

# 외형적 특징을 활용한 유사 문서 검색 시스템

## 프로젝트 요약

보고 날짜	팀 이름	작성자
2023년 9월 30일	DFRC-AI	조서연, 김준호, 박정흠

## 보고서 목차

<b>1 기술 경연 대회 주제</b>	<b>2</b>
1.1 기술 요약 (초록)	2
1.2 기술 영향력 평가	3
1.2.1 연구 목표	3
1.2.2 기술의 우수성	3
<b>2 배경지식</b>	<b>4</b>
2.1 외형적 유사성	4
2.2 Microsoft word 2007+ 문서 구조 (.DOCX)	5
<b>3 외형적 특징을 활용한 유사 문서 검색 방법</b>	<b>7</b>
3.1 개요	7
3.2 특징 추출	7
3.3 데이터 전처리	8
3.3.1 단어 전처리	9
3.3.2 단어 리스트 전처리	9
3.3.3 데이터 스케일링 : 표준화(Standardization)	10
3.3.4 데이터 스케일링 : 정규화(Nomalization)	11
3.4 유사 문서 검색	11
<b>4 평가</b>	<b>13</b>
4.1 평가 데이터 세트	13
4.2 평가 지표	15
4.3 평가 결과	16

<b>5 고찰 .....</b>	<b>17</b>
5.1 연구 결과에 대한 고찰 .....	17
5.2 특징 선정 기준 .....	19
<b>6 결론 .....</b>	<b>23</b>
<b>7 부록 1: 도구 사용 방법.....</b>	<b>24</b>
<b>8 부록 2: ms-word 외형적 서식 목록.....</b>	<b>29</b>
8.1 document.xml 특징 .....	29
8.2 style.xml 특징 .....	30
8.3 fonttable.xml 특징 .....	35
8.4 header#N.xml/footer#N.xml 특징 .....	35

## 1 기술 경연 대회 주제

### 1.1 기술 요약 (초록)

#### [ 문서의 외형적 구조를 결정하는 서식 데이터를 이용한 검색 시스템 ]

업무의 디지털화와 데이터 저장 기술의 발전으로 인해 디지털 저장 매체에 저장되는 문서 파일의 양이 증가하는 추세다. 문서 파일은 개인 또는 조직의 활동이나 업무를 기록하는 용도로 생성하기 때문에 디지털 포렌식 수사 시 필수적으로 분석해야 할 대상이다. 문서 파일이 범죄를 입증하는 증거 또는 실마리가 되는 경우, 디지털 포렌식 수사 과정에서 대상 문서 파일과 연관된 다른 문서 파일을 선별하는 선별 압수를 수행하고 있다. 따라서, 범죄 사건과 관련된 문서를 선별하기 위해 대상 문서를 질의하고 해당 문서에 대하여 저장장치 내 유사 파일 검색 방법이 필요하다. 종래의 포렌식 수사의 문서 선별 압수는 키워드 중심으로 연관 문서를 검색하는 방식을 사용했다. 키워드 중심의 문서 선별 방법은 유사한 서식이 적용되어 외형적으로는 매우 유사하나, 내용이 다른 문서들은 식별하지 못하는 한계점이 있다.

본 기술에서는 문서 파일을 대상으로 서식 데이터의 저장 구조를 파악하고 이를 활용하여 유사 서식 문서를 검색 및 군집화하는 방법론을 제안한다. 그리고 Microsoft WORD 2007+(.docx) 문서 형식을 방법론에 적용하여 제안한 방법론의 실효성을 검증한다.

## 1.2 기술 영향력 평가

### 1.2.1 연구 목표

문서 내부 구조 분석을 통해 문서의 서식 정보를 추출하고 이 정보를 활용하여 외형적으로 유사한 문서를 검색하는 방법 및 시스템 제공한다.

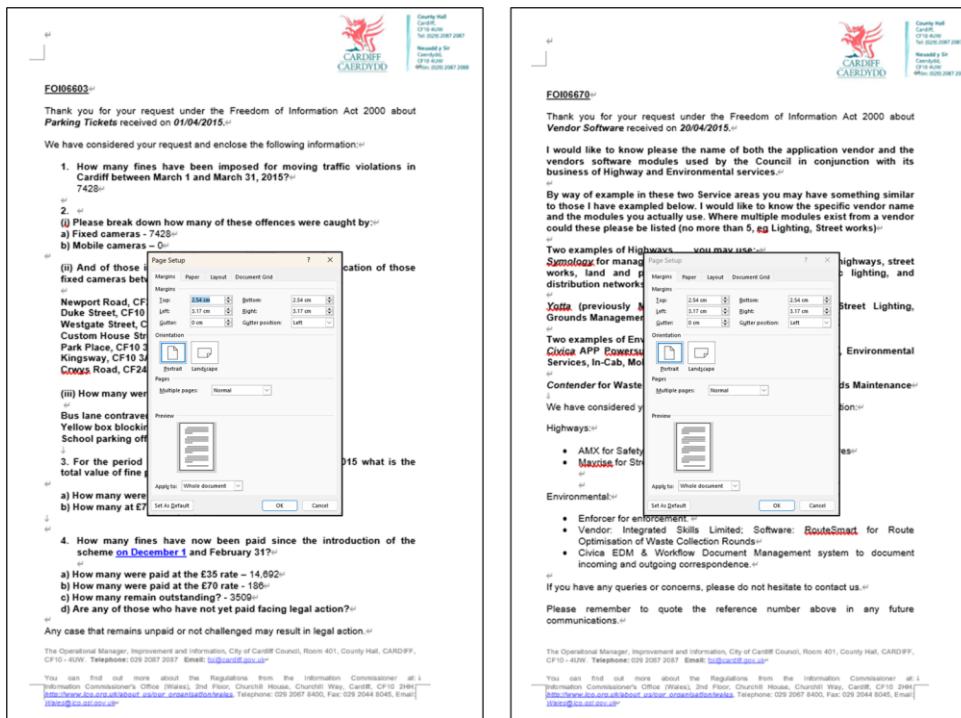
### 1.2.2 기술의 우수성

1. 비슷한 외형적 특징을 갖는 문서를 탐지하는 기술은 동일 집단 또는 밀접한 관계가 있는 집단에서 생성한 문서를 효과적으로 탐지할 수 있도록 함으로써, 디지털 포렌식 관점에서 특정 사건과 관련된 문서를 선별 수집하는 활동을 지원할 수 있음
2. 종래의 키워드 기반 문서 검색 시스템과 달리, 문서의 내용을 고려하지 않고 외형적 구조를 결정하는 서식 구조를 활용하므로, 디지털 포렌식 분야의 전통적인 검색 기능을 확장하는 데에 의의가 있음
3. 사용자 설정 서식이 존재하는 모든 문서 파일 형식에 확장하여 적용이 가능함
4. 문서 유사성(document similarity) 및 문서 이해(document understanding) 연구에 새로운 관점을 제시함  
4-1 기존 문서의 레이아웃(스타일, 여백)을 유추하는 시도들은 주로 인공지능 시스템을 활용하고 있지만, 문서 내에 적용된 실제 설정 서식 값을 추출하여 활용함으로써 인공지능 시스템에서 추론한 문서 레이아웃 정보보다 정확하고 설명 가능한 데이터를 기반으로 한 판별 시스템의 구축이 가능함
- 4-2. 인공지능 기반 시스템 구축 시 활용할 수 있는 새로운 특징 정보를 제시함
5. 텍스트에 독립적으로 문서의 외형적 구조에 기반한 유사도 측정 방법으로, 언어에 종속적이지 않기 때문에 다국어/동의어 처리를 위한 별도의 과정이 불필요함

## 2 배경지식

### 2.1 외형적 유사성

본 연구에서 다루는 문서의 외형적 유사성이란, 사용자가 문서 편집 프로그램에서 설정한 서식 설정 값의 유사성을 말한다. 문서의 내용을 시각적으로 구조화하기 위해, 사용자는 문서를 작성할 때 문서 편집 프로그램에서 제공하는 서식 설정(폰트, 문단, 스타일, 머리말, 꼬리말, 문서 여백, 테두리 등)을 지정한다. 이렇게 설정된 서식 값들은 문서 내에서 텍스트의 외형적 위치 및 스타일을 결정한다. [그림 1]의 예시와 같이 본 연구에서는 시각적으로 문서의 구조가 유사하고, 문서 내 정의된 사용자 서식 값이 동일한 것을 외형적 유사성이 높다고 판단한다.



[그림 1] 외형적 유사성이 높은 문서 예시

외형적 유사성이 높은 문서는 같은 조직의 문서 양식을 사용했을 가능성이 높다. 통상, 동일한 집단이나 조직에서는 보고서, 보도자료 같은 문서를 작성할 때, 조직 문서의 통일성과 내용 전달의 효과를 증대하기 위해 지정된 문서 양식을 공유하여 작성한다. 이에 착안하여 본 연구는 역으로 문서 양식을 만들기 위해 사용자가 설정한 문서 서식을 추출하여 외형적 유사도가 높은 문서를 식별하고, 더 나아가 문서 출처 집단을 구별한다.

## 2.2 MICROSOFT WORD 2007+ 문서 구조 (.DOCX)

Microsoft 는 2007 년 XML 과 바이너리 파일의 조합으로 이뤄진 Open Office XML(OOXML) 구조를 개발하였고, OOXML 구조가 적용된 Microsoft Office 2007+(.docx)를 출시했다. Microsoft Office 의 이후 버전에서도 OOXML 구조가 계속 사용되고 있다. 가장 최근 버전인 Microsoft Word 365 도 동일한 구조를 사용하고 있다. 데이터 저장 형태는 XML 파일 모음으로 구성된 ZIP 파일 형태이며, 데이터 구조화하여 표현하는데 사용되는 메타데이터 파일과 문서의 실제 데이터를 담고 있는 XML 파일 모음으로 구성된다. Open Office 문서를 이루는 데이터 구성과 XML 파일들은 OPC(Microsoft Open Packaging Conventions<sup>1</sup>)를 기반으로 설계되었다. OPC 아카이브에서는 각 데이터 파일을 part 라고 하며, 각 XML 은 이 part 에 해당한다. 문서에 포함된 part 인 XML 파일 목록은 저장소 루트에 있는 “[Content\_Types].xml”에 저장된다. 이 파일은 문서를 구성하는데 필요한 XML 파일과 문서에 포함된 이미지, 미디어 유형의 정보를 포함하기 때문에, Word 파일의 설정과 내용에 따라 “[Content\_Types].xml” 파일의 내용이 달라진다. “[Content\_Types].xml”에 목록으로 정리된 XML 과 이미지, 미디어 등 문서를 구성하는 파일들의 참조 관계는 각 파일에 고유 ID 를 부여하여 \_rels 파일에 정리된다.

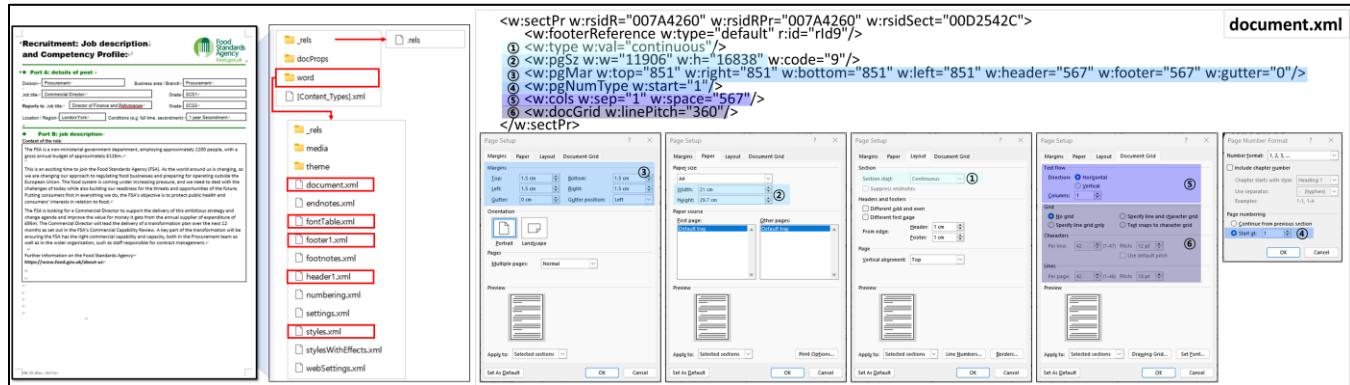
본 연구에서 주요 특징으로 사용하고 있는 서식 데이터 또한 XML 파일 형태로 관리된다. 본 연구는 document.xml, style.xml, fontTable.xml, footerN.xml/headerN.xml 4 개의 파일에 저장된 문서 서식 정보를 활용한다. 각 XML 파일은 하나의 루트 요소(element)로부터 파생되어 데이터를 설명하는 메타 정보인 요소(element)와 속성(attribute)을 가지며 구조화된 형태로 데이터를 저장한다. 각 요소(element)들은 다시 자식 요소(child element)와 자식 요소(child element)의 속성(attribute)을 가지며 확장해 나갈 수 있다. [그림 2]는 document.xml 를 하나의 예로 Word(.docx) 파일을 구성하는 xml 파일이 데이터를 저장할 때 갖는 구조를 설명한다. 각 xml 파일들은 [그림 2]처럼 하나뿐인 루트(document) 요소(element)로부터 파생되어 자식 요소(child element)를 가지며 트리 형태로 연결된 계층적 구조로 데이터를 저장한다. 각 자식 요소(child element)들은 요소(element)를 설명하는 속성(attribute)과 함께 정의될 수 있다. 예를 들어 문서의 page size 를 설정하는 [w:pgsz] 요소(element)는 넓이와 높이에 해당하는 속성(attribute)인 [w:w]와 [w:h] 로 정의된다.

```
<w:document xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas" ...
<w:body>
  <w:p w14:paraId="516733CA" w14:textId="77777777" w:rsidR="00692CF8" w:rsidRDefault="00692CF8 "/>
  <w:sectPr w:rsidR="00692CF8">
    <w:headerReference w:type="default" r:id="rId6" />
    <w:pgSz w:w="11906" w:h="16838" />
      <w:pgMar w:top="1701" w:right="1440" w:bottom="1440" w:left="1440" w:header="851" w:footer="9 92" w:gutter="0" />
      <w:cols w:space="425" />
      <w:docGrid w:linePitch="360" />
    </w:sectPr>
  </w:body>
</w:document>
```

[그림 2] DOCX XML 예시

<sup>1</sup> Microsoft. Open packaging conventions fundamentals. <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/opc/open-packaging-conventions-overview>, accessed on Sep. 2023

문서 편집기에서 다양한 서식을 설정할 수 있는데, [그림 3]은 문서 편집기 내 설정 옵션과 XML 내 저장된 설정 정보에 대한 예시이다. [그림 3]에 표시한 6 개의 설정 정보에 대한 설명은 다음 리스트와 같다.



[그림 3] Word 문서 zip 구조와 document.xml GUI 매핑 관계

### 1. element (w:type)

속성 [val]의 값인 continues는 사용자가 page setting 서식 창에서 section start로 지정한 값인 (continues)를 저장한다.

### 2. element (w:pgsz)

해당 요소는 Word 페이지의 세로와 가로의 크기를 정의한다. 속성 [w:w]는 페이지의 폭을 속성 [w:h]는 페이지의 높이를 정의한다.

### 3. element (w:pgMar)

페이지의 위 아래 양옆의 여백을 정의한다. 속성 [w:top]은 위, 속성 [w:right]은 오른쪽, 속성 [w:bottom]은 아래, 속성 [w:left]는 왼쪽, 속성 [w:header]는 머리말, 속성 [w:footer]는 꼬리말, 속성 [w:gutter]는 제본 여백을 정의한다.

### 4. element (w:pgNumType)

속성 [w:start]는 page Number Format 서식 창에서 page numbering start at에 설정된 값이 들어간다.

### 5. element (w:cols)

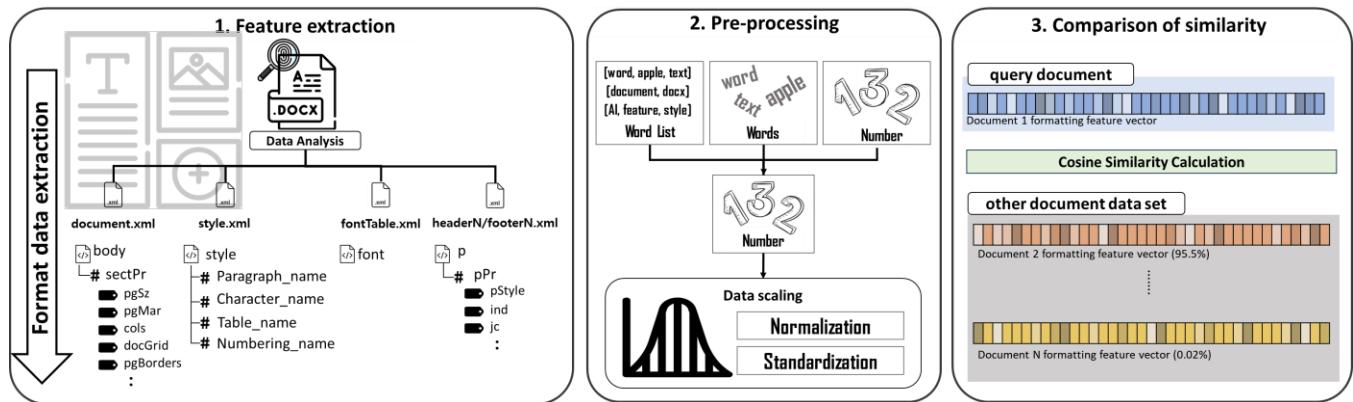
속성 [w:sep]는 문단의 열(column)의 개수를 지정하고 속성 [w:space]는 열(columns)간의 여백을 의미한다

### 6. element (w:docGrid)

해당 서식은 문서의 한 줄 당 글자 수와 한 페이지에 들어가는 줄 수를 설정할 수 있으며, 설정된 값에 따라 여백은 자동으로 조정된다. 속성 [w:linepitch]는 줄 간의 여백을 의미한다.

### 3 외형적 특징을 활용한 유사 문서 검색 방법

#### 3.1 개요



[그림 4] 서식 기반 문서 유사도 측정 절차

이번 절에서는 서식 데이터를 활용한 유사 문서 검색 방법론에 대해 설명한다. [그림 4]는 Microsoft Word 2007+ (.docx) 파일을 대상으로 외형적으로 유사한 문서를 검색하는 예시이다.

첫 번째 단계는 특징을 추출하는 것으로 Word (.docx) 파일의 데이터 저장 구조를 분석하여 본 연구에서 주요하게 다루고 있는 서식 데이터를 식별하고 추출한다.

두 번째 단계는, 추출된 각 서식 정보의 데이터 타입을 문서 유사도 계산 알고리즘에 적용할 수 있는 형태로 변환하는 전처리 과정이다. Word (.docx) 문서에서 추출된 데이터 형태는 정수형/실수형, 단어, 단어 리스트로 구분되며, 각 서식 데이터의 타입을 문서 유사도 계산 알고리즘에 적용할 수 있는 정수형/실수형으로 변환한다. 각 서식 특징들의 단위와 범주를 맞추기 위해 데이터 스케일링(Data Scaling)을 진행한다. 이 과정에는 표준화(Standardization)와 정규화(Normalization) 과정이 포함된다. 이렇게 문서를 변환하면 문서는 서식 설정을 추출하여 만든 특징 벡터(Feature Vector)로 표현된다.

세 번째 단계로, 유사도 비교 알고리즘 통해 질의 문서의 특징 벡터(Feature Vector)와 다른 문서 특징 벡터(Feature Vector)의 유사도를 계산한다. 이 과정을 통해 질의 문서와 유사한 서식 특징을 가진 문서에 대한 검색 결과를 제공한다.

#### 3.2 특징 추출

특징 추출 단계에서는, 문서 유사도를 계산할 때 사용할 서식 특징들을 식별하고, 추출한다. 서식 특징을 추출하기 위해서는 문서의 데이터 저장 구조를 분석하여 특징으로 될 수 있는 데이터를 선정하는 것이 선행되어야 한다. Word (.docx) 문서 파일의 경우 XML 파일을 PKZIP 형식으로 압축하여 저장하는 OOXML 구조로, 압축을 해제하면 문서를 구성하는 여러 XML 파일을 볼 수 있다.

본 연구에서 다루는 사용자 설정 서식 정보는 텍스트에 개별 적용된 서식 설정 값이 아닌 문서 전체에 지정된 사용자 설정 서식 값이다. 텍스트에 적용된 서식 데이터만을 고려하게 되면, 배치구조가 같아도 내용에 종속적으로 유사도가 변화하기 때문에 외형적 유사도 측정이 불가능하다. 따라서 특정 텍스트가 아닌 이에 독립적으로 문서 외형적 구조에 영향을 주는 서식 값을 고려한다.

사용되는 문서 서식 특징의 종류와 개수는 데이터 세트에 따라 유동적으로 추출된다. 모든 서식이 문서에서 필수적으로 설정되어야 하는 값이 아니기 때문에 사용자의 설정에 따라 추출되는 서식 특징의 종류와 개수가 다르다. 따라서 본 연구에서는 제안한 특징 모두를 추출하되, 그중 실제 문서에 적용된 특징들만 유동적으로 활용하여 유사도를 측정한다.

유사 문서 검색에 활용할 서식 값은 document.xml, style.xml, fontTable.xml, headerN.xml/footerN.xml에 저장된다. 각 XML에는 문서의 페이지 크기, 여백, 테두리, 스타일 리스트, 글꼴 리스트, 머리말/꼬리말 서식 등의 설정 값이 저장되어 있다. 각 파일에서 추출한 서식 값의 세부 사항은 부록에 [표 9-14]로 정리했다.

**/word/document.xml** : Microsoft Word 문서의 실제 내용을 포함하는 XML 파일이다. 사용자가 입력한 텍스트 데이터와 그에 따른 문단의 서식, 글꼴, 크기, 스타일과 같은 서식 설정 값, 그림, 표, 등 문서를 구성하는 요소들이 이 파일에 포함된다. [표 9]

**/word/style.xml** : Microsoft Word 문서에서 사용되는 스타일을 정의하는 XML 파일이다. 문서의 서식, 글꼴, 크기, 간격 등을 지정하는 스타일 요소들이 포함된다. 문서 전체에 적용되는 기본 스타일부터 문서 내 정의된 폰트 스타일까지 다양한 스타일이 포함된다. [표 10][표 11][표 12]

**/word/fontTable.xml** : 문서에 정의된 글꼴 정보를 포함하는 XML 파일이다. 문서 내 사용되는 다양한 글꼴들의 정보와 속성들이 여기에 정의된다. 각 글꼴의 이름, 크기, 스타일 등이 이 파일에 포함된다. [표 13]

**/word/headerN.xml, /word/footerN.xml** : 문서의 머리말(header)과 꼬리말(footer) 내용을 정의하는 XML 파일이다. 각 파일은 문서의 각 섹션에 대한 머리말과 꼬리말 내용을 정의한다. 머리말(header)과 꼬리말(footer)에 들어갈 텍스트, 그림, 페이지 번호 등의 이 파일에 포함된다. N은 섹션 번호를 나타내며, 여러 섹션에 대한 머리말과 꼬리말이 따로 정의된다. [표 14]

### 3.3 데이터 전처리

데이터 전처리 단계에서는, 데이터의 형태를 유사도 알고리즘을 적용할 수 있는 정수형, 실수형 단어로 변환하고, 데이터 스케일링(Data Scaling) 과정을 통해 각 특징의 단위와 범주를 일관되게 한다.

추출된 서식 데이터는 정수형, 실수형, 단어, 단어 리스트 등으로 데이터 형식이 다르다. 유사도 측정 알고리즘의 입력으로 사용하기 위해서는 추출된 서식 데이터 타입을 int/float/double 형으로 변경해야 한다. 따라서 추출된 특징의 형식이 단어, 단어 리스트로 식별된 경우는 데이터 전처리 과정이 필요하다. 데이터 형식을 변경한 서식 데이터는 각 데이터의 단위와 범위가 다르므로, 데이터의 단위와 범위를 일관성 있게 반영하기 위해 Data Scaling 을 한다. 표준화(Standardization)와 정규화(Normalization)가 과정이 이에 해당한다.

### 3.3.1 단어 전처리

범주형 단어 데이터의 전처리는 사용자가 설정할 수 있는 값에 고유한 ID를 부여하여 정수형 데이터로 변환하는 방법을 사용한다. 특정 범위로 제한된 단어가 사용자 서식 설정값인 특징이 여기에 해당한다. 예로, 페이지 방향과 같이 세로/가로의 설정에 따라 해당 필드의 값이 (portrait/landscape)로 설정되는 경우를 들 수 있다. 이때, 사용자가 설정할 수 있는 값이 2 가지이므로 각 설정값에 고유한 ID인 0/1로 부여하여 정수형 값으로 변환한다. [그림 5]는 페이지 방향을 추출하고 전처리한 것으로 범주형 데이터 전처리 예시다. 기본값인 portrait로 설정될 때는 데이터 값이 xml에 직접 명시되지 않아 Null 값으로 추출된다.

1	sectPr_pgSz_orient	1	sectPr_pgSz_orient
2		2	-1
3		3	-1
4		4	-1
5		5	-1
6		6	-1
7	landscape	7	0
8		8	-1
9		9	-1
10		10	-1
11		11	-1
12		12	-1
13		13	-1
14		14	-1
15		15	-1
16		16	-1
17		17	-1
18		18	-1
19		19	-1
20		20	-1
21		21	-1
22		22	-1
23		23	-1
24		24	-1
25		25	-1
26		26	-1
27		27	-1
28	landscape	28	0
29		29	-1

[그림 5] 범주형 데이터 전처리

### 3.3.2 단어 리스트 전처리

단어 리스트의 전처리는 단어에 고유 ID 값을 부여한 후, PCA를 이용하여 하나의 실수형으로 변환하는 방법을 사용한다. 예로, 문서 내 포함된 글꼴 명이나, 스타일 명을 들 수 있다.

단어 리스트의 원소인 단어는 단어 전처리 과정과 유사하게 단어 명칭에 고유 ID 값을 부여하여 정수형으로 변환한다. 하지만, 문서에 포함된 단어 리스트에는 서로 다른 문서라고 하더라도 동일한 글꼴이나 스타일 명 단어가 포함될 수 있다. 따라서 문서 데이터 세트에 존재하는 모든 단어 리스트의 원소 단어를 모아, 단어마다 고유 ID를 부여한다. 이렇게 생성된 단어-정수 인덱스 맵핑 사전은 빈도수가 높은 단어일수록 작은 인덱스 번호가 부여된다. 맵핑 사전을 이용하여 단어 리스트의 원소인 단어를 고유 ID로

치환하고, 단어 리스트를 정수형 리스트로 변환한다. [그림 6]의 첫 번째와 두 번째 그림은 이런 변환을 나타낸다.

변환된 정수형 리스트는 고차원 데이터이므로, 차원 축소 기법을 이용하여 저차원인 하나의 실수형 값으로 표현한다. 여기서 차원은 정수형 리스트의 크기에 해당한다. 각 문서에 포함된 단어의 개수가 다르므로 정수형 리스트의 크기, 즉 차원의 크기가 모두 상이하다. 일괄적으로 차원의 크기를 맞추기 위해 가장 큰 차원을 가진 정수형 리스트를 기준으로 패딩을 넣어 차원을 확장한다. 이후 고차원의 데이터를 축소하여 일괄적으로 저차원, 즉 하나의 실수형 데이터로 변환하기 위한 기법인 차원 축소 기법을 이용한다. 대표적인 예로 PCA(Principal Component Analysis) 기법이 있다.

1 paragraph_name	1 paragraph_name	1 paragraph_name
2 ['Normal', 'heading 1', 'heading 2', 'heading 3', 'heading 4', 'heading 5', 'heading 6', 'heading 7', 'heading 8', 'header', 'f	2 [1, 6, 7, 8, 12, 14, 15, 18, 19, 3, 2, 143, 90, 86, 39, 38, 41, 20, 28,	2 [1863811.791
3 ['Normal', '바탕글', '본문(신명조10)', '본문(종고딕10)', '작은제목(종고딕15)', '중간제목(옛체20)', '큰제목(검고딕20)', '큰제목	3 [1, 205, 313, 205, 314, 315, 316, 317, 318, 206, 319, 206, 320	3 15106686.07
4 ['Normal', '바탕글', 'Body Text', '개요 1', '개요 2', '개요 3', '개요 4', '개요 5', '개요 6', '개요 7', '쪽 번호', '머리말', '각주', '그림	4 [1, 20, 9, 73, 74, 78, 72, 71, 79, 80, 68, 50, 67, 92, 93, 94, 95, 278,	4 512543.163
5 ['Normal', 'Balloon Text']	5 [1, 5]	5 -1075834.163
6 ['Normal', 'No Spacing', 'header', 'footer', 'Balloon Text']	6 [1, 23, 3, 2, 5]	6 -1049428.517
7 ['Normal', 'List Paragraph', 'Normal (Web)', 'header', 'footer']	7 [1, 4, 1, 11, 3, 2]	7 -1061114.753
8 ['Normal', 'header', 'footer', 'Balloon Text']	8 [1, 3, 2, 5]	8 -1072382.719
9 ['Normal', 'header', 'footer']	9 [1, 3, 2]	9 -1075148.361
10 ['Normal', 'HTML Preformatted']	10 [1, 69]	10 -1068475.726
11 ['Normal', 'List Paragraph', 'header', 'footer', 'Normal (Web)', 'ecxmsnormal', 'Balloon Text']	11 [1, 4, 3, 2, 1, 11, 431, 5]	11 -1000202.746
12 ['Normal', 'List Paragraph', 'Balloon Text']	12 [1, 4, 5]	12 -1074675.519
13 ['Normal', 'Normal (Web)', 'txt13', 'header', 'footer', 'Balloon Text']	13 [1, 1, 11, 144, 3, 2, 5]	13 -916426.522
14 ['Normal', 'Balloon Text']	14 [1, 5]	14 -1075834.163
15 ['Normal']	15 [1]	15 -1076421.981
16 ['Normal', 'heading 1', 'heading 2', 'heading 3', 'Balloon Text', 'List Paragraph', 'No Spacing', 'header', 'footer', 'Subtitle']	16 [1, 6, 7, 8, 5, 4, 23, 3, 2, 22]	16 -1013618.807
17 ['Normal', 'Balloon Text']	17 [1, 5]	17 -1075834.163
18 ['Normal', 'heading 1', 'heading 2', 'heading 3', 'heading 4', 'heading 5', 'heading 6', 'heading 7', 'heading 8', 'heading 9']	18 [1, 6, 7, 8, 12, 14, 15, 18, 19, 24, 10, 97, 2, 5, 187, 87, 9, 16, 27, 4	18 5833475.963
19 ['Normal', 'List Paragraph']	19 [1, 4]	19 -1075949.139
20 ['Normal', 'List Paragraph']	20 [1, 4]	20 -1075949.139
21 ['Normal']	21 [1]	21 -1076421.981
22 ['Normal', 'List Paragraph']	22 [1, 4]	22 -1075949.139
23 ['Normal', 'List Paragraph', '바탕글', 'Balloon Text', 'header', 'footer']	23 [1, 4, 20, 5, 3, 2]	23 -1046766.329
24 ['Normal', 'List Paragraph', 'header', 'footer']	24 [1, 4, 3, 2]	24 -1071824.832
25 ['Normal']	25 [1]	25 -1076421.981
26 ['Normal', 'List Paragraph', 'Balloon Text']	26 [1, 4, 5]	26 -1074675.519
27 ['Normal', 'Balloon Text']	27 [1, 5]	27 -1075834.163
28 ['Normal', 'Balloon Text', 'List Paragraph', 'header', 'footer']	28 [1, 5, 4, 3, 2]	28 -1066849.614
29 ['Normal', 'heading 1', 'Normal (Web)', 'header', 'footer', 'List Paragraph', 'Balloon Text']	29 [1, 6, 1, 11, 3, 2, 4, 5]	29 -1051756.185
30 ['Normal', 'heading 1', 'heading 2', 'header', 'footer', 'Normal (Web)', 's0', '제1조', '바탕글', '스타일2', '스타일3', 'annotation']	30 [1, 6, 7, 3, 2, 1, 11, 98, 279, 20, 165, 209, 25, 21, 5, 113, 280, 281,	30 4304529.94
31 ['Normal', 'heading 1', 'heading 2', 'heading 3', 'heading 4', 'heading 5', 'heading 6', 'heading 7', 'heading 8', 'heading 9']	31 [1, 6, 7, 8, 12, 14, 15, 18, 19, 24, 29, 9, 16, 27, 49, 37, 28, 2, 3, 42	31 -33981.5786
32 ['Normal']	32 [1]	32 -1076421.981
33 ['Normal', 'heading 1', 'heading 2', 'heading 3', 'Balloon Text', '설명', '숫자', '열 머리글']	33 [1, 6, 7, 8, 5, 432, 433, 434]	33 -607094.0956

[그림 6] 단어 리스트 데이터 전처리

### 3.3.3 데이터 스케일링 : 표준화(STANDARDIZATION)

데이터 표준화는 전처리된 특징에 따라 데이터 단위가 다르므로, 단위를 일관되게 조정하여 유사도 계산 시 특성 간 차이를 고려할 수 있도록 하는 과정이다. 표준화를 통해 각 특징은 평균으로부터 어느 정도 오차 범위에 있는 값인지 나타낼 수 있다. 예를 들어, 여백의 단위와 고유 ID가 부여된 범주형 단어는 동일하게 1의 차이가 나더라도, 두 특징의 단위가 다르기 때문에 유사도 계산할 때 반영되어야 하는 차이가 다르다. 따라서 표준화를 적용하여 각 데이터의 특성이 달라도, 1의 차이가 동일한 효과를 반영하도록 해야 한다. 표준화가 적용된 데이터는 분산(평균에서 얼마나 떨어져 있는지)로 나타내기 때문에 모든 변수의 평균과 단위를 일관되게 반영하여 계산할 수 있다. 해당 과정은 sklearn 의 StandardScaler 를 통해 구현했다. 해당

함수를 적용하면 한 문서의 특정 특징  $f$ 에 대해서 모든 문서가 가진 특징  $f$ 의 평균과 얼마나 떨어져 있는지 확인할 수 있다. 표준화 식은 [수식 1]과 같다.

$$z_i = \frac{x_i - \bar{x}}{s}$$

### [수식 1] 표준화 식

#### 3.3.4 데이터 스케일링 : 정규화(NORMALIZATION)

데이터 정규화는 전처리된 특징의 데이터 범위가 다르므로, 각 특징이 정해진 값 사이에 오도록 조정하여 유사도 계산 결과를 해석하기 용이하게 하는 과정이다. 예를 들어, 문서의 크기와 고유 ID 가 부여된 글꼴은 데이터의 최소, 최대값이 다르기 때문에, 계산된 유사도 값이 정해진 범위의 값으로 나타내기 힘들다. 따라서, 계산된 유사도의 값이 0 과 100 사이의 값으로 나타나도록 하기 위해 정규화 과정이 필요하다. 정규화를 적용하면 전체 특징의 최대값, 최소값 범위 내에서 해당 값이 어느 정도의 위치에 속해 있는지 백분율로 표현할 수 있다. 또, 벡터의 크기를 일정하게 맞춤으로써, 벡터의 방향에 대한 정보만 고려할 수 있도록 해준다. 따라서, 정규화가 적용된 데이터는 코사인 유사도 알고리즘을 이용하여 계산한 결과 값이 -1 과 1 사이의 값으로 나타난다. 유사도 결과 값을 일정 범위 내의 값으로 나타내기 때문에 유사도 측정 결과를 해석하기 용이하다. 해당 과정은 sklearn 의 Normalizer 를 이용하여 구현했다. 해당 함수를 적용하면 한 문서의 특정 특징  $f$ 에 대해서  $f$ 특징의 최대값과 최소 값의 범위 내의 값으로 반환해 준다. 정규화 식은 [수식 2]와 같다

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

### [수식 2] 정규화

## 3.4 유사 문서 검색

유사 문서 검색 과정은 데이터 전처리 및 스케일링이 완료한 후, 특징 벡터로 나타낸 문서에 대해 문서 간 유사도를 계산하는 과정이다. 검색 대상 문서를 선정하고, 이 문서의 특징 벡터를 기준으로 다른 문서 파일의 특징 벡터들과 문서 유사도를 계산한다. 이를 통해 각 문서가 얼마나 유사한지 백분율로 제공할 수 있으며, 계산된 유사도를 기반으로 사용자에게 가장 유사한 문서를 추천한다.

유사 문서 검색 과정에서, 어떤 유사도 알고리즘을 이용하여 특징 벡터 간의 유사도를 계산할 것인지 선정해야 한다. 다양한 문서 유사도 측정 알고리즘 중에서 텍스트가 아닌 설정 서식의 수치로 유사도를 판단하는데, 어떤 알고리즘이 가장 적합한지 고려해야 한다.

문서 유사도를 계산하는데 사용되는 알고리즘으로는 자카드(Jaccard), 맨하튼(Manhattan), 유클리드(Euclidean), 코사인(Cosine), 피어슨 상관관계(Pearson Correlation Coefficient) 알고리즘 등이 있다.

자카드 유사도는 텍스트 기반으로 단어의 집합 관계로 유사도를 측정한다. 하지만 본 연구에서는 텍스트가 아닌 설정 서식의 값으로 유사도를 판단하기 때문에 자카드 유사도를 사용하기에는 적합하지 않다. 멘허튼과 유클리드 알고리즘에서는 두 문서 간의 차이(거리) 기반으로 유사도를 측정한다. 이 알고리즘들은 유사한 문서를 찾아 순위별로 추천해 주는 시스템에서는 이와 같은 계산 방식이 유용하다. 하지만 두 문서 사이의 유사도를 백분율로 표현하기는 어렵기 때문에 설정 서식의 유사성을 직관적으로 확인하기는 어렵다. 코사인과 피어슨 상관관계 알고리즘은 벡터 공간에서 문서 간 유사도를 측정하는데 사용된다. 이 알고리즘들은 벡터로 표현된 설정 서식의 값에서, 이 벡터 간의 각도를 이용하여 문서 간의 유사도를 계산할 수 있다. 또, 설정 서식의 유사성을 직관적으로 이해하기 백분율로 표현할 수 있어 사용자에게 직관적인 유사도를 제공해 줄 수 있다.

따라서 본 연구에서는 코사인과 피어슨 상관계수 알고리즘을 이용하여 문서 간 유사도를 측정한다. 이 알고리즘을 서식의 유사성을 백분율로 확인할 수 있으며 문서 간 상대적인 유사도를 직관적으로 이해할 수 있다.

### 1) 코사인 유사도

코사인 알고리즘은 두 벡터 간의 각도를 기반으로 유사도를 측정한다. 문서 정보 검색이나 클러스터링에서 사용되는 유사성 알고리즘 중 하나로, 1에 가까울수록 두 벡터는 유사하다고 판단된다. 문서  $U, V$  와 특징 벡터  $r$ , 특징 벡터를 구성하는 특징 원소  $I$ 에 대해 코사인 알고리즘은 [수식 3]과 같이 정의된다.

$$\text{Sim}(u, v)^{\cos} = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\| \cdot \|\vec{r}_v\|} = \frac{\sum_{i \in I_u \cap I_v} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u \cap I_v} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} r_{vi}^2}}$$

#### [수식 3] 코사인 유사도 수식

### 2) 피어슨 상관계수

피어슨 상관관계는 두 변수 간의 선형 상관관계를 측정하는 통계적 방법 중 하나다. 코사인 알고리즘과 유사한 식을 사용하지만, 각 특징에서 해당 객체가 가진 특징의 평균을 차감하여 객체의 평균에서 얼마나 차이가 나는지를 가지고 측정할 수 있다. 1에 가까울수록 두 벡터는 유사하다고 판단된다. 문서  $U, V$  와 특징 벡터  $r$ , 특징 벡터를 구성하는 특징 원소  $I$ 에 대해 피어슨 상관계수 알고리즘은 [수식 4]와 같이 정의된다.

$$\text{Sim}(u, v)^{\text{PCC}} = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}$$

#### [수식 4] 피어슨 상관계수 수식

## 4 평가

### 4.1 평가 데이터 세트

시스템의 성능을 측정하기 위해 NapierOne<sup>2</sup>에 공개된 Microsoft Word(.docx) 문서 데이터 세트 “DOCX-NOMAGIC-total.zip” 활용했다. 조직 로고나, 문서의 외형적 구조를 시각적으로 확인하여 조직을 식별할 수 있는 1,450 개의 문서 데이터를 분류 및 라벨링하여 사용했다. [그림 7]과 같이 조직 로고를 이용하여 1 차적으로 문서를 분류하고 2 차적으로 사람이 직접 문서의 외형적 구조를 시각적으로 확인하여 분류했다. 상세 분류는 과정은 아래 설명과 같으며, 향후 분석은 1 차적으로 분류한 세부 조직 양식에 대해서 2 개 이상의 문서를 가진 조직을 대상으로 실험했다.

**1 차 분류 (기관 로고)** : 분류의 첫번째 단계에서는 기관 로고 유무를 기준으로 문서 조직을 식별하여 문서를 분류했다. 1 차적으로 조직의 로고를 통해 분류한 카테고리 개수는 122 개다.

**2 차 분류 (시각적 확인)** : 동일 조직 로고를 갖는 문서라도, 전달되는 내용 유형에 따라 외형적 구조가 다르기 때문에 2 차적 분류가 필요하다. 이 단계에서는 사람이 직접 문서의 외형적 구조를 시각적으로 확인하여 분류했다. 2 차적으로 조직의 세부 양식을 분류한 카테고리 개수는 308 개이다

Access Northern Ireland 1

Access Northern Ireland 2

Access Northern Ireland 1

Access Northern Ireland 2

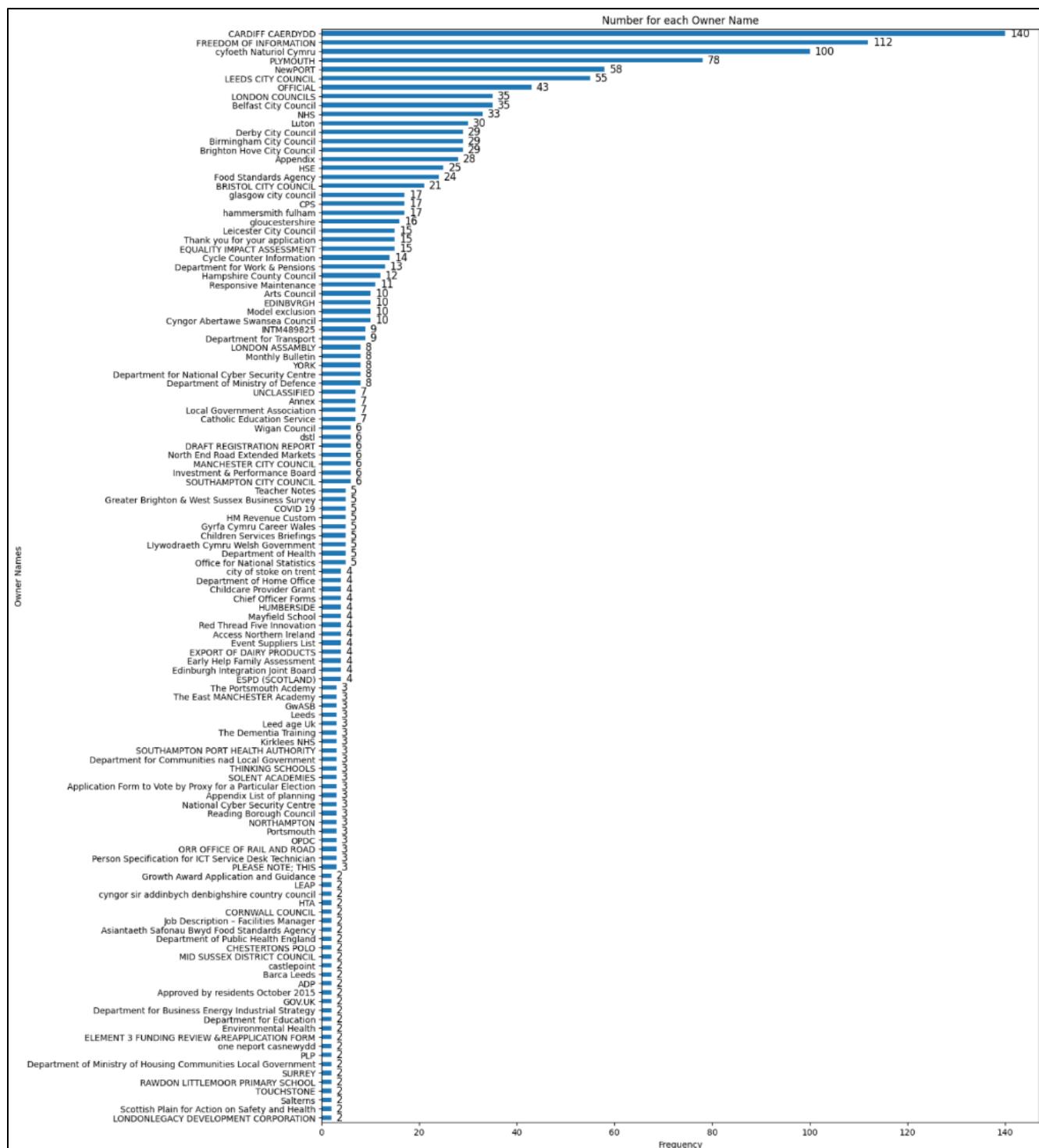
Access Northern Ireland 2

[그림 7] 외형적 구조가 유사한 데이터 라벨링

<sup>2</sup> <http://napierone.com.s3-eu-north-1.amazonaws.com/NapierOne/Data/DOCX/DOCX-NOMAGIC-total.zip>

NapierOne 은 랜섬웨어 탐지 및 포렌식 연구 발전을 목표로 공유된 사이버 보안 혼합 파일 데이터 세트이다. 데이터 세트에는 500,000 개가 넘는 개별 파일이 포함되어 있으며, docx 또한 500,000 개의 파일이 존재한다. 본 연구에서는 이 500,000 개 중 조직을 식별할 수 있는 1450 개의 데이터 셋을 선별하여 실험에 사용했다.

[그림 8]은 1 차적으로 분류한 조직별 데이터 세트의 분포를 나타내는 그라프다.



[그림 8] 1 차 분류된 조직별 문서 데잍 분포

## 4.2 평가 지표

본 시스템의 검색 정확도를 판단하기 위한 기준은 Precision(정밀도), Recall(재현율), F1-Score, Fall-out(오검출률)으로, 이는 검색 결과의 정확성과 완전성 그리고 오검출에 대한 신뢰도를 측정하는 지표다.

Precision 이란 검색 결과 중, 실제 조직 문서의 비율을 나타내며, Recall 은 시스템이 실제 조직의 문서인 것과 아닌 것을 얼마나 정확하게 검출했는지 비율을 의미한다. F1-Score 은 Recall 과 Precision 의 조화 평균을 말한다. Fall Out 은 시스템이 연관이 없는 문서를 얼마나 오탐했는지를 나타내며, 다른 정확도 측정치와 달리 낮을수록 시스템의 신뢰성이 높다는 것을 뜻한다. 이러한 측정치는 True Positives, False Positives, True Negatives, False Negative 로부터 도출된다. 각 설명은 아래와 같으며, [그림 9]는 검색 정확도 평가 기준 간의 관계를 시각적으로 나타낸다.

**True Positives:**  $|\text{selected\_docs} \cap \text{relevant\_docs}|$

시스템이 관련 문서로 판단한 것 중에서 실제 관련 문서의 개수

**False Positives:**  $|\text{selected\_docs} - \text{relevant\_docs}|$

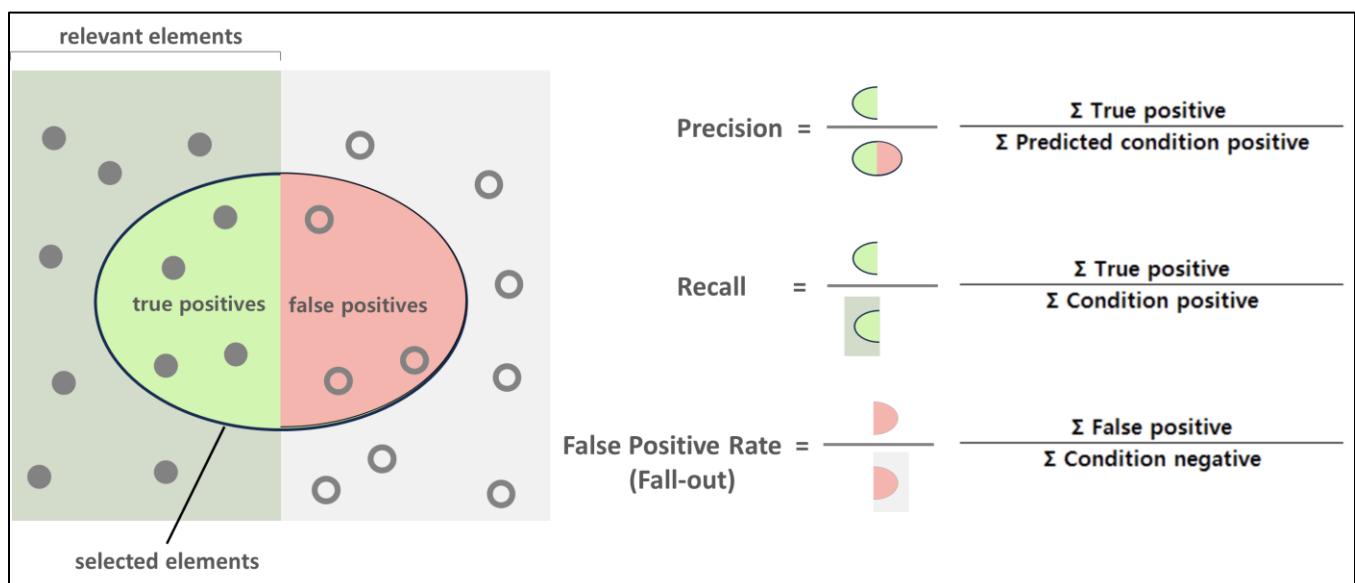
시스템이 관련 문서로 판단했지만, 실제 관련 문서가 아닌 문서의 개수

**True Negatives:**  $|\text{all\_docs} - \text{selected\_docs} - \text{relevant\_docs}|$

시스템이 관련 없는 문서로 판단한 것 중에서 실제 관련이 없는 문서의 개수

**False Negatives:**  $|\text{relevant\_docs} - \text{selected\_docs}|$

시스템이 관련 없는 문서로 판단한 것 중에서, 실제 관련 있는 문서의 개수



[그림 9] 검색 정확도 평가기준

### 4.3 평가 결과

본 검색 시스템의 검색 정확도는 코사인 유사도 및 피어슨 상관관계 알고리즘을 사용하여 1,450 번의 질의를 수행하고, 다양한 유사 문서 판단 기준(99%, 97%, 95%)에 대한 정확도를 측정했다. 2 차 분류된 조직 라벨을 정답 기준으로 사용했으며, 활용한 특징은 160 개로, 1,450 개의 문서 데이터 세트에서 발견된 특징을 모두 추출한 개수다.

코사인 알고리즘과 피어슨 상관관계 알고리즘의 검색 신뢰도를 아래 [표 1]과 [표 2]에 정리했다. 이 정확도는 1,450 개의 개별 문서들을 모두 질의했을 때, 1,450 번의 질의 결과 정확도를 평균 낸 결과다. 표의 유사 문서 판단 기준(%)은 사용자가 A 문서를 검색할 때, A-B 문서 간 유사도가 각각 99%/97%/95% 이상으로 측정된 문서들을 검색 결과로 판단하여 정확도를 측정했다는 뜻이다.

[표 1] Cosine 알고리즘 검색 신뢰도

선별된 특징(유사 문서 판단 기준 (%))	Precision(%)	Recall(%)	F1-Score(%)	Fall Out(%)
160 개(전체) 특징 사용 (99%)	76.7009	76.3263	67.6214	0.7862
160 개(전체) 특징 사용(97%)	71.1256	79.9269	66.4296	0.9994
160 개(전체) 특징 사용(95%)	68.0229	80.8274	64.6577	1.0782

[표 2] Pearson 알고리즘 검색 신뢰도

선별된 특징(유사 문서 판단 기준 (%))	Precision(%)	Recall(%)	F1-Score(%)	Fall Out(%)
160 개(전체) 특징 사용 (99%)	77.0248	76.2654	67.7814	0.7819
160 개(전체) 특징 사용(97%)	71.1431	79.9269	66.4850	0.9981
160 개(전체) 특징 사용(95%)	68.2053	80.8274	64.7793	1.0750

검색 시스템 정확도는 Recall, Precision, F1-Score 지표에서 약 70% 정확도로 나타난다. 코사인과 피어슨 상관관계 알고리즘에서 측정한 검색 결과는 매우 유사한 정확도를 보인다. 두 알고리즘 모두 동일한 유사 문서 판단 기준에서 Precision 과 Recall 결과가 유사하다.

본 검색 시스템은 특정 저장장치 내 존재하는 제한된 수의 데이터 세트에서 연관된 데이터를 추출하는 시스템이다. 질의에 따른 유사 문서 결과가 충분히 많은 인터넷에서의 검색과 달리 저장장치 내 검색은 데이터 세트에 따라 조직 별로 존재하는 문서가 유동적으로 변하고 소수로 존재한다. 정확도를 측정할 때 정답이 되는 조직별 문서 개수가 유동적으로 변하며, 정답의 개수가 소수인 환경에서는 하나의 오탐도 정확도 측정에 큰 영향을 미친다. 예를 들어, 검색한 문서 A 의 결과의 정답이 B,C,D 3 개의 문서인 경우, 검색 시스템이 B,D,E 를 추천해 주었다면 Precision, Recall 은 70(%)이 된다.

유사 판단 기준이 낮아질수록 Precision 은 낮아지는데, 이는 유사 판단 기준을 낮추면 검색 결과로 나오는 문서 개수가 늘어나면서 검색 결과에서 나타나는 검색 문서의 조직 문서가 아닌 다른 조직의 비율도 섞여 나오기 때문이다. 반면 유사 측정 기준이 낮아질수록 검색 결과에 나오지 않았었던 낮은 유사도를 갖는 조직 관련 문서를 결과로 나타내 주기 때문에 Recall 은 높아진다.

오탐 비율을 나타내는 fall-out 은 1.0782(%) 정도로 낮게 나타났는데, 이는 시스템의 검색 결과 중 연관된 조직이 아닌 다른 조직의 문서가 섞인 비율이 낮다는 것을 시사한다. 디지털 포렌식 수사 시, 특정 조직의 문서를 검색할 때, 다른 조직의 문서가 섞이는 비율이 1.0782% 정도로 낮게 나온 것은 선별 압수에 있어 긍정적인 효과를 기대할 수 있다.

본 검색 시스템은 코사인 유사도와 피어슨 상관관계 알고리즘에서 높은 정확도를 보였다. 유동적인 정답 문서 수에도 불구하고, 검색 결과의 정확도와 신뢰도를 유지하고 있으며, fall-out 이 낮아 조직 관련 문서의 신속하고 효과적인 선별을 기대할 수 있다.

## 5 고찰

### 5.1 연구 결과에 대한 고찰

본 절에서는 연구 결과에 대한 고찰로, 시스템에서 개별 문서를 질의한 결과의 상위 10 개 검색 결과를 평가한다. 개별 검색 결과의 정확도가 좋은 예시와, 나쁜 예시를 하나씩 들어 설명한다

우선, 검색 결과의 정확도가 높은 문서의 경우를 살펴본다. 이러한 문서는 조직의 서식이 명확하게 지정된 경우로 조직의 로고뿐만 아니라 머리말, 꼬리말, 표지, 스타일 등이 명확하게 지정되었다. [표 3]은 “2426-docx-nomagic.docx” 파일을 검색했을 때의 결과를 나타낸다. 질의 파일의 식별 카테고리는 “CARDIFF CAERDYDD”로, 검색 결과 파일 역시 전부 “CARDIFF CAERDYDD” 카테고리의 파일로 확인할 수 있다. 따라서 [표 3]의 정답 여부에 TRUE 값으로 표시되며, 질의 문서와 각 검색 결과 파일 간의 유사도는 1 이다. [그림 10]은 검색 결과의 정확도가 높은 질의-검색 결과 파일의 예시를 시각적으로 보인다. 이 예시에서는 조직의 로고와 함께 꼬리말, 상하좌우의 여백과 같이 서식이 명확하게 지정되었다는 것을 확인할 수 있다.

[표 3 ] 검색 정확도가 높은 문서 검색 결과

Query: 2426-docx-nomagic.docx (Category: CARDIFF CAERDYDD)				
검색 결과 파일명	1 차 카테고리	2 차 카테고리	유사도	정답 여부
2543-docx-nomagic.docx	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
2445-docx-nomagic.docx	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
2433-docx-nomagic.docx	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
2434-docx-nomagic.docx	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE

<b>2435-docx-nomagic.docx</b>	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
<b>2436-docx-nomagic.docx</b>	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
<b>2437-docx-nomagic.docx</b>	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
<b>2438-docx-nomagic.docx</b>	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
<b>2439-docx-nomagic.docx</b>	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE
<b>2440-docx-nomagic.docx</b>	CARDIFF CAERDYDD	CARDIFF CAERDYDD	1.0	TRUE

**Query: 2426-docx-nomagic.docx**



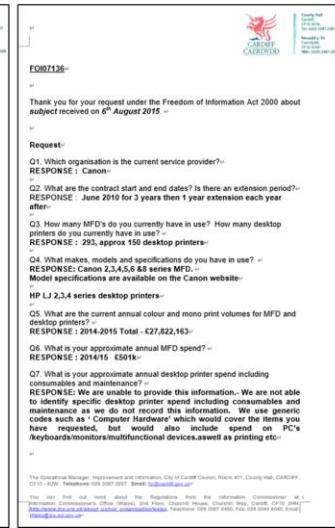
**2445-docx-nomagic.docx**



**2433-docx-nomagic.docx**



**2434-docx-nomagic.docx**



[그림 10] 검색 결과의 정확도가 높은 질의-결과 검색 파일 예시

다음으로, 검색 결과의 정확도가 낮은 문서의 경우를 살펴본다. 이러한 문서는 문서 페이지나 섹션, 스타일 지정 등의 문서 전체에 적용되는 별도의 사용자 서식 없이 기본 서식만을 사용한 경우가 대부분이다. [표 4]는 “4237-docx-nomagic.docx” 파일을 검색했을 때의 결과를 나타낸다. 질의 파일의 식별 카테고리는 “Thank you for your application”로, 검색 결과 문서의 카테고리는 이와 다르다는 것을 확인할 수 있다. 따라서 [표 4]의 정답 여부에 FALSE 값으로 표시했다. 질의 문서와 각 검색 결과 파일 간의 유사도는 모두 1로 나타났다. [그림 11]은 검색 결과의 정확도가 낮은 질의-검색 결과 파일의 예시를 시각적으로 보인다. 이러한 문서들은 주로, 본 연구의 범위에서 다루지 않는 개별 텍스트에 적용되는 서식 값, 예를 들어 글꼴의 크기, 굵기, 조직의 로고, 표 등의 서식 값을 적용했거나 기본 서식만을 사용했다는 것을 확인할 수 있다.

[표 4] 검색 정확도가 낮은 문서 검색 결과

**Query: 4237-docx-nomagic.docx (Category: Thank you for your application)**

검색 결과 파일명	1 차 카테고리	2 차 카테고리	유사도	정답 여부
<b>4237-docx-nomagic.docx</b>	Thank you for your application	Thank you for your application	1.0	TRUE

<b>0485-docx-nomagic.docx</b>	Department for Work Pensions	Department for Work Pensions 3	1.0	FALSE
<b>0321-docx-nomagic.docx</b>	Derby City Council	Derby City Council 10	1.0	FALSE
<b>4843-docx-nomagic.docx</b>	LEEDS CITY COUNCIL	LEEDS CITY COUNCIL - JOB DESCRIPTION	1.0	FALSE
<b>4162-docx-nomagic.docx</b>	LEEDS CITY COUNCIL	LEEDS CITY COUNCIL - JOB DESCRIPTION	1.0	FALSE
<b>0045-docx-nomagic.docx</b>	Luton	Luton 12	1.0	FALSE
<b>3944-docx-nomagic.docx</b>	OFFICIAL	OFFICIAL – SENSITIVE 3	1.0	FALSE
<b>3943-docx-nomagic.docx</b>	OFFICIAL	OFFICIAL – SENSITIVE 3	1.0	FALSE
<b>0320-docx-nomagic.docx</b>	OFFICIAL	OFFICIAL – SENSITIVE 2	1.0	FALSE
<b>0298-docx-nomagic.docx</b>	OFFICIAL	OFFICIAL – SENSITIVE 2	1.0	FALSE

Query: 4237-docx-nomagic.docx

The screenshot shows four document snippets side-by-side:

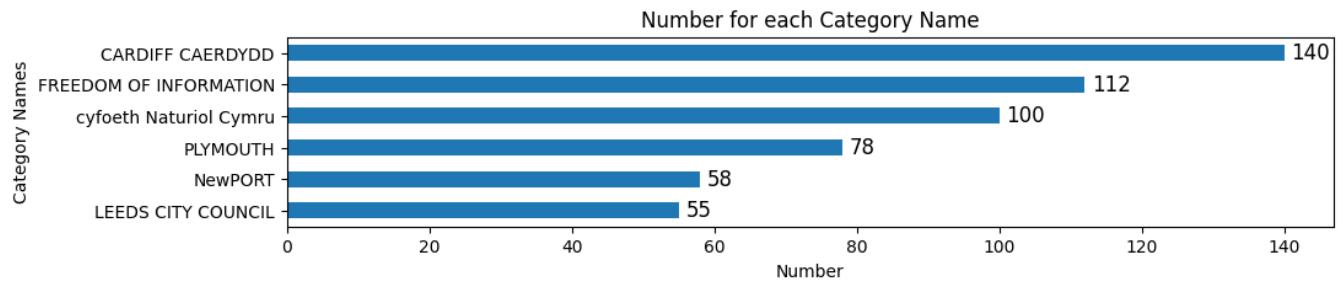
- 0485-docx-nomagic.docx**: A snippet from the Department for Work Pensions regarding high-rise fire attendance. It asks for data from 2010 to 2019, mentioning pump requirements and building regulations.
- 0321-docx-nomagic.docx**: A snippet from Derby City Council about statutory assessment training for teachers, mentioning K11 and K12 levels.
- 4843-docx-nomagic.docx**: A snippet from Leeds City Council's job description for Customer Service Team Leader, detailing responsibilities like providing local front of house management and ensuring integrated customer service provision.
- 4162-docx-nomagic.docx**: Another snippet from Leeds City Council's job description for Customer Service Team Leader, listing responsibilities such as managing staff and developing performance programmes.

[그림 11] 검색 결과의 정확도가 낮은 질의-결과 검색 파일 예시

## 5.2 특징 선정 기준

본 연구에서 선정한 사용자 설정 서식 특징들이 연구 결과에 미치는 성능을 평가한다. 이를 평가하기 위해 분류 알고리즘을 사용하여 분류 정확도를 측정한다. 또한, 분류 알고리즘으로 평가된 특징 중요도(Feature Importance)의 순위를 기반으로 선별한 특징만을 이용해 검색 정확도를 재평가한다.

분류 알고리즘 실험을 위해 1 차 분류 기준으로 50 개 이상의 문서를 가진 6 개의 조직으로 대상으로 데이터 세트를 구성했다. 이 데이터 세트는 2 차 분류 기준으로 조직 수는 42 개이며, 총 543 의 문서를 가진다. [그림 12]는 분류를 위해 추린 50 개 이상 데이터를 보유한 조직별 데이터 분포도를 시각화한 그래프다.



[그림 12] 1 차 분류 조직 중 50 개 이상 데이터를 가진 조직별 데이터 분포도

구성된 데이터 세트에 적용된 모든 특징을 추출하여 분류 알고리즘을 적용한 결과를 정리했다. [표 5]는 다양한 분류 알고리즘에서 1 차 분류된 조직을 정답으로 하여 분류의 정확도를 평가한 결과를 정리한 표다. 가장 높은 정확도를 보이는 알고리즘은 RandomForest 로, [표 6]은 RandomForest 를 이용하여 각 조직의 분류 정확도를 측정한 결과이다. Random Forest 가 98% 이상의 높은 성능으로 서식 데이터를 이용하여 조직을 분류하는 것을 통해, 서식 데이터의 조합이 조직을 식별함에 있어 유용하다는 것을 확인할 수 있다.

[표 5] 여러가지 분류 알고리즘을 적용한 서식 데이터 기반 분류 정확도

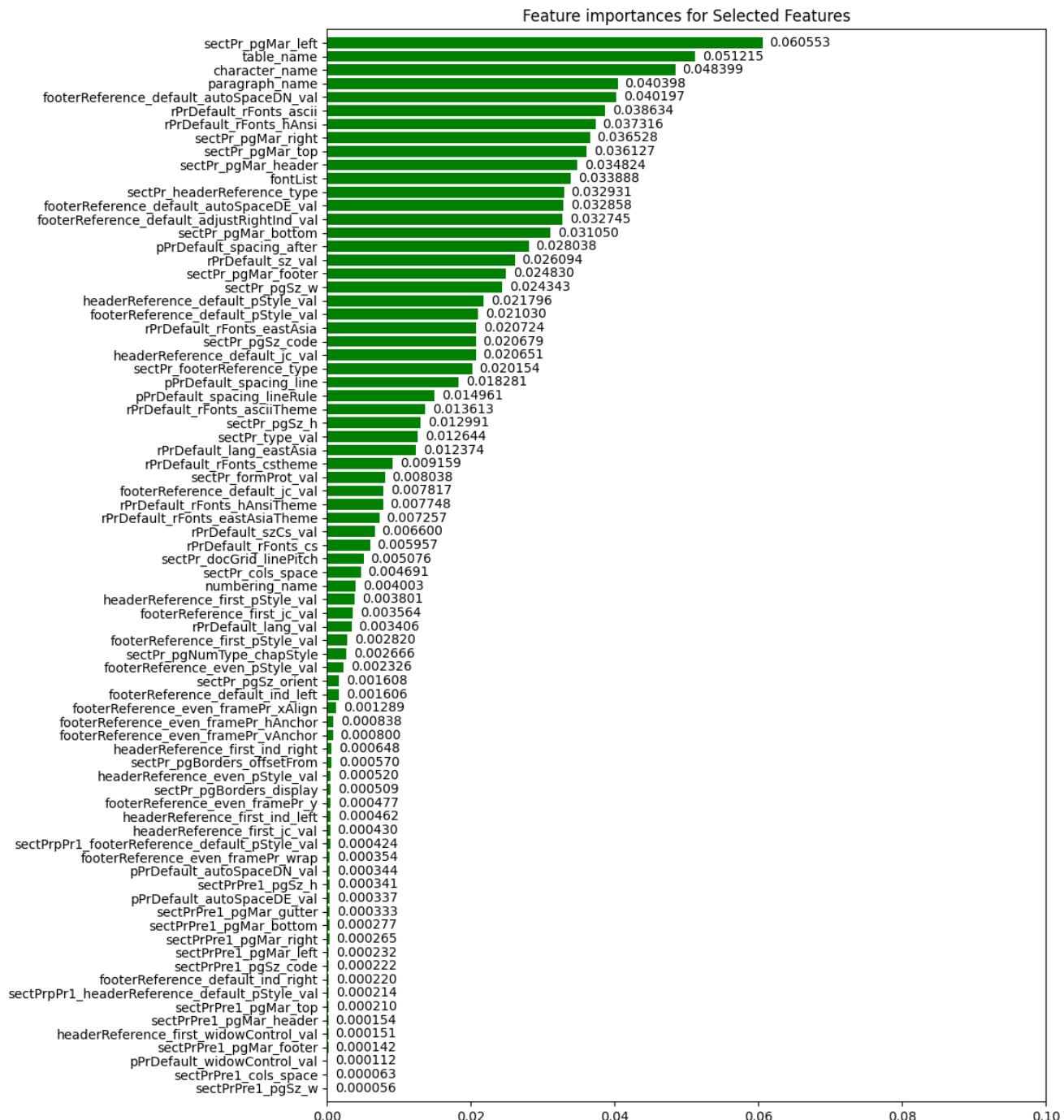
분류 알고리즘	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)
<b>Random Forest</b>	98.159509202453	97.989267676767	98.571428571428	98.222897553829
<b>ExtraTrees</b>	97.546012269938	96.296296296296	97.584058713090	96.677056068328
<b>DecisionTree</b>	95.705521472392	94.733796296296	95.705751834784	94.898842621625
<b>SVC</b>	75.460122699386	80.679603048024	72.230674912595	73.232142314729

[표 6] 서식 데이터를 활용한 Random Forest 분류 정확도

Class	Precision	Recall	F1-Score	Support
<b>CARDIFF CAERDYDD</b>	0.98	1.00	0.99	43
<b>FREEDOM OF INFORMATION</b>	0.97	1.00	0.98	31
<b>PLYMOUTH</b>	0.93	1.00	0.97	14
<b>cyfoeth Naturiol Cymru</b>	1.00	1.00	1.00	18
<b>NewPORT</b>	1.00	1.00	1.00	22
<b>LEEDS CITY COUNCIL</b>	1.00	0.91	0.96	35
<b>Accuracy</b>			0.98	163
<b>Macro Avg</b>	0.98	0.99	0.98	163
<b>Weighted Avg</b>	0.98	0.98	0.98	163

또, 중요한 특징만을 선별하여 사용한 검색 시스템을 평가하기 위해 RandomForest 를 이용하여 특징 중요도를 평가했다. [그림 13]은 RandomForest 를 이용해서 각 특징의 중요도를 측정한 결과를 나타낸다.

이 과정에서 중요도가 0으로 측정된 Feature 들은 제외하여 검색 시스템을 다시 구축했다. 특정 중요도 순위를 보면, 전체 문서에 적용한 서식들 중 문서의 왼쪽, 오른쪽, 아래, 머리말, 꼬리말과 같은 여백과 테이블 명, 글꼴 명, 문단 스타일 명 등의 서식이 상위권을 차지한다는 사실을 확인할 수 있다.



[그림 13] Random Forest로 평가된 서식 설정 중요도

Random Forest로 평가된 특징 중요도를 고려하여 중요도가 0.0000000001 이상인 특징 78 개, 0.1 이상인 특징 31 개만을 이용해서 검색의 정확도를 판별한 결과를 정리했다. [표 7]은 Cosine 알고리즘을 이용하여 선별된 특징을 적용한 검색 정확도를 보이고, [표 8]은 Pearson 알고리즘을 이용하여 선별된 특징을 적용한 검색 정확도를 보인다.

[표 7] Cosine 알고리즘 검색 신뢰도 (선별 특징 적용)

선별된 특징(유사 문서 판단 기준 (%))	Precision(%)	Recall(%)	F1-Score(%)	Fall Out(%)
160 개(전체) 특징 사용 (99%)	76.7009	76.3263	67.6214	0.7862
160 개(전체) 특징 사용(97%)	71.1256	79.9269	66.4296	0.9994
160 개(전체) 특징 사용(95%)	68.0229	80.8274	64.6577	1.0782
78 개 특징 선정 (99%)	75.5693	76.6605	67.2574	0.8156
78 개 특징 선정 (97%)	70.2941	79.7570	65.9515	1.0223
78 개 특징 선정 (95%)	67.2922	80.9562	64.3937	1.1070
31 개 특징 선정 (99%)	72.3844	77.6734	65.9586	1.0003
31 개 특징 선정 (97%)	66.4360	80.3770	63.5373	1.4247
31 개 특징 선정 (95%)	63.9289	82.0916	62.8214	1.6069

[표 8] Pearson 알고리즘 검색 신뢰도(선별 특징 적용)

선별된 특징(유사 문서 판단 기준 (%))	Precision(%)	Recall(%)	F1-Score(%)	Fall Out(%)
160 개(전체) 특징 사용 (99%)	77.0248	76.2654	67.7814	0.7819
160 개(전체) 특징 사용(97%)	71.1431	79.9269	66.4850	0.9981
160 개(전체) 특징 사용(95%)	68.2053	80.8274	64.7793	1.0750
78 개 특징 선정 (99%)	75.5884	76.5860	65.9553	1.0033
78 개 특징 선정 (97%)	70.3487	79.5535	65.8100	1.0220
78 개 특징 선정 (95%)	67.2972	80.9562	64.4014	1.1061
31 개 특징 선정 (99%)	72.3802	77.6792	65.9553	1.0033
31 개 특징 선정 (97%)	66.3279	80.3862	63.4497	1.4323
31 개 특징 선정 (95%)	64.0704	82.2457	63.0525	1.6148

분류 알고리즘에서 주요하게 평가된 특징들을 선별하여 구축한 검색 시스템의 정확도가 전체 특징을 이용한 것보다 전반적으로 수치가 낮다는 것을 확인할 수 있다.

분류 알고리즘은 특정 경계 안에 속한 특징 값을 특정 조직이라고 인식하는 것과 달리, 유사도 알고리즘은 특징 값의 차이를 계산하여 유연하게 문서 간 유사 관계를 인식한다. 또한, 분류 시스템에서 평가한 특징들은 조직의 문서 특성과 다른 조직과의 차별성을 얼마나 잘 반영하는 특징인지에 중점을 둔다. 반면, 검색 시스템은 문서에 동일하게 설정된 서식 값이 있으면 다른 조직의 문서라고 하더라도 유사도가 높아지는 것이 반영되어야 한다.

결과적으로 분류 알고리즘에서 중요도가 낮은 특징을 생략하면 분류 정확도는 향상될 수 있지만, 문서 유사 관계를 측정하는 데는 도움이 되지 않을 수 있다. 예를 들어, 모든 조직에서 같은 값으로 설정되거나 다른 조직과 명확하게 구별할 수 없을 정도로 차이가 미미한 특징은 분류 알고리즘에서 특징 중요도가 낮게 평가된다. 이런 특징들을 제외한 경우, 분류 정확도의 수치가 높아질 수 있겠지만, 개별 문서 간 특징들의 유사도를 계산하는 데는 도움이 되지 않는다.

이러한 이유로 특징을 선정하여 유사도 알고리즘을 구성할 때, 전반적으로 정확도 수치가 낮아진 것으로 보인다.

## 6 결론

본 연구에서는 기존 유사 문서 검색 시 활용하던 본문 기반의 검색 방법이 아닌, 외형적 특징을 활용한 유사 문서 검색 방법을 제안했다. 특히, 점유율이 높은 Microsoft Word 2007+(.docx) 문서 포맷을 실시 예로 활용하여 본 연구에서 제안한 방법론을 검증하였다. 그리고 공개된 데이터 세트를 활용한 유사 문서 검색 실험에서 약 77%의 정확도를 보였다.

디지털 포렌식 수사 시 수많은 문서 중 유사한 문서를 군집화하는 것은 인적 비용 및 시간적 비용을 줄일 수 있다. 기존 본문 기반 유사 문서 검색 방법은 본문의 내용을 통해 비슷한 주제를 지닌 문서를 선별하는 것이 주목적이다. 이와 함께 본 연구에서 제안한 외형적 특징을 활용한 유사 문서 검색 방법을 수사 시 활용한다면, 특정 기관이나 집단에서 작성한 문서를 추가적으로 군집화하여 효율적인 수사를 수행할 수 있을 것으로 기대한다.

## 7 부록 1: 도구 사용 방법

### 1. 외형적 유사도를 측정하고자 하는 Microsoft Word 2007+ (.docx) 문서를 수집한다.

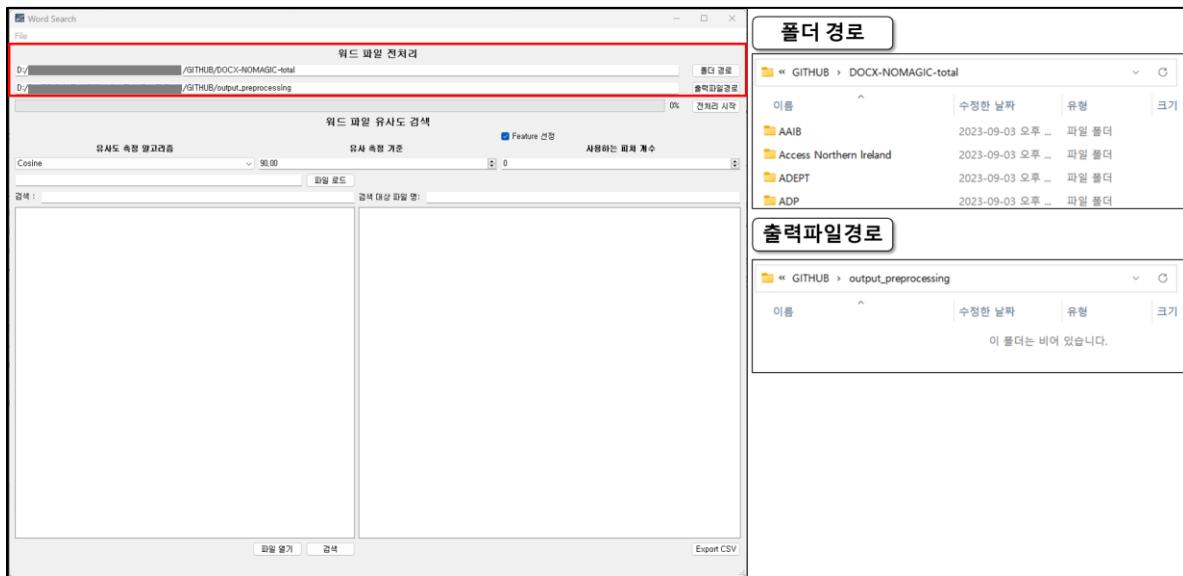
- A. 본 예시에서는 NapierOne에서 제공하는 데이터 세트를 대상으로, 조직 로고와 외형적으로 유사하다 판단되는 문서를 직접 폴더별로 분류해서 사용한다.

📁 AAIB	2023-07-19 오후 10:26	
📁 Access Northern Ireland	2023-07-05 오전 10:43	
📁 ADEPT	2023-07-19 오후 10:26	
📁 ADP	2023-07-19 오후 10:26	
📁 Annex	2023-07-19 오후 10:26	
📁 Appendix	2023-07-05 오전 10:43	
📁 Appendix List of planning	2023-07-19 오후 10:26	
📁 Application Form to Vote by Proxy for a Particular Ele...	2023-07-19 오후 10:26	
📁 Apprenticeships	2023-07-19 오후 10:26	
📁 Approved by residents October 2015	2023-07-19 오후 10:26	
📁 ARMED FORCES COVENANT	2023-07-19 오후 10:26	
📁 Arthurerry Learning Partnership	2023-07-19 오후 10:26	
📁 Arts Council	2023-07-05 오전 10:43	
📁 Asiantaeth Safonau Bwyd Food Standards Agency	2023-07-19 오후 10:26	
📁 Attain	2023-07-19 오후 10:26	
📁 Autism East Midlands	2023-07-19 오후 10:26	
📁 Barca Leeds	2023-07-19 오후 10:26	
📁 BARNSLEY	2023-07-19 오후 10:26	
📁 Access Northern Ireland 1	2023-07-19 오후 10:26	
📁 Access Northern Ireland 2	2023-07-19 오후 10:26	
DOCX-DataPreprocessing.py	2023-09-24 오후 ...	파일
DOCX_feature_extraction.py	2023-09-04 오전 ...	파일
frozen_application_license.txt	2023-09-24 오후 ...	텍스트 문서
msvcp140.dll	2023-04-05 오후 ...	응용 프로그램
python3.dll	2022-12-06 오후 ...	응용 프로그램
python310.dll	2022-12-06 오후 ...	응용 프로그램
vcomp140.dll	2023-04-05 오후 ...	응용 프로그램
WORD_LOGO.ico	2023-09-15 오후 ...	ICO 파일
WORD_main.py	2023-09-22 오후 ...	파일
WordSearch.exe	2023-09-24 오후 ...	응용 프로그램

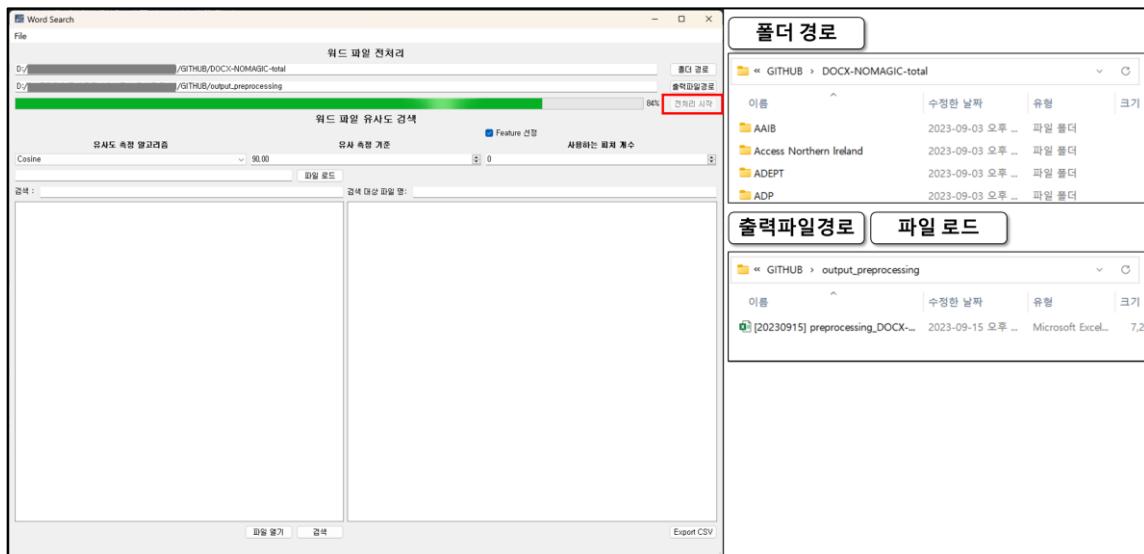
### 2. 'WordSearch.exe'를 실행한다.

📁 lib	2023-09-24 오후 ...	파일 폴더
DOCX_DataPreprocessing.py	2023-09-24 오후 ...	파일
DOCX_feature_extraction.py	2023-09-04 오전 ...	파일
frozen_application_license.txt	2023-09-24 오후 ...	텍스트 문서
msvcp140.dll	2023-04-05 오후 ...	응용 프로그램
python3.dll	2022-12-06 오후 ...	응용 프로그램
python310.dll	2022-12-06 오후 ...	응용 프로그램
vcomp140.dll	2023-04-05 오후 ...	응용 프로그램
WORD_LOGO.ico	2023-09-15 오후 ...	ICO 파일
WORD_main.py	2023-09-22 오후 ...	파일
WordSearch.exe	2023-09-24 오후 ...	응용 프로그램

**3. Word Search 프로그램을 실행하여 수집한 문서 데이터 세트의 경로를 입력하고, 문서의 외형적 특징 정보를 전처리한 결과물 CSV 파일을 저장할 경로를 입력한다.**



**4. 전처리 시작을 누르면 데이터 세트 Word 파일에서 외형적 유사도를 비교할 특징 정보를 추출하여, 출력 파일 경로 내 CSV 파일로 저장한다.**

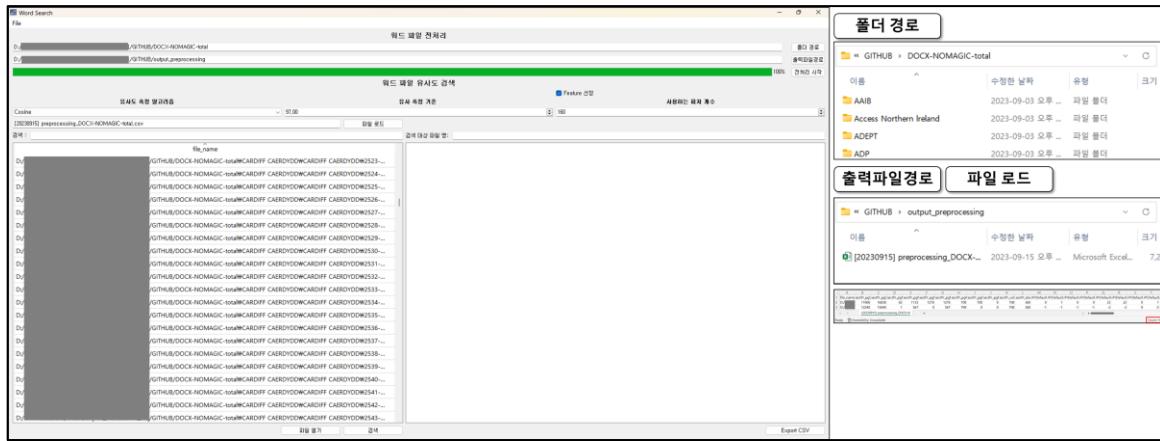


**A. 외형적 유사 문서 검색에 사용되는 특징들을 추출한 CSV 예시**

- i. ([20230915] preprocessing\_DOCX-NOMAGIC-total.csv)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	file_name	sectPr_pg1	sectPr_pg2	sectPr_pg3	sectPr_pg4	sectPr_pg5	sectPr_pg6	sectPr_pg7	sectPr_pg8	sectPr_pg9	sectPr_pg10	sectPr_pg11	sectPr_pg12	sectPr_pg13	sectPr_pg14	sectPr_pg15	sectPr_pg16	sectPr_pg17	sectPr_pg18	
2	D:\	11906	16838	62	1133	1276	1276	708	708	0	708	360	0	1	0	0	22	22	0	1
3	D:\	12240	15840	1	567	0	567	709	0	0	708	360	-1	-1	-1	-1	-2	-2	0	0

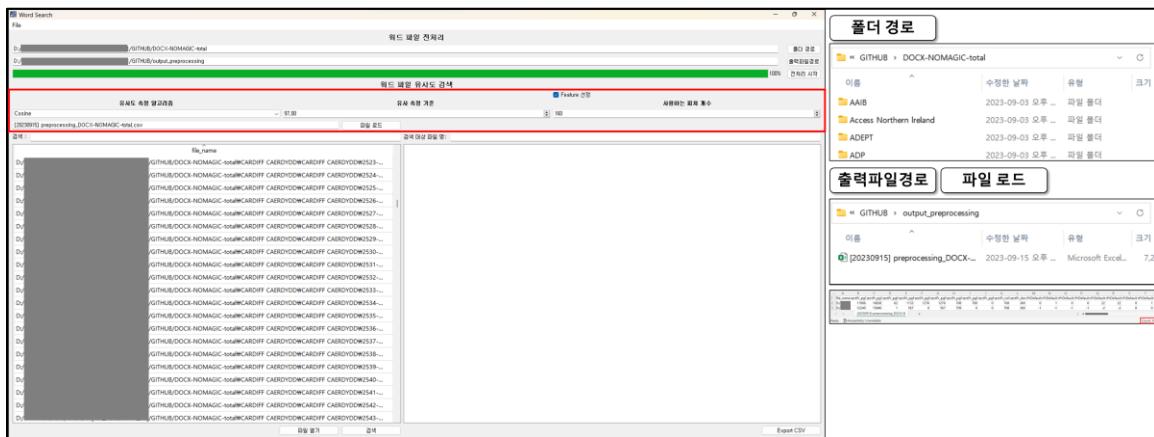
## 5. 파일 로드 버튼을 클릭한 뒤, 데이터 세트에서 외형적 특징을 추출한 CSV 파일을 선택한다.



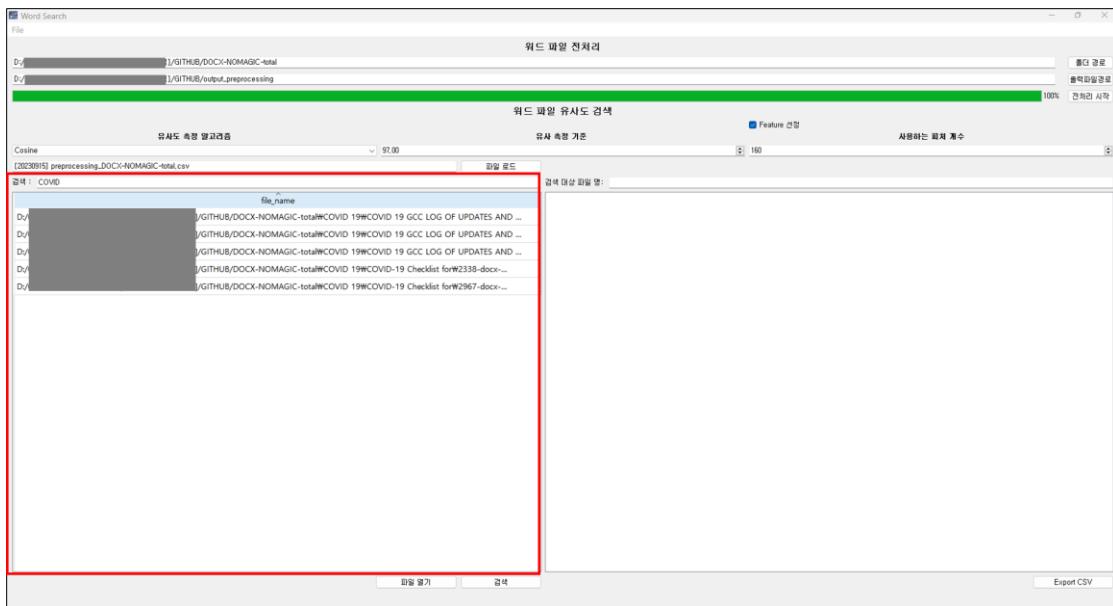
## 6. 외형적으로 유사한 문서를 검색하기 전, 검색 옵션을 설정해야 한다. 실험을 통해 본 프로그램을 가장 효과적으로 활용할 수 있는 검색 옵션을 기본값으로 설정해 놓았다. 하지만 해당 옵션은 수정해서 사용할 수 있다.

- A. [유사도 측정 알고리즘], [유사 측정 기준], [사용하는 피처 개수] 옵션을 조정 가능
  - ① 유사도 측정 알고리즘 : 문서간 유사도를 계산하는 알고리즘 (cosine/pearson)
  - ② 유사 측정 기준 : 문서 간 유사도가 유사 측정 기준 이상인 파일만 결과로 출력 (0~100)
  - ③ 사용하는 피처 개수 : 사용된 전체 피쳐들 개수는 기본값으로 설정되며, 특징 선정 체크박스를 선택하고 사용하는 피처 개수를 조정하면 해당 피처만을 이용하여 검색 가능

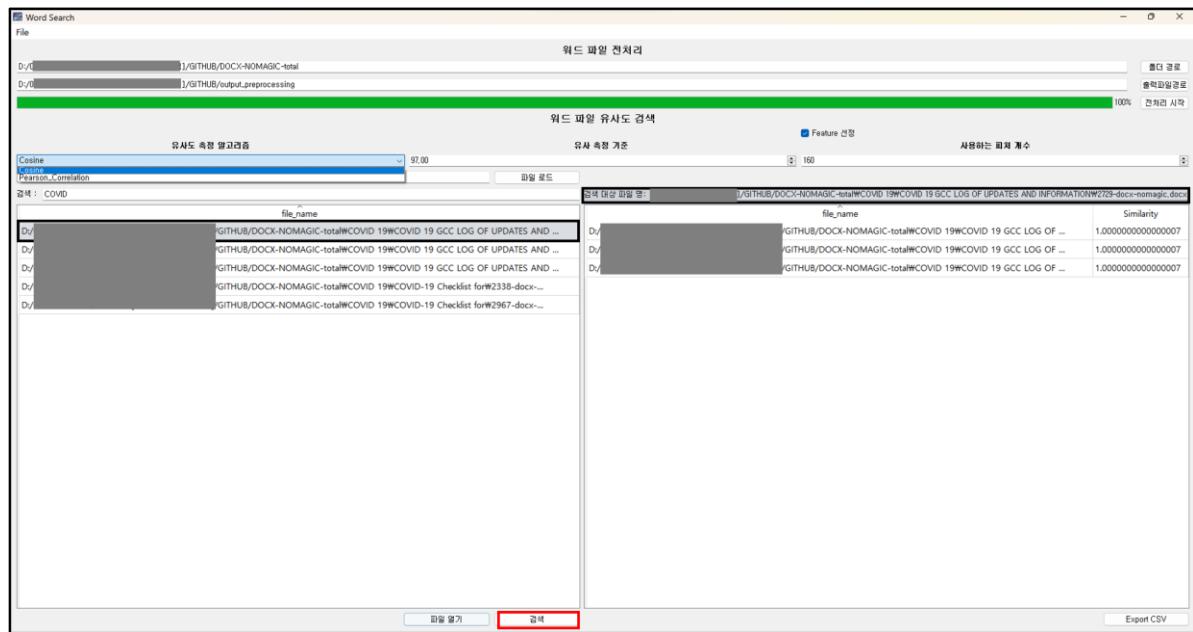
※ 이때 csv에 기재된 특징 컬럼 순서대로 검색에 사용되는 특징이 선택된다.



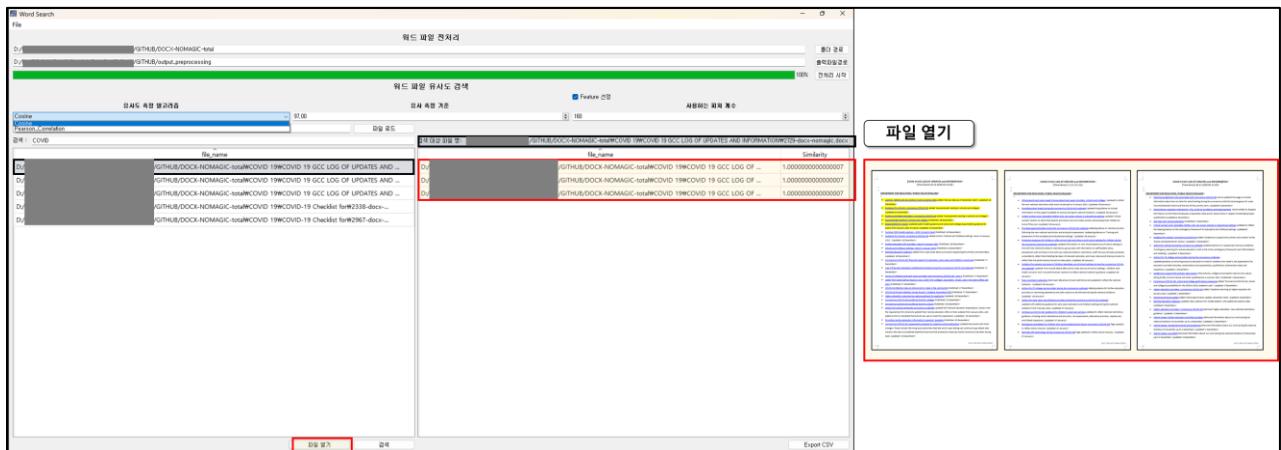
## 7. 왼쪽 검색 텍스트 박스에 검색하고 싶은 이름을 검색하면, 왼쪽 리스트 박스에 필터링된다.



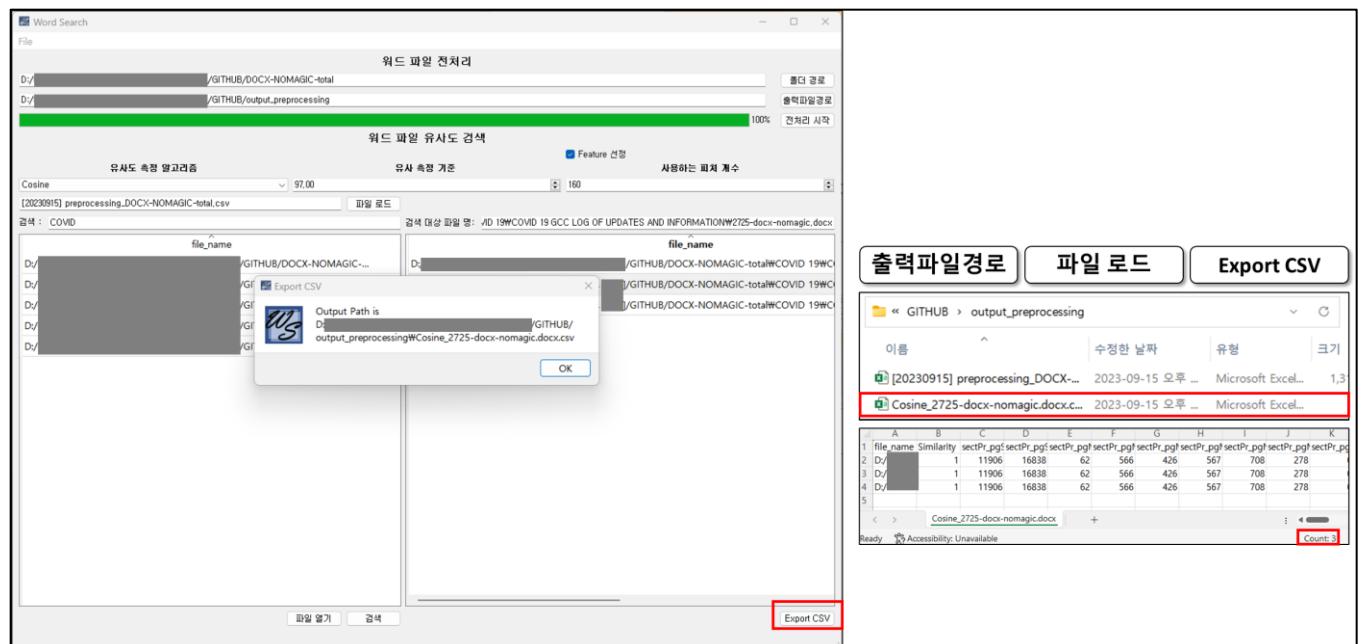
8. 왼쪽 리스트 박스에서 유사한 문서를 검색하고자 하는 파일을 클릭하면, 오른쪽 검색 대상 파일 명 텍스트 박스에 해당 파일의 이름이 입력된다. 그리고 검색 버튼을 클릭하면 검색 대상 파일과 유사한 외형적 구조를 가진 문서 목록을 보여준다.



9. 리스트 박스 내 DOCX 문서 이름을 하나 클릭한 뒤 “파일 열기” 버튼을 선택하면 해당 파일을 WORD 프로그램으로 열람할 수 있다.



10. Export CSV 버튼을 클릭하면 검색 결과 목록과 함께 유사도 계산에 사용한 특징 정보를 CSV 파일로 저장한다. 출력 파일 경로로 설정한 경로에 해당 특징 정보 파일이 저장된다.



## 8 부록 2: MS-WORD 외형적 서식 목록

본 연구에서 분석한 Word(.docx) 파일에 포함된 사용자 서식 설정값은 document.xml, style.xml, fontTable.xml, header#N.xml, footer#N.xml에 포함되며, 자세한 서식 설정값 목록을 표로 정리했다. 정리한 표가 갖는 컬럼은 파일내 속성 경로, 속성명, 속성 내용, 필수 여부로, 각 컬럼에 대한 설명은 다음과 같다.

\* **파일 내 속성 경로** : xml 파일에서 해당 서식 설정을 찾기 위해 거쳐야 할 하위 자식 요소(element) 경로를 뜻한다.

\* **속성 명** : 해당 서식 설정값이 저장되어 있는 속성(attribute)에 대한 명칭이다.

\* **속성 내용** : 해당 서식 설정값에 대한 설명이다.

\* **필수 여부** : 사용자가 특별히 설정한 값이 없어도 사전 정의된 (default) 값으로 저장되어, 사용자의 설정 여부와 관계없이 필수적으로 존재하는 속성(attribute)에 대해서는 “필수”, 사용자가 설정할 때만 존재하는 속성(attribute)에 대해서는 “선택”으로 표기했다.

### 8.1 DOCUMENT.XML 특징

/word/document.xml : Microsoft Word 문서의 실제 내용을 포함하는 XML 파일이다. 사용자가 입력한 텍스트 데이터와 그에 따른 문단의 서식, 글꼴, 크기, 스타일과 같은 서식 설정 값, 그림, 표, 등 문서를 구성하는 요소들이 이 파일에 포함된다.

[표 9] document.xml 내 추출 속성

파일 내 속성 경로	속성 명	속성 내용	필수 여부
[sectPr] - [pgSz] (페이지 크기)	w	용지 너비	필수
	h	용지 높이	필수
	orient	페이지 방향 (portrait / landscape)	선택
	code	크기가 프린터에서 지원하는 여러 용지 유형의 크기와 일치하는 경우 적절한 유형이 선택되도록 내부 용지 코드를 가져오거나 설정.	선택
[sectPr] - [pgMar] (페이지 여백)	top	페이지 위쪽 여백	필수
	right	페이지 오른쪽 여백	필수
	bottom	페이지 아래쪽 여백	필수
	left	페이지 왼쪽 여백	필수
	header	페이지 머리말 여백	필수
	footer	페이지 꼬리말 여백	필수

	gutter	페이지 제본 여백	필수
[sectPr]-[pgBorders] (페이지 테두리)	offsetFrom	페이지 여백을 기준으로 페이지 테두리의 위치 지정 (page / text)	선택
	z-order	교차하는 텍스트 및 개체를 기준으로 페이지 테두리가 배치되는 위치 지정(front / back)	선택
	display	페이지 테두리가 인쇄되는 페이지 지정(all-pages / first-page / not-first-page)	선택
[sectPr]-[pgBorders]-[Top] [sectPr]-[pgBorders]-[Left] [sectPr]-[pgBorders]-[Bottom] [sectPr]-[pgBorders]-[Right]	val	선종류(single)	선택
	sz	선 굵기(4)	선택
	space	여백(24)	선택
	color	선 색 (auto)	선택
	shadow	선 그림자 (on/off)	선택
	frame	테두리를 반전시켜 프레임 효과를 만들 것인지 여부 설정(on/off)	선택
	wx:bdrwidth	외부 네임 스페이스	선택
[sectPr]-[cols]	w	열의 너비를 가져오거나 설정합니다	선택
	space	이 열과 다음 열 사이의 공백 설정, 마지막 열에는 필요하지 않음.	선택
[sectPr]-[docGrid]	type	그리드 유형 설정	선택
	linePitch	줄 피치와 줄 간격 설정 페이지당 줄 수는 줄 사이의 간격에 맞게 자동으로 조정.	선택
	char-space	문서의 줄당 문자 수 설정	선택

## 8.2 STYLE.XML 특징

/word/style.xml : Microsoft Word 문서에서 사용되는 스타일을 정의하는 XML 파일이다. 문서의 서식, 글꼴, 크기, 간격 등을 지정하는 스타일 요소들이 포함된다. 문서 전체에 적용되는 기본 스타일부터 문서 내 정의된 폰트 스타일까지 다양한 스타일이 포함된다.

서식 추출은 styles.xml 파일 내 자식 요소 중 "rPrDefault", "pPrDefault", "style" 태그를 가진 노드의 자식 요소를 순환하며 해당 요소의 속성의 키와 값을 추출한다.

rPrDefault 는 현재 문서에 대한 기본 실행 속성 집합을 지정한다. 실제 실행 속성은 현재 요소의 rPr 하위 요소 내에 저장된다. 이 요소가 생략되면 현재 문서의 기본 실행 속성이 존재하지 않는다. 즉, 문서에 기본 실행 속성이 정의되지 않는다면 기본값은 응용프로그램에서 정의한 내용으로 표현된다.<sup>3</sup>

---

<sup>3</sup> <https://learn.microsoft.com/en-us/dotnet/api/documentformat.openxml.wordprocessing.runpropertiesdefault?view=openxml-2.8.1>

pPrDefault 는 현재 문서에 대한 기본 단락 속성 집합을 지정한다. 실제 단락 속성은 현재 요소의 pPr 자식 요소 내에 저장된다. 이 요소가 생략되면 현재 문서의 기본 단락 속성이 존재하지 않는다. 즉, 문서에 기본 실행 속성이 정의되지 않는다면 기본값은 응용프로그램에서 정의한 내용으로 표현된다.<sup>4</sup>

style 요소는 type 속성(attribute) 값으로 문단, 글자, 표, 번호 지정(Paragraph, Character, Table, Numbering\_name)을 가지고 있으며, 이 값을 통해 어떤 요소를 지정한 스타일인지 식별할 수 있다. 해당 태그의 하위 요소(name)에서 스타일 명과 함께 세부 스타일을 지정하여 style.xml 파일에 포함한다. 문서 내에 해당 필드명(paragraph\_name, character\_name, table\_name, numbering\_name)으로 정의된 내용은 없으나, [표 12]의 속성 명은 각각 문단, 글자, 표, 번호 지정에 사용된 스타일 명 리스트라는 의미로 명명된 것이다.

[표 10] style.xml 내 rPrDefault 속성 정리

파일 내 속성 경로	속성 명	속성 내용	필수 여부
[rPrDefault]-[rPr]-[rFonts]	hint	표시에 사용될 글꼴에 대한 힌트를 word 설정 (default, fareast, cs)	선택
	ascii	아스키 코드 기본 폰트	선택
	eastAsia	한글 기본 폰트 (동아시아 문자에 사용되는 글꼴)	선택
	hAnsi	안시 기본 폰트	선택
	cs	코드 기본 폰트 (복잡한 스크립트에 사용되는 글꼴)	선택
[rPrDefault]-[rPr]-[i]	val	기울임 꼴 (on/off)	선택
[rPrDefault]-[rPr]-[smallCaps]	val	소문자를 작은 대문자로 옵션 선택 (on/off)	선택
[rPrDefault]-[rPr]-[dstrike]	val	이중 취소선 효과 (on/off)	선택
[rPrDefault]-[rPr]-[Color]	val	모든 텍스트의 글꼴 색 (FF0000)	선택
[rPrDefault]-[rPr]-[kern]	val	요소는 텍스트 실행에 대한 기본 키닝(문자 간 간격)을 지정	필수
[rPrDefault]-[rPr]-[sz]	val	텍스트 실행의 문자에 대한 기본 글꼴 크기	필수
[rPrDefault]-[rPr]-[szCs]	val	텍스트 실행의 문자에 대한 기본 글꼴 크기(초기 값 저장)	필수
[rPrDefault]-[rPr]-[lang]	val	텍스트 실행의 기본 언어 (en-US) (라틴어 설정)	선택
	eastAsia	ko-KR (아시아 언어 설정)	선택
	bidi	ar-SA (복잡한 스크립트 언어 설정)	선택
[rPrDefault]-[rPr]-[rStyle]	val	해당 r 의 문자 스타일을 나타냄	필수

<sup>4</sup> <https://learn.microsoft.com/en-us/dotnet/api/documentformat.openxml.wordprocessing.paragraphpropertiesdefault?view=openxml-2.8.1>

[rPrDefault]-[rPr]-[b-cs]	val	복잡한 스크립트 문자를 굵게 설정 (on/off)	선택
[rPrDefault]-[rPr]-[i-cs]	val	복잡한 스크립트 문자를 이탤릭체로 설정 (on/off)	선택
[rPrDefault]-[rPr]-[caps]	val	소문자 텍스트를 대문자로 포맷합니다. 숫자, 문장 부호, 알파벳이 아닌 문자 또는 대문자에는 영향을 주지 않음.	선택
[rPrDefault]-[rPr]-[strike]	val	텍스트를 통해 선을 그림(on/off)	선택
[rPrDefault]-[rPr]-[outline]	val	각 문자의 내부 및 외부 테두리 표시	필수
[rPrDefault]-[rPr]-[shadow]	val	텍스트 뒤, 텍스트 아래 및 오른쪽에 그림자 추가 (on/off)	선택
[rPrDefault]-[rPr]-[emboss]	val	텍스트가 양각으로 페이지에서 올라온 것처럼 보이게 함 (on/off)	선택
[rPrDefault]-[rPr]-[imprint]	val	선택한 텍스트가 각인되거나 페이지에 눌려진 것처럼 보이게 함. (조각이라고도 함) (on/off)	선택
[rPrDefault]-[rPr]-[noProof]	val	이 실행에서 맞춤법 및 문법 오류가 무시되도록 텍스트 서식 지정. (on/off)	선택
[rPrDefault]-[rPr]-[snapToGrid]	val	현재 섹션 속성의 docGrid 요소에 지정된 문자 수와 일치하도록 줄 당 문자 수 설정 (on/off)	선택
[rPrDefault]-[rPr]-[vanish]	val	이 실행의 텍스트가 표시되거나 인쇄되지 않도록 함 (on/off)	선택
[rPrDefault]-[rPr]-[webHidden]	val	이 문서가 웹 페이지로 저장될 때 이 실행의 텍스트가 표시되지 않도록 함 (on/off)	선택
[rPrDefault]-[rPr]-[spacing]	val	문자 사이의 간격이 확장되거나 축소되는 양 나타냄(on/off)	필수
[rPrDefault]-[rPr]-[w]	val	텍스트를 현재 크기의 백분율로 가로로 늘리거나 줄임. (minInclusive = 1 / maxInclusive = 600)	선택
[rPrDefault]-[rPr]-[position]	val	기준선과 관련하여 텍스트를 옮기거나 내려야 하는 양 나타냄.	필수
[rPrDefault]-[rPr]-[highlight]	val	주변 텍스트에서 눈에 띄도록 텍스트를 강조 표시 (black/blue/cyan/green/magenta/red/yellow/white/dark-blue/dark-cyan/dark-green/dark-magenta/dark-red/dark-yellow/dark-gray/light-gray/none)	필수
[rPrDefault]-[rPr]-[u]	val	이 실행에 대한 밑줄 서식 나타냄	선택
[rPrDefault]-[rPr]-[effect]	val	이 실행에 대한 애니메이션 텍스트 효과 나타냄 (blink-background/lights/ants-black/ants-red/shimmer/sparkle/none)	필수
[rPrDefault]-[rPr]-[bdr]	val	이 실행에서 문자의 테두리 나타냄	필수
	color	테두리 색상 설정	선택
	sz	테두리 너비 설정	선택
	space	포인트의 1/8 지점에서 테두리 공간 설정	선택
	shadow	테두리에 그림자가 있는지 여부 설정(on/off)	선택
	frame	테두리를 반전시켜 프레임 효과를 만들 것인지 여부 설정(on/off)	선택
	wx:bdrwidth	외부 네임스페이스 참조	선택
[rPrDefault]-[rPr]-[shd]	val	음영 스타일 값 설정	필수

	color	전경 음영 색상 값 설정	선택
	fill	배경 채우기 색상 값 설정	선택
	wx:bgcolor	외부 네임스페이스 참조	선택
[rPrDefault]-[rPr]-[fitText]	val	텍스트가 들어갈 공간의 너비 설정.	필수
	id	맞춤 텍스트의 여러 실행을 연결하는 고유한 내부 ID 설정	선택
[rPrDefault]-[rPr]-[vertAlign]	val	기준선을 기준으로 텍스트의 세로 위치를 조정하고 가능한 경우 글꼴 크기를 변경합니다. 글꼴 크기를 줄이지 않고 텍스트를 올리거나 내리려면 위치 요소 지정 (baseline/superscript/subscript)	필수
[rPrDefault]-[rPr]-[rtl]	val	정렬 및 읽기 순서를 오른쪽에서 왼쪽으로 설정(on/off)	선택
[rPrDefault]-[rPr]-[cs]	val	텍스트가 복잡한 스크립트 텍스트인지 여부 지정 (true 또는 false).	선택
[rPrDefault]-[rPr]-[em]	val	강조 표시 유형 설정 (none/dot/comma/circle/under-dot)	필수
[rPrDefault]-[rPr]-[hyphen]	val	하이픈 넣기 스타일 (1 포인트의 20 분의 1, 1 인치의 1/1440) 단위의 양수 측정 값 설정	필수
[rPrDefault]-[rPr]- [asianLayout] 특별한 아시아 레이아웃 서식 속성을 나타냅니다.	id	여러 아시아 텍스트 실행을 연결하는 고유한 내부 ID 설정	선택
	combine	줄을 결합할지 문자를 결합할지 나타내는 값 설정 (lines/letters)	선택
	combine-brackets	결합된 텍스트 주위에 넣을 대괄호 스타일 설정 (none/round/square/angle/curly)	선택
	vert	세로 텍스트로 제대로 표시되도록 아시아 반각 문자의 회전 값 설정합니다. (on/off)	선택
	vert-compress	한 문자 단위에 맞도록 회전된 텍스트의 압축 값 설정 (on/off)	선택
[rPrDefault]-[rPr]- [specVanish]	val	텍스트를 항상 숨김으로 만드는 특수 숨겨진 속성을 나타냄 (on/off)	선택
[rPrDefault]-[rPr]-[wx:font]	val	기본 편지 병합 문서가 표시하는 활성 레코드 지정	필수
[rPrDefault]-[rPr]-[wx:sym]	val	기본 편지 병합 문서가 표시하는 활성 레코드 지정	필수
[rPrDefault]-[rPr]- [aml:annotation]	val	기본 편지 병합 문서가 표시하는 활성 레코드 지정	필수

[표 11] style.xml 내 pPrDefault 속성 정리

파일 내 속성 경로	속성 명	속성 내용	필수 여부
[pPrDefault]-[spacing] 문단 뒤의 간격과 행 간격을 포함하여 문단에 대한 기본 속성을 지정	after	문단 뒤의 기본 간격	선택
	after-lines	단락 뒤 줄 수 설정(문자 단위 사용 시).	선택
	after-autospacing	단락 뒤의 공백/줄이 자동인지 여부 설정	선택
	beforeLines	[간격]-[단락 앞] 설정된 값(단위가 "줄"이면 설정됨)	선택
	before	[간격]-[단락 앞] 설정된 값	선택
	before-autospacing	단락 앞의 공백/줄이 자동인지 여부 설정	선택
	line	줄 간격을 지정	선택
	lineRule	문단 또는 텍스트 실행의 줄 간격을 결정하는 데 사용되는 규칙을 지정(auto/exact/at-least)	선택
[pPrDefualt]-[ind]	leftChar	들여쓰기 왼쪽 간격 - 글자간격으로 설정된 경우 확인	선택
	left	들여쓰기 왼쪽	선택
	rightChar	들여쓰기 오른쪽 간격 - 글자 간격으로 설정된 경우 확인	선택
	right	들여쓰기 오른쪽	선택
	hanging	처음 이후의 모든 행에 내어쓰기 지정	선택
	hanging-chars	첫 번째 이후의 모든 줄에 대해 문자 공간 내어쓰기를 지정 (문자 단위를 사용하는 경우).	선택
	first-line	첫 번째 줄에만 들여쓰기 지정 (걸기 속성과 함께 사용할 수 없음).	선택
	firstLineChars	들여쓰기 첫줄 간격 - 글자간격으로 설정된 경우만 확인됨	선택
[pPrDefault]-[jc]	val	문단에 대한 기본 정당성을 지정 (left/center/right/both/medium-kashida/distribute/list-tab/high-kashida/low-kashida/thai-distribute)	선택

[표 12] style.xml 내 문단, 글자, 표 번호의 스타일 명 속성 리스트

파일 내 속성 경로	속성 명	속성 내용	필수 여부
type:paragraph_name	[name]-[val]	문단 스타일 리스트	필수
type:character_name	[name]-[val]	글꼴 스타일 리스트	필수
type:table_name	[name]-[val]	표 스타일 리스트	필수
type:numbering_name	[name]-[val]	번호 지정 스타일 리스트	필수
type:paragraph_name	[name]-[val]	문단 스타일 리스트	필수

### 8.3 FONTTABLE.XML 특징

/word/fontTable.xml : 문서에 정의된 글꼴 정보를 포함하는 XML 파일이다. 문서 내 사용되는 다양한 글꼴들의 정보와 속성들이 여기에 정의된다. 각 글꼴의 이름, 크기, 스타일 등이 이 파일에 포함된다.

fontTable.xml 파일 내 자식 요소 중 "font" 태그를 가진 노드의 자식 요소를 순회하면서 해당 요소의 "name"(속성)의 키와 값을 추출한다.

[표 13] fontTable.xml 문서 내 정의된 속성 정리

파일 내 속성 경로	속성 명	속성 내용	필수 여부
[font]	name	폰트 이름 리스트 가져오기	필수

### 8.4 HEADER#N.XML/FOOTER#N.XML 특징

/word/header#N.xml, /word/footer#N.xml : 문서의 머리말(header)과 꼬리말(footer) 내용을 정의하는 XML 파일이다. 각 파일은 문서의 각 섹션에 대한 머리말과 꼬리말 내용을 정의한다. 머리말(header)과 꼬리말(footer)에 들어갈 텍스트, 그림, 페이지 번호 등의 이 파일에 포함된다. N 은 섹션 번호를 나타내며, 여러 섹션에 대한 머리말과 꼬리말이 따로 정의된다.

[표 14] headerN.xml/footerN.xml 문서 내 정의된 속성 정리

파일 내 속성 경로	속성명	속성 내용	필수 여부
headerN.xml footerN.xml	pStyle	단락 스타일	선택
	keepNext	다음 단락과 함께 유지 옵션 나타냄, 이 단락과 다음 단락 사이의 페이지 나누기 방지.	선택
	keepLines	Keep Lines Together 옵션 나타냄, 단락에서 페이지 나누기 방지	선택
	pageBreakBefore	이전 페이지 나누기 옵션 나타냄. 이 단락 앞에서 페이지 나누기 강제 실행	선택
	framePr	텍스트 프레임 및 첫 문자 장식 속성을 나타냄	선택
		첫 문자 장식의 위치 설정	선택
		드롭 캡에 대해 높을 줄을 설정	선택
		프레임 너비 설정	선택
		프레임 높이 설정	선택

		프레임과 프레임 위와 아래의 텍스트 사이의 거리 설정	선택
		프레임과 프레임의 오른쪽과 왼쪽에 있는 텍스트 사이의 거리 설정	선택
		텍스트 줄 바꿈을 설정(유효: 없음 및 주변).	선택
		수평 위치/정렬을 측정할 지점 설정	선택
		수직 위치/정렬을 측정할 지점 설정	선택
		가로 위치 설정	선택
		가로 정렬 설정(위치 재정의).	선택
		세로 위치 설정	선택
		수직 정렬 설정(위치 재정의).	선택
		높이(h)를 해석하는 방법을 설정	선택
		프레임의 앵커를 현재 포함하고 있는 단락으로 잠금	선택
	widowControl	Widow/Orphan 제어 옵션을 나타냄: Word 가 단락의 마지막 줄을 페이지 상단(widow)에 인쇄하거나 단락의 첫 줄을 페이지 하단(orphan)에 인쇄하지 않도록 함	선택
	supressLineNumbers	단락 옆에 줄 번호가 표시되지 않도록 함. 이 설정은 줄 번호가 없는 문서나 섹션에는 비적용	선택
	shd	단락 음영	선택
		음영 스타일 값 설정	선택
		전경 음영 색상 값 설정	선택
		배경 채우기 색상 값 설정	선택
		외부 네임스페이스 참조됨	선택
	suppressAutoHyphe ns	자동 하이픈 넣기 방지	선택
	kinsoku	동아시아 타이포그래피 및 줄바꿈 규칙을 사용하여 페이지에서 줄을 시작하고 끝내는 문자 결정 (아시아 타이포그래피 옵션).	선택
	wordWrap	라틴어 단어 중간에서 줄 바꿈 허용(아시아 타이포그래피 옵션).	선택
	overflowPunct	문장 부호가 단락의 다른 줄 정렬을 넘어 한 문자를 계속하도록 허용. 이 옵션을 사용하지 않으면 모든 줄과 문장 부호가 완벽하게 정렬되어야 함(아시아 타이포그래피 옵션).	선택
	topLinePunct	줄 시작 부분에서 문장 부호를 압축할 수 있으므로 후속 문자가 더 가깝게 이동할 수 있음(아시아 타이포그래피 옵션).	선택
	autoSpaceDE	동아시아 텍스트와 라틴 텍스트 사이의 문자 간격을 자동으로 조정(아시아 타이포그래피 옵션).	선택
	autoSpaceDN	동아시아 텍스트와 숫자 사이의 문자 간격을 자동으로 조정(아시아 타이포그래피 옵션).	선택

	bidi	단락의 정렬 및 읽기 순서를 오른쪽에서 왼쪽으로 설정	선택
	adjustRightInd	문서 격자를 사용할 때 올바른 들여쓰기를 자동으로 조정	선택
	snapToGrid	텍스트를 문서 격자선에 맞춤(격자선이 정의된 경우).	선택
spacing		줄과 단락 사이의 간격 나타냄	선택
		단락 위의 공간 설정	선택
		단락 앞 줄 수 설정(문자 단위를 사용하는 경우).	선택
		단락 앞의 공백/줄이 자동인지 여부 설정	선택
		단락 아래의 공간 설정	선택
		단락 뒤 줄 수 설정(문자 단위 사용 시).	선택
		단락 뒤의 공백/줄이 자동인지 여부 설정	선택
		텍스트 줄 사이의 세로 간격 설정	선택
		선 속성의 해석 지정	선택
		단락 들여쓰기 나타냄	선택
ind		왼쪽 여백과 텍스트 사이의 간격을 지정. 음수 값은 텍스트를 여백으로 이동	선택
		왼쪽 여백과 텍스트 사이의 문자 간격 지정(문자 단위를 사용하는 경우). 음수 값은 텍스트를 여백으로 이동	선택
		텍스트와 오른쪽 여백 사이의 간격 지정. 음수 값은 텍스트를 여백으로 이동	선택
		텍스트와 오른쪽 여백 사이의 문자 간격 지정(문자 단위를 사용하는 경우). 음수 값은 텍스트를 여백으로 이동	선택
		처음 이후의 모든 행에 내어쓰기 지정	선택
		첫 번째 이후의 모든 줄에 대해 문자 공간 내어쓰기 지정(문자 단위를 사용하는 경우).	선택
		첫 번째 줄에만 들여쓰기 지정(걸기 속성과 함께 사용할 수 없음).	선택
		첫 줄에만 문자 간격 들여쓰기 지정(hanging-chars 속성과 함께 사용할 수 없음).	선택
		동일한 스타일의 단락 사이에 공백을 추가하지 않도록 지정	선택
		이 프레임이 겹치지 않도록 지정	선택
	jc	단락 정렬	선택
	textDirection	현재 셀, 텍스트 상자 또는 텍스트 프레임에서 단락의 방향	선택
	textAlignment	한 줄에 있는 모든 텍스트의 수직 정렬을 결정(아시아 타이포그래피 옵션).	선택
	outlineLvl	개요 수준	선택
	divId	이 단락이 현재 있는 HTML DIV 요소의 ID	선택
	aml:annotation	외부 네임스페이스 참조	선택