연구배경

20182806 서동혁

최근 컴퓨터를 이용해 사람의 언어를 처리하는 자연어 처리(NLP)는 감정분석, 카테고리 세분화, 관련 상품 추천 등 여러 분야에서 사용되어지고 있고 우리의 일상에서 자주 쓰이는 애플의 Siri나 구글 번역기가 NLP의 대표적인 응용사례이다.

국내 최대 통신사이자 디지털 플랫폼 기업인 KT는 인공지능(AI) 전문인력을 양성하기 위해 한양대학교와 손잡고 석사 과정의 'AI 응용학과'를 개설했다고 2일 밝혔다. KT는 디지털플랫폼기업(디지코)으로 도약을 선언한 KT가 AI 우수인재를 확보해 미래 성장엔진인 AI 기술력과 전문성을 강화하기 위한 결정이며 이번에 새로 개설되는 'AI 응용학과'는 자연어처리가 교육과정에 포함되어 있다고 한다. 이는 기업에서 자연어처리 기법을 중요하게 생각하고 있음을 보여주는 사례이다.

시간이 지날수록 온라인 플랫폼에서 리뷰나 댓글의 중요도가 커지며 이를 활용하는 방법들이 증가하는 추세이다. E-Commerce 업계는 고객 리뷰의 힘에 주목해 이를 적극적으로 활용하기 위해 리뷰 자체를 콘텐츠화해 제공하거나 양질의 리뷰 확보를 위한 프로모션이나 커뮤니티, 전문 시스템을 도입하는 등 고객 참여 활동에 적극적으로 나서고 있다. 또한 한국관광공사는 우리나라 문화관광 유튜브 인기 영상 외국어 댓글을 통해 영상 분야, 조회수 및 댓글수 등 6개 유형별 인기 영상 콘텐츠 총 375건, 댓글 51만3894건을 분석해 키워드를 뽑아냈다. 이후 이를 '방한관광 홍보 활용을 위한 유튜브 소셜데이터 분석' 보고서로 공개했다.

따라서 이와 관련해 파이썬을 활용해 온라인 플랫폼 중 하나인 유튜브에서 크리에이터 '김진짜'의 영상 댓글(리뷰)들로 각 영상별 **키워드를 추출해** 영상의 핵심내용 및 주요 관심 포인트들을 유추해볼 수 있는 방법론을 제시해보고자 한다.

관련연구

Pyhon(파이썬)프로그래밍 언어는 Konlpy 와 Soynlpy 2가지의 한국어 자연어처리 패키지를 제공하는데 이에 대해 설명을 해보고자 한다. 첫번째로 Konlpy는 다양한 형태소 분석, 태깅 라이브러리를 파이썬에서 쉽게 사용할 수 있도록 모아놓은 라이브러리로써 5가지의 분석기가 있다. 각각의 형태소 분석기마다 품사 태깅 방식은 조금씩 차이가 있으며 단어 사전에 등록된 단어들을 기반으로 토큰화를 실시한다는 것이 중요한 포인트이다.

아래의 <표1>은 Konlpy의 다양한 형태소 분석기들을 정리해놓은 표이다.

< 표 1>

Hannanum	KAIST Semantic Web Research Center 개발
Kkma	서울대학교 IDS 연구실 개발, 속도 느림, 상세한 품사 정보
Komoran	Shineware 개발
Mecab	일본어용 형태소 분석기를 수정
Okt	오픈소스 한국어 분석기, 띄어쓰기가 어느정도 되어있는 문장을 빠르게 분석.

두 번째로 Soynlp는 사전에 등록된 단어를 바탕으로 형태소를 분석하는 것이 아닌 말뭉치의 Cohension Score등을 통해 학습한 단어의 경계로 형태소를 분석하는 비지도학습 접근법이다.

Cohension Score란 문자열을 글자단위로 분리하여 부분문자열을 만들 때 왼쪽부터 문맥을 증가시키면서 각 문맥이 주어졌을 때 그 다음 글자가 나올 확률을 계산한 누적곱의 값으로 쉽게 말하자면 단어를 구성하는 글자들이 얼마나 자주 등장하는지 알 수 있는 값이다.

$$cohesion(c_{0:n}) = \left(\prod P(c_{0:i+1}|c_{0:i})\right)^{n-1}$$

< 그림 1>

< 그림 1> 은 Cohension Score를 구하는 수식으로써 단어의 경계에 가까워질수록 값은 커지고 경계를 넘어서면 값이 줄어든다.

L-Tokenizing란 한국어의 경우 하나의 문자열이 'L-Token + R-Token' 구조로 이루어지는데 어절의 왼쪽에 위치한 부분문자열 중 단어 점수가 가장 높은 부분을 선택한뒤 어절을 L + R로 나누는 방법이다.

< 표 2>

부분문자열	Cohension Score
아이오아	0.2
아이오아이	**0.3**
아이오아이는	0.24

< 표 2 >의 경우 L-token으로는 아이오아이가 된다.

TF-IDF 가중치는 TF값과 IDF값을 곱한 것으로, TF값은 한 문서 내에서 특정 단어가 출현한 빈도수를 의미하며 이는 한 문서 내부의 단어 출현 빈도를 모든 단어의 총 출현 횟수로 나누어 정규화한 형태이다.

IDF 값은 문서 집합에 포함되어 있는 문서 수를 특정 단어가 나타난 문서의 수로 나눈 것으로 상대적으로 많은 문서에 출현한 단어의 IDF 값은 작게, 한쪽으로 편중하여 나타난 단어의 IDF값은 크게 만든다. 단어가 등장하는 문서의 수(df(t)로 약칭)는 Zero Division을 방지하기 위해 (df(t) +1)로 변환해 사용한다.

결론적으로 TF값은 주어진 단어가 문서 내에서 많이 출현할수록 상대적으로 더 중요하다는 가정을 반영하고, IDF값이 작은 단어는 보편적인 단어로 반대로 값이 큰 단어는 문서 내에서 주요 의미를 가지는 단어로 분별하기 위한 의도를 반영하므로 이 둘을 곱해준 TF-IDF 값은 특정 문서 내에서 어떤 단어가 중요한지를 나타내주는 통계적 수치이다.

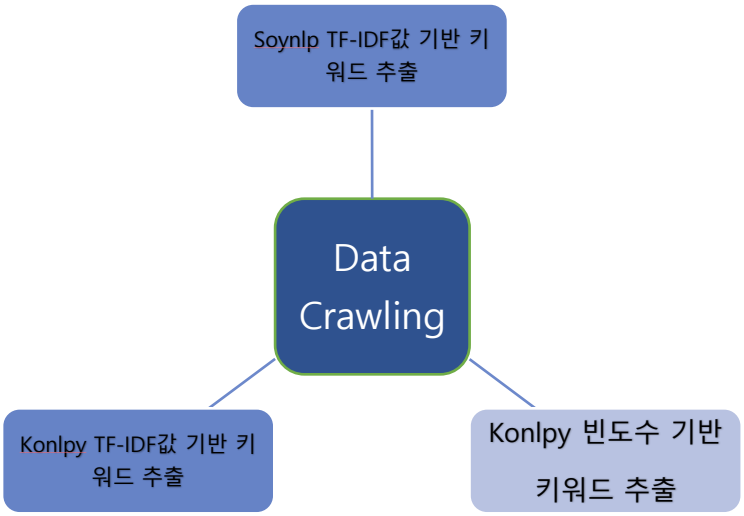
자세한 수식은 아래 <표3>으로 확인할 수 있다.

< 표 3>

TF값	$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$ $n_{i,j}$: 단어 ti가 문서 dj에서 출현한 회수 $\sum_k n_{k,j}$: 문서 dj에서 모든 단어가 출현한 회수
IDF값	$idf_i = \log \frac{\lvert D \rvert}{\lvert \{d_j \mid t_j \in d_j\} \rvert}$ $\lvert D \rvert$: 문서집합에 포함되어 있는 문서의 수 $\lvert \{d_j \mid t_j \in d_j\} \rvert$: 단어 t_j가 등장하는 문서의 수
TF-IDF 가중치	$TFIDF_{i,j} = tf_{i,j} \times idf_i$

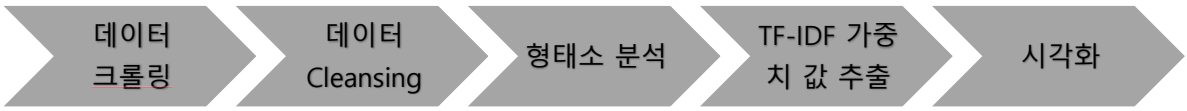
제안 방법론

<그림 2>



<그림2>는 이번 실험에서 진행하게 될 3가지 방법론이며 1. Konlpy를 활용해 키워드를 추출하는데 빈도수만을 이용해 키워드를 추출하는 방법, TF-IDF 값을 기반으로 추출하는 두 가지 방법 2. Soynlp를 사용해 TF-IDF 값을 기반으로 추출하는 방법으로 진행해 각 결과값들을 비교해 보고자 한다. 여기에서는 도출된 TF-IDF값을 통해 알맞은 키워드를 골라내는 것이 목표이긴 하지만 Konlpy에서만 성능의 비교를 위해 빈도수만을 이용해 키워드를 추출해 볼 계획이다.

<그림 2-1>

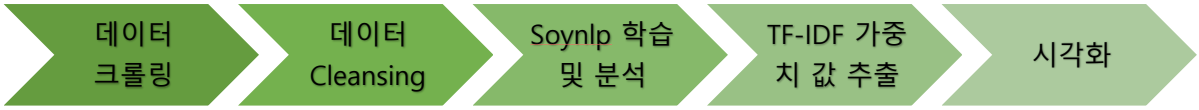


<표 4>

Kkma 사용	Okt 사용
[('안녕', 'NNG'), ('나', 'VV'), ('는', 'ETD'), ('하늘색', 'NNG'), ('과', 'JC'), ('딸기', 'NNG'), ('를', 'JKO'), ('좋아하', 'VV'), ('어', 'ECS')]	[('안녕', 'Noun'), ('나', 'Noun'), ('는', 'Josa'), ('하늘색', 'Noun'), ('과', 'Josa'), ('딸기', 'Noun'), ('를', 'Josa'), ('좋아해', 'Adjective')]

<그림 2-1>은 Konlpy를 활용하였을 때의 전체 개요로써 먼저 데이터 크롤링 및 분석에 필요없는 문자(!,?등)을 제거해주는 cleansing 작업을 수행하고 이후 형태소 분석 및 TF-IDF 가중치를 계산, 시각화 방법에는 WordCloud와 Matplotlib을 활용한 그래프로 진행할 예정이다. Konlpy의 형태소 분석기에는 다양한 분석기들이 있으나 <표 4>와 같이 이번 실험에서는 섬세한 품사 정보를 확인할 수 있는 'Kkma'와 댓글들의 띄어쓰기가 어느정도 되어있다고 생각되어서 'Okt'를 사용해 성능을 비교해 본다.

<그림 2-2>



<그림 2-2>는 Soynlp에서의 전체 개요로써 데이터 크롤링 및 cleansing을 해주고 이후에는 추가적으로 앞서 관련 연구에서 언급했던 것처럼 말뭉치 데이터를 그대로 넣어주어 학습을 시키고 Cohension score를 기반으로 L-Tokenizer를 적용해 형태소를 분리시켜 형태소를 분석한다. 나머지는 Konlpy의 방식과 똑같다. 이번 실험에서 Soynlp를 사용하는 이유는 유튜브 댓글은 트렌드에 굉장히 민감하고 젊은 세대층들이 많이 사용하므로 기존 사전에 등록되어 있지 않은 미등록 단어, 신조어 처리 문제들을 해결하기 위한 방안이다.

실험내용

1. 크롤링

앞으로의 실험방법들의 구현은 모두 파이썬으로 실행하며 크롤링 할 때는 Selenium과 BeautifulSoup 2개의 라이브러리를 사용하는데 유튜브 댓글을 크롤링 하게 될 때에는 페이지의 스크롤을 끝까지 내려주어 댓글이 화면에 모두 출력되게 해야 하는데 이는 Selenium으로 구현할 수 있고, 댓글들이 모두 로딩되면 BeautifulSoup으로 데이터를 crawling하게 된다. 이번 실험에서는 총 **4개의 영상**의 댓글들을 긁어와 분석해본다. Data cleansing 작업은 re모듈을 사용했고 제외한 단어들은 <표5> 와 같다.

삭제한 문자	비고
이모티콘	이모티콘을 텍스트로 변경함
영상 시간대 ex) 5:1	유튜브 특성상 댓글에는 해당 영상의 장면을 시간대를 표기함으로써 빠르게 찾아갈 수 있음. 그렇지만 키워드를 추출하는데에는 필요 없음.
'ㅋㅋ,ㅎㅎ'등 과 같은 자음으로 이루어진 표현	
?!., ₩n 등의 각종 특수문자	

<표5>

2-1. 형태소 분석(Konlpy)

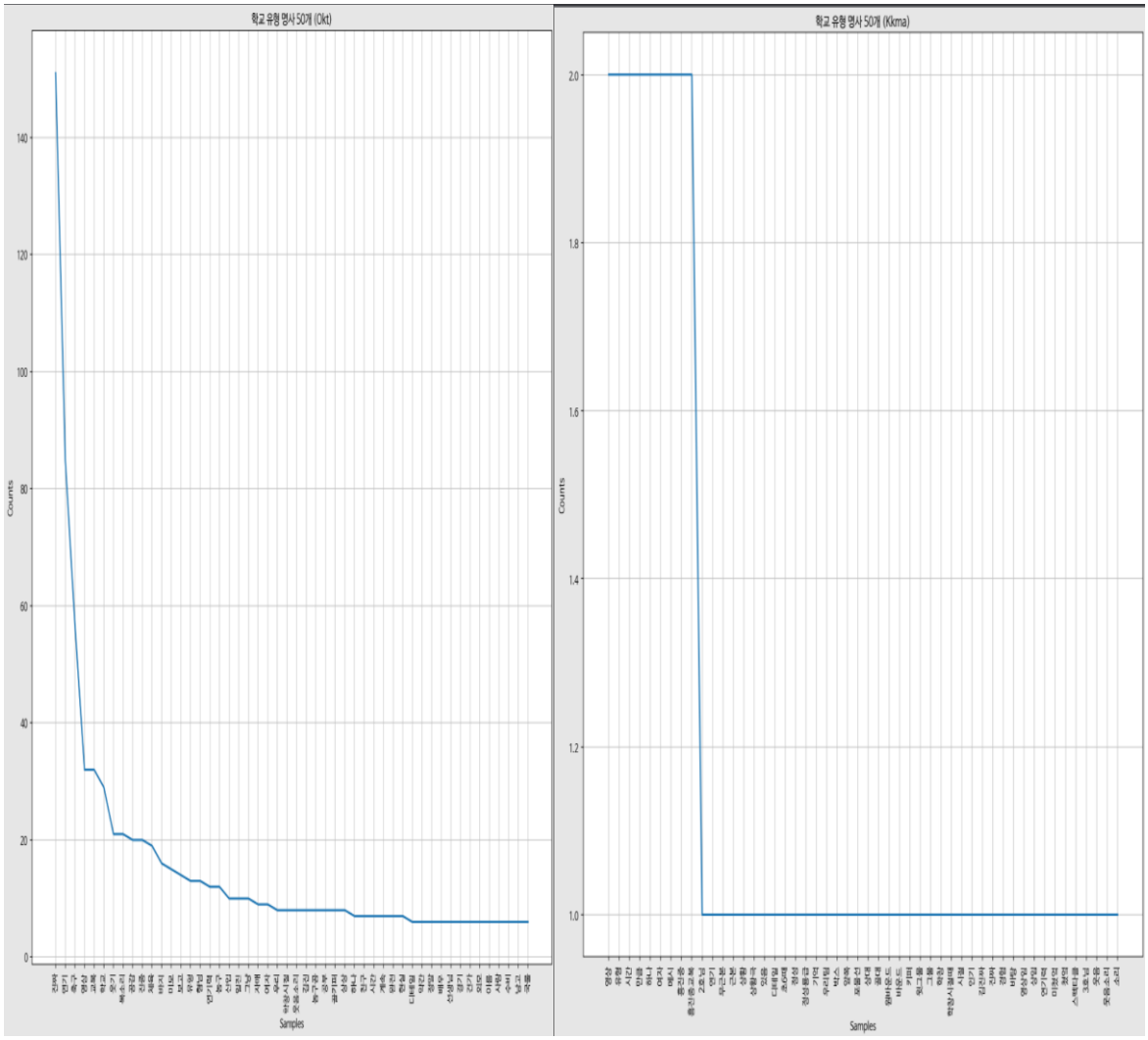
우리가 알고 싶은 것은 각 영상의 내용을 유추해볼 수 있는 '키워드'임으로 각 영상 댓글을 Kkma와 Okt의 분석기로 명사만을 추출하여 알아보도록 했고 단어의 최다 빈도 순으로 50개를 plot 그래프로 시각화 하여 대략적인 빈도수를 알아보았다. 이후 빈도순으로 뽑은 100개의 단어들을 WordCloud 형태로 시각화 하여 한눈에 알아볼 수 있도록 했다.

<그림3-1>은 영상 4개 중 하나의 결과로써 왼쪽이 Okt, 오른쪽이 Kkma로 분석한 결과이다. kkma가 okt에 비해서 비교적 명사들을 정확하게 분류하는 걸 볼 수 있다. 그 예로 실제 댓글 내용인 '흥진중'을 Kkma는 명사로 알맞게 분류했으나 Okt는 '진중'이라고 분류해버렸다. 하지만 <그림 3>을 보면 왼쪽은 Okt 형태소 분석기로 뽑은 명사 50개의 빈도그래프, 오른쪽은 Kkma 형태

소 분석기로 뽑은 빈도 그래프인데 Okt에 비해 Kkma는 단어의 빈도가 2, 1로 이루어진다. 왜 그럴까? Kkma는 품사를 너무 세세하게 분류해버려서 중복되는 단어들을 찾기가 어렵기 때문이다.

그래서 앞으로 Kkma를 활용해서는 단어별 빈도 및 TF-IDF를 계산하기에 적합하지 않다고 판단해 형태소 분석기는 Okt만을 활용하기로 했다.

<그림 3>



<그림 3-1>

2-2. 형태소 분석(Soynlp)

<그림 3-2>처럼 Soynlp는 4개의 영상별 댓글들(말뭉치)들을 각각 따로 WordExtractor에 넣어준 뒤 Ltokenizer로 token화를 진행시켰으며 마찬가지로 명사만을 추출했는데 이 때는 LRNounExtractor 보다 성능이 더 좋다고 알려진 soynlp.noun의 LRNounExtractor_v2를 이용했다.

```
84] # 학교에서 video 명사추출

    from soynlp.noun import LRNounExtractor_v2
    noun_extractor = LRNounExtractor_v2()
    nouns_4 = noun_extractor.train_extract(sorted_school)
    nouns_4
```

```
# WordExtractor
from soynlp.word import WordExtractor

word_extractor = WordExtractor(
    min_cohesion_forward=0.05,
    min_right_branching_entropy=0.0)
word_extractor.train(school_soynlp) # 학교에서임
words = word_extractor.extract()

# 토큰화 작업
from soynlp.tokenizer import LTokenizer
scores = {word:score.cohesion_forward for word,score in words.items()}
tokenizer = LTokenizer(scores = scores )

school_sy_token = tokenizer.tokenize(school_soynlp,flatten=False)
school_sy_token
```

<그림 3-2>

3. TF-IDF 가중치 값 추출

Konlpy 와 Soynlp 모두 TF-IDF값을 계산할 때 각각의 명사만을 넣어주었더니 출현빈도가 고려되지 않아 빈도수 만큼 해당하는 명사들을 만들어준 뒤 계산을 했고 ((EX) '너'의 빈도수 50번 => '너' 50개 생성) 이를 csv 파일에 column은 문서, index는 단어인 dataframe 형태로 저장했다.

<그림3-2>의 왼쪽이 soynlp로 실행한 TF-IDF값, 오른쪽이 konlpy의 okt로 실행한 TF-IDF값이다.

```
pd.read_csv('tf-idf값_soynlp.csv',encoding = 'utf-8',index_col= 0)
```

	극혐	조기축구	축구볼때	학교에서
2호님	0.000000	0.000000	0.000000	0.859441
3호님	0.114613	0.043408	0.168718	0.114322
4번	0.000000	0.000000	0.121242	0.000000
7번	0.035651	0.000000	0.031863	0.000000
8번	0.000000	0.054651	0.053105	0.000000
...	...	...	...	...
형	0.242710	0.122988	0.414764	0.085008
혼자	0.090437	0.000000	0.000000	0.000000
혼자형	0.077517	0.000000	0.000000	0.000000
회비	0.096896	0.000000	0.000000	0.000000
흥진중	0.000000	0.000000	0.000000	0.067407

150 rows × 4 columns

```
pd.read_csv('tf-idf값_okt.csv',encoding = 'utf-8',index_col= 0)
```

	극혐	조기축구	축구볼때	학교에서
가발	0.000000	0.000000	0.123174	0.000000
강두	0.108344	0.000000	0.000000	0.000000
개그	0.000000	0.000000	0.057964	0.000000
개그맨	0.042710	0.000000	0.057125	0.000000
건가	0.000000	0.000000	0.000000	0.049132
...	...	...	...	...
형님	0.000000	0.065314	0.041623	0.067947
혹시	0.054172	0.000000	0.000000	0.000000
혼자	0.162515	0.000000	0.000000	0.000000
회비	0.049828	0.066008	0.000000	0.000000
회수	0.058686	0.000000	0.000000	0.000000

136 rows × 4 columns

<그림 3-3>

4. 시각화

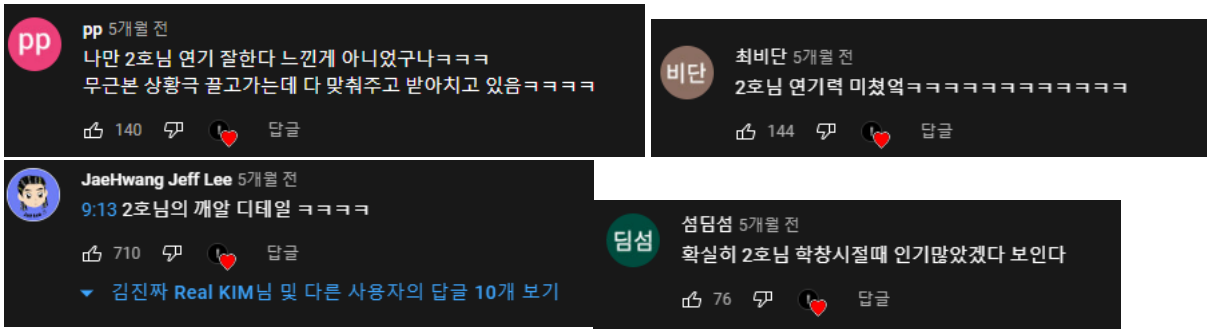
시각화 방법으로는 WordCloud 및 bar plot, line plot등을 실행했다.

결론

결론의 사진에서 왼쪽이 Soynlp기반으로 TF-IDF를 추출해 시각화를 진행한 것, 오른쪽이 Konlpy의 Okt 기법을 적용한 것이다. 물론 오른쪽만 보더라도 키워드로 뽑아낸 것들로 충분히 학교라는 곳에서 무언가 연기를 했다는 것을 알 수 있긴 하지만 실제 댓글 내용인 <그림 4>를 보면 영상에서는 사람들이 '2호님'의 연기력, 미모 등 '2호님'과 관련된 것에 큰 반응을 보이고 있다. 이번 실험에서 사용된 말뭉치데이터는 약 1000~2000개 정도밖에 되지 않아서 '누','많' 등 별의미가 없는 명사들이 뽑히긴 했으나 만약 더 많은 말뭉치로 학습을 했다면 우리가 바라는 이상적인 결과를 도출하는데 도움이 될 것이다.

실험을 통해 1) 미등록 단어들 및 신조어들을 처리할 수 있는 말뭉치 기반 학습방법인 Soynlp로 유튜브 댓글을 분석하는 것이 Konlpy보다 더 효과적이다. 2) 형태소를 정확하게 분석하는 것 (세세하게 분석)하는 것이 꼭 좋은 것만은 아니다. 3) 말뭉치 데이터가 더 많으면 많을수록 분석의 정확도가 상승할 수 있을거라고 예상된다. 라는 3개의 결론을 도출할 수 있었다.

<그림4>



인용 자료

"Konlpy API". (날짜 정보 없음). https://konlpy.org/ko/latest/.에서 검색됨

"MZ세대 부모의 똑똑한 육아법, 구독 경제 스타트업 활용". (2022년 4월 18일). Venturesquare news: https://www.venturesquare.net/853171에서 검색됨

국제섬유신문. (2022년 4월 18일). "이커머스 업계, 고객 목소리 담은 '리뷰 콘텐츠' 마케팅 주목". 국제섬유신문: https://www.itnk.co.kr/news/articleView.html?idxno=69143에서 검색됨

신승훈. (2021년 09월 02일). "KT-한양대, 석사과정에 'AI 응용학과' 개설...졸업 후 KT 연구원으로 근무". 아주경제: https://www.ajunews.com/view/20210902090646575에서 검색됨

이성직김한준. (2009). TF-IDF의 변형을 이용한 전자뉴스에서의 키워드 추출 기법. 서울시립대학교 전자전기 컴퓨터 공학부.

이현주. (2022년 2월 16일). "역시 'BTS'...한국 문화관광 유튜브 댓글에도 파급력". 미디어뉴스: https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=103&oid=003&aid=0011005801에서 검색됨