https://doi.org/10.5392/JKCA.2020.20.06.556

준 실시간 뉴스 이슈 분석을 위한 계층적·점증적 군집화

Hierarchical and Incremental Clustering for Semi Real-time Issue Analysis on News Articles

김호용*, 이승우*, 장홍준**, 서동민*
과학기술연합대학원대학교(UST)/한국과학기술정보연구원(KISTI)*, 한국과학기술정보연구원(KISTI)**

Hoyong Kim(khyong0101@ust.ac.kr)*, SeungWoo Lee(swlee@kisti.re.kr)*,
Hong-Jun Jang(hongjunjang@kisti.re.kr)**, DongMin Seo(dmseo@kisti.re.kr)*

요약

실시간으로 발생하는 뉴스 기사로부터 이슈를 분석하기 위한 다양한 연구가 진행되어 왔다. 하지만 범주에 따라 계층적으로 이슈를 분석하는 연구는 많이 진행되지 않았고, 계층적 이슈 분석을 위한 기존의 연구에서 제안하는 방식 또한 뉴스 기사 증가에 따라 군집화 속도가 느려지는 문제점이 있다. 따라서 본 논문에서는 준 실시간으로 뉴스 기사의 이슈를 분석하는 계층적·점증적 군집화 방식을 제안한다. 제안하는 군집화 방식은 샴 신경망을 이용한 가중 코사인 유사도 측정 모델 기반의 k-평균 알고리즘을 이용한 단어 군집 기반 문서 표현 방식을 통해 뉴스 기사를 문서 벡터로 표현한다. 그리고 문서 벡터로부터 초기 이슈 군집 트리를 생성하고, 새로 발생한 뉴스 기사를 해당 이슈 군집 트리에 추가하는 점증적 군집화 방식을 제안함으로써 뉴스 기사의 계층적 이슈를 준 실시간으로 분석한다. 마지막으로, 본 논문에서 제안하는 방식과 기존 방식들과의 성능 평가를 통해 제안하는 군집화 방식이 정확도 측면에서 기존 방식 대비 NMI 지표 기준 0.26 정도 성능이 향상되었고, 속도 측면에서 약 10배 이상의 성능이 향상됨을 입증하였다.

■ 중심어 : | 뉴스 기사 | 계층적 군집화 | 점증적 군집화 | 가중 코사인 유사도 | 샴 신경망 |

Abstract

There are many different researches about how to analyze issues based on real-time news streams. But, there are few researches which analyze issues hierarchically from news articles and even a previous research of hierarchical issue analysis make clustering speed slower as the increment of news articles. In this paper, we propose a hierarchical and incremental clustering for semi real-time issue analysis on news articles. We trained siamese neural network based weighted cosine similarity model, applied this model to k-means algorithm which is used to make word clusters and converted news articles to document vectors by using these word clusters. Finally, we initialized an issue cluster tree from document vectors, updated this tree whenever news articles happen, and analyzed issues in semi real-time. Through the experiment and evaluation, we showed that up to about 0.26 performance has been improved in terms of NMI. Also, in terms of speed of incremental clustering, we also showed about 10 times faster than before.

■ keyword : | News Articles | Hierarchical Clustering | Incremental Clustering | Weighted Cosine Similarity | Siamese Neural Networks |

* 본 연구는 한국과학기술정보연구원의 「국가 연구데이터 공유·확산체제 구축(K-20-L01-C04)」사업으로부터 지원받아 수행된 연구임.

접수일자 : 2020년 05월 25일
수정일자 : 2020년 06월 08일
심사완료일 : 2020년 06월 08일
교신저자 : 서동민, e-mail : dmseo@kisti.re.kr

I. 서 론

최근 IT업계에서 주목받고 있는 빅데이터(Big-Data) 기술은 대용량 데이터를 활용하고 이를 분석하여 가치 있는 정보를 추출하고, 이를 바탕으로 사회적 문제를 해결하거나 문제 상황을 예측하기 위한 정보화 기술을 말한다. 빅데이터 분석을 위하여 이미지, 영상, 텍스트 중 다양한 형태의 데이터들이 활용되는데, 그 중 텍스트 데이터를 활용하여 이슈를 분석하는 이슈 분석(Issue Analysis) 기술이 있다. 이슈 분석은 각종 포털 사이트에서 실시간으로 발생하는 사회적 이슈를 파악하기 위해 사용된다. 일례로, 빅카인즈(BIG KINDS)[1]는 다양한 언론사로부터 수집한 뉴스로 구성된 통합 데이터베이스에 빅데이터 분석 기술을 접목한 새로운 뉴스 분석을 제공해주는 사이트로, 매일 발생하는 뉴스 기사로부터 키워드를 추출하여 이슈를 분석한다.

텍스트 데이터로부터 이슈를 분석하기 위해서 여러 가지 방식이 사용되는데, 그 중 문서 표현(Document Representation) 방식과 군집화(Clustering) 방식이 있다. 텍스트 데이터에 군집화 알고리즘을 적용하기 위해서는 텍스트 데이터를 벡터로 변환해주는 과정이 필요하다. 문서 표현 방식은 이러한 텍스트 데이터로 이루어진 문서를 벡터로 표현하기 위하여 사용된다. 군집화 방식은 벡터로 표현된 문서들을 비슷한 문서들끼리 묶어 군집을 생성하기 위하여 사용된다.

이러한 문서 표현 방식과 군집화 방식을 사용하여 이슈를 분석하는 다양한 연구가 진행되어 왔다. P. Zhou, Z. Cao, B. Wu, C. Wu 그리고 S. Yu[2]는 문서 내 단어의 빈도수를 이용한 TF-IDF 방식을 수정한 JS-ID'F_{order} 방식을 제안하였고, 이를 이용하여 문서의 자질(feature)을 추출하고 문서를 벡터로 표현한 다음, Bikmeans 알고리즘[3]이라는 k-평균 알고리즘(k-means algorithm)에 양방향적인(Bidirectional) 특성을 추가한 알고리즘을 사용하여 군집을 생성함으로써 뉴스 이슈 분석 문제를 해결하였다. L. Hu, B. Zhang, L. Hou, 그리고 J. Li[4]의 연구에서는 Skip-gram 모델과 k-평균 군집화(k-means clustering)를 이용하여 뉴스 문서를 벡터로 표현하였다. 군집화 과정에서는 먼저 뉴스 기사들을 일정 시간 단위로 나누어 집합을 생성하고, 해당 집합들에 대하여 싱글 패스 온라인 군집화(Single-pass Online Clustering) 방식을 이용하여 집합 내 문서들을 이슈 단위로 재군집하였다. 이후 이벤트 병합(Event Merging)과 적응형 후처리 과정(Adaptive Post-processing)을 거쳐 뉴스 기사로부터 이슈를 분석하는 어댑티브 온라인 군집화(Adaptive Online Clustering) 방식을 제안하였다. 유홍현[5]의 연구에서는 TF-IDF를 기반으로 하는 Okapi BM25 알고리즘을 이용하여 뉴스 기사들을 원-핫 벡터(one-hot vector)로 표현한 다음, LDA를 이용하여 계층적 다중 뉴스 군집을 형성하고, 일정 단위 시간 동안 수집된 뉴스 기사들에 대하여 재군집을 진행하는 점증적 다중 뉴스 군집화 과정을 거침으로서 이슈를 계층적으로 분석하였다. 유홍현의 연구에서 제안한 군집화 방식은 기존의 이슈 분석 방식과는 다르게 계층적 다중 뉴스 군집을 생성함으로써 넓은 범주의 이슈와 좁은 범주의 이슈를 계층적으로 표현하였다. 하지만, TF-IDF 기반 문서 표현 방식을 사용하여, 뉴스 기사가 증가함에 따라 단어의 개수가 증가하고, 결과적으로 문서 벡터의 크기가 커져 메모리 비효율성 문제가 발생하였으며, 이에 따라 군집화 속도가 느려지는 현상이 나타났다.

본 연구에서는 유홍현[5]의 연구에서 제안한 계층적 다중 뉴스 군집 생성 과정 및 점증적 군집화 방식과 뉴스 기사의 시간적 특성을 활용한 어댑티브 온라인 군집화 방식을 융합하여 준 실시간 뉴스 이슈 분석을 위한 계층적·점증적 군집화 방식을 제안한다. 특히, 어댑티브 온라인 군집화 방식의 단어 군집 기반 문서 표현을 통해 뉴스 기사 증가에 따른 문서 벡터의 크기 증가 문제를 해결한다. 또한 워드 임베딩 모델 선정과 가중 코사인 유사도 측정 모델 개발 실험을 진행하고 이를 적용함으로써 제안하는 군집화 방식의 계층적 이슈 분석 정확도를 높인다. 마지막으로, 점증적 군집화 실험을 통해 속도 측면에서 기존 군집화 방식과 제안하는 군집화 방식의 이슈 군집 트리 업데이트 소요 시간을 비교한다.

본 논문의 구성은 다음과 같다. II절에서는 본 연구와 관련된 기존 연구들에 대하여 기술한다. 먼저, II의 1절에서는 본 연구의 문서 표현을 위해 사용되는 단어 임베딩 모델과 단어 군집 기반 문서 표현 방식에 대하여,

그리고 II의 2절과 3절에서는 기존의 군집화 알고리즘과 가중 유사도 측정 모델에 대하여 구체적으로 기술한다. III절에서는 본 논문에서 제안하는 군집화 방식에 대하여 자세히 기술하고, IV절에서는 군집화 정확도 성능 개선을 위한 실험들과 제안하는 군집화 방식에 대한 정확도와 속도 실험에 대하여 설명하고 실험 결과를 분석한다. 마지막으로 V절에서는 본 연구에 대한 결론을 제시한다.

II. 관련 연구

1. 기존 문서 표현 방식

문서를 벡터로 표현하는 가장 기본적인 방법은 bag-of-words를 만들고 이를 이용하여 문서를 원-핫 벡터로 표현하는 것이다. 하지만 이러한 문서 표현 방법은 문서의 개수가 증가함에 따라 단어의 개수가 증가함으로써 벡터의 차원이 증가하고, 이는 희소성(sparsity) 문제를 발생시켜 메모리가 낭비된다는 단점이 있다. 이러한 문제를 해결하기 위하여 최근에는 딥 러닝(Deep Learning) 모델을 이용한 워드 임베딩(word embedding) 방식에 대한 연구가 진행되었다. 워드 임베딩이란 자연어처리(Natural Language Processing; NLP)에서 단어를 실수형 벡터로 표현하는 방식으로, 특정 단어의 주변 단어를 예측하는 딥 러닝 모델로부터 단어 벡터를 학습하는 Skip-gram[6] 연구를 통해 폭발적인 인기를 얻게 되었다. 이후, FastText[7], GloVe[8] 등 다양한 워드 임베딩 모델에 대한 연구가 진행되었으며[9-11], 문서에 대한 벡터를 직접 학습하는 doc2vec [12] 모델도 연구되었다. 이러한 워드 임베딩 모델을 이용한 문서 표현 방식은 단어를 고밀도 벡터(dense vector)로 표현함으로써 희소성 문제를 해결하였고, 워드 임베딩 모델을 통하여 생성한 단어 벡터는 NLP 분야의 다양한 문제를 해결하는 데 좋은 성능을 보이고 있다.

1.1 FastText

FastText 모델은 Skip-gram 모델의 구조를 이용하여 subword 정보를 학습하는 모델이다. Skip-gram 모델은 딥 러닝 방식을 통해 단어의 벡터 표현을 학습하는 워드 임베딩 모델이다. 학습에 사용될 텍스트 데이터로부터 추출한 모든 단어의 개수가 N인 단어 집합을 $W=\{w_1, w_2, ..., w_N\}$라고 정의할 때, 로그-가능도(log-likelihood)에 대한 평균을 최대화하는 목적 함수(objective function)를 수식 (1)과 같이 정의한다.

$$\max \frac{1}{N}\sum_{t=1}^{N}\sum_{w_c\in C_t}\log p(w_c|w_t) \tag{1}$$

수식 (1)에서 단어 w_c는 단어 w_t의 문맥 단어(context word)로, 집합 $C_t=\{w_i|t-m\le i\le t+m, i\neq t\}$의 원소이다. 이 때, 집합 C_t는 문맥의 크기가 m인 문맥 단어 집합이다. 단어 w_t의 문맥 단어 w_c에 대한 조건부 확률(conditional probability) $p(w_c|w_t)$은 소프트맥스 함수(softmax function)를 사용하여 수식 (2)과 같이 계산한다.

$$p(w_c|w_t)=\frac{\exp\left(\boldsymbol{v'}_{w_c}^{\top}\boldsymbol{v}_{w_t}\right)}{\sum_{i=1}^{N}\exp\left(\boldsymbol{v'}_{w_i}^{\top}\boldsymbol{v}_{w_t}\right)} \tag{2}$$

수식 (2)에서 $\boldsymbol{v}_w$는 단어 w에 대한 입력 벡터(input vector)로, 단어 집합 W을 bag-of-words로 사용한 원-핫 벡터로 표현된다. $\boldsymbol{v'}_w$는 단어 w에 대한 출력 벡터(output vector)로, 입력 벡터 $\boldsymbol{v}_w$에 대한 Skip-gram 모델의 결과이다. 즉, Skip-gram 모델은 $p(w_c|w_t)$을 최대화하도록 모델을 학습하고, 이로부터 모든 단어들에 대한 벡터 표현을 얻는다. 하지만 이러한 Skip-gram 모델은 단어의 내부 구조를 벡터 표현에 반영하지 못하고, 학습하지 않은 단어를 벡터로 표현하지 못하는 OOV(Out of Vocabulary) 문제가 발생하는 문제점이 있다.

FastText 모델은 Skip-gram 모델을 학습시킬 때, 각 단어에 대하여 character n-gram 방식을 적용하여 subword 집합을 만들어 사용함으로써 단어의 내부 정보를 벡터 표현에 반영하였다. character n-gram 방식이란 단어를 n개의 문자 조합, 즉, n-gram로 나타내는 방식으로, FastText 모델에서는 단어의 시작부분과 끝부분에 기호 〈와 〉를 덧붙여 조합을 생성한다. 예를 들어, '임베딩'이란 단어에 n=3인 character n-gram 방식을 적용하여 만든 subword 집합은 {'〈임

베딩〉', '〈임베', '임베딩', '베딩〉'}이 된다. (단어 전체를 온전히 학습하기 위하여 전체 단어 '〈임베딩〉'도 추가된다.) 따라서, FastText 모델은 '임베딩'이라는 단어를 학습할 때, '임베딩'이라는 단어 하나만을 학습하는 Skip-gram 모델과 달리 , '〈임베딩〉', '〈임베', '임베딩', 그리고 '베딩〉' 이라는 4개의 단어들을 이용한다. 이를 일반화하면, 단어 w에 대해 character n-gram 방식을 적용하여 만든 subword 집합은 $W_s(w,n)=\{w_s|character\ n-grams\ of\ w, \text{and}\ w\}$로 나타낼 수 있다.

FastText 모델에서 단어 w_t의 문맥 단어 w_c에 대한 조건부 확률 $p(w_c|w_t)$은 수식 (3)과 같다.

$$p(w_c|w_t)=\frac{e^{s(w_t,w_c)}}{\sum_{i=1}^{N}e^{s(w_t,w_i)}} \quad (3)$$

소프트맥스 함수를 사용하여 $p(w_c|w_t)$을 계산할 때, exponential의 지수로 두 단어에 대한 내적 대신 수식 (4)와 같이 정의한 score 함수로부터 얻은 score 값을 사용한다.

$$s(w_t,w_c)=\sum_{w_s\in W_s(w_t,n)}\boldsymbol{v}'^{\top}_{w_s}\boldsymbol{v}_{w_c} \quad (4)$$

수식 (4)의 score 함수는 단어 w_t에 대한 subword들과 문맥 단어 w_c의 내적의 합을 계산한다. 이렇게 단어에 대한 subword를 학습함으로써 같은 문자(character)를 가지는 단어들끼리 벡터 표현을 공유할 수 있고, 이에 따라 출현 빈도가 낮은 단어들의 벡터 표현에 대한 신뢰성을 높일 수 있다. 또한, 학습되지 않은 단어들에 대한 벡터 표현도 가능하여 OOV 문제가 발생하지 않는다.

1.2 단어 군집 기반 문서 표현 방식

어댑티브 온라인 이벤트 탐지(Adaptive-online Event Detection) 연구[4]에서는 워드 임베딩 모델과 k-평균 군집화를 이용하여 뉴스 기사를 벡터로 표현하는 단어 군집 기반 문서 표현 방식을 제안하였다.

먼저, 뉴스 기사 집합 D로부터 단어집(vocabulary) v을 만들고, Skip-gram 모델을 학습시킨다. 학습된 Skip-gram 모델을 이용하여 각 단어 w_i을 d차원의 벡터 $\boldsymbol{v}_{w_i}$로 변환한다. 변환된 단어 벡터들에 대하여 k-평균 군집화를 진행하여 k개의 단어 군집 c_i을 생성한다. 생성된 단어 군집 집합 $C=\{c_1,c_2,\dots,c_k\}$을 기반으로 문서 내 단어들의 단어 군집에의 분포를 계산하여 각 문서에 대한 문서 벡터 $\boldsymbol{v}_j$을 생성한다.

TF-IDF를 이용한 원-핫 벡터 표현 방식은 bag-of-words의 크기가 증가함에 따라 벡터의 크기가 커지고, 희소성 문제로 인하여 메모리 공간 사용에 있어 비효율적이다[5]. 반면에 어댑티브 온라인 군집화의 단어 군집 기반 문서 표현 방식은 워드 임베딩 모델을 이용하여 단어를 고밀도 벡터로 표현하고, k-평균 군집화를 통해 문서 벡터를 단어 군집의 분포로 표현함으로써 메모리 효율성을 높일 수 있다. 또한, k-평균 군집화에서 단어 군집의 개수 k를 이용하여 문서 벡터의 크기를 조절할 수 있다. 본 연구에서는 이러한 이점을 사용하기 위하여 단어 군집 기반 문서 표현 방식을 적절히 변형하여 사용하였다. 변형된 문서 표현 방식은 III절에서 자세히 서술한다.

2. 기존 군집 분석 기술

군집화에는 전통적으로 토픽 모델(topic model)과 군집 분석(cluster analysis)이 사용된다. 토픽 모델은 문서 집합의 추상적인 주제, 즉, 토픽을 추출하기 위한 통계적 모델 중 하나로, 텍스트 데이터의 숨겨진 의미 구조를 발견하기 위해 사용되는 텍스트 마이닝 기법 중 하나이다. 토픽 모델 기술에는 주어진 문서에 대하여 각 문서에 어떤 토픽들이 존재하는지에 대한 확률 모델을 구하는 잠재 디리클레 할당(Latent Dirichlet Allocation; LDA)[13]이 주로 사용된다. LDA를 통하여 문서 내의 단어들이 특정 토픽에 속할 확률을 구할 수 있는데, 이를 이용하여 문서 내 단어들의 토픽 분포를 구한 다음, 비슷한 토픽 분포를 가지는 문서들끼리 묶어 군집을 생성할 수 있다[5]. 군집 분석이란 주어진 데이터들을 비슷한 특성을 가진 집단, 즉, 군집(cluster)으로 분할하여 각 군집마다의 특성을 분석하는 방법이다. 군집 분석에 사용되는 대표적인 알고리즘으로는 계층적 군집화(hierarchical clustering)와 k-평균 군집화(k-means clustering)가 있다. 계층적 군집화는 주어진 데이터 사이의 유사도를 기반으로 덴드로그램(Dendrogram)이라는 트리 형태의 계층적 군집

을 생성한다. k-평균 군집화는 k개의 중심점(centroid)을 임의로 설정하여, 주어진 데이터들과 각 중심점 사이의 거리를 계산한 다음, 데이터들을 가장 가까운 군집에 할당함으로써 군집을 생성한다.

2.1 계층적 군집화

계층적 군집화는 주어진 데이터로부터 데이터 사이의 거리 또는 유사도를 계산하여 상이 행렬(dissimilarity matrix)을 만들고 이를 이용하여 군집을 생성한다. 계층적 군집화는 초기 군집 생성 방식과 방향성에 따라 bottom-up 방식(Agglomerative Clustering)과 top-down 방식(Divisive Clustering)으로 나뉜다.

Agglomerative Clustering은 초기에 데이터를 각각 하나의 군집으로 구성하고, 매 step마다 가장 유사한 군집을 병합하는 과정을 반복하여 군집을 생성하는 방식이다. 예를 들어, N개의 데이터 $x_1, x_2, \dots, x_N$ 에 대하여 Agglomerative Clustering 방식을 이용해 군집을 생성할 때, 먼저 각 데이터를 원소로 가지는 크기가 1인 N개의 군집 $C_i = \{x_i\}$ $(i = 1, \dots, N)$을 생성한다. 이후, 상이 행렬을 이용하여 가장 유사한 두 군집을 하나의 군집으로 병합한다. 이 때, 병합된 두 군집 C_j, C_k은 이용할 수 없는 상태(unavailable)가 된다. 모든 데이터가 하나의 군집으로 병합될 때까지 이러한 과정을 반복하여 군집을 생성한다[14]. 이러한 병합 과정은 덴드로그램(dendrogram)이라는 일종의 트리(tree) 형태로 표현될 수 있는데, 이 덴드로그램을 이용하여 임계값(threshold)을 설정함으로써 원하는 개수만큼의 군집을 생성할 수 있다.

반면에, Divisive Clustering은 초기에 전체 데이터를 하나의 군집으로 구성하고, 이를 다양한 기준에 따라 분할하며 군집을 생성한다. Divisive Clustering 방식을 이용해 군집을 생성할 때, 먼저 N개의 데이터 $x_1, x_2, \dots, x_N$ 에 대하여 모든 데이터를 원소로 가지는 크기가 N인 군집 $C_0 = \{x_1, x_2, \dots, x_N\}$을 생성한다. 이후, 상이 행렬을 이용하여 군집 C_k $(k = 0, 1, \dots)$ 내의 모든 데이터에 대하여 군집 내 다른 데이터와의 상이값(dissimilarity) 평균을 계산한 다음, 상이값 평균이 가장 큰 데이터를 다른 군집으로 옮긴다. 이와 같은 작업을 특정 조건을 만족할 때까지 반복한다. 특정 조건의 예로, 임계값 δ을 설정하고 군집 C_k 내 상이값 평균이 가장 큰 데이터의 상이값 평균이 δ보다 작을 경우, 작업을 종료한다. 분할된 군집에 대하여 더 이상 분할되는 군집이 발생하지 않을 때까지 이러한 과정을 반복한다[14].

bottom-up 방식의 Agglomerative Clustering과 top-down 방식의 Divisive Clustering의 시간 복잡도(Time Complexity)는 각각 $O(n^3)$, $O(n)$으로 top-down 방식이 bottom-up 방식보다 빠르다. 또한 Divisive Clustering 방식은 초기 군집에 모든 데이터가 포함되어있기 때문에 모든 데이터에 대한 문맥 정보를 활용하여 군집을 생성할 수 있는 반면, 가장 가까운 데이터끼리 병합하며 군집을 생성하는 Agglomerative Clustering은 군집 생성 시 활용할 수 있는 문맥 정보가 비교적 협소하다[14]. 따라서 본 연구에서는 실시간에 가깝게 뉴스 기사의 이슈를 분석하기 위하여 크기가 큰 이슈 군집에 대하여 하위 이슈 군집을 생성할 때, 속도가 빠르고 문맥 정보를 활용할 수 있는 top-down 방식의 계층적 군집화 방식을 사용한다.

2.2 k-평균 군집화

k-평균 군집화는 k-평균 알고리즘을 이용하여 주어진 데이터들을 k개의 군집으로 분할하는 군집화 방식이다. k-평균 알고리즘은 N개의 D차원 입력 데이터 집합 $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$가 주어졌을 때, K개 군집의 중심점(centroid) 집합 $\mathrm{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}$을 임의로 설정하고, 중심점 $\mathbf{c}_i$와 입력 데이터 $\mathbf{x}_j$ 사이의 유클리드 거리(Euclidean Distance)를 계산하여 거리가 가장 가까운 군집에 $\mathbf{x}_j$을 할당한다. 따라서 k-평균 알고리즘은 수식 (5)와 같이 군집 내 중심점과 그 군집에 할당된 입력 데이터들과의 거리의 합이 최소가 되도록 군집을 생성한다.

$$\min \sum_{i=1}^{K} \sum_{j=1}^{N} \| \mathbf{c}_i - \mathbf{x}_j \|^2 \tag{5}$$

유클리드 거리는 두 벡터 사이의 물리적 거리를 수치적으로 표현하는 반면, 코사인 유사도는 두 벡터의 방향성을 비교하여 두 벡터 사이의 유사성을 수치적으로

표현한다. 따라서 본 연구에서는 벡터로 표현된 단어 사이의 의미적 유사성을 수치적으로 표현하기 위하여 코사인 유사도 방식을 사용하였고, 유사도 측정의 정확성을 높이기 위하여 가중 코사인 유사도 측정 모델(Weighted Cosine Similarity model)을 학습하였다. 학습된 가중 코사인 유사도 측정 모델이 적용된 k-평균 알고리즘에서 중심점과 입력 데이터 사이의 거리는 수식 (6)과 같이 측정한다.

$$\min \sum_{i=1}^{K} \sum_{j=1}^{N} \left| 1 - WCS(\mathbf{c}_i, \mathbf{x}_j) \right| \tag{6}$$

수식 (6)의 WCS는 제안하는 가중 코사인 유사도 측정 모델을 의미한다. 따라서 가중 코사인 유사도 측정 모델이 적용된 k-평균 알고리즘은 중심점과 주어진 데이터 사이의 가중 코사인 유사도를 측정하고, 이를 거리로 환산하여 군집 내 중심점과 그 군집에 할당된 데이터들과의 거리의 합이 최소가 되도록 군집을 생성한다. 제안하는 가중 코사인 유사도 측정 모델은 IV절에서 자세히 기술한다.

3. 기존 가중 유사도 측정 모델

군집 분석 중에서도 계층적 군집화와 k-평균 군집화는 군집을 생성하기 위하여 두 데이터 사이의 유사도를 측정한다. 따라서 유사도 측정의 정확성은 군집화 성능에 중요한 영향을 미친다. C. Wartena와 R. Brussee[15]의 연구는 군집화에 있어서 유사도 측정의 중요성을 보여준다. C. Wartena는 이분법 k-평균 알고리즘(bisecting k-means algorithm)을 이용하여 다양한 유사도 측정 방식에 대한 키워드 추출 및 군집화를 진행한 후, 위키피디아(Wikipedia)의 뉴스데이터를 이용하여 성능 평가를 진행하였다. 이로부터 Jensen-Shannon divergence 기반 유사도 측정 방식이 기존 코사인 유사도 측정 방식보다 더 좋은 성능을 보인다는 것을 정량적으로 증명하였다. 이외에도 기존의 유사도 측정 방식에 가중치(weight)를 추가하여 유사도 측정의 정확성을 향상시키는 연구도 진행되었다[16-18].

3.1 코사인 유사도 측정을 위한 가중치 행렬 학습

H. V. Nguyen et al.[17]의 연구에서는 얼굴 식별(Face Verification) 문제를 해결하기 위하여 두 이미지 사이의 유사도를 측정하기 위한 정확한 유사도 행렬이 필요하였고, 이를 위하여 코사인 유사도 행렬 학습(Cosine Similarity Metric Learning; CSML) 방식을 제안하였다. CSML 방식은 학습데이터로부터 변환 행렬(transformation matrix)을 학습하여 코사인 유사도의 정확도를 높이는 방식으로, 크기가 N인 학습데이터셋 S에 대하여 $S = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, l_i)\}_{i=1}^{N}$와 같이 정의할 때, 이러한 학습데이터셋 S를 이용하여 수식 (7)의 행렬 A를 학습한다.

$$CS(\boldsymbol{x}, \boldsymbol{y}, A) = \frac{(A\boldsymbol{x})^{\top} (A\boldsymbol{y})}{\|A\boldsymbol{x}\| \|A\boldsymbol{y}\|} = \frac{\boldsymbol{x}^{\top} A^{\top} A\boldsymbol{y}}{\sqrt{\boldsymbol{x}^{\top} A^{\top} A\boldsymbol{x}} \sqrt{\boldsymbol{y}^{\top} A^{\top} A\boldsymbol{y}}} \tag{7}$$

집합 S는 하나의 sample이 d차원의 입력 벡터(input vector) $\boldsymbol{x}_i, \boldsymbol{y}_i$와 두 입력 벡터 $\boldsymbol{x}_i, \boldsymbol{y}_i$에 대한 레이블(label) $l_i \in \{0, 1\}$로 이루어진 데이터셋이다. 이 때, $l_i = 1$인 입력 벡터를 positive sample, $l_i = 0$인 입력 벡터를 negative sample이라고 한다. 수식 (7)의 행렬 A는 CSML에서 학습하고자 하는 변환 행렬로, positive sample과 negative sample 사이의 margin을 크게 만드는 함수와 정규화(regularization)를 이용한 목적 함수를 정의하여 행렬 A를 학습시킨다. H. V. Nguyen et al.의 연구에서는 이러한 CSML 방식을 통해 가중치 행렬을 학습하고, 이를 코사인 유사도 측정 방식에 적용하여 LFW (Labeled Faces in the Wild) 데이터셋 기반 얼굴 식별 문제에서 가장 높은 정확도(highest accuracy)를 얻었다.

3.2 샴 신경망 기반 유사도 측정 모델 학습

S. Chopra, R. Hadsell 그리고 Y. LeCun[18]의 연구에서도 유사도 측정의 정확도를 높이기 위한 가중치 행렬을 학습하였는데, 해당 연구에서는 샴 신경망(Siamese Neural Networks)[19]이라는 두 개의 입력 데이터에 대하여 은닉층(hidden layer)의 가중치를 공유하는 쌍둥이(twin) 신경망 구조를 이용하여 얼굴 식별 문제를 해결하기 위한 딥 러닝 모델을 학습하였다. 쌍둥이 신경망 구조란 서로 다른 2개의 입력 데이터(input)를 입력받고, 신경망 최상단의 에너지 함수

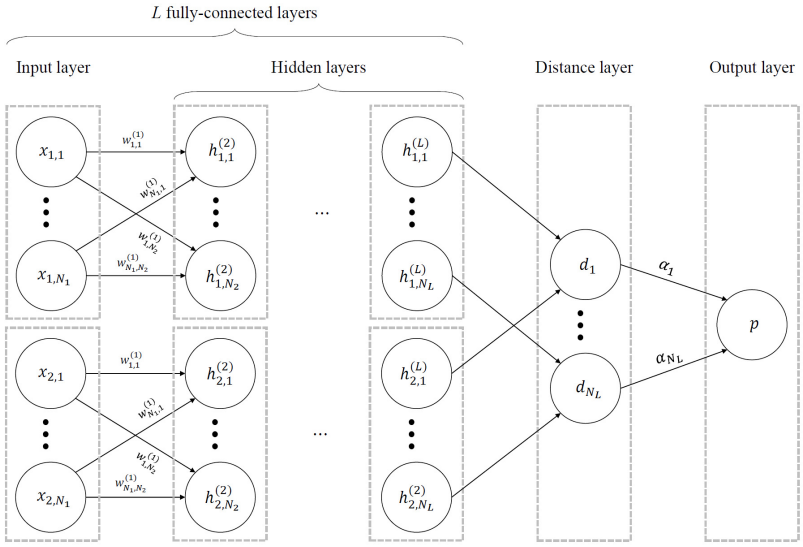


그림 1. L **fully-connected layers each with** N_l **units**
샴 신경망 구조

(energy function)에 의해 2개의 입력 데이터(input)에 대한 출력(output)이 합쳐지는 신경망 구조로 이루어져 있다. 샴 신경망의 구조는 그림 1과 같다. 그림 1은 G. Koch, R. Zemel, 그리고 R. Salakhutdinov[19]의 'a simple 2 hidden layer siamese network for binary classification with logistic prediction p'를 일반화하여 도식화한 그림이다. [그림 1]의 각 용어와 해당 구조에서의 샴 신경망 학습 방식은 다음과 같다.

$$\mathbf{x}_1 = (x_{1,i}) \in R^{N_1},\ \mathbf{x}_2 = (x_{2,i}) \in R^{N_1} \tag{8}$$

$$\mathbf{h}_{1,l} = (h^{(l)}_{1,i}) \in R^{N_l},\ \mathbf{h}_{2,l} = (h^{(l)}_{2,i}) \in R^{N_l} \tag{9}$$

$$W_{l,(l+1)} = (w^{(l)}_{i,j}) \in R^{N_l \times N_{l+1}} \tag{10}$$

$$\mathbf{b}_l = (b_i) \in R^{N_{l+1}} \tag{11}$$

$$\begin{aligned} \mathbf{h}_{1,(l+1)} &= \max(\mathbf{0}, W^{\tau}_{l,(l+1)}\mathbf{h}_{1,l} + \mathbf{b}_l), \\ \mathbf{h}_{2,(l+1)} &= \max(\mathbf{0}, W^{\tau}_{l,(l+1)}\mathbf{h}_{2,l} + \mathbf{b}_l) \end{aligned} \tag{12}$$

수식 (8)의 $\mathbf{x}_1$와 $\mathbf{x}_2$는 각각 1st 트윈(twin)과 2nd 트윈(twin)의 입력 벡터로, $\mathbf{x}_1 = \mathbf{h}_{1,1}$, $\mathbf{x}_2 = \mathbf{h}_{2,1}$가 성립한다. 수식 (9)의 $\mathbf{h}_{1,l}$와 $\mathbf{h}_{2,l}$는 각각 1st 트윈과 2nd 트윈의 l번째 층 은닉 벡터(l^{th} layer hidden vector)를 의미하며, 각 벡터의 크기는 N_l로 동일하다. 수식 (10)의 $W_{l,(l+1)}$는 l번째 층의 N_l개 유닛(unit)들과 $(l+1)$번째 층의 N_{l+1}개 유닛(unit)들을 연결하는 $N_l \times N_{l+1}$ 공유 웨이트 행렬(shared weight matrix)이고, 수식 (11)의 $\mathbf{b}_l$은 l번째 층의 공유 바이어스 벡터(shared bias vector)이다. $l \in \{1, \ldots, L\}$에 대하여 각 층에 rectified linear (ReLU)를 활성화 함수(activation function)로 사용함으로써 $\mathbf{h}_{1,(l+1)}$과 $\mathbf{h}_{2,(l+1)}$는 수식 (12)과 같이 계산된다.

$$\mathbf{d} = |\mathbf{h}_{1,L} - \mathbf{h}_{2,L}| \tag{13}$$

$$\boldsymbol{\alpha} = (\alpha_i) \in R^{N_L} \tag{14}$$

$$p = \sigma(\boldsymbol{\alpha}^{\tau}\,\mathbf{d}) \tag{15}$$

L^{th} layer 이후, 즉, L fully-connected layer 이후 거리층(distance layer)에서는 1st 트윈과 2nd 트윈의 L번째 은닉 벡터 $\mathbf{h}_{1,L}$, $\mathbf{h}_{2,L}$에 대해 L1 distance를 계산함으로써 두 벡터 사이의 유사도를 측정한다(수식 (13)). 수식 (14)의 $\boldsymbol{\alpha}$는 샴 신경망 모델이 학습하는 동안 학습되는 매개변수로, 가중치 벡터(weight vector)로서 component- wise distance에 대한 가중치를 표현한다. 출력층(output layer)에서는 거리층으로부터 구한 거리 벡터(distance vector) $\mathbf{d}$와 가중치 벡터 $\boldsymbol{\alpha}$에 대한 행렬곱(matrix multiplication)을 계산하고,

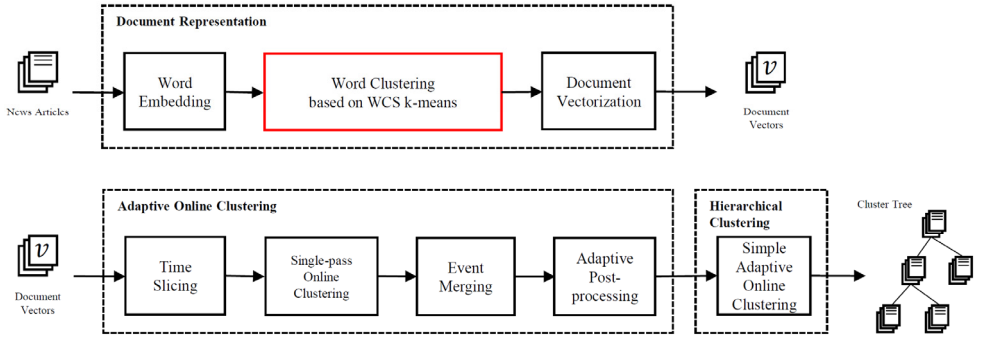


그림 2. 제안하는 초기 이슈 군집 트리 생성 과정

시그모이드 활성화 함수(sigmoid activation function) σ을 이용하여 출력을 0에서 1 사이의 값 p로 표현한다(수식 (15)).

$p(\mathbf{x}_1, \mathbf{x}_2)$을 1st 트윈의 입력 벡터 $\mathbf{x}_1$와 과 2nd 트윈의 입력 벡터 $\mathbf{x}_2$에 대한 siamese neural network with L fully-connected layers each with N_l units의 출력이라고 하자. 이 때, 모델에 대한 손실 함수(loss function) $f(\mathbf{x}_1, \mathbf{x}_2)$는 아래 수식 (17)과 같다.

$$y(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} 1 & if\ \mathbf{x}_1, \mathbf{x}_2\ are\ in\ the\ same\ class \\ 0 & otherwise \end{cases}$$

$$f(\mathbf{x}_1, \mathbf{x}_2) = y(\mathbf{x}_1, \mathbf{x}_2)\log p(\mathbf{x}_1, \mathbf{x}_2) + (1-y(\mathbf{x}_1, \mathbf{x}_2))\log(1-p(\mathbf{x}_1, \mathbf{x}_2)) + \boldsymbol{\lambda}^{\top}|\mathbf{w}|^2 \quad (16)$$

(17)

수식 (16)의 $y(\mathbf{x}_1, \mathbf{x}_2)$는 입력 벡터 쌍 $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$에 대한 레이블로, 두 벡터가 같은 클래스(class)이면 1, 아니면 0인 값을 가진다. 손실 함수는 수식 (17)과 같이 regularized binary cross-entropy 방식을 사용하였다. 최종적으로 이러한 신경망 구조와 손실 함수를 이용하여 모델을 학습한다.

S. Chopra, R. Hadsell 그리고 Y. LeCun은 샴 신경망 구조에서 fully-connected layers 대신 합성곱 신경망(Convolutional Neural Networks)을 적용하였고, 거리층에서는 유클리드 거리를 계산하여 두 얼굴 이미지에 대한 유사도를 측정하는 모델을 학습함으로써 다양한 데이터셋에 대한 얼굴 식별 문제를 해결했다.

본 연구에서는 벡터화된 단어 사이의 유사성을 보다 정확하게 측정하기 위하여 코사인 유사도를 이용한 거리층을 가지는 샴 신경망 기반 가중 코사인 유사도 측정 모델을 구현 및 학습하였다. 또한 학습된 가중 코사인 유사도 측정 모델을 k-평균 군집화에 적용하여 제안하는 군집화 방식의 이슈 분석에 대한 정확성을 향상시켰다.

III. 제안하는 군집화 방식

본 논문에서 제안하는 군집화 방식은 계층적 군집화 방식을 이용하여 초기 이슈 군집 트리를 생성하는 단계와 점증적 군집화 방식을 이용하여 이슈 군집 트리를 업데이트하는 단계로 이루어져 있다. 초기 이슈 군집 트리 생성 단계는 뉴스 기사들에 대하여 이슈 군집 트리를 생성하기 위해 한 번 실행되며, 이후 일정 단위 시간 동안 발생한 기사들은 이슈 군집 트리 업데이트 단계를 통해 이슈 군집 트리에 추가된다.

1. 초기 이슈 군집 트리 생성

본 연구에서 제안하는 초기 이슈 군집 트리 생성 과정은 [그림 2]와 같다. 초기 이슈 군집 트리는 문서 표현(Document Representation), 어댑티브 온라인 군집화(Adaptive Online Clustering), 그리고 계층적 군집화(Hierarchical Clustering)를 통해 생성된다.

1.1 문서 표현 (Document Representation)

Word Embedding. 워드 임베딩 모델을 통해 단어를 벡터로 변환하는 과정이다. 어댑티브 온라인 군집화 연구에서는 문서 표현 단계에서 수집한 뉴스 기사를 이용하여 워드 임베딩 모델을 학습하였다[4]. 하지만, 이슈는 사건이 발생한 시간과 밀접한 관련이 있고, 뉴스 기사는 발생한 사건에 대한 기록이 담긴 문서이기 때문에 일정 기간 동안 수집된 뉴스 기사만으로 워드 임베딩 모델을 학습하면 이슈 분석을 위한 단어들을 충분히 학습하지 못할 가능성이 있다. 따라서 본 연구에서는 이러한 가능성을 최대한 줄이기 위하여 12년 동안의 뉴스 기사를 수집 및 전처리하여 워드 임베딩 모델을 위한 데이터셋을 구축하였고, 이로부터 단어를 추출하여 워드 임베딩 모델을 학습하였다. 데이터셋에 대한 설명과 워드 임베딩 모델에 대한 학습 및 선정 방식에 대해서는 IV절에서 자세히 기술한다.

Word Clustering based on WCS *k*-means. 학습된 워드 임베딩 모델 내의 모든 단어들을 벡터로 변환한 후, 단어 벡터들에 대하여 IV의 3절에서 기술하는 가중 코사인 유사도 측정 모델을 적용한 *k*-평균 알고리즘(WCS *k*-means)을 이용하여 단어들에 대한 군집을 생성한다. 해당 과정에서 *k*값은 문서 벡터의 크기가 되며, 따라서 *k*값에 따라 군집의 성능이 달라진다.

본 연구에서 제안하는 초기 이슈 군집 트리는 뉴스 기사로부터 계층적 이슈를 분석한다. 일반적으로 계층이 낮아질수록 더 자세한 이슈를 다루기 때문에 이슈 군집 트리에서 계층이 낮아짐에 따라, 즉, level이 높아짐에 따라 군집의 개수가 증가하는 것이 이상적이다. 이를 위하여 *k*값을 2의 거듭 제곱 값인 256, 512, 1024로 다양하게 설정하여 초기 이슈 군집 트리를 생성하고, 생성되는 군집의 개수를 정성적으로 평가하여 이상적인 이슈 군집 트리를 생성한다고 판단되는 *k*=512를 선정하였다.

Document Vectorization. 만들어진 단어 군집을 기반으로 문서를 벡터로 변환한다[4]. 예를 들어, Word Clustering based on WCS *k*-means로부터 3개의 단어 군집을 얻었고, 각 단어 군집을 C_1, C_2 그리고 C_3라고 하자. 또한, 각 단어 군집은 $C_1 = \{w_1, w_2, w_4, w_7\}$, $C_2 = \{w_3, w_5, w_{10}\}$, 그리고 $C_3 = \{w_6, w_8, w_9\}$와 같은 단어들을 가지고 있다고 하자. 10개의 단어를 가지고 있는 뉴스 문서 $d = \{w_1, w_1, w_2, w_4, w_5, w_6, w_7, w_7, w_7, w_9\}$에서 각 단어의 단어 군집 C_1, C_2 그리고 C_3으로의 분포로 문서를 벡터로 표현할 때, 단어 군집 C_1, C_2 그리고 C_3에 속하는 단어의 개수는 각각 7개, 1개, 그리고 2개이므로, 뉴스 문서 d에 대한 문서 벡터(document vector) $\mathbf{d}$는 수식 (18)과 같이 표현할 수 있다.

$$\mathbf{d} = (0.7, 0.1, 0.2) \tag{18}$$

본 연구에서는 뉴스 기사가 증가하여 단어의 수가 증가하고, 이에 따라 문서 벡터의 크기가 커지는 기존의 TF-IDF 기반 원-핫 벡터 표현 방식과 달리, 워드 임베딩 모델을 사용하여 단어를 고밀도 벡터로 변환한 후, 단어 군집을 생성함으로써 *k*차원의 문서 벡터를 생성하는 단어 군집 기반 문서 표현 방식을 사용하여 뉴스 기사를 문서 벡터로 표현하였다. 그 결과, 문서 표현에서의 메모리 효율성을 높였고, 문서 벡터의 크기가 감소함에 따라 군집화 속도 또한 빨라졌다.

Algorithm 1: Time Slicing

Input: News documents $\mathrm{D} = \{\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_M\}$
Output: Time-Sliced Clusters D'

1 $W \leftarrow$ time window size
2 $T_i \leftarrow$ ith cluster which is divided by W
$T_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,N_i}\}$ (N_i : size of T_i)
3 $D' \leftarrow$ the set of time-sliced clusters
$D' = \{T_1, T_2, \ldots, T_C\}$ (C: size of D')

1.2 어댑티브 온라인 군집화 (Adaptive Online Clustering)[4]

Time Slicing. Algorithm 1은 어댑티브 온라인 군집화 과정의 Time Slicing에 대한 알고리즘이다. 해당 과정에서는 뉴스 기사로부터 이벤트를 추출할 때, 시간 정보가 중요하다는 점을 이용하여 시간 단위 W(e.g. 주 단위)로 뉴스 문서를 분할한다. 그 결과, C 개의 시간 분할 군집 T_i에 대한 집합 D'이 생성된다. 시간 분할 군집 T_i는 시간 단위 W로 분할된 뉴스 기사들에 대한 군집들 중 i번째 군집을 의미한다.

Algorithm 2: Single-pass Online Clustering

Input: Time-Sliced Clusters D',
Similarity Threshold δ_{sim}
Output: Event Centric Clusters $\mathbf{E}_S = \{\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_C\}$

```
1  foreach T_i ∈ D'(i = 1, ..., C) do
2    initialize event set E_i = ∅
3    for the j-th document d_{i,j} in T_i do
4
calculate the vector representation v_{i,j} of d_{i,j}
5      if E_i = ∅ then
6       create E_1, let d_{i,j} ∈ E_1, represent E_1 by v_{i,j}
7         let E_1 ∈ E_i
8      else
9        foreach event E_k ∈ E_i do
10          calculate the similarity sim(v_{i,j}, E_k)
11          let maxS = max_j sim(v_{i,j}, E_k)
12          let maxE = E_k  if sim(v_{i,j}, E_k) = maxS
13          if maxS ≥ δ_sim then
14            let d_{i,j} ∈ maxE
15         recalculate the representation of maxE
           by the centroid
16          else
17            create a new event E_new
18            let d_{i,j} ∈ E_new, represent E_new by v_{i,j}
19            let E_new ∈ E_i
20     j++
```

Single-pass Online Clustering. Algorithm 2는 어댑티브 온라인 군집화 과정의 Single-pass Online Clustering에 대한 알고리즘이다. 해당 과정에서는 시간 분할 군집 T_i에 대하여 Single-pass Online Clustering 기법을 적용하여 이벤트 집합 E_i을 생성한다. Single-pass Online Clustering은 모든 이벤트 군집의 중심점(centroid)과 데이터 사이의 유사도를 계산한 다음, 가장 높은 유사도 $maxS$ 가 임계값 δ_{sim} 이상일 경우, 해당 이벤트 $maxE$ 에 할당하고, 그렇지 않은 경우 새로운 이벤트 E_{new}을 생성하여 해당 이벤트에 데이터를 할당한다. 이 과정을 반복하여, 모든 T_i에 대한 이벤트 집합 E_i을 생성한다.

Algorithm 3: Event Merging

Input: Event Centric Clusters $\mathbf{E}_S = \{\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_C\}$
, Merging Threshold δ_{merge}
Output: Event Set $\mathrm{E}_M = \{E_1, E_2, \ldots, E_K\}$

```
1  let E_M = E_1
2  foreach event set E_t ∈ E_S(t = 2, ..., C) do
3    foreach event E_{t,i} ∈ E_t do
4      foreach event E_k ∈ E_M do
5        let c_{t,i} represent the centroid of E_{t,i}
6        let c_k represent the centroid of E_k
7        calculate the similarity sim(c_{t,i}, c_k)
8        let maxS = max_i sim(c_{t,i}, c_k)
9        let maxE = E_k  if sim(c_{t,i}, E_k) = maxS
10       if maxS ≥ δ_merge then
11         let maxE ← E_{t,i} ∪ maxE
12       recalculate the representation of maxE
         by the centroid
13       else
14         let E_{t,i} ∈ E_M
```

Event Merging. Algorithm 3은 어댑티브 온라인 군집화 과정의 Event Merging에 대한 알고리즘이다. 이벤트는 Time Slicing에서 정한 시간 단위를 초과하여 발생할 가능성이 있다. 예를 들어, Time Slicing에서 주 단위로 뉴스 기사들을 분할하였을 때, 특정 이벤트는 1달 동안 계속 이슈가 되어 Single-pass Online Clustering을 거치면서 같은 이벤트에 대한 군집들이 중복 생성되었을 가능성이 있다. 이러한 중복된 이벤트를 찾고, 하나의 이벤트로 병합하는 과정이 Event Merging이다. Event Merging에서는 모든 이벤트 집합 E_i 내의 서로 다른 이벤트 $E_{t,i}$, E_k 사이의 유사도를 계산한 다음, 가장 높은 유사도 $maxS$ 가 임계값 δ_{merge}

Algorithm 4: Adaptive Post-processing

Input: Event Set $E_M = \{E_1, E_2, \ldots, E_K\}$,
Noise Threshold δ_{noise}
Output: Event Set $E = \{E_1, E_2, \ldots, E_{K'}\}$

```
let E = ∅
foreach event E_j ∈ E_M (j = 1, ..., K) do
  calculate the noise N_j of E_j
  if N_j ≥ δ_noise then
    apply Single-pass Online Clutsering
    and Event Merging to E_j
    let the set of new events E'_j
    foreach event E ∈ E'_j do
      let E ∈ E
  else
    let E_j ∈ E
```

이상일 경우, 해당 이벤트를 $maxE$ 와 병합하고, 그렇지 않은 경우, 해당 이벤트를 새로운 이벤트 집합 E_M에 할당한다.

Adaptive Post-processing. Algorithm 4는 어댑티브 온라인 군집화 과정의 Adaptive Post-processing에 대한 알고리즘이다. 각 이벤트마다 노이즈 N_j 를 계산하여 (노이즈는 이벤트 내 문서 벡터들 사이의 유사도 값에 대한 분산을 계산하여 구한다) N_j 가 임계값 δ_{noise}보다 클 경우, 해당 이벤트 내 뉴스 문서들에 대하여 더 높은 임계값을 가지는 Single-pass Online Clustering과 Event Merging을 적용하는 후처리 작업(post-processing)을 진행한다. 후처리 작업 결과 생성된 이벤트들은 새로운 이벤트 집합 E에 할당한다.

본 연구에서는 뉴스 기사로부터 이벤트를 추출한다는 점이 본 연구의 목적인 이슈 분석과 비슷하고, 뉴스 기사의 시간적 특성을 이용하여 군집을 생성하기 때문에 어댑티브 온라인 군집화 방식을 사용하였다.

1.3 계층적 군집화 (Hierarchical Clustering)

이슈는 범주에 따라 세분화되는 특성이 있다. 예를 들어, 넓은 범주에서 '군집화'라는 이슈는 좁은 범위의

Algorithm 5: Hierarchical Clustering
-Simple Adaptive Online Clustering

Input: Event Set $E = \{E_1, E_2, \ldots, E_{K'}\}$,
Size Threshold δ_{size}
Output: Issue Clusters $E_1, E_2, \ldots, E_{MAX\_LEVEL}$

```
let level = 1, E_level = E, and E_others = ∅
repeat
  let E_level+1 = ∅
  foreach event E_i ∈ E_level do
    if size of E_i ≤ MIN_CLUSTER_SIZE then
      let E_i ∈ E_others
      remove E_i from E_level
    else
      if the size of E_i ≥ δ_size then
        re-clustering on E_i
        E_i,child ← the set of children clusters
        from re-clustering
        foreach E_i,c ∈ E_i,child do
          let E_i,c ∈ E_level+1
  level++
until level ≤ MAX_LEVEL
```

'계층적 군집화', 'k-평균 군집화' 등으로 다시 나뉠 수 있다. 본 연구에서는 뉴스 기사로부터 이러한 계층적 이슈를 분석하기 위하여 어댑티브 온라인 군집화를 통해 생성한 이벤트 집합에 계층적 군집화를 진행한다. II의 2절에서 기술한 바와 같이, 속도가 빠르고 문맥 정보를 활용할 수 있는 top-down 방식의 계층적 군집화 방식을 사용하여 level이 높아짐에 따라, 즉, down 방향으로 갈수록 세부적인 이슈를 다루는 군집 트리를 생성한다. 본 연구에서 사용한 계층적 군집화 알고리즘은 Algorithm 5와 같다.

먼저 level별로 이슈 군집을 저장하는 집합 E_{level}에 대하여 집합 $E_{level=1}$을 어댑티브 온라인 군집화로부터 생성한 이벤트 집합 $E = \{E_1, E_2, \ldots, E_K\}$로 초기화한다. $E_{level=1}$에 대한 초기화 이후, $E_{level=1}$ 내의 모든 이벤트 E_i의 크기에 따라 군집화 작업을 수행한다. E_i의 크

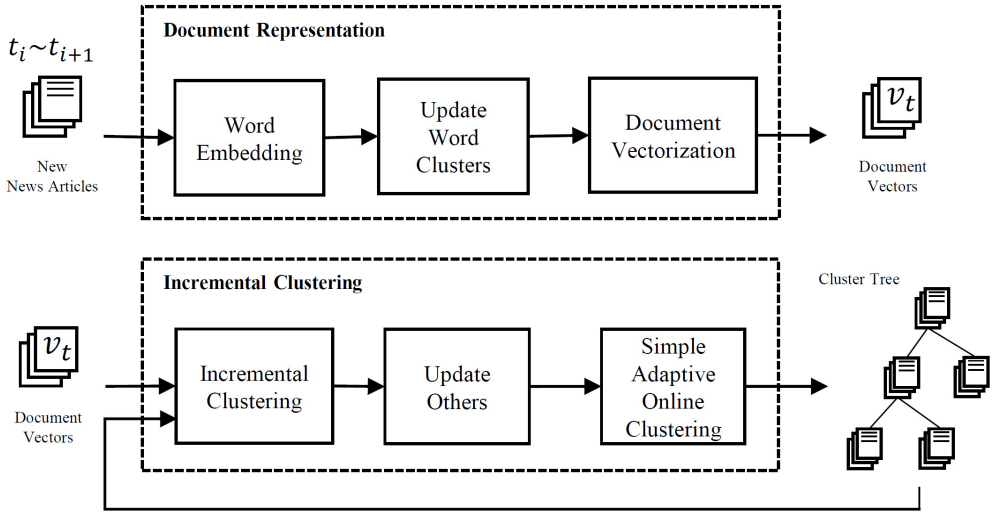


그림 3. 제안하는 이슈 군집 트리 업데이트 과정

기는 E_i에 존재하는 문서들의 개수이다. E_i의 크기가 MIN_CLUSTER_SIZE보다 작을 경우, E_i을 others라는 군집 E_{others}에 할당하고, 기존 집합 E_{level}에서 제거한다[5]. 이러한 E_{others} 할당 과정을 통하여 크기가 너무 작은 군집들을 이슈 분석에서 제외하고, 추후 E_{others}의 크기가 임계값 δ_{size}보다 커질 경우 ($the\ size\ of\ E_{others} \geq \delta_{size}$), E_{others}에 재군집을 진행하여 이슈를 분석한다. 재군집은 어댑티브 온라인 군집화의 Single-pass Online Clustering과 Adaptive post-processing을 차례로 적용하여 하위 이슈 군집 $E_{i,c}$을 생성하는 과정이다. E_i의 크기가 δ_{size}보다 클 경우, E_i에 재군집을 진행하여 $E_{i,c}$를 생성한다. 생성된 하위 이슈 군집 $E_{i,c}$는 하위 레벨 $E_{level=2}$에 할당한다. 이러한 군집 과정을 level이 MAX_LEVEL에 도달할 때까지 반복하여, 깊이(depth)가 MAX_ LEVEL인 이슈 군집 $E_1, E_2, \ldots, E_{MAX_LEVEL}$에 대한 트리를 생성한다. 본 연구에서는 MIN_CLUSTER_ SIZE=4, MAX_LEVEL=3으로 설정하여 계층적 군집화를 진행하였다.

2. 이슈 군집 트리 업데이트

본 연구에서 제안하는 이슈 군집 트리 업데이트 과정은 [그림 3]과 같다. 이슈 군집 트리 업데이트 단계에서 이슈 군집 트리는 문서 표현(Document Representation) 과정과 점증적 군집화(Incremental Clustering) 과정을 통해 새로운 뉴스 기사에 대하여 초기 이슈 군집 트리의 이슈 군집들이 업데이트된다.

2.1 문서 표현 (Document Representation)

Word Embedding. 초기 이슈 군집 트리 생성 단계의 문서 표현 과정 중 Word Embedding에서 사용된 워드 임베딩 모델과 동일한 모델을 사용하여 특정 시간 동안($t_i \sim t_{i+1}$) 발생한 뉴스 기사로부터 단어 군집에 존재하지 않는 단어들을 벡터로 변환한다.

Update Word Clusters. 새로운 단어에 대하여 초기 이슈 군집 트리 생성 단계의 문서 표현 과정에서 생성된 단어 군집들의 중심점과 새로운 단어 벡터 사이의 유사도를 계산하고, 가장 유사한 단어 군집에 새로운 단어를 할당한다. 이러한 과정을 통하여 새로운 단어에 대한 문서 표현이 가능하도록 한다.

Document Vectorization. 업데이트된 단어 군집을 기반으로 문서를 벡터로 변환한다. 변환 방식은 초기 이슈 군집 생성 단계의 문서 표현 과정 중 Document Vectorization과 동일하다.

Algorithm 6: Incremental Clustering

Input: News documents $D = \{d_1, d_2, \ldots, d_M\}$,
Issue Clusters $E_1, E_2, \ldots, E_{MAX\_LEVEL}$,
Similarity Threshold δ_{sim}
Output: Issue Clusters $E_1, E_2, \ldots, E_{MAX\_LEVEL}$

```
for the j−th document d_j in D do
  for i ← 1 to MAX_LEVEL do
    calculate the vector representation v_j of d_j
    foreach E_k ∈ E_i do
      c_k ← a centroid vector of E_k
      calculate the similarity sim(v_j, c_k)
      let maxS = max_j sim(v_j, c_k)
      let maxE = E_k if sim(v_j, E_k) = maxS
      if maxS ≥ δ_sim then
      let d_j ∈ maxE and mark maxE as updated
        recalculate the representation of maxE
        by the centroid
      else
        let E_k ∈ E_others
close not updated clusters during T
```

Algorithm 7: Update Others Cluster

Input: Others Cluster E_{others}, Size Threshold δ_{size},
Issue Cluster E_1
Output: Updated Others Cluster E'_{others}

```
if the size of E_others ≥ δ_size then
  re-clustering on E_others
  calculate the vector representation v_j of d_j
  E_others,child ← the set of children clusters
  from re−clustering
  foreach E_others,c ∈ E_others,child do
    if the size of E_others,c
      ≤ MIN_CLUSTER_SIZE then
      let E_others,c ∈ E'_others
    else
      let E_others,c ∈ E_1
```

본 연구에서는 새로 수집한 뉴스 기사에서 초기 이슈 군집 트리 생성 단계의 문서 표현 과정 중 Word Clustering based on WCS k-means를 통해 생성한 기존의 단어 군집에 존재하지 않는 단어를 추출하고, OOV에 대해서도 벡터 표현이 가능한 FastText 모델을 사용하여 새로운 단어들을 벡터로 표현한 다음, 가장 유사한 단어 군집에 추가함으로써 문서 표현에서의 확장성을 높인다. 이러한 방식은 새로운 뉴스 기사로부터 발생한 단어를 문서 표현에 반영하고, 이는 새로운 이슈에 대한 분석을 용이하게 해준다.

2.2 점증적 군집화 (Incremental Clustering)

Incremental Clustering. Algorithm 6은 점증적 군집화 과정의 Incremental Clustering에 대한 알고리즘이다. 점증적 군집화의 새로운 뉴스 기사를 문서 벡터로 표현한 다음, 초기 이슈 군집 트리 생성 단계를 통해 생성한 이슈 군집 트리의 군집들 중 가장 유사한 군집을 구한다. 해당 군집과 뉴스 기사의 유사도가 임계값 δ_{sim}보다 크면 해당 군집에 뉴스 기사를 할당한다. 그렇지 않은 경우, others 군집 $\mathbf{E}_{others}$에 할당한다[5]. 또한, duration T를 설정하고 T 시간 동안 새로운 뉴스 기사가 추가되지 않은 군집은 close하여 더 이상 이슈 군집 트리에서 제외한다. 예를 들어, $T = one\ month$를 설정한 다음, 2016년 10월까지의 뉴스 기사로부터 생성된 이슈 군집 트리에 새로운 2016년 11월, 12월 뉴스 기사를 Incremental Clustering을 통해 추가할 때, 새로운 뉴스 기사를 Incremental Clustering을 통해 이슈 군집 트리에 추가할 동안 어느 한 뉴스 기사도 추가되지 않은 군집들은 close한다. 이렇게 close 기능을 통하여 불필요한 이슈 군집과의 연산을 줄임으로써 시간을 단축시킨다. 또한, T 이후 close된 이슈와 같은 이슈가 다시 발생할 때는 해당 이슈를 새로운 군집으로 재생성함으로써 이슈가 단절되는 문제를 해결한다.

Update Others Cluster. Algorithm 7은 점증적 군집화 과정의 Update Others Cluster에 대한 알고리즘이다. E_{others}의 크기가 임계값 δ_{size}보다 커질 경우, E_{others}에 대하여 재군집(re-clustering)을 진행한다. 재군집 방식은 초기 이슈 군집 트리 생성에서의 재군집 방식과 동일하다. 재군집을 통해 생성된 하위 이슈 군

Algorithm 8: Incremental Clustering
−Simple Adaptive Online Clustering

Input: Issue Clusters $E_1, E_2, \cdots, E_{MAX\_LEVEL}$,
Updated Others Cluster E'_{others},
Size Threshold δ_{size}
Output: Updated Issue Clusters
$E'_1, E'_2, \cdots, E'_{MAX\_LEVEL}$

```
let level = 1, E'_level = ∅
repeat
  let E'_(level+1) = ∅
  foreach event set E_i ∈ E_level do
    if E_i is updated then
      if the size of E_i
          ≤ MIN_CLUSTER_SIZE then
        let E_i ∈ E'_others
      else
        if the size of E_i ≥ δ_size then
          re-clustering on E_i
          E_others,child ← the set of children
          clusters from re−clustering
          foreach E_i,c ∈ E_i,child do
            let E_i,c ∈ E'_(level+1)
        else
          let E_i ∈ E'_level
  level++
until level ≤ MAX_LEVEL
```

집 $E_{others,c}$는 기존 이슈 군집 트리의 level 1 이슈 군집에 대한 집합 E_1에 할당한다[5].

Simple Adaptive Online Clustering. Algorithm 8은 점증적 군집화 과정의 Simple Adaptive Online Clustering에 대한 알고리즘이다. 새로운 뉴스 기사가 추가된 군집 중, 크기가 임계값 δ_{size} 이상인 군집에 대하여 초기 이슈 군집 트리 생성 단계의 계층적 군집화 과정의 재군집과 동일한 방식으로 하위 이슈 군집 $E_{i,c}$를 생성한다. 생성된 $E_{i,c}$은 새로운 이슈 군집에 대한 집합 $E'_{level+1}$에 할당된다.

본 연구에서는 위와 같은 방식으로 점증적 군집화 과정을 통해 새로운 뉴스 기사로부터 새로운 이슈 군집을 생성하고 기존의 이슈 군집 트리를 업데이트함으로써 최신 이슈를 분석한다.

IV. 실험 결과

1. 데이터셋 구축

1.1 뉴스 데이터셋

본 연구에서는 워드 임베딩 모델 학습을 위하여 네이버 뉴스 사이트[20]로부터 뉴스 기사를 수집하였다. 2005년도부터 2016년까지 약 3,650만 개의 뉴스 기사에 대한 제목, 본문, 카테고리 등의 정보를 텍스트 형식으로 저장하였다. 또한, 한국어 형태소 분석기인 Mecab-ko[21]를 이용하여 제목과 본문의 문장들의 형태소를 분석하고 그 결과를 json 형식 파일로 저장함으로써 뉴스 데이터셋을 구축하였다. 최종적으로 해당 json 형식 파일들로부터 형태소가 일반 명사(NNP), 고유 명사(NNP)인 단어들만을 추출하여 다양한 FastText 모델을 학습하였다.

1.2 유의어 단어 쌍 데이터셋

표 1. 유의어 단어 쌍 데이터셋 예시

word	target_code	linked_word	linked_target_code	label
사람	12170	인간	553451	1
사내	44477	남자	368493	1
어머니	17531	엄마	17642	1
기관총	105884	남자배우	735338	0
아빠	16687	수염풍뎅이	206068	0
개	79790	호황	60686	0

본 연구에서는 유사도 측정의 정량적인 평가를 위하여 우리말샘 사전[22]으로부터 골드 스탠다드 데이터셋(gold standard dataset)을 구축하였다. 우리말샘 사전의 API를 이용하여 단어에 대한 의미 번호, 링크 정보(비슷한말, 반대말 등)를 수집하였고, 수집한 데이터로부터 두 단어에 대하여 링크 정보가 '비슷한말'인 경우 1, 그렇지 않은 경우 0으로 레이블링(labeling)하였다. 구축한 데이터셋에 대한 예시는 [표 1]과 같다.

[표 1]과 같이 우리말샘 사전에 등록된 링크 정보에

따라 '비슷한말' 관계에 있는 두 단어 (예시: 〈사람, 인간〉)의 레이블은 1, 반면에 '비슷한말' 관계가 아닌 다른 관계를 가지는 두 단어(예시: 〈아빠, 수염풍뎅이〉)의 레이블은 0으로 분류하였다. 위와 같은 방식으로 총 159,182개의 유의어 단어 쌍 데이터셋을 구축하였다.

가중 코사인 유사도 측정 모델을 학습시키거나 학습된 모델에 대한 성능을 평가할 때, 모델이 입력 데이터의 대칭성에 대하여 독립성을 유지할 수 있도록 유의어 단어 쌍의 순서가 뒤바뀐 데이터도 추가로 생성하였다. 즉, 〈어머니, 엄마 | label: 1〉이라는 유의어 단어 쌍이 존재한다면, 〈엄마, 어머니 | label: 1〉이라는 유의어 단어 쌍 또한 추가된다. 따라서 총 318,364개의 유의어 단어 쌍이 모델 학습 및 평가에 사용되었다.

1.3 군집 테스트셋

본 연구에서는 군집화 성능의 정량적인 평가를 위하여 IV의 1.1절에서 수집한 뉴스 기사 중, 2016년 10월, 11월 뉴스 기사 5,539개에 대하여 이슈 태깅 작업을 진행하여 군집 테스트셋을 만들었다. 2명의 학생들이 1차적으로 이슈를 태깅하였고, 1차 작업 완료 후, 또 다른 2명의 학생들이 태깅된 이슈를 검증하는 절차를 거쳐 최종적으로 5,539개의 군집 테스트셋을 구축하였다. 군집 테스트셋에 대한 예시는 표 2와 같다. 표 2의 'id'는 뉴스 기사의 고유 번호를 의미하고, 'level N'은 각 뉴스 기사의 N번째 계층의 이슈를 의미한다. 예를 들어, [표 2]의 첫 번째 기사의 고유 번호는 1이고, 해당 기사는 총 3계층의 이슈를 가지고 있으며 각 계층의 이슈는 1부터 3까지 순서대로 '4차 산업혁명 - 4차 산업혁명 직업 - 4차 산업혁명 유망직업'이다. 위의 5,539개의 군집 테스트셋의 각 계층별 이슈 군집의 통계 정보는 [표 3]과 같다.

[표 3]과 같이 군집 테스트셋의 level 1부터 level 3 계층까지의 군집 개수는 차례대로 1,408개, 1,594개, 그리고 251개이고, level 1의 군집은 평균 4개의 뉴스 기사를, level 2와 level 3의 군집은 평균 2개의 뉴스 기사를 가지고 있음을 알 수 있다.

2. 워드 임베딩 모델 선정

표 2. 군집 테스트셋 예시

id	level1	level2	level3
1	4차 산업혁명	4차 산업혁명 직업	4차 산업혁명 유망직업
2	4차 산업혁명	4차 산업혁명 직업	4차 산업혁명 노동시장 대응
3	4차 산업혁명	4차 산업혁명 일자리	N/A
4	5G	SK텔레콤 5G MOU체결	N/A
5	5G	5G 주파수 공급 요청	N/A
6	6차 산업화	N/A	N/A

표 3. 군집 테스트셋에 대한 통계 정보

구분	level1	level2	level3
군집 개수	1,408	1,594	251
가장 큰 군집의 크기	207	57	22
가장 작은 군집의 크기	1	1	1
군집 크기 평균	3.93	2.27	1.85
군집 크기 표준 편차	10.37	3.66	2.21

* 군집의 크기: 군집 내 뉴스 기사의 개수

본 연구에서는 군집을 생성하기 위한 단어 군집 기반 문서 표현 방식에서 워드 임베딩 모델을 이용하여 단어를 벡터로 변환한다. 따라서 워드 임베딩 모델의 성능이 군집화 모듈 성능에 영향을 미친다. 해당 절에서는 유추 검사(Analogy Test)를 통해 좋은 성능을 보이는 워드 임베딩 모델을 선정한 실험 및 결과에 대하여 기술한다.

워드 임베딩 모델의 성능을 검증하기 위한 방식으로는 워드 임베딩 모델 자체의 의미론적 특성을 비교하는 직접적인 평가 방식과 워드 임베딩 모델을 이용하여 NLP 분야 과제에서의 성능을 비교하는 간접적인 평가 방식이 있다. 직접적인 평가 방식에서는 특정 단어 쌍에 대하여 유사도를 측정하는 유사도 검사(similarity test), TOEFL의 어휘 문제를 이용하여 동의어 관계를 평가하는 동의어 검사(synonym test), 그리고 유추 관계를 평가하는 유추 검사가 있다[23]. 이 3가지 검사 중에서도 유추 검사를 통하여 학습된 워드 임베딩 모델이 의미론적·문법적 관계를 얼마나 잘 표현하는지 측정할 수 있다. 따라서 본 연구에서는 워드 임베딩 모델을 직접적으로 평가하는 방식 중, 유추 검사를 사용하여 워

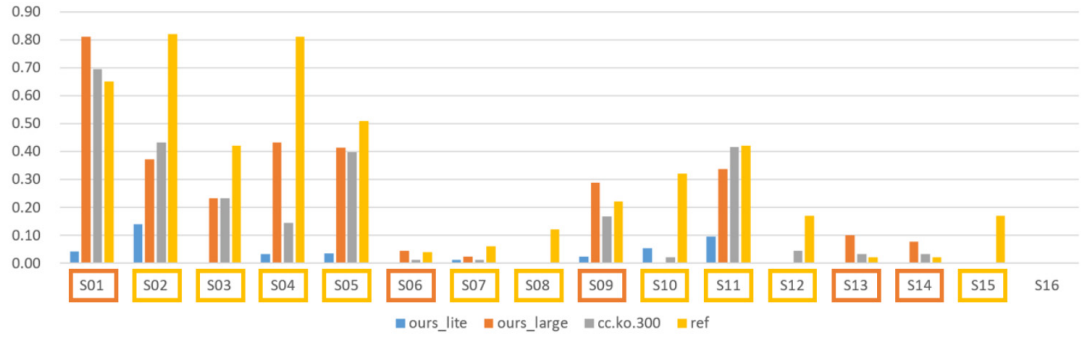


그림 4. 워드 임베딩 모델 성능 비교

드 임베딩 모델의 성능을 정량적으로 평가하고, 우리 연구에 적합한 모델을 선정하였다.

유추 검사에 사용한 데이터셋은 강형석[23] '한국어 유추 테스트셋(Korean Analogy Test Set; KATS)'을 사용하였다. KATS는 크게 의미론적 유추 범주와 문법적 유추 범주로 분류되며, 의미론적 유추 범주는 세부 섹션인 S 섹션(S01~S16)으로, 문법적 유추 범주는 G 섹션(G01~G15)으로 다시 나뉜다. 각 범주에 속하는 데이터의 개수는 의미론적 유추 범주 2,746개, 문법적 유추 범주 1,874개이다. 본 연구에서 제안하는 이슈 분석을 위한 계층적 군집화는 문법적인 요소보다 의미론적인 요소가 더 중요하므로, KATS 중 의미론적 유추 범주에 해당하는 데이터셋에 대해서만 성능을 측정하고 비교한다. 또한, 본 연구에서 수집한 뉴스 데이터셋이 Mecab을 이용하여 전처리된 점을 감안하여, Mecab으로 형태소 분석된 KATS(kor_analogy_kkma.txt)를 실험에 사용한다. KATS에 대한 각 워드 임베딩 모델의 정확도를 비교한 그래프는 [그림 4]와 같다. [그림 4]에서 각 섹션별로 표시된 정보는 섹션마다 어떤 모델이 최고 성능을 보였는지를 나타낸다.

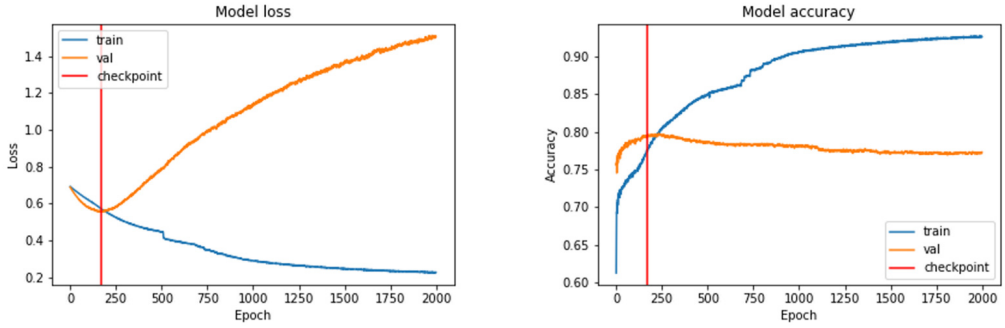
ours_lite는 수집한 뉴스 기사 중, 2016년 10월, 11월 뉴스 기사 약 1만 6천개를 학습한 FastText 모델로 전반적으로 좋지 않은 성능을 보이고 있다. cc.ko.300[24]은 공개되어있는 미리 학습된 FastText 한국어 워드 임베딩 모델로, S02, S03, S05, 그리고 S11에서는 ours_large보다 좋거나 비슷한 성능을 보이지만 다른 섹션에서는 좋지 않은 성능을 보이고 있다. 따라서 ours_lite와 cc.ko.300은 후보군에서 제외하였다.

ours_large는 수집한 뉴스 기사 약 3,650만개를 모두 학습한 FastText 모델로, 5개의 섹션에서 최고 성능을 보이고 있다. 반면에, 나무위키와 위키백과의 대용량 덤프 파일을 학습한 강형석 연구의 Skip-gram 모델인 ref는 10개의 섹션에서 최고 성능을 보이고 있다. 하지만 Skip-gram 모델인 ref는 OOV 문제가 발생하기 때문에 새로운 이슈가 발생함에 따라 워드 임베딩 모델에 학습되지 않은 단어가 출현할 가능성이 있는 우리 연구에 부적절하다. 이와 같은 이유로, 본 연구에서는 단어를 벡터로 변환하기 위한 워드 임베딩 모델로 약 3,650만개의 뉴스 기사를 학습한 FastText 한국어 워드 임베딩 모델 ours_large를 사용하였다. ours_large 모델의 학습된 단어는 총 437,487개이며, 단어 벡터의 차원은 300이다.

3. 가중 코사인 유사도 측정 모델 학습

3.1 모델 구조

본 연구에서 제안하는 샴 신경망 구조 기반 가중 코사인 유사도 측정 모델 구조는 [그림 1]의 L fully-connected layers each with N_l units 샴 신경망 구조를 기반으로 하여 모델을 구성하였다. 먼저, 구축한 유의어 단어 쌍 데이터셋으로부터 데이터를 불러온 다음, 두 단어 w_l와 w_r을 IV의 2절에서 선정한 한국어 워드 임베딩 모델을 이용하여 각각 벡터 v_l과 v_r로 변환한다. 각 벡터는 입력층(Input layer)을 통하여 은닉층(Hidden layer)로 전파(propagate)된다. 은닉층



(a) train/val loss curve (b) train/val accuracy curve

그림 5. 샴 신경망 구조 기반 가중 코사인 유사도 측정 모델 학습 결과

은 5 fully-connected layers 구조로 이루어져 있다. 각 은닉층은 300개의 유닛을 가진 입력층과 입력층에 가까운 순서대로 512, 256, 128, 64개의 유닛을 가지는 4개의 은닉층들로 구성되어 있다. 또한 학습과정에 과적합 (overfitting)을 방지하기 위하여 5 fully-connected layers의 각 은닉층에 드롭아웃(Dropout)을 적용하였다. 드롭아웃의 확률값은 0.5로 각 은닉층에 동일하게 적용하였다. 샴 신경망 구조에 따라 두 입력층을 통해 전파된 벡터들에 대하여 학습된 은닉층의 가중치는 동일하다. 해당 은닉층으로부터 얻은 v'_l과 v'_r은 코사인 거리층(Cosine Distance layer)으로 전파되어 두 벡터 사이의 유사도 $sim(v'_l, v'_r)$가 계산된다. 해당 유사도 값을 출력층(Output layer)으로 전파하여 수식 (19)에 따라 두 단어 사이의 유의 관계를 예측한다.

$$label(w_l, w_r) = \begin{cases} 1 & \text{if } sim(v'_{w_l}, v'_{w_r}) \geq 0.5 \\ 0 & \text{if } sim(v'_{w_l}, v'_{w_r}) < 0.5 \end{cases} \quad (19)$$

3.2 hyper-parameter 최적화 및 모델 선정

본 연구에서는 샴 신경망 구조에서의 hyper-parameter 최적화 실험을 진행하여 유의어 단어 쌍 학습 데이터셋에 대하여 가장 좋은 성능을 보이는 hyper-parameter를 선정하였다. 실험을 위하여 유의어 단어 쌍 데이터셋을 'train : validation : test = 0.64 : 0.16 : 0.20' 비율로 나누었고, 이 중 학습데이터셋(train)과 평가데이터셋(validation)을 사용했다.

표 4. Hyper-parameter Setting

epoch	2000
batch size	256
dropout	0.5
loss function	binary cross-entropy
activation function	ReLU
optimizer	Adam
learning rate	0.001

다양한 activation function, optimizer, learning rate 조합을 설정하고 각 조합별로 10번의 학습 결과를 도출하였다. 그 결과, 〈ReLU, Adam, 0.001〉 조합이 train accuracy: 0.8275 / validation accuracy: 0.7919 으로 가장 좋은 성능을 보였다. 따라서 학습에 사용된 hyper-parameter는 [표 4]와 같다.

선정된 hyper-parameter로 학습한 모델에 대한 train/validation의 loss curve (a)와 accuracy curve (b)는 [그림 5]와 같다. [그림 5]의 (a)는 각 epoch마다의 train/validation에 대한 loss curve를 그래프로 표현한 그림으로, epoch 173 (빨간 직선) 이후로 train loss 값은 감소하지만, validation loss 값은 증가하는 과적합 현상이 발생하는 것을 알 수 있다. 따라서 checkpoint를 설정하여 과적합이 발생하기 전, loss 값이 최소인 모델을 저장하였다. 해당 checkpoint에 해당하는 모델의 accuracy는 [그림 5]의 (b)와 같다. [그림 5]의 (b)에서 checkpoint에서의 모델의 validation accuracy는 0.7951로 최댓값에 가

		Pred	
		Pos	*Neg*
True	*Pos*	TP	FN
	Neg	FP	TN

그림 6. 이진 분류에서의 혼잡 행렬

WCS	Confusion Matrix		Pred	
			1	0
	True	1	28740	3098
		0	10058	21780

Precision	0.7408
Recall	0.9027
Accuracy	0.7934
F1-score	0.8137

CS	Confusion Matrix		Pred	
			1	0
	True	1	4038	27800
		0	28	31810

Precision	0.9931
Recall	0.1268
Accuracy	0.5630
F1-score	0.2249

그림 7. 유의어 단어 쌍 테스트셋에 대한 성능 비교 평가

까운 값을 가지는 것을 확인할 수 있다.

3.3 모델 평가

학습된 가중 코사인 유사도 측정 모델의 유의어 단어 쌍 데이터셋의 테스트셋(test)을 이용하여 학습된 모델의 성능을 평가한다. 테스트셋의 개수는 63,676개로 테스트셋에 대하여 기존 코사인 유사도를 이용한 방식과 제안하는 가중 코사인 유사도 측정 모델의 성능을 혼잡 행렬(confusion matrix)과 정밀도(Precision), 재현율(Recall), 정확도(Accuracy), 그리고 F1-score 측면에서 비교 평가한다.

혼잡 행렬은 기계 학습의 분류(classification) 문제에 대한 결과를 표현한 행렬로, 이진 분류(binary classification)에 대한 2x2 혼잡 행렬은 [그림 6]과 같이 나타낼 수 있다.

[그림 6]의 Pos와 Neg는 각각 Positive와 Negative를 의미하며, 데이터셋에서의 레이블을 나타낸다. 예를 들어, 이미지에 대하여 사람인지 아닌지를 분류하는 문제에서 '사람이다'는 Positive 레이블, '사람이 아니다'는 Negative 레이블이 된다. 크기가 N인 데이터셋에서 레이블이 Positive인 데이터의 집합을 Pos_{true}, 레이블이 Negative인 데이터의 집합을 Neg_{true}, 기계 학습 모델을 이용해 Positive라고 예측한 데이터의 집합을 Pos_{pred}, Negative라고 예측한 데이터의 집합을 Neg_{pred}라고 할 때, TP, FN, TN, FP에 대한 수식은 다음과 같다.

$$\mathrm{TP} = |Pos_{true} \cap Pos_{pred}| \tag{20}$$

$$\mathrm{FN} = |Pos_{true} \cap Neg_{pred}| \tag{21}$$

$$\mathrm{TN} = |Neg_{true} \cap Neg_{pred}| \tag{22}$$

$$\mathrm{FP} = |Neg_{true} \cap Pos_{pred}| \tag{23}$$

수식 (20)의 TP(True Positive)는 레이블이 Positive인 데이터를 Positive로 예측한 데이터의 개수를, 수식 (21)의 FN(False Negative)는 레이블이 Positive인 데이터를 Negative로 예측한 데이터의 개수를 의미한다. 또한, 수식 (22)의 TN(True Negative)는 레이블이 Negative인 데이터를 Negative로 예측한 데이터의 개수를, 수식 (23)의 FP(False Positive)는 레이블이 Negative인 데이터를 Positive로 예측한 데이터의 개수를 의미한다. 이러한 혼잡 행렬을 이용하여 정밀도, 재현율, 정확도, 그리고 F1-score를 다음과 같이 구할 수 있다.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{24}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{25}$$

$$Accuracy = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FN}+\mathrm{TN}+\mathrm{FP}} \tag{26}$$

$$\text{F1-score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{27}$$

수식 (24)은 정밀도(Precision)에 대한 수식으로, 기계 학습 모델이 Positive라고 예측한 데이터 중, 실제로 레이블이 Positive인 데이터의 비율을 나타낸다. 수식 (25)은 재현율(Recall)에 대한 수식으로, 실제로 레이블이 Positive인 데이터 중, 기계 학습 모델이 Positive라고 예측한 데이터의 비율을 나타낸다. 수식 (26)은 정확도(Accuracy)는 전체 데이터 중 기계 학습 모델이 레이블을 올바르게 예측한 데이터의 비율을 의

미한다. 수식 (27)의 F1-score는 정밀도와 재현율에 대한 조화 평균으로, 0 이상 1 이하의 값을 가지며, 1에 가까울수록 좋은 성능을 의미한다.

이와 같은 혼잡 행렬과 정밀도, 재현율, 정확도, 그리고 F1-score를 이용하여 유의어 단어 쌍 테스트셋 63,676개에 대한 성능을 평가한 결과는 [그림 7]과 같다.

유의어 단어 쌍 테스트셋에서 Positive 레이블은 '두 단어가 유의어 관계에 있다(1)'이고, Negative 레이블은 '두 단어가 유의어 관계에 있지 않다(0)'이다. 가중 코사인 유사도 측정 모델(WCS)의 정밀도는 0.7408로 기존 코사인 유사도 방식(CS)의 정밀도(0.9931)보다 낮은 수치를 보여주고 있지만, 각각의 혼잡 행렬에서 Positive 레이블에 대한 값을 확인해보면, WCS는 유의어 관계에 있는(Positive) 단어 쌍 31,838개 중 28,740개를 올바르게 예측한 반면에, CS는 31,838개 중 4,038개만을 올바르게 예측하였다. 따라서. 정밀도와 혼합 행렬을 함께 고려하였을 때, WCS가 CS보다 좋은 성능을 보임을 알 수 있다. 재현율은 WCS: 0.9027, CS: 0.1268로 WCS가 압도적으로 높은 수치를 보여주고 있고, 혼합 행렬과 함께 고려하여도 WCS가 여전히 좋은 성능을 보이고 있음을 알 수 있다. 또한 정확도와 재현율을 함께 고려한 지표인 F1-score에 대하여 WCS: 0.8137, CS: 0.2249로 WCS가 CS보다 우수한 성능을 보이고 있다.

본 논문의 모델 평가에 사용한 유의어 단어 쌍 테스트셋은 레이블이 0인 데이터 31,838개, 레이블이 1인 데이터 31,838개로 레이블에 따른 데이터의 분포가 균일하다. 따라서 단순히 정확도만으로도 어느 모델이 우수한지 평가할 수 있다. WCS와 CS의 정확도는 각각 0.7934, 0.5630으로, WCS의 정확도가 0.2265 정도 높다. 따라서 WCS가 두 단어의 유사도를 계산하는 데 있어서 기존 코사인 유사도 방식보다 우수한 성능을 보임을 알 수 있다.

본 논문에서는 우리말샘 사전으로부터 유의어 단어 쌍 데이터셋을 구축하였다. 또한 이 데이터셋을 이용하여 샴 신경망 기반 가중 코사인 유사도 측정 모델을 학습 및 평가하였다.

표 5. 이슈 분석 정확도 성능 비교 평가

구분		orig	ours	ours+wcs
ARI	level1	0.0483	0.1417	**0.2100**
	level2	0.0669	0.0771	**0.1034**
	level3	0.0624	0.0652	**0.0914**
NMI	level1	0.4601	0.7052	**0.7191**
	level2	0.6534	0.7165	**0.7215**
	level3	0.6670	0.7152	**0.7195**
F1-score	level1	0.1367	0.2106	**0.2253**
	level2	0.1175	0.1574	**0.1779**
	level3	0.1003	0.1706	**0.1809**

4. 계층적·점증적 군집화 실험

해당 절에서는 학습된 가중 코사인 유사도 측정 모델을 단어 군집 기반 문서 표현 방식의 k-평균 알고리즘에 적용하여 군집화 모듈의 성능을 정량적으로 평가한다.

4.1 계층적 이슈 분석 실험

본 연구에서는 군집 테스트셋인 3계층으로 이슈 태깅이 완료된 2016년 10월, 11월 뉴스 5,539개에 대하여 계층적 이슈 분석을 진행하였다. 또한, 군집화를 통해 분석한 이슈에 대하여 군집 결과에 대한 정확도를 측정하는 3가지 지표를 이용하여 정량적으로 평가하였다. 3가지 지표는 다음과 같다.

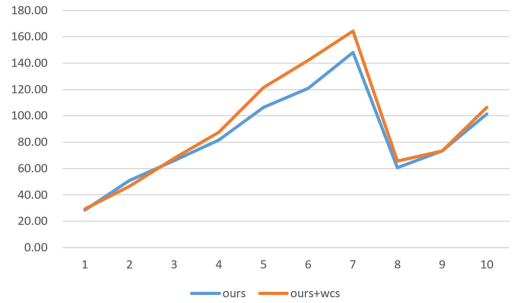
$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]} \tag{28}$$

$$\mathrm{NMI} = \frac{\mathrm{MI}(U, V)}{mean(H(U), H(V))} \tag{29}$$

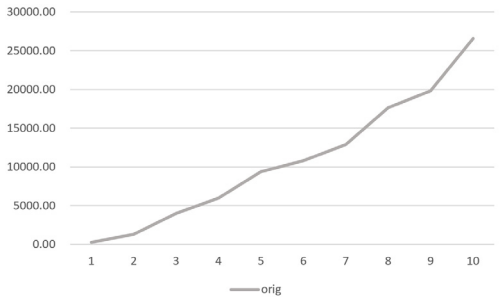
$$\mathrm{FScore} = \sum_{i=1}^{|U|} \frac{|u_i|}{N} \max_{v_j \in V} \mathrm{F1-score}(u_i) \tag{30}$$

수식 (28)의 ARI(Adjusted Rand Index)는 RI(Rand Index; $\mathrm{RI} = (a+b)/C_2^{n_{samples}}$)에 임의성(randomness)을 추가하여 확률적으로 조정한 군집 평가 지표로, 모든 데이터에 대하여 데이터 쌍(pair)을 만들고, 데이터 쌍이 정답 군집과 예측 군집에 동일하게 존재하는지에 대한 정보를 이용하여 군집의 성능을 평가한다. ARI는 -1에서 1 사이의 값을 가지며, 0에 가까울수록 낮은 성능임을, 1에 가까울수록 좋은 성능임을 의미한다[14].

수식 (29)의 NMI(Normalized Mutual Information) 는 정답 군집과 예측 군집 사이의



(a) 점증적 군집화 소요 시간 그래프



(b) 점증적 군집화 소요 시간 그래프

월		01	02	03	04	05	06	07	08	09	10
데이터 개수		3,276	3,982	3,429	3,900	4,712	4,653	4,264	5,018	5,044	6,071
소요 시간	orig	277.98	1304.92	4035.54	5964.13	9363.06	10805.29	12866.52	17641.38	19835.05	26564.70
	ours	28.53	50.98	65.98	81.52	106.38	120.82	148.03	60.64	73.22	101.31
	ours+wcs	29.21	46.49	67.78	87.63	121.47	142.05	164.50	65.52	73.35	106.37

(c) 월별 데이터 개수 및 각 군집화 방식의 점증적 군집화 소요 시간 (시간 단위: sec)

그림 8. 점증적 군집화 실험 결과

MI(Mutual Information)을 정규화(normalize)한 지표이다. MI는 두 분포 사이의 의존 관계(dependence)를 엔트로피(Entropy; H) 값으로부터 수치화한 지표로, NMI는 이러한 MI를 0 이상 1 이하의 값으로 정규화(normalize)한 지표이다. 따라서 NMI가 0에 가까울수록 두 분포가 독립적, 즉, 군집의 성능이 좋지 않다는 것을 의미하고, 1에 가까울수록 두 분포의 상관관계가 높다, 즉, 군집의 성능이 좋다는 것을 의미한다[14].

수식 (30)의 FScore는 정밀도와 재현율을 이용하여 모델의 성능을 0 이상 1 이하의 값으로 수치화하는 지표로, 0이면 정답 군집과 예측 군집이 전부 일치하지 않는다는 것을 의미하며, 1이면 두 군집이 전부 일치함을 의미한다[25].

위의 3가지 지표를 사용하여 기존의 '계층적 다중 뉴스 문서 군집화(orig)[5]'와 제안하는 군집화 방식의 성능을 평가한다. 제안하는 군집화 방식에서는 가중 코사인 유사도 측정 모델에 따른 군집화 성능 개선을 확인하기 위하여 가중 코사인 유사도 측정 모델이 적용되지 않은 군집화 방식(ours), 그리고 가중 코사인 유사도 측정 모델이 적용된 군집화 방식(ours+wcs) 2가지 방식에 대한 성능을 평가한다. orig, ours, 그리고 ours+wcs에 대한 성능 평가 결과는 [표 5]와 같다.

[표 5]에서 볼 수 있듯이, 모든 지표의 모든 레벨에서 가중 코사인 유사도 측정 모델이 적용된 군집화 모듈(ours+wcs)이 가장 좋은 성능을 보이고 있음을 알 수 있다. 이로부터 가중 코사인 유사도 측정 모델이 군집화 모듈의 계층적 이슈 분석 성능을 개선하는 데 긍정적인 영향을 미친다는 것을 알 수 있다.

4.2 점증적 군집화 실험

본 연구에서는 새로운 뉴스 기사가 발생함에 따라 이슈 군집 트리를 업데이트하는 점증적 군집화 과정의 실시간성을 테스트하였다. 실험에 사용한 데이터셋은 구글 검색 엔진[26]을 이용하여 수집한 2012년도 1월부터 10월까지의 뉴스 기사에 대하여 IV의 1.1절의 뉴스 데이터셋과 같은 전처리 과정을 적용하여 총 44,349개를 구축하였다. 실험 방법은 2012년 1월 뉴스 기사를 이용하여 초기 이슈 군집 트리를 생성하고, 2월 뉴스 기사부터 한 달 단위로 점증적 군집화를 진행하며 이슈 군집 트리를 업데이트하였다.

위와 같은 방식으로 실험을 진행한 결과는 [그림 8]과 같다. [그림 8]의 (a)는 본 연구에서 제안하는 가중 코사인 유사도 측정 모델이 적용되지 않은 군집화 방식(ours)과 가중 코사인 유사도 측정 모델이 적용된 군집

화 방식(ours+wcs)에 대한 점증적 군집화 소요 시간 그래프로, 누적되는 뉴스 기사의 개수에 비례하여 소요 시간이 증가하는 것 알 수 있다. 특히, 8월 뉴스 기사에 대하여 점증적 군집화가 진행될 때 소요 시간이 큰 폭으로 감소하는 데, 이는 III의 2.2절에서 설명한 close 기능 때문에 발생하는 현상으로, 실제로 ours에서는 7월에 생성된 군집 114개 중 102개가, ours+wcs에서는 135개 중 129개가 close되었다. 이에 따라 7월까지 생성된 이슈 군집 중, 불필요한 이슈 군집을 close함으로써 8월 뉴스 기사에 대한 점증적 군집화 과정에서 연산 시간이 크게 감소한 것으로 보인다. [그림 8]의 (b)는 기존 군집화 방식(orig)의 점증적 군집화 소요 시간 그래프로, ours와 ours+wcs와는 다르게 뉴스 기사가 증가함에 따라 군집화 소요 시간이 기하급수적으로 증가하는 것을 알 수 있다. [그림 8]의 (c)는 각 군집화 방식에 따른 점증적 군집화 소요 시간을 월별로 작성한 표로, orig의 소요 시간이 ours와 ours+wcs보다 훨씬 오래 걸리는 것을 알 수 있다.

V. 결론

본 연구에서는 뉴스 기사로부터 이슈를 준 실시간으로 분석하기 위한 계층적 군집화와 점증적 군집화 방식을 제안하였다. 계층적 군집화에서는 단어 군집 기반 문서 표현 방식을 이용하여 뉴스 기사 증가에 따른 메모리 비효율성 문제 및 연산 시간 증가 문제를 해결하였으며, 계층적 군집화의 성능을 높이기 위하여 유의어 단어 쌍 데이터셋 구축, 워드 임베딩 모델 선정, 그리고 샴 신경망 기반 가중 코사인 유사도 측정 모델 학습 실험을 진행하고 이를 군집화 방식에 적용한 다음, 구축한 군집 테스트셋을 이용하여 계층적 군집화의 성능을 정량적으로 평가·비교하였다. 그 결과, 3가지 군집 성능 평가 지표에서 가중 코사인 유사도 측정 모델이 적용된 군집화 모듈이 가장 우수한 성능을 보였으며, 특히 기존의 계층적 다중 뉴스 문서 군집화와 비교하였을 때, ARI level 2에서는 약 0.04, NMI level 1에서는 약 0.26의 성능이 향상되었다. 점증적 군집화에서는 새로운 뉴스 기사 데이터셋을 구축하고 이에 대하여 월 단위 점증적 군집화를 진행함으로써 점증적 군집화의 준 실시간성을 확인하였고, close 기능을 통하여 점증적 군집화 과정에서 이슈 군집 트리를 업데이트하는 데 소요되는 시간이 크게 감소되는 것을 확인하였다. 하지만, 이슈 분석의 정확성과 관련된 계층적 군집화 성능 측면에서 가중 코사인 유사도 측정 모델을 적용함으로써 향상되는 성능 증가폭이 1%p에서 5%p로 미미하고, 가중 코사인 유사도 측정 모델 자체 성능 또한 정확도가 78.95%로 낮은 편이다.

본 논문에서 제안하는 준 실시간 뉴스 이슈 분석을 위한 계층적·점증적 군집화는 급증하는 뉴스 기사 데이터로부터 이슈를 실시간에 가깝게 분석할 수 있음을 입증하였다. 이는 뉴스 기사뿐만 아니라 트위터, 웨이보 등 많은 사용자들이 실시간으로 글을 게시하는 소셜 네트워크 서비스에도 적용되어 소셜 네트워크 서비스 상의 이슈를 분석할 수 있음을 시사한다.

향후 연구로 샴 신경망 내의 은닉층에 Recurrent Neural Network(RNN), Convolutional Neural Network(CNN) 등 다양한 신경망 구조를 적용하여 가중 코사인 유사도 측정 모델의 성능을 더욱 높인다. 또한, 학습된 가중 코사인 유사도 측정 모델을 단어 군집 기반 문서 표현 과정뿐만 아니라 Single-pass Online Clustering, Event Merging 등 두 문서 사이의 유사도를 측정하는 다른 여러 과정에도 확장 적용함으로써 계층적 군집화의 성능을 개선한다.

참 고 문 헌

[1] https://www.bigkinds.or.kr/, 2020.05.12

[2] P. Zhou, Z. Cao, B. Wu, C. Wu, and S. Yu, "EDM-JBW: A novel event detection model based on JS-ID'Forder and Bikmeans with word embedding for news streams," The Journal of Computational Science Vol.28, pp.336-342, 2018.

[3] Y. Li, L. Liu, X. Bai, H. Cai, W. Ji, D. Guo, Y. Zhu, "Comparative study of discretization methods of microarray data for inferring transcriptionalregulatory networks," BMC

Bioinform, Vol.11, 2010.

[4] L. Hu, B. Zhang, L. Hou, and J. Li, "Adaptive online event detection in news streams," Knowledge-Based Systems, Vol.138, pp.105-112, 2017.

[5] 유홍현, 이승우, 고영중, "실시간 뉴스 기반의 이슈 분석을 위한 점증적 군집화 및 다중 문서 요약," 정보과학회논문지, 제46권, 제4호, pp.355-362, 2019.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv, 2013.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," In Transactions of the Association for Computational Linguistic, Vol.5, pp.135-146, 2017.

[8] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vetors for word representation," In EMNLP, 2014.

[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol.1, pp.2227-2237, 2018.

[10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," In NAACL-HLT, 2019.

[11] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," In NeurIPS, 2019.

[12] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," In arXiv, 2014.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," The Journal of Machine Learning Research, Vol.3, pp.993-1022, 2003.

[14] Kevin P. Murphy, *Machine learning: a probabilistic perspective*, MIT press, 2012

[15] C. Wartena and R. Brussee, "Topic Detection by Clustering Keywords," In 2008 19th International Workshop on Database and Expert Systems Application, pp.54-58, 2008.

[16] Z. J. Mansoor, P. Elham, and J. Robert, "A method of learning weighted similarity function to improve the performance of nearest neighbor," In Information Sciences, Vol.179, No.17, pp.2964-2973, 2009.

[17] H. V. Nguyen and L. Bai, "Cosine Similarity Metric Learning for Face Verification," In Asian Conference on Computer Vision, pp.709-720, 2011.

[18] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol.1, pp.539-546, 2005.

[19] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-Shot Image Recognition," In Proceedings of the 32nd International Conference on Machine Learning, Vol.37, 2015.

[20] https://news.naver.com/, 2020.05.12

[21] https://bitbucket.org/eunjeon/mecab-ko-dic/src/master/, 2020.05.12

[22] https://opendict.korean.go.kr/service/openApiInfo

[23] 강형석, 양장훈, "한국어 단어 임베딩 모델의 평가에 적합한 유추 검사 세트," 디지털콘텐츠학회논문지, 제19권, 제10호, pp.1999-2008. 2018(8).

[24] https://fasttext.cc/docs/en/crawl-vectors.html, 2020.05.12

[25] Z. Ying and K. George, "Evaluation of Hierarchical Clustering Algorithm for Document Datasets," In Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp.515-524, 2002.

[26] https://news.google.com/, 2020.05.12

저 자 소 개

김 호 용(Hoyong Kim) 준회원

- 2018년 2월 : 아주대학교 소프트웨어학과(공학사)
- 2018년 3월 ~ 현재 : 과학기술연합대학원대학교 빅데이터과학과 석사과정

〈관심분야〉 : 자연언어 처리, 텍스트 마이닝, 딥 러닝

이 승 우(Seungwoo Lee) 정회원

- 1997년 2월 : 경북대학교 컴퓨터공학과(공학사)
- 1999년 2월 : 포스텍 컴퓨터공학과(공학석사)
- 2005년 8월 : 포스텍 컴퓨터공학과(공학박사)
- 2006년 3월 ~ 현재 : 한국과학기술정보연구원 책임연구원

〈관심분야〉 : 자연언어 처리, 텍스트 마이닝, 딥 러닝, 기계학습

장 홍 준(Hong-Jun Jang) 정회원

- 2008년 2월 : 고려대학교 컴퓨터교육과(이학사)
- 2019년 2월 : 고려대학교 컴퓨터학과(공학박사)
- 2019년 6월 ~ 현재 : 한국과학기술정보연구원 선임연구원

〈관심분야〉 : 지식정보시스템, 데이터베이스, 데이터마이닝

서 동 민(Dongmin Seo) 정회원

- 2002년 2월 : 충북대학교 정보통신공학과(공학사)
- 2004년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2008년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2008년 3월 ~ 2010년 2월 : 한국과학기술원 전산학과 연수연구원
- 2010년 2월 ~ 현재 : 한국과학기술정보연구원 국가과학기술데이터본부 책임연구원

〈관심분야〉 : 빅데이터 분석, XML, 시맨틱웹, 이동객체 데이터베이스, 센서 네트워크, 네트워크 분석