# TextRank를 이용한 문서 요약

최종현

# 목차

# 1. 요약

## 1.1 프로젝트 목적

- TextRank 알고리즘을 이용하여 문서 요약 시스템 구현
- 사용자가 검색한 문서들의 내용을 요약하여 빠르게 검토할 수 있도록 서비스를 제공하고자 함

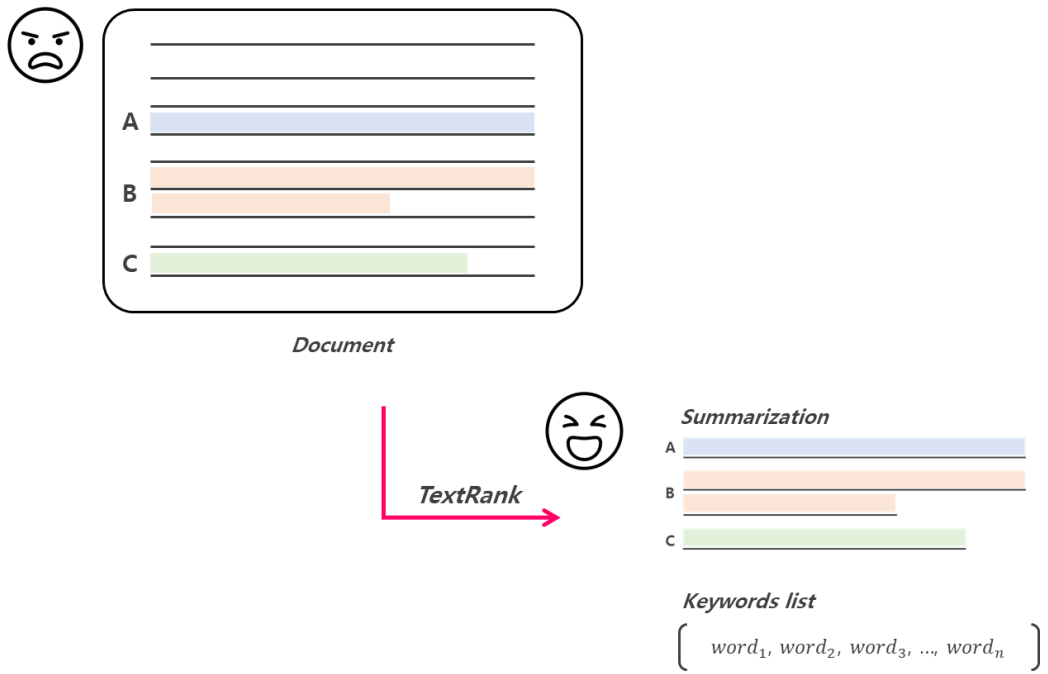
## 1.2 시스템 프로세스

① 자연어 처리(NLP) : 문장 분리 후 명사 추출

② TF-IDF 모델링 : 단어의 가중치 계산 후 TF-IDF Matrix 생성

③ 그래프 생성 : TF-IDF Matrix를 통해 Correlation Matrix 생성

④ TextRank 적용 : TextRank를 계산하여 높은 순으로 정렬




## 1.3 시스템 테스트

- [다음 뉴스]에서 제공 하는 '자동 요약' 서비스와 비교 검증함
- 시스템과 [다음]의 요약문을 ROUGE를 이용하여 검증

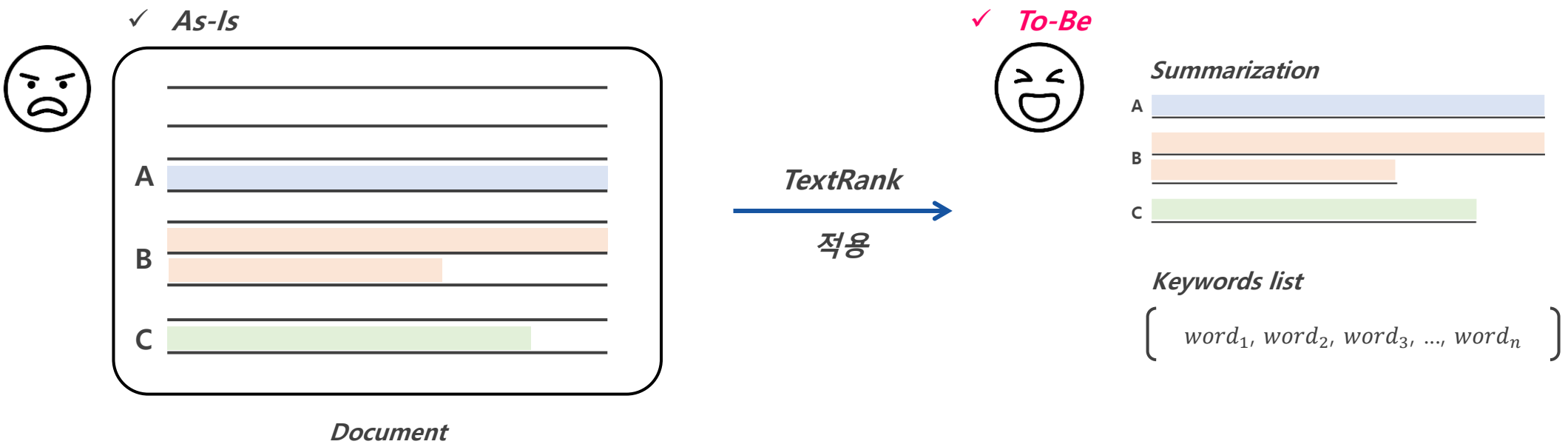
# 2. 개요

## 2.1 개요

- TextRank알고리즘을 적용하여 문서 내에서 문장들의 중요도를 계산하여 문서요약 시스템 구현
- 내용이 많은 문서에서 핵심 문장들만 추출하여 보여줌 (e.g. 3줄 요약)

## 2.2 목적

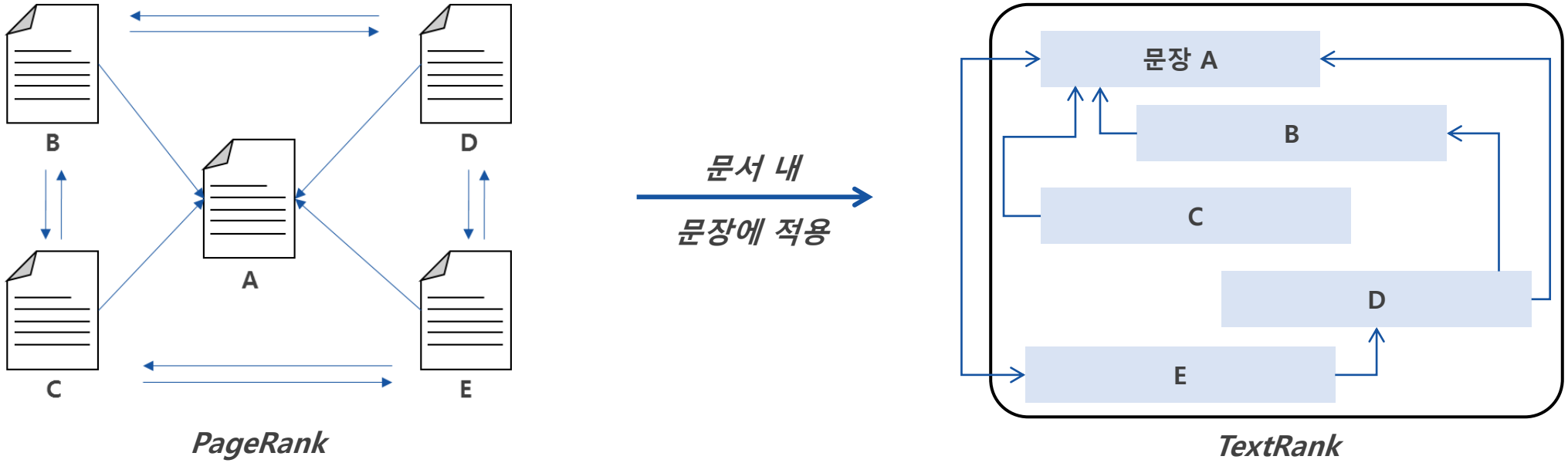
- 사용자가 검색한 문서들의 내용을 요약하여 빠르게 검토할 수 있도록 서비스를 제공하고자 함
- 사용자가 검색한 문서에서 핵심 단어들을 추출하여 제공하고자 함



# 3. TextRank

## 3.1 TextRank 란?

- Google의 **PageRank**(Brin, 1998)를 텍스트에 적용한 알고리즘
- Mihalcea(2004)이 제안한 알고리즘으로 텍스트에 관한 graph-based ranking model
- PageRank는 '중요도가 높은 웹 사이트는 다른 많은 사이트로 부터 링크를 받는다'는 점에 착안하여 Ranking을 계산하는 알고리즘
- **TextRank**는 PageRank의 사이트 대신 단어나 문장으로 대체하여 문서 내에서 문장의 Ranking을 계산하는 알고리즘




*PageRank* *TextRank*

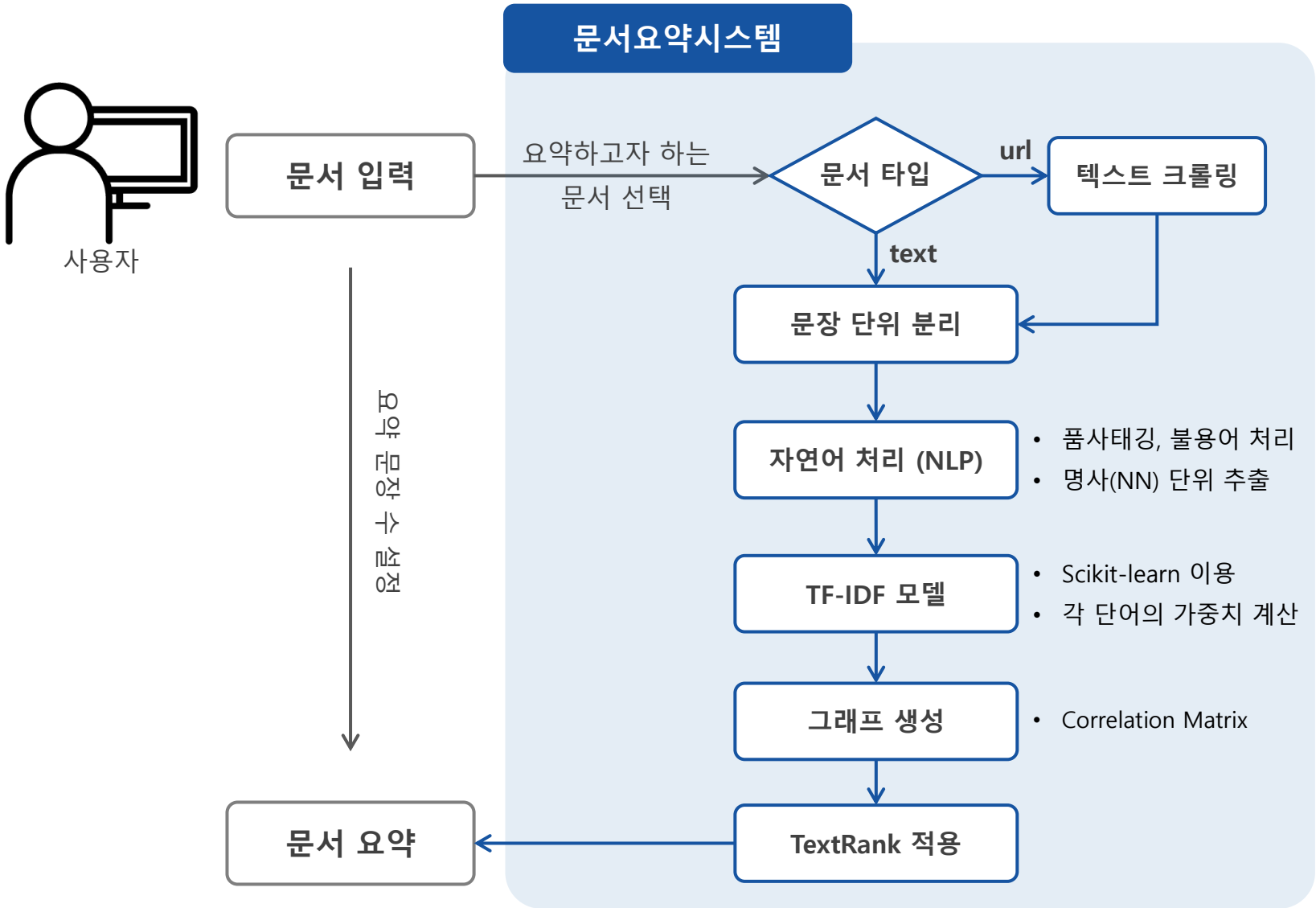
# 3. TextRank

## 3.2 TextRank 식

$$TR(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} TR(V_j)$$

- $\boldsymbol{TR(V_i)}$ : 문장 또는 단어 $(V)$ $i$에 대한 TextRank 값
- $\boldsymbol{w_{ij}}$ : 문장 또는 단어 $i$ 와 $j$ 사이의 가중치
- $\boldsymbol{d}$ : $damping\ factor$, PageRank에서 웹 서핑을 하는 사람이 해당 페이지를 만족하지 못하고 다른 페이지로 이동하는 확률로 TextRank에서도 그 값을 그대로 사용 (0.85로 설정)
- 문장 또는 단어 $V_i$에 대해 가중치 $w_{ij}$와 $w_{ji}$를 계산한 뒤 $V_i$에 대한 그래프를 구성
- TextRank $TR(V_i)$를 계산 한 뒤 높은 순으로 정렬

## 4.1 문서 요약 프로세스


문서요약시스템
사용자
문서 입력
요약하고자 하는
문서 선택
문서 타입
url
텍스트 크롤링
text
문장 단위 분리
자연어 처리 (NLP)
• 품사태깅, 불용어 처리
• 명사(NN) 단위 추출
TF-IDF 모델
• Scikit-learn 이용
• 각 단어의 가중치 계산
그래프 생성
• Correlation Matrix
TextRank 적용
문서 요약
요약 문장 수 설정

## 4.1 텍스트 크롤링

- 웹 페이지에서 텍스트를 추출하기 위해 Python 크롤링 패키지인 BeautifulSoup, Scrapy, Newspaper를 비교함
- 특정 사이트가 아닌 다양한 사이트에서 텍스트 크롤링이 가능한 **Newspaper**를 사용

## 4.2 문장 단위 분리

- TextRank를 적용하기 위한 전처리 단계
- Python 한글 자연어처리(NLP) 패키지인 KoNLPy의 꼬꼬마(Kkma) 형태소 분석기를 이용하여 문장단위로 추출

## 4.3 자연어 처리 - Natural Language Processing

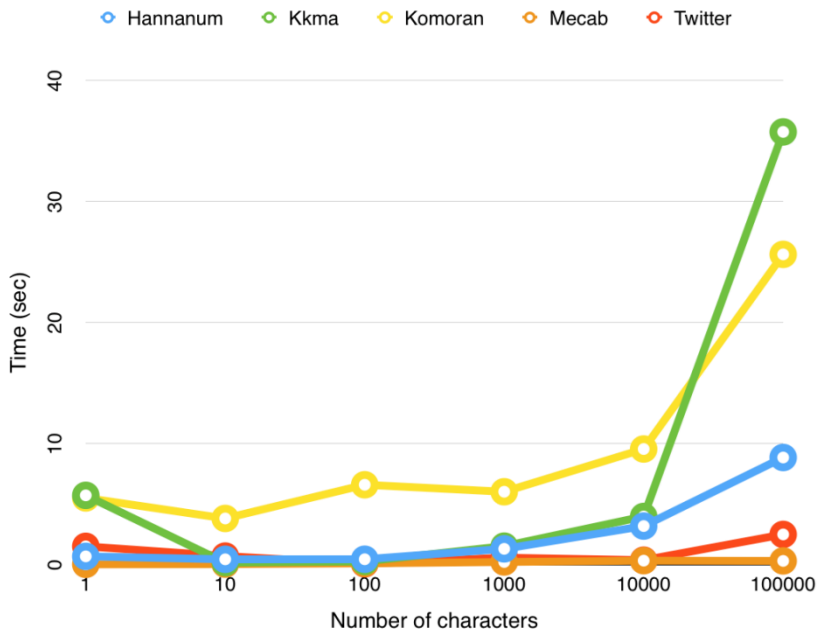
### 4.3.1 자연어 처리 란?

- 사람의 언어를 기계적으로 분석하여 컴퓨터가 이해할 수 있는 형태로 만드는 작업
- 자연어처리 기술에는 대표적으로 형태소 분석, 품사(POS)태그, 불용어 처리가 있음
- 문서의 길이와 속도를 고려하여 Python 한글 NLP 패키지인 KoNLPy 중 **Twitter** 형태소 분석기를 이용

**NLP**

| | |
|---|---|
| **형태소 분리** (Tokenization) | • 자연어 분석의 첫 단계로써, 문자열을 언어의 최소 단위인 형태소로 변환하는 작업 |
| **품사 태깅** (Part-Of-Speech Tagging) | • 형태소 분석을 통해 나온 결과 중 가장 적합한 형태의 품사를 태깅하는 작업 |
| **원형처리** (Lemmatization) | • 문장 내 단어들을 해당 단어의 어원(root word, lemma)로 변환하는 작업 (e.g. classified → classify) |
| **불용어 처리** (Stopword) | • 불용어(Stopword)는 문장 내에서 내용을 나타내는데 의미를 가지지 않는 단어들을 의미하며, 관사, 전치사, 조사 등을 제거하는 작업 |

*NLP 요소*

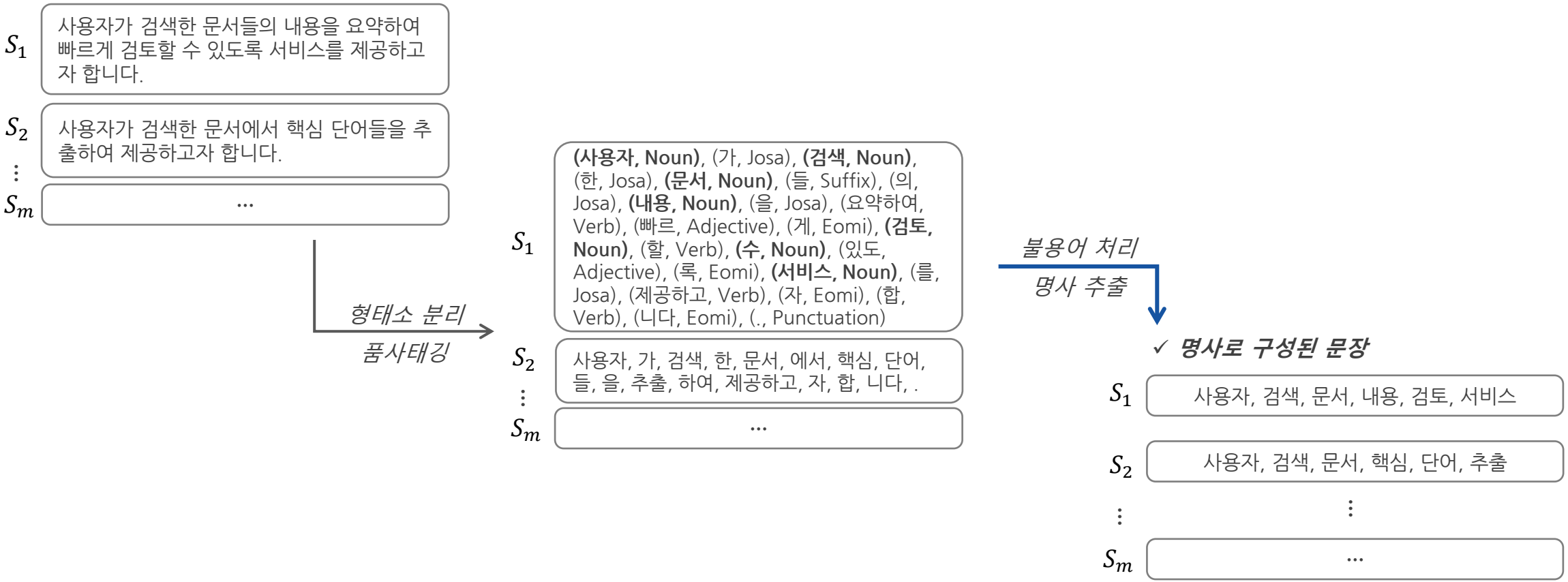



*형태소 분석기 비교*

## 4.3 자연어 처리 - Natural Language Processing

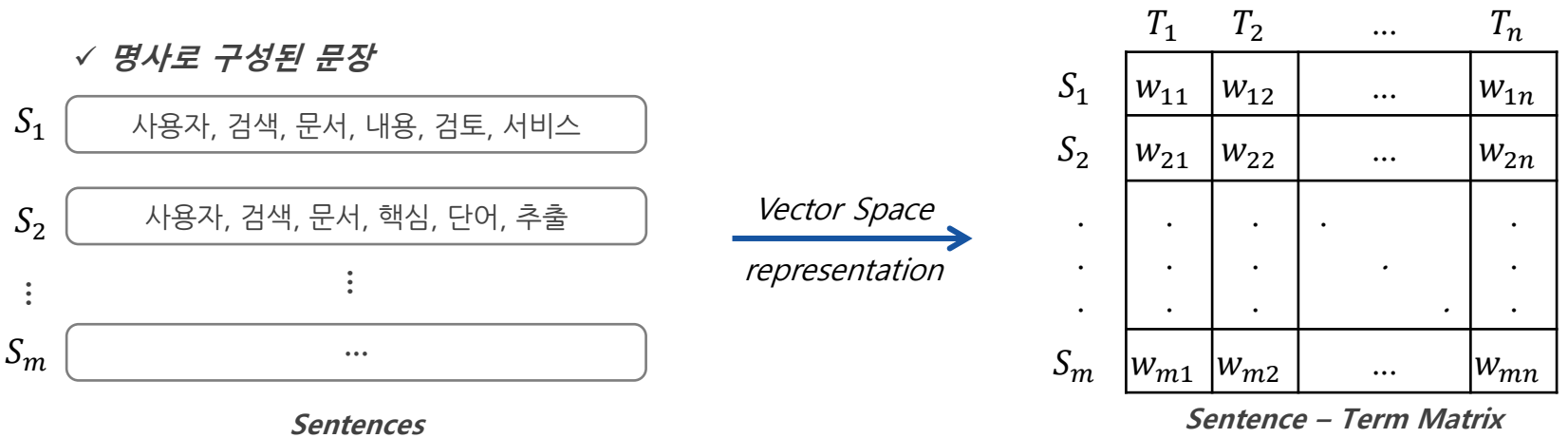
### 4.3.2 문장에서 명사 단위 추출

- TF-IDF 모델을 적용하기 위한 전처리 과정
- Twitter 형태소 분석기를 이용하여 품사태깅 후 명사단위로 추출 → *명사만으로 이루어진 문장으로 변환*



## 4.4 TF-IDF 모델

- TF-IDF(Term Frequency - Inverse Document Frequency)는 정보 검색(Information Retrieval)과 텍스트 마이닝에서 사용하는 가중치
- 여러 문서로 이루어진 문서군이 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내는 통계적 수치
- **TF(Term Frequency):** 단어빈도로 특정 단어가 문서 내에 얼만큼의 빈도로 등장하는지를 나타내는 척도
- **IDF(Inverse Document Frequency):** 역문헌 빈도수로 문서 빈도의 역수. 전체 문서 개수를 해당 단어가 포함된 문서의 개수로 나눈 것을 의미
- TF-IDF 의 가중치를 구하는 식 → $W_{i,j} = tf_{i,j} \times \log(^{N}/_{df_i})$
- Python의 머신러닝 패키지 **Scikit-learn** 을 이용하여 TF-IDF 모델링 수행
- TF-IDF 모델링 한 후 단어를 벡터로 나타내기 위해 **Sentence - Term Matrix** 생성



## 4.5 그래프 생성

- TextRank 모델을 적용하기 위한 전처리 과정
- Sentence - Document Matrix와 Sentence - Document Matrix의 전치 행렬을 곱하여 **Correlation Matrix**를 계산
- Correlation Matrix의 각 원소를 문장 $S_i$와 $S_j$ 사이의 가중치 그래프(Weighted graph)로 나타낼 수 있음

| | $T_1$ | $T_2$ | ... | $T_n$ |
|---|---|---|---|---|
| $S_1$ | $w_{11}$ | $w_{12}$ | ... | $w_{1n}$ |
| $S_2$ | $w_{21}$ | $w_{22}$ | ... | $w_{2n}$ |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| $S_m$ | $w_{m1}$ | $w_{m2}$ | ... | $w_{mn}$ |

Sentence – Term Matrix

Transpose

×

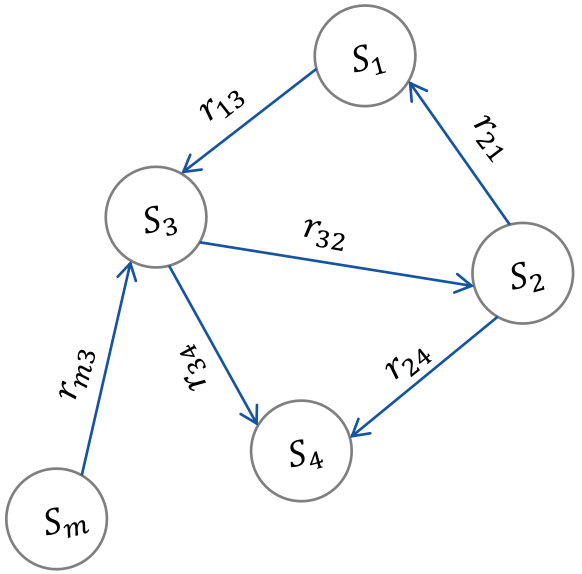
| | $T_1$ | $T_2$ | ... | $T_n$ |
|---|---|---|---|---|
| $T_1$ | $w_{11}$ | $w_{21}$ | ... | $w_{m1}$ |
| $T_2$ | $w_{12}$ | $w_{22}$ | ... | $w_{m2}$ |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| $T_n$ | $w_{1n}$ | $w_{2n}$ | ... | $w_{mn}$ |
| | $S_1$ | $S_2$ | ... | $S_m$ |

Term – Sentence Matrix

=

| | $S_1$ | $S_2$ | ... | $S_m$ |
|---|---|---|---|---|
| $S_1$ | 1 | $r_{12}$ | ... | $r_{1m}$ |
| $S_2$ | $r_{21}$ | 1 | ... | $r_{2m}$ |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| $S_m$ | $r_{m1}$ | $r_{m2}$ | ... | 1 |

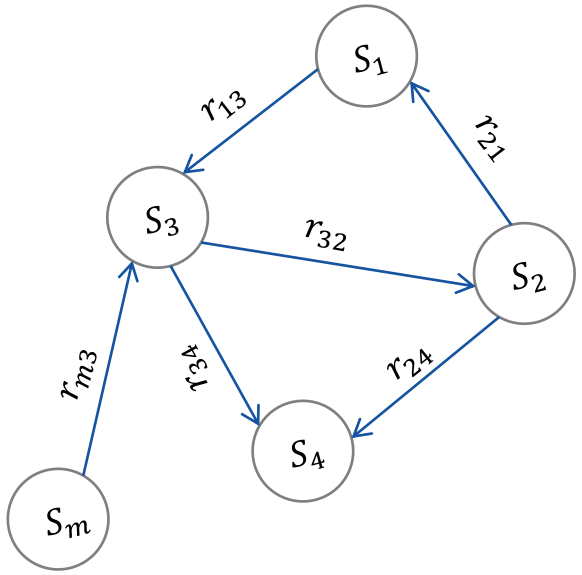
Weighted graph



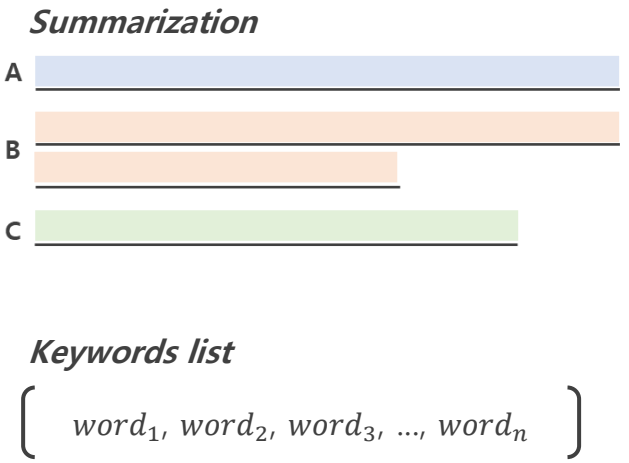
## 4.6 TextRank 알고리즘 적용

- 문장의 가중치 그래프를 이용하여 TextRank 알고리즘 적용
- TextRank 값이 높은 순으로 정렬한 뒤 요약할 문장 수 출력
- 핵심 단어 추출의 경우 단어의 가중치 그래프를 생성 후 TextRank 알고리즘 적용

$$TR(S_i) = (1-d) + d * \sum_{S_j \in In(S_i)} \frac{w_{ji}}{\sum_{S_k \in Out(S_j)} w_{jk}} TR(S_j)$$



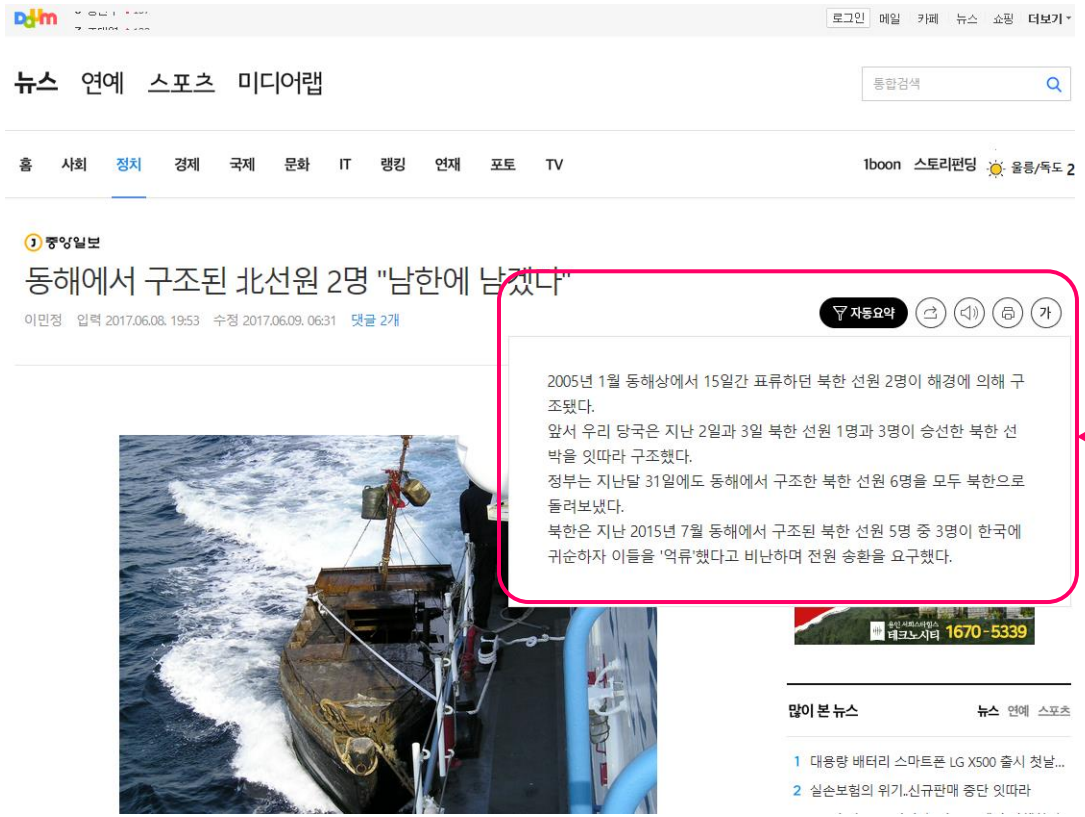

TextRank 적용



# 5. 시스템 테스트

## 5.1 테스트 데이터 설정

- 다음 뉴스 의 언론사별 뉴스 카테고리에서 “조선일보, 중앙일보, 동아일보” 선택함
- 다음 뉴스에서 제공 하는 **‘자동 요약’** 서비스와 비교 검증함
- 각 정치 부분의 기사 중 1 주일(2017년 6월 5일 ~ 6월 9일까지) 기사를 크롤링 함 → *총 785 개*



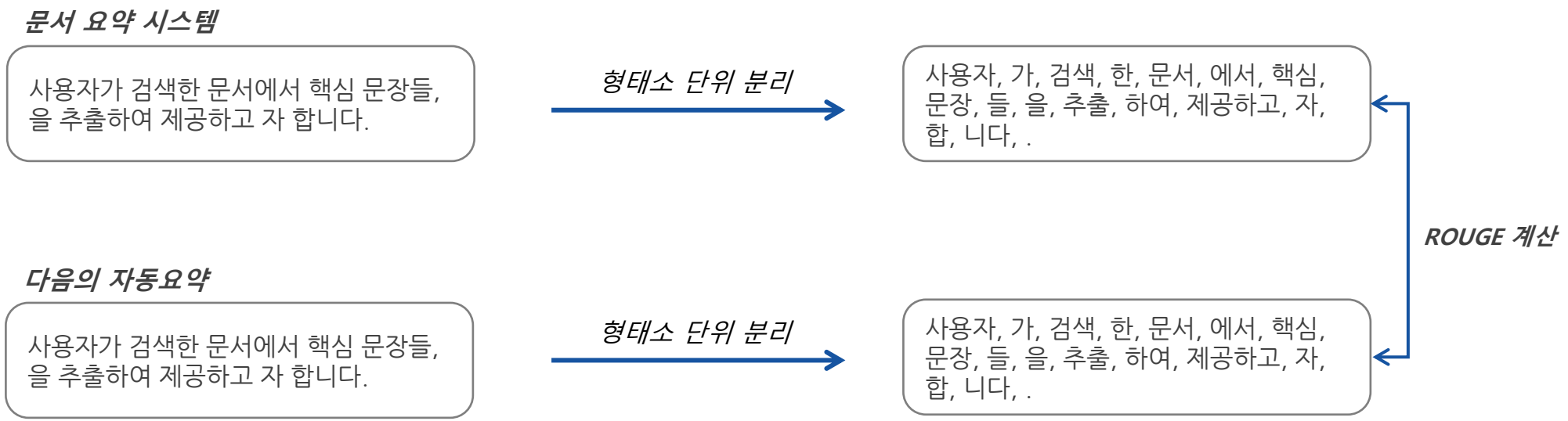

*다음 뉴스 > 자동요약*

***자동요약 서비스***

- ✓ 다음 뉴스에서 제공하는 ‘자동요약’ 서비스와 ROUGE 척도를 이용
- ✓ ‘자동요약’ 서비스가 없는 110개의 기사를 제외한 675개 기사와 ROUGE 척도를 이용하여 검증
- ✓ 하지만, 다음의 자동요약 또한 완벽한 정답 셋이라고 하기 어려움
  → 대체로, 기사 앞 문단만 가져와 ‘자동요약’이라고 서비스함(링크 참조)

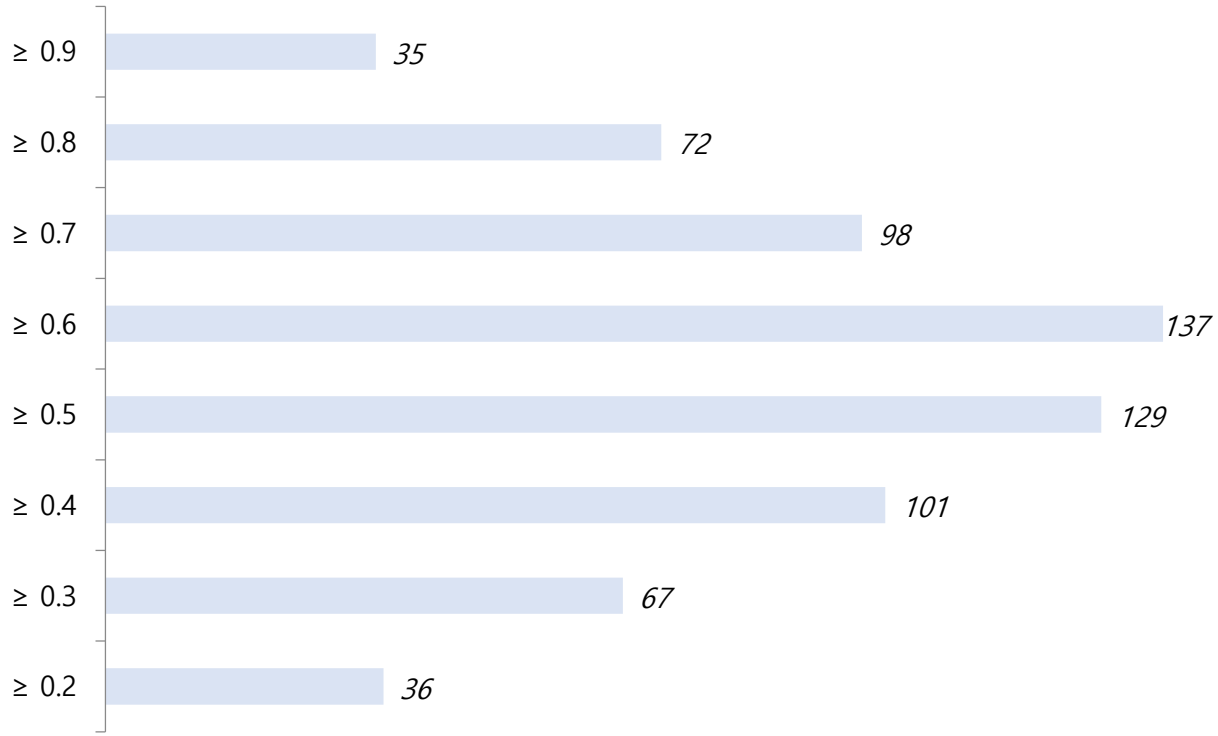
## 5.2 테스트 방법

- 문서 요약 시스템을 검증하기 위한 척도로 사람의 평가와 높은 상관 관계가 있다고 알려 져 있는 **ROUGE**(Recall-Oriented Understudy for Gisting Evaluation, CY Lin)을 이용
- ROUGE를 이용하여 시스템의 요약문과 다음이 제공하는 요약문 사이의 겹치는 단어수(n-gram)에 따라 95% 신뢰구간으로 계산
- ROUGE 계산을 하기 전, 시스템의 과 다음이 제공하는 각 요약문 별로 형태소 단위로 분리 후 ROUGE 계산을 수행함



### 5.3 테스트 결과

- 조선, 중앙, 동아 일보의 정치 부분 675개의 기사에 대한 평균 ROUGE는 0.6으로 나타남
- 다음 뉴스의 '자동요약' 서비스 또한 정확한 정답 셋이 아님을 감안한다면, ROUGE 값이 유의미 하다고 판단됨




*ROUGE 점수 분포 표*

# 6. 참고문헌


- 홍진표, 차정원. "TextRank 알고리즘을 이용한 한국어 중요 문장 추출." 한국정보과학회 학술발표논문집, 36.1C (2009.6): 311-314.
- 권영대, 김누리, 이지형. 문장 수반 관계를 고려한 문서 요약. 정보과학회논문지, 2017, 44.2: 179-185.
- Page, Lawrence, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.
- Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing order into texts." Association for Computational Linguistics, 2004.
- LIN, Chin-Yew. Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. 2004.
- 파이썬 한글 형태소 분석기 패키지 KoNLPy - http://konlpy-ko.readthedocs.io/ko/v0.4.3/
- Twitter 형태소 분석기 - https://github.com/twitter/twitter-korean-text


THANK YOU