Journal of KIIT. Vol. 16, No. 2, pp. 1-16, Feb. 28, 2018. pISSN 1598-8619, eISSN 2093-7571 1
http://dx.doi.org/10.14801/jkiit.2018.16.2.1

TF-IDF 기반 키워드 추출에서의 의미적 요소 반영을 위한 결합벡터 제안

박대서*, 김화종**

A Proposal of Join Vector for Semantic Factor Reflection in TF-IDF Based Keyword Extraction

Dae-Seo Park*, Hwa-Jong Kim**

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임
(No.R-20160906-004163, 빅데이터 자동 태깅 및 태그 기반 DaaS 시스템 개발)

요 약

최근 빅데이터를 다루기 위한 기술 개발이 활발하게 이루어지고 있으며 과거에는 부족한 정보 속에서 정확한 정보를 찾는 것이 핵심 이었다면 이제는 많은 정보 속에서 정확한 정보를 찾아내는 것이 핵심이 되어가고 있다. 정확한 정보를 찾아내는 것은 사용자 만족도 향상, 업무효율 향상 등 다양한 방향으로 긍정적인 효과를 발휘할 수 있다. 이에 본 논문에서는 기존에 연구되고 있던 통계 기반의 키워드 추출 방식에 의미 기반 키워드 추출 방식을 결합하여 새로운 키워드 추출 방법을 제안한다. 키워드 추출 방법에는 TF-IDF와 Word2vec을 활용하며 TF-IDF로는 단어 빈도수 기반으로 뉴스기사를 통계적으로 벡터화하고 Word2vec의 단어간 유사도 점수를 사용해 뉴스기사 내의 특정 단어와 다른 단어들간의 유사도 평균을 이용해 의미적으로 벡터화한다. 최종적으로 두 벡터를 결합한 결합벡터를 통해 키워드를 추출하고 본 연구의 성능을 평가한다.

Abstract

Recently, there has been a brisk technological development to handle big data. In the past, finding exact information based on insufficient information was the key. Finding the right information in a lot of information is now becoming key. Finding the right information can help improve user satisfaction and improve performance. In this paper, a new method of keyword extraction is proposed by combining the traditional statistical-based method of keyword extraction with the semantic-based method of keyword extraction. Use TF-IDF and Word2vec to extract keywords. TF-IDF vectorizes news articles based on word frequency, and Word2vec vectorizes news articles on the basis of similarity score between words. Finally, keywords are extracted with a combined vector combination and the performance of this study is evaluated.

Keywords
keyword extraction, news article, semantic vector, join vector, text analysis

* 강원대학교 컴퓨터정보통신공학과 석사과정
- ORCID: https://orcid.org/0000-0001-7421-5926
** 강원대학교 컴퓨터정보통신공학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0002-3822-390X

· Received: Dec. 22, 2017, Revised: Jan. 10, 2018, Accepted: Jan. 13, 2018
· Corresponding Author: Hwa-Jong Kim
Dept. of Computers and Communications Enginieering Kangwon National Univ., Gangwondaehak-gil 1, ChunCheon-si, Gangwon-do, Korea
Tel.: +82-33-250-6323, Email: hjkim3@gmail.com

Ⅰ. 서 론

유무선 네트워크의 발달과 스마트 기기의 도입에 따라 인터넷은 끝없는 정보들을 지속적으로 쌓아가고 있다. 과거 사람들이 도서관, 신문, 사전 등에서 정보를 습득했던 것과 비교했을 때, 현재는 앉은 자리에서 다양한 기기를 이용해 손쉽게 정보를 얻을 수 있게 되었다[1]. 하지만, 정보를 손쉽게 얻을 수 있다는 장점이 있는 반면, 많은 정보만큼 왜곡된 정보도 많이 존재한다. 인터넷 사용자가 늘어갈수록 잘못된 정보 문제는 더욱 대두될 것이기 때문에 올바른 정보를 찾아주고 사용자가 원하는 정보를 추천하는 기술이 더욱 더 필요해 질 것이다[1]-[3].

구글, 네이버 등 인기 검색엔진의 경우와는 다르게 일반적인 인터넷 뉴스 사이트에서는 질의가 제목과 내용에 포함되어 있으면 날짜 순서대로 보여주는 방식이 대부분이다[4]. 이 때문에 다시 뉴스기사를 탐색하거나 뉴스기사를 하나씩 읽어보고 좋은 뉴스인지를 판단하게 된다.

최근에는 조회수, 댓글수 등을 이용해서 사용자에게 인기뉴스기사를 제공하는 방식을 적용하여 질의에 대한 뉴스기사를 선택하고 조회수, 댓글수, 추천수를 기준으로 정렬하여 제공함으로써 신뢰성 있는 기사를 제공하는 방법이 활용되고 있다[5]. 이 방법을 통해 뉴스기사는 다수의 사용자에게 인증을 받게 되며 제목과 내용에만 의존하는 방법에 비해 효과적이다. 하지만, 조회수와 댓글의 경우 초기에 상단에 노출된 기사에 더 많이 할당되는 문제가 있으며, 추천수 방식은 사용자의 참여에 절대적으로 의존하기 때문에 한계점이 있고, 일정 횟수 이상의 추천이 이루어지지 않으면 평가하기 어렵다[5].

따라서 뉴스기사 뿐만 아니라 인터넷에서 정보를 제공하는 사이트 및 페이지에서는 사용자에 의존하지 않고 자체적으로 질의에 대한 올바른 검색결과를 제공할 수 있도록 뉴스기사의 주요 정보를 키워드로 추출하고 키워드를 이용하여 질의응답이나 관련 있는 뉴스기사 탐색에 활용한다면 단순한 검색어의 포함여부에 따른 결과보다 효과적으로 응답 뉴스를 선정할 수 있을 것이다.

이러한 필요성에 따라 본 논문에서는 국내 인터넷 뉴스기사를 활용하여 뉴스기사 내에서 의미적 요소를 반영하여 주요 키워드를 추출하는 방법에 대해 연구하고 키워드 기반의 뉴스기사 탐색을 통해 과거의 통계 방식의 키워드 추출 방식과 비교 평가하여 본 연구의 효과를 검증한다.

본 논문은 문서간의 전체 비교를 통한 뉴스기사 탐색을 대체할 수 있는 키워드 기반의 뉴스기사 탐색을 위해 새로운 키워드 추출 방식을 제안한다. 우선, 뉴스기사의 주요 키워드 추출을 위해 본 논문에서는 불필요한 뉴스기사 내의 성분과 정보를 제거하는 작업을 수행한다[6]. 다음으로 단어의 원형을 복원한다. 이를 통해 벡터화 단계에서 고려해야하는 단어 수를 줄이고 단어 간의 유사도 비교도 효율적으로 수행 할 수 있다.

주요 키워드 추출을 위한 방안으로 본 논문에서는 기존의 TF-IDF(Term Frequency-Inverse Document Frequency)를 이용한 벡터와 단어 유사도 기반의 의미벡터를 결합한 벡터화 방법을 제안하며 이후, 결합 벡터라 지칭한다. 의미벡터는 본 논문에서 제안하는 벡터로 문서에 포함된 단어들 간의 유사도를 벡터 값으로 사용해 뉴스기사 내 다른 단어들과 높은 유사도를 보이는 단어를 추출하기 위한 방법이다. 벡터화에 따라 각 뉴스기사에서 가장 높은 점수를 받은 키워드를 선택해 뉴스기사의 주요 키워드로 선정한다.

마지막으로 뉴스기사로부터 추출된 결합 벡터와 기존 키워드 추출 방법인 TF-IDF와의 성능평가를 통해 본 논문에서 제안한 키워드 추출 방법의 성능을 평가한다. 인터넷 뉴스기사는 네이버의 IT 일반 분야 뉴스기사 8만 건을 활용해 연구를 수행한다.

Ⅱ. 관련 연구

2.1 주요 활용 기술

2.1.1 TF-IDF

TF-IDF는 정보검색과 텍스트 마이닝 분야에서 활용되고 있으며 여러 문서에서 단어가 특정 문서 내에서 얼마나 중요한지를 나타내는 통계기반 기술이다[7][8]. 키워드 추출과 검색 결과의 순위 결정,

문서 유사도 측정에 활용된다. TF는 단어 빈도수를 나타내며 단어가 문서 내에서 얼마나 등장했는지를 나타내는 값이고 IDF는 DF값의 역수를 취한 값으로 DF는 문서 집합에서 단어가 얼마나 등장했는지를 표현한 값이다. 예를 들어 '스마트폰'이라는 단어는 정치 문서 집합에서는 잘 나오지 않기 때문에 IDF 값이 커지고 동시에 해당 문서의 키워드가 될 수 있다. 반대로 모바일 문서 집합에서는 '스마트폰'은 많은 문서에서 나타나기 때문에 IDF값은 낮아지게 된다. TF의 경우 계산 방법이 다양하며 IDF와 함께 표 1과 같이 나타낸다.

표 1. TF-IDF 수식 정의
Table 1. TF-IDF formula definition

Equation		Definition
TF	Frequency (basic)	$tf(t,d) = \frac{f(t,d)}{\sum f(w,d)}$ $f(t,d)$: The number of word t appeared in document d $\sum f(w,d)$: Length of document d
	boolean Frequency	$tf(t,d) = 0,1$ 1 if word t appears in document d, 0 if not
	Log scale frequency	$tf(t,d) = log(tf(t,d)+1)$ Resizing of value $tf(t,d)$
	Increment frequency	$tf(t,d) = 0.5 + \frac{0.5 \times f(t,d)}{max\{f(w,d) : w \in d\}}$ Calculation of the number of words by document length
IDF		$idf(t,D) = log\ \frac{\lvert D \rvert}{\lvert\{d \in D : t \in d\}\rvert}$ $\lvert D \rvert$: Total number of documents $\lvert\{d \in D : t \in d\}\rvert$: The number of word t appeared in document d
TF-IDF		$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$

TF는 기본적으로 문서 내의 단어 총 빈도수를 사용해 계산할 수 있지만 이 방식은 문서가 크거나 단어수가 많아질 경우 값이 지속적으로 커지는 문제가 있다. 크기 문제 개선을 위한 방법으로 3가지 방식을 정의 할 수 있으며 첫째로 불린 빈도는 단어의 출현 여부만으로 tf 값을 0과 1로 정의한다. 두 번째로 로그 스케일 빈도는 TF값에 로그를 취함으로써 크기 문제 해결과 실제 빈도수를 반영할 수 있다. 마지막으로 증가 빈도는 문서 길이에 따른 단어의 상대적 빈도를 나타내는 방식으로 최대 스케일이 1을 넘지 않는다. IDF는 전체 문서수를 해당 단어를 포함한 문서의 수로 나눈 뒤 로그를 취해 계산한다. TF-IDF는 단어가 특정 문서 내에서 빈도수가 높고 전체 문서 중 해당 단어가 포함된 문서가 적을수록 높아진다. 이를 통해 모든 문서에서 자주 나타나는 단어들을 걸러낼 수 있다.

2.1.2 벡터공간 모델

벡터 공간 모델은 텍스트 마이닝 분야에서 단어 공간 모델로 불리며 주어진 텍스트 문서를 단어들의 벡터로 나타낸 대수적인 모델이다[9]. 벡터 공간 모델은 정보 필터링과 정보 검색, 색인, 유사도 순위에 사용된다. 어떤 단어가 문서에 포함되면 해당 단어는 0이 아닌 벡터 값을 갖게 된다. 단어 가중치라 불리는 이 값을 산출하는 방법에는 여러 가지가 있으며 대표적으로 TF-IDF 방식이 있고 해당 문서에서 단어의 출현 빈도를 그대로 사용하여 벡터 공간에 적용할 수도 있다.

2.1.3 단어 임베딩

단어 임베딩은 텍스트를 구성하는 단어들을 수치화하는 방법을 말한다[10]. 이 방법론을 통해 단어를 벡터 공간에 표현할 수 있다. 과거에는 원-핫 인코딩 방식을 이용해 단어를 벡터화 하였다. 원-핫 인코딩은 N개짜리 사전이 있을 때 특정 단어를 표현하기 위해서 크기 N인 벡터를 만들고 그 단어가 해당되는 자리에 1, 나머지는 0을 넣어서 특정 단어를 벡터로 표현한 것을 말하며 Naive Bayes를 적용한 스팸 분류기가 이 방법을 사용한다. 현재에 와서는 두 가지 형태의 방법론이 존재하며 카운트 기반 방법과 예측 기반 방법으로 나뉜다. 카운트 기반 방법은 카운트 벡터, TF-IDF 벡터 등 빈도수를 이용한 방법이며 예측 기반 방법은 신경망 모델을 적용하여 단어, 문장 등이 주어지면 다음에 올 수 있는 단어를 예측하는 방식으로 동작하며 현재의 Word2vec이 여기에 해당한다. Word2vec은 CBOW와 Skip-Gram 방식으로 단어 임베딩을 수행하며 CBOW

(Continuous Bag of Words)는 여러 개의 단어를 나열한 뒤 이와 관련된 단어를 예측하고 Skip-Gram은 CBOW와는 반대로 특정 단어로부터 문맥이 될 수 있는 단어를 예측한다.

2.1.4 텍스트 분석 패키지

KoNLPy는 파이썬 기반의 한국어 정보처리 패키지로 오픈소스 소프트웨어이다[11]. 이 패키지는 간단한 용어검색부터 형태소 분석, 품사 태깅 등의 기능을 제공한다. 또한, 단독으로 사용되기보다 파이썬의 시각화, 텍스트분석 패키지와 함께 사용 가능하다[11].

Gensim은 벡터 공간 모델링을 위한 딥러닝 패키지로 TF-IDF, LDA, Word2vec 등의 기능을 제공한다. Word2vec의 경우 이 패키지를 적용하면 Numpy로 구현한 것보다 70배의 속도를 낸다고 한다. 빠른 속도를 통해 대규모 텍스트 데이터를 효율적으로 다룰 수 있도록 지원한다[12].

2.1.5 코사인 유사도

내적공간의 두 벡터간 각도의 코사인 값을 이용하여 측정된 벡터간의 유사도를 의미한다[13]. 각도가 0일 때 코사인 값은 1, 다른 모든 각도는 1보다 작기 때문에 벡터의 크기가 아닌 방향의 유사도를 판단하는 목적으로 사용되며 방향이 같을 경우 1, 직각일 경우 0, 반대 방향인 경우 -1을 갖는다. 흔히, 정보 검색, 텍스트 마이닝 분야에서 단어, 문장, 문서 벡터 간에 유사도를 측정하는데 유용하게 사용된다. 코사인 유사도는 A, B 벡터 값이 주어졌을 때, 벡터의 스칼라 곱과 크기로 식 (1)과 같이 표현한다.

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (1)$$

2.1.6 NDCG(Normalizing Discounted Cumulative Gain)

검색 및 연관뉴스 탐색 시 나타난 결과를 순위 정보까지 반영하여 평가 할 필요가 있는 경우 NDCG를 활용한다.

NDCG는 DCG값을 정규화한 값으로 실제 검색결과 순서에 따른 패널티를 적용하여 점수를 도출한다. 식 (2)는 DCG값을 표현한 것이며 rel은 검색결과 목록에서 사용자가 평가한 i번째 검색내용의 점수를 나타낸다[14].

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i} \quad (2)$$

DCG 계산 결과는 현재 검색결과의 상태를 보여주며 가장 이상적인 검색결과 점수로 DCG 값을 나누면 정규화된 DCG를 얻을 수 있는데 이를 NDCG라 한다. 식 (3)은 NDCG를 표현한다. IDCG는 현재 검색결과의 가장 이상적인 DCG 점수를 의미하며 5개 검색결과에 대해 사용자가 부여한 점수가 3, 3, 0, 2, 1인 경우 이상적인 점수는 3, 3, 2, 1, 0으로 높은 순위일수록 높은 점수를 받은 경우이다.

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad (3)$$

2.2 선행연구

키워드 추출에 관한 연구는 지속적으로 이루어지고 있으며 추출한 키워드를 활용하여 문서의 검색, 분류 등에 활용되고 있다. 인터넷 상에 공유된 문서의 키워드 추출만으로도 큰 의미를 갖지만 키워드를 기반으로 하여 문서들을 카테고리 별로 분류하고 문서를 대표하는 키워드를 검색에 활용함으로써 빠른 문서 탐색에도 기여한다. 검색에 있어서 시스템이 방대한 양의 문서 데이터를 매번 다루는 것은 시스템 자원소모가 크며 응답 시간의 지연을 야기시킬 수 있다. 이처럼, 키워드는 정보검색 분야에서 다양하게 활용되고 있으며 키워드 추출을 위한 다양한 방법들이 연구되고 있다.

[15]의 연구에서는 텍스트 마이닝 분야에서 널리 알려진 TF-IDF의 변형을 이용해 전자뉴스에서의 키워드 추출 기법을 연구하였다. 총 6가지의 TF-IDF 변형식을 고안하여 각 분야별 후보 키워드를 추출하고 후보 키워드들 간의 교차비교분석을 수행해 의미 없는 키워드를 제거하여 성능을 높였다.

표 1의 기본 TF-IDF를 바탕으로 [15]의 연구에서 제시한 6가지 TF-IDF의 변형식은 표 2와 같다.

BTF는 문서 전체 집합에서 특정 단어가 출현한 횟수의 합을 나타낸 기본 TF 식이다. NTF1은 최대 출현 단어 횟수로 특정 단어의 출현 횟수를 나누어 나타내었고 NTF2는 특정 문서에서 나타난 단어의 수를 모든 단어의 빈도수 합으로 나누어 더한 식으로 두 NTF는 정규화 된 값을 사용한다. BTF는 문서의 길이에 차이가 있을 경우를 반영하지 못하기 때문에 NTF1과 NTF2에서 문서 길이 차이로 발생할 수 있는 편차를 줄이고자 하였다.

표 2에서와 같이 [15]의 연구에서는 기존 TF-IDF 식을 변형하여 문서 집합을 대표하는 키워드를 추출하는 연구를 수행하였다. 또한, TF-IDF 변형식을 이용해 특정 분야 문서 집합의 후보 키워드를 추출하여 타 분야 문서 집합의 후보 키워드와 비교분석하여 타 분야 후보 키워드와 겹치지 않고 현재 문서 집합에서의 등장비율이 높은 경우 대표 키워드로 선출하는 방식을 적용하였다. [15]의 연구는 TF-IDF의 변형식을 제안한 점에서는 의의가 있으나 단일 문서 별 키워드가 아닌 전체 문서 집합에서 키워드를 추출하는 방식으로 연구를 진행하여 특정 분야나 문서 집합의 대표 키워드는 추출할 수 있지만 각 문서 별 키워드는 추출 할 수 없으며 검색을 목적으로 사용하는 것이 어렵기 때문에 한계가 있다.

[16]에서는 용어 클러스터링 기법을 활용하여 단일문서를 이용해 해당 문서내의 키워드 추출에 대해 연구하였으며 용어 클러스터링은 문서 내에서 관련된 용어들을 모아 클러스터를 구성하는 방법을 말한다. [16]에서는 문헌정보학 분야의 국내 학술지 20편을 대상으로 연구를 수행하였으며 기존 TF-IDF의 개념을 단일 문서에 적용하였다.

연구 방법은 그림 1과 같이 7단계로 구성된다.

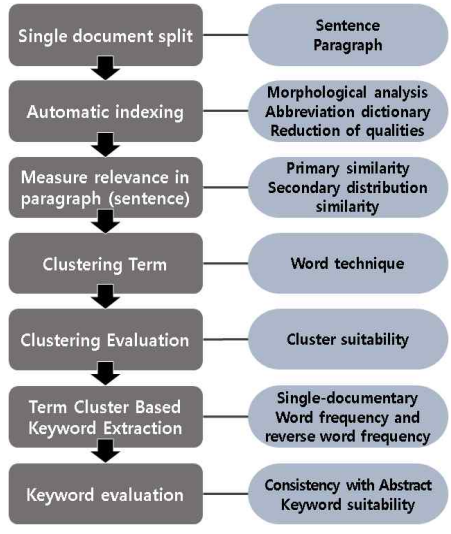

그림 1. 용어 클러스터링을 이용한 키워드 추출 연구 단계
Fig. 1. Keywords keyword extraction research step using clustering

표 2. [15]에서 제안하는 TF-IDF 변형식
Table 2. TF-IDF variant format proposed in [15]

TF-IDF Format		
TF variants	BTF (Basic Term Frequency)	$BTF_i = \sum_{f=1}^{\lvert D\rvert} n_{i,j}$ Number of word I in the entire document D
	NTF1 (Normalized Term Frequency 1)	$NTF1_i = \frac{BTF_i}{Max\{BTF_1, \cdots, BTF_{\lvert T\rvert}\}}$
	NTF2 (Normalized Term Frequency 2)	$NTF2_i = \sum_{j=1}^{\lvert D\rvert} \frac{n_{i,j}}{\sum_{k=1}^{\lvert T_j\rvert} n_{k,j}}$
TF-IDF variants	TF-IDF transform	$tfidf_i = (TFvariant) \times idf_i$ (BTF, NTF1, NTF2 Each applies = Three)
	TF-IDF Apply log scale	$ntfidf_i = (log(TFvariant)+1) \times idf_i$ Apply log scale = Three

그림 1에서 핵심 부분은 단락 내 용어 연관성 측정과 용어 클러스터링, 용어 클러스터 기반 키워드 추출 단계로 볼 수 있으며 단락 내 용어 연관성은 용어들 간의 유사도를 코사인, 피어슨 상관관계를 이용해 측정하고 비교를 통해 용어 클러스터링에 적합한 유사도 방식을 선정한다. 이후, 유사도를 통해 생성된 용어-용어 행렬로 워드 클러스터링을 수행해 문서 별로 10개의 용어 클러스터를 생성한다. 마지막으로 각 클러스터 별로 TF-IDF 방식의 키워드 추출 기법을 적용하여 각 클러스터별 키워드를 추출하고 키워드들을 해당 문서의 주요 키워드로 선정하였다. 여러 문서 단위에서 수행하는 TF-IDF를 단일 문서에 적용하기 위해 문서를 구성하는 단락, 문장 단위로 단어 빈도를 측정한다.

성능 평가는 식 (4)와 식 (5)의 추출 키워드의 적합도, 초록과의 일치도를 이용해 측정하였으며 측정결과 $tf \times ipf$가 가장 높은 성능을 보였다. 이러한 결과는 정규화를 통해 문장, 단락 내의 단어 빈도 차이를 축소화 한 것이 다른 기법에 비해 높은 결과를 낸 이유라고 설명하였다.

$$\text{초록과의 일치도} = \frac{\text{초록에 출현한 키워드 수}}{\text{문서당 추출된 키워드 총 수}} \tag{4}$$

$$\text{키워드 적합도} = \frac{\text{키워드로 적합한 용어 수}}{\text{문서당 추출된 키워드 총 수}} \tag{5}$$

[16]의 연구에서는 기존의 여러 문서를 이용해 키워드를 추출하는 TF-IDF 기법을 단일 문서로 축소 적용하였다는 점에서 의의가 있는 연구이며 용어 클러스터 기법을 활용해 관련 있는 용어들을 하나의 클러스터로 묶고 클러스터로부터 핵심 키워드를 추출하는 차별화된 연구를 수행하였다. 하지만, 연구를 적용한 논문데이터가 20개 밖에 되지 않으며 산출된 키워드를 실제 검색 서비스에 적용 시 효과에 대한 연구가 진행되지 못해 실효성에 대한 문제가 있다.

2.3 본 연구의 차별성

[15]의 연구는 유명한 키워드 추출 기법인 TF-IDF를 변형하여 전체 문서집합의 주요 키워드를 산출하였다. 하지만, 문서집합 전체에 대해 추출한 키워드는 단일 문서를 탐색하거나 요약하는 등에는 활용할 수 없기 때문에 문서별로 키워드를 추출하고 활용할 수 있는 기술이 필요하다. 본 연구에서는 각 문서별 주요 키워드 추출을 목표로 한다. 주요 키워드 추출에는 문서 내의 불필요한 문장을 제거하여 문서 내의 노이즈를 최소화 하는 방법을 적용하며 문장제거에는 통계, 의미벡터 기반의 유사도 측정방법을 이용해 불필요 문장의 선별 및 삭제 방법을 제안하고 결합 벡터 기법을 제안하여 기존 TF-IDF와의 차별성을 보인다.

[16]의 연구에서는 용어 클러스터 기반 키워드 추출 기법을 제안하였으며 단일 문서 내에서 문장과 단락을 중심으로 TF-IDF 기법을 재정의 하여 활용하였다. [16]의 연구에서는 분석 문서가 20개로 작은 규모이며 뉴스기사에 적용 시 초록과의 일치도 등의 평가는 불가능 하다. 이에 본 논문에서는 충분한 수의 국내 뉴스기사 데이터를 확보하여 분석과 성능평가에 활용하며, 다른 공인된 평가 모델을 이용한다. 또한, 본 논문에서는 전체 뉴스기사를 벡터 공간 모델링을 수행하는 Gensim 패키지에 입력해 단어 간의 유사도를 학습하고 학습된 정보를 이용해 문서 내의 단어들 간의 유사도 측정에 활용한다. 또한, 앞의 2가지 연구는 모두 통계 기반 벡터 공간 모델을 사용하기 때문에 문서의 문장 수나 핵심 키워드의 출현 빈도가 낮으면 주요 키워드를 잘 찾아내지 못하는 문제가 있다. 따라서, 문장 길이나 키워드 빈도에 영향을 적게 받는 키워드 추출 방법이 필요하다.

본 연구는 뉴스 키워드를 이용해 검색된 뉴스기사 들의 적합성을 판단하여 새로운 키워드 추출방식과 기존 키워드 추출 방식의 성능 차이를 보인다.

III. 뉴스기사 분석

3.1 데이터 수집 및 전처리

키워드 분석을 통한 뉴스기사 추천 방안의 연구 프로세스는 그림 2와 같다.

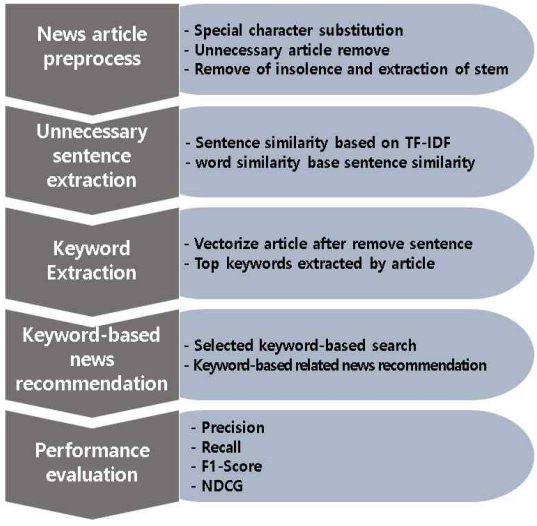

그림 2. 키워드 기반 뉴스 검색 탐색 프로세스
Fig. 2. Keyword-based news search process

본 연구에서는 네이버에서 제공하는 2017년 1월부터 2017년 9월까지의 IT 일반 뉴스기사 약 8만건을 크롤링하여 사용하였다. 크롤링의 특성 상 내용에 HTML 태그가 포함되기 때문에 크롤링 단계에서 정규표현식을 이용해 태그와 기본 특수문자를 제거하고 반복적으로 나타나는 관련 없는 안내문구도 제거하였다. 수집한 데이터의 속성은 기사의 제목, 작성일, 내용, 링크주소 4가지이며 실제 키워드 추출은 기사 내용을 분석하여 수행하였다. 제목, 작성일, 링크주소의 경우 검색 결과의 범위와 미리보기 등을 위해 사용한다.

데이터 수집 단계에서 특수문자와 HTML태그 제거 작업을 수행하였지만, 전체 뉴스기사에서 사용하는 특수문자가 다양하여 2차 제거작업이 필요하다. 수집단계에서는 모든 뉴스를 탐색해 보기가 제한되어 어떤 특수문자들이 사용되고 있는지 정확히 파악이 어려웠기 때문에 수집 후 추가 작업을 수행하였다. 이후, 뉴스기사 중 사진, 표, 이미지 등의 형태로 작성되어 있어 텍스트형태로 변환하지 못한 기사와 특정 기업의 인사이동과 같이 의미성이 떨어지는 기사를 제거하였다. 표 3은 수집된 데이터의 일부 항목이다.

뉴스기사 수집의 경우 구현상 API사용이 편리하나 일회 요청 시 수집할 수 있는 뉴스기사 수의 제한과 추후 연구에서 필요한 항목을 선택적으로 수집하기 위해 수집용 크롤러를 직접 구현하였다. 일반적으로 인터넷 뉴스기사는 현재 날짜의 목록만 보여주기 때문에 다른 기간의 뉴스기사는 검색을 필요로 한다. 따라서 본 논문에서는 여러 기간의 뉴스기사 수집 자동화를 위해 뉴스기사 목록 URL상에 날짜 파라미터를 추가하여 해당 기간의 뉴스기사 목록을 크롤링해 뉴스기사 각각의 URL을 수집하였다. 이후, 수집된 뉴스기사 URL 목록에 대해 각각 접근하여 제목, 내용, 작성기간 정보를 뉴스기사별로 수집하였다. 뉴스기사 목록과 뉴스기사 별 정보를 태그로부터 읽어내는 작업은 Python에서 제공하는 라이브러리를 활용하였다.

수집된 초기 뉴스기사의 내용에는 HTML 태그와 특수문자가 포함되어 있기 때문에 정규표현식을 이용해 해당되는 문자를 탐색하고 해당되는 문자를 제거하는 1차 전처리 작업을 수행하였다. 또한, 개별적으로 의미를 갖지 못하는 단어의 삭제를 위해 별도의 불용어 사전을 구축하였으며 불용어 사전 구축을 위해 각 뉴스기사 별로 품사태깅을 수행하였고 조사, 어미, 구두점 등 개별적으로 의미를 갖지 못하는 단어들로 사전을 구축하였다.

또한, 같은 단어이지만 접사등에 의해 프로그램상 다른 단어로 인식되는 경우를 제거하기 위해 어간 추출 작업을 수행하였다.

표 3. 수집 뉴스기사의 일부 항목
Table 3. Some items in the collection news article

	Title	Contents
1	2020 Join space technology independent nations	Korea which is late competitor of Korean space development history and present address universe development field secured
2	Accelerate innovation of new technologies such as AI and VR.	Information Technology IT industry will accelerate innovation of new technologies such as AI Virtual Reality VR, Augmented Reality AR this year.
3	Samsung Drum transporter all-in-one washing machine	Samsung Electronics is bringing in FlexWashing with a small and large washing machine in Las Vegas on July 5.

어간 추출과 품사 태깅은 KoNLPy의 트위터(Twitter) 태거를 사용하여 수행하였다. 트위터 태거를 사용하면 입력된 문서를 단어 단위로 분석하여 단어의 품사를 파악해 주기 때문에 이를 이용하면 불용어사전 구축도 용이하다. 표 4는 불용어와 어간 추출 과정을 거친 문장과 원본 문장의 예시이다.

3.2 불필요 문장 제거

본 연구에서는 뉴스기사 내에서 불필요한 문장들을 선택하기 위해 TF-IDF를 통해 계산된 문장 유사도와 단어 간 유사도 평균을 통해 계산된 결과를 반영한다. 2.1절에서 설명한 것과 같이 TF-IDF는 다른 문서까지 고려한 통계 기반 벡터화를 수행 한다. 하지만, 문서 내에서의 의미적 중요도가 반영되지 않기 때문에 빈도가 낮은 단어는 의미가 있어도 작은 값을 갖게 되는 현상이 나타난다. 따라서, 이 벡터를 이용해 문서간의 유사도를 측정하면 통계 기반의 유사도가 나오게 된다. 문장 제거를 위해서는 단순히 단어들의 빈도만을 고려하는 것보다 단어들의 의미적 요소도 고려하는 것이 훨씬 효과적이라 볼 수 있다. 따라서 문장 간의 의미적 유사도와 통계적 유사도를 측정해 결합하는 방법으로 두 문장의 유사도를 계산한다. TF-IDF를 이용한 통계적 유사도 측정은 파이썬의 기계학습 패키지인 Scikit-learn의 TfidfVectorizer를 활용 하였고 의미적 유사도는 Word2vec 모델을 전체 뉴스기사 데이터로 학습시켜 단어 간 유사도 측정에 활용하였다. 통계적 유사도는 두 문장의 TF-IDF 벡터를 코사인 유사도로 계산함으로써 두 문장이 얼마나 유사한지를 측정하고 의미적 유사도는 두 문장에서 나타난 각 단어 간 유사도의 평균값을 두 문장의 유사도로 계산한다. 구체적인 계산 방법은 수식 6을 따른다.

A, B는 문장을 의미하며 a, b는 각 문장의 단어 집합을 의미한다.

$$similarity(A,B) = \frac{\sum_{i=0}^{n} \max(similarity(a_i, b))}{len(a)} \tag{6}$$

$max(similarity(a_i, b))$는 A문장의 i번째 단어 a와 b 단어 집합과의 유사도를 측정하고 최대 유사도를 선택한다. 최종적으로 최대 유사도 집합의 평균값 계산을 위해 a집합의 크기로 나누어 A, B 문장의 유사도를 계산한다. 이렇게 계산된 각 문장 간의 유사도로부터 타 문장들과 가장 낮은 유사도를 보이는 N개 문장을 불필요 문장으로 선택한다.

표 4. 원본 문장과 단어 전처리 후 문장
Table 4. Sentence after original sentence and word preprocessing

Original sentence	Sentence after word preprocessing
The attributes of the collected data are the title, creation date, content, and link address of the article. The actual keyword extraction was done by analyzing the articles. For headings, date of creation, and link addresses, use it for previewing search results.	Collection Data Attribute Articles Title Creation Date Link Address 4 Use to preview search result scope, etc. on the date of link address when performing actual keyword extraction analysis
Word 2 vec does word logging in a CBOW and skip-Gram manner. Continuous Bags of Words (CBOW) lists several words and then estimates the words associated with them. Skip-Gram, contrary to CBOW, predicts words that can become context for a particular word.	Word 2 vec requires CBOW and Skip Gram type word logging CBOW Continuous Bags of Words List and Estimate for words related to this Skip Gram predicts the context word from the word CBOW
Remove pre-processing and non-relativity sentences from news articles. 15 keywords were extracted from each TF-IDF, semantic vector, and bound vector.	Remove the all-processing news article related sentence for news article data Extract 15 keysaut vectors from TFIDF meaning vector coupling

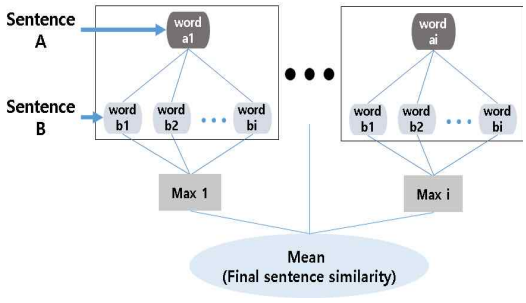

그림 3. 두 문장 사이의 의미적 유사도 측정 프로세스
Fig. 3. Semantic similarity measurement process between two sentences

"Galaxy" related word	"OLED" related word
Note – 0.91629	Organic light emitting diode – 0.88735
S – 0.74195	Flexible – 0.86161
8 – 0.72404	LCD – 0.84362
FE – 0.68881	Pannel – 0.81422
Galexy – 0.67532	Flexable– 0.76510
Return – 0.64893	Display – 0.76324
FEFa – 0.63231	AMOLED – 0.70551
UnPack – 0.60352	BOE – 0.68584
Galaxx – 0.60131	Full Active – 0.67145
"Speech Recognition" – "AI" Similarity	**"AI" – "Intellegent" Similarity**
0.74932	0.88947

그림 4. Word2vec 모델 기반 단어 유사도 측정 결과
Fig. 4. Word2vec model-based word similarity measurement result

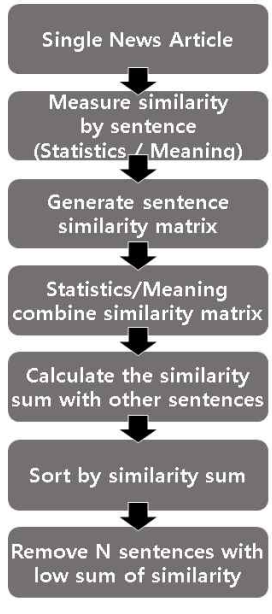

그림 5. 불필요 문장 제거 프로세스
Fig. 5. Unnecessary sentence remova process

기존 TF-IDF 방식이 단어들의 통계치를 벡터화하여 벡터공간에 대해 유사도를 측정하고 비슷한 공간에 매핑 될 때 유사하다고 본다면, 단어들 간의 유사도를 계산해 측정하는 방식은 문서 내에서 여러 문장에 같이 등장하는 단어 간에는 의미가 있다고 판단하고 A, B문장에 포함된 단어들 간의 유사도를 측정해 같이 쓰이는 단어들이 많을 경우 유사하다고 본다. 그림 3은 두 문장 사이의 단어간 유사도를 이용해 문장 유사도를 도출하는 프로세스를 나타내며 두 문장에서의 단어간 의미적 유사도의 측정은 Word2vec 모델을 활용해 계산할 수 있다.

본 논문에서는 수집한 전체 뉴스기사 8만건을 입력데이터로 학습하였으며 Word2vec 모델 생성 함수의 파라미터 설정으로 모델 성능을 향상시켰다. 그림 4는 Word2vec 모델에 의해 생성되는 단어 간 유사도를 나타낸다. Gensim의 Word2vec 모델을 이용해 그림 4의 예시와 같이 단어 간의 유사도와 해당 단어와 가장 유사한 단어 리스트를 얻을 수 있다.

기존의 TF-IDF 기반 통계 벡터와 그림 3의 두 문장 사이의 의미적 유사도 측정 방법을 이용해 그림 5과 같은 과정으로 불필요 문장 제거를 수행한다.

불필요 문장 제거는 특정 뉴스기사에 포함된 문장들간의 유사도를 측정하고 관련도가 낮은 문장을 제거하는 방식이다. 본 논문에서는 TF-IDF를 이용해 단일 뉴스기사를 벡터화하였으며 문장 별 벡터 값을 코사인 유사도 공식을 적용하여 뉴스기사 내 문장들과의 유사도를 각각 계산하여 통계 기반 문장 유사도 행렬을 생성하였다. 또한, 그림 3의 의미적 유사도 측정 방식을 이용하여 의미 기반 문장 유사도 행렬을 생성하였고 두 유사도 행렬을 합 연산하여 최종 유사도 행렬을 도출하였다. 유사도 행렬은 행과 열에 문장 번호가 순차적으로 반영되어 있다. 이 유사도 행렬의 값을 행 또는 열을 기준으로 합산할 경우 각 문장별로 유사도 총 합이 계산되며 이 값이 가장 큰 문장은 해당 뉴스기사를 대표할 수 있는 문장으로 볼수 있고 값이 작을수록 뉴스기사 내에서 관련도가 낮다고 볼수 있다.

문장 간의 유사도는 그림 6과 같은 행렬 형태로 나타낼 수 있다. 가로 축과 세로 축은 각각 문장의 번호를 나타내며 같은 숫자인 경우 문장이 같으므로 유사도가 1로 나타난다. 그림 6의 세로 축은 유

사도 총 합을 기준으로 정렬한 상태이다.

그림 6의 행렬은 “휴대폰의 지원금 상향에도 구매가 어렵다”는 기사의 행렬이며 표 5에서 0, 7, 4, 12, 2, 15번째 문장의 내용을 확인 할 수 있다. 0과 7은 상위권에 포함되어 뉴스기사를 대표할 수 있는 문장이며 중위권인 4, 12는 지원금이나 구매 둘 중 하나의 주제를 나타내고 하위권인 2, 15문장은 뉴스기사 내에서 관련성이 떨어지는 문장이다.

	0	1	2	3	4	12	13	14	15	Total
0	1.000	0.138	0.000	0.173	0.128	0.044	0.211	0.043	0.080	2.383
7	0.259	0.202	0.000	0.137	0.000	0.065	0.205	0.000	0.000	2.317
3	0.173	0.103	0.000	1.000	0.263	0.000	0.158	0.128	0.000	2.310
13	0.211	0.155	0.000	0.158	0.085	0.000	1.000	0.049	0.000	2.087
6	0.177	0.000	0.000	0.088	0.143	0.072	0.078	0.082	0.000	2.042
1	0.138	1.000	0.000	0.103	0.000	0.098	0.155	0.050	0.044	1.933
5	0.130	0.143	0.000	0.148	0.000	0.000	0.146	0.000	0.000	1.915
4	0.128	0.000	0.000	0.263	1.000	0.000	0.085	0.097	0.000	1.907
12	0.044	0.098	0.047	0.000	0.000	1.000	0.000	0.020	0.000	1.505
14	0.043	0.050	0.000	0.128	0.097	0.020	0.049	1.000	0.000	1.494
11	0.000	0.000	0.057	0.000	0.000	0.077	0.000	0.025	0.000	1.413
10	0.000	0.000	0.000	0.000	0.191	0.000	0.000	0.000	0.000	1.338
9	0.000	0.000	0.000	0.112	0.000	0.000	0.000	0.000	0.000	1.328
8	0.000	0.000	0.078	0.000	0.000	0.082	0.000	0.000	0.000	1.200
2	0.000	0.000	1.000	0.000	0.000	0.047	0.000	0.000	0.000	1.182
15	0.080	0.044	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.124

그림 6. 문서 내 문장 유사도 행렬
Fig. 6. Sentence similarity matrix

표 5. 문장과의 관련도 상중하 각 2개 문장
Table 5. Relationship with other sentences

Relevance	Sentence number	Sentence content
The top two	0	High mobile operators ′ support fee adjustment it is difficult to purchase smartphones
	7	However, it is not easy to buy a smartphone
The middle two	4	Telecommunication companies also raised the subsidy at the end of last month
	12	A source at a mobile network operator said, "The amount of remaining money was raised by the service operators to exhaust the amount of the remaining amount after the intercompany sales adding It would be difficult to buy it from an ordinary customer"
The lower two	2	The market is short of stock.
	15	This news item can not be distributed in its entirety without permission.

본 논문에서는 불필요 문장을 삭제해 고려해야하는 단어의 수를 줄이고 전체 뉴스기사 분석에 있어서 빅 데이터를 필요한 부분만 취해 효율적인 분석을 수행하고자 하였다. 문장 삭제는 각 뉴스기사의 길이 차이를 반영하기 위해 문장 수에 대한 퍼센트 비율을 설정해 제거할 문장 수를 계산하였다.

3.3 키워드 추출

뉴스기사 데이터에 대한 전처리와 뉴스기사 상에서 관련성이 떨어지는 문장을 제거하고 TF-IDF, 의미벡터, 결합 벡터에 대해서 각각 15개의 키워드를 추출하였다. 의미벡터는 3.3절의 문장 유사도 측정 부분에서 다룬 단어 간 유사도 방법을 활용하여 계산된다. 전체 뉴스기사에서 나타난 모든 단어들을 벡터공간에 표현하기 위해 총 단어 수 만큼의 차원 벡터를 생성한다. 다음으로 특정 뉴스기사에 포함된 단어들을 수치화하여 벡터공간에 표현하게 되는데 현재 단어와 뉴스기사에 포함된 단어들 간의 유사도 평균을 벡터 값으로 설정한다. 이를 통해 문서 내에서 다른 단어들과의 의미적 연관성이 높은 단어가 높은 값을 받게 된다. 이 벡터를 본 논문에서는 의미벡터라 지칭하며, TF-IDF 벡터와 의미벡터를 결합한 결합 벡터와 벡터화 방법으로 제안한다. 그림 7의 3단계를 통해 최종 의미벡터 키워드가 추출된다.

키워드 추출에 적용하는 의미벡터는 단어 간의 유사도를 측정하는 방식이 3.3절의 의미기반 문장 유사도 계산 방식과 같지만 유사도를 비교하는 대상의 차이가 있다.

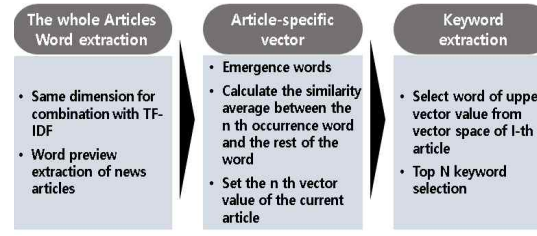

그림 7. 의미기반 벡터화 및 키워드 추출 프로세스
Fig. 7. Semantic-based vectorization and keyword extraction process

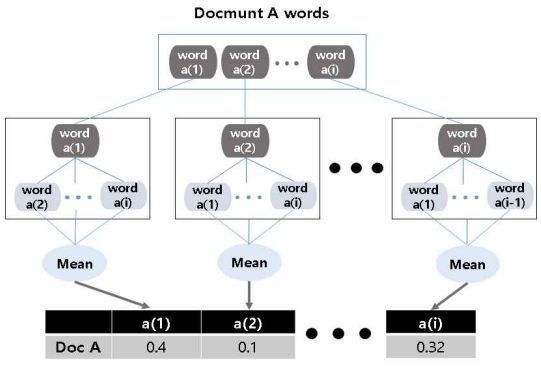

그림 8. 문서 A에서의 의미벡터 생성 과정
(단어별 유사도 측정)
Fig. 8. Semantic vector generation process in document A
(Measure similarity by word)

3.3절에서는 불필요 문장 추출을 위해 각 문장에서 나타나는 단어들 간의 유사도를 측정한 반면, 뉴스기사 벡터화에서는 뉴스기사 내에 나타난 전체 단어들 간에 유사도를 측정해 각 단어가 뉴스기사에 출현한 단어들과 얼마나 가까운 의미를 갖는지를 측정한다. 또한, 이를 통해 각 단어에 대해서 계산된 최종 유사도 값이 각 단어의 벡터 값이 된다.

그림 8은 문서 A의 단어 집합을 벡터공간에 나타내기까지의 과정을 나타냈으며 단어 a1과 a2부터 ai까지 단어들과의 유사도를 측정하고 값들의 평균을 취해 a1의 의미적 벡터 값을 생성해 벡터 공간에 입력하였다. 설명 편의상 벡터공간에는 문서 A의 단어 집단만을 표시했으나 실제로는 타 문서에 나타난 단어들도 한 벡터 공간에 모두 표시되며 문서 A에서 나타나지 않은 단어는 0의 값을 갖게 된다.

본 연구는 그림 9와 같이 전체 뉴스기사 집합으로부터 통계, 의미벡터를 생성하고 두 벡터를 결합한 결합벡터를 도출한다. 이후 각 벡터로부터 높은 수치를 갖는 단어 15개를 추출하여 키워드로 선정한다.

전체 뉴스기사 집합을 구성하는 단어를 추출 및 카운트하여 벡터화를 위한 사전 준비를 수행하였다. 최종 결합 벡터 산출을 위해 통계벡터와 의미벡터는 동일한 차원으로 구성하였으며 벡터의 각 열은 전체 뉴스기사에 출현한 단어들을 나타내며 행은 개별 뉴스기사로 설정하였다. 그림 10에서 이를 도식화 하였다.

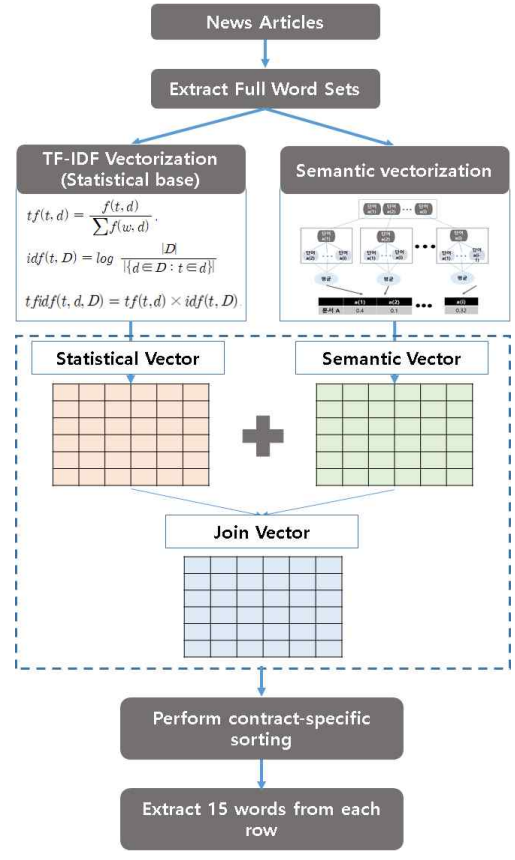

그림 9. 결합벡터 키워드 추출 프로세스
Fig. 9. Join vector keyword extraction process

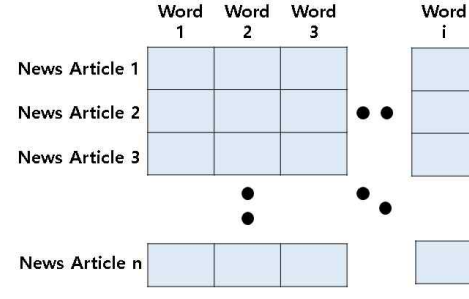

그림 10. 벡터 공간의 행, 열 의미
Fig. 10. Rows and columns of vector space

TF-IDF의 경우 정의된 수식을 활용하였으며 의미벡터는 본 논문에서 정의한 그림 9의 방법을 통해 뉴스기사 별로 벡터공간에 단어들의 점수를 설정하였다. 통계적 방법을 적용하는 TF-IDF는 뉴스기사에 출현하지 않는 단어는 자연스럽게 0으로 계산되도록 수식이 정의되어 있다.

하지만, 본 논문에서 제안한 의미벡터 생성 방법에서는 벡터공간의 모든 단어를 특정 단어와 비교하는 방식이기 때문에 실제 뉴스기사 내에서 출현하지 않은 단어는 0으로 설정하고 단어간 유사도 비교 시 고려하지 않도록 해야 한다. 따라서, 본 논문에서는 n번째 뉴스기사에 출현한 단어들을 목록화하여 해당 목록에 나타난 단어들간의 유사도를 측정하여 뉴스기사에 출현한 단어들이 올바른 의미값을 가질 수 있도록 하였다. 이후, 같은 차원의 통계벡터와 의미벡터를 합 연산하여 결합벡터를 생성하였다.

마지막으로 현재 벡터공간에는 뉴스기사 별로 단어들의 값이 순서 없이 배정되어 있기 때문에 높은 값을 갖는 단어들을 선택해야 한다. 이를 위해, 벡터의 뉴스기사 별 단어 가중치 값 정렬을 수행하였으며 가장 높은 15개의 단어를 해당 뉴스기사의 키워드로 선정하였다. 표 6은 선택된 뉴스기사의 벡터별 키워드 목록을 나타낸다.

기사내용은 AI, VR, AR, 등 신기술 혁명 가속에 관한 내용을 포함하고 있다. 표 6을 통해서 TF-IDF 벡터 기반 키워드는 ‘ar’, ‘it’, ‘앱’, ‘ai’ 등 IT 분야의 용어들이 주로 추출되었으며 의미벡터 기반 키워드는 ‘가속’, ‘경쟁’, ‘관리’등 상황이나 상태를 나타내는 키워드와 다양한 문서에서 나타날 수 있는 일반적인 키워드가 상당수 추출되었다. 결합 벡터는 두 벡터 기반 키워드의 내용이 섞여있는 형태로 나타났다. 문서의 핵심 키워드는 TF-IDF의 키워드와 같이 특정 기술 명을 나열하는 것으로도 설명할 수 있지만, 키워드 목록이 다양한 정보를 나타낼 수 있도록 의미벡터와의 결합방법을 제안한다.

표 6. 뉴스기사로부터 추출된 벡터 별 키워드
Table 6. Keywords by vector extracted from news articles

Vector	Keywords by vector
TF-IDF	ar, secretary, it, ai, sns, vision, device, app, virtual, forecast, forecast, forecast, voice, fake
Semantic vector	experience, global, relevance, domestic, google, secretary, competition, institution, device, game, acceleration, deployment, personal, anagement, management,
Join Vector	ar, secretary, device, experience, virtual, vision, global, national, international, national, air, competition, trend, forecast, agency

TF-IDF 키워드에서는 AI, IoT, IT, 디스플레이, 텔레콤 등 IT 분야의 기술 단어들이 주로 출현하였다. 의미 기반 벡터 키워드에서는 IT분야 키워드보다 가능성, 개선, 개척 등과 같이 기술의 상황을 판단할 수 있는 단어들이 주로 출현하였다. 앞의 표 6과 같이 결합 벡터 키워드는 AI, AR, 강점, 게임 등 앞의 두 벡터 키워드의 혼합 형태로 나타났다. 통계, 의미벡터를 단일로 사용했을 경우 한쪽으로 치우쳐진 결과가 나타났으며 이러한 결과를 결합 벡터를 통해 상호보완 시킴으로써 통계, 의미 양쪽의 키워드가 고르게 추출될 수 있도록 하였다.

3.4 키워드 기반 뉴스기사 검색 및 연관 뉴스기사 추천

인터넷상에서는 문서의 제목이나 내용을 기반으로 검색과 연관 문서를 찾아주는 경우가 많으며 로그 기반의 키워드 기반 연구도 수행되고 있다[17].

이에 본 논문에서는 뉴스기사 별로 키워드를 추출하여 해당 키워드를 검색어와 비교하고 유사하거나 같은 단어일 경우 해당 뉴스기사를 검색 결과로 선택한다. 첫번째로 뉴스기사의 키워드 목록과 검색어 간의 일치성을 확인하고 두 번째로 키워드 목록과 검색어 간의 의미적 유사도를 측정해 검색 결과를 도출 하였다.

연관 뉴스 기사 추천은 기존에 뉴스기사 전체를 벡터화하고 코사인 유사도 등의 유사도 계산 방법에 적용하여 찾아내는 방식이 활용되고 있으나, 지속적으로 증가하는 뉴스기사들을 첫 번째부터 마지막까지 비교하는 것은 시스템 자원적으로 비효율적이다. 따라서 결합벡터를 통해 추출된 키워드 목록을 벡터화하고 각 뉴스기사별로 키워드 벡터를 비교하여 연관 뉴스기사를 찾아내었다.

IV. 성능 평가

평가지표는 정밀도(Precision)와 재현율(Recall), F1-Score, NDCG 4가지 지표를 이용해 성능을 평가하였다. 정밀도와 재현율은 키워드에 따른 검색결과 리스트와 실제 정답결과의 리스트가 있어야지만 평

가가 가능하기 때문에 검색 키워드 목록을 선정하고 해당 키워드에 대한 정답 뉴스기사 리스트를 생성해 성능을 평가하였다. F1-Score는 정밀도와 재현율의 조화 평균값을 나타내며 최종적으로 F1-Score가 높은 경우 성능이 좋다고 평가한다. 수식 7를 통해 F1-Score를 얻을 수 있다.

$$F1-Score=2\times\frac{precision\times recall}{precision+recall} \quad (7)$$

키워드를 이용해 검색과 관련기사가 효과적으로 도출 되는 것을 목표로 하기 때문에 키워드 자체의 평가가 아닌 키워드를 활용했을 때 결과를 이용해 역으로 키워드의 성능을 평가하였다. 우선, 검색어 10개를 무작위로 선정하여 각 키워드에 해당하는 뉴스기사를 여러 사용자의 평가를 통해 선택하였다.

표 7. 벡터별 정밀도 및 재현율 결과
Table 7. Accuracy and recall results by vector

answer article	Delete sentence (%)	TF-IDF		Semantic vector		Join Vector	
		Precision	Recall	Precision	Recall	Precision	Recall
1	0	1.0	0.8	1.0	0.3	0.36	0.3
	3	0.95	0.7	0.94	0.29	0.37	0.31
	5	0.93	0.7	0.92	0.27	0.30	0.29
2	0	0.82	0.47	0.8	0.63	0.51	0.64
	3	0.75	0.34	0.71	0.60	0.49	0.63
	5	0.76	0.4	0.66	0.57	0.49	0.56
3	0	0.52	0.31	0.50	0.66	0.49	0.6
	3	0.50	0.17	0.47	0.66	0.49	0.71
	5	0.49	0.21	0.45	0.63	0.48	0.58

표 8. 벡터 별 F1-Score 결과
Table 8. F1-Score results by vector

answer article	Delete sentence (%)	TF-IDF	Semantic vector	Join Vector
		F1-Score		
1	0	0.88	0.46	0.33
	3	0.81	0.45	0.34
	5	0.80	0.42	0.30
2	0	0.60	0.70	0.57
	3	0.47	0.65	0.55
	5	0.52	0.61	0.52
3	0	0.39	0.57	0.54
	3	0.26	0.55	0.58
	5	0.30	0.52	0.52

사용자가 선택한 뉴스기사는 정답 뉴스기사로 설정한다. 또한, 해당 검색어를 이용해 키워드 기반 검색을 통해 나온 뉴스기사와의 일치도를 통해 성능을 평가하였다.

표 7은 뉴스기사 선택과 문장삭제에 따른 벡터별 정밀도, 재현율을 계산한 결과이다.

뉴스기사 선택방법은 3단계로 정의하였으며 3에 가까울수록 선택되는 뉴스기사의 기준을 엄격하게 하였다. 문장 삭제는 원본 뉴스기사와 타 문장과의 연관도가 낮은 0에서 5%에 해당하는 문장을 제거한 후 키워드를 추출해 키워드 기반으로 검색어에 해당하는 뉴스를 선정하였다. 정밀도와 재현율이 상반되는 형태로 나타나고 정확한 판단이 어려워 F1-Score를 이용해 정확한 성능을 측정하였다. 표 8은 표 7의 정밀도와 재현율에 따른 F1-Score를 계산한 결과이며 이를 통해 정확한 성능을 파악할 수 있었다.

TF-IDF와 결합 벡터의 결과는 서로 상반되는 형태를 보였는데 TF-IDF는 정답 뉴스를 낮은 기준으로 선택하고 문장 삭제를 적게 했을 때 높은 점수를 보였고 결합 벡터는 반대의 성향을 보였다.

TF-IDF는 정답 기사 선택에 있어서 엄격한 기준을 취해 나온 정답 셋에 대해서는 관련성이 낮은 문장을 삭제하여도 성능의 감소를 보였는데 이는 단어의 빈도수에 의존하기 때문에 단어가 적게 나타나면 해당 문서의 나머지 내용과는 상관없이 관련 없는 뉴스기사로 분류하게 된다는 것을 나타낸다. 반면에 결합 벡터의 경우 문장삭제 수에 따른 영향이 작게 나타났으며 이를 통해 내용이 짧거나 핵심 문장으로만 구성된 뉴스기사도 TF-IDF에 비해 적절히 선택할 수 있다는 것을 유추할 수 있었다.

그림 11은 검색어의 포함여부만을 이용해 구축한 정답기사 1에 대한 성능 그래프이다.

그림 11에 따르면 TF-IDF의 성능이 아주 높게 측정되었음을 알 수 있으나 이는 단순히 검색어가 포함된 경우를 정답셋으로 구축하였을 때의 성능이다. 따라서 보다 정밀한 정답셋에 대한 성능 평가를 실시하였을 때 그림 12와 같은 성능 결과가 나타났다.

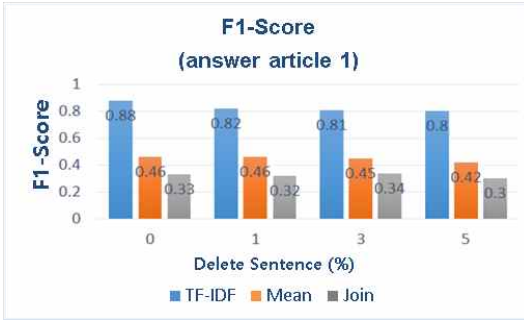

그림 11. 정답기사 1에 대한 키워드 검색 성능
Fig. 11. Keyword search performance for correct answer article 1

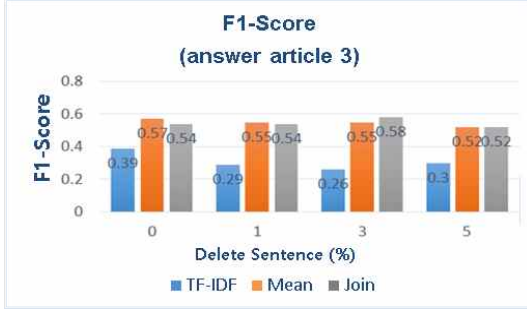

그림 12. 정답기사 3에 대한 키워드 검색 성능
Fig. 12. Keyword search performance for correct answer article 3

그림 12를 통해 의미 요소가 반영되었을 때, 성능이 높은 것을 확인 할 수 있다. 의미 벡터와 결합벡터는 성능이 비슷하지만 뒤의 NDCG 성능 결과를 통해 가장 효과적인 벡터를 도출한다.

마지막으로 선택된 기사의 키워드 목록을 벡터화하여 연관 뉴스기사 추천의 성능을 평가하였다. 본 성능평가에서는 키워드 기반 연관 뉴스기사 목록과 문서 벡터 기반 연관 뉴스기사 목록의 NDCG 점수를 산출해 성능을 비교하였다. NDCG 계산을 위해 선택된 뉴스기사의 품질을 나타내는 rel 등급은 0, 1, 2, 3 네 가지 수로 나타내며 0은 관련성이 없는 문서, 1은 약간의 관련성이 있는 문서 2는 보통의 관련성이 있는 문서 3번은 매우 연관성이 높은 문서를 나타낸다. 추천된 연관 뉴스는 유사도에 따라 랭킹화되기 때문에 정밀도나 재현율이 아닌 랭킹 기반 성능지표를 활용하였다. 문서 벡터와 각 키워드 기반의 연관기사 추천을 통해 표 9의 성능을 계산하였다.

표 9. 벡터 별 연관 기사 추천 성능 계산 결과
Table 9. Recommended performance calculation results

Delete sentence (%)	Document vector	TF-IDF	Semantic vector	Join Vector
0	0.97	0.96	0.81	0.96
1	0.95	0.90	0.99	0.98
3	0.95	0.91	0.89	0.98
5	0.99	0.94	0.98	0.97

다수 문서에 대해서 NDCG를 계산하여 평균값을 취하였다. 문서 벡터는 많은 단어를 가지고 연관도를 분석하기 때문에 예상대로 높은 성능이 나왔으며 TF-IDF와 의미벡터 키워드는 결과 값의 변동이 심하고 상대적으로 연관 기사를 잘 못 찾는 경향이 있었다. 마지막으로 결합 벡터 키워드는 문장삭제 여부와 상관없이 고른 성능을 보였으며 모든 단어들의 집합을 사용하는 문서 벡터를 대신하여 연관기사 추천에 활용 될 수 있을 것으로 판단된다. 또한, NDCG 결과와 앞의 F1-Score를 함께 고려하였을 때 결합벡터가 검색과 연관뉴스 추천에서 가장 효과적임을 알수 있다.

V. 결론 및 향후 과제

본 논문에서는 기존 TF-IDF에 단어 유사도를 활용한 의미벡터를 결합하여 뉴스기사를 벡터화하고 그로부터 뉴스기사의 주요 키워드를 추출하였다. 또한, 추출한 키워드를 이용하여 뉴스 검색과 연관 뉴스 추천 연구를 수행하였다. TF-IDF의 경우 단어 빈도수 기반의 벡터모델로 통계 기반의 모델이기 때문에 의미적인 요소가 반영되기 어렵다. 단순히, 서로 비슷한 횟수로 나타난 단어들이 유사하다고 볼 수밖에 없다. 따라서 전체 문서 집합에서 같은 문장에 나타나거나 이웃한 단어의 경우 단어 간에 유사성이 있다는 가정을 통해 수집한 뉴스기사 대상으로 Word2vec 모델을 학습하여 단어 간 유사도를 측정한다. 하나의 단어와 같은 문서 내에 등장한 단어들 간에 유사도를 측정해 해당 단어가 문서 내에서 다른 단어들과 얼마나 유사한지를 벡터 공간에 표현함으로써 의미벡터를 생성하여 활용하였다. TF-IDF와 의미벡터를 결합한 결합 벡터는 문서의 통계, 의미 적인 부분을 모두 내포하고 있는 벡터이

며 키워드 분포를 통해 두 벡터보다 효과적인 키워드 추출이 가능한 것으로 판단하였다.

정확한 성능평가를 위해 각 뉴스기사에 대해 키워드 추출을 수행하였으며 단일 검색어에 대한 질의응답을 수행한 결과 결합 벡터가 TF-IDF보다 효과가 있음을 보였고 연관 뉴스 탐색에서도 결합벡터의 성능이 높게 측정 되었다.

인터넷 뉴스기사 데이터를 활용하여 뉴스기사 추천 방법에 대해 연구함으로써 인터넷 상에 존재하는 다른 문서집합들에 대해서도 같은 방법의 연구를 수행할 수 있음을 보였다.

본 논문에서 제시한 의미벡터는 단독으로 사용되었을 때는 저조한 성적을 보였으나 TF-IDF 모델과 결합해 결합 벡터를 생성함으로써 TF-IDF에서는 찾지 못한 키워드를 반영 할 수 있었으며 불필요 문장 제거 기법과 결합해 뉴스기사 검색에서 준수한 성능을 보였다. 또한, 연관 기사 추천에서는 다른 벡터보다 안정적으로 연관 기사를 추천하였으며 이에 따라, 무조건적인 빅데이터 적용이 아닌 적절한 데이터 가공을 통해 상황에 따라 요약된 문서 데이터로도 좋은 분석 성능을 낼 수 있음을 보였다.

본 논문에서는 인터넷 뉴스 기사 중 IT 분야 뉴스기사에 대해서만 연구를 수행하였기 때문에 타 웹문서나 분야가 다른 뉴스기사에 대한 검증이 필요하다. 이러한 이유로 향후 과제로 분야별 뉴스기사와 논문 데이터를 수집해 본 연구의 확장가능성을 검증하고 다양한 문서 데이터를 학습해 유사도 측정 모델의 성능을 향상시킴으로써 고성능 결합 벡터 개발이 가능할 것이다. 끝으로 검색과 관련 문서 추천에 집중한 본 연구를 확장하여 사용자의 웹 문서 사용 이력 정보를 수집하여 키워드 기반의 맞춤형 문서 추천 연구를 수행하여 웹을 이용하는 사용자의 편의성 향상에도 기여하고자 한다.

References

[1] Yongjae Im, Sunkyoung Baek, and Seungjun, Yeon, "Select and focus on securing competitiveness in the Big Data era", The Korean Institute of Communications and Information Sciences, Vol. 29, No. 11, pp. 3-10, Oct. 2012.

[2] Jieun Son, Seoungbum Kim, Hyunjoong Kim, and Sungzoon Cho, "Review and Analysis of Recommender Systems", Journal of the Korean Institute of Industrial Engineers, Vol. 41, No. 2, pp. 185-208, Apr. 2015.

[3] Yongsoo Kim, "Research Trend of Recommendation System for Personalized Service", The Korean Institute of Industrial Engineers ie Magazine, Vol. 19, No. 1, pp. 37-42, Mar. 2012.

[4] Jiyeon Kim, "Internet Search Engine : Technological Mode that Draws User's Attention to Make Its Expertise Reinforce", Journal of Science & Technology Studies, Vol. 13, No. 1, pp. 181-216, Jun. 2013.

[5] Yangjung Ae, "News Story Salience and Users' Selective Exposure - Effects of Popularity Indications on Online News Exposure", Korean Journal of Broadcasting, Vol. 25, No. 2, pp. 264-288, Mar. 2011.

[6] Sunghae Jun, "Big Data Preprocessing using Statistical Text Mining", Journal of Korean Institute of Intelligent Systems, Vol. 25, No. 5, pp. 470-476, Oct. 2015.

[7] Daemin Park, "Natural Language Processing of News Articles : A Case of <NewsSource beta>", Communication Theories, Vol. 12, No. 1, pp. 4-52, Mar. 2016

[8] Manning, C. D, Raghavan, and P, Schutze, "HIntroduction to Information Retrieval", Cambridge University Press, pp. 100-123, 2008

[9] Dik L. Lee, Huei Chuang, and Kent Seamons, "Document Ranking and the Vector-Space Model", IEEE Software, Vol. 14, No. 2, pp. 67-75, Mar. 1997.

[10] Euisok Chung and Jeon-Gue Park, "Class Language Model based on Word Embedding and POS Tagging", KIISE Transactions on Computing Practices, Vol. 22, No. 6, pp. 315-319, Jul. 2016.

[11] Eunjeong Park and Sungzoon Cho, "KoNLPy: Korean natural language processing in Python",

Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology, pp. 133-136, Oct. 2014.

[12] Radim Rehurek and Petr Sojka, "Software framework for topic modelling with large corpora", The LREC 2010 Workshop On New Challenges For NLP Frameworks, pp. 45-50, May. 2010.

[13] Young-Bin Kwon, Seoung-Do Lee, Hyun Yang, and Yo-Han Joo, "The Analysis of the Conferences for the Computer Network Using the Miner and the Cosine Similarity based upon Keywords", The Korea Society of Information Technology Services, Vol. 11, No. 1, pp. 223-238, Mar. 2012.

[14] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu, "A Theoretical Analysis of NDCG Ranking Measures", Workshop and Conference Proceedings, pp. 1-30, Apr. 2013.

[15] Sungjick Lee and Hanjoon Kim, "Keyword Extraction from News Corpus using Modified TF-IDF", Society for e-Business Studies, Vol. 14, No. 4, pp. 59-73, Nov. 2009.

[16] Seunghee Han, "A Study on Keyword Extraction From a Single Document Using Term Clustering", Journal of the Korean Society for Library and Information Science, Vol. 44, No. 3, pp. 155-173, Aug. 2010.

[17] KilHong Joo, JooIl Lee, and WonSuk Lee, "An Associated Keywords Extraction and a Spread Clustering Methods for an Efficient Document Searching", Journal of KIIT, Vol. 9, No. 6, pp. 155-166, Jun. 2011.

저자소개

박 대 서 (Dae-Seo Park)

2015년 8월 : 강원대학교 컴퓨터정보통신공학과(공학사)
2016년 3월 ~ 현재 : 강원대학교 컴퓨터정보통신공학과(석사과정)
관심분야 : 데이터분석, 머신러닝, 딥러닝, 데이터공유 플랫폼

김 화 종 (Hwa-Jong Kim)

1982년 : 서울대학교 전자공학과 (공학사)
1984년 : KAIST 전기 및 전자과 (공학석사)
1988년 8월 : KAIST 전기 및 전자과(공학박사)
1988년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 교수
관심분야 : 데이터공유, 데이터분석, 인공지능, 머신러닝, 딥러닝