2023년 2월

석사학위 논문

수사 구조 이론 관계 레이블 기반 문서 구조 유사성 분석에 대한 연구

조선대학교 대학원

컴 퓨 터 공 학 과

서 동 원

수사 구조 이론 관계 레이블 기반 문서 구조 유사성 분석에 대한 연구

A study on the document structure similarity analysis based on Rhetoric Structure Theory Relationship Label

2023년 02월 24일

조선대학교 대학원

컴 퓨 터 공 학 과

서 동 원

수사 구조 이론 관계 레이블 기반 문서 구조 유사성 분석에 대한 연구

지도교수 김 판 구

이 논문을 공학석사학위신청 논문으로 제출함

2022년 10월

조선대학교 대학원

컴 퓨 터 공 학 과

서 동 원

서동원의 석사학위논문을 인준함

위원장 조선대학교 교 수 양희덕 (인)

위 원 조선대학교 교 수 전찬준 (인)

위 원 조선대학교 교 수 김판구 (인)

2022 년 12 월

조선대학교 대학원

목 차

ABSTRACT

서론 1

관련연구 5

A. 텍스트 분석 5

i. 형태소 분석 (Morphological Analysis) 7

ii. 구문 분석 (Syntax Analysis) 12

iii.의미 분석 (Semantic Analysis) 13

iv. 화용 분석 (Pragmatic Analysis) 14

B. 수사구조이론 16

i. RST Label 16

ii. 구문 분석 Parser 20

iii.RST Parser 23

C. 벡터 유사도 분석 25

i. 벡터 공간에서의 유사도 분석: 코사인 유사도 분석 25

ii. 텍스트 벡터화(Text Vectorization) 26

(ㄱ) 원-핫 인코딩(One-hot encoding) 27

(ㄴ) 빈도수 기반 텍스트 벡터화(TF-IDF) 27

(ㄷ) 단어 임베딩(Word Embedding) 28

문서 구조 유사도 분석 시스템 30

A. 문서 구조 유사도 분석 시스템 30
B. 실험 34

결론 39

참고문헌 40

표 목 차

표 1. 데이터의 종류 6
표 2. 한글 세종 품사 태그(왼), KKMA 품사 태그(오) 9
표 3. 영어 품사 태그 11
표 4. 수사구조이론 관계 레이블 종류 19
표 5. 사설 벡터 공간에서의 사설 데이터 분석 결과 35
표 6. 사설 벡터 공간에서의 기사 데이터 분석 결과 36
표 7. 뉴스 기사 벡터 공간에서의 사설 데이터 분석 결과 37
표 8. 뉴스 기사 벡터 공간에서의 기사 데이터 분석 결과 38

그 림 목 차

그림 1. AI 자기소개서 분석기를 활용하는 기업들의 활용 방안 2
그림 2. AI 자기소개서의 분석 내역 예시..... 3
그림 3. 한글 형태소 분석 8
그림 4. 영어 형태소 분석 10
그림 5. 화용 분석의 예시 15
그림 6. RST Tree (S: 핵, N: 위성)..... 17
그림 7. 구문 분석의 하향식 파싱 21
그림 8. 구문 분석의 상향식 파싱 22
그림 9. RST Parser 의 하향식 파싱 24
그림 10. RST Parser 의 상향식 파싱 24
그림 11. One-hot encoding 27
그림 12. Word Embedding 벡터 연산..... 29
그림 13. RST 기반 문서 구조 분석 시스템..... 31
그림 14. RST Parser 의 파싱 결과 32
그림 15. retWeb 에서 생성된 입력 데이터의 RST Tree..... 33

ABSTRACT

A study on the document structure similarity analysis based on Rhetoric Structure Theory Relationship Label

Seo DongWon
Advisor : Prof. Kim. Pan-Koo, Ph.D.
Department of Computer Engineering
Graduate School of Chosun University

The performance has been greatly improved with the development of deep learning technology, which is a text processing technique that mainly used statistical techniques. As a result, more and more companies are actively entering the text processing model into their business. It is mainly used for document analysis such as classification of customer queries and report analysis. more recently, models using document analysis techniques have also begun to be used in the recruitment process. These models significantly increased the job efficiency of the interviewer by quantifying the applicant's resume for each item.

Each document has different characteristics depending on its purpose. For example, a report used by a company or a research institute interprets the contents of an experiment or writes an article in the form of a causal relationship. Conversely, in the case of an opinion, it is possible to interpret a phenomenon, but it is written in the form of asserting and

persuading one's opinion on it. Also, the resume is written in the form of a cover letter. If the texts have the same personality, there are structural commonalities even though the content is different.

In this paper, using the rhetorical structural theory, we try to find structural commonalities that fit the characteristics of these texts, and to figure out how well they are structurally written when new texts are written. The experiment was conducted by analyzing the similarity by creating two text vector spaces for two groups of documents with different types of text.

서론

인공 신경망의 발전에 따른 자연어 처리 분야의 성능 향상은 특정 분야에서 사람과 유사하거나 더 뛰어난 정확도를 보여주기도 한다. 대표적으로 언어번역은 학습을 위한 데이터 확보와 저장공간의 발전, 하드웨어 성능 향상에 힘 입어 과거의 단어 수준, 한 문장 수준에서만 가능했던 번역 수준을 꽤나 긴 문장들의 번역에도 높은 정확도를 보이게 됐다. 그 외에도 문장을 분석해 품사를 분류하거나 중의적인 표현의 분류, 의미론적 문장 해석 등 사람의 직감능력에 해당하는 부분까지 처리가 가능해짐에 따라 그 활용도가 커지고 있다. 기업에서는 고객의 요청 사항을 요약하여 관리자에게 제공하거나, 여러 요청사항들의 내용을 분석하여 유사한 요청사항들끼리 하나의 클래스로 묶어 처리하는 기능 등을 적극적으로 활용하고 있다. 최근에는 인재채용과정에서까지 활용범위를 확대하고 있는데, 지원자의 자기소개서, 이력서 등을 분석하여 어떤 성향의 지원자인지를 대략적으로 알 수 있도록 해주는 분석 AI 자기소개서 분석기들이 등장하고 있기 때문이다.

그림 1. AI 자기소개서 분석기를 활용하는 기업들의 활용 방안

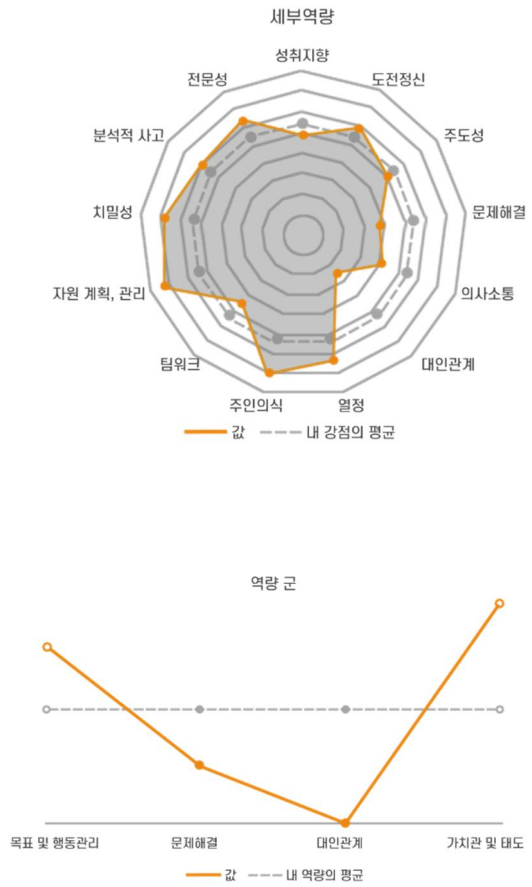


그림 2. AI 자기소개서의 분석 내역 예시

기업에서 사용하는 자기소개서 분석은 각 텍스트에서 사용된 단어들의 의미 분석을 통해 어떤 성격을 가진 문장인지를 확인,분류하여 전체 문서에 대한 성향을 그래프나 도표 등으로 표현하거나, 각 기업에서 생각하는 모범적인 합격자기소개서를 기준 삼아 유사한 표현 등을 얼마나 많이 사용했는지를 분석하여 점수를 매긴다.[6] 즉, 합격자 자기소개서와 지원자 자기소개서 간의 유사성을 분석하는 것인데, 텍스

트 간의 유사성 검사 기법은 두 텍스트를 나타내는 벡터의 위치와 모양을 보고 파악한다. Word2Vec 와 같은 텍스트의 동음이의어, 다의어 등을 잘 파악할 수 있는 워드임베딩 방법론들이 등장하면서 텍스트 간 유사도의 정확성도 크게 향상됐다. 하지만 워드임베딩 기법을 바탕으로 한 문서 간 유사도 분석은 문장 하나하나에 대한 의미적 유사성에 대한 것에 초점을 두고 있기에 문서 전체 구조에 대한 유사성이라고 보기에는 어려움이 있다. 본 논문은 텍스트 간의 구조적 관계를 나타내는 Rhetorical Structure Theory 레이블을 기반으로 두 문서가 구조적으로 얼마나 유사하게 사용되었는지에 대한 연구를 진행한다.

관련 연구

A. 텍스트 분석

데이터는 크게 정형 데이터(Structured data), 비정형 데이터(Unstructured data)로 나눌 수 있다. 정형 데이터는 행과 열을 가진 데이터베이스의 규칙에 맞게 입력된 데이터 중 그 데이터 값 자체가 의미를 가지고 있는 데이터를 말한다. 예로 '남자'와 '여자'을 값으로 가질 수 있는 '성별' 이라는 컬럼은 정형 데이터라고 볼 수 있지만, 웹 페이지 주소, 음성 데이터 등은 주소를 타고 들어가든, 해당 데이터를 가지고 음성으로 변조를 하는 작업이 추가적으로 필요하므로 이 경우는 정형 데이터라 하지 않는다. 비정형 데이터는 규칙 없는 텍스트, 음성, 영상 등의 데이터를 말한다. 규칙이 없기에 의미를 파악하는 것이 정형 데이터에 비해 어렵고 용량 자체도 매우 커 그동안 데이터로써 사용되기 어려웠다. 높은 처리 능력을 가진 CPU 등의 등장과 용량을 확보함에 따른 빅데이터 처리가 가능해짐에 따라 비정형 데이터를 통한 새로운 인사이트를 얻기 시작했다. 정형 데이터와 비정형 데이터의 성격을 둘 다 가지고 있는 반정형 데이터(Semi-structured data)라는 구분도 존재한다. 이는 정형 데이터처럼 고정된 스키마가 없지만 완전한 비정형 데이터는 아닌 데이터로, 태그 및 조직 메타데이터와 같은 구조적 요소도 가지고 있기도 한다. 객체 지향 데이터베이스 등에서 주로 보이는 HTML, XML, 그래프, 이메일을 예로 들 수 있다.

비정형 데이터는 과거에는 처리하기 힘들었으나 기술의 발전으로 인해 처리가 가능해짐에 따라 많은 새로운 인사이트를 얻을 수 있게 됐다. 그 중 특히나 텍스트 데이터는 소설, 수필, 기사, 논문 등 다양한 성격을 가진 수많은 정보를 담은 글들이 매일 같이 쏟아져 오는 상황에 SNS의 등장에 더욱 탄력을 받아 데이터 양이 기하급수적으로 증가하고 있다.

정형 데이터 (Structured data)	
일정한 규칙에 맞게 입력된 값 자체가 의미를 가진 데이터	Ex) 엑셀 데이터, 데이터베이스 등
비정형 데이터 (Unstructured data)	
일정한 규칙이 없고 정리되지 않는 데이터	Ex) 신문기사, 음성, 영상 등
반정형 데이터 (Semi-structured data)	
고정된 스키마는 없지만 구조적 요소는 가진 객체 데이터	Ex) HTML, XML, 그래프 등

표 1. 데이터의 종류

텍스트 데이터를 정제하고 분석하기 위해서는 기계학습, 통계학과 언어학을 기반으로 한 자연어 처리 기술 (Natural Language Processing, NLP)에 대한 이해가 필

요하다. 자연어는 한국어, 영어, 중국어 등의 사람이 의사소통을 위해 사용하는 언어를 말하며 컴퓨터를 사용하여 이를 분석하는 것을 컴퓨터 공학에서는 '자연어 처리(NLP)'라 부르며 언어학에서는 '전산언어학(Computational linguistic)'이라 부른다. NLP에서 자연어를 처리하는 단계로는 크게 아래와 같이 4단계가 있다.

I. 형태소 분석 (Morphological Analysis)

II. 구문 분석 (Syntax Analysis)

III. 의미 분석 (Semantic Analysis)

IV. 화용 분석 (Pragmatic Analysis)

i. 형태소 분석 (Morphological Analysis)

먼저 형태소 분석(Morphological Analysis)단계는 대부분의 자연어 처리 프로세스에서 가장 먼저 하는 작업으로, 세부적으로 단어와 단어를 분리하는 Word Segmentation(단어 분리), 텍스트를 형태소로 분리하는 Morphological Analysis(형태소 분석), 분리된 텍스트들의 품사를 찾는 Parts of speech(POS) Tagging(품사 태깅)으로 볼 수 있다. 하나의 것을 특정 단위로 분리하는 것으로 공학에서는 이러한 작업을 토큰화(Tokenization)이라 한다. 텍스트를 나누기 위해 사용되는 개념을 형태소라 하는데 의미적으로 가장 작은 단위를 뜻한다. 즉, 형태소 분석이란 텍스트를 형태소 단위로 토큰화하는 것이다. 토큰화를 위해서는 단어의 앞뒤에 붙는 접두사, 접미사를 분리해야 하며 영어의 경우는 단어를 원형으로 바꿔줘야 한다. 한

국어는 영어에 비해 상대적으로 이 과정이 어렵다고 하는데 그 이유가 영어에 비해 접두사, 접미사 분리가 까다롭기 때문이다.

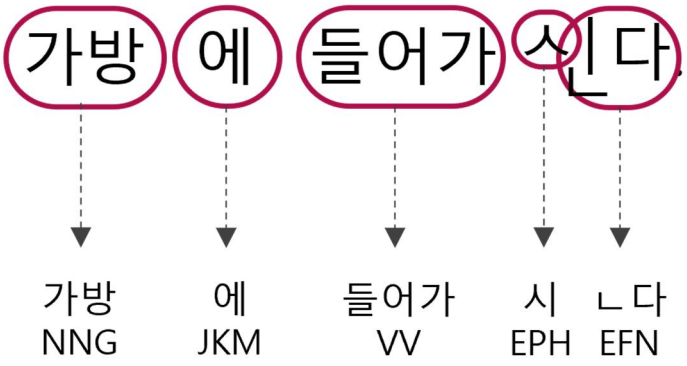


그림 3. 한글 형태소 분석

대분류	세종 품사 태그		KKMA 단일 태그 V 1.0	
	태그	설명	태그	설명
체언	NNG	일반 명사	NNG	보통 명사
	NNP	고유 명사	NNP	고유 명사
	NNB	의존 명사	NNB	일반 의존 명사
			NNM	단위 의존 명사
	NR	수사	NR	수사
	NP	대명사	NP	대명사
용언	VV	동사	VV	동사
	VA	형용사	VA	형용사
	VX	보조 용언	VXV	보조 동사
			VXA	보조 형용사
	VCP	긍정 지정사	VCP	긍정 지정사, 서술격 조사 '이다'
	VCN	부정 지정사	VCN	부정 지정사, 형용사 '아니다'
관형사	MM	관형사	MDT	일반 관형사
			MDN	수 관형사
부사	MAG	일반 부사	MAG	일반 부사
	MAJ	접속 부사	MAC	접속 부사
감탄사	IC	감탄사	IC	감탄사
조사	JKS	주격 조사	JKS	주격 조사
	JKC	보격 조사	JKC	보격 조사
	JKG	관형격 조사	JKG	관형격 조사
	JKO	목적격 조사	JKO	목적격 조사
	JKB	부사격 조사	JKM	부사격 조사
	JKV	호격 조사	JKI	호격 조사
	JKQ	인용격 조사	JKQ	인용격 조사
	JX	보조사	JX	보조사
	JC	접속 조사	JC	접속 조사
선어말 어미	EP	선어말 어미	EPH	존칭 선어말 어미
			EPT	시제 선어말 어미
			EPP	공손 선어말 어미
어말 어미	EF	종결 어미	EFN	평서형 종결 어미
			EFQ	의문형 종결 어미
			EFO	명령형 종결 어미
			EFA	청유형 종결 어미
			EFI	감탄형 종결 어미
			EFR	존칭형 종결 어미

표 2. 한글 세종 품사 태그(왼), KKMA 품사 태그(오)

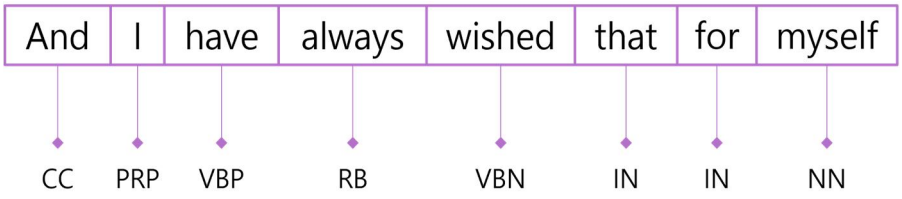


그림 4. 영어 형태소 분석

품사 태그	설명	예시
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	there is
FW	foreign word	les
IN	preposition, subordinating conjunction	in, of, like
IN/THAT	that as subordinator	that
LS	list marker	1)
MD	modal	could, will
NN	noun, singular or mass	table
NNS	noun plural	tables
NP	proper noun, singular	John
NPS	proper noun, plural	Vikings
PDT	predeterminer	both the boys
POS	possessive ending	friend's
PP	personal pronoun	I, he, it
PP$	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
SENT	Sentence-break punctuation	. ! ?

표 3. 영어 품사 태그

ii.구문 분석 (Syntax Analysis)

구문 분석(Syntax Analysis)단계는 텍스트를 이루고 있는 구성 성분들을 분해하여 성분들 사이에 있는 위계 관계를 분석하여 Parsing Tree를 만드는 것을 말한다. 문장의 구조나 기능, 문장의 구성 요소 따위를 연구하는 언어학의 통사론(Syntax) 또는 구문론이 여기에 해당된다. 구문 분석을 통해 문장이 문법적으로 옳게 작성되었는지를 판단할 수 있다.

- Korea is an asian country [O]
- Korea is an country [O]
- Korea is an some asian country [X]

첫 번째 문장 ***Korea is an asian country***를 형태소 분석을 거치면

Korea [***고유 명사, 단수형, NNP***], ***is*** [***동사, 현재형 시제, 3인칭 단수, VBZ***], ***an*** [***한정사, DT***], ***asian*** [***형용사, JJ***], ***country*** [***명사, 단수형, NN***] 로 분석이 가능하다. 구문 분석은 각 형태소에 태그된 품사의 문법 구조가 어긋난 구조인지 여부를 파악한다. 세번째 문장 ***Korea is asian an country*** 은 [***한정사, DT***] 로 태그된 ***an***이 ***asian*** [***형용사, JJ***] 와 ***country*** [***명사, 단수형, NN***] 사이에 위치하고 있다. 언어학적으로 한정사(determiner)는 명사의 의미를 명확하게 또는 불명확하게 하고, 소유관계, 수

량, 순서 등을 나타내며 등장할 수 있는 위치에 따라 맨 앞에 위치하는 '전치 한정사(predeterminer)', 중간에 위치하는 '중간 한정사(central determiner)', 맨 뒤에 나오는 '후치 한정사(postdeterminer)'로 분류되며, 명사를 기준으로 등장하는 위치이기 때문에 형용사의 위치와는 무관하다. ***an*** 과 ***some***은 분류상 중간 한정사에 해당한다. 같은 중간 한정사끼리는 중복하여 사용할 수 없다는 제약이 있다. 의미상으로도 '하나의'와 '약간의'의 의미를 가지고 있어 한 문장에 함께 존재할 수 없다.

언어학의 통사론에서 구문에 대한 분석을 할 때 유의해야할 점을 하나의 문장이 다수의 구조로 해석될 수 있는 성질인 구조적 중의성(Structural Ambiguities)으로 꼽는다. 자연어 처리의 구문 분석에서도 마찬가지로 구조적 중의성에 의한 어려움이 있는데, 다양한 구문 분석기가 같은 문장을 두고 분석을 진행하여도 각자 다른 결과를 출력하는 것에서 확인해 볼 수 있다.

iii.의미 분석 (Semantic Analysis)

의미 분석(Semantic Analysis)단계는 구분 분석 결과로 생성된 통사 구조에 따른 문장의 의미를 해석하는 단계이다. 말이나 글의 의미 또는 뜻을 연구하는 언어학의 의미론(Semantics)이 여기에 해당한다. 의미적으로 문장이 옳은지에 대한 판별을 하며, 중의성 해소와 생략된 표현 확인, 대명사가 지시하는 것이 무엇인가에 대한 것들을 판별한다. 아래의 두 문장은 문법 구조만을 보면 옳은 구조를 가지고 있다고 할 수 있다. 하지만 해석을 하게 되면 한국은 아시아 국가이다라는 뜻을 가진

첫 번째 문장은 옳은 문장이지만, '아시아 국가는 한국이다' 라는 뜻을 가진 두 번째 문장은 의미론적으로 옳지 않게 된다.

- Korea is an asian country [O]
- An asian country is Korea [X]

의미 분석의 어려움에는 표기는 같지만 뜻이 다르고 어원적으로 서로 관련이 없는 동음이의어(heteronym), 여러 뜻을 가지고 있는 다의어(polysemy) 등에 의한 단어의 모호성 때문에 의미 분석이 제대로 된 의미 분석이 되지 않는 것에 있다. 이를 해결하기 위해 등장한 것이 바로 Word2Vec와 같은 단어 임베딩 기법이다. 이들은 의미적으로 유사한 단어들을 벡터 공산 상의 가까운 위치에 배치시키도록 학습된 신경망을 이용하는데, 유사한 단어들을 확인하는 것이 가능하고 문장의 의미론적인 부분을 확인할 수 있어 동음이의어, 다의어에 의한 모호성을 어느정도는 해결할 수 있다.

iv.화용 분석 (Pragmatic Analysis)

화용 분석(Pragmatic Analysis)단계는 언어가 사용된 관련 분야에 대한 지식을 통해 문장을 통한 화자의 의도를 파악하는 단계이다. 사용된 단어가 적절하게 사용됐는지를 파악하는 언어학의 화용론 또는 어용론이 여기에 해당한다.

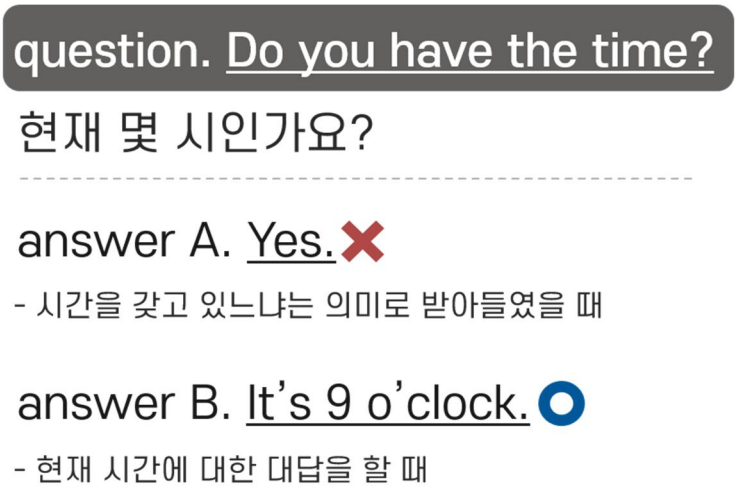


그림 5. 화용 분석의 예시

B. 수사구조이론

i. RST Label

수사구조이론(Rhetorical Structure Theory)은 텍스트들 사이에 유지되는 관계에 대한 이론이다. 1988년 William Mann, Sandra Thompson 등에 의해 발표되었으며 수사학 관계, 담화 관계라 부르는 문서 전체에 걸쳐 유지되는 텍스트 사이의 관계를 설정하여 텍스트 분석, 텍스트 생성 분야의 연구에서 주로 사용된다.[2] 수사구조이론에서 텍스트는 크게 핵과 위성으로 구분 지어지며 위성은 관계 레이블을 가지며 핵 속성을 가진 텍스트에 기여한다. 핵 속성의 텍스트는 위성의 수식을 받거나 핵 속성과의 동치 관계를 가지며, 스스로가 가진 의미만으로도 의사소통이 가능한 텍스트이다. 이는 언어학에서 말하는 술어와 주어의 관계와 유사하다. 일반적으로 잘 쓰여진 글이라 함은 각 문장들이 단순히 독립적이고 고립된 문장으로 사용되는 것이 아닌 의미적으로 연속적이고 일관성 있는 구조화된 글이다. 수사학적 분석은 글이 내포하고 있는 이러한 일관성 있는 구조를 밝히는 것을 목적으로 하고 있다. 분석의 결과는 트리(RST Tree) 형태로 표현이 가능하며 핵(Nucleus)과 위성(Satellite)이 트리의 노드(node) 또는 잎사귀(leaf)가 된다.

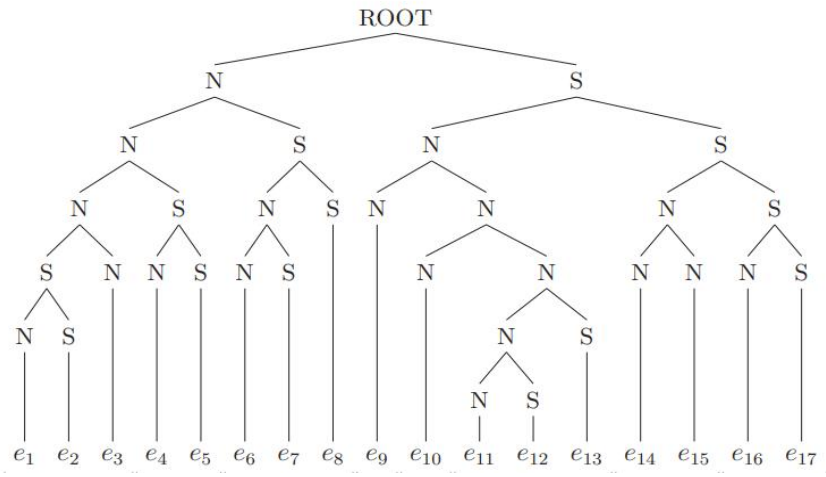
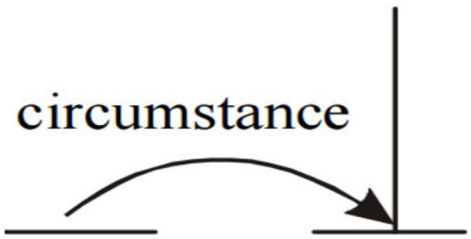


그림 6. RST Tree (N: 핵, S: 위성)

RST 에서 텍스트 사이의 관계를 표현하는 방법으로 핵 또는 위성에 관계 레이블을 부여하여 표현한다. 관계 레이블은 대상 텍스트의 속성, 방식에 따라 제시 관계, 주제 관계, 다중 핵 관계 등으로 분류할 수 있으며, 표준은 있지만 분석가에 따라 레이블을 수정하거나 삭제가 가능하다.

제시 관계는 Antithesis(반론), Background(배경), Concession(양보), Evidence(증거), Preparation(도입), Summary(요약) 등의 관계 레이블들이 포함되어 있으며 핵 속성일 경우는 텍스트를 긍정적으로 고려하거나 받아들이는 성향을 가지며, 위성 속성일 경우는 핵이 고려하는 제시적 기능을 수행한다. 주제 관계는 Circumstance(환경), Condition(조건), Elaboration(정교화), Cause(원인), Purpose(목적), Evaluation(평가), Result(결과) 등의 관계 레이블들이 포함되어 있으며 핵 속성일 경우는 관계가 연결된 내용들의 중심이며 위성 속성일 경우는 핵을 관계적으로 설명을 한다. 다중 핵 관계는 Joint(연결), Contrast(대조), Sequence(연속) 등의 관계 레이블이 포함되어 있으며 연결, 대조, 연속 등의 두 개

이상의 내용상 대등하게 연결되어 있는 관계를 말한다. 다중 핵 관계는 위성 속성을 가질 수 없다. 그 외의 관계 레이블을 기타 관계로 분류한다.

<table>
<tr><td rowspan="2">제시
관계</td><td>Antithesis(반론), Background(배경)
Concession(양보), Enablement(가능화)
Justify(정당화), Evidence(증거)
Motivation(동기화), Preparation(도입)
Restatement(재 진술), Summary(요약)</td></tr>
<tr><td> 
</td></tr>
</table>

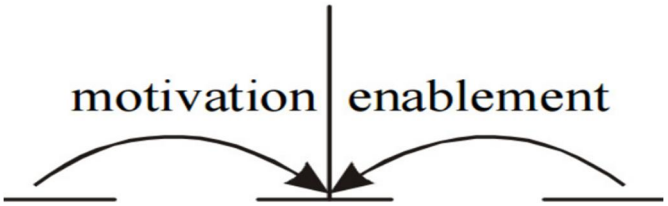
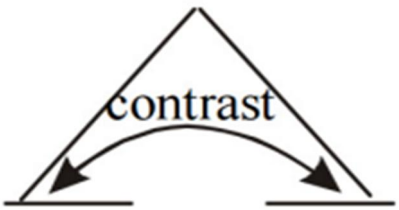
주제 관계	Circumstance(환경), Condition(조건) elaboration(정교화), evaluation(평가) interpretation(해석), means(수단) otherwise(양자택일), purpose(목적) solutionhood(해결성), Result(결과)
	
다중핵 관계	Joint(연결), Contrast(대조) Sequence(연속), Multinuclearrestatement(다핵재진술)
	

표 4. 수사구조이론 관계 레이블 분류

ii.구문 분석 Parser

RST 에서 Tree 구조를 만드는 분석은 기본적으로 텍스트 분석의 구문 분석(Syntax Analysis)단계라고 볼 수 있다. 구문 분석 단계에서 Parser Tree를 만드는 과정은 다음과 같다.

1) 어휘 분석기를 통해 문서를 토큰 스트림 단위로 분리한다.

2) 분리된 토큰 스트림으로부터 파스 트리를 생성한다.

토큰 스트림으로부터 파스 트리를 생성하는 방식에는 크게 하향식 구문 파싱(Top-Down Parsing)과 상향식 구문 파싱(Bottom-Up Parsing)이 있다. 하향식 구문 파싱은 문장의 상단부분에 해당하는 루트에서부터 터미널 노드 쪽으로 파스트리를 구성하는 것이다. 이때 노드들의 전위 순서(Preorder)는 좌측에서부터 생성하는 것이기 때문에 이를 입력 문자열에 대한 좌측 유도(Left Most Derivation)이라고도 한다.

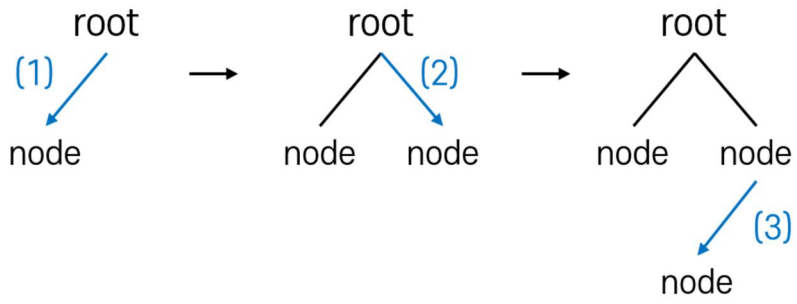


그림 7. 구문 분석의 하향식 파싱

상향식 구문 파싱은 구문 분석 트리의 가장 아래쪽의 leaf 에서 시작하여 트리의 상단 root 쪽으로 진행하면서 Parsing 하는 방식이다. 하향식 구문 파싱과는 반대로 트리의 root 기준으로 가장 최우단에서부터 생성되기 때문에 입력 문자열에 대한 최우단 유도(Right Most Derivation)이라고도 한다.

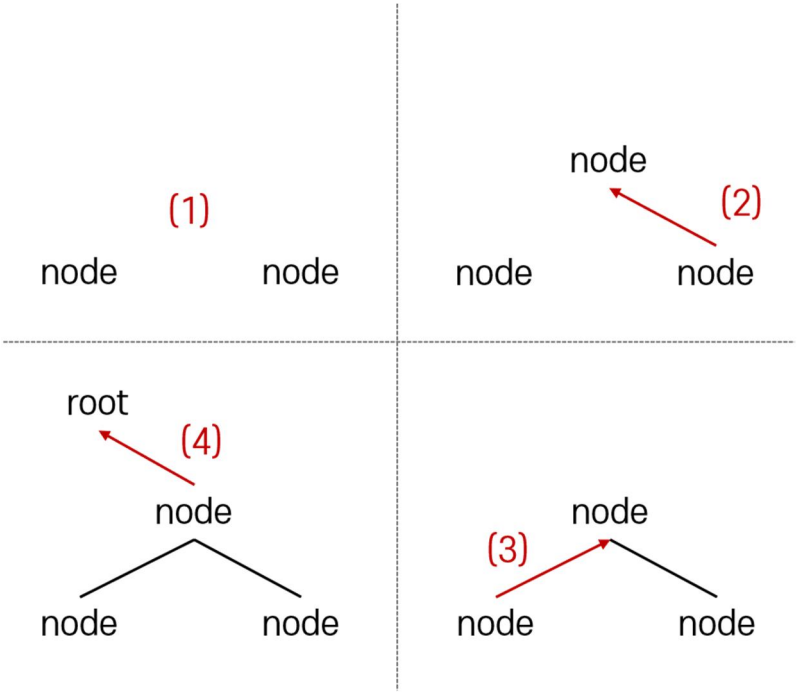


그림 8. 구문 분석의 상향식 파싱

하향식 구문 파싱은 상단에서부터 트리 순회를 root 에서 leaf 까지 진행하는데 있어서 구문이 맞지 않으면 다시 root 로 회귀해야 하는 backtracking 이 발생하는 경우가 있다. 때문에 연산과정이 길어질 수 있다는 단점이 있으며 상향식 구문 파싱은 이러한 문제점을 해결할 수 있다는 장점이 있어 구현이 다소 까다롭지만 많이 사용되고 있다.

iii.RST Parser

RST Parser는 입력 문서를 RST 관계 레이블을 가지고 있는 문장 쌍으로 표현이 가능한 형태로 Parsing하는 도구로, 구문 분석 Parser와 유사한 형태로 사용된다. 차이점은 구문 분석 Tree의 leaf에 해당하는 토큰 대신 EDU(Elementary Discourse Units)이라고 하는 사용하며 EDU 사이에는 관계성이라는 속성을 추가로 존재한다는 점과 각 EDU가 핵(N) 또는 위성(S)의 역할이 부여된다는 점이다. 학습 데이터 셋은 주로 RST-DT(RST-Discourse Treebank)를 사용한다.[4] RST-DT는 Parser 구현을 위한 밴치마킹 데이터 셋으로 385개의 월스트리트 등의 저널기사에서 사용된 문장들에 관계 주석이 달려있다. 학습 모델로는 LSTM, BEAT 등이 자주 사용된다.[5]

- **하향식 파싱 접근법 (Top-Down Approach)**

일반적은 구문 분석의 하향식 파싱과 유사하지만 위성 속성의 EDU가 핵 속성에 종속되는 구조를 가지고 있다.

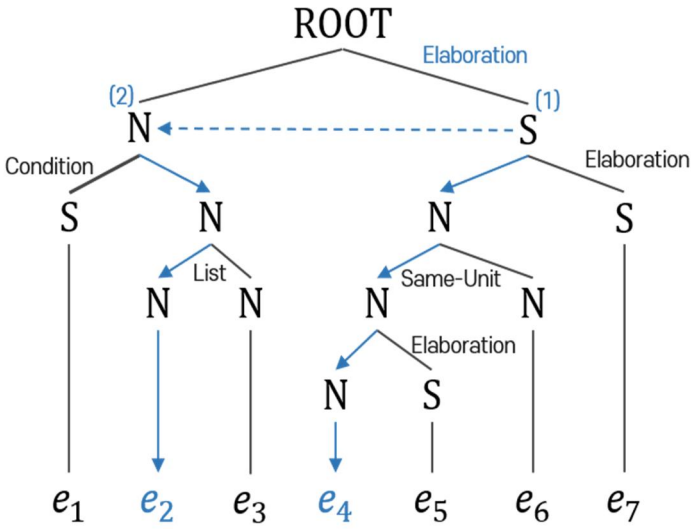


그림 9. RST Parser 의 하향식 파싱

- 상향식 파싱 접근법 (Bottom-Up Approach)

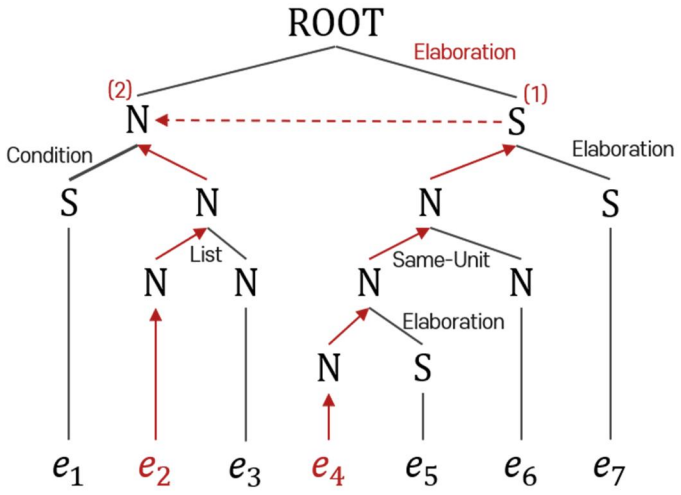


그림 10. RST Parser 의 상향식 파싱

C. 벡터 유사도 분석

i. 벡터 공간에서의 유사도 분석: 코사인 유사도 분석

코사인 유사도는 벡터공간의 두 벡터간 각도의 COS값을 이용하여 측정된 벡터 간의 유사한 정도를 말하며, 각도가 0°일 때 COS값은 최대값인 1를 가지며, 90°일 경우 0, 180°인 경우 -1의 값을 가진다. 자연어 처리 분야에서는 컴퓨터에게 자연어를 이해시키기 위해 인코딩이라는 과정을 걸치게 된다. 이때 단어들이 각각의 차원을 구성하고 문장이나 문서는 다차원의 단어들의 집합, 즉 다차원의 양수공간으로 만들어진다. 코사인 유사도는 차원 개수의 제약없이 어떤 차원에서도 적용이 가능하여 정보검색 및 텍스트 마이닝 분야에서 자주 사용되는 유사도 분석 기법이다.

벡터 A와 벡터 B가 주어졌을 때, 코사인 유사도 $\cos(\theta)$는 벡터의 스칼라 곱 연산과 벡터의 크기로 다음과 같이 구할 수 있다.

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

코사인 유사도 예시 : A = [**0 1 1 1 3 1**] B = [**2 1 2 3 1 1**]

1) 분모 계산 : 벡터의 곱셈

$$\|A\|\|B\| = \sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}$$

$$\sqrt{0^2 + 1^2 + 1^2 + 1^2 + 3^2 + 1^2} \times \sqrt{2^2 + 1^2 + 2^2 + 3^2 + 1^2 + 1^2}$$

$$= 16.124515$$

2) 분자 계산 : 벡터의 내적

$$A \cdot B = \sum_{i=1}^{n} A_i \times B_i$$

$$(0 \times 2) + (1 \times 1) + (1 \times 2) + (1 \times 3) + (3 \times 1) + (1 \times 1)$$

$$= 10$$

$$Similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{10}{16} = 0.625$$

ii.텍스트 벡터화(Text Vectorization)

컴퓨터를 통해 텍스트 처리를 함에 있어 코사인 유사도 구하는 것을 비롯하여 여러 처리를 벡터 공간에서 진행한다. 이를 위해서는 필수적으로 텍스트를 벡터로 변경해야 하는데, 컴퓨터공학에서는 이를 텍스트 벡터화(Text Vectorization)이라 한다. 대표적으로 원-핫 인코딩(One-hot encoding), 빈도수 기반 벡터화(TF-IDF), 단어 임베딩이 있다.

(ㄱ) 원-핫 인코딩(One-hot encoding)

원-핫 인코딩은 N개의 단어를 각 N차원의 벡터로 단순 표현하는 방식이다. 각 단어에 해당되는 차원의 값을 1로, 나머지는 0인 희소벡터(sparse vector)를 만든다. 즉, 각 차원이 해당되는 단어의 인덱스 역할을 한다. 원-핫 인코딩으로 만들어진 벡터의 크기는 말뭉치(corpus) 내 존재하는 단어의 수와 같다. 때문에 사용하고자 하는 말뭉치의 크기가 커질수록 속도 측면에서 성능이 크게 떨어진다. 또한, 만들어진 벡터는 서로 독립적이며 내적이 0이다. 그로 인해 각 벡터 사이에 특정한 관계성이나 유사성을 파악하기가 어렵다.

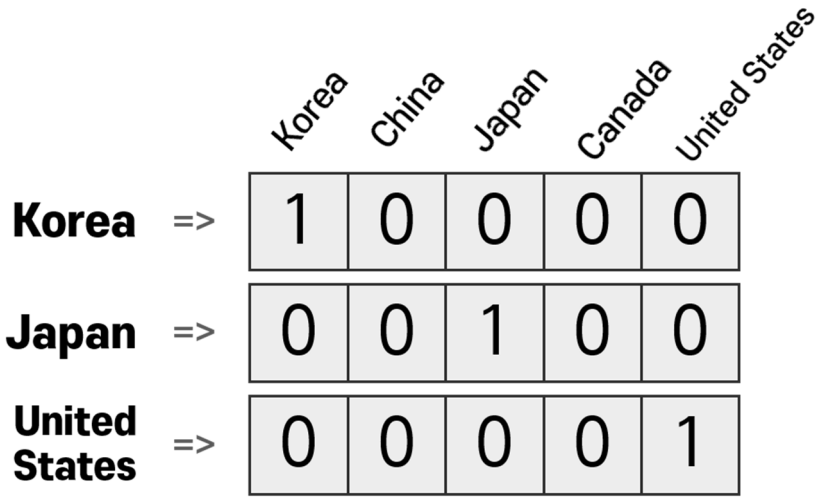


그림 11. One-hot encoding

(ㄴ) 빈도수 기반 텍스트 벡터화(TF-IDF)

TF-IDF(Term Frequency - inverse Document Frequency)는 특정 문서 내에서 등장

하는 특정 단어의 빈도와 해당 단어가 문서 전체 집합에서의 등장하는 빈도를 이용한 벡터화 방법이다. 특정 단어가 문서 내에서 자주 등장한다면 그 단어는 해당 문서에서 중요성이 높은 단어라고 볼 수 있다. 하지만 전체 문서 집합에서 봤을 경우에도 해당 단어가 자주 사용된다면 해당 단어는 전체적으로 자주 사용되는 단어임을 파악할 수 있어 중요성이 떨어지게 되는데, 이러한 원리를 이용한 것이 TF-IDF 방법론이다. 각 단어 t에 해당 문서 d와 전체 말뭉치 D에서의 TF-IDF는 다음과 같이 구할 수 있다.

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

$$tf(t,d) = f_{t,d} \qquad idf(t,D) = \log \frac{N}{|d \in D : t \in d|}$$

$tf(t,d) = f_{t,d}$의 값은 단어 t가 문서 d에 등장하는 빈도를 나타낸다. tf빈도를 구하는 산출 방식은 대표적으로 Boolean 빈도, Log 빈도, 증가 빈도 방식이 있다.

$idf(t,D)$는 역문서 빈도라고 하며, 특정 단어가 문서 전체에서 얼마나 나타났는지를 나타내는 값이다.

(ㄷ) 단어 임베딩(Word Embedding)

단어 임베딩은 비슷한 분포를 가진 단어의 주변 단어들도 비슷한 의미를 가질 것이라는 가정을 둔 단어 벡터화 기법이다. 원-핫 인코딩에서의 단어 벡터들 사이에 연결관계를 찾아보기 힘들다는 단점을 보완했다. 일반적으로 원-핫 인코딩과는 달리 미리 정해진 차원의 크기를 가지고 있는데 보통 200 ~ 300 차원을 사용한다. 대표적으로 Word2Vec, GloVe , FastText 등이 있으며, 신경망을 사용하여 학습시킨다. 단어 임베딩으로 생성된 벡터들은 연산도 가능한데, 유사한 텍스트 간의 벡터는 그림 14 의 “MAN” 에서 “WOMAN” 의 벡터와 “UNCLE” 에서 “AUNT” 벡터처럼 유사한 방향성을 가진다.

$$W(\text{MAN}) - W(WOMAN) \cong W(UNCLE) - W(AUNT)$$

$$W(\text{MAN}) - W(WOMAN) \cong W(KING) - W(QUEEN)$$

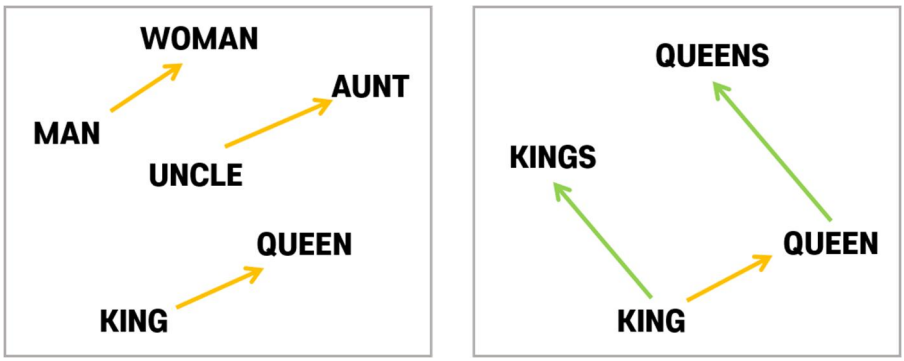


그림 12. Word Embedding 벡터 연산

문서 구조 유사도 분석 시스템

문서 구조 유사도 분석 시스템은 텍스트 간의 구조적 관계를 나타내는 Rhetorical Structure Theory 레이블을 기반으로 두 문서가 구조적으로 얼마나 유사하게 작성되었는지를 분석한다.

글은 주장, 문제 해결, 연설, 건의, 광고, 문학, 기사 등으로 나뉘는데, 각 글들의 목적에 따라 글의 구조가 크게 바뀐다. 실험에서의 문서 구조 비교를 명확하게 하기 위해 종류가 다른 두 개의 문서 그룹을 선정하였으며, 주장하는 글 속성의 글인 ‘사설’과 사실 보도 목적의 ‘스트레이트 기사’을 대상으로 실험을 진행했다.

A.문서 구조 유사도 분석 시스템

문서 구조 유사도 분석 시스템은 텍스트 간의 구조적 관계를 나타내는 Rhetorical Structure Theory 레이블을 기반으로 두 문서가 구조적으로 얼마나 유사하게 작성되었는지를 분석한다.

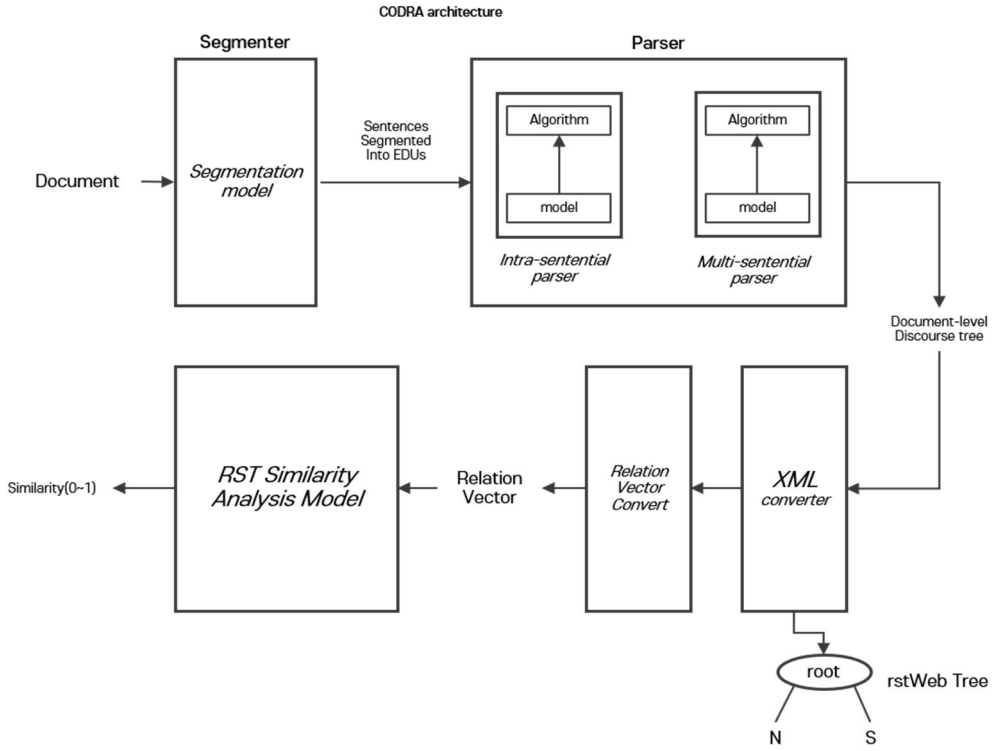


그림 13. RST 기반 문서 구조 분석 시스템

문서를 입력으로 주어지면 RST Parser 에 의해 문서를 핵과 위성으로 분리하고 각각에 알맞은 RST 레이블을 할당한다. RST Parser 로 RST-DT 벤치마킹 학습 데이터로 학습된 CODRA RST Parser 를 사용했다. 문서가 핵과 위성으로 분리되어 트리 형태의 모습이 된 것을 Document-level Discourse tree 라 부른다.

```
( Root (span 1 65)
  ( Nucleus (leaf 1) (rel2par span) (text _!The resignation of British Prime Minister Liz Truss on Thursday morning puts an
  ( Satellite (span 2 65) (rel2par Summary)
    ( Nucleus (span 2 10) (rel2par span)
      ( Nucleus (span 2 3) (rel2par span)
        ( Nucleus (leaf 2) (rel2par span) (text _!Republicans should take note of her mistakes_!) )
        ( Satellite (leaf 3) (rel2par Condition) (text _!if they want to avoid a similar debacle after the midterms and in 2
      )
      ( Satellite (span 4 10) (rel2par Elaboration)
        ( Nucleus (span 4 8) (rel2par span)
          ( Satellite (span 4 7) (rel2par Contrast)
            ( Nucleus (span 4 5) (rel2par span)
              ( Nucleus (leaf 4) (rel2par span) (text _!Truss ??? s first mistake was to push a radical economic agenda_!) )
              ( Satellite (leaf 5) (rel2par Elaboration) (text _!she did not campaign on ._!) )
            )
            ( Satellite (span 6 7) (rel2par Elaboration)
              ( Nucleus (leaf 6) (rel2par span) (text _!Her personal views_!) )
              ( Satellite (leaf 7) (rel2par Elaboration) (text _!supporting a low-tax , smaller government were telegraphed
            )
          )
          ( Nucleus (leaf 8) (rel2par span) (text _!But she did not campaign for the premiership on that agenda ._!) )
        )
        ( Satellite (span 9 10) (rel2par Elaboration)
          ( Nucleus (leaf 9) (rel2par Joint) (text _!She had promised some modest tax reductions_!) )
          ( Nucleus (leaf 10) (rel2par Joint) (text _!and offered rhetorical backing for deregulation ._!) )
        )
      )
    )
```

그림 14. RST Parser 의 파싱 결과

이후 변환된 Tree 를 XML 타입으로 변환하는 과정을 거치는데, 이는 분석이 편한 데이터 형태로 가공하는 용도도 있지만 tree 가 올바르게 생성되었는지를 확인하기 위함이다. 변환된 XML 은 rstWeb Server 를 통해 확인할 수 있는데 그림 15 처럼 RST Tree 를 시각적으로 확인할 수 있다. XML 를 입력 받아 RST 관계 레이블의 빈도수에 대한 벡터화를 진행한다. 벡터화 기법으로는 TF-IDF 를 사용했다. 각 문서에 대한 벡터를 해당 문서의 레이블 빈도수와 전체 문서 집합에서의 레이블 빈도수를 사용하여 생성한다. RST Similarity Analysis Model 은 유사도 분석의 대상이 되는 글 종류의 문서의 벡터 공간을 가진 코사인 유사도 모델이다. 사설 종류의 글에 대한 유사도를 분석하고자 한다면 사설 문서들의 벡터 공간을 가지고 있어야 한다.

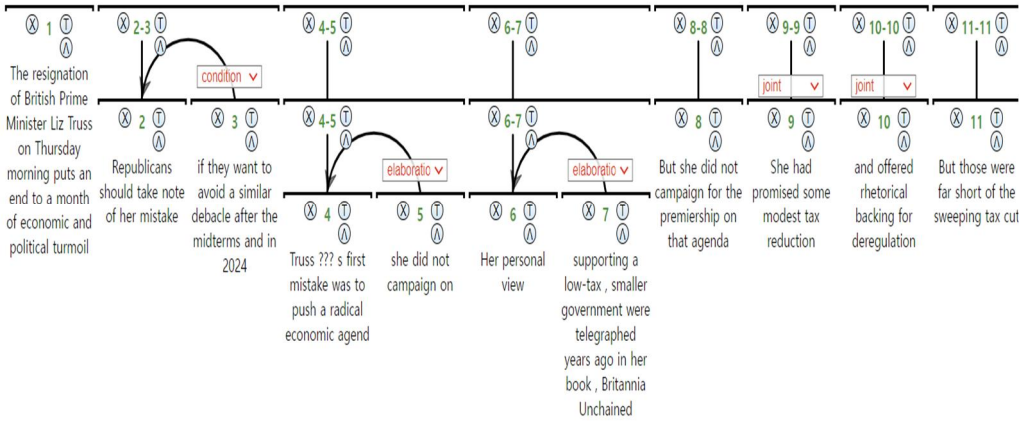


그림 15. retWeb 에서 생성된 입력 데이터의 RST Tree

B. 실험

실험 데이터는 CNN 사설 50 편과 CNN 스트레이트 기사 50 편을 사용한다. 이 중에 40 편은 각 유형 모델의 벡터 공간을 만드는 데 사용되며, 각 10 편씩 20 편을 결과 비교를 위한 테스트 데이터로 활용한다.

실험과정

1. 문서를 RST Parser 를 통해 관계 분석 후 TF-IDF 벡터를 생성한다.
2. 사설 데이터로 생성된 벡터 공간을 가진 사설 모델과 스트레이트 기사 데이터로 생성된 벡터 공간 기사 모델을 각 생성한다.
3. 각 타입 10 편의 문서를 각 모델에 입력으로 주어 나온 결과를 확인한다.

사설 테스트 데이터	유사도 최대 수치	유사도 최소 수치
TEST_Openion_1	0.943727	0.691955
TEST_Opinion_2	0.949628	0.729587
TEST_Opinion_3	0.952610	0.734187
TEST_Opinion_4	0.952159	0.752817
TEST_Opinion_5	0.949301	0.700948
TEST_Opinion_6	0.955779	0.669619
TEST_Opinion_7	0.944824	0.690797
TEST_Opinion_8	0.946626	0.673897
TEST_Opinion_9	0.959168	0.638240
TEST_Opinion_10	0.945142	0.751172

표 5. 사설 벡터 공간에서의 사설 데이터 분석 결과

표 5는 사설 벡터 공간에서의 사설 데이터에 대한 유사도 분석 결과이다. 유사도 최대 수치는 벡터 공간 내의 여러 벡터 중 입력 벡터와 가장 유사도가 높게 나오는 벡터의 수치를 의미하며, 유사도 최소 수치는 벡터 공간 내의 벡터 중 입력 벡터와 유사도가 가장 낮게 나온 벡터의 유사도를 의미한다. 테스트 사설 10편모두 최대 유사도는 94% 이상이며, 최소 수치는 63% ~ 75%이다.

뉴스 기사 테스트 데이터	유사도 최대 수치	유사도 최소 수치
TEST_NEWS_1	0.582814	0.202679
TEST_NEWS_2	0.774279	0.406535
TEST_NEWS_3	0.671781	0.291527
TEST_NEWS_4	0.639443	0.253917
TEST_NEWS_5	0.765137	0.408405
TEST_NEWS_6	0.671606	0.310026
TEST_NEWS_7	0.611238	0.347140
TEST_NEWS_8	0.705180	0.408300
TEST_NEWS_9	0.708186	0.339347
TEST_NEWS_10	0.671302	0.390844

표 6. 사설 벡터 공간에서의 기사 데이터 분석 결과

사설 벡터 공간에서의 기사 데이터 분석 결과는 최대 유사도가 77%, 최소수치는 20% 까지 낮아졌음을 확인했다.

사설 테스트 데이터	유사도 최대 수치	유사도 최소 수치
TEST_OPINION_1	0.559598	0.271349
TEST_OPINION_2	0.477768	0.162957
TEST_OPINION_3	0.694523	0.293127
TEST_OPINION_4	0.595365	0.227330
TEST_OPINION_5	0.634646	0.180274
TEST_OPINION_6	0.608422	0.230356
TEST_OPINION_7	0.668508	0.242185
TEST_OPINION_8	0.662431	0.209205
TEST_OPINION_9	0.550317	0.174063
TEST_OPINION_10	0.621515	0.306240

표 7. 뉴스 기사 벡터 공간에서의 사설 데이터 분석 결과

뉴스 기사 테스트 데이터	유사도 최대 수치	유사도 최소 수치
TEST_NEWS_1	0.898897	0.397645
TEST_NEWS_2	0.950937	0.279726
TEST_NEWS_3	0.911594	0.438372
TEST_NEWS_4	0.915325	0.373709
TEST_NEWS_5	0.943979	0.375422
TEST_NEWS_6	0.892637	0.384845
TEST_NEWS_7	0.948787	0.453087
TEST_NEWS_8	0.930083	0.182745
TEST_NEWS_9	0.974296	0.203946
TEST_NEWS_10	0.965589	0.413068

표 8. 뉴스 기사 벡터 공간에서의 기사 데이터 분석 결과

표 7와 표 8은 뉴스 기사 벡터 공간에서의 사설, 기사 데이터에 대한 분석 결과이다. 사설 데이터의 경우 최대 유사도가 약 66%, 최소 유사도가 16%의 결과를 얻었으며, 뉴스 기사의 경우 최대 유사도 96%, 최소 유사도 20%의 결과를 얻었다.

결론

본 논문은 수사 구조 이론에서 쓰이는 텍스트 사이의 관계를 나타내는 RST 관계 레이블을 사용하여 문서 구조 유사성을 분석하였다. 문서 내 등장하는 RST 레이블들의 빈도수와 TF-IDF 를 사용해 문서 구조에 대한 벡터화를 진행한 후 코사인 유사도 검사를 실시하는 방식으로 실험을 진행했다. 사설과 기사 종류의 문서 집합으로 진행한 결과 서로 다른 유형의 벡터 공간에서는 유사도 최대/최소값이 크게 떨어졌고, 같은 유형의 벡터 공간에서는 유사도 최대/최소값이 상대적으로 높게 나왔음을 확인할 수 있었다. 하지만 서로 다른 유형의 공간에서도 특정 문서의 경우에는 70%정도의 유사도가 나오기도 했는데, 이는 해당 문서가 두 종류의 글의 구조를 가지고 있을 경우라 생각된다. 하지만 문서에서 사용된 관계의 빈도수만을 가지고 벡터화를 하였기에 각 관계의 문서 내에서의 시간적 정보가 손실됐다는 부분이 존재한다. 추후 연구를 통해 문서 벡터를 생성하는 과정에서 RST 관계의 시간적 정보를 담을 수 있도록 한다면 구조 유사성의 정확도를 올릴 수 있을 것으로 기대된다.

참고 문헌

[1] Kobayashi, Naoki, et al. "Improving neural RST parsing model with silver agreement subtrees." Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021.

[2] Morey, Mathieu, Philippe Muller, and Nicholas Asher. "How much progress have we made on RST discourse parsing? A replication study of recent results on the RST-DT." Conference on Empirical Methods on Natural Language Processing (EMNLP 2017). 2017.

[3] Feng, Vanessa Wei, and Graeme Hirst. "A linear-time bottom-up discourse parser with constraints and post-editing." Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014.

[4] Joty, Shafiq, Giuseppe Carenini, and Raymond T. Ng. "Codra: A novel discriminative framework for rhetorical analysis." Computational Linguistics 41.3 (2015): 385-435.

[5] Knuth, Donald E. "Top-down syntax analysis." Acta Informatica 1.2 (1971): 79-110.

[6] 김병건. "신문의 사설· 칼럼에 나타난 ‘진보’에 대한 비판적 담화 분석." 사회언어학 24.1 (2016): 65-90.