NLP – 키워드 추출

Index

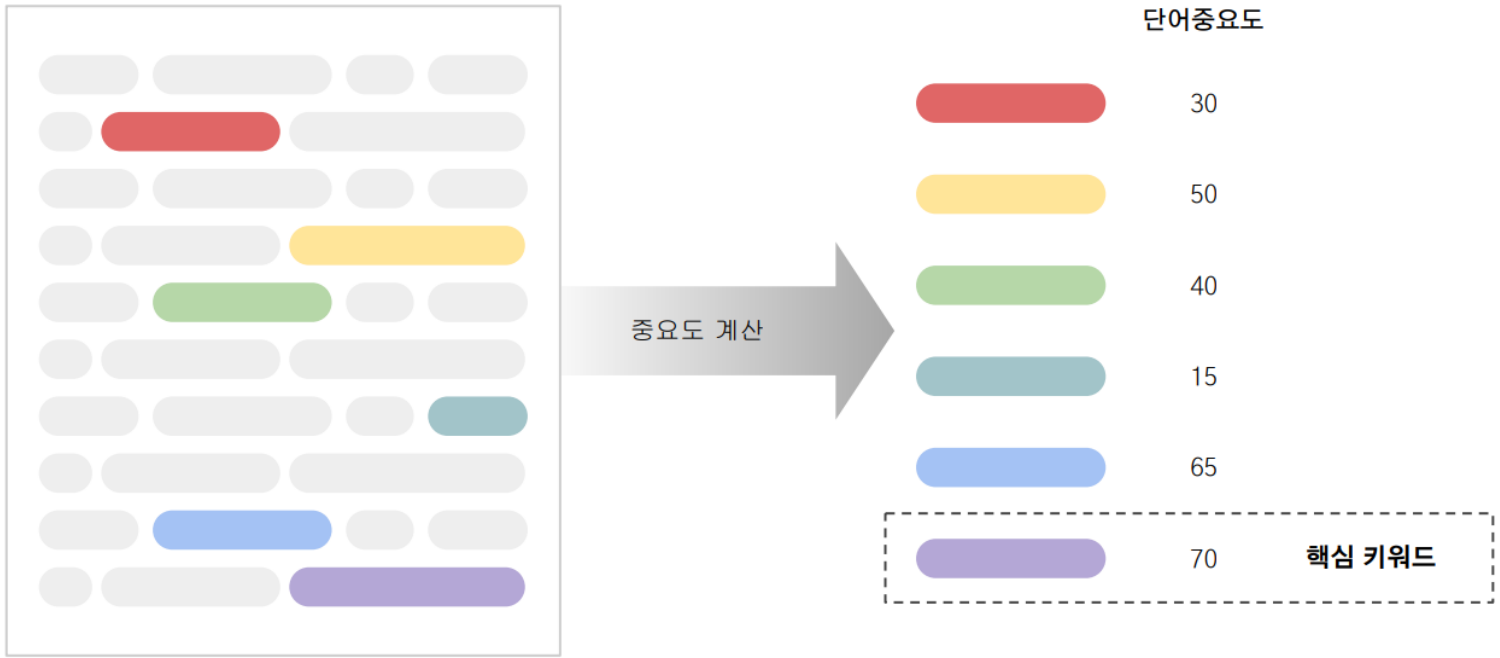
1. 키워드 추출
   - 필요성
   - 통계적 접근
2. TF-IDF
3. TextRank
   - pagerank

Keyword Extraction

키워드 추출

키워드 추출

- 키워드 추출은 문서에서 가장 중요한 단어를 자동으로 추출하는 과정
- 중요한 단어를 추출한다 → 단어의 중요성을 어떻게 판단할 것인가



필요성

- 대량 데이터 처리 가능
  키워드 추출을 활용해 대량데이터를 분석할 수 있다. 직접 문서를 읽고 주요 용어를 수동으로 식별할 수 있지만 많은 시간이 소요된다. 이 작업을 자동화하면 다른 더 중요한 작업에 집중할 수 있다.

- 추출의 일관성
  키워드 추출은 규칙과 사용자가 정의한 매개변수를 기반으로 작동한다. 따라서 텍스트 분석을 수동으로 수행 할 때 나타나는 불일치를 고려할 필요가 없다.

- 실시간 분석 가능
  소셜 미디어 게시물, 고객 리뷰, 설문 조사 또는 고객 지원에 대한 키워드 추출을 실시간으로 수행하고 제품에 대한 의견을 얻을 수 있다.

통계적 접근

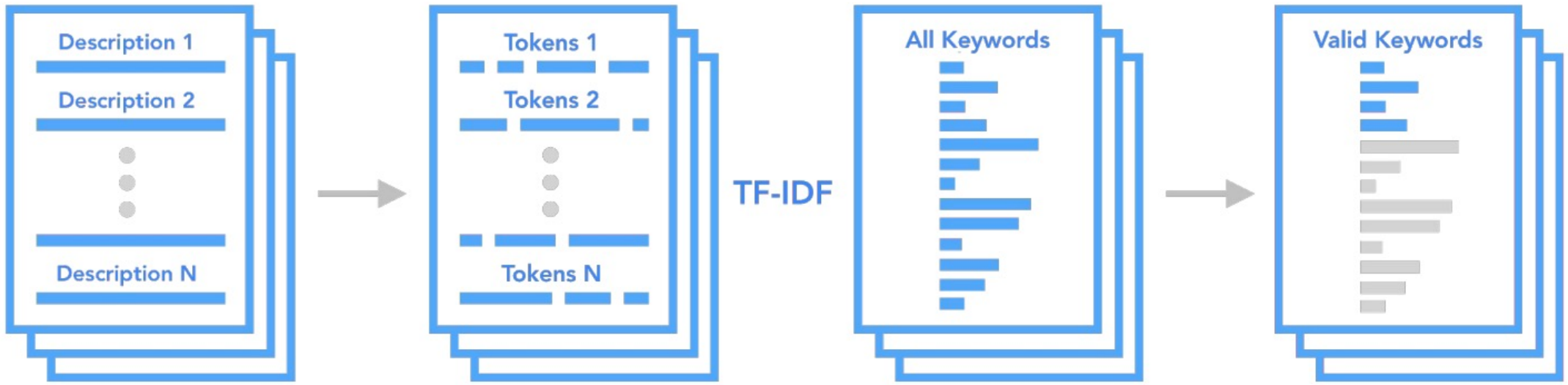
- 단어빈도
  - 단어의 등장 **빈도를 활용**하여 **중요 단어를 추출**
  - 단어 빈도 접근 방식은 문서를 단순한 **단어 모음으로 간주**
  - 단어의 의미, 구조, 문법 및 단어 순서를 고려하지 않음

- 연어 / 동시발생
  - 단어의 의미구조를 이해하기 위해서 N-gram과 같은 통계기법을 활용하여 연어나 동시발생 단어를 하나의 단어로 처리 할 수 있음
  - **연어는 연이어 함께 자주 등장하는 단어 묶음**. 예, "고객 서비스"
  - 동시 발생(co-occurrence)은 **동일 코퍼스 내에 함께 등장하는 단어 묶음**. 연어와 다르게 반드시 단어가 인접할 필요 없음.

TF-IDF

TF-IDF

TF-IDF 활용 핵심 키워드 추출



토큰화

문서 집합을
문서별로 토큰화

TF-IDF 계산

문서의 단어 별 TF-IDF
score 계산

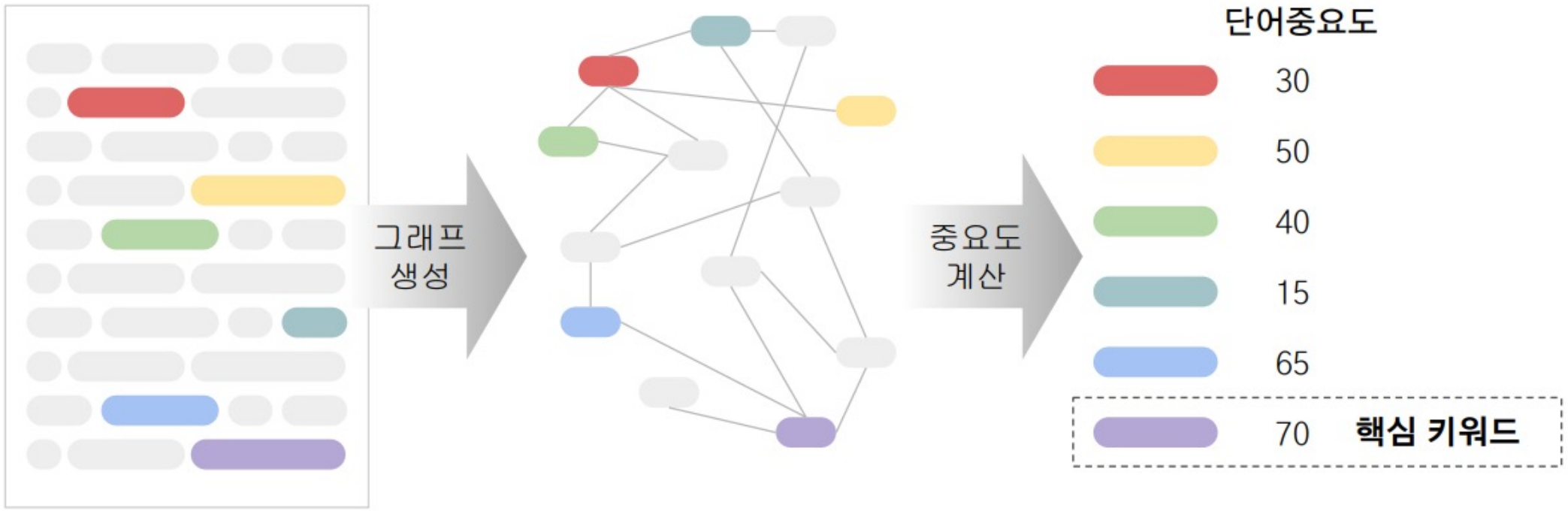
**TF-IDF score가
높은 순으로 추출**

TF-IDF score가 높은
순으로 상위부터 추출

TextRank

텍스트랭크

TextRank



Google PageRank

– The PageRank Citation Ranking: Bringing Order to the Web

1 Introduction and Motivation

The World Wide Web creates many new challenges for information retrieval. It is very large and heterogeneous. Current estimates are that there are over 150 million web pages with a doubling life of less than one year. More importantly, the web pages are extremely diverse, ranging from "What is Joe having for lunch today?" to journals about information retrieval. In addition to these major challenges, search engines on the Web must also contend with inexperienced users and pages engineered to manipulate search engine ranking functions.

However, unlike "flat" document collections, the World Wide Web is hypertext and provides considerable auxiliary information on top of the text of the web pages, such as link structure and link text. In this paper, we take advantage of the link structure of the Web to produce a global "importance" ranking of every web page. This ranking, called PageRank, helps search engines and users quickly make sense of the vast heterogeneity of the World Wide Web.

Google PageRank

- The PageRank Citation Ranking: Bringing Order to the Web

1.2 PageRank

In order to measure the relative importance of web pages, we propose PageRank, a method for computing a ranking for every web page based on the graph of the web. PageRank has applications in search, browsing, and traffic estimation.

Section 2 gives a mathematical description of PageRank and provides some intuitive justification. In Section 3, we show how we efficiently compute PageRank for as many as 518 million hyperlinks. To test the utility of PageRank for search, we built a web search engine called Google (Section 5). We also demonstrate how PageRank can be used as a browsing aid in Section 7.3.

Google PageRank

- The PageRank Citation Ranking: Bringing Order to the Web

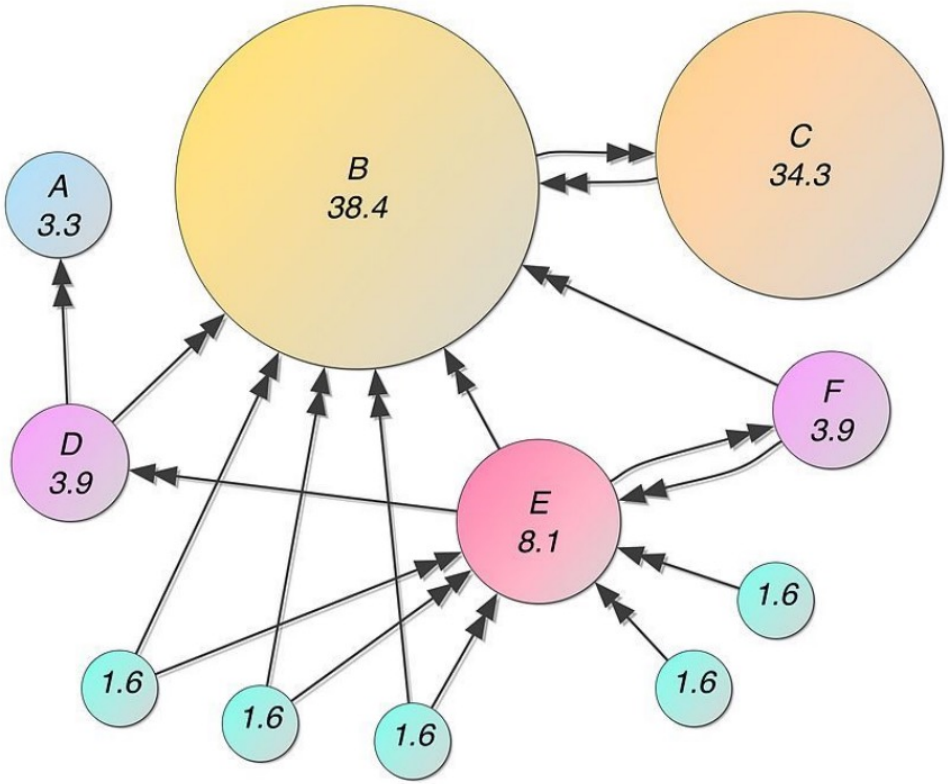
2.2 Link Structure of the Web

Web pages vary greatly in terms of the number of backlinks they have. For example, the Netscape home page has 62,804 backlinks in our current database compared to most pages which have just a few backlinks. Generally, highly linked pages are more "important" than pages with few links. Simple citation counting has been used to speculate on the future winners of the Nobel Prize [San95]. PageRank provides a more sophisticated method for doing citation counting.

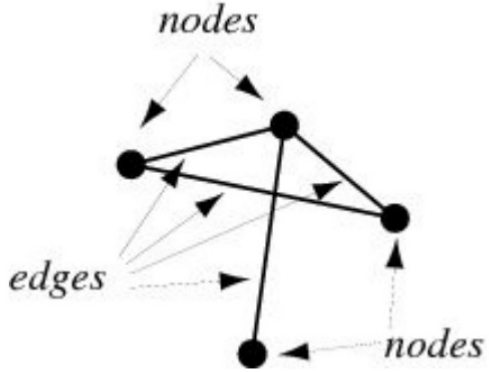
The reason that PageRank is interesting is that there are many cases where simple citation counting does not correspond to our common sense notion of importance. For example, if a web page has a link off the Yahoo home page, it may be just one link but it is a very important one. This page should be ranked higher than many pages with more links but from obscure places. PageRank is an attempt to see how good an approximation to "importance" can be obtained just from the link structure.

Google PageRank

– 인터넷상의 '링크 구조'를 통해 웹페이지의 중요도를 결정하여, 인터넷 검색 시 양질의 결과를 얻을 수 있도록 하는 알고리즘



그래프



Google PageRank

$$PR(A) = (1-d) + d\,(PR(T_1)/C(T_1) + \cdots + PR(T_n)/C(T_n))$$

A : PageRank 점수를 계산할 웹페이지(노드)

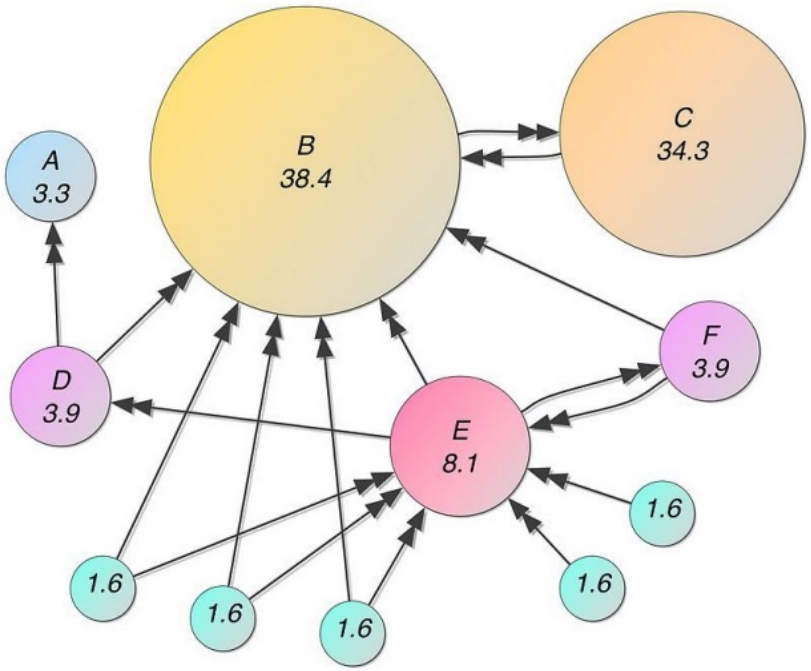
T : A를 링크하고 있는 웹페이지(노드)

PR(T) : 웹페이지T의 PageRank 점수

C(T) : 웹페이지T에서 연결된 링크 총 갯수

PR(T)/C(T) : T의 PageRank를 링크수로 나눈값

d : Damping factor



- 웹페이지 A의 pagerank는 A를 인용하고있는 T_1, T_2, ..., T_n의 pagerank를 정규화한 값의 합

TextRank: Brining Order into Texts

- https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf

0. Abstract

- In this paper, we introduce TextRank – a graph-based ranking model for text processing, and show how this model can be successfully used in natural language applications. In particular, we propose two innovative unsupervised methods for keyword and sentence extraction, and show that the results obtained compare favorably with previously published results on established benchmarks.

- Textrank는 그래프 기반 Ranking 모델 (구글의 PageRank를 텍스트분석에 적용한 모델)
- 키워드와 문장 추출을 위한 비지도 학습 방법을 제안

1. Introduction

- Graph-based ranking algorithms like Kleinberg's HITS algorithm (Kleinberg, 1999) or Google's PageRank (Brin and Page, 1998) have been successfully used in citation analysis, social networks, and the analysis of the link-structure of the World Wide Web. Arguably, these algorithms can be singled out as key elements of the paradigm-shift triggered in the field of Web search technology, by providing a Web page ranking mechanism that relies on the collective knowledge of Web architects rather than individual content analysis of Web pages. In short, a graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information.

- 그래프 기반 랭킹 알고리즘은 구글 PageRank에서 사용됨 (웹 분석에 성공적으로 적용)
- 그래프 기반 랭킹 알고리즘은 그래프의 각 vertex(node)의 중요도를 결정하는 방법

2. The TextRank Model

- Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex...

- 그래프 기반 랭킹 알고리즘은 그래프 속 vertex(노드)의 중요도를 결정하는 방법
- 기본 아이디어는 (하나의 vertex가 다른 vertex에 연결되면) "투표" 혹은 "추천"
- 많은 득표를 한 vertex가 중요한 vertex임을 의미

2 The TextRank Model

TextRank

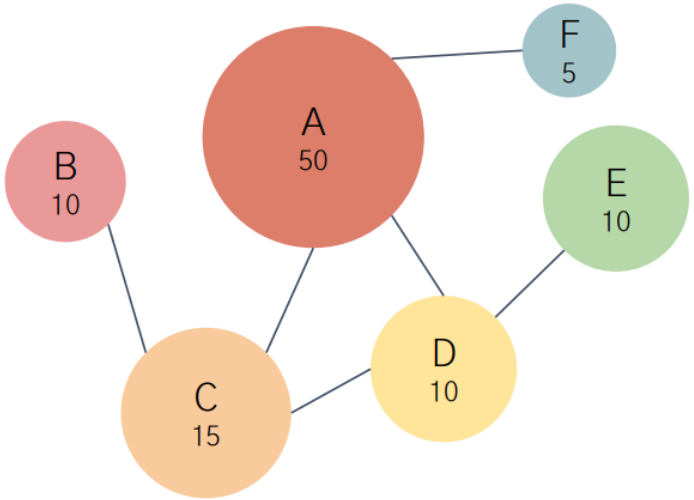
계산할 단어 노드의
스코어

Vi로 들어오는(In)
Edge 가중치의 합산

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

Damping Factor

Vi의 인접 Vertex에서 나오는
(out) edge의 가중치



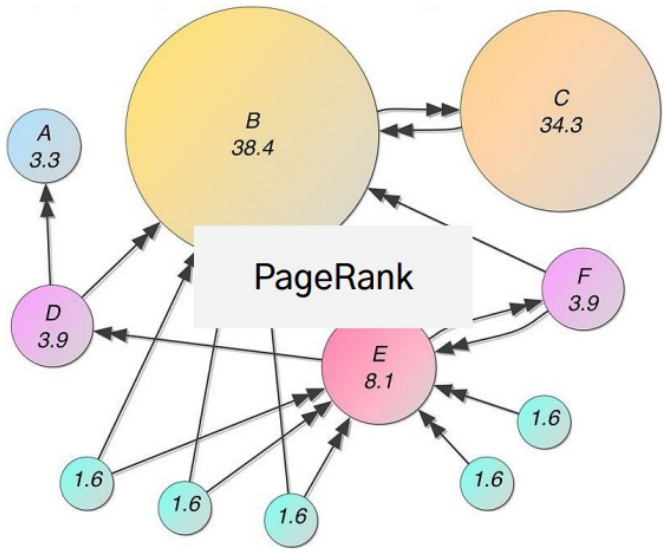
$$PR(A) = (1-d) + d(PR(T_1)/C(T_1) + \cdots + PR(T_n)/C(T_n))$$

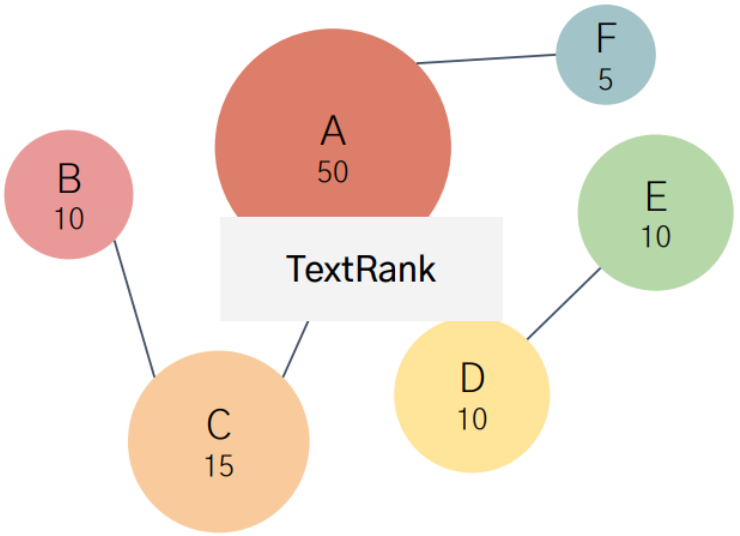
PageRank

2.1 Undirected Graphs

– Although traditionally applied on directed graphs, a recursive graph-based ranking algorithm can be also applied to undirected graphs, in which case the out-degree of a vertex is equal to the in-degree of the vertex. For loosely connected graphs, with the number of edges proportional with the number of vertices, undirected graphs tend to have more gradual convergence curves.
Figure 1 plots the convergence curves for a randomly generated graph with 250 vertices and 250 edges, for a convergence threshold of 0.0001. As the connectivity of the graph increases (i.e. larger number of edges), convergence is usually achieved after fewer iterations, and the convergence curves for directed and undirected graphs practically overlap.

– Undirected 그래프의 경우 더 점진적 수렴 곡선을 가짐 (더 빠르게 수렴한다)

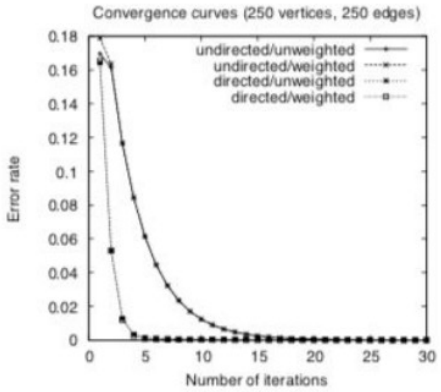




2.2 Weighted Graphs

– However, in our model the graphs are build from natural language texts, and may include multiple or partial links between the units (vertices) that are extracted from text. It may be therefore useful to indicate and incorporate into the model the "strength" of the connection between two vertices and as a weight added to the corresponding edge that connects the two vertices...

– undirected / weighted graph를 사용 시 더 적은 iteration으로 0에 수렴하는 것을 볼 수 있음



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

2.3 Text as a Graph

1. Identify text units that best define the task at hand, and add them as vertices in the graph.
2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.
3. Iterate the graph-based ranking algorithm until convergence.
4. Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

In the following, we investigate and evaluate the application of TextRank to two natural language processing tasks involving ranking of text units: (1) A keyword extraction task, consisting of the selection of keyphrases representative for a given text; and (2) A sentence extraction task, consisting of the identification of the most "important" sentences in a text, which can be used to build extractive summaries.

- 주어진 텍스트의 핵심 키워드를 추출할 수 있음
- 주어진 텍스트의 핵심 문장을 식별할 수 있음

3 Keyword Extraction

– The task of a keyword extraction application is to automatically identify in a text a set of terms that best describe the document. Such keywords may constitute useful entries for building an automatic index for a document collection, can be used to classify a text, or may serve as a concise summary for a given document. Moreover, a system for automatic identification of important terms in a text can be used for the problem of terminology extraction, and construction of domain-specific dictionaries… The simplest possible approach is perhaps to use a frequency criterion to select the "important" key words in a document.

– 키워드 추출은 문서를 대변할 수 있는 단어 집합을 자동으로 식별하는 것

3.1 TextRank for Keyword Extraction

– The TextRank keyword extraction algorithm is fully unsupervised, and proceeds as follows. First, the text is tokenized, and annotated with part of speech tags – a preprocessing step required to enable the application of syntactic filters… Next, all lexical units that pass the syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of words. After the graph is constructed (undirected unweighted graph), the score associated with each vertex is set to an initial value of 1, and the ranking algorithm described in section 2 is run on the graph for several iterations until it converges– usually for 20–30 iterations, at a threshold of 0.0001.… For this example, the lexical units found to have higher "importance" by the TextRank algorithm are (with the TextRank score indicated in parenthesis): numbers (1.46), inequations (1.45), linear (1.29), dio phantine (1.28), upper (0.99), bounds (0.99), strict (0.77)

– 1단계 : 텍스트는 품사가 태깅되어 토큰화됨
– 2단계 : 단어 윈도 (window of words)에 동시 등장한 토큰 사이는 엣지를 추가하여 그래프를 생성
– 3단계 : 0.0001을 threshold로 20–30회 반복

3.2 Evaluation

Method	Assigned		Correct		Precision	Recall	F-measure
	Total	Mean	Total	Mean			
TextRank							
Undirected, Co-occ.window=2	6,784	13.7	2,116	4.2	**31.2**	43.1	**36.2**
Undirected, Co-occ.window=3	6,715	13.4	1,897	3.8	28.2	38.6	32.6
Undirected, Co-occ.window=5	6,558	13.1	1,851	3.7	28.2	37.7	32.2
Undirected, Co-occ.window=10	6,570	13.1	1,846	3.7	28.1	37.6	32.2
Directed, forward, Co-occ.window=2	6,662	13.3	2,081	4.1	31.2	42.3	35.9
Directed, backward, Co-occ.window=2	6,636	13.3	2,082	4.1	31.2	42.3	35.9
Hulth (2003)							
Ngram with tag	7,815	15.6	1,973	3.9	25.2	**51.7**	33.9
NP-chunks with tag	4,788	9.6	1,421	2.8	29.7	37.2	33.0
Pattern with tag	7,012	14.0	1,523	3.1	21.7	39.9	28.1

Table 1: Results for automatic keyword extraction using TextRank or supervised learning (Hulth, 2003)

- Textrank 방식이 정밀도(Precision)와 F-measure에서 가장 높지만 재현율(Recall)은 지도 학습 방법에 비해 높지 않음
- 그리고 windows가 클수록 정확도는 낮아짐 (= 멀리 떨어져있는 단어가 관계를 정의할 만큼 강력하지 않음)

5 Why TextRank Works

– Intuitively, TextRank works well because it does not only rely on the local context of a text unit (vertex), but rather it takes into account information recursively drawn from the entire text (graph) … The sentences that are highly recommended by other sentences in the text are likely to be more informative for the given text, and will be therefore given a higher score… Through its iterative mechanism, TextRank goes beyond simple graph connectivity, and it is able to score text units based also on the "importance" of other text units they link to. The text units selected by TextRank for a given application are the ones most recommended by related text units in the text, with preference given to the recommendations made by most influential ones, i.e. the ones that are in turn highly recommended by other related units. The underlying hypothesis is that in a cohesive text fragment, related text units tend to form a "Web" of connections that approximates the model humans build about a given context in the process of discourse understanding.

– 텍스트 단위 (버텍스)의 로컬 컨텍스트를 고려할뿐만 아니라, 전체 텍스트(그래프)에서 재귀적으로 정보를 고려하기 때문에 Textrank가 잘 작동
– 링크하는 다른 텍스트 단위의 "중요성"을 바탕으로 텍스트 단위를 평가

6 Conclusions

– In this paper, we introduced TextRank – a graph- based ranking model for text processing, and show how it can be successfully used for natural language applications. In particular, we proposed and evaluated two innovative unsupervised approaches for keyword and sentence extraction, and showed that the accuracy achieved by TextRank in these applications is competitive with that of previously proposed state-of-the-art algorithms. An important aspect of TextRank is that it does not require deep linguistic knowledge, nor domain or language specific annotated corpora, which makes it highly portable to other domains, genres, or languages.

– Textrank는 깊은 언어 지식이나 도메인 별 corpora를 필요로 하지 않고, 다른 도메인, 장르, 언어에 적용할 수 있음

Textrank 과정 : 순서

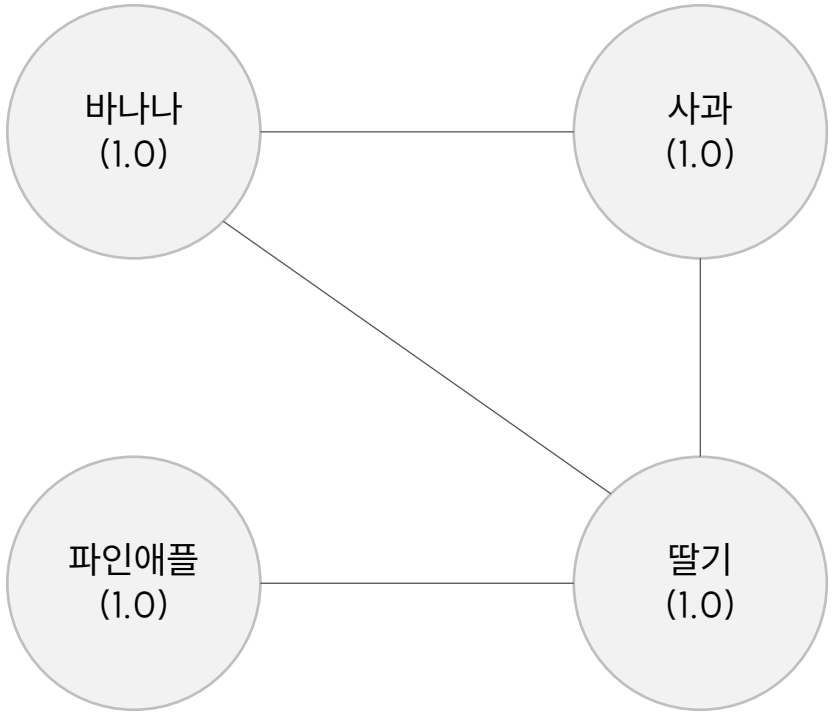
딸기 바나나 사과 딸기 파인애플

- 1단계 : 그래프 생성하기
- 2단계 : Score 계산하기
- 3단계 : 키워드 추출하기

Textrank 과정 : 그래프 생성

딸기	바나나	사과	딸기	파인애플

- window_size=2로 sliding 하며 그래프를 생성한다. (노드 초기값은 1)



Textrank 과정 : 행렬로 계산하기

계산할 단어 노드의 스코어

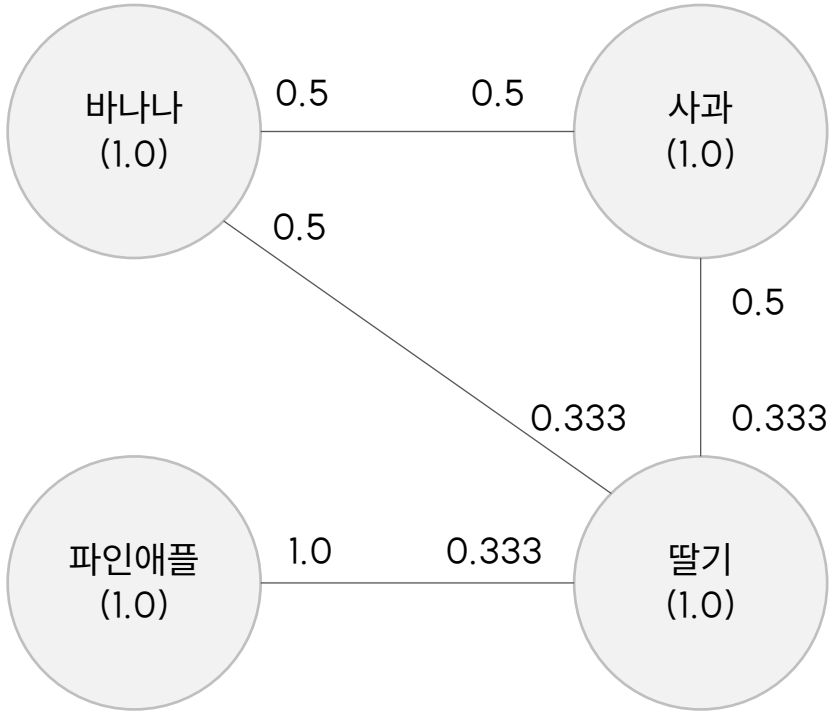
Vi로 들어오는(In) Edge 가중치의 합산

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

Damping Factor

Vi의 인접 Vertex에서 나오는 (out) edge의 가중치

Textrank 과정 : 행렬로 계산하기



노드	스코어
바나나	1.0
사과	1.0
파인애플	1.0
딸기	1.0

노드	바나나	사과	파인애플	딸기
바나나	0	0.5	0	0.5
사과	0.5	0	0	0.5
파인애플	0	0	0	1.0
딸기	0.333	0.333	0.333	0
	0.858	0.858	0.433	1.85

Textrank 과정 : 행렬로 계산하기

노드 별 스코어

노드	스코어
바나나	1.0
사과	1.0
파인애플	1.0
딸기	1.0

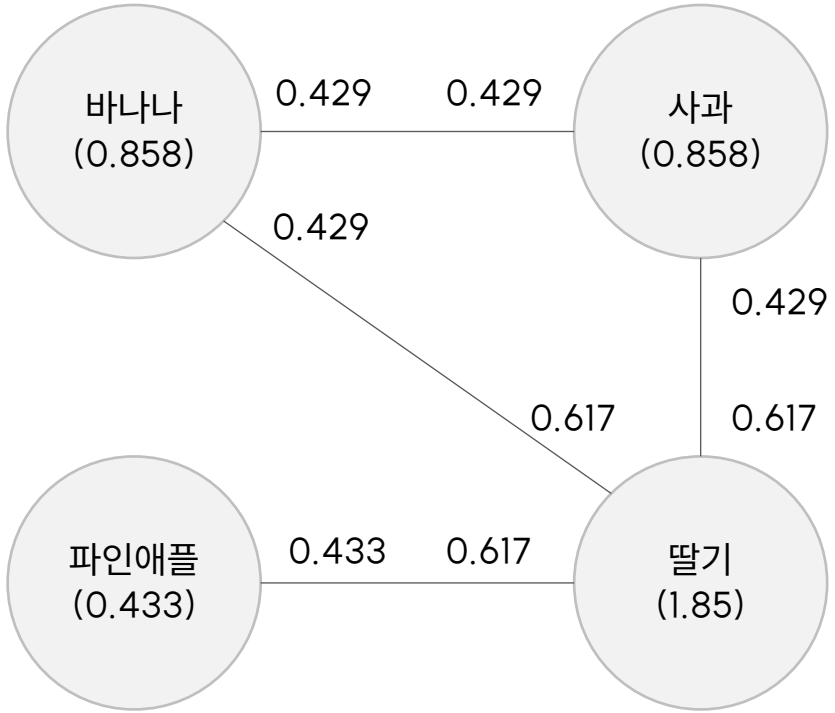
노드간 엣지 가중치 행렬 (그래프)

노드	바나나	사과	파인애플	딸기
바나나	0	0.5	0	0.5
사과	0.5	0	0	0.5
파인애플	0	0	0	1.0
딸기	0.333	0.333	0.333	0
	0.858	0.858	0.433	1.85

- 스코어 계산

S(바나나) = (1−0.85) + 0.85 x (0.5 + 0.333) = 0.858
S(사과) = (1−0.85) + 0.85 x (0.5 + 0.333) = 0.858
S(파인애플) = (1−0.85) + 0.85 x (0.333) = 0.433
S(딸기) = (1−0.85) + 0.85 x (0.5 + 0.5 + 1) = 1.85

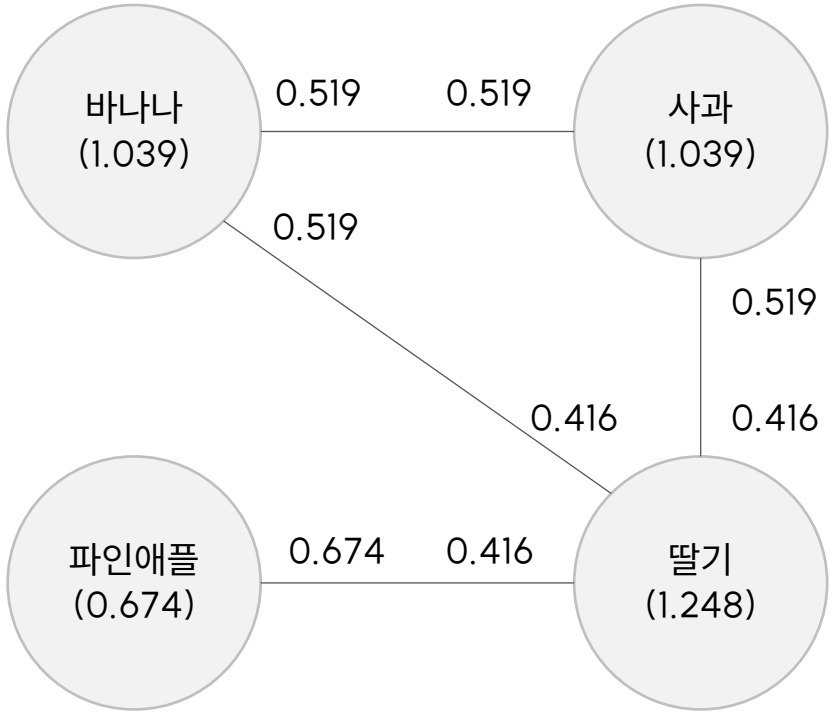
Textrank 과정 : 행렬로 계산하기



노드	이전 스코어	현 스코어
바나나	0.858	1.039
사과	0.858	1.039
파인애플	0.433	0.674
딸기	1.85	1.248

노드	바나나	사과	파인애플	딸기
바나나	0	0.429	0	0.429
사과	0.429	0	0	0.429
파인애플	0	0	0	0.433
딸기	0.617	0.617	0.617	0

Textrank 과정 : 행렬로 계산하기



노드	이전 스코어	현 스코어
바나나	1.039	0.945
사과	1.039	0.945
파인애플	0.674	0.504
딸기	1.248	1.606

노드	바나나	사과	파인애플	딸기
바나나	0	0.519	0	0.519
사과	0.519	0	0	0.519
파인애플	0	0	0	0.674
딸기	0.416	0.416	0.416	0

Textrank 과정 : 계산 완료