KJCEM 21. 6. 125~134
November 30, 2020

www.kicem.or.kr (pISSN 2005-6095 / eISSN 2465-9703)
http://dx.doi.org/10.6106/KJCEM.2020.21.6.125

문자열 유사도 알고리즘을 이용한 공종명 인식의 자연어처리 연구
- 공종명 문자열 유사도 알고리즘의 비교 -

정상원[1] · 정기창[2*]
[1]한국산업융합연구원 선임연구원 · [2]한국산업융합연구원 원장

Comparing String Similarity Algorithms for Recognizing Task Names Found in Construction Documents

Jeong, Sangwon[1], Jeong, Kichang[2*]
[1]Senior researcher, Korea Institute of Industry Convergence
[2]President, Korea Institute of Industry Convergence

Abstract : Natural language encountered in construction documents largely deviates from those that are recommended by the authorities. Such practice that is lacking in coherence will discourage integrated research with automation, and it will hurt the productivity in the industry for the long run. This research aims to compare multiple string similarity (string matching) algorithms to compare each algorithm's performance in recognizing the same task name written in multiple different ways. We also aim to start a debate on how prevalent the aforementioned deviation is. Finally, we composed a small dataset that associates construction task names found in practice with the corresponding task names that are less cluttered w.r.t their formatting. We expect that this dataset can be used to validate future natural language processing approaches.

Keywords : Natural Language Processing, String Approximation, String Matching, String Similarity

1. 서론

1.1 연구의 배경

문서 작성과 문서 편집의 디지털화와 자동화는 어떠한 산업에서도 생산성 증진을 도모한다. 가장 생활과 밀접한 예를 들자면, 데이터를 디지털 스프레드시트에 기재하기 시작한 것을 생각할 수 있다. 이를 통해 데이터의 검색, 수정, 그리고 배포 등이 쉬워졌고, 지금은 디지털 스프레드시트 없이는 사무적인 일을 처리할 수 없는 수준에 이르렀다. 또한, 스프레드시트의 여러 자동화 기능을 통해 실무자들은 예전과는 비교할 수 없는 속도로 데이터를 다룬다.

타 분야에서도 그렇듯이, 건설업계의 문서에서 사용되는 수없이 많은 단어와 공종명 등의 불규칙성은 생산적이라기보다는 비생산적이라고 표현할 수 있고, 이는 생산성 향상을 위한 정보의 표준화의 장애 요인이 된다. 특히 건설 프로젝트의 내역서에서 공종명을 표기하는데 있어서 실무자들은 업계에서 사용하기로 약속된 규격화된 명칭보다는 실무자의 관점에서 의미가 더욱 잘 통하는 명칭을 사용하는 모습을 어렵지 않게 접할 수 있고, 명칭을 디지털 문서로 만들 때 실무자의 가독성과 작성 편의만을 중시하는 방식을 선호한다. 가장 손쉽게 접할 수 있는 예로는 띄어쓰기의 남용을 생각할 수 있다. 아울러 정확한 문법 및 철자에 대한 개인의 지식 차이에 의한 표기 차이도 발생한다. 또한, 스트레드시트에 데이터를 입력할 때 하나의 셀에 공종, 규격, 단위 등의 여러 정보가 혼재된 경우도 빈번하다. 〈Table 1〉이 이러한 행태의 예시를 보여준다. 모든 표의 한글은 영어발음식 표기로 전환하여 표기하였음을 참고하기 바란다. 처음의 예시는 스프레드시트의 셀을 꽉 채우기 위해서 띄어쓰기를 남용하는 경우이다. 두 번째는 서로 다른 계층의 공종명을 혼재하는 경우이다. 세 번째는 공종명과 규격 등을 같이 표기하는 방식이며, 이는 문자열을 자동적으로 처리하려 할 때 큰 걸림돌이 될 수 있다. 본 바와 같이 사람에게는 편할 수 있

* **Corresponding author:** Jeong, kichang, Korea Institute of Industry Convergence, 501, 22, Guuigangbyeon-ro, Gwangjin-gu, Seoul, Republic of Korea
E-mail: therza@hanmail.net
Received September 9, 2020: **revised** October 13, 2020
accepted October 19, 2020

는 표기법은 장기적으로 문서들을 디지털화하고, 이를 자동으로 처리할 수 있는 자연어 처리 시스템(Natural Language Processing)을 개발하는 데에는 매우 큰 난관을 제공한다.

Table 1. Problematic input in construction-related documents

Correct Usage		Actual Usage in the Industrial Settings	
Towgong		Tow	gong
Baesu-gong	Towgong	Baesu-gong-1]	Tow gong
Tile Buchigi	Dogijil tile	Tile (Dogijil) Butchim	

자연어 처리란, 한글과 영어와 같이 인간이 사용하는 언어를 컴퓨팅을 통해 가공하고 사용하는 응용 분야나 연구를 지칭한다. 자연어는 추후 컴퓨터를 통해 사용될 것을 고려하고 만들어지지 않았기 때문에, 연구에 여러 가지 어려움이 존재한다. 과거에는 이러한 자연어에 대한 지식을 컴퓨터가 이해할 수 있게 인코딩하는 것에 주안점이 맞춰져 있었다. 문법 법칙을 일일이 손수 하드 코딩한다든지, 단어와 단어 간의 관계를 수동적으로 정의하는 등 사람의 노동력이 많이 필요한 연구 방향이 지배적이었다. 하지만, 처리해야 할 정보의 양이 큰 폭으로 늘어나고, 시스템의 행동 법칙을 명시하는(Rule-based) 방식의 한계점에 봉착하자 자연어 처리는 기계학습으로 서서히 눈을 돌리게 되고, 자연어 내의 규칙과 단어와 단어 간의 관계 등은 비지도 학습을 사용하여 자동으로 인코딩될 수 있는 방향을 추구하게 되었다. 하지만 아직 건설업계에서는 이러한 기계학습을 사용하기가 힘든데, 이는 기계학습 시스템을 학습시킬 가공된 데이터의 불충분과 자동화에 관한 관심의 부재가 원인으로 보인다(Bilal et al., 2016).

건설업계에서 데이터의 디지털화와 규격화는 광범위하고 공개적으로 이뤄지지 않고 있다. 단편적인 예를 들자면 건설 현장에서 생성되는 이미지 데이터를 생각할 수 있다. Han and Golparvar-Fard (2017)가 조사한 바에 의하면 전형적인 건설 프로젝트는 대략 400,000개 정도의 시각 데이터를 생성한다. 하지만 이러한 데이터들은 체계적으로 정리되지 않아서 사용이 어렵다(Kim, 2019). 데이터의 생산 비용과 그 경제적 가치 때문에 회사마다 각자 사유의 데이터베이스를 제작, 소지하고 있는 경우가 존재할 가능성을 열어둔다. 큰 그림 속 이러한 행태는 방대한 정보를 빠르게 처리하기 어려운 소규모 사업체에게 비생산성을 초래할 것이고, 업계에서 정보의 비대칭성에 의한 불균형이 가속화될 수 있다. 또한, 건설 관련 데이터가 연구계에 도달하지 못함으로 인해 (Lee 2019) 에서 지적하듯이 건설업계에서의 자연어 처리 연구가 지연되고, 깊이 이루어지지 못한다. 이는 데이터에 대한 접근성이 낮고, 데이터의 양과 질 또한 저조하다는 것을 시사한다.

건설업계에서 자연어 처리가 빛을 볼 수 있는 분야는 다양하다. 건설 법규문서에서 정보를 추출하여 건축물이 안전 법규 등을 준수하였는지를 자동으로 검사하는 작업(Zhang and El-Gohary, 2016)을 할 수도 있고, 건설 입찰 정보를 토대로 공사비의 증가를 예측하는 모델을 개발할 수도 있다(Williams and Gong, 2014), 또한 건설사업과 관련한 문서의 리스크를 키워드 중심으로 분석하기 위한 자료로 활용될 수 있으며(An et al., 2019), 국내 건설산업에서의 실패정보를 사례 중심으로 수집하고 활용하기 위한 기술로도 활용할 수 있다(Park, 2008). 그러나 자연어 처리에 있어서 법률상 용어를 컴퓨터가 판별할 수 있는 언어를 함수로 변환하여야 하는 과정으로서 룰셋 정의서를 개발하는 등 Kim (2012)의 접근 이외의 실무에서 쓰이는 자연어 처리에 관한 연구는 부진한 상황이다.

본 연구는 Lee et al. (2012)에서 BIM (Building Information Modeling) 정보와 Onthology(개체와 개체들간의 관계에 대한 정보)를 사용하여 표준내역항목 추론을 자동화하는 연구와 관련성이 깊다. 선행 연구에서는 타일공사를 중심으로 물량 내역서의 항목을 자동으로 추론하는 방법으로 온톨로지를 사용하는 것을 제시하였다. 하지만, 그들은 실제 문서 작성 프로그램을 만들기 위한 정보추출 및 데이터형식 변환 모듈, 시멘틱 추론 모듈 등을 구축하지 못하였음을 강조하였다. 본 연구는 개체와 개체간의 관계를 정의하는 온톨로지 방법론에 기반을 두고 있지 않고, 개체와 개체의 관계를 단지 문자열 유사도를 통하여 측량하는 방법을 채택하여 선행연구에서 요구하는 시멘틱 추론 모듈등의 필요성을 우회하는 하나의 방법을 제시한다.

1.2 연구의 목적과 기여

본 연구는 한글 건설문서를 활용한 자연어 처리 연구를 활성화하고, 앞에서 지적한 건설업계에서의 자연어 처리 연구에 걸림돌이 되는 장벽을 하나씩 허물어갈 추후 연구들의 주춧돌이 될 수 있는, 공종명을 포함하는 간단한 데이터셋을 구축한다.

다음으로, 구축된 데이터셋을 활용하여 몇몇 간단한 단어 유사도 계산 기법들의 정확성을 비교한다. 같지만 다르게 표현된 공종명을 성공적으로 인식하는 시스템은 물량 내역서를 가지고 사업비를 산출하는 작업(적산의 일부)을 자동화할 수 있다. 우리는 문자열 유사도 알고리즘을 사용해 가장 간단한 방식으로 물량 내역서의 공종명과 자체로 구축한 공종명 데이터베이스 간 인식이 가능한지를 보고자 한다. 이는 추후 개체명 인식(Named Entity Recognition) (Nadeau and Sekine, 2007)과 같은 자연어 처리 기법을 통해 건설문

서를 분석, 수정, 작성 등 더욱 복잡한 작업을 수행하는 시스템을 설계하는 데 있어 기본적인 이해를 제공할 수 있을 것으로 보인다. 마지막으로, 데이터셋은 기계학습 기반의 문자열 유사성 연구를 진행할 더욱 큰 데이터셋을 만드는 데 지침이 될 수 있을 것으로 보인다.

2. 문자열 유사성

2.1 유사성 계산

문자열 유사성, 더욱 구체적으로 본 연구에서 탐구하는 스트링 매칭은 자연어 처리 분야에서 매우 기초적인 작업이다. 이는 검색엔진이나 오타 검사와 같은 생활과 밀접한 서비스에서도 사용되는 기법으로(Lalwani et al., 2014), 기본적으로 두개의 서로 다른 문자열의 거리를 측정하는 작업을 의미한다. 두 문자열의 거리는 낱말별 차이일 수도 있고, 의미론적 거리일 수도 있다. 어떠한 거리를 사용할지는 응용시스템의 성격, 최종 목표 등에 따라 달라진다. 만약 단순 철자 오타 검사와 같이 대충 두 단어가 일치하거나 비슷한지를 확인하기만 하면 된다면 구태여 의미론적 거리를 추론해 낼 필요가 없을 것이다. 하지만, 검색하고자 하는 단어와 비슷한 개념을 도출해 내야 하는 검색엔진의 경우는 의미론적 거리가 중요할 수 있다. 더욱 구체적인 이해를 돕기 위해 예로 “아스팔트 부수기”라는 공종명과 “아스콘 깨기”라는 공종명을 생각해보자. 두 명칭을 낱말 별로 비교하면 “아스” 두 글자를 빼고는 모두 다르므로 거리가 멀 것이다. 반면에 두 공종명의 의미상 거리는 매우 가깝다는 것을 알 수 있다.

본 연구에서는 여러 가지 문자열/단어 유사성 알고리즘을 이용하여 실무자들이 작성한 건설 내역서에 나오는 공종명과 한국건설기술연구원에서 발표한 2020년 건설공사 표준품셈에 명시되어있는 가이드라인을 참조해서 작성한 타일공사와 관련한 소규모 공종명 데이터베이스를 구축하였다. 우리는 이 데이터베이스와 실무에서 발견되는 공종명의 단어 유사성의 인식 가능성을 서로 다른 문자열 유사도 알고리즘을 통해 살펴본다.

2.2 유사성 알고리즘

유사성 알고리즘으로는 대표적인 N-Gram, Hamming Distance, Levenshtein Distance, 그리고 Jaro-Winkler Distance를 사용한다. 〈Fig. 1〉의 Psedocode는 모든 거리 계산 알고리즘들이 사용되는 틀이다. DISTANCEALGO는 언급한 4가지의 알고리즘의 자리이며, 모두 공통된 형식의 출력을 가지고 있다.

ALGORITHM 1: Abstract procedure for string matching

```
Input  : W– list of known words (database)
         t – a word to match
Output: E – (matched closest word, distance) tuple
r ← 0                                   /* maximum distance counter (higher = closer) */
m ← NULL                                /* matched word */
for w ← W do
    d ← DISTANCEALGO(t, w)              /* calculates a distance between a word from the
     database and a word to match */
    if d > r then
        m ← w                           /* update matched word */
        r ← d                           /* update distance counter */
    end
end
E ← (m, r)                              /* return final result */
```

Fig. 1. Abstract procedure for string matching algorithms

2.2.1 N-Gram

N-Gram은 문자열을 N개로 이루어진 토큰으로 나누고, 두 문자열의 토큰들을 하나씩 비교하여 전체 문자열의 유사성을 판단하는 방식이다. 예를 들어 “아스팔트 부수기”라는 문자를 2개로 이루어진 토큰으로 나눈다면 “아스”, “스팔”, “팔트”, “트 “, “ 부”, “부수”, “수기” 와 같이 총 7개의 토큰으로 나누어진다. 이렇게 나눠진 토큰은 상대 언어의 토큰 배열에서 같은 위치에 있는 토큰과 비교되며, 틀리면 패널티를 준다.

2.2.2 Hamming Distance

Hamming Distance는 떠올릴 수 있는 가장 직관적인 방법으로, 두 문자열을 있는 그대로 비교하는 방법이다. 이 방법은 두 개의 문자열 A와 B를 낱말 별로 1대 1로 비교하는 방법으로, 만약 같은 위치에 있는 낱말이 다르다면 페널티를 준다. 예를 들어 “아스팔트”와 “아스콘” 두 단어를 놓고 보았을 때, “팔”과 “콘”이 틀리고, “트”와 대응하는 단어가 아스콘에는 없으므로 또한 다르다고 취급된다.

2.2.3. Levenshtein Distance

Levenshtein Distance는 편집 거리(Edit Distance)라는 개념을 사용한다. 쉽게 이야기하자면, 문자열 A가 문자열 B가 되기 위해서 몇 번의 편집을 해야 하는지를 측정하는 것이다. 편집 거리를 산정하는 가장 기본적인 편집의 방법은 총 3가지가 있다. 첫 번째로 삽입(Insertion)은 문자열 내 같은 자리에 새로운 낱말을 채워 넣는 편집이다. 두 번째로 제거(Deletion)은 문자열 내 같은 자리에 있는 낱말을 제거하는 연산이다. 그리고 마지막으로 대체(Substitution)은 문자열 내 같은 자리에 있는 낱말을 다른 낱말로 바꾸는 연산이다. “아스팔트”와 “아스콘”을 각각 A와 B라고 가정하고, A를 B로 변환시키는 편집 거리를 구해본다. 문자열의 세 번째 자리에 있는 “팔”을 “콘”으로 변화시키는 편집은 대체, 그리고 “트”는 대응하는 낱말이 없으므로 제거 연산을 해야 한다. 따라서 위 예에서 필요한 편집은 총 2개가 된다.

2.2.4. Jaro-Winkler Distance

Jaro-Winkler Distance도 Levenstein Distance와 마찬가

지로 편집 거리의 개념을 사용한다. 하지만, 위의 경우와 다르게 두 가지의 경우에 더욱 높은 관련도 점수를 할당하는데 이는 첫 번째, 같은 낱말이 문자열 내 특정한 거리 내에 있을 때. 즉, “부수기”와 “깨기”를 비교하였을 때, 낱말 “기”는 각각 문자열의 3번째와 2번째에 자리 잡고 있지만, 그 거리가 가까우므로 같은 일치하는 것으로 계산하는 것이다. 두 번째로는 문자열의 시작점부터 A와 B의 문자열이 일치하기 시작하는 것에 높은 점수를 주게 된다. 이는 알고리즘에 방향성을 부여하게 된다. 방향성은 문자열이 일치하는 방향이 같은 것을 중요시하는 것이다.

2.2.5 유사성 알고리즘의 결과물 예시

소개된 알고리즘이 반복적으로 언급된 예시 단어 “아스팔트 부수기”와 “아스콘 깨기” 사이의 거리를 어떻게 비교하는지 살펴보겠다. 모든 거리는 0과 1사이로 정규화 되었고, 1에 가까울수록 가까움을 의미한다.

Table 2. Distance example

Algorithm	Distance
Hamming	0.571
N-Gram (N=2)	0.166
N-Gram (N=3)	0.0
Levenshtein	0.714
Jaro-Winkler	0.78

의미론적인 거리를 따진다면 두 단어의 거리는 1에 매우 가까워야 한다. 하지만 〈Table 2〉에서 볼 수 있듯이 가장 가까운 거리를 계산한 알고리즘은 Jaro-Winkler이고, N-Gram은 낱말 뭉치의 크기에 따라 거리의 변화가 심하다.

2.3 데이터 수집 및 가공

테스트할 공종명 데이터는 첫째, 실제 업무에서 각기 다른 사람들이 사용하는 공종명을 포함해야 하고, 둘째로 그 실무자들이 공종명을 작성하는 양상을 토씨 하나 틀리지 않고 잘 보존해야 한다. 이는 추후 더욱 복잡한 업무를 수행할 프로그램을 만들 때, 사용하는 데이터에 최대한 손을 대지 않아도 되는 프로그램을 만드는 데 유리하기 때문이다. 따라서 본 연구에서는 공종명을 나라장터 국가종합전자조달 시스템에 올라와 있는 부분적으로만 조직적인 데이터(Semi-structured data) (Abiteboul, 1997)에서 건설 관련 입찰 문서에서 추출하였고, 그 문서에서 공종명 테스트 데이터를 추출할 때에는 공종명 이하 표에 적혀있는 그대로의 문자열을 불러왔다.

문자열 유사성 알고리즘의 성능을 평가하기 위해서 타일 공사 관련 공종명만 고려하였다. 모든 공종을 고려하는 것이 더욱 의미가 있지만, 모든 정식 공종명을 포함하는 데이터베이스가 아직 구축되어있지 않기 때문에 현실적인 제약을 따르는 것이다. 정식 공종명 데이터베이스에 존재하지 않는 깨끗하지 않은 문자열 원본과 일대일로 대응하는 정식 문자열이 있어야 한다. 따라서 위에 언급한 실제 업무에서 사용된 문자열 원본(684개)을 사용하여 검증 단계에서 사용할 정식 공종명 데이터베이스를 만들었다. 정식 공종명은 간략화를 도모하고 공종명의 구조적인 특징을 잘 이용하기 위해 〈대상〉과 〈대상을 가지고 행하는 작업〉 두 단어로 정의하였다. 다음의 〈Table 3〉에 예시를 정리하였다.

Table 3. Task name label samples

Origianl	Target Material	Action
Terrajotilecheolgeo 〈테라죠타일철거〉	Terrajotile 〈테라죠타일〉	Cheolgeo 〈철거〉
PVC Tile Buchim 〈PVC 타일 붙임〉	PVCTile 〈PVC타일〉	Buchim 〈붙임〉
Tile Apchak Buchigi 〈타일 압착 붙이기〉	Tile 〈타일〉	Apchak Buchigi 〈압착붙이기〉

3. 실험의 설계

3.1 문제의 정의

본 연구에서 탐구하고자 하는 문제는 다음과 같다. 비정형 텍스트 원문 (A)을 정형적 데이터로 쉽게 변환시킬 수 있는 비정형 텍스트 (B)로 맵핑하는 것이 최종 목표이다. 비정형 텍스트를 추후에 여러 가지 응용 과제에 손쉽게 사용될 수 있는 정형적 데이터로 변환하기는 쉽다고 가정한다. 본 문제를 모식도로 표현하면 다음 〈Fig. 2〉과 같다.

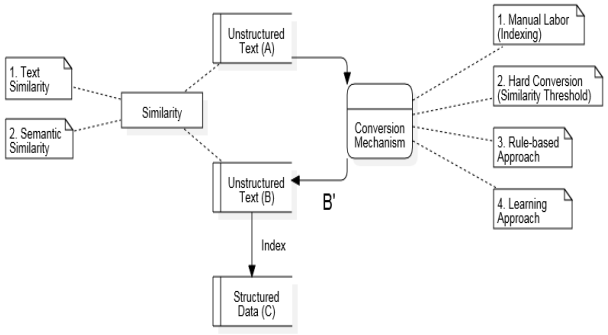

Fig. 2. Overall structure of the task

모식도의 Conversion Mechanism은 A를 B로 맵핑하는 추상적인 프로세스(Abstraction)이다. 본 프로세스의 출력은 B’로 정의한다. B’는 A와 B간의 거리일 수도 있고, 프로세스가 판단한 B와 가장 가까운 문자열일 수도 있다. 전자의 경우 B와 B’의 거리가 가까우면 가까울수록 성공적인 프로세

스로 볼 수 있고, 후자의 경우 B'와 B의 거리가 가까우면 성공적인 프로세스로 볼 수 있다.

이 프로세스는 다시 크게 네 가지의 서로 다른 방법으로 나뉠 수 있는데, 첫 번째인 Manual Labor은 A에 있는 모든 문자열에 대응하는 B 또는 C의 문자열 혹은 데이터를 수동으로 정의하여 색약(Index) 데이터베이스를 만드는 방법이다. 이 방법에는 인간 작업자의 정수가 들어가 있으므로 가장 정확할 수 있지만, 구축하는데 시간과 비용이 너무나 많이 소요되고 또한 색약에 포함되어 있지 않은 문자열은 사용할 수 없으므로 전혀 유연하지(Not generalizable) 못한 방법이다. 두 번째인 Hard Conversion, 즉 강제 변환은 서로 다른 두 문자열의 유사성 척도 등(통상 Text Similarity)의 수치를 계산한 뒤, 설정된 임계치(Threshold)에 따라 A와 B가 동일한 문자열인지 상이한 문자열인지를 판별하는 방식이다. 이는 첫 번째 방법보다 시스템 구축이 쉽다는 장점이 있고, 더욱 유연하지만, 문자열의 개념을 고려하지 못한다. 세 번째는 Rule-based, 즉 사람이 정의한 여러 법칙에 따라 A를 B로 변환시키는 것이다. 이는 이전의 방법들보다 유연할 수 있고, 시스템의 판단 경위(Decision process)를 보다 쉽게 이해할 수 있다. 하지만 법칙을 정의하는데 각기 다른 분야마다 전문가(Domain expert)가 필요하고, 이처럼 만들어진 시스템은 다른 분야에 사용하기 어려울 수 있으므로 유연성이 떨어진다. 같은 분야 내에서도 하고자 하는 작업이 조금이라도 달라지면 큰 힘을 발휘하지 못할 수 있다.

마지막으로 Learning Approach는 기계학습 등을 이용하여 프로세스가 A에서 B로 변환하는 능력을 학습할 수 있게 하는 방법이다. 이 방법은 문자열의 의미론적 유사도를 고려할 수 있고 가장 유연한 방법이라는 측면에서 매력적이다. 하지만 학습을 시키기 위해서는 통상 방대한 학습 데이터셋이 필요하므로 이는 본 논문의 범위에서 벗어나는 방법이다.

본 연구에서는 두 번째 강제 변환 방법을 통해 건설 관련 문서에 사용되는 문자열이 얼마나 쉽게 A에서 B로 변환될 수 있는지를 탐구하기 위해 A와 B 간의 유사성(Similarity), 그중에서도 첫 번째인 Text Similarity에 집중한다. Manual Labor은 그 정확도는 확실하나, 데이터베이스를 구축하는 것이 불가능하다. Rule-based 방법을 본 연구에 적용하기에는 저자들의 건설 산업의 자연어에 대한 이해가 부족하다. 마지막으로 기계학습을 사용하기에는 데이터셋이 턱없이 작다. 따라서 강제 변환 방식을 택하게 되었다. 이에 따라서 기계학습을 통해 얻을 수 있는 의미론적 거리도 자연스럽게 본 연구의 범위를 벗어나며, 고려하지 않는다.

3.2 유사성 알고리즘의 성능 비교 방법

2장에서 정의된 네 가지의 유사성 알고리즘을 비교한다. 하지만, 첫 번째 N-Gram은 N=2일 경우와 N=3일 경우의 두 가지를 사용하기 때문에 실질적으로 〈Table 4〉에 명시되었듯이 총 다섯 개의 알고리즘을 비교한다. N-Gram의 경우 문자열의 길이가 N보다 작다면 그 문자열의 길이가 N보다 커질 때까지 N을 1씩 감소시켜서 연산하였다.

Table 4. Short-hand symbols for compared algorithms

Symbol	Algorithm Full Name
N2	N-Gram (N=2)
N3	N-Gram (N=3)
HAM	Hamming Distance
LEV	Levenshtein Distance
JAR	Jaro-Winkler Distance

위 알고리즘들이 서로 다른 이유는 서로 다른 거리 계산 방법을 가지고 있기 때문인데, 이 때문에 알고리즘이 출력하는 거리를 바탕으로 이의 성능을 객관적으로 평가하는 것이 어렵다. 따라서 본 연구에서는 모든 알고리즘이 출력하는 거리값을 0~1 사이의 값으로 정규화(Normalize) 시키고, 이를 B'(A와 B사이의 거리)로 간주한다. 그 거리 B'의 값이 가장 가까운 것을 출력 단어로 선택한다. 예를 들어서, B에는 "아스팔트"와 "콘크리트"라는 단어가 있고, A에는 "아스콘"이 있다고 가정하자. 그러면 위 알고리즘들은 A의 두 단어에 대한 각기 다른 거릿값을 출력할 것이다. Levenshtein distance의 경우 "아스팔트"와 "콘크리트"에 대한 "아스콘"의 거리값은 각각 0.71과 0.43이다. 이런 경우에 이 알고리즘이 판단한 가장 "아스콘"에 근접한 단어는 "아스팔트"가 되는 것이다.

거리값이 가장 가까운 단어가 만약 A의 올바른 표기법 B라면 정답 (1)으로, 그렇지 못하면 오답 (0)으로 취급한다. 최종적으로 이 정오값을 모두 더한 뒤, 테스트 셋의 크기로 나누면 나오는 0과 1사이의 점수를 바탕으로 알고리즘들의 성능을 비교한다.

검증(Validation)은 k-Fold Cross Validation (Stone, 1974)라는 방법을 사용하였다. 이는 주로 데이터셋이 작은 연구에서 기계학습 시스템을 학습시키고 검증할 때 사용하는 방법이다. 검증의 단계는 다음과 같다. k-Fold란 먼저, N개의 데이터를(본 연구에서는 실무 공종명과 정식 공종명의 쌍) 크기가 같은 k개의 부분집합으로 나눈다. 이 중 하나의 부분집합은 검증을 위해 사용하고, 나머지 집합들은 학습을 위해 사용한다. 예를 들어 1000개의 데이터가 있고 k는 5이라고 가정한다면 200개의 검증 데이터와 800개의 학습데

이터로 나누는 것을 의미한다. 다음은 가능한 모든 순서쌍을 통해 교차검증을 하는 것이다. 위의 예에서는 총 3번의 교차검증이 이루어진다. 기계학습에서 본 기법은 통상 데이터셋의 최종 테스트(Testing)가 아니라 하이퍼파라미터등을 최적화하기 위한 모델의 검증(Validation)을 위한 것이다. 따라서 최종적인 시스템의 일반화 능력을 판단하는 척도로 사용하지 않는다.

본 연구의 경우는 이 기법을 응용하여 알고리즘들의 성능을 도출하였다. k-Fold가 최종 테스트에 사용되지 않는 이유는 학습시스템에 어떠한 이유로든 노출된 데이터는 사용하지 않는 것이 원칙이기 때문이다. 하지만 우리의 연구의 경우에는 존재하는 모든 공종의 정식 명칭이 정해져 있지 않고, 명확한 정답도 존재하지 않는다. 또한, 우리는 문자열 유사도 알고리즘의 일반화능력을 평가하고자 하는 것이 아니다. 따라서 이러한 선택을 하였다. K로는 5와 10을 사용했고, 각각의 실험에서 테스트하는 공종명과 데이터베이스의 공종명을 교차시켜 총 4개의 실험을 하였다. 두 가지를 교차시킨다는 것은, 하나의 실험에서는 테스트하는 공종명이 k개의 부분집합중 하나이고, 다른 실험에서는 테스트하는 공종명이 하나를 제외한 모든 부분집합들이 된다는 것을 의미한다. 이에 따라 서로 다른 데이터베이스의 크기에 따른 정확도의 변동을 관찰할 수 있었다. 마지막으로, 같은 조건의 실험은 각각 10번 반복하여 평균값을 취했다.

4. 결과

4.1 알고리즘의 정확도

결론적으로는 다섯가지의 알고리즘 중 Jaro-Winkler 알고리즘이 가장 좋은 성능을 보여주었다. 또한, N-Gram (N=3) 알고리즘이 가장 저조한 성능을 보였다. Hamming distance와 N-Gram (N=2)을 사용한 알고리즘은 비슷한 정확도를 보여주었고, Levenshtein 알고리즘은 중간 정도의 정확도를 보여주었다. 다음 〈Fig. 3〉에는 알고리즘별 각 4개의 실험에서 얻을 수 있었던 평균 정확도를 정리하였다. 4가지의 서로 다른 실험은 k-Fold Cross Validation에서의 k값이 5일때와 10일 때, 그리고 데이터베이스로 주어지는 데이터의 양이 클 때와 작을 때 (K가 5인 경우 데이터베이스의 크기가 크다면 총 데이터셋의 80%, 반대의 경우 20%)를 조합한 경우의 수이다. 표의 X축은 알고리즘의 약자이며 이는 〈Table 4〉에 정리되어 있다. 표의 Y축은 0과 100 사이의 정확도이다. 〈Table 5〉는 구체적인 수치이다.

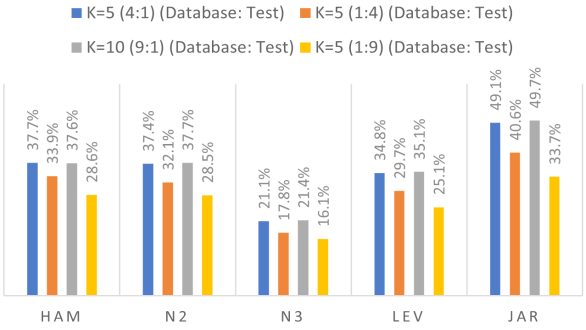

Fig. 5. Similarity algorithm comparison

Table 5. Similarity algorithm accuracy comparisons

Algorithm	K=5 (4:1) (Database: Test)	K=5 (1:4) (Database: Test)	K=10 (9:1) (Database: Test)	K=5 (1:9) (Database: Test)
HAM	37.7%	33.9%	37.6%	28.6%
N2	37.4%	32.1%	37.7%	28.5%
N3	21.1%	17.8%	21.4%	16.1%
LEV	34.8%	29.7%	35.1%	25.1%
JAR	49.1%	40.6%	49.7%	33.7%

다음으로 서로 다른 데이터베이스와 테스트 공종명의 크기에 따른 정확도를 비교한다. k가 5일 경우에서 데이터베이스가 더 크고, 테스트 공종명의 수가 상대적으로 작은 경우 (파랑)과 그 반대의 경우 (주황)을 비교하면, 전자가 더욱 뛰어나다. k가 10일 경우에도 (회색, 노랑) 데이터베이스가 더욱 클 때 정확도가 더 높다. 마지막으로, k가 10이고 데이터베이스가 작을 때 성능이 가장 저조한데 이는 실무에서 사용되는 문서에서 추출한 새로운 정보를 비교해 볼 데이터베이스가 더욱 클 때 유리할 수 있다는 것을 의미하며, 더욱 풍부한 공종명 데이터베이스를 구축하는 것이 중요하다는 것을 시사한다.

4.2 알고리즘의 정밀도(Precision)와 재현율(Recall)

탐구한 모든 알고리즘은 거리값을 출력한다. 만약 출력한 거리값이 0이라면 전혀 다른 문자열이라는 것을 의미하고, 1이라면 완전히 같은 문자열이라는 것을 의미한다. 〈Table 6〉에 각 알고리즘이 모든 실험에서 데이터셋에 존재하는 문자열들에 부여한 거리값의 평균값들을 정리하였다.

Table 6. Average thresholds for different algorithms

	HAM	N2	N3	LEV	JAR
Avg. Thresh.	0.68	0.32	0.14	0.66	0.81

Jaro-Winkler 알고리즘이 가장 너그러운 점수를 주었고, N-Gram (N=3) 알고리즘이 가장 박한 점수를 주었다. 본 연구에서의 거릿값은 알고리즘의 신뢰도 또는 자신감 (Confidence)이라고도 볼 수 있는데, 우리는 이 값을 가지

고 각 알고리즘의 정밀도와 재현율을 탐구해 보았다. 정밀도란, 얻은 결과물 중 해결하고자 하는 문제와 관련이 있는 결과의 비율을 의미하고, 재현율이란 모든 관련 있는 결과물 중 성공적으로 얻을 수 있었던 것들의 비율을 의미한다. 정밀도와 재현율을 탐구하는 이유는 추후 자동 문서작성 등의 작업에 알고리즘에 의해 인식된 단어를 사용하기 위해서는 결과물의 신뢰도를 따져 신뢰성이 낮게 판단한 문자열은 배제할 필요가 있기 때문이다. 이를 위해, 출력한 문자열 중 알고리즘이 자신이 옳다고 판단한 것과 틀린다고 판단한 것을 구분 짓기 위해 임계점을 먼저 0.7로 임의로 설정하였다. 만약 거리값이 0.7이면 이는 알고리즘이 자신의 대답이 틀렸다고 말하는 것과 같다.

다음 〈Table 7〉은 k가 5이고, 데이터베이스와 테스트 데이터의 비율이 1:9일 때의 정밀도와 재현율이다. 이 연구에서 정밀도는 임계치가 0.7일 때, 알고리즘이 일치하였다고 판단하는 문자열 중 실제로 맞는 것의 비율이다. 반면에 재현율은 임계치가 0.7일 때 모든 옳은 문자열 중 일치한다고 제대로 찾아낸 것의 비율이다. 둘 다 높은 알고리즘이 가장 유리한데, Jaro-Winkler 알고리즘이 총체적으로 가장 높았다. 다시 이야기하면, 본 알고리즘이 자신 있게 인식하였다고 생각하는 문자열 중 실제로 옳게 인식한 비율이 높고, 임계점 이상에서 더욱 많은 문자열을 옳게 인식해 낼 수 있었다는 것을 의미한다.

Table 7. Precision and recall when K=5 and (DB:Test = 1:9)

	Precision	Recall
HAM	0.413	0.327
N2	0.555	0.042
N3	0.709	0.025
LEV	0.374	0.241
JAR	0.476	0.710

공종명 인식을 통해 문서작성 등을 자동으로 하기 위해서는 최대한 자동화를 많이 시키는 것도 중요하지만, 하나를 찾아내도 올바로 찾아내는 것이 더욱 중요시되어야 한다. 왜냐하면, 자동 시스템이 만약 실수해서 옳지 않은 방향으로 문서를 작성하였으면 그 실수를 찾는데 더욱 큰 노력이 들어갈 수 있기 때문이다. 이러한 측면에서 보면 N-Gram (N=3)이 가장 유력해 보인다. 하지만 N-Gram (N=3)은 재현율이 낮아 실제로 자동으로 인식할 수 있는 공종명이 얼마 되지 않는 데에 그 단점이 있다.

만약 각 응용 프로젝트에서 문자열 유사도 알고리즘을 사용하여 공종명을 인식하려고 한다면, 알맞은 임계점을 찾아내야 한다. Receiver Operation Characteristics (ROC) curve를 사용해서 각 임계점에서 사용하고자 하는 알고리즘의 성능을 평가한 뒤, 가장 해결하고자 하는 문제에 알맞은 임계점을 사용하면 될 것이다.

5. 연구결과의 고찰

5.1 공종명의 언어 구조적 특징

연구를 진행함에 따라 우리는 상당수의 공종명이 공유하는 언어 구조적 특징을 배울 수 있었다. 모든 공종명은 고유명사로 취급할 수 있다. 하지만 이를 또 자세히 나눠보면 다음 〈Table 8〉과 같다.

Table 8. Task name generic composition

Proper Noun	Common Noun	Proverb	Ending
Chinhwangyeongpaint	-	Chil	-Gi
-	Paint	Chil	-Gi
-	Mortar	Bareuda	-Gi
Vinyltile	Cheolgeo	-	-

건설 산업에만 존재하는 매우 특수한 자재나 공정 등은 대부분 고유명사에 속한다. 예로 “H형강”이라던지 “비닐타일”등을 들 수 있다. 다음으로는 일반명사인데, 일반명사는 설치나 철거와 같이 동작성 명사가 대부분이다. 하지만 동작성 명사 또는 용언(동사, 형용사)과 어미(명사형 전성어미)의 조합일 때도 존재한다. 예를 들어 “페인트 칠하기”는 명사인 “칠”과 명사성 전성어미 “-기”가 붙은 형태이다. 또한 “모르타르 바르기”에서는 “바르다”라는 동사와 “-기”라는 명사형 전성어미가 혼합된 형태이다. 품사의 분석은 파이썬 패키지 Konlpy (Eunjeong L. Park, 2014)에 포함된 한나눔 형태소 분석기(KAIST CILab, 2014)를 사용해 분석한 데이터를 참조하였다.

언어 구조적 특성이 뚜렷하게 존재한다는 것은 표준화에 유리하다. Resource description framework (RDF) (Cambria and White, 2014)는 고정적인 구조를 정의하고, 데이터를 그 구조에 맞춰 설명하는 방식인데, 공종명의 구조적 특색 덕분에 구조적인 특성에 집중하면 빛을 발할 가능성이 상대적으로 높아질 것이다. 이러한 시스템의 일례로 (McGuinness and Van, 2004)를 들 수 있는데, Subject-Predicate-Object 모델과 같이 어떠한 분야에서 사용되는 언어를 고정적인 구조에 맞춰 분석해야 하는 경우에 유리해 보인다.

더욱 나아가서 구조적 특징은 추후 문자열 유사도 알고리즘보다 진화된 기계학습 등을 통한 공종명 비교연구에서 큰 도움이 될 수 있다. 하지만 문제점도 존재하는데, 다양한 형

태소 분석기들의 분석이 원문의 띄어쓰기에 상당한 부분 의존한다는 점에서 본 연구에서 맞닥뜨린 띄어쓰기에 문제가 있는 공종명 들을 식별하는 데 어려움이 있을 것으로 보인다.

5.2 성능 평가

각기 다른 알고리즘의 결과를 살펴보면 Jaro-winkler 알고리즘의 결과가 가장 좋다. 이는 아마도 문자열의 방향성을 고려하는 알고리즘의 특색 때문으로 추정하는데, 공종명에서 자주 발견되는 언어적 특징이(수식어 + 명사 + 명사형 동사) 위 알고리즘에서 좋은 결과를 이끌어내는 것으로 보인다. 반면에 N-Gram (N=3)이 가장 좋지 못한 결과를 보이는데, 이는 공종명에 사용되는 문자들의 최소단위가 대부분 낱말 세 개 이하이기 때문이라고 해석할 수 있다.

본 논문에서 사용한 알고리즘은 의미론적 거리를 고려하지 못한다. 이에는 장단점이 존재하는데, 장점으로는 알고리즘의 구현이 간단하고, 학습을 위한 데이터셋이 필요하지 않고, 문자열의 처리 속도가 빠르다는 것이다. 반면에 단점으로는 성능의 수준이 보유하고 있는 데이터베이스 (〈Fig. 1〉의 C)의 풍부함에 크게 좌우된다는 것이다. 만약에 세상에 존재하는 모든 공정과 그 파생형이 데이터베이스에 존재한다면 문자열 유사도 알고리즘이 그 빛을 발할 수 있을 것이다. 하지만 이러한 이상적인 데이터베이스를 사용할 때는 단점도 존재한다. 바로 알고리즘이 연산 시간이 데이터베이스에 존재하는 데이터 개체의 수에 정비례한다는 것이다. 인덱싱이나 멀티 프로세싱 같은 추가적인 성능향상 기법을 사용하지 않는다면 풍부한 데이터베이스는 느린 처리 속도를 초래할 것이다. 그러나 이러한 문제를 보완할 수 있는 빅데이터 처리를 위한 시스템의 발전이 병행되고 있어 이러한 방식으로의 발전 가능성은 고무적이다.

6. 결론

6.1 부족한 데이터와 양식의 불규칙성

문자열 유사도 알고리즘의 성능은 데이터베이스의 질에 좌지우지된다고 하였다. 또한, 기계학습을 하려고 해도 잘 정리된 커다란 데이터셋이 필요하므로 공종명 데이터베이스를 구축하는 일은 매우 중요하다.

공종명은 너무나 다양할 수 있다. 또한, 전문가 이외의 사람은 공종명 비교에 난관을 겪을 수 있다. 따라서 공종명 데이터베이스를 적산 업무 이외에서 개별적으로 단기간에 구축하는 것은 불가능해 보인다. 반면에 적산 전문가들이 실제 작업하는 환경에 데이터베이스를 구축하는데 일조할 수 있는 모듈을 통합하는 방법을 생각할 수 있다. 전문가의 작업 환경과 데이터베이스가 연동되어 있다면, 전문가는 데이터베이스에 존재하는 알맞은 공종명, 규격, 비용을 빠르게 수집해 올 수 있다. 반면에 데이터베이스는 전문가가 지속해서 추가하는 공종 데이터를 통해 그 풍부함을 불릴 수 있다.

데이터 수집에 관련해서는, 우리는 공종 데이터를 추출하기 위해 수만 개의 내역서 엑셀 파일을 수집하였다. 이 과정에서 불규칙한 양식 때문에 데이터 추출의 질이 떨어지게 되었는데, 이는 향후 빅데이터 시장에서 큰 발목을 잡을 수 있는 요소이다. 예컨대, 수집한 입찰 내역서 중 만약 조명 관련 사업의 양식은 중구난방이고, 토목 관련 사업의 양식들은 비교적 체계적인 경우, 양식에서 텍스트를 불러오는 전처리 프로그램에서 미처 고려하지 못한 대부분 양식이 조명 관련 내역서에 존재할 가능성이 커지고, 추후 연구나 프로그램 개발에서 등한시될 수 있을 것이다. 하지만 만약 산업에서 일괄적이고 체계적인 건설 프로젝트 품목 내역서의 양식을 엄격하게 따른다면 미래를 위해 균형 잡히고 더욱 질 높은 데이터를 생산할 수 있을 것이다.

6.2 사업 견적비 작성 자동화에의 응용 가능성

문자열 유사도 알고리즘을 통해 물량 내역서를 바탕으로 사업비 견적을 산출하는 작업을 자동화할 수 있음을 서론에서 밝힌 바 있다. 여러 집단의 이해관계가 얽힌 문서의 작성은 정확성과 투명성이 중요한데, 문자열 유사도를 이용한 방법처럼 자동화된 시스템을 사용하면 문서 작성 도중의 판단 경위를 모두 기록할 수 있어서 투명성을 높이는 데 도움이 될 수 있다. 반면에, 정확도는 크게 높지 않기 때문에 문자열 유사도 알고리즘을 통한 완전 자동화는 힘들어 보인다. 하지만 이는 사업비 견적 자동 산출 프로그램이 추구할 수 있는 방향을 제시하는데, 바로 전문가가 작업을 더욱 손쉽게 처리할 수 있는 추천 시스템의 개발이다. 추천 시스템은 어떠한 일 처리의 방향이나 방법을 제시해 주는 시스템으로, 전문가의 생산성을 증대시킬 수 있고, 작업의 질도 향상할 수 있다. 추천 시스템을 사용한다면 문자열 유사도 알고리즘이 생각하는 유사한 공종 몇 가지가 전문가에게 제시될 것이고, 그는 추천된 공종명을 바탕으로 쉽게 문서를 작성할 수 있을 것으로 예상된다.

6.3 사례 연구

사업 견적비 작성 자동화에서 공종명 인식이라 하면 이것이 과연 어떠한 작업인지 혼란이 올 수 있다. 따라서 실제 건설문서에서 추출한 몇 가지 내역서 항목을 사용해 간단하게나마 시연한다. 먼저 물량 내역서는 〈Table 9〉과 같이 생겼다. 자동 물량 산출이란 품명, 규격을 사용해 공종과 그 공종의 파생형 또는 변종을 인식하고, 인식된 공종을 데이터베이

Table 9. Empty quantities

Name	Specs	Measurement	Quantity	Material Cost	Labor Cost	Expenses	Total
Tileshihum 〈타일시험〉	Negyunyeolsung(shiyoutile) 〈내균열성(시유타일)〉	times 〈회〉	1				
Terrajotilecheolgeo(Badak) 〈테라죠타일철거(바닥)〉	Sohyungbreaker(Badakmortarpoham) 〈소형브레이커(바탕몰탈포함)〉	㎡	116				
Songyitile 〈송이타일〉	190*57*16	㎡	210				
Tileapchakbuchim(Batang18mm+yip5mm) 〈타일압착붙임(바탕18mm+압5mm)〉	Byuk, Tile 0.04~0.10㎡ yihaa 〈벽, 타일 0.04~0.10㎡ 이하〉	㎡	208.9				
Badaktile(200*200) 〈바닥타일(200*200)〉	200×200×7.0mm	㎡	49.34				

Table 10. Jaro-Winkler output

Name	Specs	Measurement	Quantity	Inferred	Confidence
Tileshihum 〈타일시험〉	Negyunyeolsung(shiyoutile) 〈내균열성(시유타일)〉	times 〈회〉	1	Tile Boyang 〈타일 보양〉	0.71
Terrajotilecheolgeo(Badak) 〈테라죠타일철거 (바닥)〉	Sohyungbreaker(Badakmortarpoham) 〈소형브레이커(바탕몰탈포함)〉	㎡	116	Yerrajotilecheolgeo 〈테라죠타일 철거〉	0.91
Songyitile 〈송이타일〉	190*57*16	㎡	210	Mozaictilebuchigi 〈모자이크타일 붙이기〉	0.68
Tileapchakbuchim(Batang18mm+yip5mm) 〈타일압착붙임(바탕18mm+압5mm)〉	Byuk, Tile 0.04~0.10㎡ yihaa 〈벽, 타일 0.04~0.10㎡ 이하〉	㎡	208.9	Tileapchakbuchigi 〈타일 압착붙이기〉	0.7
Badaktile(200*200) 〈바닥타일(200*200)〉	200×200×7.0mm	㎡	49.34	Badaktilebuchigi 〈바닥타일 붙이기〉	0.76

스에서 검색하여 단위와 수량을 사용해 재료비, 노무비, 경비를 계산하는 것이다. 본 연구는 규격은 배제하고 공종명만 고려하여 공종 인식을 시도한다. 규격까지 고려할 수 있어야 더욱 정확한 프로그램을 만들 수 있다.

다음은 〈Table 10〉은 Jaro-winkler 알고리즘의 출력물이다. 네 개의 실험 설정 중 k=5이고 데이터베이스와 테스트 셋의 비율이 1대 4로, 검색할 데이터베이스의 크기가 충분히 작을 때의 결과 중 일부를 추출하였다. 실전에서 사용할 때의 데이터베이스는 단위당 가격을 포함하고 있어야 하며, 이를 사용하여 〈Table 9〉의 비용을 자동 산출할 수 있다. 4.2에서

4.2에서 임계점에 대해 논한 바 있다. 만약 〈Table 10〉의 출력값에서 임계점을 0.70 초과로 설정한다면, 첫번째 개체는 Type 2 오류 (잘못된 것을 옳다고 판단), 2, 5번째와 3번째는 각각 옳은 것을 옳게 판단한 경우와 틀린 것을 틀리게 판단한 경우, 마지막으로 4번째 개체는 옳을 것을 틀리다고 판단하는 Type 1 오류에 속할 것이다. 이 예에서 만약 임계점을 "0.68 이상"으로 설정한다면 옳을 것을 옳게 판단하는 비율이 높아지겠지만, 반면에 Type 2 오류의 개수가 2개로 증가한다. 따라서 실전에서는 임계점을 연구하고 설정하는 것이 매우 중요하다고 할 수 있다.

6.4 향후의 연구 방향

본 연구에서는 건설업계에서 사용되는 공종명이 대체로 따르는 구조적 특징이 존재한다는 것을 관찰하였으며 알고리즘별 분석결과를 통해 정밀도와 재현율의 차이를 확인하였다. 그러나 문자열 유사성 알고리즘만으로는 공종명을 정확하게 인식하는 것에는 한계가 있다. 마지막으로는 현재 건설업에서 자동화와 인공지능 연구가 활발하게 이루어지는데 필요한 데이터의 질과 양이 현저히 떨어지고 부족하다는 문제점을 지적하였다.

추후 연구에서는 위 문제점을 해결하기 위한 음성적 유사도 등의 자연어 처리 분석 방법론을 병행하는 경우의 분석결과와 방대한 데이터베이스를 처리할 수 있는 효율적인 시스템 구축에 관한 연구를 통해 데이터베이스의 볼륨을 증가시켜야 한다. 또한, 데이터의 증가에 따라 기계학습을 사용해 도출한 공종명 간의 의미론적 거리를 공종명 식별에 사용할 수도 있다.

References

Abiteboul, S. (1997). Querying semi-structured data. InInternational Conference on Database Theory, Springer, Berlin, Heidelberg, pp. 1-18.

An, M.G., Kim, M.J., and Kim, Y.S. (2019). "A Blockchain-Based Risk Management for Overseas Construction Project Using Natural Language Processing." Proceedings of KICEM University Students Conference, pp. 116-119.

Bilal, M., Oyedele, L.O., Qadir, J., Munir, K., Ajayi, S.O., Akinade, O.O., and Pasha, M. (2016). Big Data in the construction industry: A review of present status, opportunities and future trends. Advanced engineering informatics, 30(3), pp. 500-521.

Cambria, E., and White, B. (2014). Jumping NLP curves: A review of natural language processing research. IEEE Computational intelligence magazine, 9(2), pp. 48-57.

Han, K.K., and Golparvar-Fard, M. (2017). Poten-tial of big visual data and building informationmodeling for construction performance analyt-ics: An exploratory study.Automation in Con-struction, 73, pp. 184 – 198.

Kim, Y.R., Lee, S.H., and Park, S.H. (2012). "Development of Rule-Set Definition for Architectural Design Code Checking based on BIM." *Korean Journal of Construction Engineering and Management*, KICEM, 13(6) pp. 143-152.

Lee, J.H. (2019). "Review on Natural Language Processing Reasearch Utilizing Unstructured Text Data in Construction Industry." *Construction Engineering and Management*, 20(2), pp. 62-66.

Lee, S.K., Kim, K.R., and Yu, J.H. (2012). Automatic Inference of Standard BOQ (Bill of Quantities) Items using BIM and Ontology. *Korean Journal of Construction Engineering and Management*, 13. 10.6106/KJCEM.2012.13.3.099.

Lalwani, M., Bagmar, N., and Parikh, S. (2014). "Efficient Algorithm for Auto Correction Using ngram Indexing." International Journal of Computer & Communication Technology (IJCCT), 3(3), pp. 23-27.

McGuinness, D. L. (2004). OWL web ontology language overview. W3C recommendation. http://www. w3. org/TR/owl-features.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), 3-26.

Park, Y.S., Oh, C.D., Jeon, Y.S., and Park, C.S. (2008). "A Web-Based Construction Failure Information System using Case-Based Reasoning." *Korean Journal of Construction Engineering and Management*, KICEM, 9(6), pp. 257-267.

Stone, M. (1974). Cross - validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, Series B (Methodological), 36(2), pp. 111-133.

Williams, T.P., and Gong, J. (2014). "Predicting construction cost overrunsusing text mining, numerical data and ensemble classifiers." Automation in Construction, 43, pp. 23 – 29.

Zhang, J., and El-Gohary, N.M. (2016). "Semantic nlp-based information extrac-tion from construction regulatory documents for automated compliancechecking." *Journal of Computing in Civil Engineering*, 30(2), 04015014.

Y.S. Kim. (2019). Automatic multi-label image classification model for construction site images (Unpublished doctoral dissertation). Graduate School, Seoul National University.

Eunjeong L. Park, Sungzoon Cho. (2014). KoNLPy: Korean natural language processing in Python. 26th Annual Conference on Human and Language Technology, pp. 1-4.

KAIST CILab (2014). Hannanum Morphological Analysis http://semanticweb.kaist.ac.kr/hannanum/index.html

요약 : 시공 서류에서 접하는 자연어는 당국에서 권장하는 언어와 크게 다르다. 일관성이 부족한 이러한 관행은 자동화를 통한 통합 연구를 방해하고 장기적으로 업계의 생산성을 저하시킬 것이다. 이 연구는 여러 문자열 유사성(문자열 일치) 알고리즘을 비교하여 여러 다른 방법으로 작성된 동일한 작업 이름을 인식하는 각 알고리즘의 성능을 비교하는 것을 목표로 한다. 우리는 또한 앞서 언급 한 편차가 얼마나 널리 퍼져 있는지에 대한 토론을 시작하는 것을 목표로 한다. 마지막으로, 우리는 실제로 발견된 시공 작업 이름을 형식에 비해 덜 복잡한 해당 작업 이름과 연결하는 작은 데이터 세트를 구성했다. 이 데이터 세트를 사용하여 미래의 자연어 처리 접근 방식을 검증 할 수 있을 것으로 기대한다.

키워드 : 자연어 처리, 문자열 유사도, 문자열 매칭, 문자열 근사값