DU-도전학기 결과보고서

과제명	효율 및 정확도 향상을 위한 표절 검사 프로그램 개발		
참여자	성명	소속	학번
	박OO	컴퓨터공학전공	
	안OO	컴퓨터소프트웨어전공	
	강OO	컴퓨터공학전공	
지도교수 의견	상기 학생들은 주 2-3회 이상 함께 모여 도전학기를 성실히 수행한 결과, 텍스트 마이닝 기반의 고도화된 표절 탐지 기술을 개발하였습니다. 학생들이 개발한 기술은 대표적인 표절 검사 프로그램인 카피킬러가 탐지하지 못하는 고도화된 표절 행위를 탐지할 수 있는 고유 기술로서 해당 기술을 논문으로 작성하여 지난 5월 개최된 대한임베디드공학회 춘계학술대회에서 발표하였습니다. 학생들이 계획한 도전학기 목표를 모두 성공적으로 달성하였다고 생각하고, 도전학기 활동을 통해 인공지능, 파이썬프로그래밍 등 전공 교과목을 심화학습하는 유익한 시간이 되었길 바랍니다. (소속) 컴퓨터공학전공 (성명) 김지연 (서명 또는 날인)		

1. 도전 과제 내용

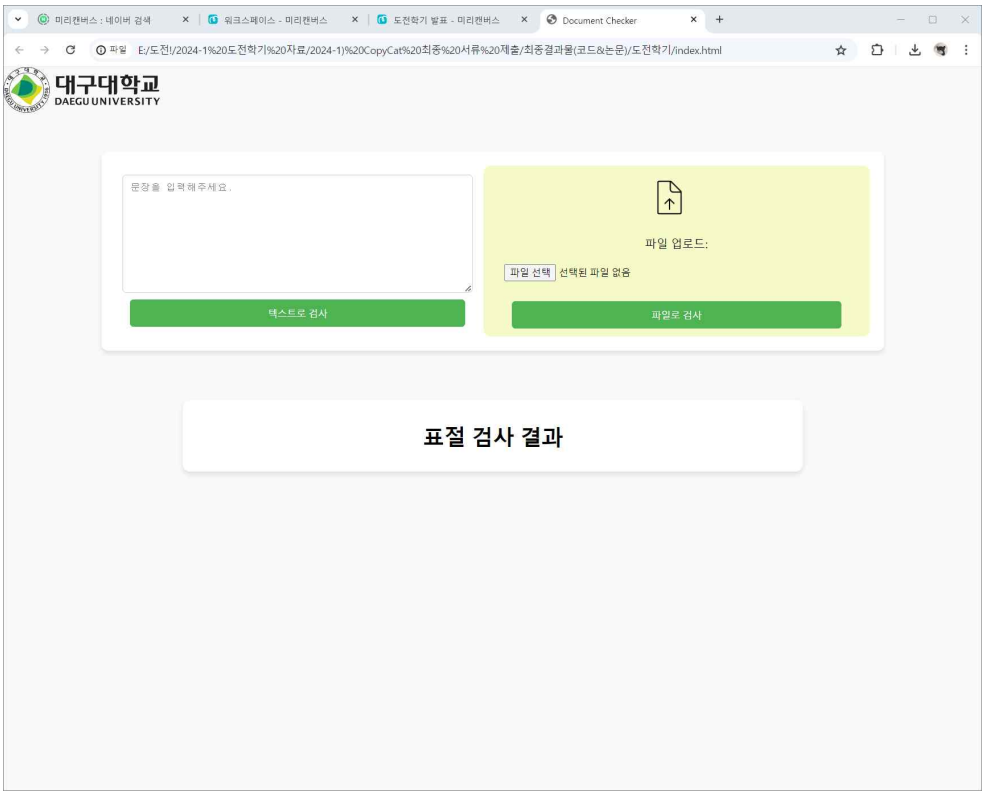
정보와 지식의 공유가 쉽게 이루어지고 디지털 기술의 발달로 정보에 대한 접근성이 증가함에 따라 지식 재산권을 침해하는 표절 행위가 더욱 심각해지고 있는 추세이다. 이러한 이유로 학문적 부정행위에 대한 경각심이 높아지고 있으며, 이러한 표절 행위호 부터 지적 재산권을 보호하기 위한 고도화된 표절 검사 시스템이 필요하다. 본 팀의 주제는 효율 및 정확도 향상을 위한 표절 검사 프로그램 개발으로 기존 제품보다 우수한 성능을 보이는 것에 대한 부분만을 강조하는 기술이라고 생각하였다. 본 팀은 이러한 기술들에서 학술적 가치를 탐구해보기 위해 텍스트마이닝을 중점으로한 표절 검사 시스템을 제시하게 되었고, 다양한 선행 연구를 찾아보았을 때, WMD(Word Mover's Distance) 특성에 맞게 변형 표절을 탐지하는 내용은 연구가 미비하다는 것을 확인하였다. 따라서 본팀은 '텍스트마이닝 기반의 고도화된 표절 검사 시스템'을 제시한다는 점에서 학술적 가치가 있다고 판단되었기에 이러한 내용을 기반해 의의를 두고 도전학기 프로젝트를 진행했다.

본 팀은 초기 도전학기 계획서 내용과 같게 데이터셋 구축을 위해 크롤러를 개발하여 국내 학술 DB인 'DBpia'에서 논문을 수집하여 데이터셋을 구축하였으며, 이를 기반으로 표절 검사 프로그램 개발을 진행할 수 있도록 하였다. 또한, 본 팀에서는 의도적으로 표절 검사 규칙을 우회하기 위해 문장 내 단어를 대체 단어로 변경하고, 문장 내의 단어 순서를 변경하는 우회 행위를 탐지하여 표절 여부를 검사하기 위해 WMD(Word Mover's Distance)를 사용하고, 뿐만 아니라 SHA-256 해시 알고리즘, 교집합 비율을 통해 문장과 문장간의 비교를 수행하였고, Doc2Vec 계산을 통해 문장-문장 비교를 수랭하였다. 또한, 웹사이트를 제작하여 본 팀에서 개발한 표절 검사 시스템을 활용할 수 있도록 진행하였다.

2. 도전 과제 수행 결과 및 성과

- 팀 공통 과제 수행 결과

본 팀은 도전학기 초기 계획대로 크롤러 개발을 통해 논문 데이터셋 구축을 진행하였고 목표했던 표절 검사 시스템 개발을 성공적으로 수행하였다. 표절 검사 시스템에서는 문장의 비교를 빠르게 수행하기 위한 해시 비교 알고리즘을 SHA-256으로 선정하여 진행하였고, WMD(Word Mover's Distance)를 통해 문장 내의 변형하여 표절한 부분도 탐지할 수 있도록 개발을 진행했다. 다음으로는 문서간의 비교를 위해 Doc2Vec을 활용하여 시스템 개발을 진행했고, 개발한 시스템의 코드를 통합하여 시각화 및 시스템 제공을 위한 웹사이트를 제작하였으며 문서 업로드 기능과 검사 기능 및 표절률을 나타내는 기능까지 성공적으로 구현했다.



3. 자기 평가

- 팀원 개별 과제 수행 결과

- 박OO : 도전 학기 초기에 계획했던대로 표절 검사 프로그램에서 비교 분석을 수행하기 위해 바른API를 사용하고 불용어 처리를 통해 단어간의 밴다이어그램을 제작하였다. 추가로 교집합 비율을 계산하여 특정값 이상일 경우 표절로 판단하는 알고리즘 및 해시값 비교를 위한 SHA-256 해시 알고리즘을 추가하였다. 또한 '카피킬러'와의 성능 비교를 진행하였으며 팀원과 함께 문서-문서 비교를 위한 Doc2Vec을 추가로 사용하여 개발을 성공적으로 진행했다.

- 안OO : 표절 검사 프로그램에서 비교 분석에서 관련 기술을 조사하고 데이터셋을 구축하기 위해 C# WinForm 애플리케이션 기반의 크롤러를 개발하고, 학술 DB에서 논문 데이터를

약 200개 수집하였다. 또한 Word2Vec 모델 계획 변경으로 인해 담당했던 해시 비교에서 WMD 측정값 계산 코드를 작성하였으며, Doc2Vec 모델을 팀원과 함께 선정 및 활용으로 비교 알고리즘을 개발하여 문서-문서 비교를 진행하고, 전체 코드 통합 및 최적화를 진행하여 웹 사이트 제작을 성공적으로 진행했다.

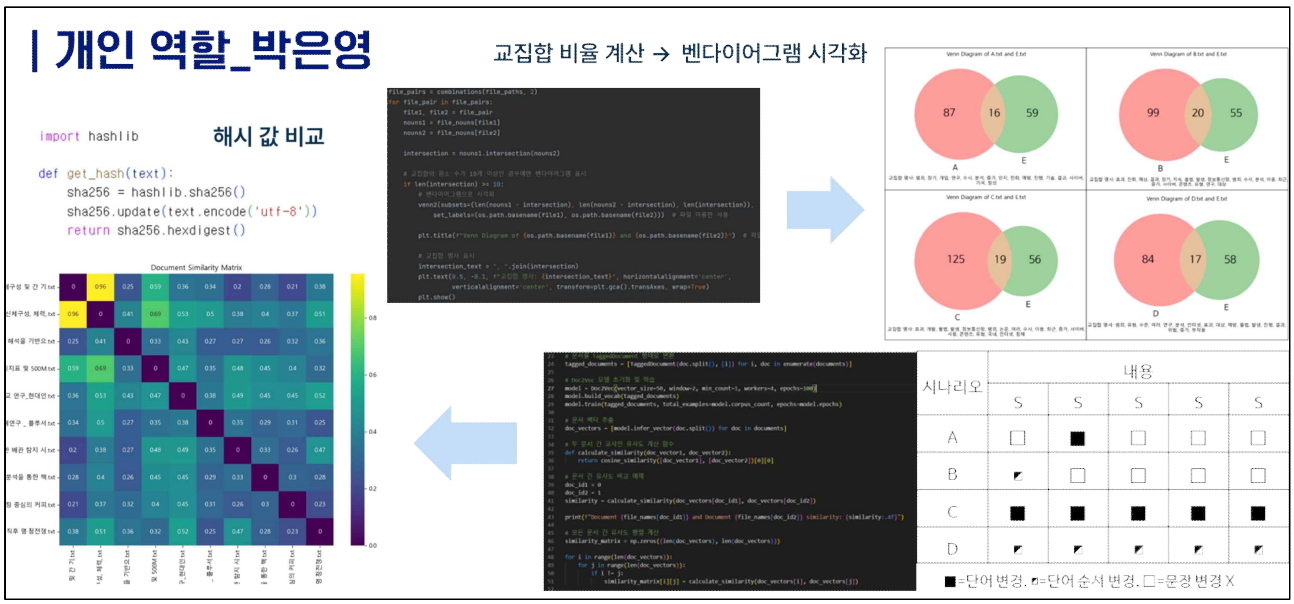
- 강OO : 표절 검사 프로그램 개발을 위해 Word2Vec 모델 알고리즘에 대한 조사 및 공부하였으며 조사를 진행하며 예시 코드 작성에서 문제를 발견하고 기존 계획인 중심성 계수에서 WMD로 변경하여 프로젝트를 진행하였다. 표절 탐지 시스템 성능 확인을 위해 단순히 문장을 복사한 '완전표절'과 문장의 단어를 변형시켜 표절한 '변형표절'과 같은 시나리오를 구성하여 팀원과 함께 성능 검증을 진행하였다. 나아가 자신의 관심 분야를 살려 개발한 시스템의 최적화 및 표절률 시각화를 위한 웹 사이트 제작을 위한 가상환경 설정과 사용자 인터페이스 구축을 성공적으로 진행했다.

▶ 팀원 모두 자신이 맡은 개인 과제를 도전 학기 초반에 세운 계획대로 잘 진행했기에 **<u>효율 및 정확도 향상을 위한 표절 검사 프로그램 개발</u>** 프로젝트 진행을 성공적으로 진행할 수 있었다.

4. 최종 결과물

- 개인(팀원별) 결과물

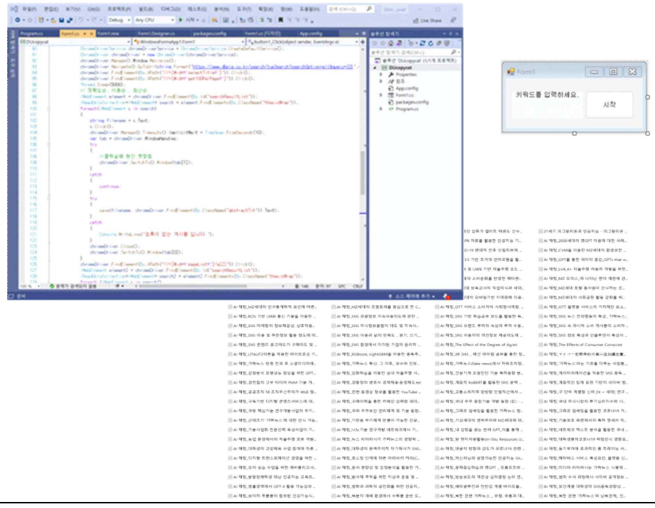
- 박OO : 표절 검사 시스템 개발 코드를 작성하며 시각화 진행, 팀원과 함께 시나리오 기반 카피킬러와의 성능 비교 진행



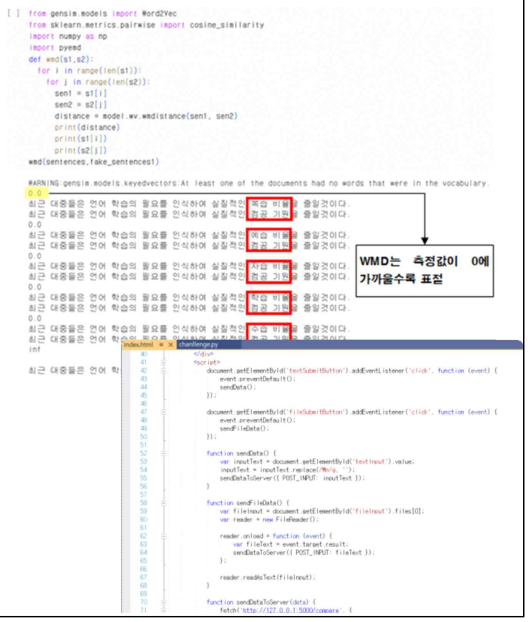
- 안OO : 논문 데이터셋 수집을 위한 크롤러 개발 진행 및 WMD 비교 코드 작성 & 팀원과 함께 웹 사이트 제작

| 개인 역할_안동휘

크롤러 개발 및 데이터 수집



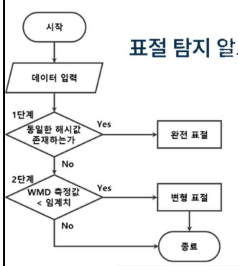
WMD 비교 코드 작성



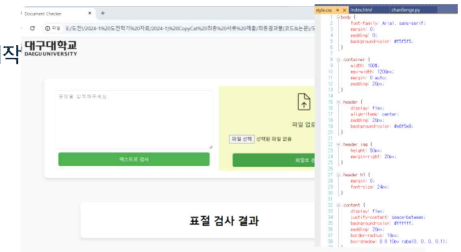
- 강OO : 변형 표절 및 완전 표절 탐지 알고리즘 제작, 카피킬러와의 성능 비교 진행, WMD 선정 및 구현 & 팀원과 함께 웹 사이트 제작

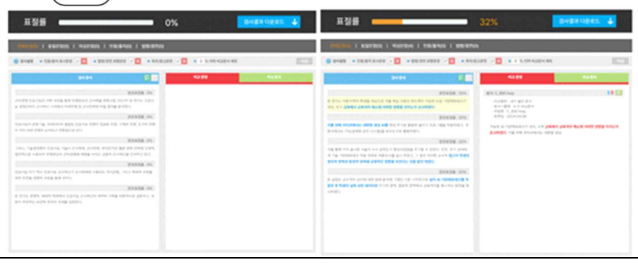
| 개인 역할_강동원

표절 탐지 알고리즘 제작

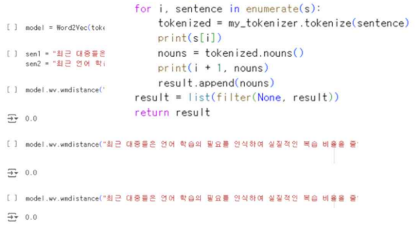


웹 사이트 프론트앤드 개발





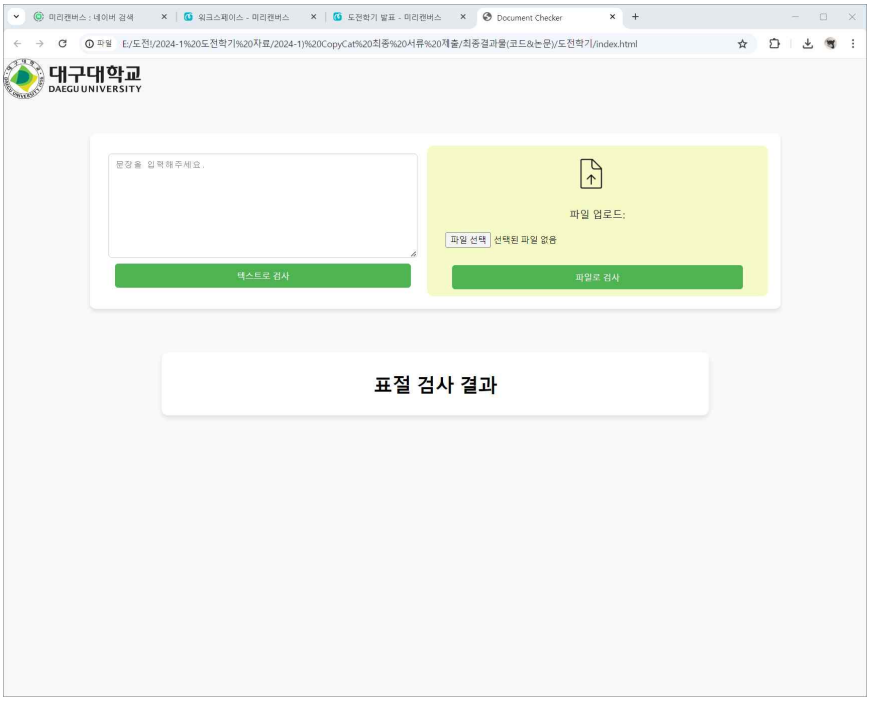
WMD 학습 코드 작성



카피킬러와의 비교 진행

시나리오	카피킬러 표절률	결과	
		해시	WMD
A	98%	완전 표절	변형 표절
B	92%	완전 표절	변형 표절
C	32%	-	변형 표절
D	0%	-	변형 표절

- **팀 공통 결과물 :** 표절 검사를 시행할 수 있도록 문서를 업로드하여 표절률을 나타내주는 웹 사이트(텍스트마이닝 기반), 학술대회 논문




```python
import sys
import baroungpy as brn
from baroungpy import Tokenizer
import re
import os
from gensim.models import Word2Vec
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
import hashlib

def get_hash(text):
    sha256 = hashlib.sha256()
    sha256.update(text.encode('utf-8'))
    return sha256.hexdigest()

def token_sentences(s):
    result = []
    for i, sentence in enumerate(s):
        tokenized = my_tokenizer.tokenize(sentence)
        print(s[i])
        nouns = tokenized.nouns()
        print(i + 1, nouns)
        result.append(nouns)
    result = list(filter(None, result))
    return result

def split_sentences(text):
    # 문장 구분자 기준으로 텍스트를 분리
    sentences = re.split('(?<=[.])\s+', text)
    print(sentences)
    return sentences
```

```python
sentences = all_files_text

tokend_sentences = token_sentences(sentences)

model = Word2Vec(tokend_sentences, vector_size=100, window=1, min_count=1, workers=4)
from flask import Flask, request
from flask_cors import CORS
app = Flask(__name__)
CORS(app)  # 모든 엔드포인트에 CORS를 적용

# POST 요청을 처리하는 엔드포인트
# POST 요청을 처리하는 엔드포인트
@app.route('/')
def index():
    return '서버가 정상적으로 실행 중입니다!'
@app.route('/compare', methods=['POST'])
def compare_sentences():
    data = request.json
    if data is None or 'POST_INPUT' not in data:
        return 'POST_INPUT을 전달해주세요.', 400

    POST_INPUT = data['POST_INPUT']
```

chanllenge.py 4 | style.css

E: > 도전! > 2024-1 도전학기 자료 > 2024-1) CopyCat 최종 서류 제출 >

```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f5f5f5;
}

.container {
    width: 100%;
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
}

.header {
    display: flex;
    align-items: center;
    padding: 20px;
    background-color: #e8f5e9;
}

.header img {
    height: 50px;
    margin-right: 20px;
}

.header h1 {
    margin: 0;
    font-size: 24px;
}

.content {
    display: flex;
    justify-content: space-between;
    background-color: #ffffff;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.upload-section,
.result-section,
.percentage-section {
    width: 30%;
    text-align: center;
    background-color: #f0f4c3;
    padding: 20px;
    border-radius: 10px;
}

.upload-section img {
    width: 50px;
    height: 50px;
}

.upload-section p,
.percentage-section p {
    font-size: 18px;
    margin: 10px 0;
}

.result-section h2 {
    margin-bottom: 20px;
    font-size: 20px;
}

.result-section p {
    font-size: 16px;
    margin: 5px 0;
}

.percentage-section h2 {
    font-size: 20px;
    margin-bottom: 10px;
}

.percentage-section p {
    font-size: 36px;
    color: red;
    margin: 10px 0;
}

.upload-section {
    width: 47.5%;
    text-align: center;
    background-color: #f0f4c3;
    padding: 20px;
    border-radius: 10px;
    height: 200px; /* 고정된 높이 추가 */
}

.text-input-section form,
.upload-section form {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    height: 100%;
}

.text-input-section textarea,
.upload-section input[type='file'] {
    width: calc(100% - 20px); /* 20px는 padding 값 */
    margin-bottom: 10px;
}

.text-input-section input[type='submit'] {
    width: calc(100% - 40px); /* 40px는 padding 값 */
}

.upload-section input[type='submit'] {
    width: calc(100% - 40px); /* 40px는 padding 값 */
    padding: 10px 20px;
    background-color: #4caf50;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-bottom: 20px; /* margin-top 대신 margin-bot
}

body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
}
```

텍스트마이닝 기반의 고도화된 표절 검사 시스템 개발

Development of an Advanced Plagiarism Inspection System based on Text Mining

†대구대학교 컴퓨터정보공학부
(Eun-Young Park, Dong-Hwi An, Dong-Won Kang, Jiyeon Kim)
(†Division of Computer and Information Engineering, Daegu University)

Abstract : 정보와 지식의 공유가 쉽게 이루어지고 디지털 기술의 발달로 정보에 대한 접근성이 증가함에 따라 지식 재산권을 침해하는 표절 행위가 더욱 심각해지고 있다. 학교 내 과제 및 논문 작성 시 원문을 변경하여 자신이 작성한 것처럼 활용하는 행위가 증가하면서 이러한 학문적 부정행위에 대한 경각심이 높아지고 있으며, 이러한 표절 행위로부터 지식 재산권을 보호하기 위한 고도화된 표절 검사 시스템 개발이 필요하다. 본 논문에서는 해시(Hash) 및 WMD(Word Mover's Distance) 기반으로 표절을 탐지하는 2단계 표절 검사 시스템을 개발하고, 대표적인 표절 검사 프로그램인 '카피킬러'와의 성능 비교를 통해 개발된 시스템의 성능을 검증한다. 의도적으로 표절 검사 시스템을 우회하는 4개 유형의 시나리오를 개발하여 표절 검사를 수행한 결과, 본 논문에서 개발한 표절 검사 시스템이 모든 시나리오를 정확하게 표절로 판단하는 것을 확인하였고, 2개 시나리오만 명확하게 표절로 판단한 '카피킬러'보다 우수한 성능을 가지는 것을 실험을 통해 검증하였다.

Keywords : Plagiarism, Text Mining, Hash, WMD(Word Mover's Distance)

Ⅰ. 서 론

표절은 타인의 저작물로부터 출처를 밝히지 않고 인용하거나 모방하여 자신의 창작물인 것처럼 사용하는 행위를 뜻하며, 타인의 저작물 가치를 훼손함으로써 심각한 학문적, 사회적 문제를 야기한다. 2022년 대학 내 과제물 중, 약 46%가 과제물 30% 이상 표절한 '표절 위험' 군으로 조사되었다[1]. 이러한 학문적 부정행위를 방지하기 위해 다양한 표절 검사 프로그램이 등장하였지만, 표절 프로그램을 우회하는 다양한 표절 시도를 탐지하는 데에는 한계가 있다. 예를 들어, 대표적인 표절 검사 프로그램인 '카피킬러'의 경우, 한 문장에서 6어절 이상 일치하는 경우 표절이라고 판단한다[2]. 그러나 표절 프로그램을 우회하기 위하여 의도적으로 한 문장에서 6어절 미만으로 표절하거나, 한 문장 내의 단어 순서를 바꾸는 경우, 또는 비슷한 단어로 대체하여 작성하는 경우에는 표절로 탐지되지 않는다. 따라서 이러한 표절 우회 행위도 탐지할 수 있는 진화된 표절 탐지 기술 개발이 필요하다.

*Corresponding Author (jyk@daegu.ac.kr)
김지연: 대구대학교 컴퓨터정보공학부

본 논문에서는 문장 단위의 표절 검사뿐 아니라, 다수의 문장에서 문맥을 파악하고, 단어 임베딩 공간에서 유사도를 계산하여 문장의 거리를 측정하는 표절 검사 기술을 제안한다. 문장 단위의 표절 검사를 위해서는 문장 전체의 해시(Hash)값 비교를 통해 단순하게 문장을 표절한 완전 표절을 검출하고, WMD(Word Mover's Distance)를 활용하여 문장의 단어를 변형하거나 순서를 바꾸는 등의 변형 표절을 탐지한다. 또한, 본 논문에서는 제안된 기술의 설계 및 실행을 검증하기 위하여 표절 검사 시스템을 개발하고, 시스템의 성능을 평가하기 위하여 학술 데이터셋을 수집하는 크롤러(Crawler)를 개발한다.

본 논문의 구성은 다음과 같다. 2장에서는 표절 탐지 기술 동향을 살펴보고, 3장에서 본 논문에서 제안하는 표절 검사 시스템 및 학술 데이터 크롤러를 개발한다. 4장에서는 개발된 시스템의 성능을 실험을 통해 분석하고, 5장에서 결론 및 향후 연구를 제시한다.

Ⅱ. 관련 연구

본 장에서는 기존에 수행된 표절 방지를 위한 관련 연구 동향을 살펴본다. 내부 DB만을 활용하는 기존의 표절 검사 기술을 개선하여 외부 웹 데이터를