

공학박사학위논문

미등록단어 문제와 데이터 부족 현상을
해결하기 위한 비지도학습 토크나이저와 추출
기반 문서 요약 기법

Unsupervised Korean Tokenizer and Extractive Document
Summarization to Solve Out-of-Vocabulary and Dearth of
Data

2019년 6월

서울대학교 대학원

산업공학과

김현중

미등록단어 문제와 데이터 부족 현상을 해결하기 위한 비지도학습 토크나이저와 추출 기반 문서 요약 기법

Unsupervised Korean Tokenizer and Extractive
Document Summarization to Solve Out-of-Vocabulary
and Dearth of Data

지도교수 조 성 준

이 논문을 공학박사 학위논문으로 제출함

2019년 6월

서울대학교 대학원

산업공학과

김 현 중

김현중의 공학박사 학위논문을 인준함

2019년 7월

위 원 장 박 종 현 (인)

부위원장 조 성 준 (인)

위 원 이 경 식 (인)

위 원 강 필 성 (인)

위 원 고 태 훈 (인)

초록

자연어처리는 사람의 언어를 컴퓨터가 이용할 수 있는 형태의 정보로 변환하거나 이를 이용하는 과업들로 이뤄진 분야이다. 토크나이징은 문장을 단어나 형태소와 같이 분석의 단위로 분해하는 과업으로, 다른 자연어처리 과업의 입력 데이터를 처리하는 기초 과업이다. 토크나이저의 성능이 좋지 않을 경우 문서 요약이나 토픽 모델링과 같은 다른 자연어처리 과업의 품질이 저하된다. 문서 요약 과업은 키워드나 핵심 문장을 통하여 문서 집합의 내용을 요약하는 과업으로, 대량의 문서 집합에 대한 탐색을 편리하게 도와주거나 문서를 인덱싱 하는데 이용될 수 있다.

그러나 자연어처리 과업은 다음의 어려움을 지닌다. 첫째, 미등록단어 문제라 불리는 현상으로, 학습 데이터에 등장하지 않은 단어를 제대로 인식하지 못할 수 있다. 둘째, 각 과업에 적합한 학습 데이터를 마련하기 어렵다. 셋째, 텍스트 데이터에는 띠어쓰기 오류 및 철자법 오류에 의하여 잘못된 자연어처리 결과가 야기될 수 있다. 영어와 달리 한국어에는 띠어쓰기와 철자법 오류가 빈번하며, 이로 인하여 단어의 경계 구분이 어려운 경우가 발생한다. 위의 어려움들은 서로가 연결되어 있다. 데이터 내 오류는 미등록단어 문제를 발생시키며 이를 해결하는 모델을 구축하기 위해서는 추가적인 학습 데이터가 필요하다. 이러한 어려움은 토크나이저와 문서 요약 외 다른 자연어처리 과업들에서도 공통적으로 발생한다.

이 논문에서는 한국어 자연어처리 과업에서 발생하는 어려움을 해결하기 위하여 한국어의 구조적 특징을 이용하는 비지도학습 자연어처리 방법들을 제안한다. 비지도 학습 방법은 학습 데이터를 이용하지 않기 때문에 다양한 도메인의 자연어처리 과업에 적용하기 용이하다. 또한 한국어의 구조적 특징은 비지도학습 기반 모델의 사전 지식

역할을 하여, 데이터로부터 효율적으로 정보를 학습할 수 있도록 도와준다.

이 논문에서는 비지도학습 한국어 자연어처리에 적합한 어절 구조인 L + [R] 와 이를 이용하는 다섯 가지 비지도학습 자연어처리 방법을 제안한다. 첫째, 미등록단어 문제를 해결하기 위하여 한국어 어절의 구조를 기반으로 작동하는 비지도학습 기반 한국어 토크나이저를 제안한다. 제안한 토크나이저는 Word Piece Model 보다 좋은 분류 성능과 학습 데이터를 이용하는 형태소 분석기와 비슷한 단어 인식 성능을 보였다. 둘째, L + [R] 구조를 기반으로 명사를 추출하는 방법을 제안하였으며, 이 역시 학습 말뭉치와 단어 사전을 이용하는 형태소 분석기보다도 뛰어난 명사 인식 능력을 보였다. 셋째, 단일 주제의 문서 집합 요약을 위한 키워드 및 핵심 문장 추출 방법을 제안하였다. 이 방법은 단어 추출 과정이 내제되어 있으며 키워드의 미등록단어 문제에 강건하다. 또한 중복되지 않는 문장들로 핵심 문장을 구성할 수 있다. 넷째, 다양한 주제로 구성된 문서 집합을 요약하기 위한 문서 군집화 기반 키워드 추출 방법을 제안하였다. 이 방법은 효율적인 문서 군집화를 위하여 초기화 과정을 개선하였으며, 개선된 Spherical k-means 방법은 기존의 알고리즘보다 수천배 빠른 초기화 계산 속도를 보였다. 또한 군집화 결과인 군집 중심값만을 이용하여 각 군집의 키워드를 추출하기 때문에 추가의 문서 요약 모델을 학습할 필요가 없다. 다섯째, 뉴스와 같이 시계열 형식으로 발생하는 문서 집합을 요약하는 방법을 제안하였다. 이 방법은 시계열 구분 방법을 이용하여 문서 집합의 주제가 변하는 시점을 기준으로 구간을 분리하며, 구간 별 키워드와 핵심 문장을 추출하여 구간 내 문서 집합을 요약한다. 제안된 문서 군집화 기반 키워드 추출 방법과 시계열 형식의 문서 집합 요약 방법은 한국어가 아닌 다른 언어에도 적용될 수 있다.

지도학습 기반 머신러닝 모델들은 다양한 과업에서 높은 정확도를 보여주지만, 학습 데이터에 대한 편향성 때문에 모델이 적용될 데이터에 적합하도록 조정이 필요하다.

이와 반대로 비지도학습 기반 방법은 모델이 적용될 데이터로부터 정보를 추출하며, 지도기반 모델들의 편향성 문제를 완화하는데 이용될 수 있다. 이 논문에서 제안하는 방법들은 비지도학습 기반으로만 작동함에도 불구하고 지도학습 기반 방법보다 좋거나 비슷한 성능을 보인다. 그러므로 제안한 방법과 지도학습 기반 방법을 상호 보완적으로 이용된다면 높은 정확도와 학습 데이터에 대한 편향성이 적은 모델로 발전할 수 있다.

주요어: 한국어 자연어처리, 비지도학습 토크나이저, 명사 추출, 키워드 추출, 핵심 문장 추출, 문서 군집화, 군집화 레이블링

학번: 2013-30314

목차

초록	i
목차	vii
표 목차	x
그림 목차	xii
제 1 장 서론	1
1.1 한국어의 구조	5
1.2 단어 임베딩 (Word embedding)	6
1.3 순차적 레이블링 (Sequential labeling)	10
1.4 머신 러닝 기반 한국어 품사 판별	17
1.5 문서 요약	20
1.5.1 키워드 추출을 이용한 토픽 레이블링	21
1.5.2 그래프 랭킹 기반 키워드와 핵심 문장 추출	23
1.5.3 딥러닝 모델을 이용한 요약 기반 문서 요약	25
제 2 장 단어 추출 기법을 이용한 미등록단어 문제 해결 및 이를 이용한 한국어 토크나이저	27
2.1 서론	27
2.2 관련 연구	29

2.3	비지도기반 한국어 단어 추출 및 이를 이용한 토크나나이저	32
2.3.1	한국어 어절의 구조 : L + [R]	32
2.3.2	음절 단위의 언어 모델을 이용한 단어 점수	34
2.3.3	단어 점수를 이용하는 비지도학습 토크나이저	35
2.4	성능 평가	38
2.4.1	영화평을 이용한 긍부정 분류 성능 평가	40
2.4.2	메타 데이터를 이용한 고유 명사 재현 능력 평가	41
2.4.3	단어 임베딩을 이용한 유사 단어 검색 성능 평가	42
2.5	결론	44
제 3 장	한국어 어절 구조를 이용한 통계 기반 명사 추출	46
3.1	서론	46
3.2	관련 연구	48
3.3	한국어 어절의 L + [R] 구조를 이용한 명사 추출	50
3.3.1	L-R 그래프를 이용한 명사 추출	50
3.3.2	세종 말뭉치를 이용한 명사 판별 분류기 학습	54
3.4	성능 평가	57
3.4.1	세종 말뭉치를 이용한 성능 평가	57
3.4.2	뉴스 기사와 온라인 문서를 이용한 성능 평가	59
3.5	결론	62
제 4 장	단일주제 문서 집합 요약을 위한 그래프 랭킹 기반 키워드와 핵심	
	문장 추출	68
4.1	서론	68

4.2	관련 연구	70
4.3	토크나이저를 이용하지 않는 키워드 및 핵심 문장 추출	74
4.3.1	부분어절 그래프와 그래프 랭킹 알고리즘을 이용한 키워드 추출	74
4.3.2	키워드 집합을 이용한 핵심 문장 선택	76
4.4	성능 평가	78
4.5	결론	84
제 5 장	다주제 문서 집합 요약을 위한 문서 군집화 알고리즘 및 군집 별 키워드 추출	86
5.1	개요	86
5.2	관련 연구	87
5.3	문서 군집화를 위하여 개선된 Spherical k-means	90
5.3.1	효율적인 Spherical k-means 초기화	91
5.3.2	군집 중심값을 이용한 문서 군집 별 키워드 추출 방법	93
5.4	성능 평가	94
5.4.1	초기화 방법의 성능 평가	95
5.4.2	문서 군집 별 키워드 추출 방법의 성능 평가	97
5.5	중심 벡터의 차원 축소와 군집 레이블을 이용한 군집화 결과 시각화	99
5.6	결론	101
제 6 장	시계열 형식의 뉴스 문서 집합 요약을 위한 거리 기반 유사 주제 구간 분리	103
6.1	개요	103
6.2	관련 연구	104

6.3	유사 주제 구간 분리를 이용한 시계열 형식의 문서 요약	108
6.3.1	시계열 분리를 위한 문서 집합의 구간 별 벡터 표현 방법	109
6.4	성능 평가	112
6.4.1	질의어 '김무성'이 포함된 뉴스 기사의 구간 분리	113
6.4.2	질의어 '박근혜'가 포함된 뉴스 기사의 구간 분리	114
6.4.3	질의어 '유시민'이 포함된 뉴스 기사의 구간 분리	116
6.5	결론	117
제 7 장 결론		122
7.1	이 논문의 기여	124
7.2	후속 연구	126
참고문헌		128
Abstract		166
감사의 글		169

표 목차

표 1.1 한국어 단어 품사 구조	6
표 2.1 기학습된 한국어 형태소 분석기를 이용한 문장 분석 예시. (N: 명사, J: 조사, V: 동사, E: 어미, VCP: 동사형 전성어미)	30
표 2.2 Example of Word Piece Model tokenization result	32
표 2.3 명사가 포함된 어절의 구조. (N: 명사, J: 조사, V: 동사, E: 어미, EP: 선어말어미, VCP: 동사형 전성어미)	33
표 2.4 Cohesion 점수와 Branching Entropy 예시	35
표 2.5 Unigram 과 Uni + Bigram 을 이용한 영화평의 긍부정 분류 과업 성능	40
표 2.6 토크나이저 별 고유 명사 재현율 (배우 이름, 영화 제목, 극 중 캐릭터 이름)	43
표 2.7 단어 임베딩 벡터를 이용하여 검색된 배우 이름의 유사어 중 배우 이름인 비율	43
표 3.1 세종 말뭉치의 형태소 품사 별 통계	46
표 3.2 ‘드라마/L’ 와 ‘시작했/L’ 를 포함하는 가장 빈번한 (L, R) 쌍 예시	51
표 3.3 세종 말뭉치의 어절을 L + [R] 구조로 변형한 예시	55
표 3.4 L 과 R 의 최소빈도수 이상 조건을 만족하는 단어의 종류	55
표 3.5 데이터셋과 판별 모델 별 5 - 교차 검증을 이용한 명사 판별 성능 비교	56
표 3.6 세종 말뭉치로부터 구축된 L – R 그래프 통계	57

표 3.7 제안하는 방법과 기학습된 모델들의 세종 말뭉치에서의 명사 인식 성능	58
표 3.8 세종 말뭉치에서 명사로 추출된 빈도수가 작은 12 개의 명사 예시 (Logistic Regression 의 판별 확률, 출현 빈도수)	59
표 3.9 세종 말뭉치에서 명사로 추출된 12 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)	59
표 3.10 제안하는 방법과 기학습된 모델들의 뉴스 기사에서의 명사 인식 성 능 (나무위키 데이터베이스를 사전으로 이용한 경우)	60
표 3.11 제안하는 방법과 기학습된 모델들의 뉴스 기사에서의 명사 인식 성 능 (온라인 한국어 사전을 이용한 경우)	61
표 3.12 뉴스 기사에서 명사로 추출된 빈도수가 작은 12 개의 명사 예시 (Lo- gistic Regression 의 판별 확률, 출현 빈도수)	61
표 3.13 뉴스 기사에서 명사로 추출된 12 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)	62
표 3.14 세종 말뭉치에서 명사로 추출된 빈도수가 작은 100 개의 명사 예시 (Logistic Regression 의 판별 확률, 출현 빈도수)	64
표 3.15 세종 말뭉치에서 명사로 추출된 100 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)	65
표 3.16 뉴스 기사에서 명사로 추출된 빈도수가 작은 12 개의 명사 예시 (Lo- gistic Regression 의 판별 확률, 출현 빈도수)	66
표 3.17 뉴스 기사에서 명사로 추출된 12 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)	67

표 4.1 KR-WordRank, TextRank, 그리고 LexRank로부터 추출된 핵심 문장의 ROUGE-1 성능	82
표 4.2 영화 '라라랜드' 리뷰로부터 KR-WordRank, TextRank, 그리고 LexRank로부터 추출된 핵심 문장 예시	83
표 4.3 영화 '라라랜드' 리뷰로부터 KR-WordRank, TextRank 을 이용하여 추출한 키워드 예시	84
 표 5.1 고차원 벡터로 표현된 7 개의 텍스트 데이터에서의 코싸인 거리 분포. D1: A6 블로그, D2: 투스칸 블로그, D3: 소나타 블로그, D4: IMDb 리뷰, D5: Reuters RCV1, D6: MovieLens 20M, D7: Yelp 리뷰 데이터 (%)	90
표 5.2 실험에 이용한 7 종류의 데이터셋	95
표 5.3 제안된 초기화 방법과 k-means++ 를 이용한 초기화 시간 (초) . .	96
표 5.4 제안된 초기화 방법과 k-means++ 를 이용한 평균 군집화 품질의 배율	97
표 5.5 IMDb 리뷰의 $k=1,000$ 문서 군집화 결과	98
표 5.6 소나타 블로그의 $k=500$ 문서 군집화 결과	99
 표 6.1 실험에 이용한 3 종류의 데이터셋	113
표 6.2 질의어 '김무성'이 포함된 뉴스의 구간 별 문서 요약 예시.	119
표 6.3 질의어 '박근혜'가 포함된 뉴스의 구간 별 문서 요약 예시	120
표 6.4 질의어 '유시민'이 포함된 뉴스의 구간 별 문서 요약 예시	121

그림 목차

그림 1.1 한국어의 구조	6
그림 2.1 L-Tokenizer 의사코드	36
그림 2.2 Max Score Tokenizer 의사코드	36
그림 2.3 띄어쓰기 오류파 포함된 문장 '파스타가좋아요' 과정	37
그림 3.1 A가 포함된 문장 예시	48
그림 3.2 제안하는 명사 추출기의 프레임워크	51
그림 3.3 어절 '드라마를'로부터 생성할 수 있는 (L, R) 쌍 예시	51
그림 3.4 제안하는 명사 추출 방법의 의사 코드	54
그림 4.1 (a) TextRank 의 co-occurrence 를 이용한 단어 그래프 예시, (b) TextRank 의 문장 간 유사도를 이용한 문장 그래프 예시	70
그림 4.2 (a) 어절을 마디로 이용한 경우, (b) 잘못된 어절을 마디로 이용한 경우	72
그림 4.3 KR-WordRank 의 키워드 추출 프레임워크	75
그림 4.4 KR-WordRank 키워드 필터링 함수 의사 코드	76
그림 4.5 KR-WordRank 의 핵심 문장 추출 프레임워크	77
그림 4.6 KR-WordRank 핵심 문장 필터링 함수 의사 코드	78
그림 4.7 WordRank 와 KR-WordRank 를 이용하여 세종 말뭉치로부터 추출한 단어의 정확도	79

그림 4.8 WordRank 와 KR-WordRank 를 이용하여 추출한 영화 ”아저씨” 리뷰의 키워드	80
그림 5.1 Lloyd k-means 의사 코드	88
그림 5.2 (a) 기대하는 초기 군집 중심값, (b) 잘못된 초기 군집 중심값 . .	89
그림 5.3 k-means++ 의 의사 코드	90
그림 5.4 개선된 Spherical k-means 의 의사 코드	91
그림 5.5 개선된 Spherical k-means 초기화 방법	91
그림 5.6 개선된 Spherical k-means 초기화 방법의 예시 (a) 한 점 c_{t-1} 이 선택된 뒤 코싸인 거리가 t_{init} 보다 작은 점들을 제거한 예시 (b) 제거된 점에서 임의로 선택된 c_t	92
그림 5.7 LDAvis 를 이용한 Spherical k-means 학습 결과 시각화 예시 . .	101
그림 6.1 시계열 분리의 기하학적 해석	107
그림 6.2 제안하는 시계열 형식 문서 집합 요약 방법의 의사 코드	109
그림 6.3 단어 빈도 (term frequency) 기반 시점 간 거리 행렬 예시	111
그림 6.4 Doc2Vec 기반 시점 간 거리 행렬 예시	112
그림 6.5 질의어 ’김무성’이 포함된 뉴스의 일별 문서 개수	113
그림 6.6 질의어 ’김무성’이 포함된 뉴스 문서의 일간 거리 행렬	114
그림 6.7 질의어 ’박근혜’가 포함된 뉴스의 일별 문서 개수	115
그림 6.8 질의어 ’박근혜’가 포함된 뉴스 문서의 일간 거리 행렬	116
그림 6.9 질의어 ’유시민’이 포함된 뉴스의 일별 문서 개수	116
그림 6.10 질의어 ’유시민’이 포함된 뉴스 문서의 일간 거리 행렬	117

제 1 장 서론

자연어처리 (natural language processing) 는 사람의 언어를 컴퓨터가 이용할 수 있는 형태의 정보로 변환하고, 이를 이용하여 과업들 (tasks) 을 수행하는 분야이다. 자연어처리 분야에는 다양한 종류의 과업이 포함되어 있다. 품사 판별과 형태소 분석은 텍스트를 단어열로 분해하는 전처리 과정이다. 단어의 특정 품사나 의미를 이해하는 객체명 인식, 키워드나 핵심 문장을 이용하여 문서 집합을 요약하는 정보 추출 과업, 사용자의 질문에 대해 적절한 답변을 탐색하는 질의 응답도 자연어처리 과업에 포함된다.

자연어처리 과업의 많은 부분은 머신러닝 기법을 이용한다. 머신러닝은 학습 방식에 따라 세 가지로 분류할 수 있다. 지도학습 기반 (supervised) 머신러닝은 객체의 특징을 기술하는 입력값 (input) 과 객체의 정답인 출력값 (output) 이 쌍으로 존재할 때 이를 이용하는 방식이다. 머신러닝 모델은 입력값으로부터 출력값을 예측하기 위하여 두 값 사이의 관계를 학습한다. 품사 판별의 경우 단어 - 품사 쌍으로 이루어진 학습 데이터를 이용하여 단어열이 입력되었을 때 이에 해당하는 품사열을 판별하는 정보를 학습한 뒤, 새로운 단어열이 입력되면 적절한 품사열을 출력한다. 비지도학습 (unsupervised) 머신러닝은 데이터에 입력값만 존재할 때 이용되는 방법이다. 품사 판별 과업에서는 두 단어가 앞 뒤에 등장하는 단어들의 분포가 비슷하면 하나의 품사로 이를 인식하는 Brown clustering [28] 방법이 이에 해당한다. 강화학습 (reinforcement learning) 은 각 입력값에 대한 출력값 대신 리워드 (reward) 가 주어졌을 때 이를 이용하는 방식으로, 모델의 출력값에 대하여 피드백을 정의할 수 있는 도메인에서 이용된다. 예를 들어 대화 시스템에서는 출력된 답변에 대한 피드백을 이용하여 답변 출력 방법을 수정하는 모델들이 제안되었다 [149, 191, 124].

자연어처리의 많은 연구들은 지도학습 머신러닝 방법을 이용한다. 품사 판별과 같은 과업은 텍스트 데이터를 벡터로 변환하는 전처리 과정에 이용되기 때문에 높은 정확도가 요구되며, 강화학습처럼 결과값에 대한 피드백을 정의하기가 어렵기 때문에 지도학습 머신러닝 방법이 이용된다. 그러나 품사 판별을 포함하여 지도학습 머신러닝 방법을 이용하는 자연어처리 모델들은 공통적으로 다음과 같은 어려움이 있다.

1. **Out of vocabulary problem** : 학습 데이터에 등장하지 않은 단어를 제대로 인식하지 못하는 문제이다.
2. **Dearth of Data** : 과업에 적합한 학습 데이터를 마련하기 어렵다.
3. **Error in data** : 띠어쓰기 오류 및 철자법 오류는 잘못된 자연어처리 과업 결과를 야기한다.

첫째, 미등록단어 (Out of vocabulary) 문제이다. 언어는 사용되는 시기와 도메인에 따라 다르기 때문에 한 종류의 학습 데이터에 모든 종류의 단어가 등장하지 않는다. 그렇기 때문에 학습 데이터에 등장하지 않은 신조어들은 품사 판별 과정에서 잘못된 단어열로 분해될 수 있다. 이는 품사 판별의 결과를 이용하는 다른 과업들의 성능을 저하시킨다. 영어처럼 공백을 기준으로 단어의 경계를 구분하는 언어에서는 단어 임베딩 기법을 이용하여 학습 데이터에 존재하지 않은 단어에 대해서도 순쉬운 품사 추정이 가능하다 [202, 145, 47]. 하지만 한국어, 중국어, 일본어와 같은 언어에서는 공백만으로는 단어의 경계를 정확히 구분하기 어렵기 때문에 단어 사전 혹은 다양한 단어와 품사가 포함되어 있는 학습 데이터가 필요하다.

둘째, 학습 데이터가 존재하지 않거나 이를 구축하기 어려운 경우들이 많다. 객체명 인식은 장소나 사람과 같은 단어의 의미 종류를 분류하는 과업이다. 품사 판별처럼 객체명 인식 과업은 각 객체명들이 태깅된 학습 데이터가 필요하지만, 도메인마다 단어

클래스가 다르기 때문에 새롭게 학습 데이터를 구축해야 한다. 예를 들어 영화 제목을 인식하는 객체명 인식 모델을 구축하기 위해서는 영화 제목이 태깅된 학습 데이터가 필요하며, 뉴스의 정치인 이름을 인식하는 모델을 학습하기 위해서는 정치인 이름이 태깅된 학습 데이터가 필요하다.

셋째, 텍스트 데이터에는 오류가 존재한다. 띠어쓰기와 같은 문법 오류나 잘못된 철자법에 의하여 한 단어가 다른 의미의 단어로 인식될 수 있다. 사람은 텍스트에 일정 수준의 오류가 포함되어 있다 하더라도 가독에 큰 어려움이 없지만 자연어처리 과업에 이용되는 많은 수의 머신러닝 알고리즘은 오류에 의하여 오작동할 가능성이 크다. 자연어처리 과업의 성능을 높이기 위해서는 노이즈를 제거하는 과정을 거쳐야 하거나 오류에 강건한 모델을 이용해야 한다. 그러나 오류를 제거하기 위해서는 추가적인 모델과 학습 데이터가 필요하다.

이 논문에서는 다음의 한국어 자연어처리 과업에서 발생하는 위의 세 가지 문제를 해결하는 방법들을 제안한다.

1. 문장에서 단어를 인식하는 토크나이징 과업
2. 키워드와 핵심 문장 추출을 통한 문서 집합 요약

이 논문에서는 비지도학습 접근 방법으로 위의 두 가지 자연어처리 과업을 해결하는 방법을 제안한다. 비지도학습 방법은 학습 데이터에 대한 의존도가 낮기 때문에 다양한 문제에 적용하기 용이하다. 또한 비지도학습 방법들은 지도학습 기반 방법들을 보완하는 역할을 할 수 있기 때문에 비지도학습 방법의 성능이 향상될수록 지도학습 기반 방법의 성능도 향상시킬 수 있다.

이 논문에서는 한국어의 구조적 특징을 이용하여 위의 과업을 해결하는 방법을 제안한다. 비지도학습 방법은 데이터로부터 학습할 수 있는 통계 정보를 이용하는데,

데이터에 대한 가정은 비지도학습 모델이 데이터로부터 무엇을 학습해야 하는지 정의하는 역할을 한다.

2 장에서는 한국어의 언어 특징을 이용하여 단어 인식 성능을 향상한 비지도학습 한국어 토크나이저를 제안한다. Word Piece Model 과 같은 비지도학습 기반 토크나이저는 언어의 구조적 특징을 이용하지 않기 때문에 낮은 단어 인식 능력을 보였다. 제안하는 방법은 한국어 어절의 구조를 이용하는 비지도기반 토크나이저로 단어 추출 기법을 이용하여 단어 간 경계 인식 능력을 높였다.

3 장에서는 한국어의 언어 특징을 이용한 통계 기반 명사 추출 방법을 제안한다. 단어는 여러 종류의 품사로 이뤄져 있으며, 한국어에서는 명사에서 가장 많은 미등록단어 문제가 발생한다. 명사는 키워드 추출이나 토퍼 모델링과 같은 자연어처리 과업에서 가장 중요한 품사의 단어이기 때문에 정확한 인식이 필요하다. 이 장에서 제안된 방법은 한국어의 어절 구조를 바탕으로 명사를 추출하며, 다양한 도메인의 명사 사전을 보완하는데 이용될 수 있다.

문서 요약 과업은 주어진 문서 집합의 주제의 다양성에 따라 접근 방법이 달라진다.

4 장에서는 문서 집합이 단일한 주제로 구성된 상황에서 정확한 키워드를 추출하기 위한 방법을 제안한다. 제안하는 방법은 토크나이저에 의존하지 않으며, 문서 집합 내 키워드에 대하여 높은 인식 능력을 보여준다. 또한 키워드 집합이 주어졌을 때 이를 이용하여 중복되지 않는 핵심 문장을 추출하는 방법을 함께 제안한다.

5 장에서는 다양한 주제로 구성된 문서 집합을 요약하기 위한 문서 군집화 기반 문서 요약 방법을 제안한다. 제안하는 방법은 효율적인 문서 군집화를 위하여 개선된 Spherical k-means 알고리즘과 군집 별 키워드를 추출하는 알고리즘으로 구성되어 있다. 제안하는 방법은 키워드 추출을 위하여 추가적인 모델을 요구하지 않기 때문에 대량의 문서 집합 요약에 효율적이다.

6 장에서는 뉴스와 같이 시계열 형식으로 구성된 문서 집합을 요약하는 방법을 제안한다. 뉴스는 시간이 변함에 따라 문서 집합을 구성하는 주제들이 변화하지만, 이에 관련된 레이블 정보를 구축하기 어려운 상황이 많다. 이를 해결하기 위하여 시간에 따라 주제가 변하는 시점을 탐지하여 뉴스 집합을 구간으로 나눈 뒤, 각 구간의 주제를 요약하는 방법을 제안한다.

1 장에서는 위 과업과 관련된 한국어의 구조, 단어 임베딩 기법, 순차적 판별 알고리즘, 한국어 품사 판별, 추출 기반 문서 요약 기법의 내용을 정리한다.

1.1 한국어의 구조

한국어의 문장은 어절로 구성되어 있으며, 어절은 띄어쓰기를 기준으로 나뉘는 단위이다. 한 어절은 하나의 단어 혹은 여러 개의 단어로 구성될 수 있다. 그리고 단어는 하나 이상의 형태소로 구성된다. 한국어의 단어 품사 종류는 5언 9품사이며, 각 단어는 그 자체로 형태소 품사를 지닌다 (표 1.1). 단 형용사와 동사는 용언의 어간과 어미 형태소로 구성된다.

”이것은 예문입니다”라는 문장은 세 개의 어절로 구성되어 있으며, ”이것은” 어절은 ”이것/명사 + 은/조사”로 구성된다 (그림 1.1). ”입니다”는 하나의 단어인 ”입니다/형용사”로 구성되어 있는데, 이는 ”이/형용사 어간 + ㅂ니다/어미”라는 두 개의 형태소로 구성된다. ”예문”은 하나의 형태소가 하나의 단어를 이루며, 하나의 단어가 하나의 어절을 이룬 경우이다.

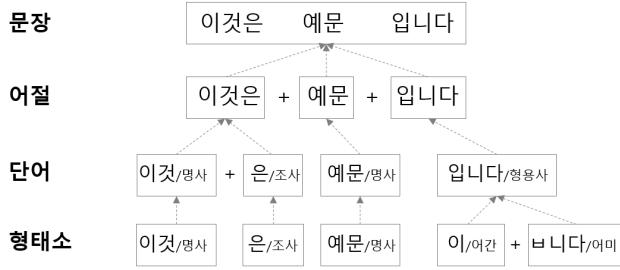


Figure 1.1: 한국어의 구조

단어의 종류는 의미를 지니며 새로운 단어가 만들어지는 열린 집합 (open classes)과 문법 기능을 수행하며 새로운 단어가 잘 만들어지지 않는 닫힌 집합 (closed classes)으로 나뉘어진다 [91]. 대명사, 관형사, 조사, 어미는 문법 기능을 수행하는 단어와 형태소이기 때문에 닫힌 집합에 해당한다. 체언의 수사도 숫자의 종류가 고정되어 있기 때문에 닫힌 집합에 해당한다. 명사나 감탄사는 새로운 개념을 표현하기 위하여 새로운 단어가 만들어지기 때문에 열린 집합에 해당한다. 동사와 형용사는 명사와 전성어미가 합쳐진 형태로 새로운 어간이 만들어지기 때문에 형태소 수준에서는 닫힌 집합이며, 단어 수준에서는 열린 집합에 해당한다.

	체언	명사	대명사	수사
불변어	수식언	관형사	부사	
	관계언	조사		
	독립언	감탄사		
가변어	용언	동사	형용사	

Table 1.1: 한국어 단어 품사 구조

1.2 단어 임베딩 (Word embedding)

언어 모델 (language model) 은 문장의 생성 확률을 표현하기 위한 방법으로, 식 1.1 처럼 n-gram 에 기반하여 문장의 확률을 정의할 수 있다 [92].

$$P(w_{1:p}) = \prod_i P(w_i | w_{i-n+1:i-1}) \quad (1.1)$$

$$P(w_i | w_{i-n+1:i-1}) = \frac{\#(w_{i-n+1:i})}{\#(w_{i-n+1:i-1})}$$

그러나 n-gram 기반 언어 모델은 n-gram의 종류가 증가함에 따라 모델의 크기가 기하급수적으로 증가한다. 또한 단어 간 의미적 유사성을 표현하지 못하는 단점이 있다. 이를 해결하기 위하여 Feed Forward Neural Network 기반 언어 모델이 제안되었다 [18]. [18] 의 언어 모델은 각 단어 w_i 를 고정된 크기의 임베딩 벡터 e_i 로 표현한 뒤, $e_{i-n+1:i-1}$ 단어의 벡터를 연결한 입력값으로 e_i 단어를 예측하는 2 층 Feed Forward Neural Network 를 제안하였다. 이는 식 1.2 처럼 임베딩 벡터 $e_{i-n+1:i-1}$ 를 입력값 x 로 이용한다. 입력값은 (H, d) 와 (W, U, b) 의 2 개의 레이어를 거쳐 e_i 를 예측하는데 이용된다.

$$y = b + Wx + U \cdot \tanh(d + Hx) \quad (1.2)$$

$$x = (e_{i-n+1}, \dots, e_{i-1}), y = e_i$$

그 결과 1.2 은 비슷한 문맥에서 등장하는 단어를 비슷한 벡터로 학습하였다. 하지만 식 1.2 은 오랜 학습 시간이 필요하였으며, 입력값과 출력값의 벡터 공간의 크기가 다르기 때문에 두 개의 임베딩 공간을 학습해야 한다. [145] 는 식 1.2 의 입력값과 출력값의 벡터 공간의 크기를 동일하게 만들기 위하여 $n - 1$ 개 단어의 벡터값의 평균을 입력값으로 변환한 Word2Vec 을 제안하였다. Word2Vec 은 단어의 앞 뒤 모든 문맥을 이용하기 위하여 앞 뒤에 등장하는 단어들의 임베딩 벡터의 평균을 입력값으로 이용한다 (식 1.3).

$$P(w_i|w_c) = \frac{\exp(e_i^T e_c)}{\sum_j \exp(e_j^T e_c)}$$

$$e_c = \text{avg}(e_{i-n}, \dots, e_{i-1}, e_{i+1}, \dots, e_{i+n})$$
(1.3)

하지만 식 1.3 의 클래스의 개수는 여전히 단어 개수처럼 큰 숫자이기 때문에 학습에 오랜 시간이 걸렸다. 이를 해결하기 위하여 Noise Contrastive Estimation [75] 에 기반한 negative sampling 기법을 적용하였다 [146].

GloVe 는 두 단어가 사용자가 정의한 간격 안에 함께 등장한 빈도수 $X_{i,j}$ 를 직접 예측하는 리그레션 모델을 이용하여 단어의 임베딩 벡터를 학습하였다 [164]. 식 1.4 처럼 두 단어 i, j 의 임베딩 벡터 e_i, e_j 의 내적에 bias b_i, b_j 를 더한 값이 co-occurrence 의 로그값 $\log(X_{i,j})$ 에 가까워지도록 e_i 와 b_i 를 학습한다. 자주 등장한 단어에 대하여 임베딩 벡터가 더 잘 학습되도록 증가함수 $f(X_{i,j})$ 를 이용하여 각 단어의 손실양에 가중치를 더한다.

$$\text{minimize} \sum f(X_{i,j}) \cdot (e_i^T e_j + b_i + b_j - \log(X_{i,j}))^2$$
(1.4)

Word2Vec 과 GloVe 는 [77] 의 ”비슷한 의미를 지닌 단어는 비슷한 문맥에서 등장한다”는 가정을 기반으로 임베딩 벡터를 학습한다. GloVe 는 j 를 고정하면 Word2Vec 처럼 단어 w_i 에 대한 소프트맥스 리그레션으로 해석할 수 있다. 그렇기 때문에 Word2Vec 과 GloVe 는 비슷한 단어 임베딩 벡터를 학습한다 [122].

또한 negative sampling 을 이용하는 Word2Vec 의 한 종류인 Skip-gram (SGNS) 은 단어 간 co-occurrence 행렬에 Shifted Point Mutual Information (Shifted PMI) 를 적용한 공간과 같음이 증명되었다 [121]. Word2Vec 의 임베딩 벡터 e_w, e_c 의 내적은 두 단어의 co-occurrence 의 PMI 값에서 negative samples 의 개수인 k 의 로그값을 뺀 것과 동치임이 증명되었다 (식 1.5). 즉 한 단어는 문맥 단어와의 co-occurrence matrix

에 Shifted PMI 를 적용한 벡터로 표현할 수 있다 [121].

$$e_w \cdot e_c = \log \left(\frac{\#(w, c) \cdot |D|}{\#(w) \cdot \#(c)} \cdot \frac{1}{k} \right) = \log \left(\frac{\#(w, c) \cdot |D|}{\#(w) \cdot \#(c)} \right) - \log(k) \quad (1.5)$$

또한 식 1.6 처럼 문맥 단어와의 co-occurrence 벡터에 Singular Value Decomposition (SVD) 를 적용 경우 Word2Vec 처럼 분산 표상 표현 공간으로 변형이 가능하다 [122]. W^{SVD} 는 SVD 에 의하여 학습된 w 의 임베딩 벡터이며, C^{SVD} 는 c 의 임베딩 벡터이다.

$$\begin{aligned} M_d &= U_d \cdot \Sigma_d \cdot V_d^T \\ W^{SVD} &= U_d \cdot \Sigma_d^{0.5} \\ C^{SVD} &= V_d \cdot \Sigma_d^{0.5} \end{aligned} \quad (1.6)$$

그러나 Word2Vec, GloVe, [121] 와 같은 단어 임베딩 방법은 빈도수가 작은 단어나 학습 데이터에 등장하지 않은 단어에 대해서는 정확한 벡터를 학습하기 어렵다. FastText 는 빈도수가 적은 단어와 미등록단어의 임베딩 벡터를 표현하기 위하여 한 단어를 구성하는 부분단어 (subwords) 의 임베딩 벡터 합으로 각 단어를 표현한다 [26]. 예를 들어 'where' 라는 단어의 임베딩 벡터는 단어의 경계를 표현하기 위하여 단어 앞 뒤에 (,) 를 추가한 뒤, '(where)' 의 3글자에서 6 글자의 부분단어인 '(wh, whe, ..., where)' 의 임베딩 벡터를 학습한다. 그 뒤 단어 'where' 의 단어의 벡터는 이를 구성하는 모든 부분단어 벡터의 합으로 표현한다. 부분단어의 벡터는 Word2Vec 을 이용하여 학습한다. 그 결과 비슷한 문맥에서 이용되는 부분단어는 비슷한 임베딩 벡터로 표현되며, 형태와 문맥이 비슷한 단어는 부분단어들의 벡터에 의하여 서로 비슷한 벡터로 표현된다.

이러한 단어 임베딩 벡터는 이후 다른 자연어처리 과업의 사전학습된 (pre-trained)

벡터로 이용되며 [101], 미등록단어 문제 해결에 이용되기도 한다. 임베딩 벡터가 비슷한 단어는 문맥이 비슷하기 때문에 품사 정보가 포함된 단어들의 임베딩 벡터와의 유사도를 이용하여 품사 정보가 없는 단어의 품사를 추정할 수 있다.

그러나 단어 임베딩 방법도 미등록단어 문제를 모두 해결하지는 못한다. FastText는 학습데이터에 등장하지 않은 단어에 대하여 형태적 유사성을 바탕으로 임베딩 벡터를 추정하기 때문에 단어가 새로운 문맥에서 등장할 경우 이를 반영할 수는 없다.

1.3 순차적 레이블링 (Sequential labeling)

순차적 레이블링 기법은 길이가 n 인 입력 시퀀스 $x_{1:n}$ 에 대하여 가장 적절한 출력 시퀀스 $y_{1:n}$ 을 출력하는 머신러닝 알고리즘이다 (식 1.7).

$$\arg \max_{y_{1:n}} P(y_{1:n}|x_{1:n}) \quad (1.7)$$

식 1.7를 정의하는 방법에 따라 다양한 순차적 레이블링 기법이 제안되었다. Hidden Markov Model (HMM)은 Markov property를 이용하는 방법으로, 오래전부터 순차적 레이블링에 이용되었다 [105]. HMM은 식 1.8처럼 두 종류의 패러매터로 구성되어 있다. $P(y_i|y_{i-1})$ 은 전이 확률 (transition probability)로, 출력값 y_i 는 이전 시점의 출력값 y_{i-1} 에 영향을 받는다. $P(x_i|y_i)$ 은 생성 확률 (emission probability)로, 출력값 y_i 가 주어졌을 때 입력값 x_i 가 발생할 확률이다. HMM은 출력, 입력 시퀀스의 생성 확률을 최대화 하는 출력 시퀀스를 탐색한다.

$$P(y_{1:n}|x_{1:n}) = \prod_{i=1}^n P(y_i|y_{i-1})P(x_i|y_i) \quad (1.8)$$

그러나 HMM은 다음의 이유로 자연어처리 과업에 적합하지 않다. 첫째는 학습

데이터에 등장하지 않은 데이터의 생성 확률이 0으로 정의된다. 품사 판별의 경우 입력 시퀀스 단어열에 대한 품사열을 탐색해야 하지만, 학습 말뭉치에 등장하지 않은 단어에 대한 출력 확률이 0으로 정의된다. 이러한 문제를 해결하기 위하여 학습 데이터에 등장하지 않은 x_i 는 규칙 기반으로 처리하는 방법들이 이용되기도 하였다 [27].

둘째로 ”unguaranteed independence problem” 이 발생한다. 품사 추정을 위해서는 앞 뒤에 등장한 단어들을 문맥으로 이용해야 한다. 하지만 HMM 은 x_i, y_i 와 y_i, y_{i-1} 간의 상관성만 학습하며, 연속된 단어로 표현되는 앞 뒤의 문맥 정보는 이용하지 않는다.

이러한 문제를 해결하기 위하여 Maximum Entropy Markov Model (MEMM) 과 Conditional Random Field (CRF) 가 제안되었다 [138, 107]. 이들은 입력 시퀀스를 벡터로 표현한 뒤, 소프트맥스 리그레션을 이용하여 적합한 출력 시퀀스를 탐색한다. 입력 시퀀스를 벡터로 표현하기 위하여 사용되는 잠재 함수 (potential function) 필터를 이용하여 시퀀스의 한 시점을 Boolean 벡터로 표현한다. 예를 들어 입력 시퀀스 $x_{1:3} = [3.2, 2.1, -0.5]$ 가 주어질 때, 아래처럼 한 개의 잠재 함수 F_1 를 이용하면 (1, 3) 크기의 벡터화된 입력 시퀀스 $h_{1:3}$ 를 얻을 수 있다.

- $F_1 = 1 \text{ if } x_i > 0 \text{ else } 0$
- $h_{1:3} = [1, 1, 0]$

아래와 같이 두 개의 잠재 함수 F_1, F_2 를 이용하면 (2, 3) 크기의 $h_{1:3}$ 를 얻는다.

- $F_1 = 1 \text{ if } x_i > 0 \text{ else } 0$
- $F_2 = 1 \text{ if } x_i > 3 \text{ else } 0$
- $h_{1:3} = [(1, 1), (1, 0), (0, 0)]$

잠재 함수는 명목 변수열 입력시퀀스도 벡터로 변환할 수 있다. 입력 시퀀스가 '[이 것, 은, 예문, 이다]' 와 같은 단어열 일 때, 아래의 잠재 함수를 이용하면 (3, 4) 크기의 $h_{1:3}$ 를 얻을 수 있다.

- $F_1 = 1$ if $x_{i_1} = \text{이것}$ and $x_i = \text{은}$ else 0
- $F_2 = 1$ if $x_{i_1} = \text{이것}$ and $x_i = \text{예문}$ else 0
- $F_3 = 1$ if $x_{i_1} = \text{은}$ and $x_i = \text{예문}$ else 0
- $h_{1:3} = [(0, 0, 0), (1, 0, 0), (0, 0, 1), (0, 0, 0)]$

잠재 함수의 가장 큰 장점은 모델링에 이용할 수 있는 변수를 사용자가 임의로 정의할 수 있다는 점과 입력 시퀀스가 벡터로 변환되어도 해석이 가능하다는 점이다. 식 1.9 처럼 잠재 함수가 이전 출력값 y_{i-1} 을 이용할 경우 Markov property 를 지닌다. 잠재 함수는 주변 입력값에 대한 정보를 벡터로 변환할 수 있기 때문에 앞 뒤 단어의 문맥을 이용하는 품사 판별 문제에 이용되었다. MEMM 은 잠재 함수를 이용하여 입력 시퀀스를 벡터로 변환한 뒤, 앞부분부터 순차적으로 소프트맥스 리그레션을 이용하여 출력값을 탐색한다 [138].

$$P(y_{1:n}|x_{1:n}) = \prod_{i=1}^n \frac{\exp(\sum_{j=1}^m \lambda_j f_j(x, i, y_i, y_{i-1}))}{\sum_{y^i} \exp(\sum_{j=1}^m \lambda_j f_j(x, i, y_i, y_{i-1}^i))} \quad (1.9)$$

그러나 MEMM 은 두 가지 문제점을 지니고 있다. 첫째는 레이블에 대한 편향성으로 (label bias), 입력 시퀀스의 매 순간마다 소프트맥스 리그레션에 의한 지역적 정규화 (local normalization) 가 이뤄지면 자주 등장하지 않은 레이블 y_{i-1} 이 선호되는 현상이 발생한다[107, 106, 8]. 출력값의 종류가 다양할 경우 자주 등장하는 출력값 y_{i-1} 다음에 y_i 가 등장할 확률은 대부분 작은 값을 지니지만 y_i 의 종류가 다양하지 않은 경우에는

대부분 큰 값의 확률을 지니기 때문에 확률에 왜곡이 발생한다.

두번재는 길이에 대한 편향성으로 (length bias), 시퀀스 분할 (sequence segmentation) 과 레이블링 문제를 동시에 풀어야 하는 상황에서 길이가 짧은 출력 시퀀스를 선호하는 현상이 발생한다. 시퀀스 분할 문제는 $P(y_{1:m}|x_{1:n})$, $m \leq n$ 이 최대인 $y_{1:m}$ 을 탐색하는 문제로, 입력 글자열을 구분하여 단어열로 만드는 문장 분할 문제가 이에 해당한다. 한국어나 일본어는 글자열 $c_{1:n}$ 이 주어졌을 때 이를 단어열 $x_{1:m}$ 로 구분하고, 구분된 단어열에 품사열 $y_{1:m}$ 을 부여해야 한다 (식 1.10).

$$P(x_{1:m}, y_{1:m}|c_{1:n}) \quad (1.10)$$

이를 위해 MEMM 을 이용하면 m 이 작을수록 적은 수의 확률 곱셈이 이뤄지므로, 길이가 긴 단어를 선호하는 (길이가 짧은 출력 시퀀스를 선호하는) 현상이 발생한다 [106].

CRF 는 지역적 정규화에 의한 두 종류의 문제를 해결하기 위하여 식 1.11 처럼 전역적 정규화 (global normalization) 를 통하여 출력 시퀀스 $y_{1:n}$ 를 탐색한다 [107].

$$P(y_{1:n}|x_{1:n}) = \frac{\exp(\sum_{j=1}^m \sum_{i=1}^n \lambda_j f_j(x, i, y_i, y_{i-1}))}{\sum_{y'} \exp(\sum_{j'=1}^m \sum_{i=1}^n \lambda_j f_j(x, i, y'_i, y'_{i-1}))} \quad (1.11)$$

CRF 는 이후 상호 참조 (Co-reference resolution) [140], 객체명 인식 (Named Entity Recognition) [168, 148, 131, 172, 174], 의존 구문 파싱(parsing) [178, 66], 품사 판별 [200, 69] 등 자연어처리의 다양한 순차적 레이블링 문제에 이용되었다 [39, 139].

식 1.11 에 로그를 취하면 식 1.12 처럼 기술할 수 있다. $F(x_{1:n}, y_{1:n})$ 는 입력과 출력 시퀀스에 대한 벡터 표현이며, λ 는 각 변수의 계수이다. $x_{1:n}$ 에 대한 $y_{1:n}$ 의 점수는 F

에 의하여 생성된 변수의 계수 합이며, 모델은 학습데이터에 존재하는 $y_{1:n}$ 와 가능한 모든 출력 시퀀스의 점수 합과의 차이가 최대가 되도록 λ 를 학습한다.

$$\log P(y_{1:n}|x_{1:n}) = \lambda \cdot F(x_{1:n}, y_{1:n}) - \log \sum_{y_{1:n}} \exp F(x_{1:n}, y_{1:n}) \quad (1.12)$$

정답 시퀀스와 예측 시퀀스와의 점수 차이가 최대화 되도록 계수 λ 를 학습할 수도 있다 (식 1.13). 이는 Support Vector Machine 처럼 마진을 최대화 하는 판별기가 되며, 이를 Structural Support Vector Machine 이라 한다 [199, 201]. 식 1.13 은 출력 시퀀스가 정답 시퀀스가 되면 더이상 학습되지 않지만, 식 1.11 은 정답 시퀀스와 그 외의 모든 시퀀스와의 점수 차이가 커지는 방향으로 지속적으로 학습이 일어난다. 그 결과 1.13 은 과적합의 가능성이 적으며 안정적인 λ 가 학습된다.

$$\lambda \cdot (F(x_{1:n}, y_{1:n}) - F(x_{1:n}, \hat{y}_{1:n})) \quad (1.13)$$

잠재 함수에 의하여 생성되는 변수가 Markov property 를 따른다면 식 1.12 처럼 $F(x, y_{i-1}, y_i, i)$ 로 기술할 수 있으며, 출력 시퀀스 값의 전이 성질을 이용하는 이러한 모델을 전이 기반 순차적 레이블링 (transition based sequence labeling) 이라 한다 [24].

$$\lambda \cdot \left(\sum_i F(x_{1:n}, y_{i-1}, y_i, i) - F(x_{1:n}, \hat{y}_{i-1}, \hat{y}_i, i) \right) \quad (1.14)$$

이들은 Markov property 를 따르지만 지역적 정규화를 하지 않기 때문에 레이블이나 길이에 대한 편향성의 위험이 상대적으로 적다.

잠재 함수를 이용하여 입력 시퀀스를 벡터로 변환할 경우, 그 형태는 sparse Boolean 벡터이다. Long-Short Term Memory network (LSTM) 이나 Gated Recurrent Unit (GRU) 와 같은 Recurrent Neural Network (RNN) 계열 신경망 모델은 워드 임베딩

시퀀스 형태의 입력값을 이용할 수 있다 [38, 81]. LSTM 과 GRU 는 게이트 메커니즘 (gating mechanism) 을 이용하여 입력 시퀀스 $x = [x_1, x_2, \dots, x_n]$ 의 정보를 허든 벡터 시퀀스 $h = [h_1, h_2, \dots, h_n]$ 에 누적한다. GRU 는 식 1.15 처럼 매 시점 i 마다 업데이트 게이트 z_i , 리셋 게이트 r_i 를 이용하여 새로운 메모리 컨텐츠 \tilde{h}_i 를 만들어 허든 벡터 h_i 를 업데이트 한다. 출력값은 허든 벡터 h_i 에 선형 판별식을 통하여 선택된다.

$$\begin{aligned} z_i &= \sigma(W^z \cdot x_i + U^z \cdot h_{i-1}) \\ r_i &= \sigma(W^r \cdot x_i + U^r \cdot h_{i-1}) \\ \tilde{h}_i &= \tanh(W \cdot x_i + r \circ U \cdot h_{i-1}) \tag{1.15} \\ h_i &= z_i \circ h_{i-1} + (1 - z_i) \circ \tilde{h}_i \\ y_i &= \text{softmax}(Vh_i) \end{aligned}$$

GRU 는 LSTM 의 구조를 간소화 한 것으로, LSTM 은 GRU 보다 1 개 많은 게이트와 메모리셀 \tilde{c}_i 추가로 학습된다. 게이트는 입력 게이트 i_i 삭제 게이트 f_i , 출력 게이트 o_i 로 구성되어 있다.

$$\begin{aligned} i_i &= \sigma(W^i \cdot x_i + U^i \cdot h_{i-1}) \\ f_i &= \sigma(W^f \cdot x_i + U^f \cdot h_{i-1}) \\ o_i &= \sigma(W^o \cdot x_i + U^o \cdot h_{i-1}) \\ \tilde{c}_i &= \tanh(W^c \cdot x_i + r \circ U^c \cdot h_{i-1}) \tag{1.16} \\ c_i &= f_i \circ c_{i-1} + i_i \circ \tilde{c}_i \\ h_i &= o_i \circ \tanh(c_i) \\ y_i &= \text{softmax}(Vh_i) \end{aligned}$$

게이트 메커니즘은 하든 벡터에 입력 시퀀스의 앞 뒤의 일부 정보를 선택적으로 이용할 수 있도록 도와준다. 이러한 모델은 단방향적인 정보만을 순차적으로 저장할 수 있기 때문에 역방향의 독립된 RNN 계열 네트워크를 동시에 학습하는 Bidirectional LSTM (BiLSTM)이나 Bidirectional GRU (BiGRU)가 제안되었다 [71].

잠재 함수에 의하여 생성되는 변수 공간은 단어 개수와 변수 종류에 비례하여 매우 커지며, 비슷한 의미를 지니는 모든 변수가 서로 독립적으로 학습되는 단점이 있다. 하지만 워드 임베딩을 통하여 서로 비슷한 문맥에서 등장하는 단어는 비슷한 벡터로 표현할 수 있게 되었고, 워드 임베딩 벡터만 학습할 수 있다면 말뭉치에 등장하지 않은 단어에 대해서도 향상된 성능으로 품사 판별이나 객체명 인식이 가능하다 [47, 108].

위의 모델들은 식 1.17처럼 출력 시퀀스 간의 관계에 대한 제약이 없다.

$$S(x_{1:n}, y_{1:n}) = \sum_{i=1}^n f_\theta(x_i, y_i) \quad (1.17)$$

식 1.18처럼 이전 출력값과의 관계를 함께 학습하기 위하여 LSTM-CRF 모델이 제안되었으며, 객체명 인식과 품사 판별 과업에 이용되었다 [135, 173, 108]. $A(y_{i-1}, y_i)$ 는 전이 모델의 역할을 한다.

$$S(x_{1:n}, y_{1:n}) = \sum_{i=1}^n (A(y_{i-1}, y_i) + f_\theta(x_i, y_i)) \quad (1.18)$$

입력 시퀀스가 단어열이 아닌 글자열일 경우에도 객체명 인식과 같은 작업이 가능하다 [73]. 입력 단어가 학습 데이터에 존재하지 않아 임베딩 벡터를 계산하지 못하는 경우를 방지하기 위하여 Convolutional Neural Network (CNN) 필터를 이용하여 단어의 글자로부터 임베딩 벡터를 학습하여 입력 시퀀스로 입력하는 LSTM + CNN 모델도 제안되었다 [37].

뉴럴 네트워크를 이용하는 전이 기반 순차적 레이블링 방법도 제안되었다 [239, 47, 3]. 전이 기반 모델은 피드 포워드 네트워크를 이용하여 점수 함수를 구축할 수 있으며, 입력 시퀀스로부터 정보를 추출하기 위하여 CNN 필터도 함께 이용될 수 있다 [47]. 이들은 중국어의 단어 분할 작업을 위해 이용되기도 하였다 [230, 31, 15].

1.4 머신 러닝 기반 한국어 품사 판별

품사 판별은 주어진 문장을 단어열로 인식하는 과정이다. 이전에는 한국어의 품사 판별을 규칙 기반 모델이 이용되기도 하였으나 [247, 251], 이들은 규칙의 관리가 어려울 뿐 아니라 확장성이 떨어지기 때문에 최근에는 학습말뭉치를 이용하여 머신 러닝 모델을 학습한 품사 판별기가 이용된다.

품사 판별은 두 가지 과정으로 구성된다. 첫째 주어진 문장에서 가능한 모든 종류의 단어열 후보를 생성하고, 둘째 이들 중에서 가장 적절한 단어열 후보를 선택한다.

영어는 공백을 기준으로 단어의 경계가 구분되기 때문에 순차적 레이블링 방법이 이용되어 입력된 단어열 $w_{1:n}$ 에 대한 품사열 $t_{1:n}$ 을 탐색한다. 그러나 한국어의 어절은 한 개 이상의 단어로 구성되거나 문장 내에 띄어쓰기 오류가 존재할 수 있기 때문에 글자열 $c_{1:n}$ 을 단어열 $w_{1:m}$ 과 품사열 $t_{1:m}$ 을 동시에 구분해야 한다 (식 1.19) [249]. 즉 한국어의 품사 판별 문제는 시계열 데이터의 분리와 레이블링이 동시에 이뤄진다. 주어진 문장에서 모든 종류의 연속된 음절인 $w_{p:q}$ 를 단어 후보로 탐색할 경우 그 종류는 2^{n-1} 개이기 때문에 단어 사전을 이용하여 문장에서 단어열 후보를 생성한다.

$$\arg \max_{w_{1:m}, t_{1:m}} P(w_{1:m}, t_{1:m} | c_{1:n}) \quad (1.19)$$

한국어의 단어는 불변어와 가변어로 구성되어 있으며 (표 1.1), 불변어는 단어의 형태가 변하지 않기 때문에 사전을 이용하여 주어진 글자열이 단어인지 확인할 수 있

다. 가변어에 속하는 동사와 형용사를 용언이라 하며, 이들은 한 의미를 지니는 단어가 여러 종류의 표현형을 가진다. 예를 들어 '가다/동사'는 '가는데', '갔던'처럼 다양한 문맥에 맞춰 그 형태가 변하며, 이를 용언의 활용이라 한다. 용언의 모든 표현형이 포함된 단어 사전을 이용할 경우 불변어와 가변어는 동일한 방법으로 품사를 판별할 수 있다. 하지만 실제로 사용되는 모든 종류의 표현형을 사전으로 만들기가 어렵기 때문에 표현형으로부터 어간과 어미를 인식하는 형태소 분석이 이용된다.

용언은 어간과 어미 두 종류의 형태소로 구성된 단어로, 의미를 지니는 어간은 고정되며 다양한 종류의 어미가 결합되어 표현형이 만들어진다. 용언의 규칙활용은 어간과 어미의 형태가 변하지 않는 활용을 의미하며, '가/어간 + 는데/어미 → 가는데/동사'는 규칙활용에 속한다. 어미 중에는 '-ㄴ다고'처럼 종성으로 시작하는 어미가 있으며, '가/어간 + ㄴ다고/어미 → 간다고/동사' 역시 규칙활용에 포함된다. '가/어간 + 았어/어미 → 갔어/동사'처럼 어간과 어미의 결합 부분에서 두 형태소의 형태가 변하는 활용을 불규칙활용이라 하며, 이들은 'ㄷ불규칙', 'ㄹ불규칙'처럼 국문법에 의하여 변형과정의 규칙이 정의되어 있다 [241].

형태소 분석기를 구현할 시에는 형태소 결합 규칙을 직접 이용하지 않는다. 한 종류의 표현형은 여러 번 등장하며, 이를 초, 중, 종성으로 분해한 뒤 형태소 결합 규칙의 부합여부를 매번 확인하는 과정은 매우 큰 계산 비용을 요구한다. 이러한 문제를 해결하기 위하여 [244] 은 기학습된 원형 복원 사전을 이용하는 방법을 제안하였다. 용언은 어간과 어미가 결합되는 부분에서 형태가 규칙적으로 변하기 때문에 그 종류가 유한하다. '가 + 았 → 갔'의 규칙은 '-가'로 끝나는 어간과 '-았'으로 시작하는 모든 어미의 활용에 이용될 수 있다. 이 규칙을 적용하면 음절 단위의 확인만으로 '나갔어 → 나가/어간 + 았어/어미'로 복원할 수 있다.

단어 사전의 확인과 형태소 분석을 통하여 생성된 단어열 후보들은 순차적 레이블링

의 방법을 통하여 주어진 문장과의 적합성이 평가된다. 한 단어 w_i 의 품사를 추정하기 위하여 앞 뒤에 등장하는 단어인 $(w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2})$ 나 글자열 $c_{i-p:i-1}$ 나 [114], 앞 단어의 띠어쓰기 유무 혹은 단어의 길이와 같은 정보를 파생 정보를 자질 (features)로 이용할 수 있다 [151].

Conditonal Random Field (CRF) 와 같은 모델은 사용자에 의하여 적절한 자질을 설계하여야 하며, 잠재 함수에 의하여 생성되는 자질은 지역적인 정보만을 포함한다. 문장 내 모든 단어를 바탕으로 적절한 문맥을 파악하기 위하여 Recurrent Neural Network 과 어텐션 메커니즘을 이용하는 품사 판별기도 제안되었다 [248]. 그러나 그 성능은 기존의 CRF 기반 모델에 근접한 수준이었다.

음절 단위의 품사 판별 모델도 제안되었다. 이들은 객체명 인식 과업에서 자주 이용되는 B, I 품사 구조를 이용하여 식 1.19 대신 식 1.20 를 탐색한다 [233].

$$\arg \max_{t_{1:n}} P(t_{1:n} | c_{1:n}) \quad (1.20)$$

예를 들어 문장 '이것은 예문이다'의 경우 '이것/N, 은/J, 예문/N, 이다/Adj' 로 나뉘어지는데 이를 음절 단위의 품사로 표현하면 '이/B-N, 것/I-N, 은/B-J, 예/B-N, 문/I-N, 이/B-Adj, 다/I-Adj' 로 구분한다 [245, 243, 242]. B 는 단어의 시작을 의미하기 때문에 B 를 기준으로 음절을 병합하면 단어열로 복원이 가능하다. 음절 단위의 품사 판별을 위한 자질을 만들기 위하여 Convolutional Neural Network (CNN) 를 이용하는 모델도 제안되었다 [252]. [252] 은 잠재 함수에 의하여 생성되는 비슷한 의미의 자질들이 하나의 CNN 필터 안에 학습될 수 있음을 보였다. 그러나 음절 단위의 품사 판별은 단어 사전에 존재하지 않는 단어로 품사 판별을 수행할 가능성성이 있기 때문에 일반적으로 단어 사전 기반의 품사 판별기보다 낮은 성능을 보인다 [155, 183, 232, 248, 252].

고속의 품사 판별기를 만들기 위하여 기분석 사전을 이용하는 방법도 제안되었다

[250, 246]. 한국어의 품사 판별과 형태소 분석기는 초, 중, 종성의 구분, 스트링 탑업의 절제 (slicing), 해싱 기반 탐색 등 계산 비용이 큰 작업들로 구성되어 있다. 하지만 문서 집합은 소수의 빈번한 어절들로 구성되어 있기 때문에 동일한 작업이 반복된다. 품사 판별 모델이 이용하는 단어 사전이 변하지 않는다면 한 어절 내에서 발생할 수 있는 단어 후보는 유한하다. 빈번한 어절에서 발생하는 단어열 후보를 메모리에 저장한 뒤 이를 재이용하면 동일한 작업의 반복 없이 고속으로 단어열 후보 생성이 가능하다.

1.5 문서 요약

문서 요약 (summarization) 과업은 키워드와 핵심 문장을 이용하여 문서 집합의 내용을 요약하는 과업이다 [224]. 문서 요약의 과업은 접근 방식에 따라 추출 기반 (extractive approach) 접근법과 요약 기반 (abstractive approach) 접근법으로 나뉘어진다 [224].

추출 기반 방식 접근법은 문서 집합을 대표 할 수 있는 단어 혹은 문장을 데이터에서 선택한다. 이 접근 방식은 TextRank [144] 와 같은 그래프 랭킹 기반 방법들이 오랫동안 이용되었다. TextRank 는 통계 기반으로 작동하기 때문에 학습 데이터에 의존하지 않으며 적은 학습 비용으로도 학습이 가능하다 [161, 153]. 요약 기반 접근 방식은 딥러닝을 기반 자연어처리 기술을 이용하여 최근에 급부상하고 있는 접근 방식으로, 문서 집합의 내용을 요약할 수 있는 새로운 문장을 생성한다 [152]. 하지만 요약 기반 접근 방법은 정답 요약 문장을 학습 데이터로 요구하며, 모델의 학습 비용이 크다. 최근에는 두 접근 방식의 장점을 모두 이용하기 위한 연구들도 제안되고 있다 [16, 19, 74].

1.5.1 키워드 추출을 이용한 토픽 레이블링

키워드 추출은 토픽 모델링 분야에서 다양한 방법이 제안되었다. Latent Dirichlet Allocation (LDA) 는 토픽 모델링에서 가장 많이 이용되는 방법으로, 문서를 토픽 확률 벡터로, 토픽을 단어 확률 벡터로 표현한다 [23]. Latent Dirichlet Allocation 는 Singular Vector Decomposition (SVD) 을 이용하여 문서와 단어를 토픽 공간의 벡터로 표현하는 Latent Semantic Indexing (LSI) [109] 보다 해석력이 좋으며, Probabilistic Latent Semantic Indexing (pLSI) [82] 보다 안정적인 학습과 새로운 문서에 대한 토픽 벡터 추정이 가능하다. 그러나 Latent Dirichlet Allocation로부터 학습된 토픽 벡터에는 높은 확률을 가지지만 정보력이 적은 단어들이 (junk terms) 존재하며 [154], 토픽 벡터의 크기는 모델링에 이용된 단어 개수이기 때문에 해석이 어렵다. 이러한 점을 해결하기 위하여 각 토픽을 해석할 수 있는 토픽 키워드를 추출하기 위한 다양한 방법들이 제안되었다.

단어의 생성 확률로 표현되는 각 토픽 벡터에서 확률값이 큰 단어를 키워드로 선택하는 방법들도 제안되었지만 [192, 40, 206], 각 토픽이나 문서 집합에서 자주 등장하는 단어는 정보력이 적은 단어일 가능성성이 높다 [166, 154, 42]. 대신 다음의 두 조건을 만족할수록 정보력이 큰 키워드일 가능성성이 높다 [41].

1. **saliency** : 키워드가 해당 문서 집합을 대표하는가?
2. **distinctiveness** : 한 문서 집합의 키워드를 이용하여 해당 문서 집합과 다른 문서 집합을 구분할 수 있는가?

한 문서 집합의 키워드는 해당 문서 집합을 대표해야 하기 때문에 문서 집합 내 많은 문서에서 등장해야 한다. 하지만 한 문서 집합의 키워드는 해당 문서 집합과 다른 문서 집합을 구분할 수 있어야 한다. 이는 상반되는 기준으로, 한 문서 집합에서만

등장하는 단어는 소수의 문서에서만 등장할 가능성이 높고, 다수의 문서 집합에서 등장하는 단어는 다른 문서 집합에서도 등장할 가능성이 높다. 그렇기 때문에 위의 두 기준을 함께 고려하는 방법들이 제안되었다 [21, 154, 197]. 토픽과 단어 간의 Point Mutual Information (PMI) 을 계산하면 한 토픽에 자주 등장하는 단어를 선택할 수 있다 [154, 197, 147]. 식 1.21 처럼 토픽 내 단어 생성 확률 $P(w|t)$ 를 문서 집합 전체의 단어 분포 $P(w)$ 로 나누면 한 토픽에 유독 자주 등장하는 단어를 선택할 수 있다. 하지만 $P(w)$ 가 매우 작은 단어는 높은 PMI 를 지니기 때문에, 토픽 내 생성 확률이 큰 몇 개의 단어를 선택한 뒤 이들에 대해서만 PMI 를 계산해야 한다 [154, 7].

$$score(w, t) = \frac{P(w|t)}{P(w)} \quad (1.21)$$

[21] 도 단어의 빈도수와 토픽 간 구분력을 모두 고려하는 FREX 라는 지표를 제안하였다. [193] 은 토픽 내 단어 생성 확률과 토픽 별 생성 확률의 분산의 곱을 키워드 점수로 이용하였다. 이 방법은 한 토픽에서 자주 등장하며, 여러 토픽에서 다른 분포로 등장하는 단어를 키워드로 선택한다. [120] 는 뉴스 문서의 카테고리 정보를 토픽으로 이용하였는데, 각 카테고리 별 단어 빈도의 순위의 편차로 단어의 빈도수를 나누는 키워드 점수를 제안하였다. 이는 문서의 클래스를 토픽으로 고려하면 문서 분류를 위한 변수 선택법들과도 비슷하다 [110, 165].

하지만 상반되는 두 가지 기준이 존재할 때 이를 하나의 지표로 종합하면 왜곡된 해석을 야기할 수 있다 [42]. [190]는 saliency 와 distinctiveness 를 표현하는 두 가지 지표를 계산한 뒤, 사용자가 가중 평균의 가중치를 직접 조절할 수 있는 인터페이스인 LDAvis 를 제안하였다. 사용자는 λ 를 조절하면서 키워드 점수 $r(w|t)$ 를 재정의 하며 토픽 공간을 해석한다.

$$r(w|t)_\lambda = \lambda \times \frac{P(w|t)}{P(w)} + (1 - \lambda) \times P(w|t) \quad (1.22)$$

1.5.2 그래프 랭킹 기반 키워드와 핵심 문장 추출

그래프는 정보를 표현할 객체를 마디 (node)로 정의하고, 객체 간의 관계를 호 (edge)로 정의한다. 각 호에는 가중치 (weight)가 할당되며, 두 마디의 거리 혹은 유사도의 값을 가중치로 이용할 수 있다. 그래프 랭킹 알고리즘은 그래프에서 각 마디의 중요성을 정의하는 방법으로, PageRank [157]와 HITS [104]가 이에 해당한다. PageRank는 웹 문서 간의 하이퍼링크 구조를 이용하여 문서 간의 상대적 중요도를 계산하기 위하여 제안되었다. 이는 '중요한 웹 문서로부터 링크를 (backlink) 받는 문서는 중요한 문서다'는 가정에 기반한다. 하이퍼링크로 구성된 웹 문서 그래프는 방향성을 지니는 유방향 그래프이다. 한 문서의 랭크 $PR(u)$ 는 식 1.23처럼 문서 u 로 링크를 지니는 다른 문서들의 랭크 $PR(v)$ 의 평균으로 정의된다.

$$PR(u) = \sum_{v \in v \rightarrow u} \frac{PR(v)}{N_v} \quad (1.23)$$

모든 마디가 인바운드와 아웃바운드가 존재한다면 식 1.23은 Markov property를 따르기 때문에 수렴 상태가 (steady state) 존재하여 랭크의 값이 수렴한다. 랭크의 계산은 반복적 학습으로 계산될 수 있다. 모든 마디의 랭크는 마디 개수의 역수인 $\frac{1}{N}$ 으로 초기화 한다. $k+1$ 번째 반복 단계에서의 각 마디의 랭크는 k 번째 반복 단계에서의 인바운드 마디의 랭크 값의 평균으로 정의된다.

$$PR(u)_{k+1} = \sum_{v \in v \rightarrow u} \frac{PR(v)_k}{N_v} \quad (1.24)$$

그러나 웹 문서는 아웃바운드를 지니지 않는 문서가 존재할 수 있기 때문에 bias

를 추가한다 (식 1.25). 식 1.24 를 따라 랭크 값을 업데이트 한 값과 초기화 값 $\frac{1}{N}$ 을 $c : 1 - c$ 의 비율로 가중 평균한다. $(1 - c) \times \frac{1}{N}$ 은 모든 마디를 $1 - c$ 의 비율로 랜덤하게 연결한 것과 같은 효과를 지니기 때문에 steady state 를 얻을 수 있다.

$$PR(u) = c \times \sum_{v \in v \rightarrow u} \frac{PR(v)}{N_v} + (1 - c) \times \frac{1}{N} \quad (1.25)$$

HITS 는 각 마디가 hub 와 authority 라는 두 개의 랭크값을 할당 받으며, 랭크를 계산하는 반복 단계마다 정규화 과정이 있는 점이 다르다 (식 1.26).

$$\begin{aligned} hub(u) &= \sum_{v:(v \rightarrow u)} authority(v) \\ authority(u) &= \sum_{v:(v \rightarrow u)} hub(v) \end{aligned} \quad (1.26)$$

식 1.26 를 반복 계산할 경우, 그래프 전체의 랭크의 합이 계속 증가하기 때문에 매 반복단계마다 hub 와 authority 벡터의 크기를 일정하게 만들기 위하여 L2 정규화를 한다. 하지만 HITS 와 PageRank 는 모두 중요한 마디는 다른 중요한 마디와 연결되어 있다는 가정에 기반하기 때문에 비슷한 학습 결과를 보인다.

웹 문서 그래프에서 중요한 마디를 정의하기 위하여 제안된 그래프 랭킹 방법은 문서 요약 과업에도 사용되었다. TextRank [144] 는 단어 그래프로부터 키워드를 추출하는 방법이다. 문장의 단어를 마디로, 문장 내 두 단어가 함께 등장한 비율을 가중치로 정의하여 단어 그래프를 구성한 뒤 PageRank 알고리즘을 적용하여 랭크가 높은 단어를 키워드로 선택한다. 한 문서의 모든 문장을 마디로, 모든 문장 간의 유사도를 흐의 가중치로 정의하면 문장 그래프를 만들 수 있고, 여기에 PageRank 를 적용하여 랭크가 높은 문장을 선택함으로써 핵심 문장을 추출할 수도 있다. 이때는 문장을 단어열로 잘 나눌 수 있는 토크나이저가 필요하다. 단어와 문장 그래프를 구성하는 방법에 따라 다

양한 변형 방법들이 제안되었다. 문장 간의 유사도를 검색 엔진의 질의어 - 문서 유사도 함수인 BM25 [169] 방법을 이용하는 방법이나 [17], 문장 간의 코싸인 유사도를 이용하는 LexRank [60] 가 제안되었다. 이들은 모두 중요한 단어에 인접한 단어는 중요한 단어이며, 중요한 문장에 인접한 문장은 중요한 문장이라는 가정에 기반한다.

단어나 문장 그래프는 문장이 단어열로 잘 분해되었다는 가정을 한다. 하지만 키워드에 미등록단어 문제가 발생하면 잘못된 문서 요약 결과가 발생한다. 이러한 문제를 해결하기 위하여 단어 추출과 키워드 추출을 동시에 수행하는 방법이 제안되었다. WordRank [35] 는 중국어와 일본어처럼 띠어쓰기가 존재하지 않는 문서집합에서 비지도학습으로 단어를 추출하기 위하여 제안된 방법으로, 문장 내 존재하는 모든 부분단어 (subwords) 간의 인접 빈도를 이용하여 부분 단어 그래프를 구성한다. 그 뒤 HITS 알고리즘을 이용하여 각 마디의 중요도를 계산하면 토크나이저를 이용하지 않으면서도 단어와 키워드가 추출된다.

그러나 WordRank 는 한국어 텍스트에 적용하기가 어렵다. 비록 오류가 존재하더라도 한국어는 띠어쓰기를 기반으로 단어의 경계를 판단할 수 있으며, 문서 집합의 모든 부분단어를 키워드의 후보로 이용할 경우 조사나 어미에 높은 랭크가 할당된다.

이러한 문제점을 해결하기 위하여 KR-WordRank [98] 가 제안되었다. 이는 띠어쓰기 기준으로 나뉘어진 어절의 왼쪽 부분부터 시작하는 부분단어 집합 L 과 어절의 오른쪽 부분부터 시작하는 부분단어 집합 R 로만 부분단어 그래프를 구성한다. 또한 한국어 어절 구조를 이용하는 후처리 과정을 통하여 어절이 단어로 추출되는 경우를 방지한다.

1.5.3 딥러닝 모델을 이용한 요약 기반 문서 요약

딥러닝 모델 기반 인코더 - 디코더 네트워크들을 이용하여 지도학습 방식으로 요약문을 생성하는 문서 요약 방법도 제안되었다 [171]. 그러나 Recurrent Neural Network

기반 인코더 디코더 모델이 기억할 수 있는 단어의 개수에 제약이 있기 때문에 중요한 단어들을 제대로 인식하지 못하여 잘못된 핵심 문장을 생성한다. 이러한 문제를 해결하기 위하여 인코더에 입력되는 입력 문서에서 키워드를 추출하여 디코더에 입력하거나 [152], 어텐션 메커니즘을 이용하여 입력 데이터에서 중요한 단어를 선택하는 방식이 제안되었다 [176, 74]. 즉 최근의 요약 기반 모델들은 추출 기반 방법과 함께 상호 보완적으로 이용되는 방향으로 발전하고 있다.

제 2 장 단어 추출 기법을 이용한 미등록단어 문제 해결 및 이를 이용한 한국어 토크나이저

2.1 서론

단어는 텍스트를 이루는 기본 단위이다. 머신러닝을 이용한 텍스트 분석을 위해서는 주어진 문장을 단어열로 분해하고 이를 벡터로 표현하는 전처리 과정이 필요하다. 이를 위하여 형태소 분석기나 품사 판별기와 같은 토크나이저가 이용된다. 영어를 포함한 많은 언어들은 공백을 기준으로 단어가 나뉘어진다. 하지만 교착어에 포함되는 한국어는 공백을 기준으로 어절이 나뉘며, 하나의 어절은 한 개 이상의 단어 혹은 한 개 이상의 형태소가 결합되어 만들어진다. 그렇기 때문에 한국어의 토크나이징 과업에는 글자열로 이뤄진 어절을 단어 혹은 형태소로 분해하는 단어 분리 (word segmentation) 문제가 포함된다.

토크나이징 과업의 목적은 이를 이용하는 자연어처리 과업의 종류에 따라 다르다. 문서 분류나 문서 군집화와 같은 작업에서는 문서에 대한 분석이기 때문에 각 문서의 정보를 잘 표현할 수 있는 질 좋은 벡터를 만드는 것이 토크나이저의 목적이다. 이 경우에는 문장이 반드시 정확한 단어로 표현될 필요는 없으며, 최근에 제안된 글자 단위의 문서 분류 모델이나 [231], 단어 임베딩 방법인 FastText 는 문장을 단어가 아닌 부분단어 (subword) 로 표현한다 [25, 90].

하지만 토픽 모델링이나 키워드 추출 과업은 사람이 그 결과물을 해석하는 것이 목적인 경우가 많으며, 토크나이징 단계에서 단어가 제대로 인식되지 않으면 토픽을 설명하는 단어들을 해석할 수 없게 된다 [76]. 분석 혹은 해석의 단위가 단어인 자연어처리 과업에서는 토크나이징의 목적은 단어를 제대로 인식하는 것이다.

토크나이저의 접근 방법은 두 가지로 나뉘어진다. 지도학습 기반 접근 방법은 학습 말뭉치를 이용하여 단어 사전과 어절을 단어로 분해하는 확률 모델을 학습한다. 이들은 학습 말뭉치를 이용하여 모호성을 해결하는 모델을 학습한다. 예를 들어 어절 '서울대 공원에'는 '서울대/명사 + 공원/명사 + 에/조사' 혹은 '서울/명사 + 대공원/명사 + 에/조사'로 분해될 수 있는데, 학습 말뭉치로부터 어떤 결과가 더 적합한지 판단하는 모델을 학습한다. 그러나 학습 말뭉치 기반으로 토크나이저를 학습할 경우에는 말뭉치에 자주 등장한 패턴으로 모호성을 해결하는 편향성이 존재한다. 말뭉치에 '서울/명사 + 대공원/명사'의 경우가 더 많이 등장했다면 정답이 '서울대/명사 + 공원/명사'인 경우에도 '서울 + 대공원'으로 단어를 인식한다. 즉 지도학습 기반 접근 방법은 주어진 문장을 학습 데이터의 패턴에 맞춰 해석한다.

이와 반대로 학습 데이터나 사전을 이용하지 않으면서 데이터 기반으로 학습된 통계 정보를 이용하는 비지도학습 토크나이징 방법도 제안되었다. 명사, 동사, 형용사, 부사는 새로운 단어가 발생하는 열린 집합에 해당하는 품사로 [91], 학습 데이터를 기반으로 구축된 토크나이저는 다른 도메인의 텍스트에서만 이용되는 단어를 제대로 인식하지 못할 수 있다. 이러한 미등록단어 문제를 해결하기 위하여 중국어와 일본어 자연어처리 연구에서 비지도학습 단어 추출 연구가 활발히 연구되었다. 최근에는 인공 신경망 기반 번역 연구에서 다양한 언어에 공통으로 적용하기 위한 목적으로, Word Piece Model (WPM)이라는 비지도학습 토크나이저가 제안되었다 [177].

그러나 WPM 을 한국어에 적용할 경우 상대적으로 빈도수가 작은 단어들이 음절 단위로 분해되거나, 빈도수가 높은 어절이 단어열로 분해되지 않고 어절 형태로 처리되는 단점이 있다. 예를 들어 뉴스 기사 문서에서는 '오늘의', '오늘은' 과 같이 빈번한 어절 들은 '오늘 + 의' 나 '오늘 + 은'으로 분해되지 않을 수 있다. 그 결과 토픽 모델링이나 키워드 추출의 결과에 같은 단어를 포함한 다양한 어절이 모두 등장한다.

본 연구에서는 이러한 문제를 해결하기 위하여 한국어에 적합한 비지도학습 단어 추출 방법과 이를 이용한 토크나이저를 제안한다. 제안하는 단어 추출 기법은 음절 단위의 언어모델에 기반하여 단어 점수를 정의하며, 한국어 어절의 구조적 특징을 이용하여 토크나이징을 수행한다.

본 논문의 2.2 에서는 미등록단어의 원인과 이를 해결하기 위한 비지도학습 단어 추출 기법 및 토크나이저들에 대하여 살펴본다. 2.3 에서는 한국어의 어절 구조를 이용한 단어 추출 기법과 이를 이용한 두 가지 토크나이저에 대하여 설명한다. 2.4 에서는 제안된 방법의 성능을 평가한다. 이를 위하여 온라인에서 수집된 영화 리뷰 데이터를 이용하여 감성 분석, 고유 명사 인식 과업, 단어 임베딩을 이용한 유사 단어 탐색 과업을 수행한다. 2.5 장에서는 제안된 알고리즘을 효율적으로 이용할 수 있는 방법 및 제안된 알고리즘의 한계점에 대하여 논의한다.

2.2 관련 연구

한국어의 토크나이징 과업에는 학습 말뭉치와 순차적 레이블링 (sequential labeling) 알고리즘을 기반으로 작동하는 품사 판별기나 형태소 분석기가 이용된다. [160, 185] 이들은 사전을 기반으로 문장에서 단어 혹은 형태소 후보를 만든 뒤, 이들을 조합하여 주어진 문장에 가장 적절한 단어 혹은 형태소 열을 판단한다 [27]. 형태소 분석에는 Hidden Markov Model, Maximum Entropy Markov Model, Conditional Random Field, Structural Support Vector Machine, discrimination based sequential labeling 과 같은 다양한 순차적 레이블링 기법이 이용된다 [105, 138, 106, 199, 201, 24, 151]. 최근에는 딥러닝 모델 기반 순차적 레이블링 방법도 품사 판별에 이용된다 [239, 47].

영어처럼 공백을 기준으로 단어의 경계가 구분되는 언어의 경우에는 공백을 기준으로 문장을 단어열로 분해한 뒤 각 단어의 품사를 추정하는 문제를 해결하도록 모델을

학습한다. 하지만 중국어나 일본어처럼 단어의 경계가 없거나 한국어의 어절처럼 공백만으로는 어절의 경계만 구분되는 언어의 경우에는 사전을 이용하여 문장에서 생성할 수 있는 가능한 모든 단어 후보를 만든 뒤, 가장 적절한 단어를 선택하는데 순차적 레이블링 알고리즘이 이용된다. 이 때 사전에 등록되지 않은 단어는 제대로 인식되지 않는 문제가 발생하는데, 이를 미등록단어 문제라 한다. 특히 한국어의 형태소 분석은 “알려지지 않은 단어는 알려진 형태소의 조합으로 구성된다”고 가정하기 때문에 미등록단어를 여러 개의 형태소로 잘못 분해하는 현상이 발생한다. 표 2.1 은 문장 ‘아이오아이는 걸그룹 이름이다’을 기학습된 형태소 분석기로 분석한 예시다. ‘아이오아이’는 두 모델이 학습하지 못한 미등록단어이기 때문에 이를 ‘아이오 + 아이’로 잘못 인식한다.

Table 2.1: 기학습된 한국어 형태소 분석기를 이용한 문장 분석 예시. (N: 명사, J: 조사, V: 동사, E: 어미, VCP: 동사형 전성어미)

정답 단어열	아이오아이/N + 는/J + 걸그룹/N + 이름/N + 이다/V
꼬꼬마 형태소 분석기	아이오/N + 아이/N + 는/J 걸/그룹/N + 이름/N + 이/VCP + 다/E
트위터 한국어 처리기	아이오/N + 아이/N + 는/J + 걸그룹/N + 이름/N + 이다/J

분석해야 할 문서 집합에서 ‘아이오아이’라는 5음절이 자주 등장하였다면 사람은 이를 하나의 단어라 추정할 수 있지만, 각 문장을 독립적으로 처리하는 형태소 분석기는 이러한 문서 집합 전체로부터 얻을 수 있는 정보를 활용하기 어렵다.

중국어와 일본어는 띄어쓰기를 이용하지 않기 때문에 미등록단어가 발생할 경우 정확한 단어의 경계를 인식하는 것이 더욱 어려워진다. 이를 해결하기 위하여 다양한 단어 추출 방법들이 제안되었다. Branching Entropy (BE) 는 글자 경계에서는 다양한 글자들이 위치한다는 [77] 의 가정을 이용하여 단어 경계를 정의한다 [88]. 식 2.1 는 문장의 부분글자인 $c_{p:q}$ 의 단어 시작 경계 LBE 와 단어 종료 경계 RBE 의 단어 점수 정의이다. RBE 는 단어 $c_{p:q}$ 오른쪽에 등장하는 글자 c_{q+1} 의 분포의 엔트로피로 정의된다. $c_{p:q}$ 가 단어라면 그 오른쪽과 왼쪽에는 모두 다양한 글자가 위치하고, $c_{p:q}$ 가 단어의

부분이라면 등장하는 글자의 수가 제한되어 엔트로피가 작기 때문이다. 이로부터 단어 w 의 단어 점수는 $\min(LBE(w), RBE(w))$ 로 정의된다.

$$\begin{aligned} LBE(c_{p:q}) &= - \sum_{c_{p-1}} P(c_{p-1}|c_{p:q}) \times \log P(c_{p-1}|c_{p:q}) \\ RBE(c_{p:q}) &= - \sum_{c_{q+1}} P(c_{q+1}|c_{p:q}) \times \log P(c_{q+1}|c_{p:q}) \end{aligned} \quad (2.1)$$

Accessor Variety (AV) 는 글자 경계에 등장하는 다른 글자의 종류로 단어의 경계 점수를 정의한다 [62]. 그리고 단어 점수 w 는 두 경계의 점수의 최소값인 $\min(LAV(w), RAV(w))$ 로 정의한다.

$$\begin{aligned} LAV(c_{p:q}) &= \#(c_{p-1}|c_{p:q}) \\ RAV(c_{p:q}) &= \#(c_{q+1}|c_{p:q}) \end{aligned} \quad (2.2)$$

정의된 단어 점수를 이용하여 주어진 데이터의 문장을 단어열로 분해하는 비지도 학습 중국어 단어 분리 알고리즘들이 제안되었다 [235, 63]. 이들은 문장을 단어열로 나누었을 때 문장 내 단어 점수의 합 혹은 평균 단어 점수가 최대가 되는 단어열을 탐색하는 방식으로 작동하며, 데이터에 자주 등장하는 주요한 단어들에 대하여 좋은 인식 능력을 보였다. 하지만 이들은 사전을 이용하는 지도학습 기반 토크나이저의 변수로 이용되기도 하였다 [234, 236, 237, 195, 239]. 지도학습 기반으로 학습된 모델을 이용하여 모호성 해결과 빈도수가 낮은 단어에 대한 인식능력을 높이고, 단어 추출 방법을 이용하여 데이터에 자주 등장하는 미등록단어의 인식능력을 보완하였다. 이처럼 사전 기반 모델과 단어 추출 기반 방법을 함께 이용함으로써 토크나이저가 데이터에 존재하는 모든 문장을 종합적으로 해석할 수 있도록 하였다 [234].

Minimum Description Length (MDL) 기법도 비지도학습 토크나이저에 이용되었

다 [102, 80, 240]. 문장은 단어의 조합이며, 최소한의 단어로 문장을 설명할 수 있다고 가정한다. 그러나 단어는 빈도수가 큰 소수의 단어와 빈도수가 작은 다수의 단어로 이뤄지기 때문에 MDL 의 가정과 단어 분포가 일치하지 않는다 [137]. 그렇기 때문에 언어적 성질을 반영한 제약조건이나 다른 기준을 함께 이용해야 한다 [137, 80].

Word piece model (WPM)은 재귀적 인공신경망처럼 모델이 학습할 수 있는 단어의 개수가 제한적일 때 사용되는 비지도학습 토크나이저이다 [177]. WPM 은 Byte-Pair Encoding 방법을 이용하여 데이터에서 자주 등장하는 글자 단위의 엔그램을 추출한다 [184]. 표 2.2은 WPM 의 예시로, 'makers' 와 같이 자주 이용되는 단어는 그대로 인식되지만 'Jet' 처럼 자주 이용되지 않는 단어는 'J' 와 'et'로 나뉘어진다. 하지만 지나치게 빈번한 'makers' 는 'make + rs' 로 나뉘어지지 않는다. 한국어에서는 빈번한 어절이 단어로 나뉘어지지 않게 된다. 예를 들어 뉴스 기사 문서에서는 '오늘의', '오늘은' 과 같이 빈번한 어절들은 '오늘 + 의' 나 '오늘 + 은'으로 분해되지 않는 경우들이 발생한다. 반대로 빈도수가 매우 작은 고유명사들은 음절 단위로 분해될 가능성이 높다.

Table 2.2: Example of Word Piece Model tokenization result

Sentence : Jet makers feud over seat width with big orders at stake,
Word pieces : _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

이는 WPM 역시 MDL 처럼 데이터의 모든 문장을 최소한의 부분단어로 표현하기 위한 방법이기 때문이다.

2.3 비지도기반 한국어 단어 추출 및 이를 이용한 토크나이저

2.3.1 한국어 어절의 구조 : L + [R]

한국어의 한 어절은 여러 개의 형태소로 구성될 있다. 예를 들어 어절 '보강수업을' 은 '보강/명사 + 수업/명사 + 을/조사' 로 구성되며, 동사 '시작했어'는 '시작/명사 +

하/동사형전성어미 + 았/선어말어미 + 어/어말어미'로 구성된다. 위의 두 예시는 표 2.3처럼 의미를 표현하는 부분과 문법을 표현하는 부분으로 나누어 표현할 수 있다. 복합명사 '보강수업'을 하나의 명사로 취급하면 '보강수업을'은 '보강수업/명사 + 을/조사'로 표현할 수 있다. '시작했어'의 어미들은 명사 '시작'을 과거형 동사 형태로 변환하는 문법 기능을 수행하기 때문에 이들을 하나의 단어 '했어'로 취급한다면 '시작/명사 + 했어/복합형태소'로 표현할 수 있다. 혹은 '시작/명사 + 하/동사형전성어미'를 동사의 어간으로 취급한다면 '시작하/동사 + 았어/복합형태소'로 표현할 수 있다.

Table 2.3: 명사가 포함된 어절의 구조. (N: 명사, J: 조사, V: 동사, E: 어미, EP: 선어말어미, VCP: 동사형 전성어미)

어절	형태소 분석 결과	제안하는 방법의 어절 구조
보강수업	보강/N + 수업/N	보강수업/L + "/R
보강수업을	보강/N + 수업/N + 을/J	보강수업/L + 을/R
시작했어	시작/N + 하/VCP + 았/EP + 어/E	시작/L + 했어/R
시작했어	시작/N + 하/VCP + 았/EP + 어/E	시작했어/L + 어/R

용언은 어간과 어미가 결합되어 어절을 이룬다. '하/동사어간 + 라고/종결어미'는 '하라고'라는 동사 어절을 이루며, 어간은 어절의 왼쪽에 어미는 어절의 오른쪽에 위치한다. 그 외 감탄사, 부사, 관형사는 각각 한 단어가 하나의 어절을 이룬다.

즉 어절은 의미를 지니는 단어와 문법 기능을 수행하는 단어의 결합으로 표현할 수 있다. 의미를 지니는 부분은 어절의 왼쪽에 위치하므로 L로, 문법 기능을 하는 부분은 어절의 오른쪽에 위치하므로 R로 표현할 수 있다. 그러나 '보강수업'처럼 의미를 지니는 형태소로만 어절이 구성될 수도 있는데, 이때는 어절에 L만 존재한다. 이를 정리하면 한국어의 어절은 L + [R] 형태라 정의할 수 있다.

용언이 불규칙활용 되는 경우에도 L + [R] 형태로 표현할 수 있다. '시작했어'의 경우 '시작하/동사'를 L로 가정하면 '시작했 + 어'로 표현한다.

L + [R] 관점은 전통적인 형태소 분석으로 변환할 수 있다. R은 문법 기능을 수

행하는 복합형태소이며 이를 전통적인 형태소로 복원하는 기분석 테이블을 이용하면 $L + [R]$ 관점으로 분석한 결과와 전통적인 형태소 분석의 결과를 상호 변환할 수 있다 [244].

2.3.2 음절 단위의 언어 모델을 이용한 단어 점수

이 논문에서는 L 을 구성하는 음절들 간의 연관성을 새로운 단어 점수 척도로 이용 한다. '보강수'이라는 단어 다음에 높은 확률로 '업'이라는 글자가 등장한다면 '보강수'는 단어의 경계가 아니라는 의미이다. 하지만 '보강수업' 다음에 등장하는 글자의 확률의 최대값이 작다면 이는 '보강수업'이 단어의 경계라는 의미이다. Cohesion 점수는 단어를 구성하는 음절의 언어모델이며, 식 2.3 으로 정의된다 [95].

$$cohesion(C_{1:n}) = \left(\prod_{i=1}^{n-1} P(C_{1:i+1}|C_{1:i}) \right)^{\frac{1}{n-1}} \quad (2.3)$$

Cohesion 점수는 단어를 이루는 음절들 간의 연관성을 측정하기 때문에 1음절 단어에 대해서는 정의하지 않는다. 또한 어절이 $L + [R]$ 구조이기 때문에 L 에 대한 단어 점수만 정의되어도 어절을 L 과 R 로 구분할 수 있다. 그렇기 때문에 어절의 왼쪽에 위치하는 2음절 이상의 부분단어에 대해서만 점수를 정의한다. 길이가 n 인 단어에 대하여 $n - 1$ 번의 조건부 확률을 누적하여 곱하면 길이가 긴 단어일수록 그 값이 작아진다. 이를 보정하기 위하여 누적곱에 $\frac{1}{n-1}$ 승을 취하였다. 이는 표 2.4 처럼 'AB'의 빈도수와 'ABC'의 빈도수가 같다면 'ABC'가 단어일 가능성이 더 높도록 만들며, 이는 복합명사와 같은 단어에 대하여 높은 단어 점수를 부여한다.

Cohesion 점수는 Branching Entropy 와 결합되어 사용될 수 있다. L 이 단어라면 높은 Cohesion 점수를 지니며, 그 경계에 다양한 R 이 위치한다면 Branching Entropy 값도 크다. L 이 단어의 부분이어도 높은 Cohesion 점수를 지닐 가능성이 있지만,

Table 2.4: Cohesion 점수와 Branching Entropy 예시

단어	빈도수	Cohesion 점수	Right-side, Branching Entropy	Cohesion × Right-side Branching Entropy
A	1000	-	0.325	-
AB	100	0.1	1.055	0.287
ABC	100	0.316	0.742	0.664
ABCD	10	0.215	0	0.215
ABCE	5	0.171	0	0.171
ABCF	3	0.144	0	0.144
ABCG	2	0.126	0	0.126
ABH	50	0.224	0	0.224
ABI	100	0.316	0	0.316

Branching Entropy 는 작은 값을 가질 가능성이 높다. 그렇기 때문에 Branching Entropy 는 Cohesion 점수를 보완할 수 있다. 그러나 한국어에서 이용되는 음절의 종류는 중국어의 글자의 종류보다 매우 작기 때문에 모호성이 발생하여 앞서 언급된 [235, 63] 를 그대로 이용할 경우 문장이 1음절의 단어들로 분해되는 현상이 발생한다. 어절의 왼쪽에 등장하는 2음절 이상의 모든 부분단어에 대하여 Cohesion 과 RBE 를 계산한 뒤, 이를 곱하여 최종 단어 점수로 이용할 수 있다.

2.3.3 단어 점수를 이용하는 비지도학습 토크나이저

이 장에서는 앞서 정의한 단어 점수를 이용하는 두 종류의 토크나이저를 제안한다. 문장에 띄어쓰기 오류가 존재하지 않는다면 어절은 $L + [R]$ 이라 가정할 수 있으며, 한 어절에서 단어 점수가 가장 큰 L 로 어절을 이분하면 $L + R$ 로 어절이 분해된다 (그림 2.1). 이를 L-Tokenizer 로 정의하며, 이는 뉴스 기사와 같이 띄어쓰기 오류가 거의 존재하지 않음을 가정할 수 있는 텍스트에 적용하기에 적합하다.

```

D: word - score dictionary
s: sentence
w: eojeol, separated by white space

def tokenize(s, D):
    tokens ← []
    for w in split(s):
        scores ← {}
        for e in range(2, len(w)+1):
            sub ← w[:e]
            scores[sub] = D.get(sub, 0)
        l ← find sub having the largest value
        r ← w[len(l):]
        tokens.append([l, r])
    return tokens

```

Figure 2.1: L-Tokenizer 의사코드

그러나 블로그나 소셜미디어와 같이 온라인 공간에서 임의의 사용자에 의하여 작성되는 다수의 텍스트는 띠어쓰기 오류가 포함되어 있다. 이때는 공백으로 구분되는 단위가 여러 개의 어절일 수 있기 때문에 L-Tokenizer 를 이용하면 R 에 여러 개의 어절이 포함될 수 있다. 이러한 경우에 이용할 수 있는 Max Score Tokenizer 를 제안한다 (그림 2.2).

```

D: word - score dictionary
s: sentence
w: eojeol, separated by white space

def tokenize(s, D):
    subs ← scan subword score (s, D)
    subs ← sort subs by score in reverse order
    tokens ← []
    while subs is not empty:
        t ← pop subs
        tokens.append([t])
        subs ← remove overlapped sub with t
    return tokens

```

Figure 2.2: Max Score Tokenizer 의사코드

그림 2.2 은 띠어쓰기가 잘 지켜지지 않은 문장을 사람이 읽을 때에는 익숙한 단어부터 인식되는 것을 모사하여 작동한다. 그림 2.3은 '파스타가 좋아요'라는 문장에 대하여 네 가지 부분단어와 단어 점수가 주어진 예시이다. 첫 단계에서는 점수가 알려진 모든

부분단어에 대하여 단어 점수를 확인한다. 두번째 단계에서는 점수 기준으로 테이블을 정렬한다. 이는 가장 익숙한 단어부터 인식하는 과정이다. 세번째 단계에서는 점수가 가장 높은 단어인 '파스타'를 선택한 뒤, 이와 겹치는 다른 모든 부분단어를 테이블에서 제거한다. 네번째 단계에서는 테이블에 단어 후보가 존재하지 않을 때까지 3 단계를 반복한다. 그 결과 '파스타가좋아요' 문장은 '파스타 + 가 + 좋아 + 요'로 분리된다.

(a) Initialize tokenize tables			
subword	begin	end	score
파스	0	2	0.3
파스타	0	3	0.7
스타	1	3	0
스타가	1	4	0
타가	2	4	0
타기종	2	5	0
가좋	3	5	0
가좋아	3	6	0
좋아	4	6	0.5
좋아요	4	7	0.2
아요	5	7	0

(b) Sort entries by score			
subword	begin	end	score
파스타	0	3	0.7
좋아	4	6	0.5
파스	0	2	0.3
좋아요	4	7	0.2
스타	1	3	0
스타가	1	4	0
타가	2	4	0
타기종	2	5	0
가좋	3	5	0
가좋아	3	6	0
아요	5	7	0

(c) Select most probable subword & remove overlapped others			
subword	begin	end	score
파스타	0	3	0.7
좋아	4	6	0.5
파스	0	2	0.3
좋아요	4	7	0.2
스타	1	3	0
스타가	1	4	0
타가	2	4	0
타기종	2	5	0
가좋	3	5	0
가좋아	3	6	0
아요	5	7	0

(d) Repeat (c) while candidates exist			
subword	begin	end	score
파스타	0	3	0.7
좋아	4	6	0.5
파스	0	2	0.3
좋아요	4	7	0.2
스타	1	3	0
스타가	1	4	0
타가	2	4	0
타기종	2	5	0
가좋	3	5	0
가좋아	3	6	0
아요	5	7	0

```
scores = {
    '파스': 0.3,
    '파스타': 0.7,
    '좋아요': 0.2,
    '좋아': 0.5
}
```

Figure 2.3: 띠어쓰기 오류파 포함된 문장 '파스타가좋아요' 과정

이처럼 데이터의 띠어쓰기 오류 수준에 따라 서로 다른 전략의 토크나이저를 선택할 수 있다. L-Tokenizer 는 어절이 L + [R] 로 이분된다는 사실을 사전 지식으로 이용하는 것과 같다. 또한 두 토크나이저 모두 문장에서 부분단어를 잘라내고 단어 점수를 확인하는 작업으로만 이뤄져 있기 때문에 확률 모델을 이용하는 토크나이저들과 비교하여 계산량이 매우 작다.

2.4 성능 평가

한국국립국어원에서 배포하는 세종 말뭉치에는 문장 원문과 이를 구성하는 형태소가 태깅되어 있으며 [97], 다양한 한국어 형태소 분석기들은 세종 말뭉치의 품사 구조를 변형하거나 각자의 단어 사전을 추가하여 배포되고 있다. 형태소 분석기들의 성능을 향상하기 위한 연구에서도 세종 말뭉치는 평가 데이터로 자주 이용된다. 하지만 제안하는 방법은 미등록단어가 자주 발생하는 환경에서 사용하기 위한 방법이기 때문에 세종 말뭉치를 이용한 실험에서는 이러한 환경을 재현하기가 어렵다. 또한 WPM 을 비롯한 비지도학습 단어 추출 방법은 특정 주제의 문서 집합이나 특정 날짜의 뉴스와 같이 텍스트 데이터의 주제가 한정적일 때 잘 작동한다. 하지만 세종 말뭉치는 다양한 종류의 문서로부터 다양한 단어와 패턴을 학습하기 위한 데이터로, 단어 추출기가 잘 작동하는 데이터가 아니다 [98]. 또한 띠어쓰기도 잘 지켜지고 있는 데이터이기 때문에 토크나이저가 이용되는 환경과 차이가 있다.

그렇기 때문에 미등록단어 문제와 띠어쓰기 오류가 빈번한 온라인 공간에서 수집한 영화평 데이터를 이용하여 제안하는 모델의 성능을 평가하였다. 네이버 영화로부터 각 영화의 영화평이 5000 개 이상 기록된 152 개의 영화로부터 약 320 만개의 영화평을 수집하였다. 영화평에는 다양한 슬랭 표현 및 배우, 영화 속 캐릭터, 영화 제목과 같은 다양한 고유 명사가 포함되어 있으며, 이들을 제대로 인식하는 것은 영화평 분석의 품질에 큰 영향을 준다.

토크나이저의 성능 평가의 접근법은 정답 단어를 이용하여 결과의 품질을 직접 평가하는 방법과 토크나이저 결과를 이용하는 다른 자연어처리 과업의 성능을 통하여 간접적으로 평가하는 두 가지로 나뉘어진다. 특히 토크나이저는 그 자체가 자연어처리의 최종 과업이 아닌 경우가 많기 때문에 후자의 접근법이 자주 이용된다 [43]. 온라인 공간에서 수집한 영화평 데이터를 이용하여 세 종류의 자연어처리 과업을 수행하였다.

첫째는 영화평과 평점을 이용한 감성 분석으로, 이를 통하여 각 토크나이저의 문장에 대한 벡터 표현 능력을 평가한다.

둘째는 고유 명사의 재현율 측정이다. 각 영화에 등장한 배우나 캐릭터명과 같은 정보는 메타데이터로부터 수집할 수 있으며, 이들은 기학습된 형태소 분석기에 등록되지 않은 고유명사일 가능성이 높다. 이들의 재현율을 측정함으로써 미등록단어의 인식 능력을 평가한다.

셋째는 단어 임베딩을 통한 유사 단어 검색 과업이다. 토크나이징의 결과가 문맥을 보존하는 단어열이라면 단어 임베딩 벡터 역시 잘 학습되어 영화 배우의 유사어는 영화 배우로 검색이 되어야 한다. 영화 배우의 이름에 대해서는 메타 데이터로부터 정확한 사전을 구할 수 있기 때문에 영화 배우 이름의 유사어가 실제로 영화 배우의 이름인지 확인하였다.

KoNLPy 는 파이썬 환경에서 다양한 한국어 형태소 분석기를 이용할 수 있도록 도와주는 라이브러리이다 [160]. 그 중 트위터 한국어 분석기는 세종 말뭉치 뿐 아니라 한국어로 작성된 트윗을 분석할 수 있는 단어 사전을 포함하고 있다. 또한 문장 분석 속도도 매우 빠르기 때문에 [160], 지도학습 기반 형태소 분석기로 이를 비교 실험에 이용하였다.

WPM 은 토크나이저가 이용하는 유닛의 개수를 사용자가 직접 정해야 한다 [212]. 유닛의 개수에 따른 성능의 차이를 확인하기 위하여 3,000 개부터 50,000 개로 유닛의 개수를 다르게 설정하며 실험을 진행하였다. 이후 WPM3000 은 3,000 개의 유닛을 이용하는 WPM 모델을 의미한다. 제안하는 방법은 Cohesion 점수만 이용하는 경우와 Branching Entropy 를 함께 이용하는 두 경우를 모두 비교하였다. 각각의 모델을 'Cohesion' 과 'CSBE' 로 표기한다. 또한 띠어쓰기만을 이용하는 토크나이저도 비교 실험에 이용하였다.

2.4.1 영화평을 이용한 긍부정 분류 성능 평가

영화평에는 1 부터 10 점까지 평점이 함께 부여되어 있다. 평점은 개인마다 기준이 다를 수 있기 때문에 점수를 그대로 이용하기 보다는 긍정과 부정으로 범주화 하여 이용하는 것이 좋다 [158, 159]. 1 부터 3 점을 부정으로, 8 부터 10 점을 긍정으로, 그 외의 점수는 개인의 편차가 클 수 있기 때문에 이용하지 않았다. [210, 90] 는 Bigram 과 Logistic Regression 모델 혹은 나이브 베이지안 모델을 이용하는 모델을 문서 분류의 기본 모델로 이용할 것을 제안하였다. 이에 따라 L2 정규화를 이용하는 Logistic Regression 모델에 Unigram 만 이용하는 경우와 Uni, Bigram 을 함께 이용하는 두 경우의 모델을 이용하였으며, 교차 검증 (cross validation) 을 통하여 일반화 성능을 측정하였다.

Table 2.5: Unigram 과 Uni + Bigram 을 이용한 영화평의 긍부정 분류 과업 성능

model	unigram		uni + bigram	
	accuracy	rank	accuracy	rank
WPM3000	89.12%	10	92.67%	9
WPM5000	89.56%	9	92.95%	8
WPM10000	91.69%	8	93.47%	3
WPM20000	92.23%	4	93.41%	4
WPM30000	92.43%	2	93.35%	6
WPM50000	92.65%	1	93.32%	7
Cohesion	92.27%	3	93.48%	2
CSBE	92.05%	5	93.63%	1
띄어쓰기	92.04%	6	92.13%	10
트위터 한국어 분석기	91.91%	7	93.39%	5

표 2.5 은 토크나이저 별 영화평의 긍부정 분류 성능이다. 긍부정 분류에서는 Bigram 이 중요한 정보를 표현한다. 단어 '재미'는 긍부정을 명확히 표현하지 못하지만 '재미 + 없다'는 이를 표현할 수 있다. 하지만 토크나이저들은 '재미없다'를 '재미 + 없다' 로 구분하기 때문에 Bigram 을 함께 이용하는 경우 Unigram 만 이용하는 경우보다

성능이 증가함을 확인할 수 있다.

제안된 방법은 트위터 한국어 분석기보다 높은 성능을, 많은 수의 유닛을 이용하는 WPM 과도 비슷한 성능을 보인다. 특히 제안하는 모델이 Bigram 을 이용하는 경우에 가장 높은 성능을 보이는데, 이는 영화평 도메인에서 사용되는 표현들을 잘 인식함을 의미한다.

띄어쓰기를 토크나이저로 이용하는 경우에는 Unigram 을 이용하는 경우와 Bigram 을 함께 이용하는 경우의 성능 차이가 거의 없는데, 이는 Unigram 안에 이미 '재미없다'와 같은 어절이 포함되어 있기 때문이다. 그리고 이 성능은 트위터 한국어 분석기의 Unigram 을 이용하는 경우보다도 높게 나타났다.

유닛의 개수가 작은 WPM 은 Bigram 을 함께 이용할 때 성능이 향상되지만, 유닛의 개수가 많은 WPM 은 Bigram 을 함께 이용하여도 성능 향상이 잘 이뤄지지 않는다. 이는 적은 유닛의 WPM 을 이용하여 생성된 Bigram 이 많은 유닛을 이용하는 WPM 의 유닛인 경우가 많기 때문이다. 실제로 WPM3000 에서 Bigram 을 함께 이용하면 약 54k 의 자질이 만들어졌다.

그 외 다른 판별 모델을 이용하여도 영화평의 긍부정 판별의 성능은 93 % 후반을 넘기기 어려웠는데, 이는 영화평 데이터에는 '드디어 개봉했네'처럼 긍부정을 판단하기 어려운 짧은 문장들이 포함되어 있기 때문이다.

2.4.2 메타 데이터를 이용한 고유 명사 재현 능력 평가

영화평 데이터에 등장하는 다양한 단어들을 정확히 알 수는 없지만, 메타데이터로부터 영화 배우의 이름, 극 중 캐릭트 명, 영화 제목과 같은 일부 고유 명사에 대한 리스트를 만들 수 있다. 각 토크나이저들이 이들을 제대로 인식할 수 있는지 평가하였다. '한효주'는 리뷰와 메타 데이터에 자주 등장하는 배우 이름이며, 이는 '한효주 + 가', '한효주 + 의' 처럼 조사와 결합되어 이용된다. 그러나 '한효주'가 기학습된 형태소

분석기의 사전이나 WPM 의 유닛으로 등록되어 있지 않다면 '한효 + 주가'나 '한효 + 주의' 처럼 잘못된 단어열로 분해될 수 있다.

표 2.6 는 영화평 데이터에 각각의 토크나이저를 적용한 뒤, 메타 데이터에 등록된 고유 명사가 단어로 존재하는지 확인한 결과이다. 트위터 한국어 분석기에는 유명한 영화 배우의 이름들이 등록되어 있기 때문에 배우 이름에서 높은 인식능력을 보이지만, 극중 캐릭터 명이나 영화 제목은 미등록 단어일 가능성성이 높다. 제안하는 방법은 데이터에 자주 등장하는 단어들을 추출하기 때문에 배우 이름 뿐 아니라 캐릭터 명이나 영화 제목을 제대로 인식할 수 있다. Cohesion 만 이용하는 경우보다 Branching Entropy 와 함께 이용하는 경우 단어의 인식률이 더 높은데, 이는 배우 이름이나 캐릭터 이름과 함께 특정 조사가 자주 이용될 Cohesion 은 조사를 포함한 어절을 단어로 선택하기 때문이다. R 의 다양성을 정량화하는 Branching Entropy 를 함께 이용할 경우 어절 내 L 과 R 의 경계가 더욱 명확하게 드러날 수 있다. 또한 제안하는 방법은 모든 종류의 단어에서 사전을 이용하는 트위터 한국어 분석기와 비슷한 단어 인식 능력을 보임을 확인할 수 있다. 그러나 WPM 의 경우 유닛의 개수를 증가하여도 인식되는 고유 명사의 개수가 적었다. 이는 WPM 의 문장의 분해 단위가 단어가 아닌, 자주 등장하는 부분단 어이기 때문이다. WPM 은 유닛의 개수를 증가하면 새로운 단어를 유닛으로 추가하는 것이 아니라, 자주 등장하는 어절을 유닛으로 추가한다.

2.4.3 단어 임베딩을 이용한 유사 단어 검색 성능 평가

토크나이징의 결과가 문맥을 보존하는 단어열이라면 단어 임베딩 벡터 역시 잘 학습되어 영화 배우의 유사어는 영화 배우로 검색이 되어야 한다. 영화평에 각 토크나이저를 적용한 뒤, Word2Vec [145] 을 이용하여 단어 임베딩 벡터를 학습하였다. 단어 임베딩 모델은 단어의 빈도수가 작으면 학습이 잘 이뤄지지 않기 때문에 최소빈도수 10 을 넘는 단어만을 학습에 이용하였다. 메타 데이터에 등록된 영화 배우의 이름을 입력하여 상위

Table 2.6: 토크나이저 별 고유 명사 재현율 (배우 이름, 영화 제목, 극 중 캐릭터 이름)

model	배우 이름	영화 제목, 극 중 캐릭터 이름	전체
WPM3000	0	0.13	0.05
WPM5000	0	0.13	0.05
WPM10000	52.05	27.21	43.3
WPM20000	63.93	50.42	59.17
WPM30000	65.99	54.49	61.94
WPM50000	63.21	56.62	60.89
Cohesion	82.23	71.29	78.38
CSBE	88.4	82.08	86.17
띄어쓰기	36.09	31.09	34.33
트위터 한국어 분석기	92.95	77.14	87.38

10 개의 유사어를 검색한 뒤, 이들이 배우 이름인지 확인하였다.

표 2.7 의 첫번째 열은 토크나이저에 의하여 재현된, 빈도수 10 이상의 배우 이름의 개수이다. 사전을 이용하는 트위터 한국어 분석기와 Cohesion 을 이용하는 모델은 800 명이 넘는 이름을 제대로 인식하였음을 확인할 수 있다. 두번째 열은 유사어가 실제로 배우 이름인 비율이며, 세번째 열은 검색된 유사어의 빈도수를 고려한 가중 평균이다.

Table 2.7: 단어 임베딩 벡터를 이용하여 검색된 배우 이름의 유사어 중 배우 이름인 비율

model	재현된 단어 개수	유사어 중 배우 이름 비율	단어 빈도수를 고려한 유사어 중 배우 이름 비율
WPM3000	0	0	0
WPM5000	0	0	0
WPM10000	64	59.06	75.67
WPM20000	159	66.54	77.71
WPM30000	220	67.27	76.2
WPM50000	309	63.66	73.97
Cohesion	847	54.91	73.64
CSBE	649	58.72	71.21
띄어쓰기	696	45.22	68.93
트위터 한국어 분석기	822	64.77	80.87

제안한 모델은 트위터 한국어 분석기와 비슷한 개수의 단어 재현과 근접한 유사어

검색 능력을 보였다. 이는 제안하는 모델이 단어 사전을 이용하는 모델과도 비슷한 문맥 보존 능력이 있음을 의미한다. 하지만 WPM은 단어 임베딩에 이용된 배우의 이름 개수 자체가 상대적으로 매우 적다. 이는 일부의 단어에 대해서만 제대로 인식되고, 많은 단어들이 문맥 정보를 잘 반영하지 못하는 단위로 나뉘어졌음을 의미한다.

2.5 결론

이 논문은 한국어의 어절 구조를 $L + [R]$ 로 가정한 뒤, 미등록단어가 발생하는 L 부분의 단어를 통계 기반으로 추출하는 방법과 이를 이용하는 비지도학습 토크나이저를 제안하였다. 제안하는 방법은 데이터의 띠어쓰기 오류 수준에 따라서 L-Tokenizer와 Max Score Tokenizer를 선택할 수 있다. 제안하는 방법은 온라인 공간에서 수집된 영화평 데이터를 이용하여 성능을 평가하였다. 문장의 긍부정 판별, 고유 명사의 인식, 단어 임베딩을 이용한 유사어 검색 과업을 통하여 제안하는 방법은 WPM 보다도 좋은 단어 인식 능력 및 문서의 벡터 표현 능력이 있으며, 사전을 이용하는 트위터 한국어 분석기와도 비슷한 토크나이징 능력이 있음이 확인되었다.

그러나 음절 단위의 언어모델은 데이터의 부분단어가 모호성이 있을 때 잘 작동한다. 뉴스 데이터에서는 '카메라'라는 단어가 자주 등장하며 나라 이름 '카메룬'은 상대적으로 적게 등장하는데, 이 경우에는 $P(\text{카메룬} | \text{카메})$ 의 값이 매우 작기 때문에 오히려 cohesion('카메') \leq cohesion('카메룬')이 되며, '카메룬'의 Branching Entropy가 이를 상쇄할만큼 크지 않다면, '카메룬의 → 카메 + 룬의'로 잘못 분리된다. 이를 해결하기 위해서는 R 의 정보를 함께 이용하여 '카메룬 + 의'의 토크나이징 점수가 '카메 + 룬의'의 토크나이징 점수보다 크도록 정의할 수 있어야 한다.

혹은 가능하다면 주제가 동일한 문서만을 나눠서 Cohesion 점수를 학습하는 것이 좋다. 아프리카와 관련된 뉴스들에 대해서만 Cohesion을 학습하였다면 '카메룬'의 빈

도수가 충분하여 높은 점수를 얻을 수 있다. 영화 리뷰에서는 영화 별로 리뷰를 나눠서 Cohesion 과 Branching Entropy 를 학습할 수도 있다.

제안된 방법은 단어 사전의 구축을 위하여 전문가의 수작업을 요구하지 않기 때문에, 한국어의 어절 구조가 유지되는 임의의 도메인에 적용할 수 있으며, 간단한 계산 과정만으로 토크나이징이 가능하기 때문에 대량의 문서 집합에 적용하기에 용이하다. 또한 기존의 [234, 237, 239] 연구들처럼 기학습된 모델의 미등록단어 인식 성능을 보완하는데 이용될 수도 있다. 그러나 앞서 언급한 것처럼 음절 단위의 언어모델은 빈도수가 작은 일부 단어들에 대해서 단어 점수가 제대로 학습되지 않는 단점이 있다. 이러한 점을 보완한다면 미등록단어 문제를 해결하는 더 좋은 토크나이저로 발전할 것으로 기대한다.

제 3 장 한국어 어절 구조를 이용한 통계 기반 명사 추출

3.1 서론

단어는 의미를 표현하며 새로운 단어가 만들어지는 열린 집합과 문법 기능을 수행하는 닫힌 집합으로 분류할 수 있다.[91]. 열린 집합의 단어는 새로운 단어가 만들어지기 때문에 미등록단어 문제가 발생한다. 한국어에서 명사는 가장 많이 이용되는 단어로, 세종 말뭉치의 29.12 %에 해당하며, 형태소 어미, 조사, 동사와 형용사의 어간이 각각 19.17 %, 17.11 %, 13.53 % 씩 등장한다 (표 3.1). 또한 명사는 가장 많은 종류의 단어로 구성된 집합이다. 세종 말뭉치의 통계에서 약 14만여종의 명사가 각각 평균적으로 49 번 정도 등장함에 비하여 289 종의 조사는 약 1만 4천번 정도 등장함을 확인할 수 있다. 이는 명사의 다양성이 매우 큼을 의미한다.

Table 3.1: 세종 말뭉치의 형태소 품사 별 통계

형태소 품사	출현 빈도수 (비율)	고유 개수	평균 출현 빈도수
명사	7,124,644 (29.128%)	144,294	49.38
용언의 어미	4,688,406 (19.168%)	3,142	1,492.17
조사	4,184,235 (17.107%)	289	14,478.32
용언의 어간	3,308,743 (13.527%)	7,989	414.16

정확한 명사 인식은 텍스트 분석의 성능을 향상시킨다. 키워드를 이용한 문서 요약이나 토픽 모델링은 단어 분포를 이용하여 그 결과를 해석한다. 명사가 제대로 인식되지 않으면 주요한 단어들이 생략되어 분석 결과의 품질이 떨어진다.

문장을 단어로 인식하는 과정에서 품사 판별기나 형태소 분석기가 이용된다. 특히 형태소 분석기는 ”학습 데이터에 등장하지 않은 단어는 알려진 형태소의 조합으로 구성”된다고 가정한다. 하지만 한국어는 한자를 이용하는 언어이기 때문에 각 음절이

단어인 경우가 많으며, 이는 미등록단어를 여러개의 짧은 형태소로 분해하는 결과를 야기한다. 또한 한국어에서 이용되는 음절의 수가 작기 때문에 음절 간의 모호성이 발생한다. 세종 말뭉치에서는 300 글자가 말뭉치 전체의 89.48 % 를, 1,000 글자가 말뭉치 전체의 96.08 % 를 차지하며, 이들로부터 수백만개의 단어가 형성된다. 표 2.1 는 기학습된 형태소 분석기인 꼬꼬마와 트위터 한국어 처리기를 이용하여 예문 '아이오아이는 결그룹 이름이다'의 형태소 분석을 한 결과이다. 두 모델에는 '아이오아이'가 학습 데이터에 등장하지 않았기 때문에 '아이오'와 '아이'라는 두 명사로 분해하였다.

Conditional Random Field (CRF), Structured Support Vector Machine (S-SVM) 나 Hidden Markov Model (HMM) 과 같은 순차적 레이블링 기반 품사 판별기는 앞 뒤에 등장하는 단어를 이용하여 미등록단어의 품사를 추정할 수 있다 [186, 185, 151, 114]. 그러나 모든 단어 후보에 대해 품사를 추정하는 것은 계산량이 많고 부정확한 추정을 할 수 있기 때문에 품사 판별기와 형태소 분석기는 단어 사전에 등장하는 단어에 대해서만 품사를 추정한다.

이러한 문제를 해결하기 위해서는 사용자에 의한 사전 추가가 필요하다. 하지만 새로운 도메인의 문서를 분석하기 전에는 어떠한 단어가 이용되는지 예상하기 어렵기 때문에, 현실적으로는 문서 분석 후 잘못 분해된 단어를 교정하는 후처리 작업이 수행된다.

명사의 오른쪽에 등장하는 단어들의 분포를 이용하면 명사에 대한 추정이 가능하다 [119]. 그림 3.1 의 예문에서 단어 'A'의 오른쪽에는 '-는', '-에서', '-로' 와 같이 자주 이용되는 조사들이 등장한다. 이러한 정보를 바탕으로 단어 'A'가 명사라고 추정할 수 있다.

이 장에서는 어절의 오른쪽에 등장하는 부분어절의 분포를 이용하여 데이터 기반으로 명사를 추출하는 방법을 제안한다. 한 어절에서 명사 오른쪽에는 빈칸, 조사들이

Figure 3.1: A가 포함된 문장 예시

문장 1: ”A는 좋아”
문장 2: ”A에서 만나”
문장 3: ”A로 가자”

자주 등장하며, 표 3.1에서 살펴볼 수 있듯이 조사의 종류는 제한적이기 때문에 이를 명사 추출의 변수로 이용할 수 있다. 또한 조사는 닫힌 집합에 속하는 단어이기 때문에 도메인에 상관없이 명사를 추출하는 정보로 이용할 수 있다.

3.2 관련 연구

CRF, S-SVM,이나 HMM 같은 순차적 레이블링을 이용하는 품사 판별기는 새로운 단어의 품사를 추정할 수 있다. [186, 185, 151, 114]. 단어 후보 x_i 의 품사를 추정하는 정보로 앞 뒤에 등장하는 단어인 $(x_{i-1}, t_{i-1}, x_{i+1}, t_{i+1})$ 를 이용하면 x_i 의 품사 추정이 가능하다 [186, 185, 151, 114]. 하지만 형태소 분석이나 품사 추정 과정에서 가능한 모든 단어 후보 x_i 에 대하여 품사를 추정하는 것은 지나치게 큰 계산 비용이 듈다. 그렇기 때문에 현실적으로는 문맥 정보는 사전에 등장한 단어 후보 x_i 가 여러 개의 품사를 지니는 모호성을 해결하는데에만 이용된다. 또한 학습 데이터에 존재하는 단어만 문맥정보 x_{i-1}, x_{i+1} 로 이용할 수 있다.

규칙 기반으로 한국어의 미등록단어 문제를 해결하려는 연구도 제안되었다. 단어의 앞음절 (prefix)이나 뒷음절 (suffix)을 기반으로 품사를 추정하는 방법 [119]과 음절 단위의 템플릿 기반으로 새로운 단어를 인식하는 방법 [84]이 제안되었다. 하지만 템플릿 기반 방법은 새로운 단어에 템플릿의 음절이 포함되어 있을 경우 실패하는 경우가 많다. 예를 들어 “~/명사 + 은/조사” 템플릿은 ‘은’으로 끝나는 명사를 잘못 인식하게 되는데, 명사는 그 자체로 어절을 이루는 경우가 존재하기 때문에 ‘손나은’이라는 고유 명사가 ‘손나’처럼 잘못 인식되는 경우가 발생한다. 이를 해결하기 위하여 예외 규칙과

오토마타를 이용한 명사 추출 방법도 제안되었지만 [116], 이는 오토마타에 포함된 규칙에 해당하는 명사만 추출할 수 있다는 단점이 있다. 예를 들어 'ㅆ'이나 'ㅋ'은 명사에 자주 등장하지 않는 자음이기 때문에 이러한 자음이 포함된 신조어들은 규칙 기반으로 추출될 수 없다.

단어 임베딩은 같은 품사의 단어를 탐색하는데 이용될 수 있다. 단어 임베딩 벡터는 의미가 비슷한 단어를 비슷한 벡터로 표현할 뿐 아니라, 같은 품사의 단어들 역시 비슷한 벡터로 표현한다 [18]. 그렇기 때문에 Word2Vec 과 같은 모델을 이용하여 단어 임베딩 벡터를 학습한 뒤, 알려진 명사를 이용하여 명사를 확장할 수 있다 [146]. 하지만 Word2Vec 은 문장이 단어열로 정확히 나뉘어져 있다고 가정하며, 한국어는 토크나이징이 이뤄지지 않으면 단어 임베딩 벡터를 학습할 수 없다. 그리고 토크나이징은 단어 사전이 제대로 구축되어 있을 때 좋은 품질을 보이기 때문에 두 방법은 서로가 서로의 전제 조건이 된다. FastText [26] 는 토크나이저를 이용하지 않는 단어 임베딩 방법이지만, 이 방법은 단어의 형태적 유사성과 문맥을 함께 보존하는 단어 임베딩 방법이기 때문에 명사의 유사어로 해당 명사를 포함하는 어절을 학습한다.

그러나 그림 3.1 에서 볼 수 있듯이 어절 내의 단어 분포 정보를 이용하면 명사를 쉽게 추정할 수 있다. [188] 학습된 형태소 분석기의 복제 실험을 수행하였는데, 앞 뒤 문맥을 모두 이용하는 형태소 분석기를 이용하여 어절들을 분석한 뒤 어절을 형태소 열로 변환하는 함수를 학습하였다. 이는 앞 뒤에 등장하는 단어들을 변수로 이용하지 않음을 의미하는데, 그럼에도 불구하고 98.31 % 의 재현율을 보였다. 즉 다수의 어절의 형태소 분석은 그림 3.1 처럼 한 어절 내에서 가능함을 의미하며, 이러한 관점은 명사 추출에도 이용될 수 있다.

3.3 한국어 어절의 L + [R] 구조를 이용한 명사 추출

이 장에서는 주어진 문서 집합에서 어절의 구조적 특징과 통계를 이용하여 명사를 추출하는 알고리즘을 제안한다. 이는 수작업으로 명사 사전을 구축하는 비용을 줄여줄 뿐 아니라, 수작업에 의하여 놓치기 쉬운 단어들도 사전에 포함할 수 있다는 장점이 있다. 한국어의 어절은 그림 1.1처럼 여러 개의 복합 형태소로 구성되어 있다. 하지만 단어 추출을 위해서는 2.3.1 절처럼 어절 구조를 단순하게 정의해야 한다.

3.3.1 L-R 그래프를 이용한 명사 추출

제안하는 명사 추출 방법은 한국어 어절의 L + [R] 구조를 이용한다. 명사는 의미를 지니기 때문에 L에 해당하며, R로는 조사와 그 외의 복합 형태소가 등장한다. 명사는 전성어미와 결합되어 동사나 형용사로 이용될 수 있는데, 이 때 사용되는 전성어미는 '-하, -되, -이, -있'처럼 제한적이며 동사나 형용사의 어간과 같은 형태이다. 즉 형태적으로는 명사의 오른쪽에 조사와 용언의 표현형이 등장할 수 있으며, 이들의 종류는 제한적이다. 용언은 어간이 포함된 음절이 L에 해당하며 R로는 어미 혹은 어미가 포함된 복합형태소가 등장한다. 그 외의 품사는 한 형태소가 하나의 어절을 이룬다. 그렇기 때문에 한 어절을 (L, R)로 분해한 뒤, R을 이용하여 L이 명사인지 판별할 수 있다.

그림 3.1에서 'A'를 명사로 추정할 수 있는 근거는 모든 문장에서의 'A'에 대한 R 분포다. 한 단어 L이 명사라면 R에 다양한 조사와 용언이 분포하며, L이 용언이라면 R에 다양한 어미들이 분포할 것이다. 이처럼 데이터의 모든 어절의 전역적인 정보를 이용하면 정확한 단어의 추정이 가능하다 [234].

그러나 어절이 주어졌을 때 L과 R의 경계를 알지 못한다. 제안하는 모델은 이러한 문제를 해결하기 위하여 그림 3.2 같이 세 단계로 구성되어 있다.



Figure 3.2: 제안하는 명사 추출기의 프레임워크

첫 단계에서는 주어진 모든 어절을 가능한 조합으로 이분하여 L-R 그래프를 만든다. 어절 내에 R 이 없을 수도 있기 때문에 그림 3.3 처럼 한 어절이 L 이 될 수도 있다. 어절 '드라마를'은 네 종류의 (L, R) 쌍으로 분해되며, 데이터의 모든 (L, R) 조합의 빈도수를 계산하여 이를 두 마디 L 과 R 사이의 호의 가중치로 정의한다.

Figure 3.3: 어절 '드라마를'로부터 생성할 수 있는 (L, R) 쌍 예시

`[(드, 라마를), (드라, 마를), (드라마, 를), (드라마를, ")]`

'드라마/L'는 명사이기 때문에, 이와 함께 등장한 R 은 조사가 다수이다. 표 3.2 의 왼쪽 행은 하루의 뉴스 문서 집합에서 학습한 L-R 그래프이다. '드라마'의 R 에는 '-를', 87 회, '-의' 67 회 처럼 조사들이 다수 등장하며, 용언의 어간인 '시작했/L'의 R 에는 어미들이 다수 등장함을 확인할 수 있다.

Table 3.2: '드라마/L' 와 '시작했/L' 를 포함하는 가장 빈번한 (L, R) 쌍 예시

'드라마'를 포함한 상위 10개의 (L, R)	'시작했'를 포함한 상위 10개의 (L, R)
(드라마,) = 278	(시작했, 다) = 2900
(드라마, 를) = 87	(시작했, 습니다) = 229
(드라마, 의) = 67	(시작했, 고) = 170
(드라마, 가) = 60	(시작했,는데) = 75
(드라마, 는) = 50	(시작했, 던) = 69
(드라마, 애) = 47	(시작했, 다른) = 68
(드라마, 에서) = 42	(시작했, 을) = 67
(드라마, 로) = 26	(시작했, 다고) = 50
(드라마, 나) = 18	(시작했, 으며) = 44
(드라마, 틱한) = 17	(시작했, 어요) = 41

두번째 단계에서는 학습된 L-R 그래프에서 길이가 2 이상인 L 에 대하여 R 의 빈도

수 벡터를 이용한 명사 유무를 판별한다. 한국어는 표의 문자의 성격이 있기 때문에 1 음절 단어는 명사인 경우가 많으며, 추출이 되어도 해석이 모호한 경우가 많다. 또한 L 의 빈도수와 R 의 종류가 사용자가 정의한 최소값 이상인 경우에 대해서만 판별 작업을 수행한다. L 의 단어 후보가 실제로 명사라면 다양한 조사들이 R 에 등장할 것이지만, 한 번 등장한 L 은 R 이 단어의 일부인지 혹은 조사인지를 확인하기 어렵다. 또한 한 종류의 R 과 함께 등장하는 L 은 명사가 아닌 다른 단어의 부분글자일 가능성성이 높기 때문이다. 단, L 중 길이가 긴 단어부터 명사 유무를 판단한다. '아이디/L', '아이디어/L' 모두 명사이기 때문에 이들과 연결된 R 은 대부분 조사이다. 하지만 명사로만 이루어진 '아이디어'라는 어절로부터 '아이디/L + 어/R' 가 생성된다. '어/R' 은 용언의 어미 표현형으로 자주 등장하는 부분단어이기 때문에 '아이디어/L'의 빈도수가 '아이디/L' 보다 클 경우, '아이디/L'가 명사가 아닌 것으로 판단될 수 있다. 이러한 문제를 해결 하기 위하여 L 의 길이가 긴 단어부터 명사 점수를 계산한 뒤, 사용자에 의하여 지정된 명사 점수의 임계값보다 클 경우 L-R 그래프에서 해당 단어가 포함된 모든 (l, r) 쌍을 그래프에서 제거한다. '아이디어/L' 가 명사로 추출된 뒤에는 ('아이디/L', '어/R') 이 존재하지 않기 때문에 '아이디/L' 역시 명사로 추출될 가능성이 높아진다. L 의 명사 판별에 이용하는 판별기는 세종 말뭉치를 이용하여 학습한다. 세종 말뭉치를 이용한 판별기 학습에 대한 내용은 다음 장에서 다룬다.

마지막 단계에서는 후처리 과정을 통하여 잘못 추출된 명사를 걸러낸다. 가능한 모든 부분어절 L 에 대하여 명사 유무를 판별했기 때문에 명사 후보에는 단어의 부분 글자들이 포함되어 있다. 명사로 잘못 판단된 L 은 세 종류로 분류할 수 있다. 첫번째는 명사의 마지막 어절이 조사와 같은 글자인 경우이며, 이는 " $N_{sub} + J$ " 형태이다. 예를 들어 명사 '떡볶이' 는 그 자체로 어절을 이루는 경우가 많으며, '-이'가 포함되어 있기 때문에 "떡볶/명사 + 이/조사" 로 판별될 수 있다. 하지만 '떡볶' 다음에 등장하는 글

자는 대부분 '이' 이기 때문에 Branching Entropy 값이 매우 작고, '뛰복이' 다음에는 다양한 조사도 등장하기 때문에 큰 Branching Entropy 값을 지닌다. 이처럼 명사로 추출된 단어의 마지막 음절이 조사에 해당하며 이를 제거한 부분음절 역시 명사로 추출된 경우, Branching Entropy 를 계산하여 그 값이 임계값 (예, 0.5) 보다 작은 부분 어절은 명사가 아닌 것으로 재분류 한다.

두번째 종류는 조사의 일부 글자가 명사에 포함된 경우로, “ $N + J_{sub}$ ” 형태이다. 예를 들어 어절 '대학생과의'는 '대학생/명사 + 과의/조사'로 구분되어야 한다. 하지만 '-의' 역시 자주 이용되는 조사이기 때문에 '대학생과/명사 + 의/조사'로 판별되는 경우가 발생한다. 이때는 위의 경우와 반대로 명사 '대학생'은 다른 조사들과 함께 등장한 경우가 많기 때문에 높은 Branching Entropy 를 지니지만 '대학생과' 다음에 등장하는 글자는 대부분 '-의' 이기 때문에 낮은 Branching Entropy 를 지닌다. 그러므로 명사로 추출된 단어의 마지막 음절이 조사의 일부이며 이를 제거한 부분음절 역시 명사로 분류된 경우, 긴 단어의 Branching Entropy 가 임계값 (예, 0.5) 보다 작다면 이를 명사가 아닌 것으로 재분류 한다.

마지막 종류는 복합 명사이다. 어절 '소수집단의' 는 '소수/명사 + 집단/명사 + 의/조사'로 분류되어야 하지만, L-R 그래프에 의하여 '소수집단'이 명사로 추출된다. 만약 '소수'와 '집단'이 자주 이용되는 명사라면 이는 각각 명사로 추출된다. 길이가 긴 명사가 길이가 짧은 다른 명사열로 구성될 경우 이를 분해하여 복합명사를 분리할 수 있다. 하지만 분석 목적에 따라서는 '유엔안전보장이사회'처럼 복합명사를 그대로 이용하기도 한다. 그러므로 마지막 종류의 후처리는 사용자의 선택에 의하여 실행한다.

제안하는 명사 추출 방법의 의사코드는 3.4 과 같다. 제안된 방법은 사용자에 의하여 정의된 명사 접수의 최소값 t , l 과 r 의 최소 빈도수 c , 그리고 문장열로 구성된 D 가 주어졌을 때, 세 가지 과정을 거쳐 D 로부터 명사를 추출한다. 첫째, D 와 c 를 이용

하여 L-R 그래프를 생성한다. 둘째, 길이가 긴 l 부터 명사 점수를 계산하여 그 점수가 t 보다 클 경우 이를 추출된 명사 집합 N 에 추가한다. 그 뒤, l 이 포함된 어절로부터 만들어진 모든 (l^*, r^*) 을 L-R 그래프에서 제거한다. 셋째, 후처리 과정을 거쳐 잘못 추출된 명사들을 제거한다.

D : dataset
 t : threshold of noun score
 c : minimum count of word

```

def extract_nouns ( $D, t, c$ ):
     $G = ((L, R), E) \leftarrow$  construct L-R graph ( $D, c$ )
     $N =$  for  $l$  in reverse sort by length ( $L$ ):
         $s \leftarrow$  noun score ( $l$ )
        if  $s \geq t$ :
            remove  $(l^*, r^*)$  from  $G$  such as  $l$  is left substring of  $l^* + r^*$ 
             $N \leftarrow L \cup \{l\}$ 
     $N \leftarrow$  postprocessing ( $N$ )
    return  $N$ 
```

Figure 3.4: 제안하는 명사 추출 방법의 의사 코드

3.3.2 세종 말뭉치를 이용한 명사 판별 분류기 학습

세종 말뭉치는 L-R 그래프에서 L 이 명사인지를 판단하는 판별기를 학습하는데 이용될 수 있다. 세종 말뭉치는 구어체와 문어체의 문서들로 이뤄진 형태소 분석용 말뭉치로, 1,560,437 개의 고유어절이 10,807,777 번 등장하는 1,054,566 문장으로 이뤄진 학습 데이터이다. 이 중 명사와 용언이 L 로 위치한 어절을 학습 데이터로 이용하였다.

세종 말뭉치는 어절을 형태소 수준으로 분해한 데이터이기 때문에 이를 $L + [R]$ 구조로 변형하였다. 복합 명사는 단일 명사로 변환하며, 전성어미와 명사가 포함된 어절은 전성어미 이후의 복합 형태소를 하나의 단어로 변환하였다. 용언은 어간을 L 로 어미를 R 로 변형하였으며, R 이 복합 형태소일 경우에는 이를 하나의 단어로 변형하였다. 어간이 활용된 경우에는 어간이 포함된 음절을 L 로 정의하였다.

Table 3.3: 세종 말뭉치의 어절을 $L + [R]$ 구조로 변형한 예시

형태소 분석 관점	$L + [R]$ 구조 관점
졸업/N + 논문/N + 의/J	졸업논문/N + 의/R
시작/N + 하/VCP + 았/E + 다/E	시작/N + 했다/R
하/V + 았/E + 다/E	했/V + 다/R

그 결과 197,651 종류의 L 과 22,394 종류의 R 이 생성되었으며, 이들의 최소빈도 수가 임계값보다 작은 경우는 데이터에서 제거하였다 (표 3.4. 학습 데이터의 레이블은 L 이 명사인지의 유무를 Boolean 벡터로 표현하였다.

Table 3.4: L 과 R 의 최소빈도수 이상 조건을 만족하는 단어의 종류

최소 빈도수	고유 L 개수	고유 R 개수
$L \geq 0, R \geq 0$ (without filtering)	197,651	22,394
$L \geq 5, R \geq 5$	50,233	5,361
$L \geq 10, R \geq 10$	31,550	3,515
$L \geq 30, R \geq 15$	15,106	2,770

판별 모델에 따른 성능 차이를 확인하기 위하여 5 - 교차 검증 (5-folds Cross validation) 을 이용하였다. 교차 검증의 평가 과정에 이용되는 데이터는 학습에 이용되지 않기 때문에 명사의 단어 추출 능력을 의미한다.

표 3.5 는 L2 regularization + Logistic Regression (L2LR), L1 regularization + Logistic Regression (L1LR), Bernoulli Naïve Bayes (BNB), Linear kernel + Support Vector Machine (SVM-L), RBF kernel + Support Vector Machine (SVM-RBF), Feed forward neural network (FNN) 의 명사 판별 성능이다. 피드 포워드 네트워크의 히든 레이어의 구조에 따라 각각 세 종류를 실험하였다. $h=(5,)$ 는 5개 유닛으로 구성된 1개의 히든 레이어를 이용한 경우이며, $h=(50,10)$ 은 각각 50개, 10개의 유닛으로 구성된 두 개의 히든 레이어를 이용한 네트워크를 의미한다. L1 정규화 모델은 R 의 일부를 이용하지 않기 때문에 본 실험에서는 사용하지 않았다. 데이터의 종류에 따른

영향력도 확인하기 위하여 표 3.4에 기록된 세 종류의 학습 데이터를 모두 이용하였다. 각각은 $(50,223 \times 5,361)$, $(31,550 \times 3,515)$ 그리고 $(15,106 \times 2,770)$ 크기의 행렬이다. 모든 벡터는 L의 빈도수의 영향을 제거하기 위하여 L2 정규화를 통한 유닛 벡터로 변형하였다.

Table 3.5: 데이터셋과 판별 모델 별 5 - 교차 검증을 이용한 명사 판별 성능 비교

		Dataset		
Algorithm		$L \geq 30, R \geq 15$	$L \geq 10, R \geq 10$	$L \geq 5, R \geq 5$
L2LR	L2LR $\lambda=1e-5$	0.9951	0.9947	0.9946
	L2LR $\lambda=1e-3$	0.9963	0.9956	0.9957
	L2LR $\lambda=0.01$	0.9964	0.9954	0.9955
	L2LR $\lambda=0.25$	0.9963	0.9949	0.9951
	L2LR $\lambda=1$	0.9958	0.9942	0.9945
	L2LR $\lambda=4$	0.9948	0.9935	0.9939
L1LR	L1LR $\lambda=0.25$	0.9959	0.9956	0.9938
	L1LR $\lambda=1$	0.9951	0.9948	0.9936
	L1LR $\lambda=4$	0.9942	0.994	0.9926
SVM-L	SVM-L $\lambda=0.1$	0.9941	0.9946	0.9936
	SVM-L $\lambda=1$	0.9938	0.9946	0.994
	SVM-L $\lambda=10$	0.9928	0.9936	0.9928
SVM-RBF	SVM-RBF $\lambda=0.1$	0.9877	0.9947	0.9938
	SVM-RBF $\lambda=1$	0.8341	0.9948	0.9942
	SVM-RBF $\lambda=10$	0.8341	0.9939	0.9929
BNB		0.9939	0.99	0.986
FNN	FNN $h=(5,)$	0.9968	0.9957	0.9951
	FNN $h=(20,)$	0.9968	0.996	0.9951
	FNN $h=(50,10)$	0.9964	0.9959	0.9949

대부분의 모델은 99 % 이상의 명사 판별 능력을 보이지만 (표 3.5), SVM-RBF 모델은 regularization 변수 값에 따라 성능의 편차가 존재했다. 이는 RBF 커널을 이용할 경우 R 벡터의 유clidean 거리를 기반으로 모델이 학습되지만, R 빈도벡터는 sparse 형태이며, 이때는 코싸인이나 자카드 척도가 적합하기 때문이다. 하지만 선형 커널을 이용하는 SVM-L은 내적을 기반으로 작동하기 때문에 안정적인 성능을 보였으며, Support Vectors의 개수 역시 선형 커널을 이용할 때에는 3 ~ 6 % 이지만, RBF 커널을 이용하면 20 ~ 31 % 을 보였다. 이 결과로부터 내적 혹은 확률을 기반으로 작동하는 분류 모델이라면 그 종류에 관계없이 안정적인 성능을 보임을 알 수 있고, 모

델의 해석력을 위하여 Logistic Regression 모형을 이용하였다. 세종 말뭉치를 이용한 판별기는 한 번 학습한 뒤 명사를 추출할 때마다 재사용 할 수 있다.

3.4 성능 평가

제안하는 방법의 정량적인 성능 평가를 위하여 기학습된 한국어 형태소 분석기인 (1) 꼬꼬마 형태소 분석기, (2) 한나눔 형태소 분석기, 그리고 (3) 트위터 한국어 처리 기와 명사 인식 능력을 비교하였다. 그러나 위 모델들이 이용한 학습 데이터에는 세종 말뭉치가 포함되어 있다고 알려져 있다. 세종 말뭉치에는 정확한 단어의 품사 정보가 포함되어 있기 때문에 이를 이용한 명사 인식 능력을 평가하였으며, 기학습된 한국어 형태소 분석기가 겪는 미등록단어 상황을 재현하기 위하여 온라인에서 수집한 뉴스 기사에서의 명사 인식 능력을 추가로 평가하였다. 평가를 위하여 L2 regularization 이 포함된 Logistic Regression 을 이용하였으며, 세종 말뭉치를 이용하여 L 과 R 의 빈도수가 각각 30, 15 이상인 학습 데이터 ($L \geq 30, R \geq 15$) 로 분류 모델을 학습하였다.

3.4.1 세종 말뭉치를 이용한 성능 평가

제안하는 방법을 이용하여 세종 말뭉치에서 L-R 그래프를 구축하였다 (표 3.6). 이들 중 10 번 이상 등장한 62,448 개의 L 에 대하여 명사 판별 유무를 판단하였으며, 후처리 과정을 거쳐 48,248 개의 단어가 명사로 추출되었다.

Table 3.6: 세종 말뭉치로부터 구축된 L – R 그래프 통계

# of subwords in L	# of subwords in R	# of edges
173,620	71,297	2,022,472

표 3.7 는 기학습된 모델과 제안하는 방법의 명사 인식 정밀도 (precision) 와 재현율 (recall) 이다. 최소 빈도수가 10 이상인 명사만 성능 평가에 이용하였으며, 복합

명사는 단일 명사로 취급하였다. 예를 들어 기학습된 분석기가 '명사 + 명사 + 조사'로 분류한 어절의 경우 두 개의 명사를 합쳐 하나의 명사로 취급하였다. 제안하는 방법은 96 % 의 정밀도와 95.8 % 의 명사 인식 재현율을 보여준다. 그러나 꼬꼬마 형태소 분석기는 세종 말뭉치를 이용하여 학습하였음에도 불구하고 제안하는 방법보다 낮은 정밀도를 보였다. 이는 주로 고유명사에서 잘못된 분석을 수행하였기 때문인데, 사람 이름 '전형은'과 '전형/명사', '은/조사'가 모두 학습 데이터에 존재하더라도 '전형은'은 자주 등장하지 않는 단어이기 때문에 확률 모형에 기반하여 형태소 분석을 할 경우 자주 이용되는 '전형/명사 + 은/조사'로 오분류하는 경우가 발생한다. 하지만 단어 사전을 이용하는 트위터 한국어 분석기는 세종 말뭉치에 등장하는 명사 사전을 직접 이용하기 때문에 제안하는 방법보다 높은 재현율을 보였다.

Table 3.7: 제안하는 방법과 기학습된 모델들의 세종 말뭉치에서의 명사 인식 성능

	정밀도	재현율	F1 점수
제안하는 방법	0.960	0.958	0.959
꼬꼬마 형태소 분석기	0.920	0.963	0.941
한나눔 형태소 분석기	0.866	0.897	0.881
트위터 한국어 처리기	0.945	0.874	0.908

표 3.8 는 제안하는 방법이 세종 말뭉치로부터 추출한 빈도수가 낮은 12 개의 명사 예시이다. 괄호는 각각 (Logistic Regression 의 판별 확률, 등장 빈도수)이다. 이들은 주로 고유 명사인데, 고유 명사들은 상대적으로 등장하는 빈도수가 낮더라도 다양한 조사들과 함께 이용되기 때문에 명사로 추출되었다. 표 3.14 에는 빈도수가 낮은 명사 100 개의 예시, 표 3.9 는 명사 확률과 빈도수를 함께 고려한 12 개 명사 예시이다. 대부분 여러 맥락에서 이용되는 일반 명사들이 추출되었으며, 이를 역시 다양한 조사와 함께 등장하기 때문에 명사로 잘 추출되었음을 확인할 수 있다. 즉 제안하는 방법은 일정 수준 이상 등장한 명사에 대해서는 추출이 잘 이뤄짐을 확인할 수 있다. 표 3.15에는 추출된 명사 중 빈도수가 높은 명사 100 개의 예시가 기록되어 있다. 여기에는

'것으'와 같이 잘못 추출된 예시도 포함되어 있다. '것으'는 주로 '것으로'의 맥락에서 등장하는데, 제안하는 방법이 1음절 명사 '것'을 추출하지 않았기 때문에 후처리 과정을 통과한 오류이다.

Table 3.8: 세종 말뭉치에서 명사로 추출된 빈도수가 작은 12 개의 명사 예시 (Logistic Regression 의 판별 확률, 출현 빈도수)

전대미문 (0.998, 17)	생산자물가 (0.998, 19)	런닝머신 (0.998, 10)	가와 (0.998, 36)
루츠 (0.998, 11)	다민족 (0.998, 14)	몇발 (0.998, 11)	마을공동 (0.998, 14)
벽산 (0.998, 13)	당중앙위원회 (0.998, 28)	법률구조 (0.998, 22)	짱마 (0.998, 11)

Table 3.9: 세종 말뭉치에서 명사로 추출된 12 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)

자신 (0.988, 16087)	머리 (0.982, 6369)	이야기 (0.943, 9209)	연구 (0.935, 8279)
우리 (0.925, 44821)	시간 (0.940, 13144)	각각 (0.997, 2233)	생활 (0.946, 7005)
사회 (0.973, 19184)	나라 (0.962, 8903)	당시 (0.981, 4521)	어머니 (0.926, 9036)

3.4.2 뉴스 기사와 온라인 문서를 이용한 성능 평가

세종 말뭉치는 각 단어에 대한 품사 정보가 기술되어 있기 때문에 정량적인 측정이 가능하다. 하지만 제안하는 방법과 기학습된 형태소 분석기들은 모두 세종 말뭉치를 학습 데이터로 이용하기 때문에 미등록단어 문제가 발생하는 실제 문제에 대한 평가가 따로 이뤄져야만 한다.

사람 혹은 사건의 이름과 같은 미등록단어가 포함되어 있는 뉴스 기사에서 제안하는 방법과 기학습된 형태소 분석기의 명사 인식 성능을 측정하였다. 그러나 뉴스에 등장한 단어의 품사 정보가 없기 때문에 정확한 정량적 평가가 어렵다. 이를 해결하기 위하여 온라인 한국어 단어 사전 (네이버 한국어 사전)과 나무위키 데이터베이스의 페이지 타이틀을 근사 명사 사전으로 이용하였다. 일반적으로 이용되는 명사는 온라인 한국어 사전에 포함되어 있지만 고유명사들은 한국어 사전에 등재되지 않은 경우가 많다. 그러나 나무위키 데이터베이스는 웹공간에서 집단적으로 작성되는 위키피디아 형태의

문서 집합으로, 다양한 사건 및 은어까지 기술되어 있기 때문에 다양한 미등록단어가 포함되어 있다. 평가 시 한 어절에서 여러 개의 명사가 인식되는 경우에는 최장일치법을 이용하였다. 예를 들어 어절 '아이오아이는'에서는 그룹 이름 '아이오아이'와 '아이', 그리고 컴퓨터 도메인의 '아이오 (IO)'가 명사로 인식되는데, 이 때는 '아이오아이'를 어절의 명사로 선택하였다.

뉴스 기사에서의 명사 인식 성능을 평가하기 위하여 2016년 10월 20일의 뉴스 기사 30,092 건을 웹 공간에서 수집하였다. 표 3.10 와 3.11 는 각각 나무위키 데이터베이스와 온라인 한국어 사전을 명사 사전으로 이용하였을 경우, 뉴스 기사에서의 명사 인식 성능이다. 뉴스 기사에 등장하는 모든 명사가 사전에 기록되어 있거나 사전의 모든 명사가 뉴스에 등장하는 것이 아니기 때문에 표 3.7 보다는 낮은 정밀도와 재현율을 보이지만, 제안하는 방법과 기학습된 모델들에 대하여 동일한 평가를 수행하기 때문에 상대적인 명사 인식 능력은 평가할 수 있다.

기학습된 모델은 학습 데이터에 등장한 단어에 대해서는 인식이 잘 이뤄지지만, 미등록단어에 대해서는 낮은 재현율을 보인다. 반면, 제안하는 방법은 기학습된 모델보다 높은 재현율을 보이며, 비슷한 정밀도, 높은 F1 점수를 보여준다. 이는 제안하는 방법은 기학습된 모델보다 새로운 명사들에 대한 인식 능력이 좋음을 의미한다.

Table 3.10: 제안하는 방법과 기학습된 모델들의 뉴스 기사에서의 명사 인식 성능 (나무위키 데이터베이스를 사전으로 이용한 경우)

	정밀도	재현율	F1 점수
제안하는 방법	0.8672	0.5484	0.6719
꼬꼬마 형태소 분석기	0.8968	0.2941	0.4429
한나눔 형태소 분석기	0.8778	0.3228	0.4721
트위터 한국어 처리기	0.8897	0.3467	0.499

뉴스 기사는 단어의 품사가 태깅된 데이터가 아니기 때문에 정확한 재현율을 계산하기는 어렵다. 한 단어가 명사를 포함한 여러 종류의 품사로 인식되는 경우에는 이를

Table 3.11: 제안하는 방법과 기학습된 모델들의 뉴스 기사에서의 명사 인식 성능 (온라인 한국어 사전을 이용한 경우)

	정밀도	재현율	F1 점수
제안하는 방법	0.9797	0.4721	0.6372
꼬꼬마 형태소 분석기	0.9931	0.3461	0.5133
한나눔 형태소 분석기	0.9858	0.4005	0.5695
트위터 한국어 처리기	0.9843	0.3928	0.5615

명사로 재분류 하였다. '대한'은 동사의 활용 형태이거나 '대한민국'의 '대한'일 수 있는데, 한 번이라도 명사로 인식된 단어에 대해서는 모두 명사로 취급하였다. 그렇기 때문에 정밀도는 상향되어 평가되지만, 제안하는 방법과 기학습된 모델에 대하여 동일한 기준이 적용되었기 때문에 상대적인 성능 평가가 가능하다.

제안하는 모델은 뉴스 데이터를 이용한 경우에도 기학습된 형태소 분석기보다 높은 명사 인식 능력을 보였으며, 특히 고유명사를 잘 인식하였다 (표 3.10).

표 3.12 와 표 3.16 는 뉴스 기사에서 추출된 빈도수가 낮은 명사의 예시이며, 표 3.13 와 3.17 는 빈도수가 높은 명사의 예시이다. 세종 말뭉치를 이용한 실험 결과와 비슷하게 제안하는 방법론은 뉴스 기사에서도 명사를 잘 추출하며 '규제기관'이나 '다음날' 같은 복합명사 뿐 아니라 '플리마켓' 같은 고유명사도 잘 추출할 수 있음을 확인하였다. 또한 '2017학년도'나 '8화' 와 같은 명사구 역시 하나의 명사로 추출된다.

Table 3.12: 뉴스 기사에서 명사로 추출된 빈도수가 작은 12 개의 명사 예시 (Logistic Regression 의 판별 확률, 출현 빈도수)

제품기획 (1, 39)	친동생 (1, 17)	특혜입학 (1, 32)	다음날 (1, 104)
매일 (1, 1635)	강화함 (1, 20)	성도 (1, 15)	2017학년도 (1, 72)
8화 (1, 23)	규제기관 (1, 13)	2005년 (1, 264)	플리마켓 (1, 21)

Table 3.13: 뉴스 기사에서 명사로 추출된 12 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)

무단 (0.999, 21605)	재배 (0.964, 20610)	지난 (0.995, 14054)	오후 (0.992, 7711)
재배포 (0.997, 20443)	20일 (0.943, 20870)	뉴시스 (0.997, 9950)	함께 (0.976, 7946)
금지 (0.986, 19959)	기자 (0.750, 29222)	이번 (0.995, 7755)	저작권자 (0.999, 7556)

3.5 결론

명사는 열린 집합의 단어이기 때문에 신조어나 도메인 별로 이용되는 전문 용어에 의하여 미등록단어 문제가 발생한다. 그러나 명사가 제대로 인식되지 않으면 키워드 추출이나 동의어 분석과 같은 텍스트 분석의 품질이 저하된다. 이 장에서는 단순화한 한국어의 어절 구조인 L + [R] 을 이용하는 통계 기반 명사 추출기를 제안하였다. 명사는 어절의 L 에 위치하며, 단어의 오른쪽에 위치하는 R 의 분포를 이용하여 L 이 명사인지 판별할 수 있다.

세종 말뭉치와 뉴스 기사를 이용하여 제안하는 방법과 기학습된 형태소 분석 기의 명사 인식 능력을 비교하였다. 두 종류의 데이터를 이용한 실험 모두에서 기학습된 모델보다 높은 재현율 및 가장 높은 정밀도와 F1 점수를 보였으며, 특히 고유 명사와 같은 도메인 별로 용례가 다른 명사들을 잘 추출하였다. 그렇기 때문에 추출된 명사집합은 기학습된 형태소 분석기의 사용자 사전으로 추가되어 이용할 수 있다.

그러나 제안된 방법은 다음의 한계가 있다. 첫째, 어절 내 단어의 분포만을 이용하기 때문에 잘못 추출되는 명사들이 존재한다. 대화체에서는 '그리고는', '그리고서'와 같은 관용어구들이 자주 이용되는데, '-는, -서' 와 같은 조사에 의하여 '그리고'가 명사로 추출될 수 있다. 이는 둘째, 학습 데이터에 등장하지 않은 R 이 포함된 어절에서는 R 의 일부가 합쳐져 명사로 추출될 수 있다. '시작합니다만'은 '시작/명사 + 합니다만/R' 으로 인식되어야 하나, 구어체인 '합니다만'이 세종 말뭉치에 존재하지 않기 때문에 '시작합니다'가 명사로 추출될 수 있다. 셋째, 제안하는 방법은 기학습된 형태소 분석기의

사용자 사전을 보강하는 것일 뿐, 기학습된 모델을 보강하지는 못한다.

그럼에도 불구하고 한국어에서 가장 많이 이용되며 미등록단어 문제도 가장 많은 명사를 자동으로 추출할 수 있다는 점에 의의가 있다. 이 방법은 맞춤법이 틀린 명사도 단어로 인식할 수 있기 때문에 맞춤법 교정과 함께 품사 판별을 하는 모델로도 확장이 가능하다.

Table 3.14: 세종 말뭉치에서 명사로 추출된 빈도수가 작은 100 개의 명사 예시 (Logistic Regression 의 판별 확률, 출현 빈도수)

전대미문 (0.998, 17)	생산자물가 (0.998, 19)	런닝머신 (0.998, 10)	가와 (0.998, 36)
루츠 (0.998, 11)	다민족 (0.998, 14)	몇발 (0.998, 11)	마을공동 (0.998, 14)
벽산 (0.998, 13)	레포 (0.998, 50)	법률구조 (0.998, 22)	쌍마 (0.998, 11)
학생처 (0.998, 13)	희대 (0.998, 17)	워렌 (0.998, 13)	현도 (0.998, 11)
필생 (0.998, 13)	대한상 (0.998, 32)	도광 (0.998, 10)	란상 (0.998, 10)
멸망시 (0.998, 23)	당중앙위원회 (0.998, 28)	육당 (0.998, 11)	대학강 (0.998, 11)
토말 (0.998, 12)	업무협 (0.998, 18)	합복 (0.998, 10)	두대 (0.998, 11)
신강 (0.998, 11)	타종 (0.998, 17)	매번 (0.998, 141)	세브란스 (0.998, 25)
초감각 (0.998, 10)	서낭 (0.998, 31)	정책자 (0.998, 13)	태환 (0.998, 18)
새턴 (0.998, 14)	항일 (0.998, 146)	지례 (0.998, 116)	쌍마자동차 (0.998, 10)
오종 (0.998, 12)	흔외 (0.998, 13)	도범 (0.998, 10)	덕원 (0.998, 15)
앨리 (0.998, 43)	열역학 (0.998, 30)	감상선 (0.998, 45)	통합전 (0.998, 10)
한국산업 (0.998, 16)	가계신용 (0.998, 14)	죄단 (0.998, 27)	파죽 (0.998, 10)
지역환경 (0.998, 14)	한국과학기술 (0.998, 50)	이정연씨 (0.998, 22)	사회체 (0.998, 60)
외향 (0.998, 39)	린다 (0.998, 28)	한국기독교 (0.998, 14)	안암 (0.998, 23)
우랄 (0.998, 18)	일렉트로닉 (0.998, 10)	뿔뿔이 (0.998, 63)	방신영 (0.998, 10)
재크 (0.998, 14)	오텐 (0.998, 13)	초유 (0.998, 15)	건설기 (0.998, 12)
현종 (0.998, 18)	연쇄살인 (0.998, 18)	도큐 (0.998, 12)	가향 (0.998, 22)
주간한국 (0.998, 13)	방송문화 (0.998, 12)	이바노프 (0.998, 13)	국가주 (0.998, 69)
매스 (0.998, 414)	경기민요 (0.998, 10)	농공 (0.998, 16)	해마다 (0.998, 401)
통상교섭본부 (0.998, 10)	경영상 (0.998, 33)	선거관리 (0.998, 13)	범패 (0.998, 57)
지용 (0.998, 44)	천혜 (0.998, 37)	옥황 (0.998, 11)	세계여성 (0.998, 14)
우르 (0.998, 184)	보슈 (0.998, 26)	이크 (0.998, 20)	선제 (0.998, 127)
비정형 (0.998, 12)	파초 (0.998, 10)	무반 (0.998, 34)	수간 (0.998, 12)
김덕수 (0.998, 17)	김영동 (0.998, 10)	양질 (0.998, 69)	반부 (0.998, 30)

Table 3.15: 세종 말뭉치에서 명사로 추출된 100 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)

자신 (0.988, 16087)	머리 (0.982, 6369)	이야기 (0.943, 9209)	연구 (0.935, 8279)
우리 (0.925, 44821)	시간 (0.940, 13144)	각각 (0.997, 2233)	생활 (0.946, 7005)
사회 (0.973, 19184)	나라 (0.962, 8903)	당시 (0.981, 4521)	어머니 (0.926, 9036)
자기 (0.982, 13801)	시대 (0.979, 6250)	기준 (0.998, 1845)	자리 (0.945, 6920)
사람 (0.880, 53855)	정도 (0.941, 11979)	현재 (0.970, 5805)	약간 (0.992, 2294)
하나 (0.962, 18922)	정보 (0.983, 5462)	최고 (0.995, 2534)	전화 (0.969, 4570)
한국 (0.963, 16477)	정치 (0.959, 8911)	국가 (0.964, 6423)	지역 (0.951, 6178)
인간 (0.974, 12364)	스스로 (0.991, 4036)	의미 (0.941, 8930)	서울 (0.914, 9662)
세계 (0.970, 12362)	그들 (0.946, 10513)	생각 (0.812, 31930)	조선 (0.968, 4602)
문제 (0.947, 17843)	더욱 (0.976, 6172)	공동 (0.989, 3313)	결과 (0.954, 5858)
미국 (0.976, 10757)	정부 (0.961, 8110)	일반 (0.974, 5172)	세상 (0.951, 6057)
문화 (0.976, 9604)	개인 (0.984, 4944)	생명 (0.980, 4488)	경우 (0.853, 17191)
모두 (0.969, 10734)	대학 (0.956, 8678)	동안 (0.951, 7661)	문학 (0.948, 6279)
소리 (0.962, 11897)	마음 (0.941, 10643)	그것 (0.854, 21254)	다음 (0.892, 11766)
이상 (0.963, 11723)	고개 (0.992, 3445)	관계 (0.943, 8379)	영화 (0.954, 5746)
경제 (0.979, 8355)	국민 (0.974, 6000)	기업 (0.962, 6109)	북한 (0.970, 4292)
때문 (0.890, 28019)	각종 (0.998, 1908)	교육 (0.942, 8205)	시민 (0.960, 5171)
사람들 (0.925, 18332)	여자 (0.936, 10822)	얼굴 (0.936, 8721)	변화 (0.957, 5353)
이러 (0.965, 10441)	현대 (0.979, 5278)	중심 (0.973, 4841)	중국 (0.953, 5705)
그녀 (0.937, 15761)	어떤 (0.916, 13572)	중요 (0.943, 7941)	세기 (0.978, 3617)
사이 (0.974, 8705)	자체 (0.981, 4889)	학교 (0.944, 7773)	얘기 (0.916, 8879)
것으 (0.952, 12063)	민족 (0.976, 5409)	가장 (0.913, 10956)	효과 (0.983, 3091)
역사 (0.973, 8286)	필요 (0.928, 11273)	아버지 (0.943, 7747)	가치 (0.978, 3442)
일본 (0.962, 10184)	대부분 (0.986, 4093)	아닌 (0.952, 6825)	언어 (0.967, 4320)
전체 (0.990, 4730)	않은 (0.964, 6795)	여성 (0.938, 8058)	최근 (0.966, 4386)

Table 3.16: 뉴스 기사에서 명사로 추출된 빈도수가 작은 12 개의 명사 예시 (Logistic Regression 의 판별 확률, 출현 빈도수)

제품기획 (1, 39)	친동생 (1, 17)	특혜입학 (1, 32)	다음날 (1, 104)
매일 (1, 1635)	강화함 (1, 20)	성도 (1, 15)	2017학년도 (1, 72)
8화 (1, 23)	규제기관 (1, 13)	2005년 (1, 264)	플리마켓 (1, 21)
베이직 (1, 50)	민정 (1, 834)	아시아계 (1, 11)	헬스조선 (1, 73)
178회 (1, 14)	주중 (1, 65)	구약 (1, 41)	1962년 (1, 23)
상충 (1, 20)	학교법인 (1, 51)	1984년 (1, 34)	부정입학 (1, 22)
스킨 (1, 115)	슬립 (1, 22)	스마트시티 (1, 28)	1996년 (1, 73)
클리닉 (1, 13)	선거사무장 (1, 22)	전야 (1, 27)	레터링 (1, 15)
높임 (1, 11)	깻잎 (1, 19)	물약 (1, 20)	1989년 (1, 94)
소폭 (1, 191)	이태성 (1, 19)	저지대 (1, 14)	밑바닥 (1, 13)
오키나와 (1, 12)	얼굴인식 (1, 15)	신라시대 (1, 13)	매료 (1, 29)
분업화 (1, 20)	강력팀장 (1, 12)	35세 (1, 13)	전자신문 (1, 1660)
상대사업자 (1, 11)	징용 (1, 27)	이용고객 (1, 12)	취재원 (1, 358)
불용 (1, 22)	연평도 (1, 13)	23살 (1, 12)	새마을 (1, 337)
1993년 (1, 42)	경직 (1, 51)	민음사 (1, 12)	문화체육관광 (1, 297)
11월1일 (1, 23)	애청자 (1, 51)	비범 (1, 33)	맨투맨 (1, 195)
2001년 (1, 80)	오래전 (1, 60)	발각 (1, 18)	정보통신 (1, 156)
풍력 (1, 58)	전임 (1, 53)	이론적 (1, 11)	박원순 (1, 146)
조난 (1, 25)	경량 (1, 135)	서민금융 (1, 48)	엠넷 (1, 129)
원청 (1, 17)	시판 (1, 18)	지방간 (1, 26)	신산 (1, 88)
정무수석 (1, 13)	90년대 (1, 32)	기내 (1, 44)	87년 (1, 79)
27회 (1, 15)	선판매 (1, 11)	시청역 (1, 11)	1988 (1, 74)
병역의무자 (1, 14)	연초 (1, 116)	김인권 (1, 16)	축산 (1, 60)
키움 (1, 114)	도덕 (1, 60)	카공족 (1, 13)	유해성 (1, 60)
거침 (1, 77)	북경 (1, 34)	성사 (1, 81)	라임 (1, 54)

Table 3.17: 뉴스 기사에서 명사로 추출된 12 개의 명사 예시 (정렬 기준 = 판별 확률 × 출현 빈도수)

무단 (1, 21605)	가능 (0.994, 4849)	투자 (0.873, 4549)	이런 (1, 2716)
재배포 (0.997, 20443)	19일 (0.927, 5573)	저작 (0.67, 7628)	대통령 (0.674, 5941)
금지 (0.987, 19959)	공개 (0.985, 4789)	조사 (0.955, 3631)	당시 (0.91, 3253)
재배 (0.964, 20610)	최근 (0.983, 4729)	모두 (0.985, 3400)	발생 (0.978, 2811)
20일 (0.943, 20870)	세계 (0.987, 4688)	필요 (0.978, 3428)	모습 (0.752, 4678)
기자 (0.75, 29222)	방송 (0.84, 6421)	확인 (0.954, 3575)	국민 (0.784, 4293)
지난 (0.995, 14054)	시작 (0.97, 4473)	설명 (0.958, 3538)	정부 (0.732, 4876)
뉴시스 (0.997, 9950)	이후 (0.984, 4323)	한편 (0.976, 3393)	현재 (0.778, 4266)
이번 (0.995, 7755)	제보 (0.993, 4178)	사용 (0.913, 3798)	이용 (0.871, 3393)
오후 (0.992, 7711)	다양 (1, 4079)	발표 (0.964, 3399)	경우 (0.767, 4304)
함께 (0.976, 7946)	기업 (0.872, 5340)	기대 (0.953, 3464)	보도 (0.972, 2677)
저작권자 (1, 7556)	사진 (0.553, 12985)	문화 (0.922, 3684)	실시 (0.992, 2546)
진행 (0.948, 8123)	국내 (0.926, 4591)	중국 (0.804, 4814)	현장 (0.88, 3216)
때문 (0.999, 6799)	국회 (0.971, 4139)	출현 (0.959, 3362)	글로벌 (0.999, 2476)
한국 (0.846, 9386)	이상 (0.892, 4809)	지역 (0.794, 4894)	마련 (0.965, 2655)
서울 (0.509, 25359)	사랑 (0.914, 4578)	연합뉴스 (0.806, 4687)	지원 (0.815, 3716)
관련 (0.976, 6680)	운영 (0.916, 4537)	계획 (0.903, 3731)	시장 (0.764, 4206)
뉴스1 (0.961, 6321)	자신 (0.906, 4592)	참여 (0.96, 3276)	증가 (0.964, 2597)
참석 (0.984, 5941)	우리 (0.704, 7452)	문제 (0.796, 4737)	처음 (0.991, 2439)
예정 (0.996, 5741)	이날 (0.763, 6340)	공감 (0.775, 4955)	반영 (0.998, 2401)
뉴스 (0.687, 11340)	북한 (0.864, 4909)	11월 (0.98, 3090)	의혹 (0.797, 3752)
제공 (0.96, 5781)	제작 (0.907, 4390)	기록 (0.919, 3425)	브랜드 (0.928, 2747)
오전 (1, 5298)	대표 (0.621, 9242)	상황 (0.851, 3961)	사이 (0.89, 2979)
경제 (0.95, 5361)	스타 (0.946, 3951)	판단 (0.971, 3019)	영화 (0.706, 4703)
미국 (0.846, 6730)	사람 (0.69, 7391)	생각 (0.847, 3963)	대비 (0.99, 2391)

제 4 장 단일주제 문서 집합 요약을 위한 그래프 랭킹 기반 키워드와 핵심 문장 추출

4.1 서론

문서 집합은 내용을 대표하는 몇 개의 키워드와 요약 문장을 통하여 요약될 수 있으며, 이러한 단어와 문장을 추출하는 자연어처리 과업을 문서 요약 (summarization)이라 한다 [224].

문서 요약의 과업은 접근 방식에 따라 추출 기반 (extractive approach) 접근법과 요약 기반 (abstractive approach) 접근법으로 나뉘어진다 [224]. 추출 기반 방식 접근법은 문서 집합을 대표 할 수 있는 단어 혹은 문장을 데이터에서 선택하는 방법이다. 이 접근 방식은 TextRank [144] 와 같은 그래프 랭킹 기반 방법들이 오랫동안 이용되었으며, [161, 153], 최근에는 딥러닝 모델을 이용한 방법도 제안되고 있다 [171]. 요약 기반 접근 방식은 딥러닝을 이용한 자연어처리 기술을 이용하여 최근에 급부상하고 있는 접근 방식이다. 최근의 단어 임베딩을 바탕으로 한 문장 생성 방법의 발전은 [18, 57, 219, 152] 질 좋은 요약 문장의 생성을 가능하게 하였다 [152]. 하지만 요약 기반 접근 방식은 정답 요약 문장을 학습 데이터로 요구하며, 모델의 학습 비용이 크다. 이와 반대로 추출 기반 접근 방식은 통계 기반으로 작동하기 때문에 학습 데이터가 필요하지 않으며, 데이터에 존재하는 문장에서 핵심 문장을 선택하기 때문에 문법 오류가 적고 의미가 명확한 문장으로 문서를 요약한다. 최근에는 두 가지 방식의 장점을 모두 이용하기 위한 연구들도 제안되고 있다 [16, 19, 74].

문서 요약 작업은 문서 내 주제의 다양성에 따라서도 접근 방법이 달라진다. 문서 집합 내 주제가 다양할 경우에는 각각의 주제들을 모두 요약할 수 있어야 한다 [224].

이를 해결하기 위하여 문서 집합을 비슷한 주제의 부분 집합으로 나눈 뒤 요약 과업을 수행하는 작업도 제안되었다 [65, 64]. 하지만 하나의 문서를 요약하거나 문서의 개수가 여러 개더라도 문서 내 주제의 다양성이 적다면 TextRank 와 같은 그래프 랭킹 기반 방법을 이용할 수 있다.

문서 요약 방법은 토크나이저의 성능에 의존한다. TextRank 는 토크나이저를 이용하여 문장을 단어열로 변환한 뒤, 특정 간격 내의 두 단어가 함께 등장한 빈도수를 이용하여 단어 그래프를 만들거나 문장 간 유사도를 바탕으로 문장 그래프를 만든다. 만약 중요한 단어가 토크나이징 단계에서 제대로 인식되지 않는다면, 이 단어는 키워드로 선택되지 못한다. 딥러닝 기반 요약 방법 역시 토크나이징 단계에서 단어가 제대로 인식되지 않으면 이에 해당하는 단어 임베딩 벡터를 생성하지 못한다. 영어와 같이 공백을 기준으로 단어의 경계가 명확히 인식되는 언어에서는 큰 문제가 발생하지 않지만, 교착어의 한 종류인 한국어에서는 키워드가 미등록단어로 인식될 경우 요약 결과의 품질이 크게 떨어진다.

이를 해결하기 위하여 앞의 2 장과 3 장에서 소개한 미등록단어 문제 해결을 위한 단어 추출 기법과 토크나이저가 이용될 수도 있다. 하지만 위 방법들은 문서 집합의 규모가 클 경우에 잘 작동하는 방법이다. 4 장에서는 토크나이저에 의존하지 않으며 한국어 미등록단어 문제와 키워드, 핵심 문장 추출을 동시에 해결하는 그래프 랭킹 기반 추출 기반 문서 요약 방법을 제안한다. 제안하는 방법은 문서 내 주제의 종류가 단일한 상황을 위한 방법으로, 키워드 추출 과정에 단어 추출 단계가 포함되며 문서 내에서 자주 등장하는 단어에 대한 단어 추출 능력이 상대적으로 좋기 때문에 키워드 추출에 용의하다. 또한 키워드 추출 결과를 이용한 핵심 문장 추출 방법도 함께 제안한다.

4.2 관련 연구

문서 내 주제가 단일할 경우에는 그래프 랭킹을 이용한 추출 기반 문서 요약 방법이 자주 이용되었다. TextRank [144] 는 대표적인 방법으로, 키워드와 핵심 문장을 선택하여 문서 집합을 요약한다. TextRank 는 ”중요한 단어와 함께 등장하는 단어는 중요하다”는 가정으로 키워드를, ”중요한 문장과 비슷한 문장과 비슷한 문장은 중요하다”는 가정을 바탕으로 핵심 문장을 추출한다.

TextRank 를 이용한 키워드 추출 과정은 다음의 과정으로 이뤄진다. 첫째, 토크나이저를 이용하여 문장을 단어열로 분해한다. 단어 중 명사, 동사, 형용사, 부사의 품사만을 이용하며, 그 외의 다른 품사 단어는 제거한다. 모든 품사를 이용할 경우 문법 기능을 수행하며 빈도수가 높은 단어들이 키워드로 선정되기 때문이다.

둘째, 한 문장에서 사용자가 지정한 간격 (window) 안에 떨어진 두 단어의 co-occurrence 빈도를 계산한 뒤, 각 단어를 마디로 두 단어의 co-occurrence 빈도를 호의 가중치로 정의한다. 단어 간 간격은 일반적으로 5 부터 7 로 설정한다. 그림 4.1 (a) 는 한국어 문장에서 생성된 단어 그래프의 예시이다.

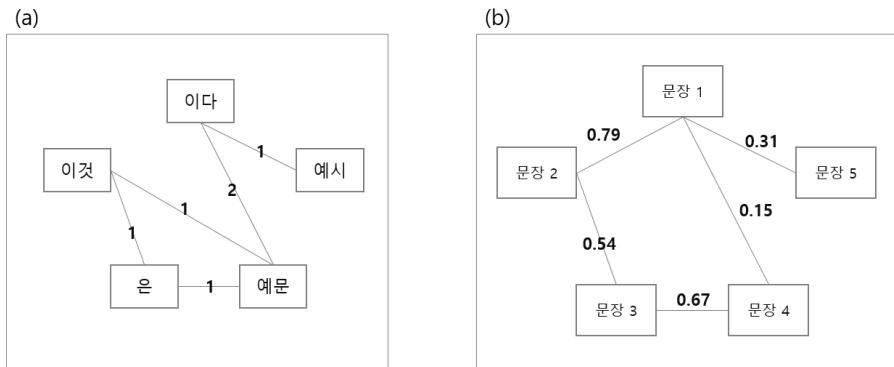


Figure 4.1: (a) TextRank 의 co-occurrence 를 이용한 단어 그래프 예시, (b) TextRank 의 문장 간 유사도를 이용한 문장 그래프 예시

셋째, PageRank [157] 를 이용하여 단어 그래프 내의 각 마디의 랭크를 계산한다. PageRank 의 식 1.25 에서는 모든 호의 가중치가 동일하지만, 그림 4.1 (a) 에서는 호마다 가중치가 다르다. 그렇기 때문에 식 4.1 처럼 한 마디에서 다른 마디로 이동하는 가중치의 합이 1 이 되도록 정규화 한 가중치를 이용하여 랭크를 계산한다. 빈도수가 높은 단어는 다른 단어들과 함께 등장한 빈도가 높기 때문에 연결된 마디의 가중치의 합이 크다. 그렇기 때문에 TextRank 는 문서 집합 내에서 빈도수가 높은 단어에 높은 랭크를 부여한다. 하지만 빈도수가 조금 작더라도 랭크가 높은 다른 단어들과 같은 문장에서 자주 등장하는 단어의 랭크도 높아진다.

$$PR(u) = c \times \sum_{v \in v \rightarrow u} \frac{w_{vu}}{\sum_{w \in v \rightarrow w} w_{vw}} PR(v) + (1 - c) \times \frac{1}{N} \quad (4.1)$$

문법 기능을 수행하는 조사나 어미는 다른 단어들보다 상대적으로 빈번하게 출연하기 때문에 (표 3.1), 이들을 모두 단어 그래프에 포함하면 조사나 어미가 가장 높은 랭크를 가지게 학습된다. 그러므로 반드시 무의미한 단어를 제거한 뒤 단어 그래프를 생성해야 한다.

TextRank 의 키워드의 후보는 단어 그래프를 구성하는 모든 단어들이기 때문에 토크나이저가 중요한 단어를 정확히 인식하지 않으면 그 단어는 키워드가 될 수 없다. 특히 소셜미디어나 영화평과 같은 데이터에서는 고유명사가 미등록단어인 문제가 빈번하며 이들이 키워드일 가능성성이 높다. 그러므로 TextRank 를 이용하여 키워드를 추출할 때에는 단어 사전을 반드시 보강해야 한다.

WordRank [35] 는 이러한 문제를 해결할 수 있는 방법이다. WordRank 는 중국어와 일본어처럼 띄어쓰기가 이뤄지지 않은 언어에서 그래프 랭킹을 기반으로 단어를 추출하는 방법으로, 문장 내 가능한 모든 부분단어를 생성한 뒤, 인접한 부분단어의 빈도수를 기반으로 부분단어 그래프를 만든다. WordRank 는 ”부분단어 그래프에서는

단어와 연결된 마디가 단어이며, 잘못된 부분단어와 연결된 마디는 잘못된 부분단어”라는 가정을 바탕으로 단어를 추출한다. 그럼 4.2 (a) 처럼 단어나 어절의 앞 뒤에는 다른 단어나 어절이 등장하며, 그 종류가 다양하다. 하지만 그림 4.2 (b) 처럼 단어가 아닌 부분단어의 앞 뒤에는 소수의 부분 단어들이 등장한다. 그렇기 때문에 부분단어 그래프에 PageRank 를 적용하면 단어인 마디들의 랭크가 높게 계산된다. 부분단어 그래프는 TextRank 가 이용하는 단어 그래프와 같은 구조이기 때문에 단어를 추출함과 동시에 키워드를 추출하는 능력이 있다.

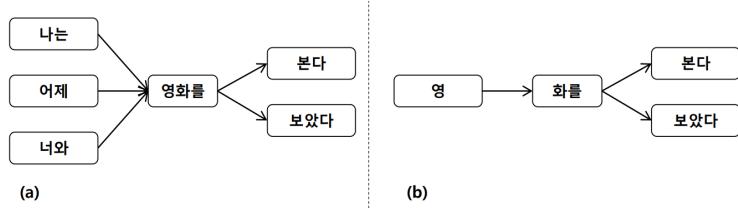


Figure 4.2: (a) 어절을 마디로 이용한 경우, (b) 잘못된 어절을 마디로 이용한 경우

그러나 WordRank 를 한국어 데이터에 적용하면 다음과 같은 문제가 발생한다. 첫째, 옳지 않은 글자가 단어로 추출될 수 있다. 아래의 예시는 WordRank 의 1 음절부터 3 음절까지의 단어후보이다. 첫 문장에서는 두 어절에 걸쳐진 ’의날씨’는 단어 후보가 되어서는 안되지만, WordRank 는 이 글자를 단어 후보로 선택한다. 만약 다른 문장에서도 두 어절에 걸쳐 ’의날씨’라는 글자가 자주 등장한다면 WordRank 는 ’의날씨’를 핵심 단어로 선택한다.

- 문장 : ’오늘의 날씨는 좋습니다’, ’내일의 날씨도 좋습니다’
- WordRank 단어 후보 : 오, 오늘, 오늘의, 늘, 늘의, 늘의날, 의, 의날, 의날씨, ...
- 한국어 단어 후보 : 오, 오늘, 오늘의, 날, 날씨, 날씨는, 좋, 좋습, 좋습니, 내, 내일, 내일의, ...

둘째, 정보성이 적은 1음절 단어 혹은 형태소가 키워드로 추출된다. 조사는 그 종류가 적지만 대부분의 문장에서 등장하는 단어이며 이들은 대부분 길이가 1, 2 음절이다. 어미는 단어는 아니지만 용언의 말미에 반드시 등장하는 형태소이며, 이들의 종류는 다양하지만 실제 문장에는 몇몇 어미들이 매우 빈번히 등장한다. 이들은 문법 기능을 하는 단어 혹은 형태소로 의미를 지녀야하는 키워드로 써는 부적합하다. 하지만 이들은 주요한 단어들의 앞 뒤에 등장하기 때문에 주요한 단어들과 co-occurrence 가 크다.

셋째, 키워드를 포함하는 어절이 중복되어 키워드로 추출된다. 예를 들어 영화 '아저씨'의 리뷰에는 배우 이름 '원빈'이 자주 등장하며, 이 단어가 포함된 '원빈은', '원빈이', '원빈이다'와 같은 어절들도 다수 존재한다. WordRank 가 이용하는 부분글자 그래프는 '원빈'의 랭크가 계산되었을 때, '원빈은'의 랭크 값을 낮출 수 없기 때문에 '원빈'을 포함한 많은 어절들이 중복적으로 추출된다.

TextRank 를 이용한 핵심 문장 추출은 다음의 과정으로 이뤄진다. 첫째, 토크나 이저를 이용하여 문장을 단어열로 분해한다. 이 과정은 단어 그래프를 만드는 과정과 동일하다.

둘째, 각 문장을 그래프의 마디로, 두 문장 간 유사도를 마디 간 가중치로 정의한다. TextRank 는 식 4.2 과 같은 문장 간 유사도를 이용한다.

$$sim(s_1, s_2) = \frac{|\{w_k | w_k \in S_1 \& w_k \in S_2\}|}{log|S_1| + log|S_2|} \quad (4.2)$$

셋째, PageRank [157] 를 이용하여 문장 그래프 내의 각 마디의 랭크를 계산한다. 랭크가 높은 상위 k 개의 문장을 핵심 문장으로 선택한다.

식 4.2 은 두 문장에 공통으로 등장하는 단어의 개수를 두 문장의 길이의 로그값의 합으로 나눈 것으로, 문장의 길이가 길어질수록 분모의 증가분이 줄어들기 때문에 긴 문장에 큰 가중치를 부여하는 경향이 있다. 또한 빈번한 단어로 구성된 문장일수록

분자가 크다. 그러므로 TextRank 는 문서 집합 전체에서 자주 등장하는 단어들로 구성된 긴 문장을 핵심 문장으로 선택한다. 이후 BM25 [169] 를 문장 간 유사도 함수로 이용하거나 [17], 코싸인 유사도를 이용하는 LexRank [60] 가 제안되었다.

하지만 TextRank 를 이용하여 핵심 문장을 추출할 때에는 두 가지 문제가 발생 한다. 첫째, 토크나이저가 단어를 제대로 인식하지 않으면 문장 간 유사도가 정확하게 계산되지 않는다.

둘째, 선택된 핵심 문장 간의 다양성이 보장되지 않는다. TextRank 에서 형태가 매우 유사한 두 문장 중 하나의 문장의 랭크가 높다면 다른 문장도 랭크가 높으며, 상위 k 개의 문장을 선택하는 과정에서 중복된 문장이 선택될 가능성이 있다. 중복적인 핵심 문장은 정보력이 적기 때문에, 핵심 문장은 다양한 내용으로 구성되어야 한다.

이를 해결하기 위하여 핵심 문장 간의 다양성을 유도하는 연구도 제안되었다 [141, 161]. 핵심 문장의 품질을 목적식으로 정의하면 주어진 문장에서 핵심 문장을 선택하는 문제는 정수 최적화 문제로 정의할 수 있다. 목적식에 선택된 핵심 문장들이 중복적 일 경우 그 값이 적어지는 항목을 추가하거나 핵심 문장들이 서로 비슷하지 않도록 제약식을 추가할 수 있다.

4.3 토크나이저를 이용하지 않는 키워드 및 핵심 문장 추출

4.3.1 부분어절 그래프와 그래프 랭킹 알고리즘을 이용한 키워드 추출

KR-WordRank 는 위 세가지 문제를 해결하기 위하여 제안된 그래프 랭킹 기반 한국어 키워드 추출 방법으로, 주어진 데이터로부터 직접 단어를 추출하는 능력이 있기 때문에 미등록단어 문제에서 자유롭다 [98].

KR-WordRank 는 다음과 같은 과정으로 학습된다 (그림 4.3). 첫째, 문장을 어절 단위로 분해하여 빈도수를 계산한다.

둘째, 2.3.1 장에서 제안한 $L + [R]$ 구조를 이용하여 각 어절에서 L 과 R 에 해당하는 부분어절의 빈도를 계산한다. 띄어쓰기 오류가 존재한다 하여도 잘못 추출된 L 과 R 의 빈도수는 상대적으로 작기 때문에 이들이 키워드로 선택될 가능성은 적다.

셋째, L 과 R 별로 빈도수가 같으면, 한 부분어절에 포함되는 짧은 부분어절을 제거한다. 그림 4.3에서 '원/L'과 '원빈/L'이 모두 2 번 등장하였기 때문에 '원/L'은 '원빈/L'을 지칭하는 중복된 마디다.

넷째, 어절 간에는 부분어절 R 과 L 사이의, 어절 내에는 부분어절 L 과 R 사이의 빈도수를 두 마디 간 가중치로 정의한다. 이 과정을 통하여 앞선 예제의 '은하'는 그래프의 마디에서 제외된다.

다섯째, PageRank 를 이용하여 마디의 랭크를 계산한다. 랭크는 단어 가능성 점수 임과 동시에 키워드 점수이다.

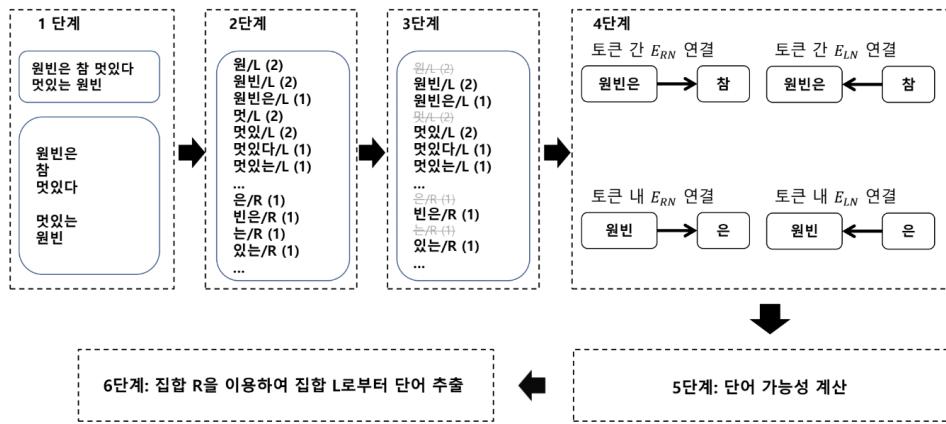


Figure 4.3: KR-WordRank 의 키워드 추출 프레임워크

여섯째, 필터링 과정을 거쳐 $L + R$ 형태의 키워드를 제거한다. 부분어절 R 중 랭킹이 높은 순으로 r_k 개의 부분어절을 선택한다. 이는 이 도메인에서 이용되는 조사 혹은 어미 집합이며, 키워드는 부분어절 L 에서만 선택한다. 부분어절 L 중 랭킹이 높은 순으로 정렬한 뒤, 랭킹이 낮은 l 이 이미 선택된 키워드 w 와 $[r_1, \dots, r_{r_k}]$ 의 조합이 아니면

키워드 집합에 추가한다 (그림 4.4). 위의 예시에서 '원빈'이 키워드로 선택되면 '원빈 + [이], 은, 이다' 는 키워드로 선택되지 않는다.

```
L = [l1, l2, ..., lL] : L set  
R = [r1, r2, ..., rk] : R set  
def select_keywords(L, Rk):  
    KW = : keyword set  
    For c in L:  
        if c is not form of kwi + rj:  
            KW ← KW ∪ c  
    return KW
```

Figure 4.4: KR-WordRank 키워드 필터링 함수 의사 코드

4.3.2 키워드 집합을 이용한 핵심 문장 선택

한 문서 집합을 대표하는 핵심 문장은 그 집합의 키워드를 다수 포함해야 한다. TextRank 는 문서 집합 전체에서 자주 등장하는 단어들로 구성된 문장을 핵심 문장으로 선택하며 길이가 긴 문장을 선호하는 경향이 있기 때문에 키워드가 포함될 가능성이 높다. 그러나 핵심 문장들 간의 다양성은 보장되지 않으며, 토크나이징 결과를 이용하여 문장 간 유사도를 계산한다.

이 장에서는 앞서 제안한 키워드 추출 방법을 이용하여 핵심 문장을 추출하는 방법을 제안한다 (그림 4.5).

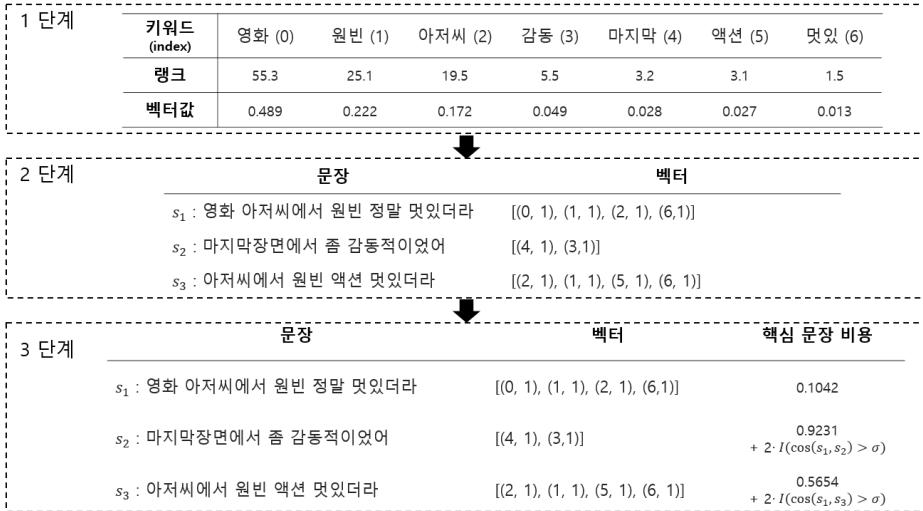


Figure 4.5: KR-WordRank 의 핵심 문장 추출 프레임워크

첫 단계에서 키워드 추출을 통하여 학습한 랭크를 벡터로 변환한다. PageRank 를 이용하여 학습한 랭크 값은 지수 분포를 따르는 경향이 있는데, 순위가 낮은 키워드의 랭크를 상대적으로 높이기 위해서 랭크의 $1/2$ 승을 취할 수도 있다. 이를 키워드 벡터 (KV) 라 명한다.

두번째 단계에서는 2 장에서 제안한 Max Score Tokenizer 를 이용하여 각 문장을 키워드의 포함 유무 벡터로 표현한다. 키워드의 랭크를 단어 점수로 이용하면 문장에서 키워드를 단어로 분리 할 수 있다.

세번째 단계에서는 각 문장 벡터와 키워드 벡터의 유사도를 이용하여 키워드가 많이 포함된 문장을 핵심 문장으로 선택한다 (그림 4.6). 키워드들이 다수 포함된 문장이라면 문서 집합을 대표할 가능성이 높기 때문이다. 이를 위하여 키워드 벡터 KV 와 문장 간의 코싸인 거리를 계산하여 이를 핵심 문장 비용으로 정의한 뒤, 핵심 문장 비용이 가장 적은 문장 S_i 를 첫번째 핵심 문장으로 선정한다. 그 뒤 문장 S_i 와 다른 문장 간의 거리가 σ 보다 작은 문장들에 페널티 비용을 추가한다. 이후 핵심 문장 리스트

KS 의 크기가 k 가 되기 전까지 비용이 가장 작은 문장을 선택하는 과정을 반복한다. 이 과정은 제안한 방법이 키워드를 다수 포함한 문장들 중에서 서로 이질적인 문장을 핵심 문장으로 선택하도록 유도한다.

```

 $KV$  : keyword vector
 $S$  : sentence vectors

def select_keysentences( $S, KV, \sigma, k$ ):
     $\sigma$  : minimum distance between selected keysentences
     $k$  : number of keysentences
     $KS = []$  : keysentence list
     $C = dist(KV, S)$  : initial keysentence score
    While  $|KS| < k$ :
         $i \leftarrow \arg \min_C$ 
         $KS \leftarrow KS + [S_i]$ 
         $C \leftarrow C + I(dist(S, S_i) < \sigma)$ 
    return  $KS$ 

```

Figure 4.6: KR-WordRank 핵심 문장 필터링 함수 의사 코드

4.4 성능 평가

제안하는 방법은 토크나이저를 이용하지 않는 키워드 추출과 이를 이용한 핵심 문장 추출로 이뤄져 있기 때문에 각 과업에 대한 성능을 각각 측정하였다.

제안하는 키워드 추출 방법의 단어 추출 능력을 평가하기 위하여 세종 말뭉치와 영화평 데이터를 이용하였다. 세종 말뭉치는 문장을 구성하는 형태소가 태깅되어 있기 때문에 단어 추출 능력에 대한 정량적인 평가가 가능하다. 제안하는 방법은 2.3.1 장의 $L + [R]$ 구조를 가정하기 때문에 세종 말뭉치의 어절을 $L + [R]$ 구조로 변형하였다. 키워드는 해석이 가능한 독립언이기 때문에 체언과 용언의 표현형, 관형사, 부사, 감탄사를 정답 단어 집합으로 이용하였다. WordRank, 1음절을 제외한 부분단어로 이뤄진 WordRank, 그리고 R_k 의 크기를 300, 400, 500 으로 설정한 KR-WordRank 의 단어 인식 능력을 평가하였다 (그림 4.7). 세종 말뭉치에 등장한 단어의 재현율을 정확도로 이용하였다. 그 결과 WordRank 는 20 % 가 되지 않는 단어 인식 능력을 보이지만,

제안하는 방법은 R 크기의 설정에 따라 70 % 정도의 단어 인식 능력을 보였다.

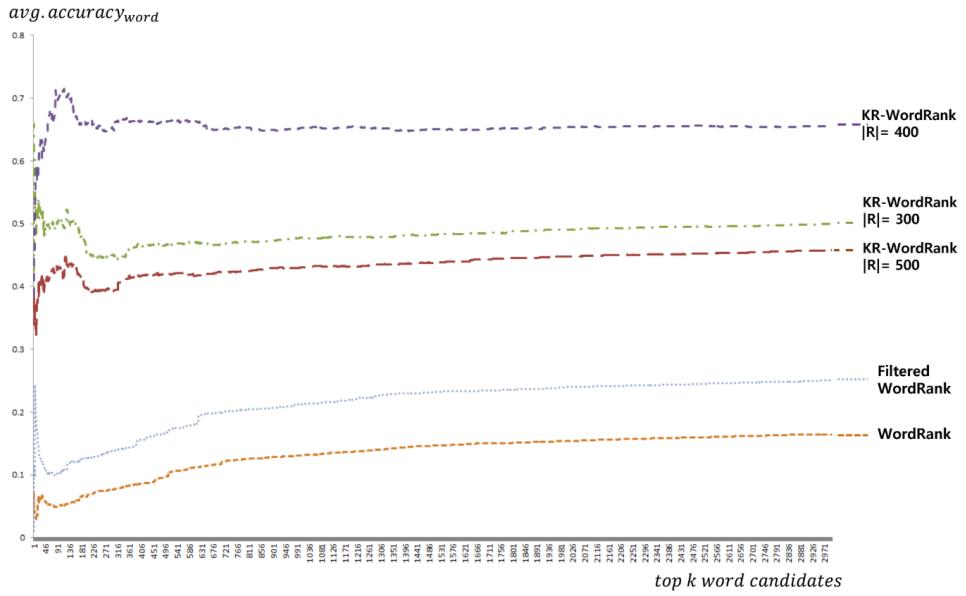


Figure 4.7: WordRank 와 KR-WordRank 를 이용하여 세종 말뭉치로부터 추출한 단어의 정확도

이는 3 장에서 제안한 명사 추출기의 재현율과 비교하면 매우 낮은 수치인데, 비교 실험에 이용한 WordRank 와 KR-WordRank 는 문서 집합의 주제가 단일할 때 좋은 성능을 보이기 때문이다. 그러나 세종 말뭉치는 여러 주제의 문서들이 모여있는 말뭉치이기 때문에 다양한 문서에서 상대적으로 많이 등장하는 단어들만 제대로 인식된다. 그렇기 때문에 영화 ”아저씨”의 영화평에서 WordRank 와 KR-WordRank 를 이용하여 키워드를 출한 결과를 정성적으로 평가하였다 (그림 4.8). 문제 종류 A 는 해석이 불가능하거나 단어가 아닌 경우, B 는 조사나 어미가 결합된 어절, C 는 조사나 어미가 단어로 추출된 경우, D 는 복합명사인 경우이다. 문서 집합의 주제가 단일하더라도 WordRank 는 해석이 불가능한 1 음절을 키워드로 선택하는 경우가 많으며, 이는 대부분 조사나 어미로 추정된다. WordRank 가 추출한 2 음절 이상의 키워드에는 ’원빈의’

와 같이 키워드와 조사가 결합된 경우와 2 음절 이상의 조사나 어미가 키워드로 추출된 경우가 많았다. 하지만 KR-WordRank 는 영화 '아저씨'를 대표하는 키워드들이 잘 추출되었으며, '한국영화'처럼 영화 도메인에서 자주 이용되는 복합명사도 키워드로 추출됨을 확인할 수 있다. 즉 제안하는 방법은 단일한 주제의 문서로 이뤄진 문서 집합을 대표하는 키워드를 추출할 수 있다.

순위	(a) WordRank를 적용한 결과 (길이가 1음절 이상인 경우)		(b) WordRank를 적용한 결과 (길이가 2음절 이상인 경우)		(c) KR-WordRank를 적용한 결과 (길이가 1음절 이상인 경우)	
	단어(빈도수)	문제 종류	단어(빈도수)	문제 종류	단어(빈도수)	문제 종류
1	이(14448)	A	영화(10559)*		영화(10559)*	
2	영화(10559)*		원빈(9248)*		원빈(9243)*	
3	다(14351)	A	액션(3767)*		정말(2783)*	
4	원빈(9248)*		정말(2783)*		액션(3766)*	
5	액션(3767)*		최고(4255)*		최고(4251)*	
6	정말(2783)*		진짜(2105)*		진짜(2105)*	
7	도(5684)	A	대박(2009)*		대박(1988)*	
8	한(7061)	A	연기(2460)*		너무(1859)*	
9	만(6004)	A	너무(1860)*		연기(2460)*	
10	가(5036)	A	아저씨(1281)*		아저씨(1280)*	
11	아(5769)	A	감동(1109)*		완전(1073)*	
12	최고(4255)*		스토리(1022)*		감동(1109)*	
13	고(11221)	A	완전(1074)*		스토리(1022)*	
14	는(8919)	A	원민의(1390)	B	보고(1875)*	
15	에(5386)	A	한국영화(1041)	D	한국영화(944)	D
16	지(6908)	A	원빈이(1078)	B	그냥(789)*	
17	진짜(2105)*		보고(1875)*		평점(1245)*	
18	나(4069)*		하지만(718)*		데이큰(753)*	
19	말(4501)	A	지만(1995)	C	본(1441)*	
20	기(4988)	A	그냥(789)*		굿(578)*	
21	화(11114)	A	네요(1670)	C	줌(557)*	
22	의(4621)	A	하고(960)	C	한국(1867)*	
23	대박(2009)*		평점(1245)*		이런(573)*	
24	연기(2460)*		데이큰(753)*		또(966)*	
25	어(4585)	A	한국(1868)*		배우(685)*	
26	요(6073)	A	이영화(504)	D	내용(566)*	
27	로(3066)	A	최고의영화(519)	D	처음(676)*	
28	너무(1860)*		애서(753)	C	연기력(437)*	
29	대(4137)	A	배우(685)*		이렇게(345)*	
30	아저씨(1281)*		내용(566)*		잔인(1608)*	

Figure 4.8: WordRank 와 KR-WordRank 를 이용하여 추출한 영화 "아저씨" 리뷰의 키워드

제안하는 핵심 문장 추출 방법의 목적은 중복되지 않은 문장으로 핵심 문장을 구성하

는 것이다. 그렇기 때문에 제안하는 방법과 TextRank 에 의하여 선택된 핵심 문장들이 얼마나 다양한 키워드들을 포함하고 있는지 확인하였다. [141, 161] 와 같은 정수 최적화 기반 핵심 문장 추출 방법은 키워드를 정의하지 않기 때문에 비교 실험에서는 이용하지 않았다.

제안하는 방법의 핵심 문장 추출 능력을 평가하기 위하여 영화평 데이터에서 키워드와 핵심 문장을 추출한 뒤, ROUGE-N [128] 를 이용하여 문서 요약 능력을 평가하였다. ROUGE-N 은 핵심 문장에 정답 문장의 n-gram 이 얼마나 재현되었는지를 측정한 수치로, 정답 문장에 포함된 단어가 많이 포함된 문장일수록 좋은 요약문으로 평가한다. 영화평 데이터는 온라인 공간에서 여러 사람들에 의하여 작성된 한줄평 (최대 140 글자) 을 영화별로 모은 뒤, 각 영화별로 하나의 가상 문서를 생성하였다. 네이버 영화로부터 각 영화의 영화평이 5000 개 이상 기록된 152 개의 영화로부터 약 320 만개의 영화평을 수집하였다. 특히 영화평 데이터에는 중복되거나 비슷한 문장이 다수 존재하기 때문에, 다양한 문장으로 핵심 문장을 구성하는 능력을 평가하기에 적합하다.

그러나 각 영화마다 정답 요약 문장을 만드는 과정에는 주관이 개입될 수 있으며, 다른 평가 데이터를 이용할 때마다 정답 데이터를 만들어야 하기 때문에 확장 가능한 방법이 아니다. 이러한 한계점을 해결하기 위하여 키워드 집합을 정답 문장으로 가정한 뒤, 핵심 문장들이 실제로 다양한 키워드를 포함하고 있는지를 확인하였다. 키워드는 하나의 단어로 구성되어 있기 때문에 ROUGE-1 을 이용하였다.

각 영화마다 TextRank 와 코싸인 유사도를 문장 유사도로 이용하는 LexRank, 그리고 KR-WordRank 을 이용하여 키워드와 핵심 문장을 추출하였다. 제안하는 방법의 사용자 정의 패러메터인 σ 는 0.5 로 설정하였으며, TextRank 와 LexRank 는 코모란 토크나이저를 이용하여 형태소 분석을 수행하였다.

그 결과 제안하는 방법은 키워드와 핵심 문장의 개수에 상관없이 언제나 TextRank

나 LexRank 보다 더 높은 ROUGE-1 점수를 보였다 (표 4.1). 또한 키워드 개수를 고정할 경우 핵심 문장의 개수가 늘어남에 따라 ROUGE-1 점수도 함께 증가하는데, 이는 제안하는 방법은 새로운 핵심 문장을 추가할 때마다 이전에 선택된 핵심 문장과 다른 문장들을 추가함을 의미한다. 코싸인 거리를 이용하는 LexRank 는 상대적으로 매우 낮은 성능을 보이는데, 이는 코싸인 척도는 길이가 매우 짧은 문장에 대하여 아주 큰 문장 간 유사도를 부여하여 1, 2 단어로 이뤄진 문장들이 핵심 문장으로 선택되었기 때문이다. 반면 TextRank 는 길이가 긴 문장을 선호하기 때문에 LexRank 보다 더 높은 키워드의 재현율을 보였다.

Table 4.1: KR-WordRank, TextRank, 그리고 LexRank로부터 추출된 핵심 문장의 ROUGE-1 성능

# keywords	# keysents	KR-WordRank	TextRank	LexRank
10	3	0.942	0.752	0.156
10	5	0.983	0.81	0.166
10	10	0.998	0.855	0.197
10	20	1	0.904	0.244
10	30	1	0.92	0.291
20	3	0.758	0.603	0.078
20	5	0.873	0.69	0.084
20	10	0.966	0.767	0.101
20	20	0.994	0.839	0.135
20	30	0.998	0.869	0.17
30	3	0.631	0.496	0.053
30	5	0.771	0.587	0.056
30	10	0.911	0.688	0.069
30	20	0.98	0.778	0.093
30	30	0.994	0.82	0.12
50	3	0.48	0.363	0.032
50	5	0.621	0.452	0.034
50	10	0.804	0.565	0.042
50	20	0.929	0.674	0.058
50	30	0.968	0.731	0.078

아래는 각 방법에 의하여 '라라랜드'의 영화평에서 추출한 핵심 문장의 예시이다 (표 4.2). 제안한 방법과 TextRank 모두 해당 영화의 내용을 잘 대표하는 문장들이

핵심 문장으로 선택되었다. 이들은 표 4.3 의 키워드를 다수 포함하고 있다. LexRank의 경우 다른 방법론보다 짧은 문장이 핵심 문장으로 선택되었음을 확인할 수 있다.

Table 4.2: 영화 '라라랜드' 리뷰로부터 KR-WordRank, TextRank, 그리고 LexRank로부터 추출된 핵심 문장 예시

추출 방법	핵심 문장
KR-WordRank	여운이 크게 남는 영화 엠마스톤 너무 사랑스럽고 라이언고슬링 남자가 봐도 정말 매력적인 배우인듯 영상미 음악 연기 구성 전부 좋았고 마지막 엔딩까지 신선했는데 애탤하구요 30중반에 감정이 많이 메말라 있었는데 오랜만에 가슴이 촉촉해지네요
	영상미도 너무 아름답고 신나는 음악도 좋았다 마지막 세バス찬과 미아의 눈빛교환은 정말 마음 아팠음 영화관에 고딩들이 엄청 많던데 고딩들은 영화 내용 이해를 못하더라— 사랑을 깊게 해본 사람이라면 누구나 느껴볼수있는 며먹함이 있다
	정말 영상미랑 음악은 최고였다 그리고 진선했다 음악이 너무 멋있어서 연기를 봐야 할지 노래를 들어야 할지 모를 정도로 그리고 보고 나서 생각 좀 많아진 영화 정말 이 연말에 보기 좋은 영화 인 것 같다
	무언의 마지막 피아노연주 완전 슬픔ㅠ보는이들에게 꿈을 상기시켜줄듯 또 보고 싶은 내생에 최고의 뮤지컬영화였음 단순할수 있는 내용에 뮤지컬을 가미시켜제즈음악과 춤으로 지루할틈없이 빠져서봄 ost너무좋았음
	처음엔 초딩들 보는 그냥 그런영화인줄 알았는데 정말로 눈과 귀가 즐거운 영화였습니다 어찌보면 뻔한 스토리일지 몰라도 그냥 보고 듣는게 즐거운
	그러다가 정말 마지막엔 너무 아름답고 슬픈 음악이 되어버린 시사회 보고 왔어요 꿈과 사랑에 관한 이야기인데 뭔가 진한 여운이 남는 영화예요
TextRank	시사회 갔다왔어요 제가 라이언고슬링팬이라서 하는말이아니고 너무 재밌어요 꿈과 현실이 잘 보여지는영화 사랑스런 영화 전 개봉하면 또 볼생각입니다 횡홀하고 따뜻한 꿈이었어요 imax로 또 보려합니다 좋은 영화 시사해주셔서 감사해요
	시사회에서 보고 있는데 여운转折었다 엠마스톤과 라이언 고슬링의 캐미가 도입부의 강렬한음악좋았고 예고편에 나왔던 오디션 노래 감동적이어서 눈물나왔다ㅠ
	이영화는 위플래쉬처럼 꼭 영화관에봐야한 색감 노래 배우 환상적인 영화
	방금 시사회로 봤는데 인생영화 하나 또 탄생했네 롱테이크 촬영이 예술 영상이 넘나 아름답고 라이언고슬링의 멋진 피아노 연주 엠마스톤과의 춤과 노래 눈과 귀가 호강한다
LexRank	재미를 기대하면 약간 실망할수도 있지만 충분히 훌륭한 영화
	좋다 좋다 정말 너무 좋다 그 말 밖엔 인생영화 등극 ㅠㅠ
	음악도 좋고 다 좋고 좋고 좋고 다 좋고 씁쓸한 결말 뭔가 아쉽다

표 4.3 는 제안하는 방법과 TextRank 를 이용하여 추출한 키워드의 예시이다. LexRank 는 TextRank 의 문장 간 유사도를 변형한 방법이기 때문에 TextRank 의 키워드를 이용한다. 제안하는 방법은 토큰나이저를 이용하지 않으며 부분어절을 키워드로 선택하기 때문에 용언의 표현형의 일부를 단어로 선택한다. 그 결과 '재미'나 '재밌'처럼 원형은 같지만 표현형이 다른 두 부분단어가 모두 키워드로 선택되는 단점이 있다. 하지만 형태소 분석 과정을 거치지 않았음에도 불구하고 제안하는 방법은 명사, 동사, 형용사, 부사와 같이 의미를 지니는 단어들을 키워드로 추출하였다.

Table 4.3: 영화 '라라랜드' 리뷰로부터 KR-WordRank, TextRank 을 이용하여 추출한 키워드 예시

KR-WordRank		TextRank	
영화 (201.34)	현실 (15.21)	영화/NNG (173.00)	번/NNB (20.26)
너무 (81.80)	생각 (14.94)	보/VV (128.93)	거/NNB (19.67)
정말 (40.62)	지루 (13.81)	좋/VA (65.55)	최고/NNG (19.18)
음악 (40.52)	다시 (13.62)	하/VV (52.02)	때/NNG (19.15)
마지막 (38.73)	감동 (13.61)	것/NNB (47.43)	사람/NNG (19.04)
뮤지컬 (23.24)	보는 (12.49)	같/VA (45.37)	여운/NNP (17.55)
최고 (21.85)	좋아 (12.01)	영화/NNP (43.89)	뮤지컬/NNP (16.94)
사랑 (20.69)	재밌 (11.91)	음악/NNG (43.59)	나오/VV (16.54)
꿈을 (20.47)	재미 (11.41)	꿈/NNG (41.43)	듯/NNB (16.11)
아름 (20.36)	좋고 (11.39)	있/VV (40.79)	영상미/NNG (15.95)
영상 (20.33)	계속 (11.16)	없/VA (35.94)	지루/XR (15.66)
여운이 (19.51)	조금 (10.95)	마지막/NNG (31.92)	처음/NNG (15.25)
진짜 (19.10)	느낌 (10.94)	수/NNB (30.08)	장면/NNG (15.15)
노래 (18.77)	처음 (10.76)	사랑/NNG (28.25)	감동/NNG (15.14)
보고 (18.60)	결말 (10.60)	아름답/VA (26.47)	가/VV (15.03)
좋았 (17.66)	연기 (10.52)	현실/NNG (24.82)	만들/VV (13.50)
그냥 (16.60)	장면 (10.38)	되/VV (23.91)	들/VV (13.24)
스토리 (16.27)	그리고 (10.36)	노래/NNG (23.39)	남/VV (13.21)
좋은 (15.67)	하는 (10.28)	생각/NNG (23.19)	느낌/NNG (13.13)
인생 (15.41)	있는 (10.17)	스토리/NNP (21.35)	말/NNG (13.13)

4.5 결론

이 장에서는 한국어의 문서 요약 과정에서 발생할 수 있는 미등록단어 문제 해결을 해결하고 요약 문장 내 다양성을 유도하는 추출 기반 문서 요약 방법을 제안하였다. 단일한 주제로 구성된 문서 집합에서의 성능을 평가하기 위하여 온라인에서 수집된 영화평 데이터를 이용하여 정성적으로 성능을 평가하였다. 그 결과 토크나이저를 이용하지 않았음에도 각 영화를 대표하는 단어를 키워드로 추출할 수 있음을 확인하였다. 또한 제안된 핵심 문장 추출 방법은 TextRank 보다 다양한 키워드를 포함하는 문장들로 핵심 문장을 구성하였다.

그러나 제안된 방법은 다음의 한계점이 있다. 앞서 언급한 것처럼 제안한 방법은

음절 단위로 분리된 부분 단어 중에서 키워드를 선택하기 때문에 용언의 표현형이 다른 부분어절을 중복하여 키워드로 선택할 가능성 있다. 또한 핵심 문장에 띄어쓰기나 문법 오류가 포함되어 있다 하더라도 이를 교정하지 않기 때문에 비문들로 문서 집합이 요약될 수 있다. 문서 요약 과업은 그 결과를 사용자가 직접 해석하는 경우가 많기 때문에 이러한 오류를 교정하는 후처리 과정이 추가된다면 사용자가 느끼는 요약의 품질을 향상시킬 수 있다.

더하여 제안된 방법을 적용하기 전, 주어진 문서 집합을 구성하는 주제가 단일한지 확인하는 과정이 필요하다. 한 영화에 대한 영화평들로 이뤄진 문서 집합의 경우에는 이를 구성하는 주제가 비슷할 것이라 예상할 수 있지만, 이를 가정하기 어려운 상황들이 많다. 이 장에서는 주어진 문서 집합이 단일한 주제로 구성되었다고 가정한 뒤 적용할 수 있는 방법을 제안하였지만, 주어진 문서 집합이 단일 주제로 구성되었는지를 판단하는 방법이 우선적으로 적용되어야 한다.

제 5 장 다주제 문서 집합 요약을 위한 문서 군집화 알고리즘 및 군집 별 키워드 추출

5.1 개요

문서 집합을 구성하는 주제가 단일하다면 문서 요약을 위하여 4 장의 방법이 이용될 수 있다. 하지만 문서 집합에 여러 종류의 주제가 섞여있을 때에는 그래프 랭킹 기반 방법은 정보력이 적은 혼한 단어를 키워드로 선택한다 [70, 129, 65, 64].

문서 집합이 여러 주제의 문서들로 구성되어 있을 때에는 문서 집합을 주제 별로 나눈 뒤, 각 주제 별로 문서를 요약해야 한다. 이를 위하여 Latent Dirichlet Allocation [23] 와 같은 토픽 모델링을 통하여 주제를 학습 한 뒤 토픽 레이블링 기법 [190] 을 이용하여 주제 별 키워드를 탐색하거나, 문서 군집화를 이용하여 문서 집합을 단일한 주제의 부분 집합으로 구분한 뒤, 각 군집별로 문서를 요약할 수 있다 [207, 59, 203].

그러나 샘플링 기법을 기반으로 학습하는 Latent Dirichlet Allocation 는 문서 집합의 크기가 클 경우 많은 계산량이 필요하며 [226], 불용어제거와 같은 전처리 과정이 잘 이뤄지지 않으면 학습이 제대로 되지 않거나 불필요한 토픽과 단어들이 주요한 토픽과 단어로 해석되기도 한다 [50, 154]. Latent Dirichlet Allocation 는 하나의 문서에 여러 개의 주제가 존재한다는 가정하지만 뉴스나 블로그와 같이 길지 않은 문서는 사실 하나의 주제를 가지는 경우가 존재한다. 이와 같은 경우에는 하나의 문서에 하나의 토픽이 존재한다고 가정하는 문서 군집화 기법이 이용할 수 있다 [54, 220, 215].

5 장에서는 문서가 하나의 주제로 구성되어 있다고 가정할 때, 문서 군집화 기법을 이용하여 여러 주제로 구성된 문서 집합을 요약하기 위한 방법을 제안한다. 그러나 대량의 문서에 대한 k-means 군집화 기법의 적용 과정에는 여전히 효율성을 개선할

부분이 존재하며, 각 군집마다 키워드를 추출하기 위해서는 추가적인 모델 학습이 필요하다. 5 장에서는 효율적인 문서 군집화를 위하여 k-means 를 개선하여, 군집화 결과를 이용하여 추가적인 모델 학습 없이 군집 별 키워드를 추출하는 방법을 제안한다.

5.2 관련 연구

문서 군집화는 각 문서가 속한 집합에 대한 레이블이 없는 상황에서, 문서 간 유사성을 바탕으로 비슷한 문서를 묶는 비지도학습 과업이다 [218, 221, 214]. 이를 위해서 다양한 군집화 알고리즘이 이용될 수 있다. DBSCAN [61]과 같은 밀도 기반 군집화 알고리즘, 계층 구조 기반 군집화 알고리즘 [189], 혹은 그래프 기반 군집화 알고리즘 [44] 는 다양한 군집화 문제에 이용되었다. 하지만 이들의 계산 비용은 $O(n^2)$ 이상이기 때문에 대량의 군집화에 적합하지 않다.

반면 k-means 는 군집화 방법은 $O(n)$ 의 계산 비용으로 학습할 수 있기 때문에 대량의 문서 군집화에 효과적이다 [46, 78, 89, 45]. k-means 는 k-partition 문제의 해법을 계산하는 근사 알고리즘으로, 데이터를 k 개의 부분 집합으로 나누었을 때 각 부분 집합 내의 점들 간의 분산이 최소화 되는 해를 탐색한다 5.1.

$$\operatorname{argmin}_C \sum_{i=1}^k \sum_{x \in C_i} |x - c_i|^2 = \operatorname{argmin}_C \sum_{i=1}^k |C_i| \operatorname{Var}(C_i) \quad (5.1)$$

C : cluster index

C_i : cluster i

c_i : the centroid of cluster i

x : a vector of a data point

k : a number of clusters

[134] 에서 제안된 k-means 는 빠른 계산 속도와 적은 메모리 사용, 분산 계산이 쉬

운 구조적 특징 때문에 대량의 데이터 군집화에 널리 이용되고 있다. k-means 는 그림 5.1 의 과정으로 학습된다. 초기 군집 중심값 (initial centroids) 를 선정한 뒤, 데이터의 모든 점을 가장 가까운 군집 중심값의 레이블로 할당한다. 각 군집에 속한 데이터의 평균을 새로운 군집 중심값으로 설정한 뒤, 위 과정을 모델이 수렴할 때까지 반복한다.

```

D: dataset
k: a number of clusters

def kmeans (D, k):
    C  $\leftarrow$  Initialize k centroids with random sampling
    L  $\leftarrow$  Assign all points to its closest centroid
    while (not converged):
        C  $\leftarrow$  Update centroids by averaging assigned date points
        L  $\leftarrow$  Reassign all the points to its closest centroid
    return C, L

```

Figure 5.1: Lloyd k-means 의사 코드

문서 군집화에는 문서 간 거리 척도의 설정이 중요하다. 일반적으로 문서는 Bag-of-Words Model 과 같은 고차원 sparse 벡터나 Doc2Vec 과 같은 고차원의 분산 표상 표현 (distributed representation) 으로 표현된다 [112, 49]. Bag-of-Words Model 로 표현되는 문서 간의 유사도는 두 문서에 공통으로 등장한 단어 정보가 중요하지만, 유 클리디언 거리는 이를 제대로 표현하지 못하기 때문에 코싸인 거리 혹은 자카드 거리와 같은 척도를 이용해야 한다 [85]. 또한 문서가 분산 표상 표현으로 표현된다 하더라도 임베딩 공간에서 두 객체의 유사도는 코싸인 거리를 이용하는 것이 좋다 [122]. 유클리디언 거리 대신 코싸인 거리를 이용하는 k-means 를 Spherical k-means 라 하며 문서 군집화 과업에 자주 이용된다 [53, 30].

그러나 k-means 는 여전히 개선될 여지가 있다. k-means 는 초기의 군집 중심값이 잘 설정되어야 안정적인 학습 결과를 얻을 수 있다 [9]. 그림 5.2 (a) 처럼 데이터의 전체 영역에 펼쳐진 형태로 초기 군집 중심값이 선택되면 빠르고 안정적인 수렴이 가능하지만, 5.2 (b) 처럼 한 지역에서 초기 군집 중심값이 선택되면 불안정한 학습 결과를

보인다.

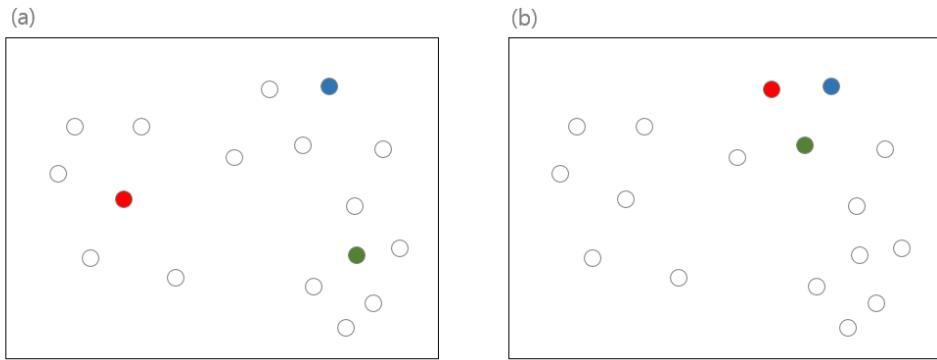


Figure 5.2: (a) 기대하는 초기 군집 중심값, (b) 잘못된 초기 군집 중심값

이러한 초기화 문제를 해결하기 위하여 다양한 k-means 초기화 방법이 제안되었다. Global k-means 는 순차적으로 초기 군집 중심값을 탐색하는 방법이다 [127, 11]. Global k-means 는 데이터의 평균 벡터를 첫번째 군집의 초기값으로 이용한다. 그 뒤 임의로 새로운 한 점을 선택하여 $k = 2$ 인 k-means 를 학습하여 두 개의 군집 중심값을 얻는다. 계속하여 새로운 점을 추가하여 $k = 3$ 인 k-means 를 학습한다. 이 과정을 군집의 개수가 k 가 될때까지 반복한다. 그러나 이 과정은 $O(nk^2)$ 의 계산 비용을 요구하기 때문에 군집의 개수가 클 경우에는 매우 큰 학습 비용을 야기한다.

k-means++ [9] 은 이러한 문제를 해결한 k-means 초기화 알고리즘으로, 그림 5.3 처럼 작동한다. 처음 한 점을 임의로 선택한 뒤, 다른 모든 점들 과의 거리 $d(x)$ 를 계산한다. 거리의 제곱을 정규화한 확률 분포를 기반으로 다른 한 점을 임으로 선택한다. 그 뒤 선택된 점들과 데이터의 모든 점들 간의 거리를 계산한 뒤, 선택된 점들 기준으로 가장 작은 거리값을 $d(x)$ 로 이용한다. 이 과정을 k 개의 점이 선택되기 전까지 반복한다. 이 방법은 이미 군집 초기값으로 선택된 점과 그 주변의 점들이 선택될 확률을 작게 만들도록써 데이터 전체에 펼쳐진 점들을 군집 초기값으로 선택한다. 그러나 이 방법 역시 여전히 $O(nk)$ 의 계산 비용이 요구된다.

- | |
|--|
| 1. Select point c_1 randomly |
| 2. Select next point c_t with probability $\frac{d(x)^2}{\sum_{x \in D} d(x)^2}$ |
| 3. Repeat step 2 until k points are chosen as initial points. |

Figure 5.3: k-means++ 의 의사 코드

k-means++ 은 서로 떨어진 점들을 초기 군집 중심값으로 선택할 수는 있지만, 데이터의 차원이 클 경우 효율적이지는 않다. 표 5.1 는 실험에 이용한 7 종류의 고차원 데이터에서의 점들 간 코싸인 거리 분포이다. 고차원 공간에서는 대부분의 문서 간 코싸인 거리가 1에 가깝기 때문에 그림 [9]에서 계산된 확률 분포는 균등 분포에 가깝다.

Table 5.1: 고차원 벡터로 표현된 7 개의 텍스트 데이터에서의 코싸인 거리 분포. D1: A6 블로그, D2: 투스칸 블로그, D3: 소나타 블로그, D4: IMDb 리뷰, D5: Reuters RCV1, D6: MovieLens 20M, D7: Yelp 리뷰 (%)

Distance range	Dataset						
	D1	D2	D3	D4	D5	D6	D7
<= 0.7	0.249	0.59	0.323	0.272	0.045	1.456	0.01
0.7 - 0.8	0.378	1.121	0.455	7.751	0.067	2.386	0.286
0.8 - 0.9	1.628	3.89	1.984	57.271	0.316	12.458	11.073
0.9 - 1.0	97.745	94.399	97.239	34.706	99.573	83.7	88.632

5.3 문서 군집화를 위하여 개선된 Spherical k-means

문서 군집화와 문서 요약을 위하여 개선된 Spherical k-means 알고리즘은 두 가지 제안하는 방법이 추가되었다. 첫째, 코싸인 거리 척도를 이용하는 고차원 벡터 공간에서 효율적으로 작동하는 초기화 방법을 이용한다. 둘째, k-means 의 학습 결과인 군집 중심값을 이용하여 군집 별 키워드를 추출한다.

그림 5.4 은 제안하는 문서 군집화 알고리즘의 의사 코드이다. 제안하는 방법은 대량의 문서 군집화를 위한 빠른 초기화 알고리즘과 군집 대표 벡터를 이용한 군집 레이블링

기능이 포함되어 있다. 사용자는 α 와 t_{init} 를 통하여 초기화에 이용할 점들의 개수와 초기화된 점들 간의 거리를 조절하며 k_0 와 k_1 을 이용하여 군집 별 키워드 및 후보 단어의 개수를 조절한다.

```

 $D$ : dataset
 $k$ : a number of clusters
 $t_{init}$ : minimum distance for initializer
 $k_0$ : number of keyword candidates
 $k_1$ : number of selected keywords

def improved_kmeans ( $D, k, \alpha, t_{init}, k_0, k_1$ ):
     $C \leftarrow$  Initialize  $k$  centroids with sparse initializer ( $D, k, \alpha, t_{init}$ )
     $L \leftarrow$  Assign all points to its closest centroid
    while (not converged):
         $C \leftarrow$  Update centroid. Averaging their belonging points
         $L \leftarrow$  Reassign all the points to its closest centroid
     $Z \leftarrow$  cluster labeling ( $C, L, k_0, k_1$ )
    return  $C, L, Z$ 
```

Figure 5.4: 개선된 Spherical k-means 의 의사 코드

5.3.1 효율적인 Spherical k-means 초기화

k-means 의 초기 군집 중심값이 그림 5.2 (b) 처럼 선택되면 불안정한 군집화 결과가 학습될 수 있다. 이를 개선하기 위하여 k-means++ 가 제안되었지만, $O(nk)$ 의 초기화 비용을 요구하며 고차원 벡터 공간에서는 효율적으로 작동하지 않는다. 고차원 벡터에서 서로 떨어진 초기 군집 중심값을 효율적으로 선택하기 위하여 그림 5.5 의 초기화 방법을 제안한다.

1. Construct set D_{init} , $\alpha \times k$ randomly selected data from data D
2. Select c_i randomly from D_{init}
3. Remove x_j where $x_j \in D_{init}$ and $\cos(ac_i, x_j) \geq t_{init}$
4. Repeat 2 ~ 3 until k centroid points are selected or D_{init} becomes empty
5. If the number of selected centroid is less than k , then select remaining points from $D - D_{init}$ randomly

Figure 5.5: 개선된 Spherical k-means 초기화 방법

제안하는 방법은 입력 데이터 D 에서 군집의 개수 k 의 α 배수의 점을 임의로 선택하여 D_{init} 을 만든다. α 는 사용자에 의하여 설정되는 패러메터이며, 실험을 통하여 1.5에서 10의 값이 적절함을 확인하였다. 제안하는 초기화 방법은 D_{init} 에서 임의로 한 점을 선택한 뒤 D_{init} 의 다른 점들과 코싸인 거리가 사용자 입력 패러매터인 t_{init} 보다 작은 점들을 D_{init} 에서 모두 제거한다. 그 뒤 다른 한 점을 D_{init} 에서 임의로 선택한 뒤, 이를 k 개의 점이 선택되거나 D_{init} 이 공집합이 될 때 까지 반복한다. 만약 k 개의 점을 선택하지 못한 경우에는 D_{init} 에 속하지 않은 임의의 점을 선택하여 k 개의 초기 군집 중심값을 선택한다.

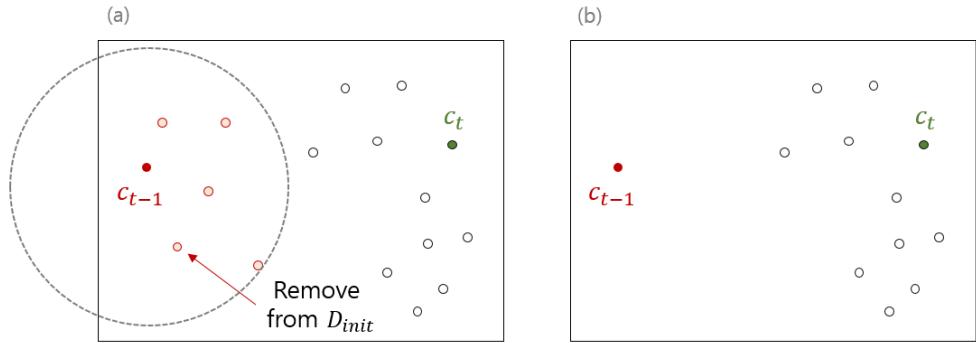


Figure 5.6: 개선된 Spherical k-means 초기화 방법의 예시 (a) 한 점 c_{t-1} 이 선택된 뒤 코싸인 거리가 t_{init} 보다 작은 점들을 제거한 예시 (b) 제거된 점에서 임의로 선택된 c_t

표 5.1에서 알 수 있듯이 대부분의 문서 간 코싸인 거리가 1에 가깝기 때문에 D_{init} 에서 임의의 점을 선택하여도 선택된 점들 간의 거리 역시 1에 가까우며, 선택된 점들과 거리가 t_{init} 보다 작은 점들을 제거하였기 때문에 한 지역에서 여러 개의 군집 초기값이 선택될 가능성은 매우 작다 (그림 5.6). 하지만 최대 $(\alpha \times k)^2$ 번의 거리 계산만 필요하기 때문에 문서 집합의 크기가 클 경우 적은 거리 계산만으로 군집 초기값을 선택할 수 있다 ($\alpha \times k \ll n$).

5.3.2 군집 중심값을 이용한 문서 군집 별 키워드 추출 방법

문서 군집화 과정은 각 문서가 속한 클래스의 정보가 존재하지 않는 상황에서 이용되기 때문에 군집 별 키워드에 대한 정답 데이터도 존재하지 않는다. 각 군집에 4 장에서 제안된 방법이나 TextRank 와 비지도학습 문서 요약 방법을 적용할 수도 있으나, 이는 추가적인 모델 학습을 요구한다. 문서 군집화와 비슷한 분야인 토픽 모델링 분야에서도 비슷한 문제를 해결하기 위하여 비지도학습 토픽 레이블링 기법이 활발히 연구되었다 [154, 190, 42, 192, 41, 23].

문서 집합의 주제가 단일할 경우에는 빈도수가 높은 단어가 문서 집합을 대표하는 키워드일 가능성이 높다. 하지만 모든 군집에서 빈도수가 높은 단어는 정보를 지니지 않은 단어일 가능성이 높기 때문에 각 군집을 구분할 수 있는 단어를 군집의 키워드로 선택하는 것이 좋다. 그러나 군집 간 구분력이 좋은 단어는 빈도수가 작아 각 군집을 대표하지 못할 수 있다. 제안하는 군집 별 키워드 추출 방법은 군집 간 구분력과 군집의 대표성을 모두 고려하여 키워드를 추출한다.

식 5.2 은 제안하는 군집 별 키워드 추출 방법으로, $s(w, C_i)$ 은 군집 C_i 에서의 단어 w 의 군집 구분력 점수이다. $p_i(w)$ 는 군집 C_i 에서의 단어 w 의 등장 비율이며, $p_{-i}(w)$ 는 군집 C_i 을 제외한 다른 문서 집합에서의 단어 w 의 등장 비율이다. $p_i(w)$ 와 $p_{-i}(w)$ 가 비슷하면 단어 w 는 군집과 상관성이 적은 단어임을 의미하며, $p_i(w)$ 가 $p_{-i}(w)$ 보다 클 경우에는 군집 C_i 에 유독 등장한 단어임을 의미한다. $s(w, C_i)$ 의 범위는 $[0, 1]$ 로 1 에 가까울수록 군집 C_i 구분하는 단어임을 의미한다.

$$s(w, C_i) = \frac{p_i(w)}{p_i(w) + p_{-i}(w)} \quad (5.2)$$

그러나 빈도수가 매우 작은 단어는 한 군집에서만 등장하여 $s(w, C_i)$ 의 점수가 1 일 가능성이 높다. 이를 방지하기 위하여 각 군집별로 $p_i(w)$ 기준 상위 k_0 개의 단어

를 키워드 후보로 선택한 뒤, $s(w, C_i)$ 기준 상위 k_1 개의 단어를 키워드로 선택한다. 선택된 키워드는 한 군집에서 자주 등장하는 단어임과 동시에 C_i 에서 유독 등장하는 단어이다. 각 문서가 Doc2Vec 과 같은 분산 표상 벡터로 표현된 경우에는 문서의 단어 빈도 벡터를 추가로 계산해야 하지만, 문서가 Bag-of-Words Model로 표현된 경우에는 군집 중심값이 $p_i(w)$ 로 이용될 수 있다. 이 때는 다른 군집의 크기를 가중치로 이용하여 다른 군집 중심 중심값의 가중 평균을 $p_{-i}(w)$ 로 이용할 수 있다. 이 방법은 [229, 156]처럼 정답 데이터를 필요로 하거나 새로운 모델을 학습하지 않고도 빠르게 키워드를 추출한다.

5.4 성능 평가

제안하는 문서 군집화 방법은 코싸인 척도를 이용하는 고차원 공간에서 효율적으로 작동하는 초기화 방법과 문서 군집화 결과를 이용하여 키워드를 추출하는 두 가지 방법으로 구성되어 있다. 제안된 k-means 초기화 방법의 성능을 확인하기 위해서는 초기화 과정에서의 계산 속도 감소량과 초기화 방법에 의한 군집화 품질의 변화를 정량적으로 확인하였다. 하지만 문서 군집화 결과를 이용한 키워드 추출 과업의 경우 객관적인 정답 데이터를 확보하기 어렵기 때문에 예시를 통한 정성 평가를 수행하였다.

제안된 방법의 성능을 확인하기 위하여 7 종류의 문서 집합 데이터를 실험에 이용하였다 (표 5.2). A6, 투스칸, 소나타 블로그는 네이버 블로그에서 수집된 문서 집합으로, 각각의 질의어가 포함된 블로그들로 구성되어 있다. IMDb 리뷰 데이터는 2,514 개의 리뷰로 이뤄진 데이터로, IMDb로부터 수집되었다. RCV1, MovieLens 20M, Yelp 리뷰는 공개된 데이터이다. RCV1은 단어 문서 행렬 형태로 배포된다. MovieLens 20M은 '사용자 - 영화' 시청 내역 데이터이지만, 희소 행렬로 표현된 형태가 문서의 단어 빈도 행렬과 비슷하다. Yelp 리뷰 데이터는 사용자에 의하여 작성된 식당 리뷰와 평점

데이터로, 실험에서는 리뷰 텍스트만 이용하였다.

Table 5.2: 실험에 이용한 7 종류의 데이터셋

데이터	문서 개수	단어 개수	단어 빈도 행렬의 0이 아닌 값의 개수	단어 빈도 행렬의 0이 아닌 값의 비율
A6 블로그	63,153	91,302	18,051,341	0.313 %
투스칸 블로그	105,755	81,497	29,192,999	0.339 %
소나타 블로그	229,253	85,129	60,861,803	0.312 \$
IMDb 리뷰	1,228,348	68,049	181,411,713	0.217 %
Reuter RCV1	804,414	47,236	60,915,113	0.160 %
MovieLens 20M	138,493	131,262	20,000,263	0.110 %
Yelp 리뷰	5,261,669	27,247	365,341,887	0.255 %

5.4.1 초기화 방법의 성능 평가

제안된 초기화 방법의 성능을 확인하기 위하여 초기화 계산 속도와 문서 군집화 품질을 확인하였다.

표 5.3 는 제안된 초기화 방법과 k-means++ 의 초기화 계산 시간이다. 제안된 방법은 α 가 커짐에 따라 계산 시간이 비례하여 커지는 경향이 있으나, 모든 데이터에서 k-means++ 보다 훨씬 빠른 초기화 속도를 보여준다. 제안된 방법은 D_{init} 에서만 문서 간 거리를 계산하므로 $n / (\alpha \times k)$ 에 비례하여 효율성이 증가한다. 즉 k 가 일정할 경우 문서의 크기가 클수록 더욱 효율적인 초기화가 가능하다.

Table 5.3: 제안된 초기화 방법과 k-means++ 를 이용한 초기화 시간 (초)

Data	α	k			
		10	20	50	100
A6 blogs	1	x 288	x 268	x 260	x 233
	3	x 265	x 257	x 213	x 150
	5	x 248	x 226	x 166	x 99
	10	x 217	x 159	x 100	x 54
	k-means++	5 s	10 s	25 s	52 s
Tucson blogs	1	x 464	x 487	x 397	x 367
	3	x 388	x 440	x 306	x 244
	5	x 358	x 376	x 261	x 172
	10	x 312	x 279	x 160	x 102
	k-means++	7 s	16 s	41 s	82 s
Sonata blogs	1	x 777	x 941	x 860	x 777
	3	x 785	x 841	x 614	x 495
	5	x 707	x 770	x 534	x 376
	10	x 600	x 615	x 330	x 208
	k-means++	16 s	33 s	86 s	175 s
IMDb review	1	x 1165	x 1257	x 2137	x 2253
	3	x 803	x 714	x 1988	x 1787
	5	x 1180	x 1172	x 1715	x 1381
	10	x 815	x 1062	x 1301	x 866
	k-means++	41 s	84 s	214 s	431 s
Reuters RCV1	1	x 511	x 686	x 819	x 892
	3	x 439	x 713	x 850	x 772
	5	x 520	x 685	x 672	x 678
	10	x 518	x 639	x 622	x 425
	k-means++	13 s	28 s	71 s	146 s
MovieLens 20M	1	x 193	x 215	x 218	x 210
	3	x 202	x 213	x 214	x 184
	5	x 202	x 204	x 186	x 145
	10	x 189	x 172	x 144	x 103
	k-means++	4 s	9 s	24 s	49 s
Yelp reviews	1	x 484	x 876	x 1535	x 3092
	3	x 368	x 908	x 1508	x 2917
	5	x 362	x 903	x 1877	x 1595
	10	x 351	x 598	x 1143	x 2120
	k-means++	81 s	164 s	421 s	848 s

제안된 초기화 방법이 군집화 품질에 미치는 영향력을 확인하기 위하여 '군집 내 평균 거리', '군집 간 평균 거리', 'Silhouette 점수'를 이용하여 군집화 품질을 측정하였다. Silhouette 점수는 군집화 품질을 측정하는 척도 중 하나로 [170, 123], 식 5.3 으로 정의된다. 군집 내 평균 거리와 가장 가까운 다른 군집 간 거리 차가 클수록 더 높은 점수가 계산된다. '군집 내 평균 거리'는 감소할수록 '군집 간 평균 거리'는 증가할수록, 'Silhouette 점수'도 증가할수록 군집화 품질이 향상됨을 의미한다.

$$s(x) = \frac{b(x) - a(x)}{\max(a(x), b(x))} \quad (5.3)$$

$a(x)$: 점 x 가 속한 군집 내 평균 거리

$b(x)$: 점 x 와 가장 가까운 다른 군집의 점들 간 평균 거리

표 5.4 는 각 척도별로 제안된 방법의 군집화 품질의 평균을 계산한 뒤, k-means++를 이용한 군집화 품질로 나눈 값이다. k 다를 경우 각 군집화 품질 값의 경향이 달라지므로 같은 k 를 기준으로 군집화 품질의 배율을 계산하였다. 그 결과 군집화 품질 기준과 데이터셋에 관계없이 대부분 1 에 가까운 값임을 확인하였고, 이는 제안하는 방법에 의한 군집화 품질의 손실은 없음을 의미한다. 제안한 방법은 고차원 데이터에서 대부분 점들 간의 코싸인 거리가 1 에 가깝다는 특징을 이용하여 효율적으로 초기 군집 중심값을 선택하였다. 제안한 초기화 방법에 의하여 선택된 점들 역시 충분히 떨어진 점들이기 때문에 군집화 품질은 여전히 안정적이다.

Table 5.4: 제안된 초기화 방법과 k-means++ 를 이용한 평균 군집화 품질의 배율

Data	Intra-cluster distance	Inter-cluster distance	Silhouette score
A6 blogs	0.986	1.001	1.124
Tucson blogs	1.006	1.000	0.998
Sonata blogs	1.005	1.000	1.038
IMDb reviews	0.999	1.000	0.984
Reuters RCV1	0.999	1.001	1.019
MovieLens 20M	1.002	0.996	1.102
Yelp reviews	1.001	0.991	0.980

5.4.2 문서 군집 별 키워드 추출 방법의 성능 평가

앞선 초기화 방법의 성능을 평가하기 위해서 모든 데이터에 동일한 k 를 이용하여 문서 군집화를 수행하였다. 하지만 IMDb 리뷰의 의미있는 군집화 결과를 학습하기

위해서 군집의 개수를 1,000 으로 설정하였다.

표 5.5 은 IMDb 의 군집화 결과에 제안된 군집 별 키워드 추출 방법이 적용된 5 개 군집의 예시이다. 각 군집을 구분할 수 있는 단어들이 키워드로 선택되었기 때문에 군집의 의미를 쉽게 해석할 수 있다. 표 5.5 와 5.6 의 두번째 열은 군집 별 키워드이며 첫번째 열은 키워드를 통하여 해석한 군집 별 레이블이다. 표 5.5 의 첫 군집은 영화 "Titanic" 과 관련된 문서들이, 다섯 번째 군집은 영화 "Matrix" 와 관련된 문서들이 하나의 군집을 이뤘음을 알 수 있다. 하지만 다른 세 개의 군집은 비슷한 주제의 영화들이 각각 하나의 군집을 이뤘음을 해석할 수 있다. 예시에서 살펴볼 수 있듯이 제안한 군집 별 키워드 추출 방법은 각 군집의 의미를 해석할 수 있는 키워드를 제공한다.

Table 5.5: IMDb 리뷰의 $k=1,000$ 문서 군집화 결과

군집 의미	군집 별 키워드
"Titanic"	iceberg, zane, sinking, titanic, rose, winslet, camerons, 1997, leonardo, leo, ship, cameron, dicaprio, kate, tragedy, jack, disaster, james, romance, love, effects, story
Marvel comics	zemo, chadwick, boseman, bucky, panther, holland, cap, infinity, mcu, russo, civil, bvs, antman, winter, ultron, airport, avengers, marvel, captain, superheroes, stark, evans, america, iron, spiderman
Alien sci-fi	skyline, jarrod, balfour, strause, invasion, independence, cloverfield, angeles, district, los, worlds, aliens, alien, la, budget, scifi, battle, cgi, day, effects, war
Horror	gayheart, loretta, candyman, legends, urban, witt, campus, tara, reid, legend, alicia, englund, leto, scream, murders, slasher, helen, killer, student, teen, summer, cut, horror, final, sequel, scary
"Matrix"	neo, morpheus, neos, oracle, trinity, zion, architect, hacker, reloaded, revolutions, wachowski, fishburne, machines, agents, matrix, keanu, smith, reeves, agent, jesus, machine, computer, fighting, fight, real

표 5.6 는 $k = 500$ 으로 설정한 소나타 블로그 군집화 결과에 제안된 군집 별 키워드 추출 방법이 적용된 5 개 군집의 예시이다. 첫번째 군집은 제주도 여행과 렌트카 광고에 관련된 문서들이 하나의 군집으로 묶였으며, 두번째 군집은 중고차 매매에 관련된 문

서들이 하나의 군집으로 묶였다. 제안된 키워드 추출 방법은 각 군집을 구분할 수 있는 단어를 키워드로 선택하기 때문에 '자동차'나 '세단'과 같은 단어보다 차량 모델명이나 여행사 이름과 같은 단어가 키워드로 선택되었다. 네번째와 다섯번째 군집은 각각 가수 아이비의 노래인 "유혹의 소나타"와 관련된 문서가, "광염 소나타"를 비롯한 일제 강점기 소설과 관련된 문서가 군집을 이뤘다. 이들은 다른 문서 집합에 비하여 문서의 개수가 적은 군집이지만, 군집화 결과의 품질이 좋을 경우 제안한 키워드 추출 방법은 해석력이 좋은 단어를 키워드로 제공한다.

Table 5.6: 소나타 블로그의 $k=500$ 문서 군집화 결과

군집 의미	군집 별 키워드
렌트카 광고	제주렌트카, 부산출발제주도, 제주신, 이끌림, 제주올레, 왕복항공, 불포함, 제주도렌트카, 064, 롯데호텔, 자유여행, 객실, 제주여행, 특가, 해비치, 제주시, 제주항, 티몬, 2박3일, 올레, 유류, 항공권, 조식, 제주도여행, 제주공항, 2인
중고차 매매	최고급형중고, 최고급, 프리미어, 프라임, 2011년식, YF소나타TOP, 2010년식, 풀옵션, 2011년, YF소나타PR, 1인, Y20, 2010년, 완전무사고, 판매완료, 군포, 검정색, YF쏘나타, 2011, 하이패스, 2010, 무사고, 등급, 파노라마, 허위매물
클래식 장르	금관악기, 아이엠, Tru, 트럼펫, 트럼, 나팔, 금관, 텔레만, Eb, 호른, 오보에, Tr, Concerto, 하이든, 협주곡, Ha, 악기, 연주하는, 오케, 오케스트라, 독주, 악장, 작곡가, 곡
아이비, "유혹의 소나타"	Song, 공부할, 부른, 노래, 가사, 부르는, 가수, 보컬, 목소리, 발라드, 명곡, 신나, 들으면, 듣기, 유혹의, 앨범, 아이비, 제목
"광염 소나타" 및 일제강점기 소설들	백성수, 발가락, 현진, 이광수, 김유, 자연주의, 친일, 평양, 운수, 유미, 저지르, 야성, 탐미, 김동인, 복녀, 광염, 닮았다, 사실주의, 광기, 저지, 1920, 단편소설, 범죄, 감자, 동인, 한국문학

5.5 중심 벡터의 차원 축소와 군집 레이블을 이용한 군집화 결과 시각화

토픽 모델링 분야에서는 Latent Dirichlet Allocation 와 같은 토픽 모델링의 학습 결과를 시각적으로 표현하기 위한 연구들이 제안되었다 [190, 42, 192, 41, 72, 167]. 특히 [42] 는 토픽 모델링처럼 고차원 공간을 해석할 경우에는 t-SNE [136, 204] 처럼

고정된 시각화 결과를 이용하면 공간을 왜곡하여 이해할 가능성이 높기 때문에 상호작용이 가능한 (interactive) 시각화 방법을 이용해야 한다고 언급하였다. LDAvis 는 토픽 모델링의 결과인 '문서 - 토픽' 확률 행렬과 '토픽 - 단어' 확률 행렬이 주어질 경우, 차원 축소 방법과 키워드 추출 방법을 이용하여 상호작용이 가능한 시각화된 토픽 모델링 결과를 제공한다 [190].

문서 군집화 결과는 토픽 모델링으로 해석할 수 있다. Latent Dirichlet Allocation [23] 나 Probabilistic Latent Semantic Indexing [82] 는 한 문서에 여러 개의 토픽이 존재할 수 있다고 가정하지만, k-means 는 하나의 문서에 하나의 토픽이 존재한다고 가정한다. 문서 군집화 결과의 '문서 - 토픽' 확률 벡터는 문서 별 토픽에 대한 ont hot 벡터이며, '토픽 - 단어' 확률 행렬은 군집 내 단어의 출현 비율 행렬이다.

LDAvis 는 각 토픽별로 키워드를 선택하기 위하여 두 종류의 키워드 점수를 조합 한다. 식 5.4 의 $P(w|t)$ 는 토픽 별 단어 출현 확률이며 자주 등장한 단어를 키워드로 선택함을 의미한다. $P(w|t) / P(w)$ 은 토픽 별 단어 출현 확률을 문서 집합에서의 출현 확률로 나눈 값으로, 특정 토픽에만 자주 등장한 단어를 키워드로 선택함을 의미한다.

$$r(w|t) = \lambda P(w|t) + (1 - \lambda) \times \frac{P(w|t)}{P(w)} \quad (5.4)$$

k-means 의 군집화 결과인 군집 중심값에 L1 정규화를 적용하면 군집 별 단어 출현 확률을 얻을 수 있으며, 이를 식 5.4 의 $P(w|t)$ 로 이용할 수 있다. 식 5.2 를 이용하면 특정 군집에만 자주 등장한 단어를 찾을 수 있으며, 이를 식 5.4 의 $P(w|t) / P(w)$ 로 이용할 수 있다.

LDAvis 와 제안한 키워드 점수를 이용하면 그림 5.7 처럼 군집화 결과를 시각화하여 표현할 수 있다.

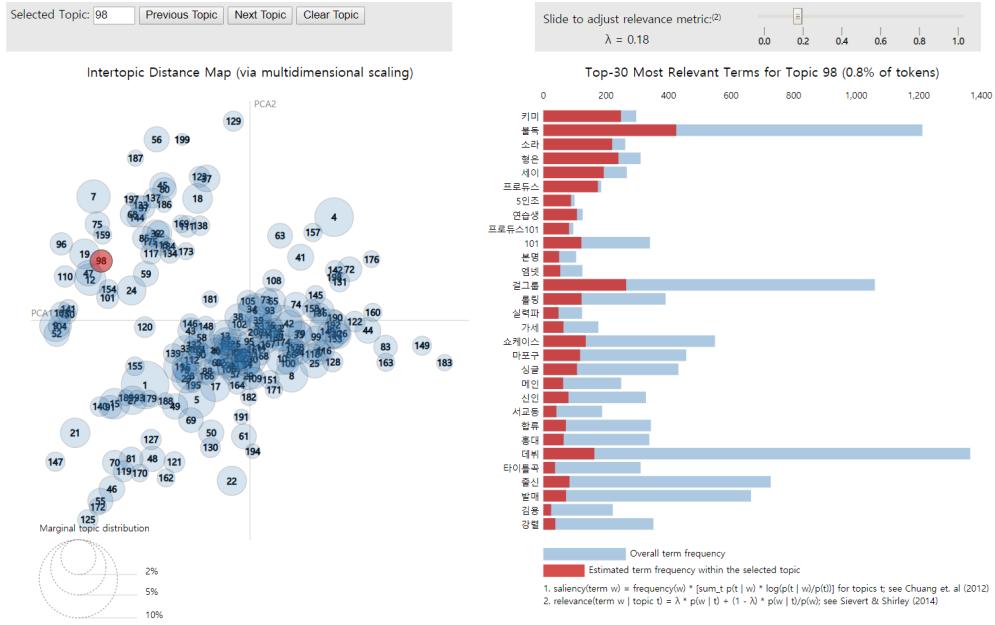


Figure 5.7: LDAvis 를 이용한 Spherical k-means 학습 결과 시각화 예시

5.6 결론

이 장에서는 문서 집합이 다양한 주제의 문서들로 구성된 경우, 문서 군집화 기법을 이용하여 단일한 주제의 문서들로 군집을 구성한 뒤 각 군집의 키워드를 추출함으로써 문서 집합을 요약하는 방법을 제안하였다. 문서 집합의 크기가 클 경우에도 효율적인 Spherical k-means 학습이 가능하도록 초기화 함수를 개선하였으며, 추가적인 키워드 추출 모델의 학습 없이 군집 중심값만을 이용하여 각 군집의 키워드를 추출하는 방법도 함께 제안하였다. 제안된 방법은 7 개의 데이터를 이용한 실험을 통하여 군집화 품질의 손실없이 효율적으로 문서 군집화 학습이 가능함이 확인되었으며, 데이터의 크기가 클수록 효율성이 증가하였다. 제안된 군집 별 키워드 추출 방법은 각 군집의 의미를 해석할 수 있는 단어들을 키워드로 제공함을 정성 평가를 통하여 확인하였다. 또한 토픽 모델링의 시각화 방법인 LDAvis 를 이용하면 Spherical k-means 의 학습 결과도

시각화 할 수 있었다.

4 장에서 제안된 핵심 문장 추출 방법은 문서 집합과 키워드 집합이 주어질 경우 핵심 문장의 추출이 가능하다. 5 장에서는 제안된 방법은 각 군집 별 키워드를 제공하기 때문에 4 장의 방법을 함께 이용할 경우 군집 별 핵심 문장 추출도 가능하다.

제안된 방법은 k-means 알고리즘의 한계점을 개선할 경우 더 정확한 문서 요약이 가능하다. 첫째로 k-means 알고리즘은 적절한 군집의 개수 k 를 사용자가 직접 설정해야 한다. 군집의 개수가 지나치게 작으면 여러 주제가 하나의 군집으로 합쳐져 한 군집의 키워드를 해석하기 어려우며, 군집의 개수가 지나치게 클 경우 한 주제가 여러 개의 중복된 군집으로 나뉘어지기 때문에 식 5.2 이 잘 정의되지 않는다. Silhouette 점수는 군집의 개수를 결정하는데 이용할 수 있는 척도이지만 고차원 벡터 공간에서는 잘 작동하지 않는다 [5]. 또한 Silhouette 점수는 수학적인 척도일 뿐, 사람이 느끼는 군집화 품질을 잘 반영하지는 못한다 [154]. 그렇기 때문에 문서 군집화 과정에서 적절한 군집의 개수를 정의할 수 있는 새로운 방법이 연구되어야 한다.

둘째로 k-means 알고리즘은 클래스의 크기가 서로 다르거나 각 클래스 별 문서의 개수가 다르더라도 각 군집의 크기가 비슷해지는 uniform effect [216] 가 발생하며, 이로 인하여 크기가 작은 클래스가 독립된 군집으로 학습이 되지 않는 문제가 존재한다. 이들은 다른 군집에 아웃라이어로 포함되는데, 제안된 키워드 추출 방법은 군집을 구분 할 수 있는 단어를 키워드로 선택하기 때문에 크기가 작은 클래스로부터 추출된 단어에 의하여 잘못된 문서 요약 결과가 제공될 수 있다. 이를 해결하기 위해서는 클래스의 크기와 문서 개수에 영향을 적게 받는 문서 군집화 방법이 연구되어야 한다.

제 6 장 시계열 형식의 뉴스 문서 집합 요약을 위한 거리 기반 유사 주제 구간 분리

6.1 개요

트위터나 뉴스와 같은 소셜 미디어에서 발생하는 문서들은 시간이 변함에 따라 문서 집합의 주제들이 달라진다. 토픽 탐색 및 추적은 (Topic Detection and Tracking, TDT) 새롭게 발생하는 주제를 인지하거나 추적하는 자연어처리 과업이다 [67]. 문서 집합을 시간의 변화를 기준으로 요약하면 문서 집합 전체의 변화에 대한 이해를 높일 수 있다. 예를 들어 특정 질의어에 대한 소셜 미디어의 시간의 흐름에 따른 문서 집합을 요약 함으로써 해당 질의어에 대한 트렌드를 이해할 수 있다.

문서 집합이 여러 주제의 문서들로 구성되어 있을 때에는 문서 집합을 주제 별로 나눈 뒤, 각 주제 별로 문서를 요약할 수 있다. 이를 위하여 5장에서 문서 군집화와 군집 별 키워드 추출을 이용하는 방법을 제안하였다. 문서의 작성자나 작성 시간과 같은 외부 정보가 함께 주어지는 경우들이 존재하며 이들은 비슷한 주제의 문서 집합을 탐색하는데 이용될 수 있다 [132, 14]. 특히 문서의 생성 시간은 동일한 주제의 문서를 탐색하는데 유용하게 이용될 수 있는데, 트위터나 뉴스와 같은 소셜 미디어에서는 시간의 변화에 따라 큰 트렌드가 존재하며 [55] 비슷한 시간대에 생성된 문서들은 비슷한 주제들로 구성될 가능성이 높다 [22, 6]. 하지만 작성자, 시간, 단어처럼 서로 다른 정보로 구성된 데이터는 점들 간 거리를 정의하기 어렵기 때문에 군집화 알고리즘을 적용하기 어렵다 [79, 29].

이러한 문제를 해결하기 위하여 Latent Dirichlet Allocation [23] 의 학습에 시간 정보를 함께 이용하는 방법들이 제안되었다 [22, 211, 86]. 문서 집합을 기간 별로 나누

어 각 기간 별로 토픽 모델을 학습한 뒤, 외부에서 정의된 질의어나 키워드를 중심으로 관련 주제의 문서 집합을 요약하는 방법들이 제안되었다. 그러나 사용자에 의하여 나뉘어지는 기간은 토픽의 변화 정보를 반영하지 못하는 한계점이 있다. 예를 들어 뉴스 문서에서는 한 사건에 관련된 문서가 짧게는 하루 혹은 수 주처럼 다양한 기간에 걸쳐 발생되지만, 임의로 정해진 기간으로 문서 집합을 구분할 경우 동일한 주제의 문서 집합이 여러 개의 토픽으로 나뉘어 학습될 수 있다. 또한 모든 문서를 이용하여 토픽 모델을 학습한 뒤 질의어를 중심으로 문서 집합을 요약할 경우, 질의어가 포함된 문서의 크기에 따라 질의어에 관련된 토픽이 제대로 학습되지 못할 가능성이 존재한다. 질의어가 주어질 경우 시간의 흐름에 따라 발생하는 문서 집합은 많은 수의 문서로 구성된 큰 주제와 작은 수의 문서로 구성된 작은 주제들로 이뤄져 있는 경우가 많은데, 모든 문서를 이용하여 토픽 모델을 학습할 경우 질의어 외의 정보에 의하여 이러한 패턴이 제대로 학습되지 않을 수 있다.

6 장에서는 특정 질의어에 대하여 시간에 따라 발생하는 문서 집합을 요약하는 방법을 제안한다. 이 장에서는 한 질의어에 대하여 시간 별로 큰 주제가 존재하는 상황에서 큰 주제를 기준으로 문서 집합의 발생 기간을 나눈 뒤, 각 기간 별로 키워드와 핵심 문장을 추출하는 방법을 제안한다. 큰 주제의 변화를 인식하기 위하여 시계열 구분 (time-series segmentation) 방법을 이용하며, 이 방법으로 구분된 각 구간은 5 장에서의 군집으로 해석할 수 있다. 그러므로 각 기간 별 문서 집합을 구분하는 키워드를 추출함으로써 해당 기간의 문서 집합의 주요 주제를 요약할 수 있다.

6.2 관련 연구

Latent Dirichlet Allocation 은 문서 집합으로부터 Dirichlet 분포를 따르는 문서의 토픽 확률 벡터 α 와 토픽의 단어 확률 벡터 β 를 학습한다. Latent Dirichlet Allocation

는 문서 집합을 구성하는 주제의 종류가 변하지 않는다고 가정하기 때문에 시간의 흐름에 따라 발생하는 모든 문서 집합에 동일한 토픽 - 단어 벡터를 이용하여 문서를 토픽 확률 벡터로 표현한다. 그러나 기간에 따라 서로 다른 토픽이 발생하며, 이러한 특징을 반영하기 위하여 기간별로 문서 집합을 나눈 뒤 여러 개의 토픽 모델을 학습하는 방법이 제안되었다 [22, 143, 6]. 이와 같은 방법은 다음의 한계점을 지니는데, 첫째 인접한 기간의 토픽은 서로 비슷할 수 있으나 독립적인 토픽 모델들은 이러한 정보를 이용할 수 없다. 둘째, Latent Dirichlet Allocation 은 모든 기간마다 사용자에 의하여 지정된 개수의 토픽이 존재한다고 가정한다. 데이터 기반으로 토픽의 개수를 추정하는 방법도 제안되었지만 [209], 토픽 별 분포가 불균형적이거나 토픽 별로 등장하는 단어가 명확히 구분되지 않는 상황에서는 좋지 못한 성능을 보인다. 셋째, 적절한 기간을 나누는 기준이 없다. 뉴스 문서의 경우 한 사건마다 하루 혹은 수주에 걸쳐 다양한 기간 동안 문서가 발생하기 때문에 임의로 기간을 정할 경우 한 토픽이 여러 개로 나뉘어지거나 여러 토픽들이 하나의 토픽으로 학습될 수 있다.

이러한 단점을 해결하기 위하여 토픽 모델링 과정에서 시간에 대한 정보를 이용하기 위한 다양한 연구들이 제안되었다. discrete-time Dynamic Topic Model (dDTM) [22] 나 Online LDA [6] 는 각 시점 t 에 대한 β_t 가 이전 시점의 β_{t-1} 와 유사하도록 식 6.1 과 같은 제약조건을 추가하였다. 이 과정에서 β_{t-1} 에 등장하지 않은 단어가 t 시점에 등장할 때 이를 인식하지 못하는 미등록문제를 해결하는 연구도 제안되었다 [227].

$$\beta_t \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I) \quad (6.1)$$

Latent Dirichlet Allocation 모델은 사용자가 설정한 토픽의 개수가 실제 존재하는 토픽의 개수와 비슷할 때 좋은 성능을 보인다 [7, 111]. 학습에 이용하는 문서 집합마다 토픽의 개수가 다를 수 있지만 dDTM 은 매 기간별로 동일한 개수의 토픽이 존재한

다고 가정한다. 이러한 문제를 해결하기 위하여 Locality Sensitive Hashing (LSH) 을 이용하는 방법이 제안되었다 [213]. LSH 는 Random Projection (RP) [20] 에 기반한 방법으로, 원 공간에서 유사한 두 벡터를 하나의 집합인 버킷 (bucket) 으로 그룹화한다 [87, 33]. 각 버킷은 비슷한 문서들로 구성되어 있으며 이를 토픽으로 생각할 수 있다. 버킷의 개수는 제한이 없기 때문에 새로운 토픽을 탐색하거나 [56], 주어진 문서와 유사한 토픽의 문서가 존재하는지 탐색하는데 이용될 수 있다 [228, 196]. 문서 군집화도 새로운 토픽의 탐색에 이용되었다 [222, 213]. 각 문서 군집을 하나의 토픽으로 가정한 뒤 이전 시점에 생성된 군집과의 거리가 사용자가 정의한 임계값보다 클 경우 새로운 군집을 생성함으로써 토픽의 개수에 제한이 있는 문제를 해결하였다.

dDTM 나 Online LDA 를 학습하기 전 사용자에 의하여 문서 집합을 기간 별로 나누어야 하는 문제를 해결하기 위하여 continuous-time Dynamic Topic Model (cDTM) 은 임의의 두 시점 $t - k, t$ 에 대한 제약조건을 식 6.2 의 $\Delta_{t-k,t}$ 처럼 두 시점에 대한 연속 함수로 정의함으로써 이러한 한계점을 완화하였다 [208].

$$\beta_t \sim \mathcal{N}(\beta_{t-1}, \Delta_{t-k,t}) \quad (6.2)$$

혹은 시계열 분리 모델을 이용하여 동일한 토픽으로 이루어진 구간을 탐색하는 방법도 제안되었다. [103] 은 Hidden Markov Model (HMM) 을 이용하여 매 시점마다 발생하는 관측값 o_t 으로부터 각 시점의 상태 (state) s_t 를 추정하는데, 인접한 시점의 상태 s_t, s_{t-1} 가 변화할 때 모델의 학습 비용을 증가하도록 비용 함수를 개선함으로써 유사한 관측값이 발생하는 구간을 분리하였다. [198, 86] 는 기간 별 단어의 발생 빈도를 o_t 로 이용하여 단어를 기준으로 구간을 분리한 뒤 dDTM 을 학습하는 방법을 제안되었다. 그러나 이러한 접근법은 상관 관계가 있는 여러 단어들을 동시에 이용하지 못한다.

여러 종류의 변수를 이용하여 시계열 데이터의 구간을 분리하기 위한 방법도 제안되

었다. [94, 225] 은 Feed Forward Neural Network 나 Convolutional Neural Network 를 이용하여 여러 종류의 항목으로 이뤄진 입력값을 벡터로 표현한 뒤 Recurrent Neural Network 에 시간의 흐름에 따른 패턴의 변화를 학습시켰다. [34] 은 값이 존재하지 않거나 데이터 생성 시간이 연속적인 상황을 해결하며 시계열 구간을 나누는 방법을 제안하였다. 이러한 방법은 텍스트와 같은 고차원의 데이터에서도 이용되었는데, [100] 은 임베딩 방법을 이용하여 대화의 문장을 벡터로 표현한 뒤 동일한 토픽의 대화를 구분하는 방법을 제안하였다. 그러나 이들은 학습 데이터를 이용하는 지도학습 기반 방법으로, 학습에 등장한 패턴만 인식하는 한계가 있다.

[99] 은 구간 별 패턴에 대한 레이블이 없는 상황에서 동일한 패턴의 구간을 탐색하기 위한 방법을 제안하였다. 각 시점 x_t 가 벡터 y_t 로 표현될 경우 시계열의 구간 분리는 시간이 변함에 따라 특정 지역에서 다른 지역으로 벡터들이 크게 이동하는 시점을 찾는 문제로 해석할 수 있다 (그림 6.1).

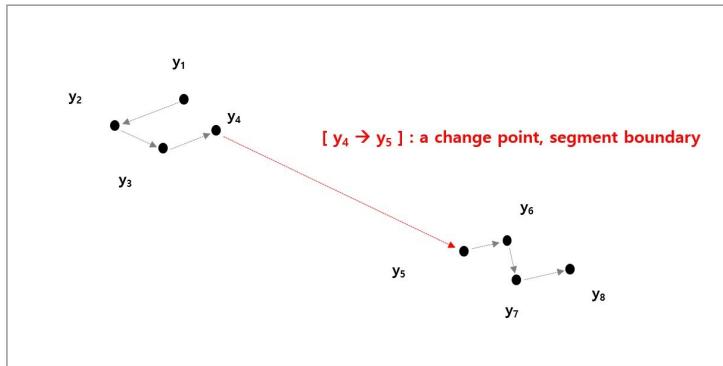


Figure 6.1: 시계열 분리의 기하학적 해석

하지만 시계열 벡터값의 표현에 분산이 클 경우 인접한 두 시점 간의 거리를 기반으로 시계열을 분리하면 매우 작은 길이의 구간으로 나뉘어질 수 있다. 이러한 한계를 완화하기 위하여 식 6.3 처럼 인접한 주변의 모든 벡터들 간의 거리의 가중 평균을

이용하는 방법도 제안되었다 [68].

$$S_t = \sum_{i=-s}^s \sum_{j=-s}^s w_{i,j} D_{(t+i,t+j)} \quad (6.3)$$
$$D_{p,q} := |y_p - y_q|$$

시점들이 각 시점의 특징을 잘 설명할 수 있는 벡터로 표현된다면 임의의 시계열 데이터에 대하여 식 6.3 의 적용이 가능하다. [99] 는 수치형 및 명목형 변수가 혼합된 차량의 주행 과정 중 수집된 센서 데이터를 이용하여 동일한 패턴으로 주행하는 구간을 탐색하는 방법을 제안하였다. 또한 각 패턴을 대표하는 핵심 입력값들을 선택함으로써 구간을 요약하였다.

6.3 유사 주제 구간 분리를 이용한 시계열 형식의 문서 요약

이 장에서는 한 질의어에 대한 뉴스 문서를 요약하기 위한 방법을 제안한다. 한 질의어에 대한 뉴스 기사는 시점 t 별로 주요 주제가 존재하며, 시점에 따라 주요 주제가 변화한다. 하지만 시점 별 주요 주제에 대한 정답 데이터가 존재하지 않기 때문에 비지도기반으로 주요 주제가 변하는 시점을 탐지하여야 한다.

제안하는 시계열 형식 문서 집합을 요약하는 방법의 의사 코드는 6.2 처럼 다섯 단계로 구성되어 있다.

첫 단계에서는 시계열 형식의 문서 집합 D 를 시점 별로 벡터 X 로 표현한다. 이를 위하여 Bag-of-Words Model 과 같은 고차원 벡터의 표현 방법이나 Doc2Vec 과 같은 분산 표상 표현 (distributed representation) 의 방법이 모두 이용될 수 있다.

두번째 단계에서는 벡터로 표현된 시계열 형식의 X 를 이용하여 각 시점마다 주제의 변화 점수 B 를 계산한다. 이를 위하여 [99] 에서 제안한 거리 기반 구간 분리 방법이

이용될 수 있다.

세번째 단계에서는 주제의 변화 점수 B 가 m_b 보다 크고 길이가 m_l 보다 큰 구간을 분리한다. 뉴스 기사의 경우 주요 주제는 오랜 기간 동안 기사가 지속되기 때문에 구간의 최소 길이 제약은 주요 주제를 탐색하는데 유용하다.

네번째 단계에서는 각 구간 s 마다 키워드를 추출한다. s 를 문서의 군집으로 여길 경우 s 외의 다른 구간과의 단어 분포 차이로부터 키워드를 추출하는 5장의 식 5.2 을 이용할 수 있다.

마지막 단계에서는 추출된 키워드 집합 KW 과 4장의 그림 4.6의 핵심 문장 추출 방법을 이용하여 각 구간 s 를 요약하는 핵심 문장을 추출한다.

```
D: documents stream
w: window length
ml: minimum segment length
mb: minimum boundary score
k0: number of keyword candidates
k1: number of selected keywords
k2: number of keysentences
σ: minimum distance between selected key-sentences

def summarize_documents_stream (D, ml, mb, k0, k1, k2):
    KW = [] : keyword list
    KS = [] : keysentence list
    X ← encode(D)
    B ← compute change point score(X, w)
    S ← segment documents stream(X, B, ml, mb)
    for s in S:
        KWs, KVs ← segment labeling (S, s, k0, k1)
        KSs ← seleft keysentences (s, KVs, σ, k2)
        KW+ = KWs
        KS+ = KSs
    return KW, KS
```

Figure 6.2: 제안하는 시계열 형식 문서 집합 요약 방법의 의사 코드

6.3.1 시계열 분리를 위한 문서 집합의 구간 별 벡터 표현 방법

문서 혹은 문서 집합을 벡터로 표현하기 위하여 다양한 방법이 이용될 수 있다. 시계열 분리가 잘 되기 위해서는 동일한 주제를 포함하는 시점의 벡터들은 비슷한 값으로,

다른 주제를 포함하는 시점의 벡터들은 서로 다른 값으로 표현되어야 한다. 적절한 벡터 표현을 탐색하기 위하여 다음의 과정을 이용하였다. 수집된 뉴스 문서들은 생성된 날짜를 기준으로 병합하여 각 날마다 하나의 가상 문서 D_t 를 생성한다. 토크나이저를 이용하여 문서를 단어열로 분해한 뒤 단어 빈도 벡터 (term frequency, TF), TF-IDF [194], 그리고 Doc2Vec [112] 을 이용하여 D_t 를 X_t 로 벡터화 하였다.

벡터 표현 방법의 품질을 정성적으로 측정하기 위하여 벡터로 표현된 X_t 간의 거리 행렬을 이용하였다. 점들 간 거리 행렬은 시간의 변화에 따른 패턴의 변화를 시각적으로 이해할 수 있도록 도와주기 때문에 [14], 음악과 같은 시계열 형식의 데이터의 구조를 시각적으로 표현하는 목적으로 이용된다 [162, 163, 142].

그림 6.3 은 TF 를 이용하여 한 시점을 벡터화 한 뒤 시점 간의 거리를 계산한 행렬이며, 그림의 좌측 하단부터 우측 상단으로 이동하는 방향으로 시간이 증가한다. 진한 검정색일수록 X_i, X_j 간의 벡터 간 거리가 가까움을 의미하는데, (30, 30) 에서 (50, 50) 사이와 같은 진한 색의 정사각형은 해당 기간동안 비슷한 문서들이 생성되었음을 의미한다.

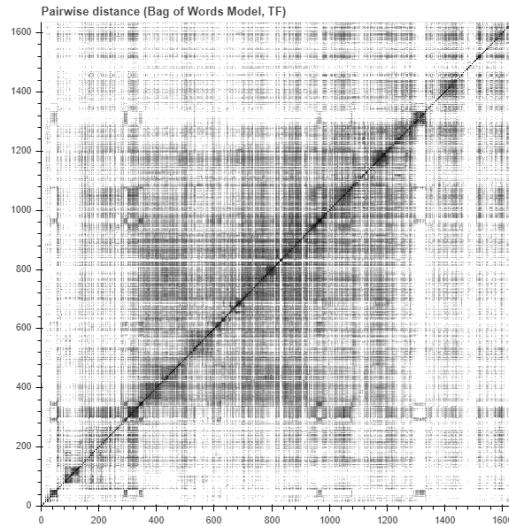


Figure 6.3: 단어 빈도 (term frequency) 기반 시점 간 거리 행렬 예시

하지만 한 질의어를 중심으로 수집된 뉴스 문서들은 같은 카테고리에 속할 가능성이 높으며, '기자', '오늘'과 같이 대부분의 뉴스 문서에서 등장하는 불용어들이 존재한다. 이들의 영향력을 줄이기 위하여 TF-IDF 가 이용될 수 있다. TF-IDF 는 단어가 등장한 문서의 비율이 1에 가까울수록 가중치가 0에 가까워지며, 이 값을 단어 빈도에 곱함으로써 불용어의 영향력을 줄인다 (식 6.4).

$$TFIDF(d, t) = TF(d, t) \times \log \frac{N}{DF(t)} \quad (6.4)$$

TF-IDF 를 이용하여 시점 간 거리 행렬을 그리면 그림 6.6 처럼 그림 6.3 보다 더 뚜렷한 정사각형들이 존재함을 확인할 수 있는데, 이는 TF-IDF 를 이용하면 주제가 변하는 시점이 뚜렷히 표현됨을 의미한다.

Doc2Vec 을 이용하여 문서를 벡터로 표현할 경우에는 그림 6.4 처럼 매 시점 별로 비슷한 벡터가 생성된다. 이는 앞서 설명한 것처럼 모든 문서 집합에 공통으로 등장하는 불용어들 뿐 아니라 뉴스 기사에는 특정한 형식과 관용어구가 존재하기 때문이며,

Word2Vec 을 단어의 임베딩 벡터로 이용하면 토픽이 다르더라도 문맥이 비슷한 단어에 의해 서로 다른 토픽의 문서도 비슷한 벡터로 표현될 수 있다. 이를 해결하기 위해서는 토픽 간 유사도를 보존하는 단어 임베딩 벡터를 이용하거나 [150, 133], TF-IDF 와 같이 소수의 단어에 의하여 문서 간 거리가 구분되는 벡터 표현 방법을 이용해야 한다.

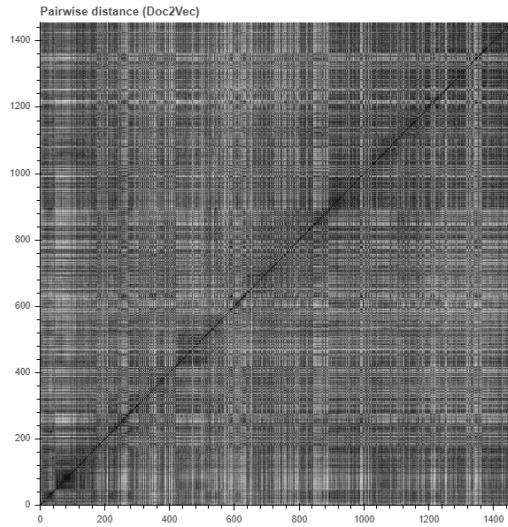


Figure 6.4: Doc2Vec 기반 시점 간 거리 행렬 예시

6.4 성능 평가

제안된 방법의 성능을 평가하기 위하여 네이버 뉴스로부터 2013-01-01 부터 2019-03-10 기간에 한국의 정치인 이름이 포함된 뉴스 기사를 수집하였다. 정성적 성능 평가를 위하여 세 명의 질의어가 포함된 뉴스를 이용하였다. 각 데이터셋은 각각 질의어 '김무성', '박근혜', '유시민'이 포함된 뉴스 기사이다 (표 6.1).

Table 6.1: 실험에 이용한 3 종류의 데이터셋

질의어	고유 날짜 수	뉴스 개수	고유 단어 개수	일평균 뉴스 개수
김무성	1,636	223,590	69,597	136.67
박근혜	2,260	1,339,266	249,152	592.60
유시민	446	18,239	13,619	40.90

세 종류의 데이터셋 모두 3 장에서 제안한 명사 추출기와 2 장의 L-TOKENIZER (그림 2.1)를 이용하여 각 문서를 단어열로 구분하였다. 날짜를 기준으로 작성된 뉴스 기사를 합쳐 가상의 문서를 생성한 뒤, TF-IDF를 이용하여 각 일자 별 벡터 X_t 를 학습하였다.

6.4.1 질의어 '김무성'이 포함된 뉴스 기사의 구간 분리

질의어 '김무성'이 포함된 뉴스는 2014년 중순부터 2016년 중순, 그리고 2016년 말에 많은 수의 뉴스가 발생하며, 간헐적으로 짧은 기간 많은 양의 뉴스가 발생한다 (그림 6.5). 빨간 선은 제안된 방법에 의하여 탐색된 구간의 경계선이다. 새로운 사건이 발생할 때 많은 양의 뉴스가 작성되는데, 제안된 방법은 일별 뉴스 문서의 개수 정보를 이용하지 않았음에도 주제가 바뀌는 순간을 인식하였음을 알 수 있다.

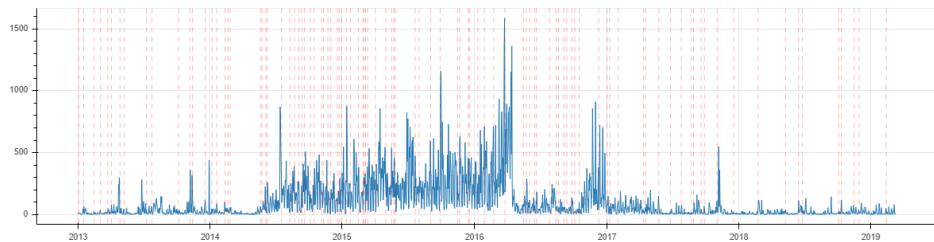


Figure 6.5: 질의어 '김무성'이 포함된 뉴스의 일별 문서 개수

그림 6.6는 일간 거리 행렬로, 짧은 기간 내에 비슷한 주제의 뉴스들이 발생하였음을 확인할 수 있다. 1,636 일의 뉴스는 길이가 2 일 이상인 99 개의 구간으로 나뉘어졌으며, 각 구간의 평균 길이는 8.74 일이다.

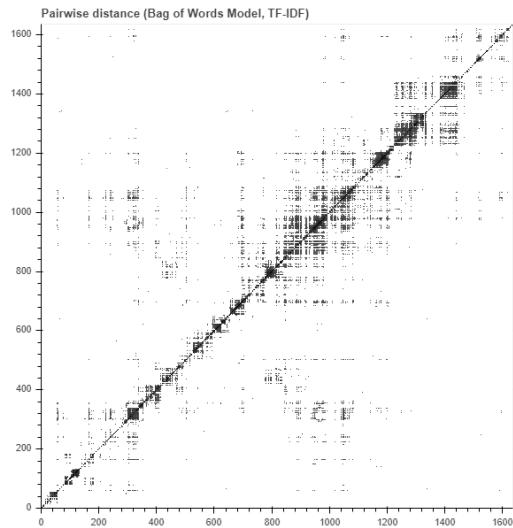


Figure 6.6: 질의어 '김무성'이 포함된 뉴스 문서의 일간 거리 행렬

표 6.2는 그림 6.5에서 뉴스의 개수가 급증한 기간이 포함된 구간의 키워드와 핵심 문장의 예시이다. 2016년 12월에는 당시 새누리당 내 의원들의 당파 논쟁으로 인하여 여러 개의 보수 정당들이 생성되는 시기이며, 2017년 11월에는 신규 보수 정당의 의원들이 당시 자유한국당으로 재입당하는 과정에서 의원들간의 논쟁이 격화되던 시기이다. 해당 시기 김무성 의원 역시 바른 정당으로 탈당 후 자유한국당으로 복당하였는데, 해당 사건에 관련된 키워드와 핵심 문장들로 해당 시기가 요약되었음을 알 수 있다. 이러한 정치적 사건 외에도 일명 '노루패스'로 불리는 해프닝 역시 하나의 구간으로 탐색됨을 볼 수 있다. 5월 23일 발생한 '노루패스' 사건은 이후 여러 패러디로 이어졌는데, 이와 같이 시기성이 짧은 사건도 하나의 구간으로 분리되어 요약될 수 있다.

6.4.2 질의어 '박근혜'가 포함된 뉴스 기사의 구간 분리

질의어 '박근혜'가 포함된 뉴스는 '김무성'이 포함된 뉴스보다 상대적으로 많이 발생하였으며, 탄핵 과정 및 국정농단 조사 기간인 2016년 말부터 2017년 초까지 급증하였다

(그림 6.7). 2,260 일간의 뉴스는 길이가 2일 이상인 68 개의 구간으로 나뉘어졌다. 이전의 예시처럼 그림 6.7에서도 뉴스의 개수가 급증할 때 구간이 나뉘어지나, 각 구간의 평균 길이는 32.5 일로 매우 길다.

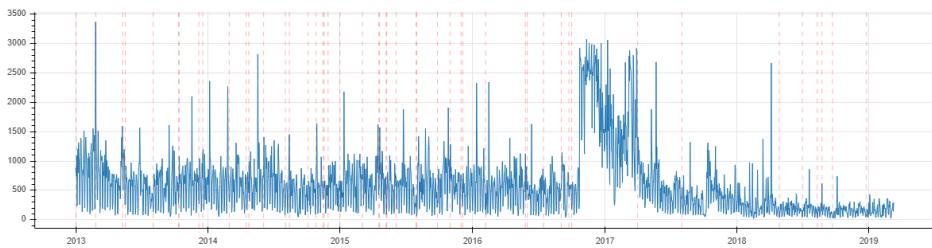


Figure 6.7: 질의어 '박근혜'가 포함된 뉴스의 일별 문서 개수

구간이 긴 이유는 질의어 '박근혜'가 하나의 의미로 이용되지 않았기 때문이다. 대통령 재임 시절에는 '박근혜 정부'처럼 정부를 지칭하는 표현으로 이용되었으며 재임 이후에는 국정농단 사건과 관련된 많은 기사에서 세부 주제와 관계없이 질의어가 자주 등장하였기 때문이다. 그 결과 큰 트렌드가 변할 때에만 문서 집합의 벡터가 변하였다. 이처럼 한 기간에 여러 주제가 포함되어 있을 경우에는 이들을 포함하는 상위 주제가 변하는 지점을 경계로 구간이 나뉘어진다.

표 6.3에서 살펴볼 수 있듯이 질의어 '박근혜'가 포함된 뉴스는 대선 당선과 인수위 원회, 탄핵 시국, 국정농단 조사 및 판결과 같은 큰 사건을 중심으로 구간이 나뉘었음을 확인할 수 있다. 제안하는 방법은 큰 트렌드를 기준으로 구간을 나누기 때문에 각 기간 별로 여러 개의 세부 주제가 존재한다. 그럼에도 불구하고 각 구간에서 자주 등장하며 다른 구간에서 등장하지 않는 단어를 중심으로 키워드를 선택하였기 때문에 해당 군집의 큰 트렌드를 이해할 수 있는 키워드와 핵심 문장들로 구간이 요약되었다.

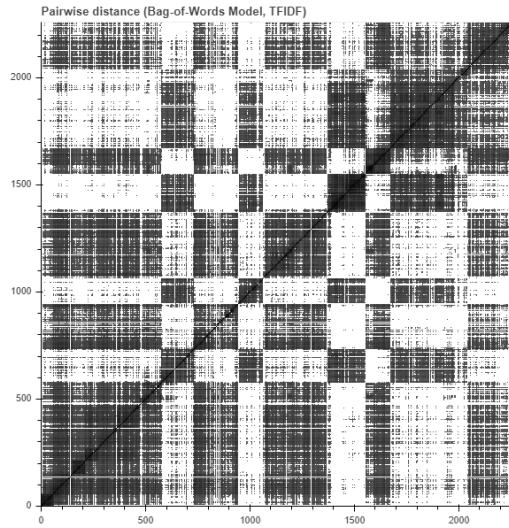


Figure 6.8: 질의어 '박근혜'가 포함된 뉴스 문서의 일간 거리 행렬

6.4.3 질의어 '유시민'이 포함된 뉴스 기사의 구간 분리

반면 질의어 '유시민'이 포함된 뉴스는 이전의 뉴스들보다 적은 수의 뉴스가 생성되었으며, 2016년 부터 2018년 까지 주기적으로 많은 뉴스가 발생하였다 (그림 6.9). 이는 해당 기간에 진행한 방송프로그램 '썰전'과 '알쓸신잡'을 중심으로 뉴스가 생성되었기 때문이며, 해당 프로그램의 기사를 중심으로 구간이 나뉘어졌다. 그렇기 때문에 446 일의 뉴스가 91 개의 짧은 기간들로 나뉘어졌다. 그림 6.10 에서도 각 일자별로 비슷한 주제가 생성되는 경우가 적음을 확인할 수 있다.

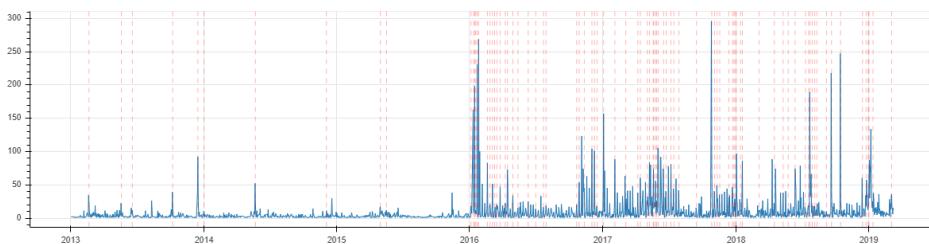


Figure 6.9: 질의어 '유시민'이 포함된 뉴스의 일별 문서 개수

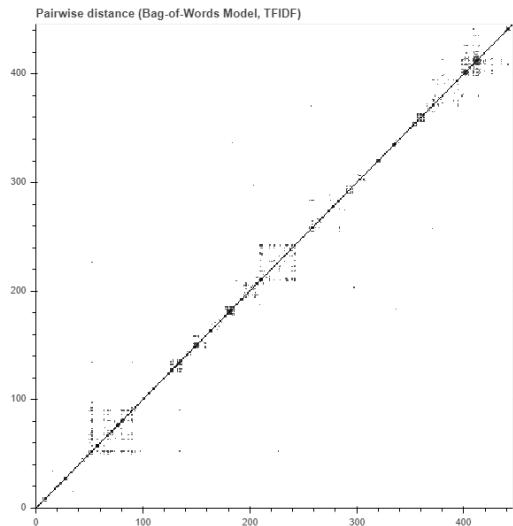


Figure 6.10: 질의어 '유시민'이 포함된 뉴스 문서의 일간 거리 행렬

그 결과 표 6.4 의 구간 별 문서 요약의 결과는 이전 두 데이터와 달리 진행 중이던 방송프로그램의 내용에 관련된 내용처럼 세부적인 주제를 요약하는 키워드와 핵심 문장으로 구성되었다.

6.5 결론

이 장에서는 시계열 형식으로 발생하는 문서 요약을 위하여 문서 집합을 비슷한 주제를 포함하는 구간으로 구분한 뒤, 각 구간을 요약하는 방법을 제안하였다. 문서가 생성된 시간 정보는 주제의 변화를 인식하는데 유용한 정보이지만, 단어의 빈도와 문서 생성 시간처럼 서로 다른 정보로 구성된 데이터 간에는 거리를 정의하기 어렵기 때문에 군집화 알고리즘이 이용되기 어렵다. 이러한 문제를 해결하기 위하여 시간이 변함에 따라 문서 집합의 주제가 변하는 지점을 구간의 경계로 탐색하는 방법을 제안하였다. 문서 집합을 생성 시간 단위로 병합하여 벡터로 표현한 뒤, 시간이 변함에 따라 벡터 간 거리가 멀어지는 지점을 구간의 경계로 선택하였다. 각 구간은 문서 군집화 방법의

군집으로 해석할 수 있기 때문에 5 장에서 제안한 군집 중심값 기반 키워드 추출 방법을 이용하여 키워드를 추출한 뒤 4 장에서 제안한 핵심 문장 추출 방법을 적용하여 구간을 요약하였다.

제안한 방법의 성능을 평가하기 위하여 네이버 뉴스로부터 정치인 이름이 포함된 뉴스를 수집하였다. 다수의 사람들에게 알려진 유명 정치인들의 뉴스 기사에 제안한 방법을 적용한 뒤 요약 결과를 해석하는 정성적 평가를 진행하였다. 생성되는 뉴스의 개수가 많고 주제가 다양할 경우 다양한 주제를 아우르는 큰 주제를 중심으로 구간이 길게 나뉘며, 반대의 경우에는 자세한 주제를 중심으로 짧은 여러 개의 구간이 나뉘어졌다. 하지만 구간의 길이와 관계없이 해당 기간의 사건을 설명하는 대표적인 키워드와 문장이 추출됨을 확인하였다. 즉 제안한 방법은 기간 별 문서 집합을 구성하는 주제의 세분성 (granularity) 을 조절하며 구간을 분리한다.

그러나 제안된 방법은 시간의 변화에 따른 큰 트렌드만을 파악할 수 있다. 큰 주제는 여러 개의 세부 주제를 바탕으로 계층적 구조를 이루며, 시간에 따른 계층적 문서 요약이 가능할 경우 주제의 세분성을 조절하며 특정 기간의 요약 결과를 제공할 수 있다.

또한 제안하는 방법은 TF-IDF 를 기반으로 각 시점의 벡터로 표현하였다. 하지만 Bag-of-Words Model 은 각 시점이 매우 큰 차원의 벡터로 표현되며 데이터 희귀성 문제를 겪을 수 있다. 이를 해결하기 위하여 Doc2Vec 과 같은 분산 표상 표현 방법이 이용되기도 하지만, 토픽 정보를 활용하지 않는 단어 임베딩 방법을 이용하여 문서를 벡터로 표현할 경우 대부분 시점의 문서들이 비슷한 벡터로 표현되는 한계가 있다. 이를 해결하기 위하여 토픽 모델 기반 단어 임베딩 방법과 이를 이용하는 문서 임베딩 방법이 연구되어야 한다.

Table 6.2: 질의어 '김무성'이 포함된 뉴스의 구간 별 문서 요약 예시.

날짜	요약
2016-12-10 ~ 2016-12-30	<p>개혁보수신당, 보수신당, 가칭, 신당, 창당, 개혁, 중도, 분당, 비대위원장, 보수, 탈당, 유승민, 주호영, 비상시국회, 분열, 친박계, 모임, 황영철, 나경원, 정책, 친박, 선출, 고민, 의원, 비박, 선언</p> <ul style="list-style-type: none"> - 김무성 유승민 등 새누리당 비박근혜 계 의원 34명이 탈당을 선언하고 신당의 가칭을 보수신당이라고 정했다 - 28일 국회 의원회관에서 주호영 원내대표 김무성 유승민 의원 등이 참석한 가운데 개혁보수신당 정강정책 토론회가 열리고 있다 - 김무성 전 대표가 친박 지도부로는 당 재건이 어렵다며 탈당을 시사해 비대위원장 선출이 분당 분수령이 될 거란 관측이 나옵니다 - 아시아경제 윤동주 기자 김무성 새누리당 전 대표가 26일 국회에서 열린 가칭 개혁보수신당 창당 준비위원회에 참석하고 있다 - 김무성 전 새누리당 대표가 친박근혜계를 향해 대통령의 정치적 노예들이라며 강하게 비판하며 중도 보수 창당을 고민하고 있다고 밝혔다
2017-05-25 ~ 2017-05-26	<p>유병재, 패러디, 반납, 방송인, 노룩, 노룩패스, 공항, 네티즌들, 웃음, 캐리어, 화제, 제목, 공개, KBS, 커뮤니티</p> <ul style="list-style-type: none"> - 서울경제 방송인 유병재가 김무성 의원의 캐리어 노룩 패스를 패러디해 눈길을 끌었다 - 그룹 투포케이 역시 26일 오전 KBS 2TV 뮤직뱅크 리허설을 가는 길에 김무성의 노룩패스 패러디를 해 웃음을 자아냈다 - 앞서 김무성 의원이 공항에서 수행원을 보지 않고 캐리어를 굴려 전달하는 모습이 포착돼 큰 화제를 모은 바 있다 - 김무성 의원 등 전 새누리당 소속 의원 27명이 오는 5월31일까지 1년치 세비를 국가에 반납할지 관심이 쏠리고 있다 - 김무성 의원의 행동은 온라인에서 지탄을 받았고 해외 온라인 커뮤니티 데딧 등에서 한국인의 스웨그라는 제목으로 공개되기까지 했다
2017-11-02 ~ 2017-11-15	<p>탈당계, 재입당, 복당파, 입당, 강길부, 지위, 정양석, 의총, 제출, 간담회, 탈당, 선언, 이종구, 황영철, 김용태, 집단, 당사, 기자회견, 김영우, 교섭단체, 의원총회, 친박, 창당, 홍준표</p> <ul style="list-style-type: none"> - 재입당 국회의원 간담회에서 바른정당을 탈당한 김무성 강길부 김영우 김용태 이종구 황영철 정양석 홍철호 의원과 손을 잡고 기념촬영을 하고 있다 - 바른정당은 지난 8일 김무성 의원 등 8명이 일제히 탈당계를 제출한 뒤 자유한국당에 복당 교섭단체 지위를 잃었다 - 홍준표 자유한국당 대표가 9일 오전 서울 여의도 당사에서 열린 바른정당 탈당파 의원들의 재입당 간담회에서 김무성 의원과 대화를 나누고 있다 - 바른정당의 김무성 의원 등 8명이 6일 탈당을 선언하기로 했다 이로써 바른정당은 창당 9개월여 만에 절반으로 조개지게 됐다 - 실제 강성 친박계는 김무성 의원 등 8명의 탈당파 복귀만으로도 의총 소집을 요구하는 등 조직적 반발 기류를 드러내고 있다

Table 6.3: 질의어 '박근혜'가 포함된 뉴스의 구간 별 문서 요약 예시

날짜	요약
2013-01-02 ~ 2013-02-24	<p>내정자, 후보자, 인사청문, 의혹, 삼청동, 지명, 취임식, 핵실험, 김용준, 장관, 대통령직인수위원회, 업무, 청와대, 오후, 부처, 인수위원회, 대통령직, 위원장, 논의, 검토, 방안</p> <ul style="list-style-type: none"> - 정 총리 후보자는 8일 삼청동 인수위에서 기자회견을 갖고 박근혜 정부의 초대 국무총리로 지명된 소감에 대해 이같이 말했다 - 박근혜 대통령 당선인과 김용준 대통령직인수위원회 위원장이 30일 오후 서울 종로구 삼청동 대통령직 인수위원회에서 열린 정무분과 업무보고에 참석했다 - 20일 정홍원 국무총리 후보자의 국회 인사청문회를 시작으로 박근혜 정부의 17개 부처 초대 장관 내정자들의 청문회가 줄줄이 이어진다 - 박근혜 대통령 당선인은 이번주 초 총리 후보자와 청와대 인선을 함께 발표하는 방안을 검토하고 있는 것으로 전해졌습니다 - 박근혜 정부 초대 국무총리로 지명된 김용준 내정자의 검증 과정에서 재산 문제가 가장 먼저 도마에 오를 것으로 보인다
2016-10-02 ~ 2016-12-02	<p>하야, 국정농단, 최순실, 촛불, 퇴진, 시국, 탄핵, 게이트, 비선, 특검, 광화문, 행진, 검찰, 수사, 집회, 대국민, 시민, 의혹, 분노, 재단, 혐의, 헌법, 사태, 조사, 개입, 사건, 촉구, 요구, 사과, 임명, 연설</p> <ul style="list-style-type: none"> - 비선 실세 최순실 씨의 국정농단 사태의 책임을 물어 박근혜 대통령의 퇴진을 촉구하는 5차 주말 촛불집회가 26일 열린다 - 774억 원을 모금해 세웠는데 재단에 박근혜 대통령 측근인 최순실 차은택 씨가 개입했다는 의혹은 현재 검찰 수사가 진행 중입니다 - 5일 오후 서울 광화문광장에서 박근혜 정권 퇴진 집회에 가족 단위로 참여한 시민들이 촛불을 들고 행진하고 있다 - 박영수 변호사가 박근혜 게이트 수사의 특별검사로 임명되었다 대통령은 특검 수사에 적극적으로 협조하고 직접 조사에도 응하겠다고 한다 - 국정농단으로 박근혜 대통령의 하야와 탄핵을 요구하는 대학가의 시국선언이 번지고 있다
2018-02-18 ~ 2018-04-27	<p>1심, 구형, 세월호, 징역, 이명박, 유죄, 피고인, 선고, 한국당, 참사, 4월, 대변인, 국정, 혐의, 법원, 역사, 재판, 원장, 검찰, 인정, 사건, 조사, 뇌물, 소환, 지난해, 회장, 구속, 받은, 판단</p> <ul style="list-style-type: none"> - 국정농단 사건으로 징역 30년을 구형받은 박근혜 전 대통령의 1심 선고가 오는 4월 6일 내려집니다 - 또 박근혜 전 대통령이 지난해 3월 21일 검찰 조사를 받은 지 358일만에 소환된 전직 대통령이 됐습니다 - 법원은 박근혜 전 대통령의 삼성그룹 뇌물수수 혐의와 관련해 앞서 일부 유죄가 인정된 최순실씨와 동일한 판단을 내렸다 - 법원이 최순실 박근혜 1심 선고를 근거로 이 부회장 항소심 재판부의 판단이 잘못됐다고 본다면 뇌물액수가 커질 가능성이 크다 - 그는 이명박 박근혜 전직 대통령이 구속된 데 대해 한국당을 향해 쓴 목소리도 냈다

Table 6.4: 질의어 '유시민'이 포함된 뉴스의 구간 별 문서 요약 예시

날짜	요약
2016-03-04 ~ 2016-03-04	<p>자본, 무전취식, 삼청각, 시술, 20만, 고급, 세종문화회관, 직권상정, 국회의장, 직원, 테러방지법, 필리버스터, 홍보, 식사, 문구, 가득, 몰려, 방청객, 국가, 외모, 원인, 넥타이</p> <ul style="list-style-type: none"> - 이어 유시민은 고급 시술전문 성형외과나 피부과에서 조용히 한다며 요즘은 외모도 신체 자본이다이라고 말했다 - 이날 유시민 작가는 국회의장이 국가비상사태라고 테러방지법을 직권상정했다고 말문을 열었다 - 유시민은 방청객만 가득하고 의원석은 텅텅 비어있다면서 편의점에 알바생 하나 있고 손님들만 몰려 있는 것과 같은 것이라고 답했다 - 한편 이날 전원책과 유시민은 필리버스터 삼청각 무전취식 논란 등 다양한 사안을 놓고 토론을 펼쳤다 - 유시민은 세종문화회관 직원인 정팀장의 행각을 지적하며 20만 9천 원 짜리 밥 구경도 못해본 사람이 99.9% 다라고 말했다
2017-07-29 ~ 2017-07-29	<p>지식인, 보성, 시즌2, 순천, 통영, 출연진, 제작, 양정우, 도시, 28일, 기획, 지식</p> <ul style="list-style-type: none"> - 이후 편집 영상에서 유시민은 여행지와 관련한 풍부한 지식을 드리내 최고의 지식인다운 면모를 보였다 - 제가 유시민 선생님의 이름을 많이 팔고 다녔죠라고 말한 양정우 PD는 섭외 비하인드에 대해 털어놓기도 했다 - 통영 순천 보성 강릉 경주 공주 부여 세종 춘천 전주 등 전국 10개 도시를 돌아다녔다 - tvN 알쓸신잡은 28일 마지막 방송에서 출연진이 한 공간에 모여 그간 방송에서 전하지 못한 뒷얘기를 나눴다 - 한편 알쓸신잡 후속으로는 삼시세끼 바다목장편이 방송되며 시즌2는 현재 기획 중이다
2018-05-25 ~ 2018-05-25	<p>관전, 군사적, 불안감, 멸균, 남북고위급회담, 취소, 후보자, 제재, 포인트, 마감, 네거티브, 단일화, 발제, 북한, 무소속, 비핵화, 한반도, 불안, 보장</p> <ul style="list-style-type: none"> - 24일 방송된 JTBC 썰전에서 유시민 작자가 남북고위급회담 취소에 대한 자신의 의견을 밝혔다 - 그리고 북한이 원하는 게 있다면서 군사적 안전 보장과 국제 무대에서의 제재 철회를 꼽았다 - 이어 이런 것은 보건학적으로 설명이 가능하다 북한은 오랜 시간 주체사상 외 모든 다양한 의견을 멸균했다고 덧붙였다 - 6.13 지방선거의 두 번째 관전 포인트 네거티브입니다 선거 시즌만 되면 고질적으로 되풀이 되는 네거티브 논란 이번에도 시동이 걸렸습니다 - 이어 북한은 아직도 비핵화를 한반도 전체 비핵화로 이해하고 있다 그것은 미국 전략자산 배제하는 것을 포함한다고 덧붙였다

제 7 장 결론

인공지능 분야에서는 오래전부터 인간의 언어를 이해하고 처리할 수 있는 머신에 대한 연구가 이뤄졌다. 인간의 언어로 기술된 문제를 해결하기 위해서는 음성의 텍스트화, 파싱을 통한 텍스트의 구조 파악, 각 문제의 해결 및 인간의 언어로 표현되는 결과 도출 등의 다양한 세부 문제를 해결해야 한다. 자연어처리 분야는 이러한 과업들로 구성되어 있으며 머신러닝을 이용한 다양한 해법들이 제안되고 있다. 최근에는 뉴럴 네트워크 기반의 머신러닝 모델 발전에 힘입어 몇몇 자연어처리 과업에서 진일보된 성과를 얻고 있다. 특히 정보를 분산 표상으로 처리하는 방식은 단어나 문맥 같은 의미적인 정보를 학습하는데 탁월하였으며 번역, 추론, 대화 모델, 문장 생성과 같이 의미를 기반으로 해법을 찾아야 하는 분야의 급격한 발전을 이끌었다.

최근 발전 속도는 자연어처리의 세부 문제마다 다르다. 뉴럴 네트워크 기반 언어 모델처럼 단어 임베딩 정보를 이용하거나 어텐션 메커니즘을 이용하여 이전에는 활용하지 못했던 정보를 활용한다 [90, 12, 130]. 최근에는 단어의 의미를 넘어 문맥의 의미까지 학습할 수 있는 셀프 어텐션 방법들도 제안되고 있다 [205, 52]. 그러나 단어 임베딩 정보가 큰 역할을 하지 않는 문장 분류의 분야에서는 어텐션을 이용하기 전까지 발전이 더뎠다 [223]. 다른 예시로 토픽 모델링의 분야에서는 여전히 Latent Dirichlet Allocation [23] 를 기반으로 한 모델들이 이용되고 있다 [4].

대부분의 자연어처리 과업에는 토크나이징 과정이 포함되어 있다. 토크나이징은 인간의 언어를 컴퓨터가 이해할 수 있는 단위로 표현하는 첫번째 작업이다. 문장 형식의 텍스트를 단어, 형태소, 혹은 과업에 적합한 단위의 토큰으로 분해하여 인식한다. 전통적인 Bag-of-Words Model 의 문서 표현 방식부터 BERT 와 같은 모델까지 모두

토크나이저를 이용하여 입력 데이터를 정제한다. 토크나이저의 성능이 향상될수록 다양한 과업의 품질 향상에 도움을 줄 수 있다. 이 과업은 인간의 언어와 다른 자연어처리 과업들을 이어주는 교두보 역할을 한다.

그러나 토크나이저는 미등록단어 문제, 데이터 부족, 오류에 의하여 성능이 저하될 수 있다. 많은 종류의 토크나이저들은 단어나 형태소를 인식하기 위하여 사전을 이용 하며, 모호성을 해결하기 위하여 학습 말뭉치를 기반으로 한 모델을 학습한다. 하지만 언어는 시간과 도메인에 따라 변하기 때문에 하나의 학습 말뭉치로 학습된 토크나이저에는 늘 미등록단어 문제가 발생한다. 특히 교착어에 속하는 한국어는 단어나 형태소의 경계가 공백으로 구분되지 않기 때문에 사전을 이용하여야 하는데, 중국어의 영향으로 표의문자의 성격을 지니는 한국어에서는 미등록단어들이 음절 단위로 분해되는 현상이 발생한다. 문법이나 철자법 오류도 정확한 단어의 인식을 어렵게 만든다. 이러한 문제를 해결하기 위해서는 오류를 교정하거나 분석할 데이터에 적합한 학습 데이터가 보강되어야 하지만, 매번 새로운 학습 데이터를 보강하는 것은 현실적인 접근 방법이 아니다.

위의 세 가지 어려움은 문서 집합을 키워드나 핵심 문장으로 요약하는 문서 요약 과업에서도 발생한다. 문서 요약을 위하여 학습기반 모델을 이용할 경우 어떤 단어나 문장이 중요한 정보를 포함하고 있는지 태깅된 학습 데이터를 마련하기 어렵다. 또한 토크나이저를 이용하여 데이터를 처리하는 단계에서 중요한 단어가 제대로 인식되지 않으면 잘못된 키워드가 추출되어 문서 요약의 품질이 저하된다.

이러한 문제를 해결하기 위하여 이 논문에서는 한국어 어절 구조를 이용한 비지도 학습 자연어처리 방법들을 제안하였다. 비지도학습 접근법은 학습 데이터에 대한 의존도가 낮기 때문에 다양한 도메인에 쉽게 적용할 수 있다. 또한 비지도학습 접근법에 언어적 사전 지식인 어절 구조를 함께 이용하면 비지도학습 방법들이 필요한 정보만을

효율적으로 학습할 수 있다.

7.1 이 논문의 기여

이 논문은 다음의 기여를 하였다.

비지도학습 자연어처리를 위한 한국어 어절 구조인 $L + [R]$ 을 제안하였다. 한국어의 어절은 한 개 이상의 형태소로 구성되며, 전통적인 형태소 분석 관점에서는 이들을 모두 분해하여 인식한다. 하지만 비지도학습 단어 추출 문제의 관점에서는 형태소 분석의 관점은 지나치게 복잡하다. 한국어의 어절은 의미를 지니는 형태소들을 어절의 왼쪽에 (L), 문법 기능을 수행하는 형태소들을 어절의 오른쪽에 (R) 위치시킨다. 각 기능별로 이분된 복합형태소를 하나의 단어나 형태소로 취급하면 어절 구조를 단순화 할 수 있다. 미등록단어 문제는 의미를 지니는 L 부분에서 발생하기 때문에 단어 추출을 해야 하는 대상을 한정할 수 있다. 또한 L 과 R 을 구성하는 복합형태소를 개별 형태소로 분해할 수 있기 때문에 전통적인 형태소 분석과 상호 호환할 수 있다.

2 장에서 한국어에 적합한 비지도학습 토크나이저를 제안하였다. 미등록단어 문제 가 빈번한 경우에는 데이터를 기반으로 단어를 추출하여 토크나이징을 수행하는 비지도학습 토크나이저가 이용될 수 있다. 가장 널리 이용되는 Word Piece Model (WPM) 은 모든 언어에 공통적으로 이용할 수 있는 토크나이저로, 그 목적은 단어를 정확히 구분하는 것이 아닌 데이터를 최소한의 유닛으로 표현하는 것이다. 이러한 한계점을 개선하기 위하여 $L + [R]$ 구조를 이용하는 비지도학습 토크나이저를 제안하였다. 음절 단위의 언어모델과 Branching Entropy 를 이용하여 통계 기반으로 단어의 경계를 학습 하고, 띄어쓰기 오류 수준에 따라 적용할 수 있는 두 종류의 토크나이저 L-Tokenizer 와 Max Score Tokenizer 를 제안하였다. 제안한 방법은 단어 인식 과업과 문장 분류 문제 에서 WPM 보다 좋은 성능을 보였으며, 특히 사전을 이용하는 기존 형태소 분석기와도

유사한 단어 인식 성능을 보였다.

3 장에서 R 의 분포를 이용한 명사 추출기를 제안하였다. 명사는 미등록단어 문제 가 가장 빈번한 단어이기 때문에 다른 단어보다도 높은 인식 성능이 요구된다. 명사는 어절의 왼쪽에 등장하며, 그 오른쪽에는 조사나 용언으로 전성하는 복합형태소가 R 로 등장한다. 이러한 특징을 이용하여 R 의 분포를 기반으로 L 이 명사인지 판단하는 명사 추출기를 제안하였다. 제안된 방법은 정확도를 높이기 위하여 L 의 판별 순서를 정의하는 방법 및 후처리 과정을 통해 잘못 추출된 명사를 제거하는 과정을 포함한다. 제안한 방법은 사전 기반으로 작동하는 세 종류의 한국어 형태소 분석기보다도 더 높은 명사 인식 능력을 보였다. 문법 기능을 하는 복합형태소들은 도메인에 따라 그 종류가 잘 바뀌지 않기 때문에 제안한 방법은 다양한 도메인에서도 좋은 성능을 보인다.

4 장에서 미등록단어 문제를 해결하며 단일 주제로 이뤄진 문서 집합을 요약하는 키워드, 핵심 문장 추출 방법을 제안하였다. 문서 집합을 요약하기 위하여 키워드가 이용될 수 있는데, 토크나이징 단계에서 중요한 단어가 제대로 인식되지 않는다면 잘못된 문서 요약 결과가 만들어진다. 키워드를 이용한 문서 요약 과업에서는 다른 단어보다도 키워드에 대한 정확한 인식이 우선되어야 한다. 이를 위하여 L + R 구조와 그래프 랭킹 알고리즘을 이용하여 토크나이저를 이용하지 않으며 키워드를 직접 추출하는 방법을 제안하였다. 또한 추출된 키워드를 이용하여 핵심 문장을 선택하는 방법도 함께 제안하였다. 이 방법에는 핵심 문장의 다양성을 유도하는 방법도 포함되어 있다. 제안한 방법은 토크나이저를 이용하지 않으면서 높은 정확도로 키워드를 인식하며, 다양한 내용의 문장들로 핵심 문장을 구성할 수 있다.

5 장에서 다양한 주제로 이뤄진 문서 집합을 요약하기 위한 군집화 기반 키워드 추출 방법을 제안하였다. 문서 집합의 크기가 클 경우에도 효율적인 군집화 학습이 가능하도록 효율적으로 작동하는 Spherical k-means 초기화 알고리즘을 제안하였으며,

이는 기존의 초기화 알고리즘인 k-means++ 보다 수백배에서 수천배의 학습 속도 향상을 보여줬다. 하지만 각 군집마다 문서 요약을 하기 위해서는 추가적인 모델 학습이 필요하다. 이를 해결하기 위하여 군집 중심값을 이용하여 키워드를 추출하는 방법을 제안하였다. 여러 주제로 이뤄진 문서 집합은 각 주제를 구분하며 한 주제를 대표할 수 있는 단어를 키워드로 선별해야 한다. 군집 중심값을 이용한 간단한 연산만으로 이러한 단어를 선택하는 방법을 제안하였으며, 군집 별 키워드 추출 결과의 정성 분석을 통하여 제안한 방법이 효과적임을 확인하였다.

6 장에서 시계열 형식의 문서 집합을 주제로 나뉘어진 구간 별로 요약하는 방법을 제안하였다. 문서 집합은 생성 시점이 변화함에 따라 이를 구성하는 주제들도 변화한다. 그리고 문서의 생성 시점 정보는 주제의 변화를 파악하는데 유용한 정보이다. 그러나 시점 별 주제에 대한 정보를 획득하기 어렵다. 이러한 문제를 해결하기 위하여 거리 기반 시계열 구간 분리 방법을 적용하여 트렌드가 급변하는 구간을 탐색한 뒤 각 구간의 문서 집합을 요약하는 비지도학습 기반 방법을 제안하였다. 제안된 방법은 2 장과 3 장의 단어 추출 방법과 토크나이저를 이용하여 각 문서를 질 좋은 벡터로 표현한 뒤, 시계열 구간 분리 방법을 이용하여 문서 집합을 주제가 비슷한 구간으로 나눈다. 각 구간은 4 장과 5 장의 키워드 및 핵심 문장 추출 방법을 이용하여 요약된다. 제안된 방법은 약 7 여년의 기간동안 발생한 정치 도메인의 뉴스 문서의 요약에 적용되었다. 그 결과 각 정치인 별 큰 트렌드를 중심으로 구간이 나뉘며, 해당 구간을 잘 대표하는 키워드와 핵심 문장으로 요약이 됨을 정성적으로 확인하였다.

7.2 후속 연구

위에서 다룬 토크나이징 과업과 문서 요약 과업에는 지도학습 기반 방법들도 이용된다. 지도학습 방법은 학습 데이터와 이를 이용하는 모델의 구조에 의하여 풀 수 있는

문제의 범위가 한정적이고 학습데이터에 편향된 분석을 수행하지만, 학습된 범위의 문제에 대해서는 높은 분석 성능을 보여준다. 반면 비지도학습 방법은 학습 데이터에 대한 의존성을 낮기 때문에 다양한 종류의 데이터에서 지도학습 방법보다 높은 적응력을 보여준다. 이 논문에서 제안한 방법들은 비지도학습 방식의 접근만으로도 학습 데이터를 이용하는 모델과 비슷하거나 더 좋은 성능을 보여주었다. 비지도학습 방법과 지도학습 방법은 서로 다른 정보를 이용하여 문제를 해결하기 때문에 서로 상호 보완적으로 이용될 수 있다. 그렇기 때문에 비지도학습으로 자연어처리 문제를 해결하려는 연구가 발전되어야 한다. 하지만 이 논문에서는 제안한 비지도학습 방법들이 어떻게 지도학습 방법들과 보완적으로 이용될 수 있을지에 대한 논의를 하지 않았다.

학습 말뭉치로부터 사전과 확률 모델을 학습한 토크나이저는 학습 말뭉치와 비슷한 문서 집합에 등장한 알려진 단어에 대해서는 정확한 처리가 가능하지만, 미등록단어를 제대로 인식하지 못하거나 모호한 상황에서는 학습 말뭉치에 편향된 확률 모델을 이용하여 이를 해결한다. 2 장에서 토크나이저가 적용되는 도메인에서의 단어 경계를 인식할 수 있는 방법과 3 장에서 주어진 데이터에 등장하는 명사들을 추출하는 방법이 지도기반 토크나이저에 결합될 경우, 미등록단어 문제를 완화하며 분석할 데이터의 도메인에 맞춰 모호성을 해결할 수 있다.

문서 요약 과업에서도 4 장과 5 장에서 제안된 방법과 지도학습 방법이 보완적으로 이용될 수 있다. 뉴럴 네트워크를 이용하는 지도학습 문서 요약 방법은 학습 데이터를 이용하여 적합한 요약 문장에 대한 기준을 학습하는데, 도메인별로 동일한 기준의 적용되지 않을 경우 학습 데이터에 대한 편향성이 발생한다. 이러한 문제를 완화하기 위하여 4 장과 5 장에서 제안된 방법을 문서 요약에 적합한 키워드와 핵심 문장의 기준으로 이용할 수 있다. 이에 대한 연구를 통하여 지도학습 문서 요약 방법의 학습 데이터에 대한 의존도를 낮출 것으로 기대한다.

참고 문헌

- [1] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Osama Ahmad Alomari. Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. *Expert Systems with Applications*, 84:24–36, 2017.
- [2] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.
- [3] Chris Alberti, David Weiss, Greg Coppola, and Slav Petrov. Improved transition-based parsing and tagging with neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1354–1359, 2015.
- [4] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.
- [5] Hélio Almeida, Dorgival Guedes, Wagner Meira, and Mohammed J Zaki. Is there a best quality metric for graph clusters? In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 44–59. Springer, 2011.

- [6] Loulwah AlSumait, Daniel Barbar, and Carlotta Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *2008 eighth IEEE international conference on data mining*, pages 3–12. IEEE, 2008.
- [7] Loulwah AlSumait, Daniel Barbar, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. *Machine Learning and Knowledge Discovery in Databases*, pages 67–82, 2009.
- [8] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*, 2016.
- [9] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [10] Olivier Bachem, Mario Lucic, Hamed Hassani, and Andreas Krause. Fast and provably good seedings for k-means. In *Advances in Neural Information Processing Systems*, pages 55–63, 2016.
- [11] Adil M Bagirov. Modified global k-means algorithm for minimum sum-of-squares clustering problems. *Pattern Recognition*, 41(10):3192–3199, 2008.

- [12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [13] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.
- [14] Ramnath Balasubramanyan and William W Cohen. Block-lda: Jointly modeling entity-annotated text and entity-entity links. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 450–461. SIAM, 2011.
- [15] Miguel Ballesteros, Chris Dyer, and Noah A Smith. Improved transition-based parsing by modeling characters instead of words with lstms. *arXiv preprint arXiv:1508.00657*, 2015.
- [16] Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [17] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606*, 2016.
- [18] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

- [19] Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. Abstractive multi-document summarization via phrase selection and merging. *arXiv preprint arXiv:1506.01597*, 2015.
- [20] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.
- [21] Jonathan Bischof and Edoardo M Airoldi. Summarizing topical content with word frequency and exclusivity. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 201–208, 2012.
- [22] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.
- [23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [24] Bernd Bohnet and Joakim Nivre. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465. Association for Computational Linguistics, 2012.

- [25] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
- [26] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [27] Thorsten Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics, 2000.
- [28] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.
- [29] Ryan P Browne and Paul D McNicholas. Model-based clustering, classification, and discriminant analysis of data with mixed type. *Journal of Statistical Planning and Inference*, 142(11):2976–2984, 2012.
- [30] Christian Buchta, Martin Kober, Ingo Feinerer, and Kurt Hornik. Spherical k-means clustering. *Journal of Statistical Software*, 50(10):1–22, 2012.
- [31] Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. Fast and accurate neural word segmentation for chinese. *arXiv preprint arXiv:1704.07047*, 2017.

- [32] Marco Capó, Aritz Pérez, and Jose A Lozano. An efficient approximation to the k-means clustering for massive data. *Knowledge-Based Systems*, 117:56–69, 2017.
- [33] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388. ACM, 2002.
- [34] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.
- [35] Songjian Chen, Yabo Xu, and Huiyou Chang. A simple and effective unsupervised word segmentation approach. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
- [36] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*, 2016.
- [37] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.
- [38] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

- [39] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics, 2005.
- [40] Jason Chuang, Sonal Gupta, Christopher Manning, and Jeffrey Heer. Topic model diagnostics: Assessing domain relevance via topical alignment. In *Proceedings of the 30th International Conference on machine learning (ICML-13)*, pages 612–620, 2013.
- [41] Jason Chuang, Christopher D Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the international working conference on advanced visual interfaces*, pages 74–77. ACM, 2012.
- [42] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 443–452. ACM, 2012.
- [43] Tagyoung Chung and Daniel Gildea. Unsupervised tokenization for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 718–726. Association for Computational Linguistics, 2009.

- [44] Aaron Clauset, Mark EJ Newman, and Christopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
- [45] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.
- [46] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pages 561–580. Springer, 2012.
- [47] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.
- [48] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [49] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.
- [50] William M Darling. A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 642–647, 2011.

- [51] Anjali R Deshpande and LMRJ Lobo. Text summarization using clustering technique. *International Journal of Engineering Trends and Technology*, 4(8):3348–3351, 2013.
- [52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [53] Inderjit S Dhillon, Yuqiang Guan, and Jacob Kogan. Iterative clustering of high dimensional text data augmented by local search. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 131–138. IEEE, 2002.
- [54] Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1-2):143–175, 2001.
- [55] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 536–544. Association for Computational Linguistics, 2012.
- [56] Weicong Ding, Mohammad Hossein Rohban, Prakash Ishwar, and Venkatesh Saligrama. Topic discovery through data dependent and random projections. In *International Conference on Machine Learning*, pages 1202–1210, 2013.
- [57] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recur-

- rent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.
- [58] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1 -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.
- [59] Güneş Erkan and Dragomir R Radev. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 365–371, 2004.
- [60] Gunes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.
- [61] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [62] Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93, 2004.
- [63] Haodi Feng, Kang Chen, Chunyu Kit, and Xiaotie Deng. Unsupervised segmentation of chinese corpus using accessor variety. In *International Conference on Natural Language Processing*, pages 694–703. Springer, 2004.

- [64] Katja Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics, 2010.
- [65] Katja Filippova and Michael Strube. Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 177–185. Association for Computational Linguistics, 2008.
- [66] Jenny Rose Finkel, Alex Kleeman, and Christopher D Manning. Efficient, feature-based, conditional random field parsing. *Proceedings of ACL-08: HLT*, pages 959–967, 2008.
- [67] Jonathan G Fiscus and George R Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking*, pages 17–31. Springer, 2002.
- [68] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, volume 1, pages 452–455. IEEE, 2000.
- [69] Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2010.

- [70] Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 40–48. Association for Computational Linguistics, 2000.
- [71] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer, 2005.
- [72] Brynjar Gretarsson, John O’donovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):23, 2012.
- [73] Mourad Gridach. Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91, 2017.
- [74] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.
- [75] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.
- [76] David Hall, Daniel Jurafsky, and Christopher D Manning. Studying the history of ideas using topic models. In *Proceedings of the conference on empirical*

- methods in natural language processing*, pages 363–371. Association for Computational Linguistics, 2008.
- [77] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [78] Kaiming He, Fang Wen, and Jian Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2938–2945, 2013.
- [79] Christian Hennig and Tim F Liao. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):309–369, 2013.
- [80] Daniel Hewlett and Paul Cohen. Fully unsupervised word segmentation with bve and mdl. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 540–545. Association for Computational Linguistics, 2011.
- [81] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [82] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

- [83] Gumwon Hong and Hae-Chang Rim. Korean spacing by improving viterbi segmentation. In *Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on*, pages 75–80. IEEE, 2007.
- [84] Jeen-Pyo Hong and Jeong-Won Cha. A new korean morphological analyzer using eojeol pattern dictionary. In *Proceedings of the Korean Information Science Society Conference*. Korean Institute of Information Scientists and Engineers, 2008.
- [85] Anna Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56, 2008.
- [86] Jiajia Huang, Min Peng, Hua Wang, Jinli Cao, Wang Gao, and Xiuzhen Zhang. A probabilistic method for emerging topic tracking in microblog stream. *World Wide Web*, 20(2):325–350, 2017.
- [87] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.
- [88] Zhihui Jin and Kumiko Tanaka-Ishii. Unsupervised segmentation of chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 428–435. Association for Computational Linguistics, 2006.

- [89] Zhongming Jin, Cheng Li, Yue Lin, and Deng Cai. Density sensitive hashing. *IEEE transactions on cybernetics*, 44(8):1362–1371, 2013.
- [90] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
- [91] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.
- [92] Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London, 2014.
- [93] Mi-young Kang, Sung-won Jung, and Hyuk-chul Kwon. Category-pattern-based korean word-spacing. *Lecture notes in computer science*, 4285:288, 2006.
- [94] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *arXiv preprint arXiv:1801.04503*, 2018.
- [95] H Kim. Cleansing noisy text using corpus extraction and string match. *Master's Thesis, Seoul National University*, 2013.
- [96] Han Kyul Kim, Hyunjoong Kim, and Sungzoon Cho. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266:336–352, 2017.
- [97] Hung-Gyu Kim and Beom-Mo Kang. 21st century sejong project-compiling korean corpora. In *Proceedings of the 19th International Conference on Computer Processing of Oriental Languages*, 2001.

- [98] Hyun-Joong Kim, Sungzoon Cho, and Pilsung Kang. Kr-wordrank: An unsupervised korean word extraction method based on wordrank. *Journal of Korean Institute of Industrial Engineers*, 40(1):18–33, 2014.
- [99] Hyunjoong Kim, Han Kyul Kim, Misuk Kim, Jooseoung Park, Sungzoon Cho, Keyng Bin Im, and Chang Ryeol Ryu. Representation learning for unsupervised heterogeneous multivariate time series segmentation and its application. *Computers & Industrial Engineering*, 130:272–281, 2019.
- [100] Seokhwan Kim, Rafael Banchs, and Haizhou Li. Exploring convolutional and recurrent neural networks in sequential labelling for dialogue topic tracking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 963–973, 2016.
- [101] Yoon Kim. Convolutional neural networks for sentence classification. 2014.
- [102] Chunyu Kityz and Yorick Wilksz. Unsupervised learning of word boundary with description length gain. In *Proceedings of the CoNLL99 ACL Workshop. Bergen, Norway: Association for Computational Linguistics*, pages 1–6, 1999.
- [103] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.
- [104] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.
- [105] Anders Krogh. Hidden markov models for labeled sequences. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3*

Conference C: Signal Processing (Cat. No. 94CH3440-5), volume 2, pages 140–144. IEEE, 1994.

- [106] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.
- [107] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [108] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
- [109] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.
- [110] Christine Largeron, Christophe Moulin, and Mathias Géry. Entropy based feature selection for text categorization. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 924–928. ACM, 2011.
- [111] Jey Han Lau, Nigel Collier, and Timothy Baldwin. On-line trend analysis with topic models. twitter trends detection topic model online. *Proceedings of COLING 2012*, pages 1519–1534, 2012.

- [112] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.
- [113] Changki Lee and Hyunki Kim. Automatic korean word spacing using pegasos algorithm. *Information Processing & Management*, 49(1):370–379, 2013.
- [114] Changki Lee, Junseok Kim, Jeonghee Kim, and Hyunki Kim. Joint models for korean word spacing and pos tagging using structural svm. *Journal of KIISE: Software and Applications*, 40(12):826–832, 2013.
- [115] Chanhee Lee, Seolwha Lee, Kuekyeng Kim, and Heuiseok Lim. A data-driven spacing correction system using character-level unidirectional lstm. *Proceedings of the Korean Information Science Society Conference*, pages 660–662, 2017.
- [116] Do-Gil Lee, Sang-Zoo Lee, and Hae-Chang Rim. An efficient method for korean noun extraction using noun patterns. *Journal of KIISE: Software and Applications*, 30(1):173–173, 2003.
- [117] Do-Gil Lee, Sang-Zoo Lee, Hae-Chang Rim, and Heui-Seok Lim. Automatic word spacing using hidden markov model for refining korean text corpora. In *Proceedings of the 3rd workshop on Asian language resources and international standardization-Volume 12*, pages 1–7. Association for Computational Linguistics, 2002.

- [118] Do-Gil Lee, Hae-Chang Rim, and Dongsuk Yook. Automatic word spacing using probabilistic models based on character n-grams. *IEEE Intelligent Systems*, 22(1), 2007.
- [119] Hyeyoung Lee, Jong-seok Lee, Byeong-do Kang, and Seung-weon Yang. Functional expansion of morphological analyzer based on longest phrase matching for efficient korean parsing. *Journal of Digital Contents Society*, 17(3):203–210, 2016.
- [120] Sungjick Lee and Han-joon Kim. News keyword extraction for topic tracking. In *2008 Fourth International Conference on Networked Computing and Advanced Information Management*, volume 2, pages 554–559. IEEE, 2008.
- [121] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.
- [122] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.
- [123] Joshua Lewis, Margareta Ackerman, and Virginia de Sa. Human cluster evaluation and formal quality measures: A comparative study. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34, 2012.
- [124] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

- [125] You Li, Kaiyong Zhao, Xiaowen Chu, and Jiming Liu. Speeding up k-means algorithm by gpus. *Journal of Computer and System Sciences*, 79(2):216–229, 2013.
- [126] Jiye Liang, Liang Bai, Chuangyin Dang, and Fuyuan Cao. The k -means-type algorithms versus imbalanced data distributions. *IEEE Transactions on Fuzzy Systems*, 20(4):728–745, 2012.
- [127] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.
- [128] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.
- [129] Chin-Yew Lin and Eduard Hovy. From single to multi-document summarization. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, 2002.
- [130] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.
- [131] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [132] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*, pages 665–672. ACM, 2009.

- [133] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [134] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [135] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.
- [136] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [137] Pierre Magistry and Benoît Sagot. Can mdl improve unsupervised chinese word segmentation? In *Sixth International Joint Conference on Natural Language Processing: Sighan workshop*, page 2, 2013.
- [138] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598, 2000.
- [139] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics, 2003.

- [140] Andrew McCallum and Ben Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Advances in neural information processing systems*, pages 905–912, 2005.
- [141] Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*, pages 557–564. Springer, 2007.
- [142] Brian McFee and Dan Ellis. Analyzing song structure with spectral clustering. In *ISMIR*, pages 405–410, 2014.
- [143] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.
- [144] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.
- [145] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [146] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

- [147] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*, pages 262–272. Association for Computational Linguistics, 2011.
- [148] Einat Minkov, Richard C Wang, and William W Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005.
- [149] Kaixiang Mo, Yu Zhang, Shuangyin Li, Jiajun Li, and Qiang Yang. Personalizing a dialogue system with transfer reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [150] Christopher E Moody. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*, 2016.
- [151] Seung-Hoon Na, Seong-Il Yang, Chang-Hyun Kim, Oh-Woog Kwon, and Young-Kil Kim. Crfs for korean morpheme segmentation and pos tagging. In *Proc. of 24th Annual Conference on Human and Cognitive Language Technology*, pages 12–15, 2012.
- [152] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

- [153] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*, 2018.
- [154] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 215–224. ACM, 2010.
- [155] Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.
- [156] Aytuğ Onan, Serdar Korukoğlu, and Hasan Bulut. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57:232–247, 2016.
- [157] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.
- [158] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.
- [159] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing- Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

- [160] Eunjeong L Park and Sungzoon Cho. Konlp: Korean natural language processing in python. In *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*, pages 133–36, 2014.
- [161] Daraksha Parveen, Hans-Martin Ramsler, and Michael Strube. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1954, 2015.
- [162] Jouni Paulus and Anssi Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 59–68. ACM, 2006.
- [163] Jouni Paulus, Meinard Müller, and Anssi Klapuri. State of the art report: Audio-based music structure analysis. In *ISMIR*, pages 625–636. Utrecht, 2010.
- [164] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [165] Alexandrin Popescul and Lyle H Ungar. Automatic labeling of document clusters. *Unpublished manuscript, available at <http://citeseer.nj.nec.com/popescul00automatic.html>*, 2000.
- [166] Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D. Manning, and Daniel A. McFarland. Topic modeling for the social sciences. In *NIPS*

2009 Workshop on Applications for Topic Models: Text and Beyond, Whistler, Canada, December 2009.

- [167] Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D Manning, and Daniel A McFarland. Topic modeling for the social sciences. In *NIPS 2009 workshop on applications for topic models: text and beyond*, volume 5, page 27, 2009.
- [168] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics, 2011.
- [169] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- [170] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [171] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.
- [172] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

- [173] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.
- [174] Sunita Sarawagi and William W Cohen. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192, 2005.
- [175] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.
- [176] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.
- [177] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- [178] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.
- [179] Saeed Shahrivari and Saeed Jalili. Single-pass and linear-time k-means clustering based on mapreduce. *Information Systems*, 60:1–12, 2016.

- [180] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.
- [181] Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. Document summarization using conditional random fields. In *IJCAI*, volume 7, pages 2862–2867, 2007.
- [182] Xiaobo Shen, Weiwei Liu, Ivor Tsang, Fumin Shen, and Quan-Sen Sun. Compressed k-means for large-scale clustering. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [183] Yanxin Shi and Mengqiu Wang. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In *IJcAI*, pages 1707–1712, 2007.
- [184] Yusuke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University, 1999.
- [185] Kwang-Seob Shim. Made: Morphological analyzer development environment. *Journal of Internet Computing and Services*, 8(4):159–171, 2007.
- [186] Kwang-Seob Shim and Jae-Hyung Yang. High speed korean morphological analysis based on adjacency condition check. *Journal of KIISE: Software and Applications*, 31(1):89–99, 2004.

- [187] Kwangseob Shim. Automatic word spacing based on conditional random fields. *Korean Journal of Cognitive Science*, 22(2):217–233, 2011.
- [188] Kwangseob Shim. Cloning of korean morphological analyzers using pre-analyzed eojeol dictionary and syllable-based probabilistic model. *KIISE Transactions on Computing Practices*, 22(3):119–126, 2016.
- [189] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973.
- [190] Carson Sievert and Kenneth E Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.
- [191] Satinder P Singh, Michael J Kearns, Diane J Litman, and Marilyn A Walker. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*, pages 956–962, 2000.
- [192] Justin Snyder, Rebecca Knowles, Mark Dredze, Matthew Gormley, and Travis Wolfe. Topic models and metadata for visualizing text corpora. *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 5–9, 2013.
- [193] Yangqiu Song, Shimei Pan, Shixia Liu, Michelle X Zhou, and Weihong Qian. Topic and keyword re-ranking for lda-based topic modeling. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1757–1760. ACM, 2009.

- [194] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.
- [195] Weiwei Sun and Jia Xu. Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 970–979. Association for Computational Linguistics, 2011.
- [196] Narayanan Sundaram, Aizana Turmukhametova, Nadathur Satish, Todd Mostak, Piotr Indyk, Samuel Madden, and Pradeep Dubey. Streaming similarity search over one billion tweets using parallel locality-sensitive hashing. *Proceedings of the VLDB Endowment*, 6(14):1930–1941, 2013.
- [197] Matt Taddy. On estimation and selection for topic models. In *International Conference on Artificial Intelligence and Statistics*, pages 1184–1193, 2012.
- [198] Yusuke Takahashi, Takehito Utsuro, Masaharu Yoshioka, Noriko Kando, Tomohiro Fukuhara, Hiroshi Nakagawa, and Yoji Kiyota. Applying a burst model to detect bursty topics in a topic model. In *International Conference on NLP*, pages 239–249. Springer, 2012.
- [199] Ben Taskar, Dan Klein, Mike Collins, Daphne Koller, and Christopher Manning. Max-margin parsing. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.
- [200] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Pro-*

- ceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*, pages 173–180. Association for Computational Linguistics, 2003.
- [201] Ioannis Tschantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484, 2005.
- [202] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.
- [203] Shiva Twinandilla, Satriyo Adhy, Bayu Surarso, and Retno Kusumaningrum. Multi-document summarization using k-means and latent dirichlet allocation (lda)–significance sentences. *Procedia Computer Science*, 135:663–670, 2018.
- [204] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.
- [205] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [206] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112. ACM, 2009.

- [207] Xiaojun Wan and Jianwu Yang. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2008.
- [208] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.
- [209] Chong Wang, John Paisley, and David Blei. Online variational inference for the hierarchical dirichlet process. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 752–760, 2011.
- [210] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics, 2012.
- [211] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.
- [212] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

- [213] Dominik Wurzer, Victor Lavrenko, and Miles Osborne. Tracking unbounded topic streams. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1765–1773, 2015.
- [214] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.
- [215] Pengtao Xie and Eric P Xing. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874*, 2013.
- [216] Hui Xiong, Junjie Wu, and Jian Chen. K-means clustering versus validation measures: a data-distribution perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):318–331, 2008.
- [217] Hui Xiong, Junjie Wu, and Jian Chen. K-means clustering versus validation measures: a data-distribution perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):318–331, 2009.
- [218] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.
- [219] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

- [220] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.
- [221] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3861–3870. JMLR. org, 2017.
- [222] Xintian Yang, Amol Ghoting, Yiye Ruan, and Srinivasan Parthasarathy. A framework for summarizing and analyzing twitter feeds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 370–378. ACM, 2012.
- [223] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.
- [224] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. Recent advances in document summarization. *Knowledge and Information Systems*, 53(2):297–336, 2017.
- [225] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 103–111. ACM, 2017.

Wide Web, pages 351–360. International World Wide Web Conferences Steering Committee, 2017.

- [226] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1351–1361. International World Wide Web Conferences Steering Committee, 2015.
- [227] Ke Zhai and Jordan Boyd-Graber. Online latent dirichlet allocation with infinite vocabulary. In *International Conference on Machine Learning*, pages 561–569, 2013.
- [228] Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. Self-taught hashing for fast similarity search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25. ACM, 2010.
- [229] Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. Keyword extraction using support vector machine. In *International Conference on Web-Age Information Management*, pages 85–96. Springer, 2006.
- [230] Meishan Zhang, Yue Zhang, and Guohong Fu. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 421–431, 2016.

- [231] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.
- [232] Yue Zhang and Stephen Clark. Joint word segmentation and pos tagging using a single perceptron. *Proceedings of ACL-08: HLT*, pages 888–896, 2008.
- [233] Hai Zhao, Chang-Ning Huang, and Mu Li. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165, 2006.
- [234] Hai Zhao and Chunyu Kit. Incorporating global information into supervised learning for chinese word segmentation. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 66–74. Citeseer, 2007.
- [235] Hai Zhao and Chunyu Kit. Exploiting unlabeled text with different unsupervised segmentation criteria for chinese word segmentation. *Research in Computing Science*, 33:93–104, 2008.
- [236] Hai Zhao and Chunyu Kit. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *IJCNLP*, pages 106–111, 2008.
- [237] Hai Zhao and Chunyu Kit. Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Sciences*, 181(1):163–183, 2011.

- [238] Ying Zhao, George Karypis, and Usama Fayyad. Hierarchical clustering algorithms for document datasets. *Data mining and knowledge discovery*, 10(2):141–168, 2005.
- [239] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, 2013.
- [240] Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. An efficient algorithm for unsupervised word segmentation with branching entropy and mdl. *Information and Media Technologies*, 8(2):514–527, 2013.
- [241] 강승식. 음절 특성을 이용한 한국어 불규칙 용언의 형태소 분석. *정보과학회논문지 (B)*, 22(10):1480–1487, 1995.
- [242] 김선우 and 최성필. Bidirectional lstm-crf 기반의 음절 단위 한국어 품사 태깅 및 띠어쓰기 통합 모델 연구. *정보과학회논문지*, 45(8):792–800, 2018.
- [243] 심광섭. 형태소 분석기 사용을 배제한 음절 단위의 한국어 품사 태깅. *인지과학*, 22(3):327–345, 2011.
- [244] 심광섭. 음절 단위의 한국어 품사 태깅에서 원형 복원. *정보과학회논문지: 소프트웨어 및 응용*, 40(3):182–189, 2013.
- [245] 심광섭. 한국어 형태소 분석을 위한 음절 단위 확률 모델. *정보과학회논문지*, 41(9):642–651, 2014.
- [246] 심광섭. 기분석 어절 사전과 음절 단위의 확률 모델을 이용한 한국어 형태소 분석기 복제. *정보과학회 컴퓨팅의 실제 논문지*, 22(3):119–126, 2016.

- [247] 양승현 and 김영섭. 부분 어절의 기분석에 기반한 고속 한국어 형태소 분석 방법. *정보과학회논문지: 소프트웨어 및 응용*, 27(3):290–301, 2000.
- [248] 이건일, 이의현, and 이종혁. Sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 품사 태깅. *한국정보과학회 학술발표논문집*, pages 693–695, 2016.
- [249] 이동주, 연종흠, and 이상구. 한국어 문장의 띄어 쓰기 오류 교정과 최적 형태소 분석을 위한 통합 확률 모델. *한국정보과학회 학술발표논문집*, 38(1A):237–240, 2011.
- [250] 이충희, 임준호, 임수종, and 김현기. 기분석사전과 기계학습 방법을 결합한 음절 단위 한국어 품사 태깅. *정보과학회논문지*, 43(3):362–369, 2016.
- [251] 최재혁 and 이상조. 양방향 최장일치법에 의한 한국어 형태소 분석기에서의 사전 검색 횟수 감소 방안. (구) *정보과학회논문지*, 20(10):1497–1507, 1993.
- [252] 카카오. khaiii: Kakao hangul analyzer. <https://github.com/kakao/khaiii>, 2018.

Abstract

Unsupervised Korean Tokenizer and Extractive Document Summarization to Solve Out-of-Vocabulary and Dearth of Data

Hyunjoong Kim

Department of Industrial Engineering

The Graduate School

Seoul National University

Natural language processing is interested in converting a human language into computerusable information to solve real problems. Among its various sub-tasks, tokenization is a fundamental data pre-processing task that aims to detect words or morphemes from its input sentence. Therefore, ineffective tokenization degrades the quality of other subsequent natural language processing tasks such as document summarization, in which documents are condensed into several key words or sentences.

All of these various natural language processing tasks share common intrinsic challenges such as out-of-vocabulary problem, lack of labeled training data and inevitable grammatical or spelling errors within the input texts. Among these various issues, grammatical or spelling errors are especially fatal for natural language

processing for Korean. As the readability of the Korean texts are not significantly affected by these errors, these errors are simply overlooked in the training data, thereby causing tokenization to be increasingly difficult. Furthermore, these overlooked grammatical or spelling errors are one of the main reasons behind the out-of-vocabulary problem in Korean. Unfortunately, there is simply not enough training data to resolve these errors.

In this paper, I propose various unsupervised Korean natural language processing methods to overcome these issues. Due to its unsupervised nature, it can be easily applied to various domains that lack labeled training data. Instead of relying on labeled data, I utilize the structures of Eojeol, a basic unit of a Korean word, as a prior knowledge for capturing the patterns in the Korean language.

Based on dividing Eojeol into a L + [R] structure, I propose four new unsupervised natural language processing methods for Korean. To overcome the out-of-vocabulary problem, I create two Eojeol based unsupervised Korean tokenizers that not only outperforms Word Piece Model but also performs at a similar level as supervised Korean tokenizers that are trained on labeled datasets and dictionaries. Furthermore, I devise a novel noun extraction method that is superior than the trained Korean morpheme analyzers. Beside these pre-processing tasks, I also propose key words and sentences selection method that removes redundant sentences and summarizes a set of Korean documents without relying on any tokenizer. For summarizing a document set that consists of documents with various topics, I additionally propose an improved document clustering method and effective cluster labeling method. It is up to several thousand times than existing k-means clustering

algorithm during initialization. Finally, I propose summarization method for time series formed document set. It first devides the data it into several segments based on time point of topic change, then the keyword and keysentence extraction methods proposed in previous chapter are applied to summarize each segment. Our improved clustering and time-series formed documents summarization methods can be applied to different languages not only Korean.

Supervised machine learning approaches are inevitable in natural language processing. However, unsupervised methods offer additional insight into Korean language that supervised methods fail to capture. As our unsupervised methods perform at a similar level as their supervised counterparts, integrating both approaches will provide unprecedented improvement in the world of Korean natural language processing.

Keywords: Korean natural language processing, Unsupervised tokenizer, Noun extraction, Keyword extraction, Key-sentence extraction, Document clustering, Clustering labeling

Student Number: 2013-30314

감사의 글

이십대의 후반에 박사 과정을 시작하여 삼십대의 중반에 작은 논문을 완성하였습니다. 그시간 동안 곁에서 가르침과 응원을 주신 분들께 감사의 글을 올립니다.

박사 과정 동안 얻은 것 중 가장 값진 것은 지식이 아닌 지식을 대하는 태도였습니다. 배움은 남을 위한 것이어야 하며 작은 것이라 하더라도 가진 지식과 기술로 사회의 문제를 해결하는데 기여해야 함을 알려준 이에게 감사를 드립니다. 그리고 그 배움의 방향이 때로는 지적인 호기심이어도 충분하며, 그 성과가 작고 이루는데 오래 걸리더라도 괜찮다고 응원해 주신 분들께 감사를 드립니다. 한 명의 연구자가 되기 위해서는 연구의 철학을 가져라 조언해주고, 하나의 졸업 논문은 이후에 같은 고민을 하는 이들의 시작점이 되어야 한다고 중심을 잡아주신 분들께 감사를 드립니다.

나의 배움에 시간과 애정을 할애해준 모든 이들에게 감사를 드립니다. 세미나, 발표, 수업에서 완성되지 않은 저의 이야기를 들어주고, 의문과 질문으로 저의 이야기를 좀 더 완성된 형태로 다듬어 주셔서 감사합니다. 그리고 많은 이들과 이야기를 할 수 있는 기회들을 주셔서 감사합니다. 특히 오랜 시간 마주 앉아 여러 주제의 토론을 함께 해준 친구들에게 감사를 드립니다.

곁에서 삶을 응원해 준 분들께 감사를 드립니다. 배움 중 하나는 앞으로 어떻게 살아야 하는지에 대한 기준이었습니다. 기쁜 순간에도 힘든 순간에도 함께 해준 친구들에게, 올바른 선택들을 할 수 있게 이끌어준 이들에게 감사를 드립니다. 응원과 애정을 주고 받은 모든 이들에게 감사를 드립니다.

덕분에 지금까지의 고민들이 쌓여 작은 논문과 작업들이 만들어 졌습니다. 앞으로도 부족한 부분들을 채워 다른 이들에게 도움이 되는 작업으로 발전 시키겠습니다.

마지막으로 어떻게 살아야 하는지 늘 삶으로 가르침을 주시는 부모님께 감사를 드립니다.