공 학 석 사 학 위 논 문

자연어 처리를 이용한
배터리 소재 물성값 추출 자동화

2024년 2월

국 립 부 경 대 학 교 대 학 원

에 너 지 자 원 공 학 과

김 지 연

공 학 석 사 학 위 논 문

자연어 처리를 이용한
배터리 소재 물성값 추출 자동화

지도교수 여 병 철

이 논문을 공학석사 학위논문으로 제출함

2024년 2월

국 립 부 경 대 학 교 대 학 원

에 너 지 자 원 공 학 과

김 지 연

김지연의 공학석사 학위논문을 인준함

2024년 2월 16일

위 원 장 공학박사 김 동 훈 (인)

위 원 공학박사 여 병 철 (인)

위 원 공학박사 이 병 주 (인)

목 차

List of Tables

List of Figures

Automatic Extraction of Battery Materials properties
Using Natural Language Processing

Ji Yeon Kim

Department of Energy Resource Engineering, The Graduate School,
Pukyong National University

Abstract

Batteries play an essential role in various fields of modern society. The duration of device usage and the speed of battery charging depend on the battery's capacity and performance. Particularly, the anode material, a key component in storing lithium ions, critically affects the charging speed and lifespan of the battery. Consequently, battery-related research is focused on developing new anode materials that offer higher capacities and improved performance. However, experimenting with different properties of batteries for the development of anode materials can be costly and pose safety risks. Additionally, the lack of systematic organization of numerous research findings often leads to redundant experiments. While referring to academic papers to obtain necessary property values is a solution, manual search and analysis by individuals have limitations and risk missing crucial information. To address these issues, this study has developed an automated model using Natural Language Processing (NLP) technology. It automatically extracts and organizes the names of materials and property values from a vast number of research papers on battery anode materials. This allows researchers to quickly identify

the core topics of papers, clearly understand the content, track research trends, and swiftly grasp key keywords.

1. 서론

배터리는 현대 사회에서 중요한 역할을 하는 요소 중 하나로 자리매김하였다. 휴대폰에서부터 전기차, 그리고 에너지 저장 장치(Energy Storage System, ESS)에 이르기까지 다양한 제품과 시스템에서 핵심적인 역할을 수행하는 배터리는 용량과 성능이 중요한 결정 요소가 된다. 배터리의 용량은 사용자가 기기를 얼마나 오랫동안 사용할 수 있는지를 나타내며, 성능은 배터리의 효율성, 안정성, 충전 속도 등 다양한 요인을 포함한다. 즉, 용량과 성능이 우수하지 않은 배터리는 사용자의 편의를 방해하고, 기기의 신뢰성과 효율성을 떨어뜨릴 수 있다.

이러한 이유로 최근 몇 년 동안 배터리 연구의 트렌드는 기존에 사용되던 배터리보다 더 높은 용량, 더 긴 수명, 더 짧은 충전 시간 등 더 높은 성능을 가진 배터리를 개발하는 데 있다. 이를 위해서는 배터리의 핵심 구성요소 중 음극재의 개발이 필수적이다(Majdi et al., 2021). 음극재는 배터리 내에서 리튬 이온을 저장하는 중요한 구성요소로, 이는 배터리의 충전 속도와 수명에 결정적인 역할을 한다. 배터리는 양극과 음극에서의 화학 반응을 통해 전기 에너지를 발생시키는데, 이 과정에서 중요한 역할을 하는 것이 음극재이다. 음극재는 배터리의 전체 저장 용량을 결정하는데, 용량이 높은 음극재가 더 많은 전력을 저장하고 공급할 수 있게 한다. 그 결과, 충전 시간과 전체 배터리의 수명이 영향을 받게 된다.

주요 음극재로 사용되고 있는 물질은 흑연이다(Wu et al, 2003). 흑연은 비교적 낮은 비용, 높은 전력밀도 및 매우 긴 사이클 수명이라는 균형 잡힌 특성을 가지고 있다(Zhang et al., 2020). 하지만 충전 시 부피 팽창과 구조 변화로 인한 안정성 문제가 발생하여 흑연보다 우수한 성능의 음극재

개발이 연구의 주요 관심사로 부상하고 있다.

음극재 연구를 위해서는 배터리의 물성값을 정확하게 측정하고 이해해야 한다. 배터리 물성값은 전압(Voltage), 용량(Capacity), 잔존용량(State of Charge), 전류밀도(Current Density), 주기(Cycle) 등으로 나뉘며, 이러한 값을 실험으로 얻기 위해서는 많은 비용과 안전 문제가 발생할 수 있다.

국내 SCI 논문 증가 그래프를 살펴보면, 매년 배터리 관련 논문의 수가 지속적으로 증가하는 추세임을 확인할 수 있다(Fig 1). 또한, 연구 과정에서 얻어진 다양한 실험값들이 계속해서 나오고 있지만, 이러한 결과값들은 데이터베이스에 충분히 정리되거나 저장되지 않는 상황이다. 이로 인해 연구자들은 동일한 실험을 하는 경우가 빈번하게 나타나고 있다. 관련 논문을 직접 찾아서 필요한 물성값을 참고하는 방법도 있지만, 방대한 양의 논문을 사람이 수작업으로 검색하고, 읽고, 분석하는 데에는 한계가 있다(Hiszpanski et al, 2020). 모든 자료를 사람이 일일이 읽고 분석하는 것은 시간과 노력이 많이 소요될 뿐만 아니라, 중요한 정보를 놓칠 위험도 있다.

이 문제점에 대응하기 위한 대안으로 자연어처리 기술(Natural Language Processing, NLP)을 제시할 수 있다. 자연어처리 기술을 활용하면 대규모의 논문 데이터의 처리와 분석을 효율적으로 할 수 있다. 또한, 다양한 논문 간의 유사성을 파악하여 관련 연구나 비슷한 주제를 가진 논문들을 분류하는 것도 가능하다. 이를 통해 논문의 핵심 주제와 개념을 신속하게 식별하며, 연구의 중요 내용을 명확하게 이해할 수 있다. 더불어 전반적인 연구 동향과 주요 키워드를 빠르게 파악하는 것이 가능하다.

따라서 본 연구에서는 가장 잘 알려진 순환신경망(Recurrent Neural Networks, RNN) 혹은 트랜스포머(Transformer) 기반 자연어처리 기술을 활용하여 약 2,000개의 배터리 음극재 관련 연구논문에서 소재이름과 물성

값을 자동으로 추출하고 정리하는 자동화 모델을 개발하였다.

Fig 1. 한국 SCI 논문 수 및 점유율 추이(2007년~2021년)

2. 연구 방법

2.1 데이터 수집

웹에서 원하는 키워드의 문헌 데이터를 사람이 일일이 수집하는 것 또한 많은 시간이 소요되기 때문에 데이터를 수집하는 과정부터 자동화 기술이 필요하다. 따라서 파이썬의 Selenium 라이브러리를 이용한 크롤링으로 논문 데이터를 수집하였다. 크롤링(Crawling)은 웹사이트의 HTML (Hypertext Markup Language) 페이지를 그대로 가져와서 데이터 및 정보 자원을 자동화된 방법으로 수집, 분류, 저장하는 것을 말한다. 크롤링하는 소프트웨어를 크롤러(Crawler)라고 부르는데, 크롤러는 URL(Uniform Resource Locator)에서 시작하여 페이지를 방문 및 다운로드하고 색인을 생성한 다음 페이지 내의 하이퍼링크를 따라 다른 페이지에 연결한다. 이 과정(Fig. 2)은 미리 정의된 수의 페이지를 방문하거나 더 이상 관련 페이지를 찾을 수 없을 때까지 계속된다(Torkestani Akbari., 2012). 크롤링 하기 전, 각 사이트 별로 루트 경로의 robot.txt에 명시된 로봇룰을 따라서 크롤러를 만들었는지 확인해야한다.

본 연구에서는 논문 데이터를 수집할 출판사로 ACS(American Chemical Society)를 채택한 다음, 음극재 관련 논문 수집을 위해 'battery anode'를 키워드 조건으로 설정하여 수집 범위에 제한을 두었다(Fig. 3). 수집한 데이터 파일은 생성된 HTML 폴더에 저장된다(Fig. 4). 이 단계를 통해 총 1,956개의 HTML 파일을 수집하였다.

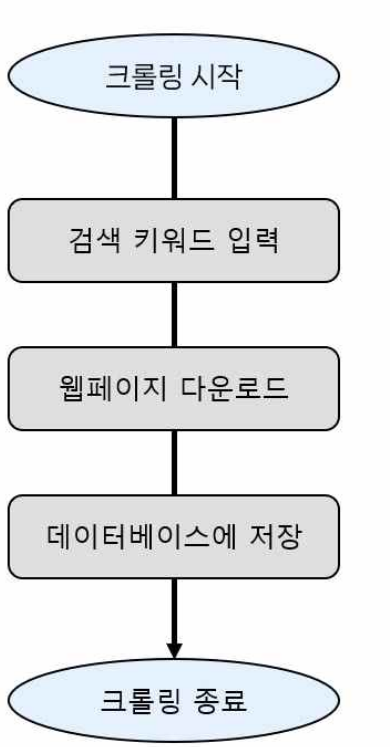

Fig 2. Web Crawling Cycle Flowchart

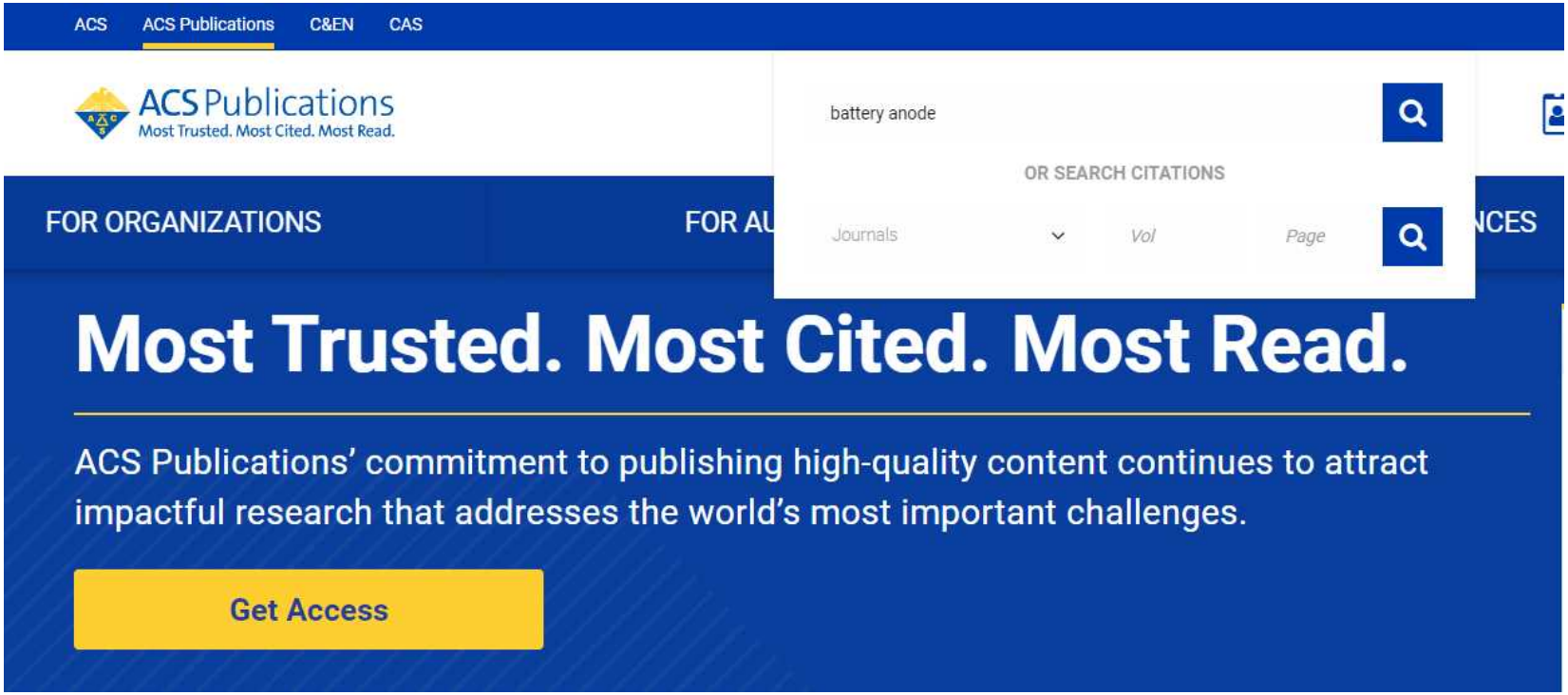

Fig 3. Enter the keyword 'battery anode' on the ACS publisher's website

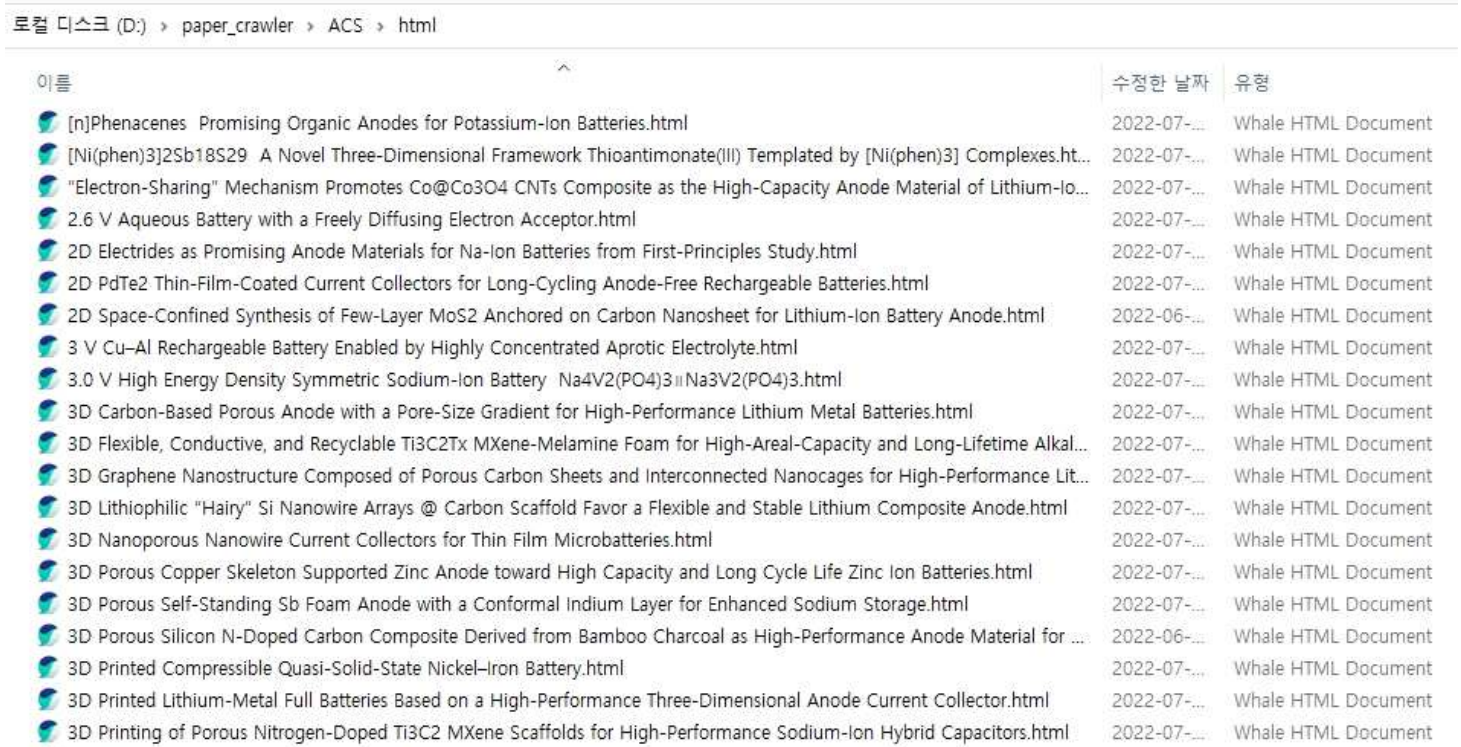

Fig 4. View html file data stored in the specified folder

2.2 데이터 전처리

자연어처리 기술을 사용하기 위해서는 앞서 수집한 HTML 파일을 전처리해야 한다. 전처리는 파싱(Parsing)과 토큰화(Tokenization) 두 가지 단계로 진행된다.

2.2.1 파싱(Parsing)

첫 번째 단계는 파싱이다. 파싱(Parsing)이란 문자열의 구조를 분석해서 원하는 정보를 얻는 것을 말한다. 이를 통해 복잡한 문자열 데이터를 우리가 원하는 형태나 구조로 변환할 수 있다. 본 연구에서는 복잡한 형태의 HTML 파일을 파싱하여 표 및 그래프 등을 제외한 논문 1,805개의 Abstract만 JSON 문자열의 파일로 수집했다. 논문의 핵심이 되는 내용과 실험 결과값 등이 Abstract에 담겨있어 전문을 분석하지 않아도 물성값을 충분히 얻을 수 있기 때문에 Abstract만 수집하였다. 또한, 이렇게 하면 전문을 분석할 때보다 토큰화 및 태깅에 소요되는 시간을 절약할 수 있다.

파싱 작업은 Fig 5.에서 확인할 수 있듯이, 파서(parser)를 통해 수행된다. 파서는 주어진 규칙 또는 문법을 기반으로 입력 문자열을 분석하고, 해당 문자열의 구조와 의미를 파악하여 적절한 형태의 출력을 생성한다.

```python
class ACSParser(Parser):
    def __init__(self):
        super().__init__()

        def img_downloader(url, savename):
            with Image.open(BytesIO(requests.get(url).content)) as im:
                im.save(savename)

        self.img_downloader = img_downloader

    def _refine_text(self, string):
        string = super()._refine_text(string)

        string = re.sub('\(( )+\)', '', string)
        string = re.sub(' (-)', '', string)

        return string

    def _get_metadata(self, soup, data):
        data['title'] = soup.select_one('.article_header-title')
        data['author'] = soup.select('.loa .hlFld-ContribAuthor')
        data['journal'] = soup.select_one('.aJhp_link')
        data['doi'] = soup.select_one('.article_header-doiurl')

        references = []
        for refer in soup.select('#references .NLM_citation'):
            refer.select_one('.citationLinks').decompose()

            references.append(refer)

        data['references'] = references

        return data
```

Fig 5. Parsing code partial extract

2.2.2 토큰화(Tokenization)

토큰화 단계를 시작하기 전에 먼저 JSON 문자열 파일로 저장된 데이터들을 모두 TXT 형식의 파일로 전환해야 한다. 토큰화 과정은 주어진 텍스트 데이터를 토큰(token) 단위로 나누는 작업이기 때문이다.

토큰은 단어, 숫자, 구두점 등이 될 수 있으며, 이는 파싱할 텍스트의 유형과 문법에 따라 달라진다. 본 연구에서는 토큰의 기준을 단어(word)로 하는, 단어 토큰화(word tokenization)를 하였다(Fig 6).

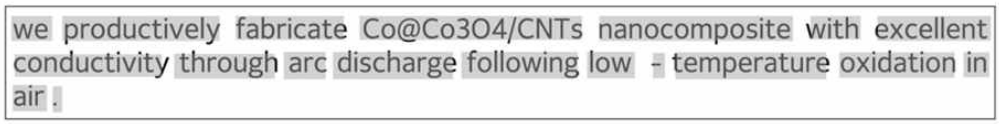

Fig 6. Tokenization Examples

사용한 함수는 NLTK(Natural Language Toolkit) 라이브러리에서 제공하는 Word_tokenize이며, 주어진 텍스트를 나누며 단어로 인식하는 지표는 띄어쓰기가 된다.

2.3 데이터 시각화

2.3.1 Doc2Vec

시각화 하기에 앞서, 텍스트 데이터의 고차원 특성을 효과적으로 수치 벡터로 변환하기 위해 Doc2Vec 단계를 가진다. 텍스트 데이터는 자체적으로는 연속적인 수치 데이터가 아니기 때문에 시각화를 위한 알고리즘을 직접 적용하기 어렵다. Doc2Vec (또는 Paragraph Vector)는 단어 시퀀스에 대한 고정 길이의 특징 벡터를 학습하는 기술이다(Mikolov et al., 2013). 이 기술은 Word2Vec 기술을 확장하여 개발되었다(Kim et al, 2019). Doc2Vec은 텍스트를 고차원에서 저차원의 연속적인 벡터로 변환해준다.

Doc2Vec에는 크게 두 가지 방법이 있다. 첫 번째는 Distributed Memory (DM) 방식으로, 문서의 문맥과 함께 단어들을 예측하는 방식이다(Le et al, 2014). 두 번째는 Distributed Bag of Words (DBOW) 방식으로, 문서 내의 단어들을 예측하면서 문서의 벡터만 학습하는 방식이다(Le et al, 2014). 본 연구에서는 문서 내 단어의 순서를 고려하고, 각 단어와 문맥을 함께 고려하여 문서의 벡터를 학습하는 DM 방식을 사용하였다(Fig 7).

Doc2Vec 모델을 학습시키고, 학습된 모델을 사용하여 각 문서를 벡터로 표현한 다음 'KMeans' 함수를 사용하여 문서의 벡터화된 표현을 기반으로 유사한 문서들을 군집화하였다(Fig 8). KMeans는 sklearn 라이브러리의 군집화(clustering) 알고리즘이다. 군집화는 관측 데이터를 여러 그룹으로 나누는 것을 목적으로 한다. 여기서 사용된 KMeans는 k-평균 알고리즘이라고도 하며, 각 군집의 중심을 찾아 그 중심에 가장 가까운 데이터 포인트들을 그 군집에 할당하는 방식으로 동작한다(Ikotun et al, 2023). 이 함수를 사용함으로써, 데이터의 분석과 시각화가 더 용이해졌다.

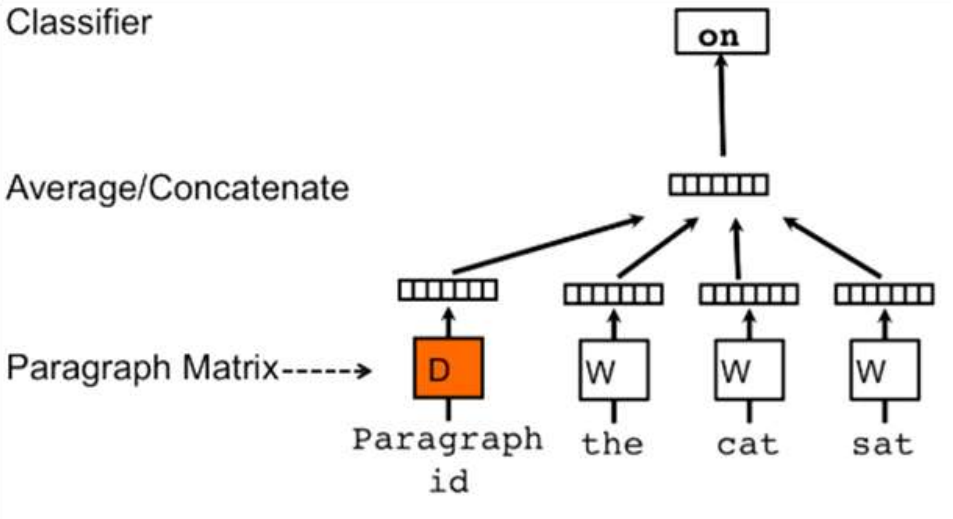

Fig 7. PV-DM method

(출처: Christou Despina, Feature extraction using Latent Dirichlet Allocation and Neural Networks: A case study on movie synopses, (2016))

```
# Clustering
labels = KMeans(n_clusters=5, random_state=0).fit(abstract_vectors).labels_
print(labels.shape)

labels_list = []
for i in range(1772):
   labels_list.append(labels[i])

print(labels_list)

file_name = "label_list_0.txt"
with open(file_name, 'w+',encoding='utf-8') as file:
    for i in range(1772):
        if labels_list[i] == 0:
            file.write(str(labels_list[i]) + '    ' + abstract_data[i] + '\n')


file_name = "label_list_1.txt"
with open(file_name, 'w+',encoding='utf-8') as file:
    for i in range(1772):
        if labels_list[i] == 1:
            file.write(str(labels_list[i]) + '    ' + abstract_data[i] + '\n')
```

Fig 8. Clustering with 'KMeans' function

2.3.2 t-SNE와 PCA

본 연구에서는 데이터 시각화를 위해 t-SNE(t-Distributed Stochastic Neighbor Embedding)와 PCA(Principal Component Analysis) 두 가지 방법을 사용하였다. t-SNE와 PCA는 둘 다 고차원 데이터를 저차원으로 시각화하거나 축소하는 기법이지만, 그 방식과 주된 용도에 있어 차이점이 있다.

1. PCA(주성분 분석): 선형적인 방법으로 데이터의 분산을 최대화하는 주축들을 찾아 데이터를 축소한다(Salem et al, 2019). 이는 원래의 데이터 성질을 유지하려는 데 중점을 둔다(Salem et al, 2019). 장점은 원본 데이터의 구조와 방향성을 더욱 잘 이해할 수 있게 도와준다. 또한, 계산 비용이 낮아 큰 데이터 셋에서도 빠르게 처리할 수 있다.(Fig 9).

```python
# Visualization: PCA
def show_pca(X_data, name_data, labels):
    pca = PCA(n_components=2).fit_transform(X_data)

    plt.figure()
    for label in set(labels):
        indices = [i for i, l in enumerate(labels) if l == label]
        plt.scatter(pca[indices, 0], pca[indices, 1], label=label)
    for i, name in enumerate(name_data):
        plt.text(pca[i, 0], pca[i, 1], name, fontdict={'weight': 'bold', 'size': 9})
    plt.xlabel("1st PC")
    plt.ylabel("2nd PC")
    plt.legend()
    plt.show()


show_pca(abstract_vectors, tag_vectors, labels)
```

Fig 9. Partial Excerpt from PCA visualization Code

2. t-SNE: 비선형적인 방식으로 데이터 포인트 간의 유사도를 최대한 보존하려 하며, 주로 데이터의 클러스터링 구조나 패턴을 시각화하는 데에 사용된다. 장점은 t-SNE는 데이터의 복잡한 비선형 관계와 클러스터 구조를 잘 드러내 데이터 내의 패턴이나 그룹화를 시각적으로 구별하기 용이하게 만들어준다(Fig 10)(Zhou, et al, 2018).

```python
# Visualization: t-SNE
def show_tsne(X_data, name_data, labels):
    tsne = TSNE(n_components=2).fit_transform(X_data)
    df = pd.DataFrame(tsne, index=name_data, columns=['x', 'y'])

    plt.figure()
    for label in set(labels):
        indices = [i for i, l in enumerate(labels) if l == label]
        plt.scatter(df.iloc[indices]['x'], df.iloc[indices]['y'], label=label)
    for word, pos in df.iterrows():
        plt.annotate(word, pos, fontsize=10)
    plt.xlabel("t-SNE component 0")
    plt.ylabel("t-SNE component 1")
    plt.legend()
    plt.show()


show_tsne(abstract_vectors, tag_vectors, labels)
```

Fig 10. Partial Excerpt from t-SNE visualization Code

2.4 데이터 태깅

태깅 과정은 품사 태깅과 BIO 태깅 두 단계로 구성되어 있다.

2.4.1 품사 태깅(Part-Of-Speech tagging)

품사 태깅(Part-Of-Speech tagging), 또는 문법 태깅으로 불리는 이 과정은 텍스트 내 단어를 해당 품사에 맞게 분류하는 것을 의미한다(Chiche et al, 2022). 태깅을 위해 Penn TreeBank의 POS Tagset(Table. 1)을 사용하였으며, 대략 36가지의 세부품사가 있고 각 품사는 정해진 약자로 표시된다. BIO 태깅 전에 품사 태깅 단계를 가진 이유는 BIO 태깅을 할 때 품사 중 명사에 해당하는 단어만 필요하기 때문이다.

Table 1. Penn TreeBank POS Tagset

Tag	Description	Tag	Description	Tag	Description
CC	Coordinating conjunction	NNS	Noun, plural	TO	to
CD	Cardinal number	NNP	Proper noun, singular	UH	Interjection
DT	Determiner	NNPS	Proper noun, plural	VB	Verb, base form
EX	Existential there	PDT	Predeterminer	VBD	Verb, past tense
FW	Foreign word	POS	Possessive ending	VBG	Verb, gerund or present participle
IN	Preposition or subordinating conjunction	PRP	Personal pronoun	VBN	Verb, past participle
JJ	Adjective	PRP$	Possessive pronoun	VBP	Verb, non-3rd person singular present
JJR	Adjective, comparative	RB	Adverb	VBZ	Verb, 3rd person singular present
JJS	Adjective, superlative	RBR	Adverb, comparative	WDT	Wh-determiner
LS	List item marker	RBS	Adverb, superlative	WP	Wh-pronoun
MD	Modal	RP	Particle	WP$	Possessive wh-pronoun
NN	Noun, singular or mass	SYM	Symbol	WRB	Wh-adverb

2.4.2 B-I-O 태깅

문장에서 명명된 엔터티(예: 인물 이름, 위치, 조직 이름)를 자동으로 식별하고 분류하는 작업을 개체명 인식(Named-Entity Recognition)이라 한다(Shelar et al, 2020). 이러한 개체명 인식을 위해 사용되는 방법 중 한 가지로 BIO 태깅이 있는데, 이는 텍스트 내의 개체명을 이루는 각 음절 또는 단어에 'B', 'I', 'O' 중 하나의 태그를 지정하는 작업을 말한다. 여기서 B는 Begin, I는 Inside, O는 Ouside의 약자이다(Zhang et al, 2006). 이러한 BIO 태깅을 통해 기계는 텍스트에서 인물명, 시간, 장소, 수량 등에 해당하는 표현이 무엇인지 파악하는 지표를 얻을 수 있게 된다.

B, I, O 각 태그 뒤에는 주석이 달리는데, 이 주석들은 사전에 선정한 토픽의 약자이다. 본 연구에서는 Anode name, Capacity, Current density, Cycle 총 4가지의 토픽을 선정하였고, 주석이 되는 각 토픽의 약자는 Table. 2를 통해 확인할 수 있다. 주석을 정하고 나면 Table. 3과 같은 태깅 규칙을 세운 후, 본격적으로 태깅을 시작한다.

BIO 태깅 방법은 다음과 같다.

1. 인물명, 단체명, 장소명과 같은 개체명을 찾는다.
2. 개체명의 토큰을 확인한다.
3. 개체명이 단일 토큰일 경우 B-주석, 토큰이 다수일 경우 처음 토큰에만 B-주석으로 태그하고 두 번째 토큰부터 I-주석으로 태그한다.
4. 개체명이 아닌 것에 Outside를 의미하는 O를 태그 한다.

Table 2. B-I-O Tagging

name	Tag	name	Tag
Anode Name	B-AN	Cycle	B-CY
	I-AN		I-CY
Capacity	B-CP	etc.	O
	I-CP		
Current Density	B-CD		
	I-CD		

Table 3. B-I-O Tagging 규칙

No.	Case	Example	Apply
1	Anode/Cathode Name	Si anode	Si(B-AN) anode(I-AN) →Cathode일 경우 B-CA/I-CA
2	Capacity	1548mAh/g (또는 mAh g-1)	1548(B-CP) mAh/g(I-CP)
3	Current Density	311 mA/g (또는 mA g-1)	311(B-CD) mA/g(I-CD)
4	Cycle	219 cycles	219(B-CY) cycles(I-CY)

2.5 자연어처리 모델

태깅한 데이터를 사용하여 개체명 인식 단계에 들어간다. 개체명 인식(Named Entity Recognition, NER)은 텍스트에서 특정한 정보(예: 사람 이름, 조직 이름, 위치, 날짜, 시간 등)를 식별하고 분류하는 자연어 처리 작업이다(Li et al, 2020). 본 연구에서는 개체명 인식을 위해 순환신경망 기반 자연어처리 모델인 BiLSTM(Bidirectional Long Short-Term Memory)과 BiLSTM-CRF(Conditional Random Field), 트랜스포머 기반 BERT(Bidirectional Encoder Representations from Transformers, BERT)를 사용하였다.

2.5.1 BiLSTM 모델

양방향 장단기 메모리 모델(BiLSTM)은 장단기 메모리 모델(Long Short-Term Memory, LSTM)을 양방향으로 확장한 모델이다(Fig 11). 과거의 정보만을 고려하여 현재 상태를 예측하는 LSTM와는 달리, BiLSTM은 과거의 정보와 미래의 정보를 동시에 고려한다는 이점이 있다(Siami-Namini et al, 2019). 각 시점에서 두 개의 메모리 셀을 사용하여, 하나는 과거의 정보를 전달하고 다른 하나는 미래의 정보를 전달한다. 이러한 구조 덕분에 BiLSTM은 주어진 시점의 문맥을 양방향으로 고려하여 텍스트의 의미를 더 정확하게 인식할 수 있다. 특히, 자연어처리에서 문장의 의미를 파악하거나 개체명 인식, 감성 분석 등의 작업에 효과적으로 사용된다(Hameed et al, 2020).

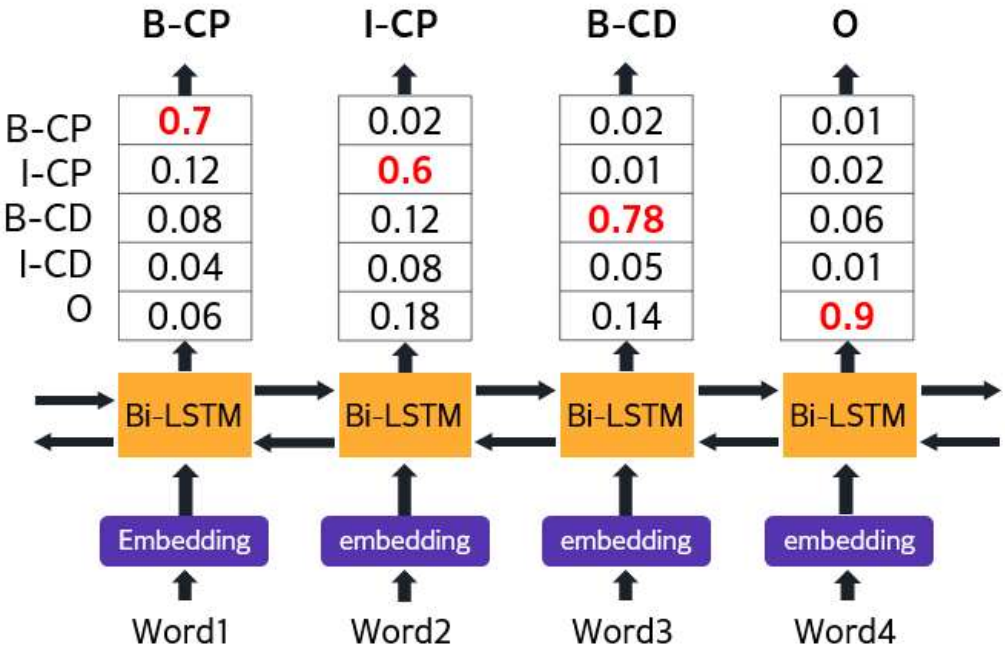

Fig 11. BiLSTM 모델

2.5.2 BiLSTM-CRF 모델

Fig 12.를 보면 알 수 있듯이, 기존 BiLSTM에 CRF층이 추가된 구조이다. BiLSTM과 BiLSTM-CRF의 차이를 결정하는 CRF는 조건부 랜덤 필드라 불리며, 일련의 항목들이 순서대로 나열된 형태의 시퀀스 데이터(텍스트 데이터)에 대한 라벨을 예측하기 위한 확률 모델이다. BiLSTM의 출력은 CRF 레이어에 공급되며, CRF는 전체 시퀀스를 고려하여 최적의 라벨 시퀀스를 예측한다(Huang et al, 2015). 문장 내의 각 단어에 대한 풍부한 문맥 정보를 제공하는 BiLSTM의 장점과, 시퀀스의 각 부분에 대한 최적의 라벨링을 결정하기 위해 전체 시퀀스를 고려하는 CRF의 장점이 결합되어 뛰어난 성능을 보인다(Wei et al, 2019).

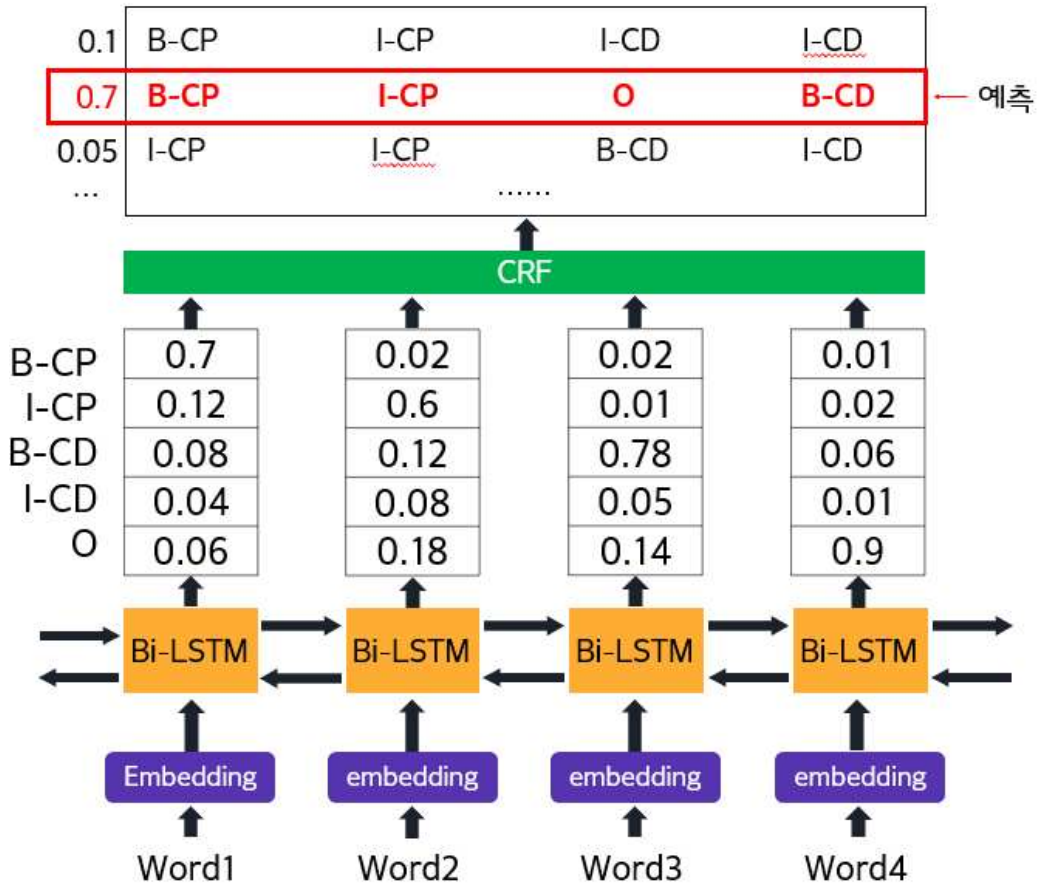

Fig 12. BiLSTM-CRF 모델

2.5.3 BERT 모델

BERT 모델은 문장을 양방향으로 읽어 문맥을 파악하는 BiLSTM과는 달리 Transformer라는 구조를 사용하여 문장 내 모든 단어 간의 관계를 동시에 파악한다(Fig 13). BERT 모델은 여러 개의 층(layer)으로 구성되어 있으며, 각 층은 'Multi-head Self-Attention' 메커니즘과 'Feed Forward Neural Network (FFNN)'로 이루어져 있다. 각 층의 출력은 'Add & Norm' 단계를 거쳐 다음 층으로 전달된다(Rizk et al, 2022). 이로 인해, 주어진 단어나 문구가 그 문장 내에서 어떤 의미로 쓰였는지 더욱 정확하게 이해할 수 있다. MLM(Masked Language Model) 방식을 사용하여 임의로 가린 단어를 문장의 전체 문맥을 바탕으로 예측하며 학습한다(Devlin et al, 2018). 또한 문장 길이 고정에 구애받지 않고, 다양한 길이의 텍스트를 효과적으로 처리할 수 있다.

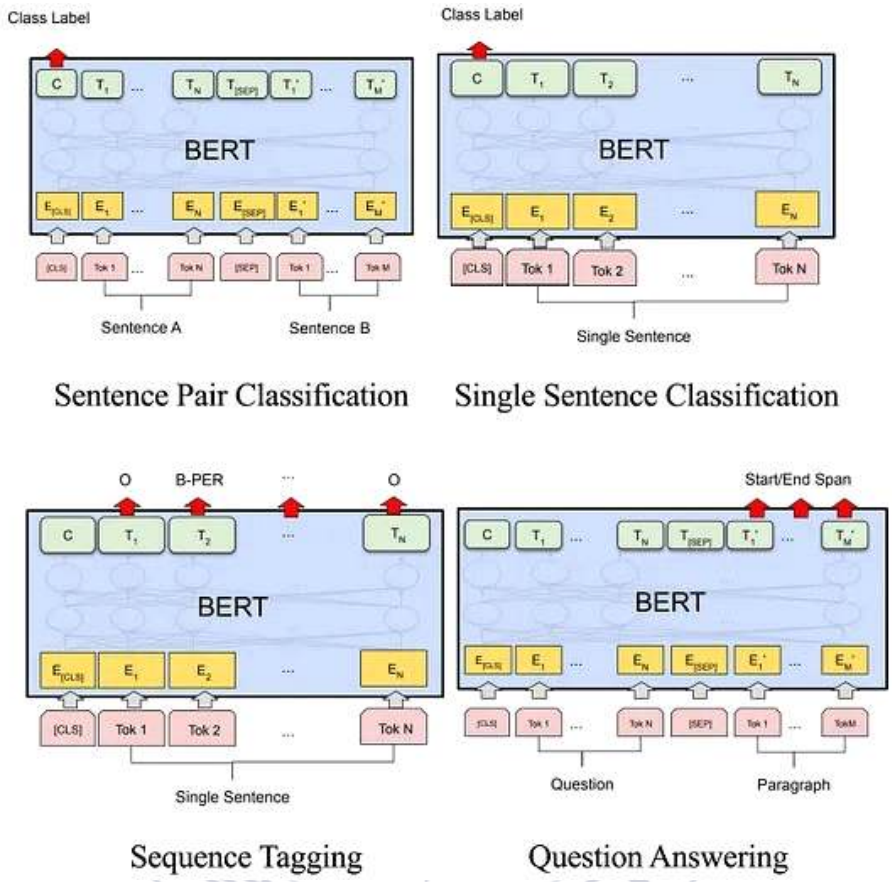

Fig 13. BERT 모델

(출처: Devlin Jacob. Chang Ming-Wei, Lee Kenton, Toutanova Kristina, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *[Submitted on 11 Oct 2018 (v1), last revised 24 May 2019 (this version, v2)]*)

3. 연구 결과

3.1 데이터 태깅 결과

B-I-O 태깅을 진행할 때, 저자마다 논문 작성 스타일과 물성값 단위 표시 방식이 조금씩 상이해서 사전에 태깅 규칙을 세웠음에도 태깅 과정에서 어려움을 겪었다. 이를 해결하기 위해 추가적으로 새로운 규칙을 세우고 태깅을 완성하였다(Table 4).

순서대로 POS tagging과 BIO tagging을 끝마치고 나면, Fig 14와 같은 결과가 나온다.

Table 4. Tagging 단계에서 어려웠던 점

No.	Case	Example	Apply
1	수치 뒤에 단위가 바로 오지 않음	숫자and 숫자mAh/g (단위는 mA/g 등 다양)	and를 기준으로각각 따로 태깅 →숫자(B-CP) and 숫자(B-CP)mAh/g(I-CP)
2	수치 뒤에 and 및 문장이 온 후 다음 수치와 연결됨	숫자 and (문장) 숫자mAh/g	and를 기준으로각각 따로 태깅 →숫자(B-CP) and (문장) 숫자(B-CP)mAh/g(I-CP)
3	이름 뒤에 Anode가 바로 오지 않음	이름 as the anode	이름만 태깅
4	Capacity 단위가 다른 형식	mAh/cm, mAh cm-2	mAh/g와 동일하게 B-CP로 태깅
5	Current Density 단위가 다른 형식	mA/cm2, mA cm-2 Wh/kg, Wh/g	mA/g와 동일하게 B-CD로 태깅
6	Current Density 단위와 유사	W/kg, W/g	다음은 전력밀도 단위. CD가 아님.

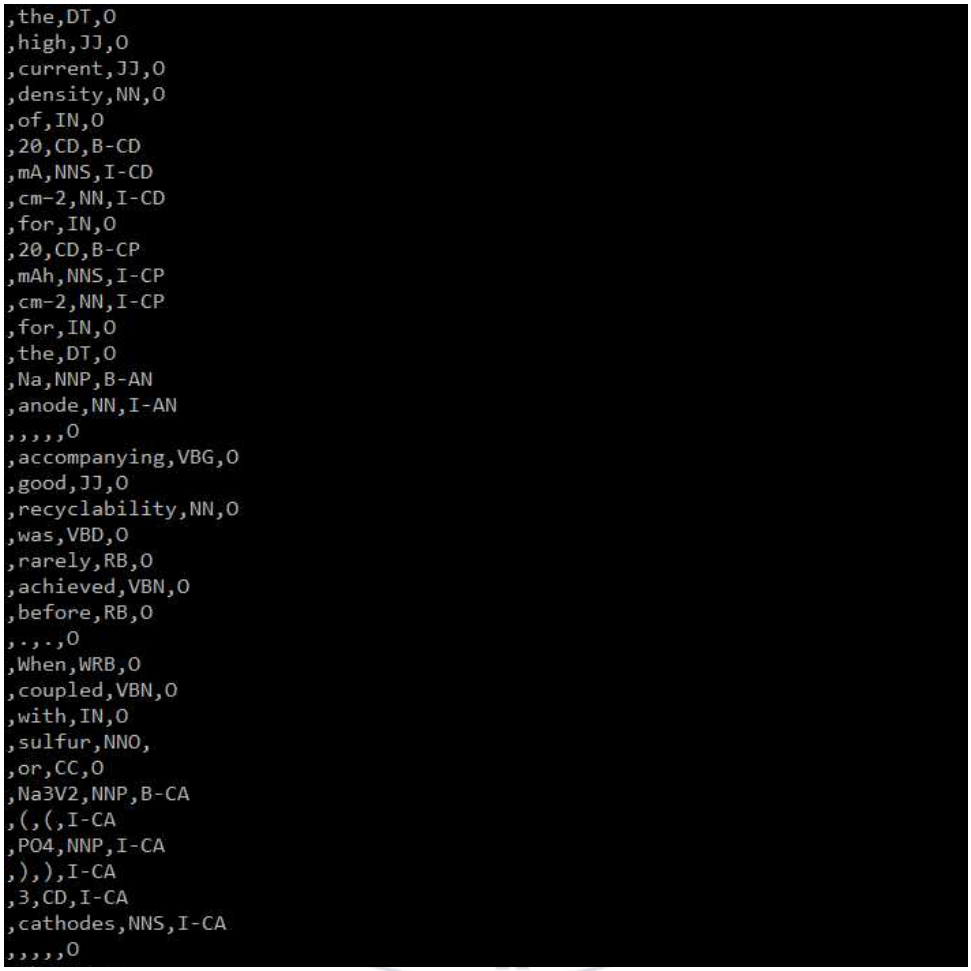

Fig 14. Finished POS tagging and BIO tagging

3.2 데이터 시각화 결과

1. PCA

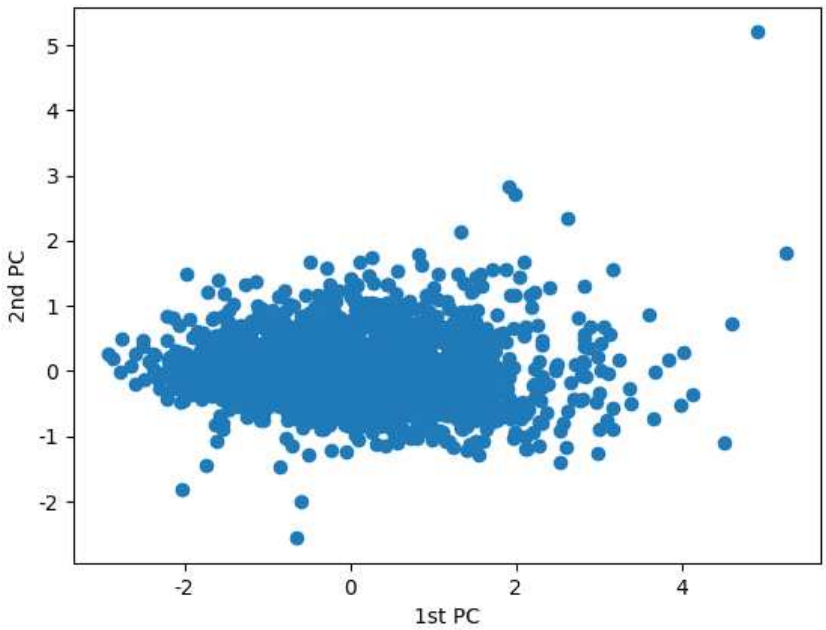

Fig 15. PCA Visualization Results

2. t-SNE

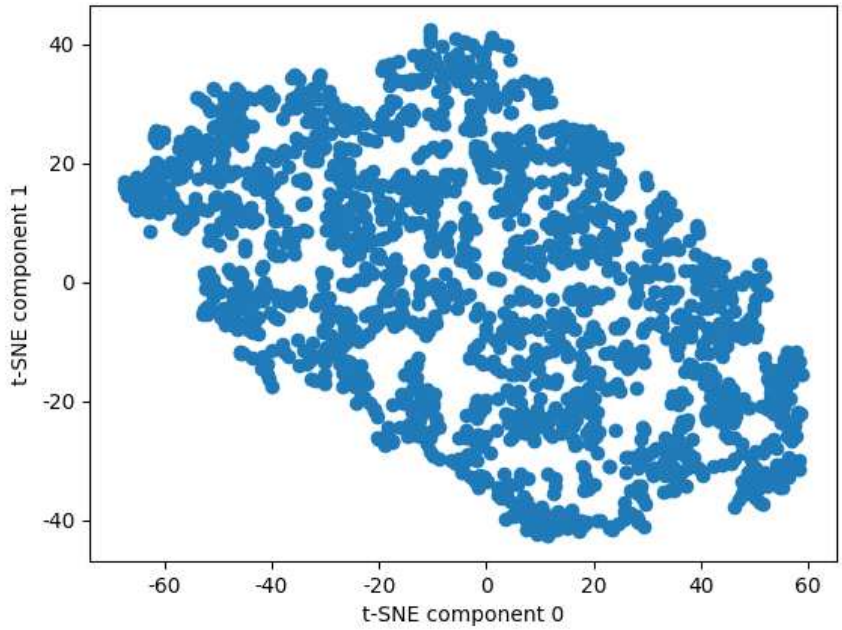

Fig 16. t-SNE Visualization Results

Doc2Vec 모델로 벡터화 한 고차원 데이터를 차원 감소 기술 t-SNE와 PCA로 시각화 하였을 때의 결과는 다음과 같다(Fig 15)(Fig 16).

PCA그래프를 보면, 응집되어 있지 않고 따로 떨어져 찍힌 점들이 존재하는 것을 볼 수 있다. 원인을 파악하기 위해 이 중에서 군집으로부터 멀리 떨어진 점 5개를 선정하여 초록을 분석해보았다(Fig 17).

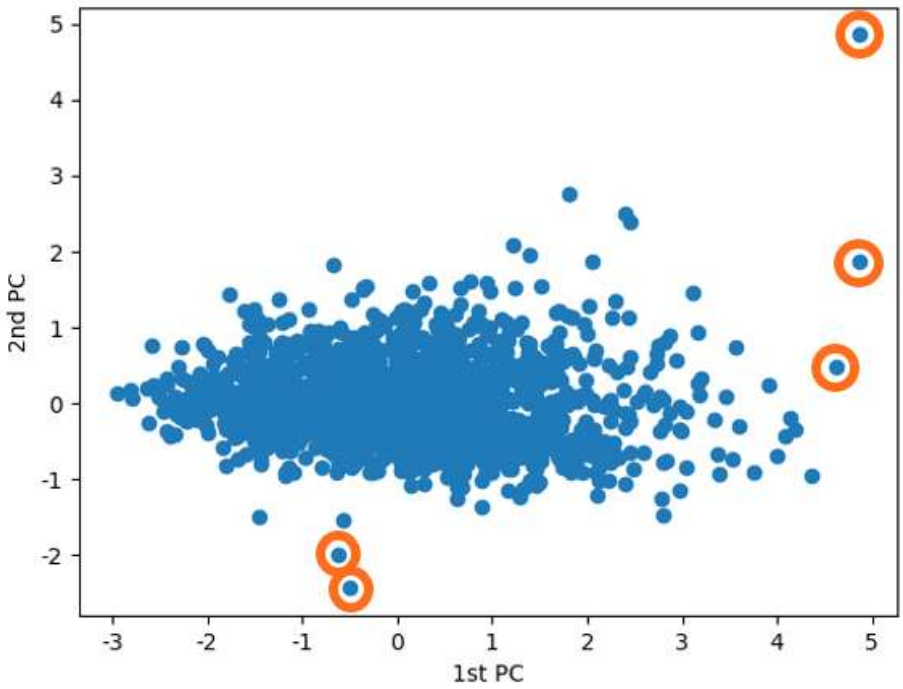

Fig 17. 5 points away from cluster in PCA graph

Table 5. 군집으로부터 떨어진 5개의 점 초록 분석

No.	Abstract
1	리튬 이온 배터리 제조
2	전해질에서 실리콘 나노와이어의 리튬화 용량
3	리튬 이온 배터리 전극용 현탁액
4	충전식 리튬 이온 배터리의 성능 향상
5	탄소 에어로젤 구조 하이브리드 제조

분석결과, 5개의 점에 포함된 논문 모두 제목이나 초록에서 ‘battery anode’ 라는 키워드를 포함하고 있었다. 그러나 Table 5에서 볼 수 있듯, 이들의 연구 내용은 배터리 음극재와는 거리가 먼 내용이었다. 이를 통해 그래프 상에서 멀리 떨어진 점들은 키워드와 관련성이 떨어지는 논문일 가능성이 높으며, 이를 제외한 나머지 점들은 그래프에서 하나의 군집을 형성하고 있음을 확인할 수 있다. 이는 데이터 수집 단계에서 설정한 키워드 ‘battery anode’에 부합하는 문서들이 적절하게 수집되었음을 나타낸다.

3.3 자연어 처리 모델 학습 결과

본 논문에서는 계산된 F1-score를 기반으로 BiLSTM, BiLSTM-CRF, BERT 세 가지 모델의 성능을 분석하였으며, 이를 막대 그래프(Fig 17)를 통해 시각적으로 비교하여 제시하였다. F1-score는 분류작업에서 사용되는 성능지표로, 정밀도(Precision)와 재현율(Recall)의 조화평균을 나타낸다.

$$F1-score = 2 \times \frac{recall \times precision}{recall + precision} \quad (1)$$

훈련 데이터에 대해서 BiLSTM의 F1-score는 96%, BiLSTM-CRF의 F1-score는 94%, BERT의 F1-score는 94%로 세 가지 모델 모두 높은 성능을 보였다. 하지만 주목해야 할 점은 바로 테스트 데이터이다. BiLSTM 모델과 BiLSTM-CRF 모델은 테스트 데이터에서는 순서대로 56%, 64%로 상당한 성능 감소를 보였다. 이는 모델이 훈련 데이터에 과적합 되었음을 시사한다. 반면 BERT 모델은 훈련 데이터에서 94%, 테스트 데이터에서 95%의 F1-score 계산 결과가 나왔다. 세 가지 모델 중 가장 높은 테스트 데이터 성능을 보인 것은 물론, 훈련 데이터와 가장 편차가 적은 성능을 보였다. 이는 BERT의 사전 훈련된 컨텍스트 기반 표현이 배터리 소재 물성값을 포함한 복잡한 자연어 문맥을 이해하는데 매우 효과적임을 시사한다.

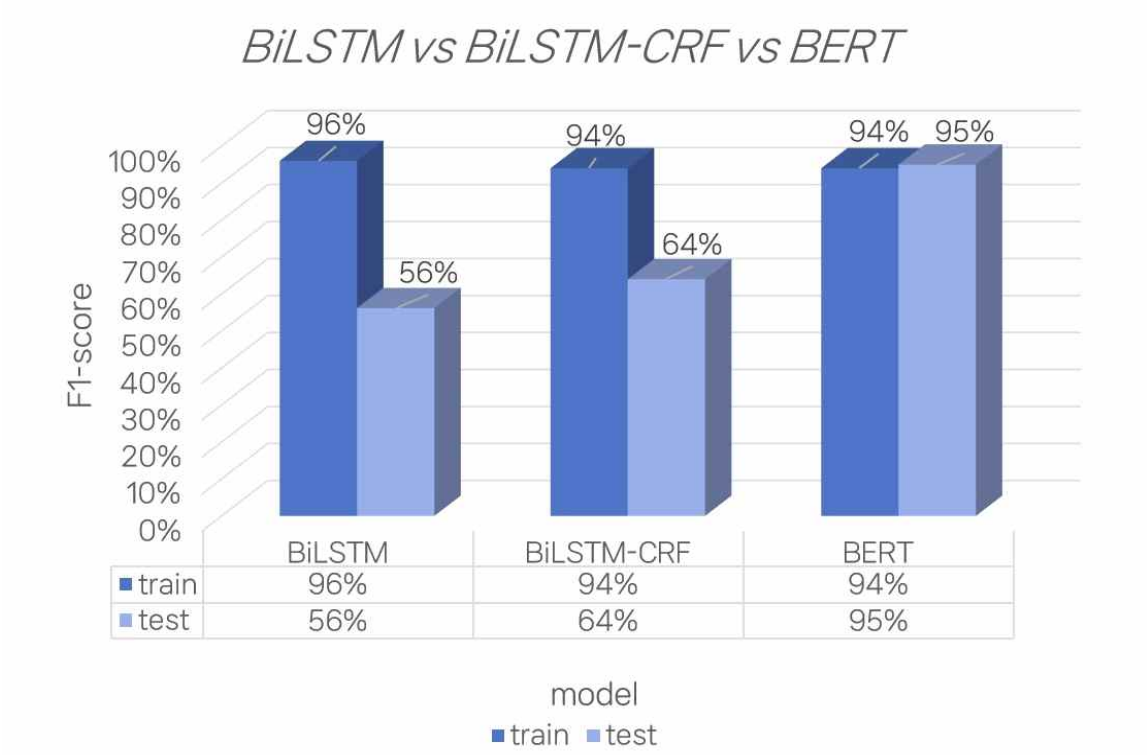

Fig 18. BiLSTM vs BiLSTM-CRF vs BERT

4. 토의

본 연구의 목적인 자연어처리 기술을 이용하여 배터리 소재 물성값 추출하는 자동화기술은 배터리 소재 개발에 있어 중요한 의미를 갖는다. 왜냐하면, 배터리 소재-물성 데이터베이스의 구축과 확장을 가속화하고, 배터리 소재 연구자들이 새로운 소재를 제안하기 위한 영감을 제공할 수 있기 때문이다. 그런데, 본 연구의 한계점으로 사용된 데이터셋의 크기와 다양성이 제한적이라는 점을 들 수 있다. 따라서 향후 연구에서는 논문에 포함된 그래프와 표에서도 물성값을 추출하는 등의 과정을 추가하여 더 넓은 범위의 데이터를 수집할 예정이다. 그리고 배터리 소재와 연관된 키워드를 검색한 데이터셋을 활용하여 모델의 일반화 능력을 강화하고, 모델의 해석 가능성을 향상시키는 방향으로 노력할 것이다. 또한, 모델의 해석 가능성을 높이기 위해 중간 레이어의 활성화와 같은 모델 내부의 메커니즘을 분석하는 추가적인 연구가 필요하다. 이러한 연구는 모델의 결정을 이해하고, 잠재적인 오류를 수정하는 데 도움이 될 것이다.

5. 결론

본 연구는 배터리 음극재의 소재이름과 물성값을 추출하는 다양한 자연어처리 모델의 성능을 평가하였다. 특히, BERT 모델이 뛰어난 성능을 보임으로써, 복잡한 자연어 문맥에서 중요한 정보를 효과적으로 추출할 수 있는 강력한 도구임을 확인하였다. 이러한 결과는 배터리 소재 데이터베이스 구축에 있어 BERT의 적용 가능성을 시사하며, 배터리 소재 실험 연구자에게 있어서 이미 보고된 배터리 문서에 숨어있는 소재별 물성데이터를 직접 수집하는 데 번거로움을 덜어내는 기반을 마련하였다.

그런데 아쉬운 점으로, 모델의 과적합 문제와 데이터셋의 한계점은 향후 연구에서 주의 깊게 고려해야 할 중요한 과제임을 드러냈다. 향후 연구에서는 더 다양하고 균형 잡힌 데이터셋을 활용하여 모델의 일반화 능력을 강화하고, 모델의 해석 가능성을 향상시키는 방향으로 노력할 필요가 있다.

이 연구는 배터리 소재 물성값 추출을 위한 자연어처리 기술의 적용에 관한 새로운 방향을 제시하였으며, 이는 소재 과학자들이 보다 신속하고 정확한 소재 성능 분석을 수행할 수 있게 하는데 기여할 것이다.

최종적으로, 본 연구의 결과는 소재 과학 분야의 지속적인 발전과 혁신을 위한 기반을 제공하며, 이는 배터리 기술의 미래에 있어 중요한 영향을 미칠 것으로 기대된다.

참고 문헌

1. Majdi, H.S., Latipov, Z.A., Borisov, V. et al. Nano and Battery Anode: A Review. Nanoscale Res Lett 16, 177 (2021).

2. Wu Yuping, Elke Rahm, Rudolf Holze, Carbon anode materials for lithium ion batteries, Journal of Power Sources, Volume 114, Issue 2, 2003.

3. Zhang Hao, Yang Yang, Ren Dongsheng, Wang Li, He Xiangming, Graphite as anode materials: Fundamental mechanism, recent progress and advances, Energy Storage Materials, Volume 36, 2021.

4. Hiszpanski, A. M., Gallagher, B., Chellappan, K., Li, P., Liu, S., Kim, H., Han, J., Kalikihura, B., Buttler, D. J., & Han, T. Y.-J. Nanomaterial Synthesis Insights from Machine Learning of Scientific Articles by Extracting, Structuring, and Visualizing Knowledge. Journal of Chemical Information and Modeling, 60(6), 2876 - 2887. (2020).

5. Torkestani Akbari, J. An adaptive focused Web crawling algorithm based on learning automata. Appl Intell 37, 586 - 601 (2012).

6. Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.

7. Kim Donghwa, Seo Deokseong, Cho Suhyoun, Kang Pilsung, Multi-co-training for document classification using various document representations: TF - IDF, LDA, and Doc2Vec, Information Sciences, Volume 477, Pages 15-29 (2019).

8. Le. Quoc. V., Mikolov Tomas. "Distributed Representations of Sentences and

Documents." arXiv:1405.4053 [cs.CL], 2014.

9. Ikotun Abiodun M., Ezugwu Absalom E., Abualigah Laith, Abuhaija Belal, Heming Jia, K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, Information Sciences, Volume 622, Pages 178-210, 2023.

10. Salem Nema, Hussein Sahar, Data dimensional reduction and principal components analysis, Procedia Computer Science, Volume 163, Pages 292-299, 2019.

11. Zhou, H., Wang, F., & Tao, P. t-Distributed Stochastic Neighbor Embedding Method with the Least Information Loss for Macromolecular Simulations. *Journal of Chemical Theory and Computation*, 14(11), 5499 - 5510. (2018).

12. Chiche Alebachew & Yitagesu Betselot. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9, Article number: 10. (2022).

13. Shelar Hemlata, Kaur Gagandeep, Heda Neha & Agrawal Poorva, Named Entity Recognition Approaches and Their Comparison for Custom NER Model, Science & Technology Libraries, 39:3, 324-337, (2020).

14. Zhang, R., Kikui, G., & Sumita, E. Subword-based Tagging by Conditional Random Fields for Chinese Word Segmentation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*(pp. 193 - 196). (2006).

15. Li, J., Sun, A., Han, J., & Li, C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1), 50-70. (2020).

16. Siami-Namini, S., Tavakoli, N., & Siami Namin, A. The Performance of

LSTM and BiLSTM in Forecasting Time Series. In 2019 IEEE International Conference on Big Data (Big Data). IEEE. (2019).

17. Hameed Zabit, Garcia-Zapirain Begonya. Sentiment Classification Using a Single-Layered BiLSTM Model.In IEEE Access (Volume: 8). IEEE. (2020).

18. Huang, Z., Xu, W., & Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv:1508.01991 [cs.CL]. (2015).

19. Wei Hao, Gao Mingyuan, Zhou Ai, Chen Fei, Qu Wen, Wang Chunli, Lu Mingyu.Named Entity Recognition From Biomedical Texts Using a Fusion Attention-Based BiLSTM-CRF. In IEEE Access (Volume: 7). IEEE. (2019).

20. Rizk Rodrigue, Rizk Dominick, Rizk Frederic, Kumar Ashok, Bayoumi Magdy. A Resource-Saving Energy-Efficient Reconfigurable Hardware Accelerator for BERT-based Deep Neural Network Language Models using FFT Multiplication. In 2022 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE. (2022).

21. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]. (2018).