enrollment on College Degre...	National Education Longitudinal Study	National Education Longitudinal Study	national education longitudinal study
1	2f26f645-3dec-485d-b68d-f013c9e05e60	Educational Attainment of High School Dropouts...	National Education Longitudinal Study	National Education Longitudinal Study	national education longitudinal study
2	c5d5cd2c-59de-4f29-bbb1-6a88c7b52f29	Differences in Outcomes for Female and Male St...	National Education Longitudinal Study	National Education Longitudinal Study	national education longitudinal study
3	5c9a3bc9-41ba-4574-ad71-e25c1442c8af	Stepping Stone and Option Value in a Model of ...	National Education Longitudinal Study	National Education Longitudinal Study	national education longitudinal study
4	c754dec7-c5a3-4337-9892-c02158475064	Parental Effort, School Resources, and Student...	National Education Longitudinal Study	National Education Longitudinal Study	national education longitudinal study

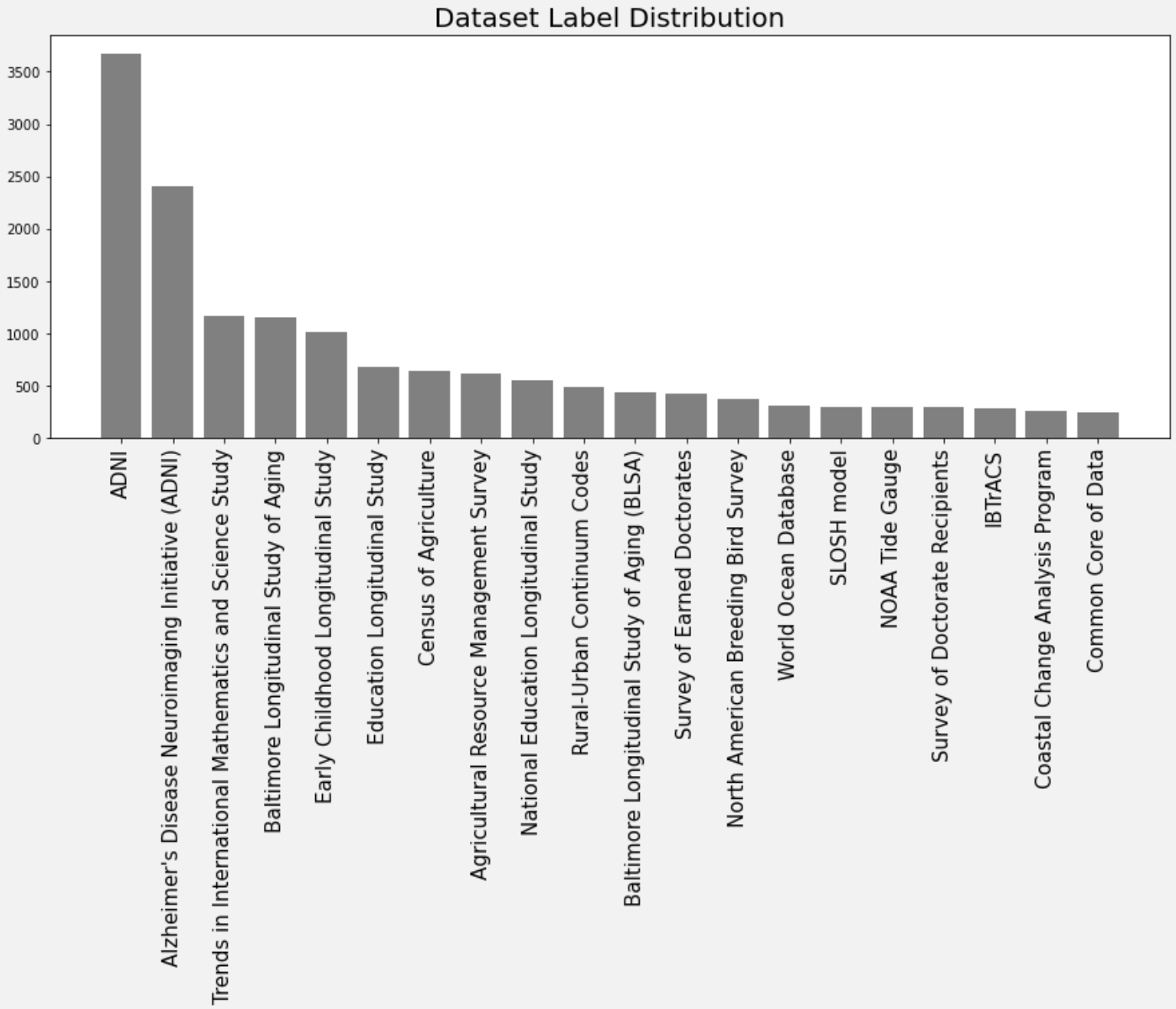
1. Data Exploration and Visualization

Count dataset titles

```
Counter(train_df.dataset_title)
```

```
Counter({'National Education Longitudinal Study': 550,
         'NOAA Tide Gauge': 441,
         'Sea, Lake, and Overland Surges from Hurricanes': 312,
         'Coastal Change Analysis Program': 326,
         'Aging Integrated Database (AGID)': 3,
         "Alzheimer's Disease Neuroimaging Initiative (ADNI)": 6144,
         'Baltimore Longitudinal Study of Aging (BLSA)': 1589,
         'Agricultural Resource Management Survey': 660,
         'Beginning Postsecondary Student': 461,
         "The National Institute on Aging Genetics of Alzheimer's Disease Data Storage Site (NIAGADS)": 22,
```

1. Data Exploration and Visualization



2. Word Cloud

```
title = train_df.dataset_title.unique()
n = 1 # 라벨 몇개까지 할건지
for i in range(n):
#     print(i,"th")
    ind = np.where(train_df.dataset_label == title[i])
    if i!=2 :
        txt = ''
        ind = np.array(ind)
        id = train_df.Id[ind[0,0]]
        json = pd.read_json(dataset_path/'train'/(id+'.json'))
        for j in range(len(json['text'])):
            txt += json['text'][j]
        print(txt)
```

```
This study used data from the National Education Lon
tudents on college degree attainment. The study also
college students versus students whose parents had a
ifferent amounts of dual enrollment course-taking an
Dual enrollment programs offer college-level learnin
ity to earn college credits for students while still
The intervention group in the study was comprised of
nt program while in high school (n = 880). The study
ts who also attended a postsecondary school but who
ograms allow high school students to take college co
ove college attainment, especially among low-income
abling students to accumulate college credits toward
The study reported program impacts on two outcomes:
mined for various subgroups of students, which are d
The study reported, and the WWC confirmed, that dual
e and (b) a bachelor's degree.
```

```
# clean text
txt = re.sub('[^a-zA-Z]', ' ', txt)
txt
```

2. Word Cloud

Lowercase, stopwords, lemmatization

```
lower_txt = txt.lower()
word = lower_txt.split()
print(len(word))
print(word[:20])
```

```
1771
['this', 'study', 'used', 'data', 'from', 'the',
f', 'dual', 'enrollment', 'programs', 'for']
```

```
word = [i for i in word if not i in stopwords.words('english')]
print(len(word))
print(word[:20])
```

```
1048
'nels', 'examine', 'effects', 'dual', 'enrollment', 'programs',
 'students', 'college', 'degree', 'attainment', 'study']
```

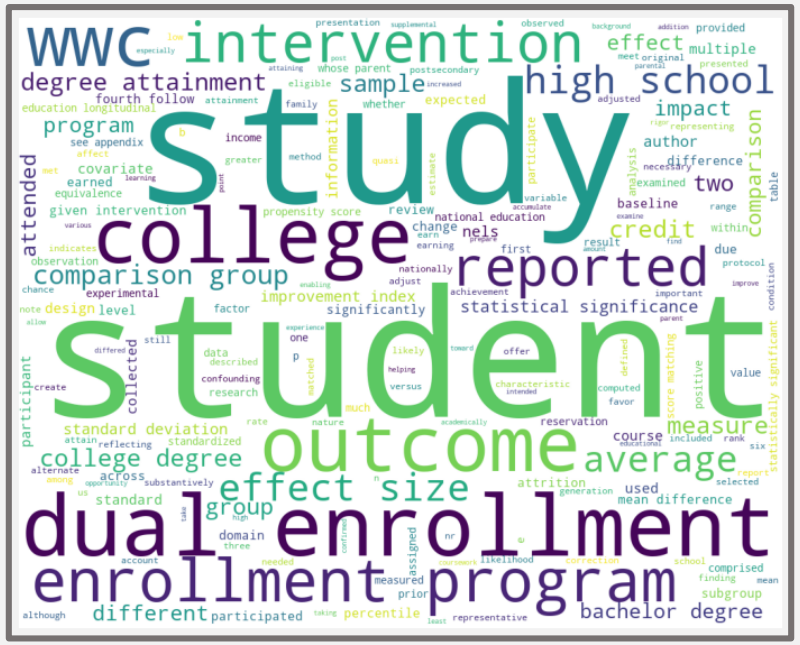
```
# Lemmatization
wordnet_lemmatizer = WordNetLemmatizer()
word = [ wordnet_lemmatizer.lemmatize(w) for w in word]
print(len(word))
print(word[:20])
```

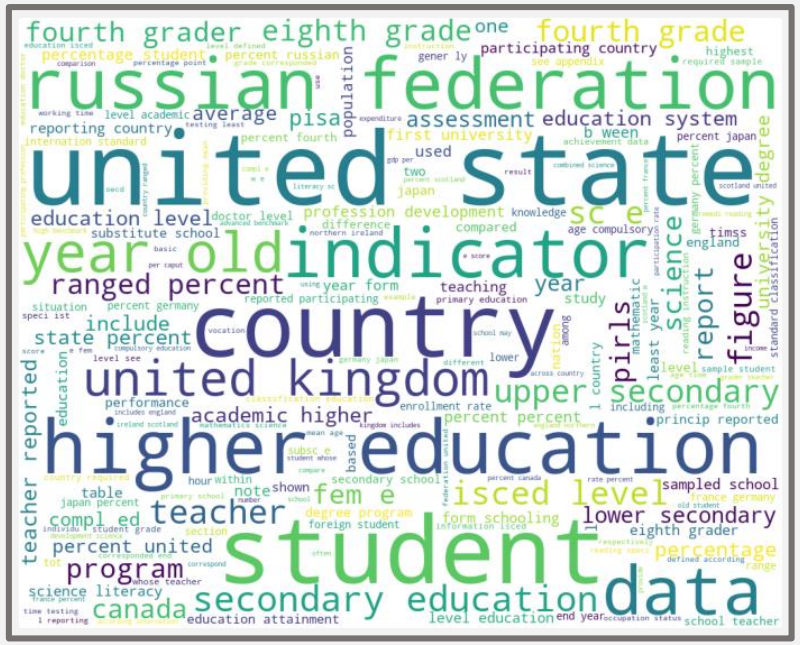
```
1048
'nels', 'examine', 'effect', 'dual', 'enrollment', 'program',
 'student', 'college', 'degree', 'attainment', 'study']
```

2. Word Cloud

Word cloud

```
def displayWordCloud(data = None, backgroundcolor = 'white', width=1000, height=800 ):
    wordcloud = WordCloud(stopwords = STOPWORDS,
                          background_color = backgroundcolor,
                          width = width, height = height).generate(data)
```





218	Seoyoung & YeonSeok	0.695	6	1d

3. LDA

JSON -> Pandas Dataframe

```python
# Path to the JSON files
train_files = glob.glob("../input/coleridgeinitiative-show-us-the-data/train/*.json")
test_files = glob.glob("../input/coleridgeinitiative-show-us-the-data/test/*.json")

# load json file using pandas
df_train = pd.DataFrame()
for count,ele in enumerate(train_files,len(train_files)):
    df_train = pd.concat([df_train, pd.read_json(ele)])

df_train.to_csv("df_train.csv",index=False)

df_test = pd.DataFrame()
num = []
for count,ele in enumerate(test_files,len(test_files)):
    df_test = pd.concat([df_test, pd.read_json(ele)])
    num.append(len(df_test))
# convert dataframe to csv file
df_test.to_csv("df_test.csv",index=False)
```

section_title	text
Foreword	The International Standard Classification of E...
Introduction'	A Guide to the International Interpretation of...
The Importance of International Data Comparabi...	Several persuasive arguments can be made for i...
Feasibility of Crosswalking U.S. Data to ISCED	The U.S. educational system differs in detail ...

3. LDA

Data preprocessing

```
def docs_preprocessor(docs):
    tokenizer = RegexpTokenizer(r'\w+')
    for idx in range(len(docs)):
        docs[idx] = str(docs[idx]).lower()  1
        docs[idx] = tokenizer.tokenize(docs[idx])  2

    docs = [[token for token in doc if not token.isdigit()] for doc in docs]  3

    docs = [[token for token in doc if len(token) > 3] for doc in docs]  4

    lemmatizer = WordNetLemmatizer()
    docs = [[lemmatizer.lemmatize(token) for token in doc] for doc in docs]  5

    return docs
```

1. 소문자
2. 단어 쪼개기 (tokenize)
3. 숫자 지우기 (숫자가 포함된 단어제외)
4. 한 글자 단어 삭제
5. 표제어 추출 (Lemmatisation)

3. LDA

Data preprocessing

```
dictionary = Dictionary(docs)
corpus = [dictionary.doc2bow(doc) for doc in docs]
```

```
print(corpus[0])
print(dictionary[0])
```

```
[(0, 1), (1, 1), (2, 2), (3, 2), (4, 1), (5, 1), (6, 2), (7, 2), (8, 1), (9, 1),
2), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 3), (28, 2), (29, 1), (30,
application
```

3. LDA

Topic Modeling

: 텍스트 본문의 숨겨진 의미 구조를 발견하기 위해 사용되는 텍스트 마이닝 기법

Latent Dirichlet Allocation, LDA (잠재 디리클레 할당)

- **가정** : 문서들은 토픽들의 혼합으로 구성되어져 있으며, 토픽들은 확률 분포에 기반하여 단어들을 생성한다

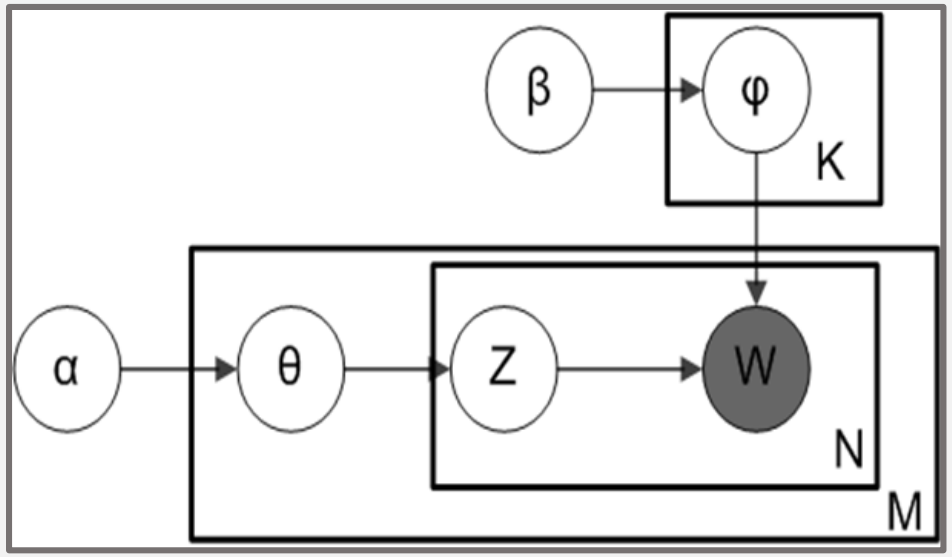
-> 데이터가 주어지면, LDA는 문서가 생성되던 과정을 역추적

각 단어나 문서들의 집합에 대해 숨겨진 주제를 찾아내어 문서나 키워드별로 주제끼리 묶어주는 비지도학습 알고리즘

3. LDA

1. 모든 문서와 문서 속 단어들에게 임의의 토픽 번호 부여
2. 각 문서의 토픽 분포 계산 ex) 문서1 : 토픽 A 100%
3. 각 토픽의 단어 분포 계산 ex) 토픽A : 사과 20%, 바나나 40%, 먹어요 40%
4. 단어 하나를 제외한 나머지 토픽-단어, 문서의 분포 고정
5. 미분류된 키워드의 토픽을 선정

-> 반복 : 가장 높은 확률을 가진 토픽에 해당 단어와 문서가 분류됨



Alpha, beta, K : 확률 분포 하이퍼파라미터
M : 문서 개수
N : 문서에 속한 단어개수
Theta : 문서의 토픽 확률분포
Phi : 주제의 단어
Z : 해당 단어가 속한 토픽의 번호
W : 실제 관측 가능한 단어

3. LDA

```
# parameters.
num_topics = 10
chunksize = 500
passes = 20
iterations = 100
eval_every = 1

temp = dictionary[0]
id2word = dictionary.id2token

%time model = LdaModel(corpus=corpus, id2word=id2word, chunksize=chunksize, \
                       alpha='auto', eta='auto', \
                       iterations=iterations, num_topics=num_topics, \
                       passes=passes, eval_every=eval_every)
```

```
topics = model.print_topics(num_words=4)
n = 0
topic_kw = []
for topic in topics:
    print(topic)
    wp = model.show_topic(n,topn=4)
    topic_keywords = " ".join([word for word, prop in wp])
    topic_kw.append(topic_keywords)
    print(topic_keywords)
    print()
    n+=1
```

```
(0, '0.011*"area" + 0.010*"during" + 0.008*"figure" + 0.008*"site"')
area during figure site

(1, '0.023*"water" + 0.023*"land" + 0.018*"coastal" + 0.018*"state"')
water land coastal state

(2, '0.015*"estimate" + 0.012*"greater_than" + 0.011*"would" + 0.010*"value"')
estimate greater_than would value
```

3. LDA

Results

```
0 번째 문서의 topic 비율은 [(0, 0.16226344), (2, 0.6534533), (4, 0.17565688)]
1 번째 문서의 topic 비율은 [(5, 0.77686316), (7, 0.22103836)]
2 번째 문서의 topic 비율은 [(1, 0.14337048), (3, 0.053312372), (5, 0.698619),
3 번째 문서의 topic 비율은 [(0, 0.08920176), (1, 0.04028297), (2, 0.07993211),
02), (9, 0.02901987)]
4 번째 문서의 topic 비율은 [(1, 0.4206185), (2, 0.30517423), (5, 0.24029988),
5 번째 문서의 topic 비율은 [(2, 0.25612217), (8, 0.7391999)]
```

	0	1	2
0	2.0	0.2949	[(0, 0.042030405), (2, 0.29487184), (4, 0.2832...
1	5.0	0.5739	[(2, 0.40562811), (5, 0.57393754)]
2	2.0	0.3678	[(2, 0.36777958), (5, 0.24499376), (6, 0.15704...
3	5.0	0.5304	[(2, 0.25071985), (5, 0.53038305), (6, 0.07159...

Submission ?

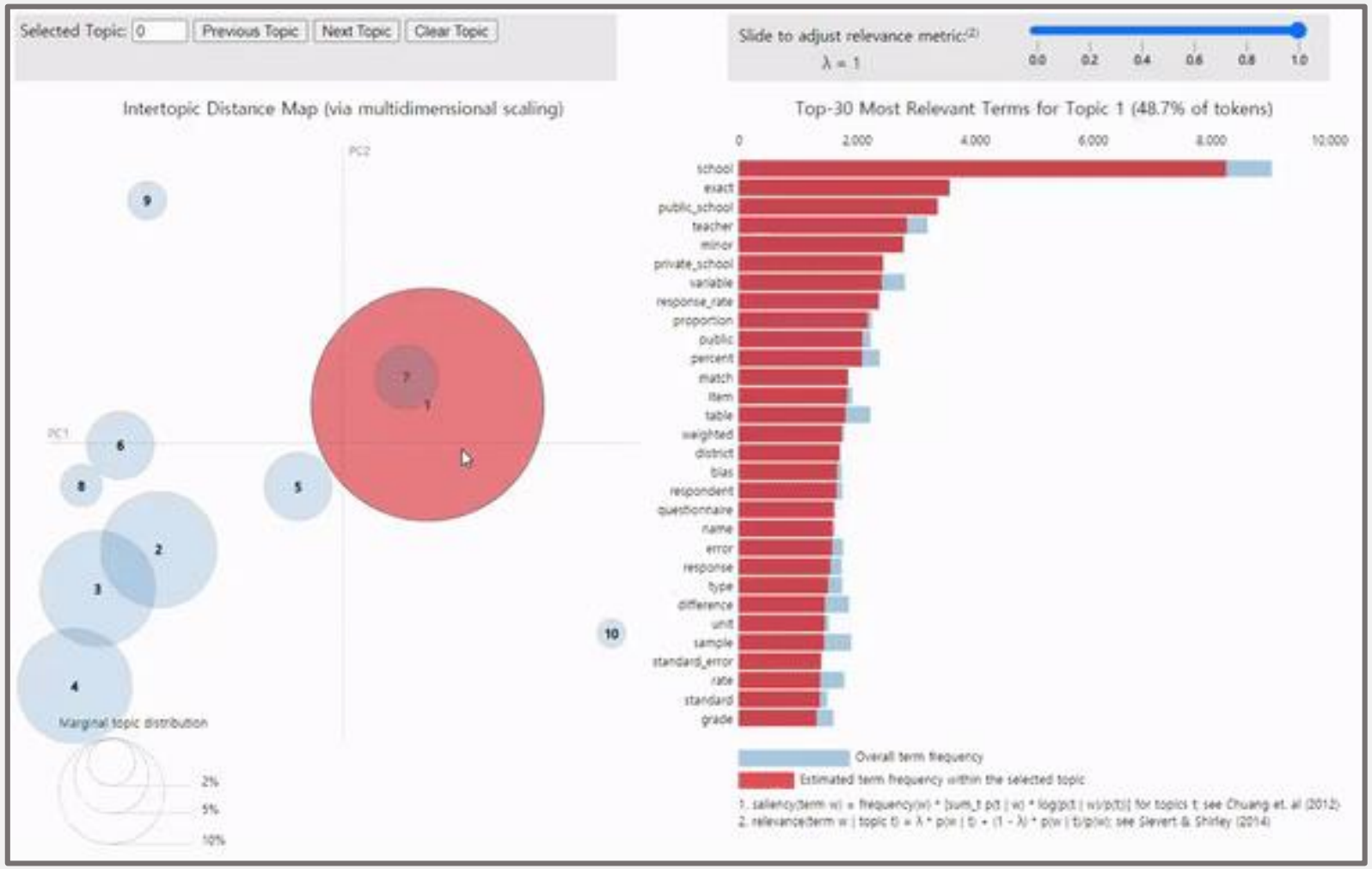
	Id	PredictionString
0	2100032a-7c33-4bff-97ef-690822c43466	student school teacher child
1	2f392438-e215-4169-bebf-21ac4ff253e1	student school teacher child
2	3f316b38-1a24-45a9-8d8c-4e05a42257c6	associated_with patient brain group
3	8e6996b4-ca08-4c0b-bed2-aaf07a4c6a60	associated_with patient brain group

LDA : 방대한 양의 문서들이 어떤 내용을 말하고 있는지에 대한 큰 맥락들을 크게 묶어주는 기법

>> 의미 있는 인사이트를 얻으려면 더 많은 과정이 필요

3. LDA

```
pyLDAvis.gensim.prepare(model, corpus, dictionary)
```



Kaggle Competition

[1] Coleridge Initiative - Show US the Data,
https://www.kaggle.com/c/coleridgeinitiative-show-us-the-data/overview

References

[1] Simple EDA and preprocessing of DataFrame,
https://www.kaggle.com/tanlikesmath/simple-eda-and-preprocessing-of-dataframe
[2] [ShowUsTheData] Topic Modeling with LDA,
https://www.kaggle.com/subinium/showusthedata-topic-modeling-with-lda
[3] **텍스트분석 – 토픽모델링(LDA)**
, http://bigdata.emforce.co.kr/index.php/2020072401/