한글 문서의 단어 동시 출현 정보에 개선된 TextRank를 적용한 키워드 자동 추출 기법

송광호○, 민지홍, 김유성
인하대학교, 정보통신공학과
crossofjc@gmail.com, newindia89@gmail.com, yskim@inha.ac.kr

Keyword Automatic Extraction Scheme with Enhanced TextRank using Word Co-Occurrence in Korean Document

KwangHo Song○, Ji-Hong Min, Yoo-Sung Kim
Department of Information and Communication Engineering, Inha University

요 약

문서의 의미 기반 처리를 위해서 문서의 내용을 대표하는 키워드를 추출하는 것은 정확성과 효율성 측면에서 매우 중요한 과정이다. 그러나 단일문서로부터 키워드를 추출해 내는 기존의 연구들은 정확도가 낮거나 한정된 분야에 대해서만 검증을 수행하여 결과를 신뢰하기 어려운 문제가 있었다. 따라서 본 연구에서는 정확하면서도 다양한 분야의 텍스트에 적용 가능한 키워드 추출 방법을 제시하고자 단어의 동시출현 정보와 그래프 모델을 바탕으로 TextRank 알고리즘을 변형한 새로운 형태의 알고리즘을 동시에 적용하는 키워드 추출 기법을 제안하였다. 제안한 기법을 활용하여 성능평가를 진행한 결과 기존의 연구들보다 향상된 정확도를 얻을 수 있음을 확인하였다.

주제어: 자동 키워드 추출, 단어 동시출현 정보, TextRank 알고리즘, 그래프 모델, 한글문서

1. 서론

정보검색, 문서 군집화, 문서 분류 등 다양한 텍스트 처리 분야의 연구들은 공통적으로 각 텍스트가 기술하고 있는 의미를 잘 표현하는 문단이나 문장 또는 단어를 선별하는 단계를 포함한다. 이렇게 선별된 속성들은 이후의 처리 단계에서 중요한 영향을 미치기 때문에 이를 정확하고 효율적으로 선별해내는 것은 매우 중요하다. 특히 단어(Word)는 텍스트 마이닝 연구에서 가장 널리 쓰이는 주요 속성으로써 주제를 대표할 수 있는 적은 수의 대표 단어(Representative Term)를 주어진 텍스트로부터 자동추출해내는 키워드 추출(Keyword Extraction)[1] 연구들이 활발히 진행되고 있다[2][3][4].

키워드 추출은 대체로 여러 문서들의 집합에서 나온 대량의 단어들 중에서 각 문서들을 구별하는데 핵심적 역할을 하는 단어를 추려내는 방법으로 이루어진다. 그러나 [5]에 따르면 문서 집합에서 추출된 단어들 중 실제 각각의 문서를 구별하는 핵심적인 역할을 할 수 있는 단어 즉, 각 문서가 가진 고유성을 잘 기술할 수 있는 단어들은 약 10% 정도로 매우 적은 수이기 때문에 각각의 문서들이 가진 의미를 잘 표현할 수 있는 단어들을 각 문서마다 걸러내는데에 많은 어려움이 따른다. 따라서 최근에는 문서 군집이 아닌 단일 문서를 대상으로 그 문서만의 키워드를 추출해 내는 연구들이 활발히 진행되고 있다. 특히 최근 국내에서도 단일문서의 중요 속성(Keyword 또는 Keyphrase)을 추출하는데 있어서 TF-IDF 또는 그 변형식을 사용하는 단순한 빈도기반 접근법[2]에서 벗어나 단어의 동시출현정보를 활용[3]하거나 TextRank 알고리즘 등의 다양한 알고리즘을 적용[4]하는 등의 시도들이 이루어지고 있다.

그러나 [2]와 [3]의 연구들은 실제 활용하기에 부족한 수준의 정확도를 보였으며, [4]의 연구는 [2,3]보다 나은 성능을 보이긴 하였으나 제한된 영역의 데이터에 대한 검증결과만을 제시하였다. 따라서 본 논문에서는 단어의 동시출현 정보와 TextRank 알고리즘을 변형한 새로운 형태의 알고리즘을 함께 적용하여 비교적 정확도가 높은 방법을 제안하고 다양한 주제를 가진 텍스트 데이터를 이용한 평가를 진행하여 제안한 방법이 다양한 분야의 텍스트에서도 균일한 성능을 보이는지에 대한 검증결과를 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 [2-4] 연구들과 본 연구에서 활용할 단어의 동시출현정보와 TextRank알고리즘에 대해 간단히 알아본다. 3장에서는 단어 간의 동시출현 정보와 TextRank 알고리즘[6]을 변형한 새로운 형태의 알고리즘을 함께 적용한 키워드 추출 방법에 대해 제안하고, 4장에서는 그 방법을 실제 데이터에 적용하여 성능을 평가하는 실험을 제시한다. 마지막으로 본 논문의 결론과 향후연구에 대해서는 5장에서 기술한다.

2. 관련 연구

문서에서 키워드를 추출해내는 방법에는 기본적으로 문서에 나타난 단어들의 빈도를 이용하는 빈도기반 접근법과 그에 더해 문서들이 가진 언어학적 요소들을 가미

하는 특징기반 접근법 그리고, 최근 연구되기 시작한 기계학습기반 접근법 등이 있다[7]. 그 중 빈도기반 접근법은 1958년 Luhn[8]이 처음 제안한 후 지금까지도 가장 많이 연구되는 방법이다.

빈도기반 접근법을 활용한 연구는 최근까지도 벡터공간모델을 바탕으로 TF-IDF 기반 공식들을 적용하는 방법이 주로 연구되었다. 그러나 이 방법은 키워드 추출에 있어서는 정확도가 낮은 문제점을 갖고 있었고 그에따라 [2]에서는 TF-IDF 뿐 아니라 다양한 변형공식들을 적용하고 그에 더해 공식 적용 전 단계에 용어클러스터링을 도입하여 그 성능을 더 향상시키려 하였다. 그 결과 [2]에서 제안된 방법으로 추출한 키워드와 제목에 출현한 단어를 비교했을 때 정확도가 최대 약 52%로 나타났다.

이렇게 기존 연구들이 정확도 측면에서 큰 개선을 이루지 못하면서 최근의 빈도기반 접근법의 연구는 단어들의 동시출현 정보와 같은 추가적인 정보를 활용하거나 이를 바탕으로 다른 공간 모델에 적용하는 등의 새로운 방법들을 시도하고 있다[3,4]. 먼저 [3]에서는 텍스트를 복합 어절 단위로 분할 한 후 분할된 영역에서 동시에 출현한 단어의 빈도를 계산하여 동시출현 행렬을 구성한 후 행렬의 값을 이용해 각 단어마다 가중치를 계산하여 문서의 주제어를 추출하는 방식을 제안하였다. 그 결과 추출된 키워드와 저자가 선정한 키워드 단어를 비교했을 때 정확도가 최대 36%로 나타났다.

[4]에서는 기존의 연구들이 사용한 벡터공간모델 대신 동시출현 관계에 있는 단어들을 이용한 그래프모델을 적용하였다. 텍스트를 문장으로 구분하고 문장에 출현한 단어들을 Vertex로 표현하고, 함께 출현한 단어들을 Edge로 연결시킨 연결그래프를 만든 후 이를 TextRank 알고리즘[6]으로 훈련하여 주요단어들로 이루어진 중요 문장을 추출하는 연구를 진행하였다. TextRank 알고리즘[6]이란 PageRank 알고리즘을 텍스트에 맞게 변형,적용한 알고리즘으로 중요한 단어는 다른 다양한 단어들과 함께 나온다는 점을 이용하여 단어 그래프를 구성하고 그래프의 Vertex V_i의 중요도(Score) $S(V_i)$를 [식 1]을 이용해 계산한다. 여기서 $In(V_i)$은 V_i로 들어오는 Edge의 집합을 의미하고 $Out_Degree(V_j)$는 V_j에서 나가는 Edge의 개수를 의미하며 d는 0과 1사이의 임의의 실수을 사용한다.

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{S(V_j)}{Out\_Degree(V_j)} \ , (0 \le d \le 1) \qquad \text{(식 1)}$$

[4]에서도 [식 1]을 이용하여 단어의 중요도를 계산하였다. 다만 [4]는 중요 문장을 추출하는 것이 최종 목적이므로 [식 2]와 같이 두 V_i와V_j가 속한 문장들 사이의 코사인 유사도 $Sim(V_i, V_j)$의 비율을 $Out_Degree(V_i)$를 대신하여 적용하는 방식으로 식을 변경했다.

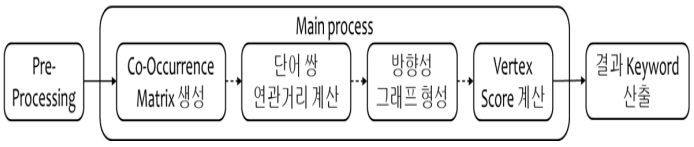
$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{Sim(V_i,V_j) * S(V_j)}{\sum_{k=0}^{N-1} Sim(V_i,V_j)} \ , (0 \le d \le 1) \qquad \text{(식 2)}$$

그 결과 연구진에서 자체적으로 준비한 정답 문장과 추출된 키워드로 이루어진 중요 문장의 정확도를 비교했을 때 약 65%의 정확도를 얻었다. 이는 알고리즘이 얻고자 하는 목적에 따른 [식 1]의 변형이 유효한 효과를 보일 수 있다는 것을 보여준다. 그러나 이 연구의 연구자들이 스스로 지적한 바와 같이 이 실험결과는 '정치,경제' 라는 특정 분야로 제한된 텍스트 데이터를 이용하여 이루어진 실험을 통해 나온 결과이므로 다양한 분야의 데이터에 적용한 결과에 대한 검토가 부족하다.

따라서 본 논문에서는 정확도 높은 키워드 추출을 위해 단어의 동시출현 정보와 TextRank 알고리즘을 변형한 알고리즘을 동시에 적용한 새로운 키워드 추출 기법을 제안한다. 또한 이를 다양한 분야의 데이터에 적용하는 실험을 통해 일반적 데이터에 대한 적용가능성을 살피고 기존 연구들과 동일한 평가방법을 사용하여 성능비교실험을 진행, 그 성능과 타당성을 제시한다. 다만 [4]의 연구의 경우는 대조군을 저자들이 직접 정의하여 실험의 재현성이 없고 본 논문의 주제인 키워드 추출과는 [4]의 연구목적이 주요문장추출로서 상이하므로 성능을 직접적으로 비교하는 실험은 진행하지 않는다.

3. 단어의 동시출현 정보에 개선된 TextRank를 적용한 키워드 자동 추출

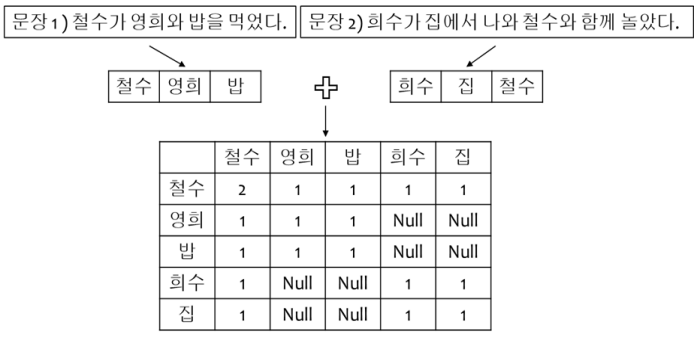
앞서 언급한 바와 같이 본 논문에서는 정확도 높은 키워드 추출을 위해 단어의 동시출현정보와 TextRank 알고리즘을 변형한 알고리즘을 동시에 적용한 새로운 키워드 추출 기법을 제안하고자 한다.



[그림 1] 키워드 추출 프로세스 개요도

이 기법은 [그림 1]과 같이 총 3단계의 과정으로 이루어진다. 첫 번째는 전처리(Pre-Processing) 단계로 문서를 문장단위로 분리하고 그 문서에서 출현 문장 수가 2개를 이상인 명사들을 추출하여 그 단어와 출현 문장 및 문장길이, 문장 내 위치 등을 저장한다. 이때 그 단어가 복합(Compound)명사일 경우 복합명사를 구성하는 구성명사도 함께 출현한 것으로 처리한다.

두 번째 단계인 Main Process는 다시 4가지의 세부단계로 나뉜다.



	철수	영희	밥	희수	집
철수	2	1	1	1	1
영희	1	1	1	Null	Null
밥	1	1	1	Null	Null
희수	1	Null	Null	1	1
집	1	Null	Null	1	1

[그림 2] 동시출현 단어 행렬 생성 예

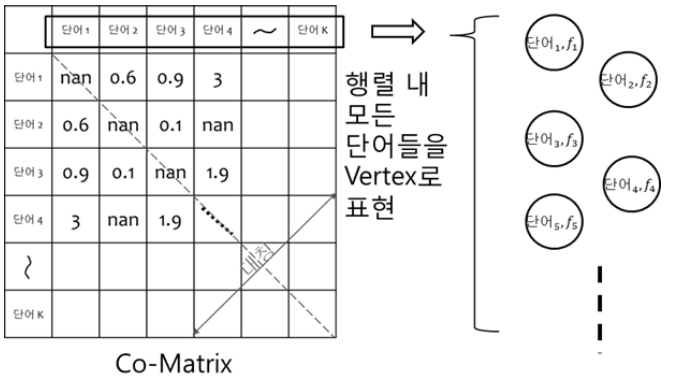
먼저 [그림 2]의 예시와 같이 각 문장에 함께 출현한 단어쌍들의 출현횟수를 저장할 동시출현 단어 행렬(Co-Occurrence Matrix, CO-Matrix)을 생성한다. 이 행

렬의 대각성분 즉, 동일한 단어가 교차하는 성분은 문서 전체에서 그 단어가 출현한 문장의 개수이고 서로 다른 두 단어가 교차하는 성분은 그 두 단어가 함께 나온 문장의 개수이다. 행렬을 생성한 후에는 다음 단계인 그래프 구성 시 사용하기 위해 각 동시출현 단어쌍마다 동시출현 수(S_{ij}^1)와 독립 출현 수(S_{ij}^2, S_{ij}^3) 그리고 동시 미출현 수(S_{ij}^4)를 [표 1]과 같이 계산해 둔다.

문장 별 단어 출현 여부		단어i(W_i)	
		출현	미출현
단어j(W_j)	출현	S_{ij}^1	S_{ij}^2
	미출현	S_{ij}^3	S_{ij}^4

[표 1] 문장 별 단어 출현 여부

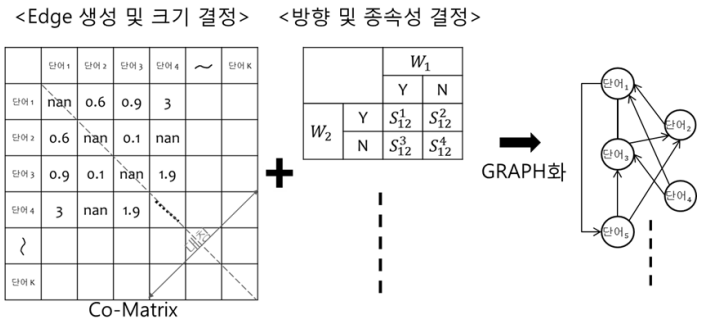
계산을 마친 후 TextRank 알고리즘을 응용하기 위해 그래프를 구성한다. 기존 연구에서는 비방향성 그래프를 사용했지만 본 연구에서는 Vertex간의 영향관계를 반영하기 위해 방향성 그래프로 구성된다.



[그림 3] Vertex 생성

먼저 그래프의 Vertex는 [그림 3]의 예와 같이 CO-Matrix에 속한 모든 단어마다 만들어지며 각 Vertex가 하나의 단어와 그 단어의 Score를 저장한다. Score의 초기값은 보통 1을 설정하는 것이 일반적이나 본 연구에서는 단어가 전체 문서에서 출현한 비율을 반영하기 위해 [식 3]과 같이 단어의 문장출현빈도를 활용한다.

$$\text{단어 } W_i\text{의 문장출현빈도}(f_i) = \frac{\text{단어}(w_i)\text{ 출현 문장 수}}{\text{전체 문장 수}} \quad \text{(식 3)}$$



[그림 4] Edge 형성

그래프의 나머지 구성요소인 Edge는 [그림 4]의 예시와 같이 두 Vertex 사이를 연결하며 말단의 단어들이 동시출현 관계임을 표시한다. Edge는 방향과 가중치를 갖는데 방향은 Edge 양 말단의 Vertex간의 '의존관계'와 '영향력'에 의해 결정되고 가중치는 그들의 동시출현 관계에 의한 '단어쌍 연관거리'로 결정된다.

먼저 Edge의 방향은 앞서 언급한 바와 같이 두 가지를 기준으로 정해진다. 첫 번째 기준은 Edge 양쪽의 Vertex들 사이의 의존관계여부이다. 의존관계란 Edge 양쪽으로 연결된 단어 중 하나(이하 종단어)가 자신의 출현이 반대쪽 단어(이하 주단어)의 출현에 의존하는 관계를 말한다. 즉, S_{ij}^1가 0보다 크면서 S_{ij}^2 또는 S_{ij}^3이 0인 관계를 의미한다. 이러한 관계를 설정하는 이유는 종단어가 출현할 확률이 주단어에 의존적이기 때문에 주단어의 Vertex가 가진 Score를 계산할 때 독립관계의 단어보다 더 큰 순영향을 미치도록 하기 위함이다. 이에 따라 Edge의 방향은 Edge의 양쪽 Vertex들이 의존관계를 가질 때 종단어에서 주단어로 향한다. 그러나 Edge가 서로 의존관계가 없는 독립쌍이라면 Edge의 방향은 두 번째 기준인 Vertex가 문서에서 갖는 영향력 차이에 의해 결정되며 Edge는 양 Vertex가 갖는 영향력에 따라 작은 쪽에서 큰 쪽으로 향한다. 여기서 영향력은 해당 문서에서 중요한 단어는 다른 다양한 단어들과 함께 나온다는 TextRank의 기본가설과 해당 문서에서 중요한 단어는 중요하지 않은 단어보다 자주 등장한다는 TF의 기본가설에 따라 Vertex마다 [식 3]의 문장출현빈도와 [식 4]의 Density의 곱으로 구한다. 여기서 Degree는 Vertex가 가진 Edge의 수를 의미하고 따라서 분모의 Possible Maximum Degree, 즉 최대가능 Edge개수는 자신을 제외한 그래프에 존재하는 Vertex의 수로써 문서에 출현했던 모든 종류의 단어들을 의미한다.

$$\text{Vertex } V_i\text{의 } Density(\rho) = \frac{Degree(V_i)}{Possible\_Maximum\_Degree} \quad \text{(식 4)}$$

Edge의 가중치는 Edge 양쪽의 Vertex V_i와 V_j의 '단어쌍 연관거리'로 결정된다. [식 5]의 단어쌍 연관거리는 Edge 양끝 단어들의 동시출현관계의 연관성이 얼마나 강한 것인지 나타내기 위해 사용되는 가중치로 값이 작을수록 연관성이 강하며 이는 [식 6]의 '동시출현점수'와 [식 7]의 '인접도'의 곱으로 나타낸다. 이때 [식 6]의 동시출현점수는 두 단어가 전체 출현 횟수 대비 독립적으로 출현하는 비율을 동시에 출현하는 비율로 나눈 값으로 크기가 작을수록 동시에 출현하는 비중이 높아 서로 연관성이 강한 관계임을 나타낸다. 또한 [식 7]의 인접도는 두 단어가 한 문장에서 동시에 출현한다면 물리적으로 얼마나 가까운 거리에서 출현하는 관계인지를 나타내는 것으로 두 단어가 동시에 출현한 경우 실제 두 단어 사이의 거리(l_c)를 문장 전체의 길이(L_c)로 나눈 비율들의 합을 전체 동시출현횟수(S_{ij}^1)로 나눈 값이다. 따라서 인접도 또한 동시출현점수와 마찬가지로 값이 작을수록 서로 연관성이 높은 관계임을 나타낼 수 있다.

$$V_i, V_j\text{간 연관거리}(D_{ij}) : P_{ij} * A_{ij} \quad \text{(식 5)}$$

$$V_i, V_j\text{간 동시출현점수}(P_{ij}) = \frac{\left(\frac{s_{ij}^2}{\sum_{k=1}^{3} s_{ij}^k}\right) + \left(\frac{s_{ij}^3}{\sum_{k=1}^{3} s_{ij}^k}\right)}{\frac{s_{ij}^1}{\left(\sum_{k=1}^{3} s_{ij}^k\right)}} \quad \text{(식 6)}$$

$$V_i, V_j\text{간 인접도}(A_{ij}) = \frac{\sum_{c=1}^{s_{ij}^1} l_c / L_c}{s_{ij}^1} \quad \text{(식 7)}$$

모든 그래프가 형성되고 나면 Main Process의 마지막 단계로 각 Vertex가 가진 중요도를 [식 8]에 따라 계산하며 d는 [식 9]와 같이 계산 대상 Vertex의 Edge중 Inner Edge의 비율로서 문서에서 중요 단어는 앞서 언급한 바와 같이 영향력이 크므로 Inner Edge의 비율이 높다.

$$S(V_i) = \underbrace{S_0(V_i)}_{①} + \underbrace{d * adj_{factor}}_{②} - \underbrace{(1-d) * (anti_{factor})}_{③} \text{ (단, } S_0 = f, 0 \le d \le 1) \text{ (식 8)}$$

$$d = \frac{In\ Degree(V_i)}{Degree(V_i)} \quad \text{(식 9)}$$

[식 8]은 보이는바와 같이 크게 세 부분으로 나누어진다. 첫 ①부분인 $S_0(V_i)$는 계산 대상 Vertex의 초기 값으로써 앞서 언급한 바와 같이 [식 3]에서 구해 저장했던 f_i이고 두 번째 ②부분은 그래프에서 대상 Vertex가 Inner Edge에 의해 받을 순영향 즉 '+' 에 관한 부분이다. 이 부분의 내부 요소인 adj_factor는 Edge의 의존관계 여부에 따라 [식 10] 또는 [식 11]로 계산된다. 특히 두 Vertex V_i와 V_j사이의 관계가 의존관계일 경우엔 주단어의 중요도를 높이는데에 직접적으로 영향을 준다고 보아 Edge가 가진 거리에 의해 그 영향력이 감쇠되지 않으나 독립일 경우엔 간접적으로 영향을 준다고 보아 V_j의 중요도가 Edge의 거리의 제곱에 반비례하여 합산된다. 이는 물리학의 역제곱법칙의 개념을 차용한 것인데 거리가 멀수록, 다시 말해 동시출현의 연관성이 적을수록 더 적은 영향을 주도록 하기 위함이다. 마지막 ③부분은 그래프에서 대상 Vertex가 Outer Edge에 의해 받을 악영향 즉 '-' 에 관한 부분이다. 이 부분의 내부 요소인 anti_factor는 Edge의 의존관계여부에 관계없이 [식 12]에 의해 계산된다. 이는 대상 Vertex가 Outer Edge에 의해 잃어버리는 것이므로 자신이 잃는 양은 Edge의 거리나 의존관계여부와 관계없이 일정하기 때문이다.

의존관계일 경우 : $adj_{factor} = \sum_{j \in In(V_i)} \frac{S(V_j)}{Out\ Degree(V_j)}$ (식 10)

독립일 경우 : $adj_{factor} = \sum_{j \in In(V_i)} \frac{S(V_j)}{Out\ Degree(V_j) * (D(V_{ji}))^2}$ (식 11)

$$anti_{factor} : \sum_{j \in Out(V_i)} \frac{S(V_i)}{Out\ Degree(V_i)} \quad \text{(식 12)}$$

이와 같이 모든 Vertex의 Score를 계산하고 나면 [5]와 같이 최종적으로 그들 중 상위 10%의 Score를 갖는 Vertex의 단어들을 키워드로 선정하게 된다.

4. 실험 및 평가

3장에서 제안한 키워드 추출기법을 이용하여 다음과 같은 2가지 실험을 진행하였다. 실험 대상은 [표 2]와 같이 [4]에서 실험 대상 분야로 선정했던 '정치,경제' 분야와 더불어 '컴퓨터, 물리학, 질병, 방위산업, 생태학' 를 더한 6개 분야의 논문들로 하였으며 각 분류 당 5개씩 총 30개 논문을 사용하였다. 각 분야를 구성하는 논문들은 [표 2]에서 보이는 바와 같이 전체 평균 156개 문장, 245개 단어들로 이루어져 있으며 분야별로는 컴퓨터 분야가 문서 당 평균 206개 문장으로 가장 많은 문장으로 이루어져있고 평균 단어 수에서는 정치,경제 분야가 문서 당 평균 296개로 가장 많은 단어를 갖고 있다.

문서 분야	문서 당 평균 문장의 수	문서 당 평균 단어의 수
컴퓨터	206	224
물리학	153	220
질병	139	256
방위산업	120	240
생태학	148	232
정치,경제	171	296
전체 평균	156	244

[표 2] 분야별 문장 및 단어 수 평균

먼저 첫 번째 실험에서는 [2]에서 평가한 바와 같이 추출된 키워드와 제목에 출현한 단어를 비교하여 정확도를 평가하는 실험을 진행하였다. 그 결과는 [표 3]와 같이 전체 평균 74.0%의 정확도를 보여 [2]의 결과인 52%의 정확도보다 향상된 정확도를 보였다.

문서 분야	분야별 정확도 평균 ((제목∩검출단어)/(제목))
컴퓨터	76.7%
물리학	70.4%
질병	80.3%
방위산업	77.2%
생태학	67.9%
정치,경제	71.8%
전체 평균	74.0%

[표 3] 제목에 출현한 단어 검출 정확도

다음은 [3]에서 수행한 바와 같이 추출된 키워드와 저자가 선정한 키워드 단어를 비교했을 때의 정확도를 평가하는 실험을 진행하였다. 실험 대상은 앞선 실험과 동일하며 결과는 [표 4]와 같이 분야마다 편차가 있지만 평균적으로 66.0%의 정확도를 보여 [3]의 결과인 36%의 정확도보다 향상된 정확도를 보였다.

문서 분야	분야별 정확도 평균 ((키워드∩검출단어)/(키워드))
컴퓨터	67.5%
물리학	61.3%
질병	79.6%
방위산업	73.9%
생태학	42.1%
정치,경제	71.7%
전체 평균	66.0%

[표 4] 저자 선정 키워드에 출현한 단어 검출 정확도

마지막으로 [6]에서 수행한 바와 같이 추출된 키워드

중 관련 전문가가 옳은 것으로 판정한 키워드의 비율을 Precision으로 하여 정확도를 산출하였다. 비록 [6]은 영어로 이루어진 문서로 실험하여 직접적인 비교에 큰 의의가 있는 것은 아니나 [표 5]에서 보이는바와 같이 평균적으로 40.0%의 Precision을 보여 [6]의 결과인 31.2%보다 나은 성능을 보였다.

문서 분야	분야별 Precision 및 평균 ((TP)/(TP+FP))
컴퓨터	35.5%
물리학	43.3%
질병	39.1%
방위산업	42.0%
생태학	38.9%
정치,경제	41.2%
전체 평균	40.0%

[표 5] 검출 단어의 분야별 Precision

위 세 실험의 결과로부터 제안한 방식이 기존 TF-IDF 기반 키워드 추출법이나 단어의 동시출현 정보만을 사용하는 방법보다 나은 결과를 얻을 수 있음과 동시에 기존 TextRank보다도 나은 성능을 얻을 수 있음을 확인할 수 있다.

5. 결론

본 논문에서는 정확하면서도 다양한 분야의 텍스트에 적용 가능한 키워드 추출 방법을 제시하고자 단어의 동시출현 정보와 그래프 모델을 바탕으로 TextRank 알고리즘을 변형한 새로운 형태의 알고리즘을 동시에 적용하는 키워드 추출 기법을 제안하였다. 그 결과 정확도 성능이 기존 연구들 대비 최대 31%의 증진됨을 볼 수 있었다. 또한 여러 분야의 텍스트 데이터들에 적용을 하여 검증함으로써 텍스트의 분야에 크게 영향을 받지 않는 방법임을 보였다. 그러나 실험과정에서 나타난 복합명사 처리 문제나 형태소 분석기의 Stemming 성능 문제로 인한 성능 하락 등은 Vertex 병합 또는 합성명사의 구성명사분석 등의 방법을 고안하여 보완하여야 할 과제이다.

참고문헌

[1] Y. J. Kumar et al., “A Review on Automatic Text Summarization Approaches”, Journal of Computer Science, 12(4), pp.178-190, 2016

[2] 한승희, “용어클러스터링을 이용한 단일문서 키워드 추출에 관한 연구”, Journal of the Korean Society for Library and Information Science ,44(3), pp.155-173, 2010

[3] 이장호, 윤성로, “짧은 문서에 대한 키워드 추출 알고리즘 성능 향상을 위한 새로운 방법”, 한국정보과학회 동계학술발표회 논문집, pp.578-580, 2015

[4] 홍진표, 차정원, “TextRank 알고리즘을 이용한 한국어 중요 문장 추출”, 한국정보과학회 학술발표논문집, 36(1C), pp.311-314, 2009

[5] G. K. Palshikar, "Keyword Extraction from a Single Document Using Centrality Measures", Proceedings of 2nd international conference on pattern recognition and machine intelligence, Vol. 4815, pp.503-510, 2007

[6] R. Mihalcea, P. Tarau, "TextRank: Bringing order into texts", Proceedings of EMNLP'04, pp.404-411, 2004

[7] B.Lott, “Survey of Keyword Extraction Techniques”, UNM Education, 2012

[8] H. P. Luhn, “The automatic creation of literature abstracts”, IBM J. Res. Dev., 2: pp.159-165. 1958