SMART UP

길진성 김태성 김현욱 정선진

주간 활동 보고서

목차

주간 진행 사항

- 회의 & 활동 진행 시간 및 내용
- 개선된 텍스트 요약 알고리즘 개발

<주간 활동 보고서 10 주차>
-2024.05.06 ~ 2024.05.12-

-주간 진행 사항

<회의 및 활동 진행 시간>
2024.05.06 (월) 14:00 ~ 22:00 / 2024.05.08 (수) 15:00 ~ 19:00

- 총 활동 시간 : 12 시간 00 분

-회의 및 활동 내용

>> [스마트업] 디스코드 오프라인 개발 (05.06)~(05.08)

-espnet 적용을 위해 라즈베안 os 안에 우분투 os 를 멀티 os 로 설치

-입력 장치를 통해 test.wav 파일 생성 후 espnet 으로 stt 전환

-요약 알고리즘 알고리즘 구현을 위한 기존 - 알고리즘들의 구성 코드 추출

-데이터 멀티프로세싱을 위해 스택내에 데이터 분할 알고리즘 개발 진행중

-equal partitioning 을 통해 알고리즘 구현
05.10 오프라인 중간 발표 준비

개선된 텍스트 요약 알고리즘 개발

저희는 가벼운 단일 알고리즘보다 더 성능이 좋지만 마찬가지로 가볍게 작동시킬 수 있는 알고리즘 개발을 위하여 두가지 기술을 혼합하여 개발하기로 계획 하였습니다.

사용하고자 하는 알고리즘은 추출 기반 요약 알고리즘으로 이 방법은 원본 텍스트에서 중요한 문장이나 구절을 추출 하여 요약하는 것입니다.

이 중요도를 결정하기 위한 기준으로 단어의 빈도나 문장의 길이, 키워드의 중요성 등을 고려하여 결정하고

PAGERANK, TF-IDF 같은 기술들이 존재 합니다.

저희는 PageRank 알고리즘을 변형한 TextRank 알고리즘과 TF-IDF 를 결합하여
더 좋은 성능의 알고리즘을 개발하고자 합니다.

우선 TextRank 는 상술한대로 PageRank 알고리즘의 변형으로 문장간 유사성을 측정하여 중요한 문장을 추출합니다.

텍스트 랭크는 페이지 랭크와 유사하게 작동하는데 그 작동 방식은 다음과 같습니다.

1. 텍스트를 그래프로 변환:

    주어진 텍스트를 문장 단위로 분할한후,

    각 문장을 노드로 표현하는 그래프를 생성합니다. 이 생성을 할 때,
    문장 간의 유사도에 따라 간선으로 연결됩니다.

2. 유사도 계산:

    문장 간의 유사도를 측정하기 위해 코사인 유사도 또는 다른 유사성 메트릭을 사용합니다.

    유사도를 기준으로 그래프의 간선에 가중치를 할당합니다.

3. 그래프 랭킹:

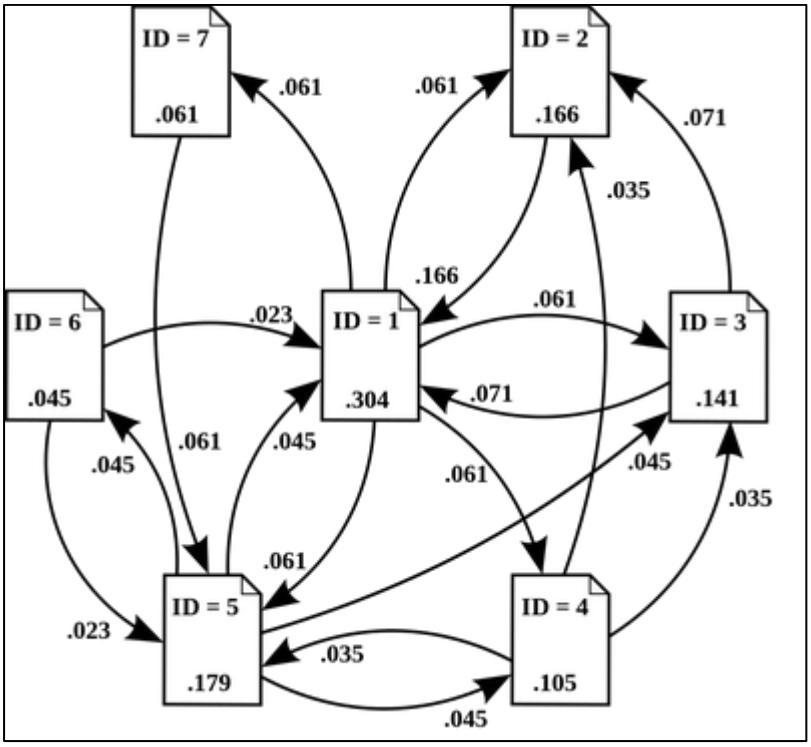
    그래프의 각 노드(문장)에 대해 페이지 랭크와 유사한 방법으로 중요도를 계산합니다.

각 노드의 중요도는 해당 노드와 연결된 다른 노드들의 중요도에 따라 조정됩니다.

4. 중요한 문장 추출:

계산된 중요도에 따라 문장을 순위화하고, 사용자가 원하는 요약 길이에 맞게 문장을 선택합니다.

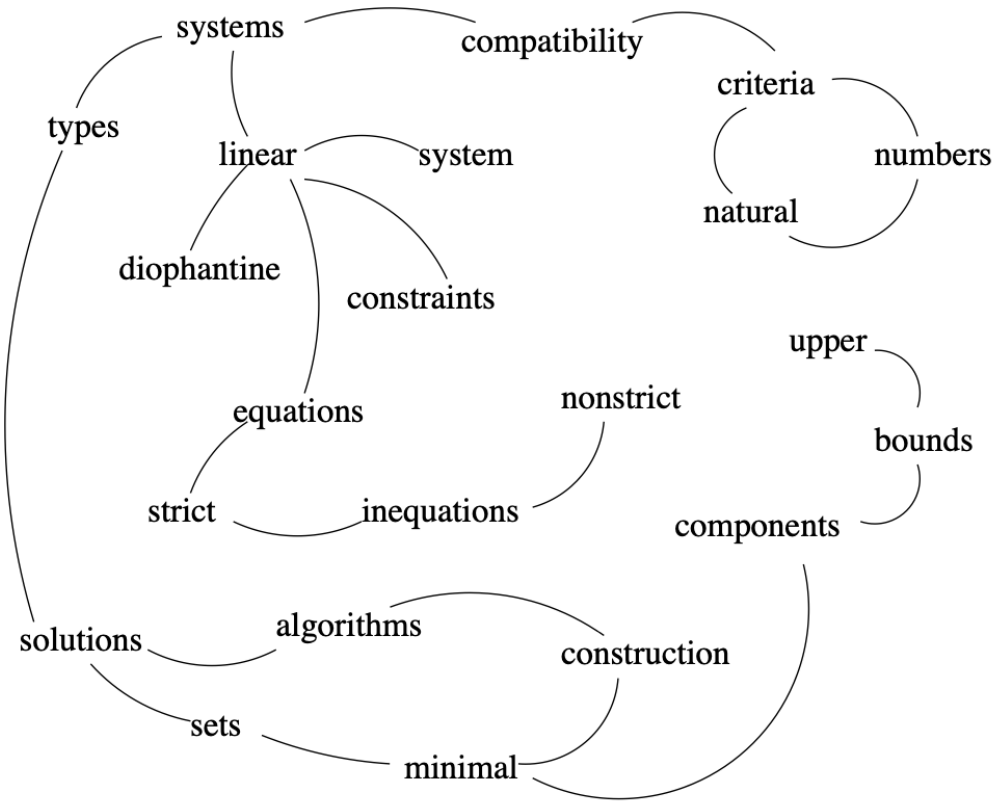
텍스트 랭크 다음과 같은 그림으로 표현 할 수 있습니다.



각 노드는 문장에 해당하고 가운데 노드의 .304 를 가중치를 의미합니다.
여러 문장이 ID=1 을 참조하여서 가중치가 가장 높고 핵심 요약으로 사용하기 알맞은 문장이 됩니다.

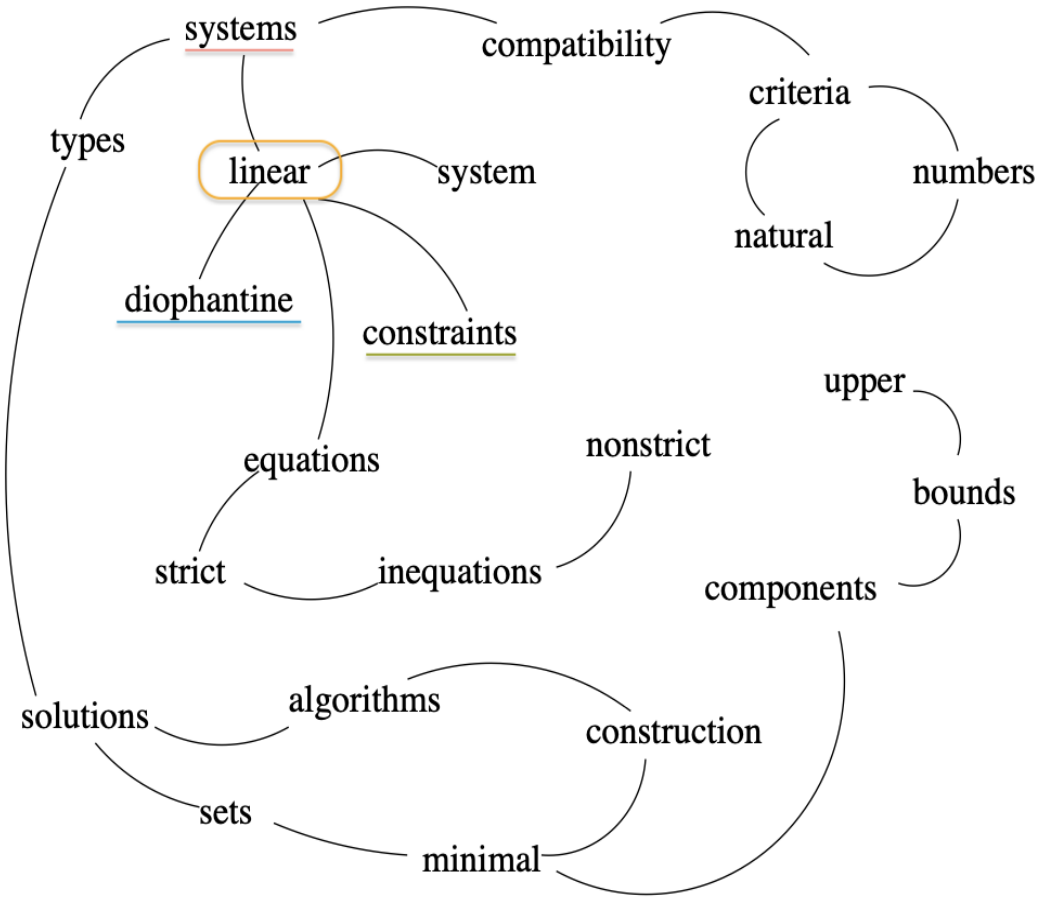
텍스트 랭크에서 단어의 가중치를 설정하고 추출하는 것이 실제로 작동되는 예시는 다음과 같습니다.

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.



각 문장들의 단어에서 단어가 어떤 것을 참조하는지 그래프로 표현하고 이에 따라 가중치를 알수 있게 됩니다.

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.



이런 식으로 중요 단어를 추출 할 수 있게 되고 문장에서도 같은 방식으로 가중치를 설정합니다.

이와 결할 할 기술로는 TF-IDF 로 결정했습니다.

1. 단어 빈도 (Term Frequency, TF):
    2. 문서 내에서 한 단어가 얼마나 자주 등장하는지를 측정합니다.
    3. 이는 해당 단어가 문서 내에서 중요한 역할을 하는지를 파악하는 데 사용됩니다.
    4. 일반적으로 문서 내에서 한 단어의 빈도는 해당 단어가 나타나는 횟수로 측정됩니다.
5. 역문서 빈도 (Inverse Document Frequency, IDF):
    6. 해당 단어가 전체 문서 집합에서 얼마나 일반적인지를 측정합니다.
    7. 이는 해당 단어가 특정 문서에서 중요한지를 판단하는 데 사용됩니다.
    8. 일반적으로 역문서 빈도는 다음과 같이 계산됩니다: 전체 문서의 수를 해당 단어를 포함하는 문서의 수로 나눈 후, 그 결과에 로그를 취합니다.

TF-IDF 는 단어 빈도와 역문서 빈도를 측정하여 단어의 중요성을 결정 할수 있는 방법으로
TF(Term Frequency, 단어빈도)는 문서내에서 한 단어가 얼마나 자주 등장하는지를 측정하고
해당 단어가 문서 내에서 중요한 역할을 하는지 파악합니다

IDF(Inverse Document Frequency,역문서 빈도) 는 해당 단어가 전체 문서 집합에서 얼마나
일반적인지를 측정하고 해당 단어가 문서에서 중요한지를 판단합니다.

TF-IDF 는 은,는,이,가 같은 조사나 자주 사용되지만 의미가 적은 단어들을 걸러내고
중요단어를 추출하는데 도움을 줍니다.

<TF-IDF 계산 방법>

$$TF(t, d) = \frac{\text{문서 d 에서 단어 t 가 등장한 횟수}}{\text{문서 d 에 등장한 모든 단어의 수}}$$

$$IDF(t, D) = \log\left(\frac{\text{총 문서의 개수}}{\text{단어 t 를 포함하는 문서의 수}}\right)$$

$$TF\text{-}IDF(t,d,D) = TF(t, d) * IDF(t, D)$$

모든 문서에서 등장하는 단어는 중요도가 낮으며, 특정 문서에서만 자주 등장하는 단어는 중요도가 높습니다.

텍스트 랭크와 TF-IDF 를 결합하는것은 다음과 같은 장점이 있다 생각하여 각각을 조합하여 개발할 예정입니다.

1. 다양한 정보 고려:

   TF-IDF 는 단어의 빈도와 역문서 빈도를 고려하여 중요한 단어를 추출하고, TEXTRANK 은 문장 간의 유사성을 고려하여 중요한 문장을

추출합니다. 두 알고리즘을 결합하면 단어 수준과 문장 수준에서의 중요성을 모두 고려할 수 있습니다.

2. 문맥 파악:

TF-IDF 는 각 단어의 중요성을 독립적으로 계산하지만, TEXTRANK 는 문장 간의 관계를 고려하여 중요한 문장을 선정합니다. 두 알고리즘을 함께 사용하면 단어의 중요성과 문장의 중요성을 모두 고려하여 요약을 생성하므로 문맥을 더 잘 파악할 수 있습니다.

3. 요약의 정확성 향상:

TF-IDF 와 TEXTRANK 은 각각 단어의 중요성과 문장의 중요성을 고려하여 요약을 생성합니다. 이를 결합하면 보다 정확한 요약을 생성할 수 있으며, 중요한 정보를 누락시키지 않고 요약할 수 있습니다.