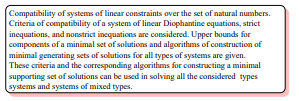
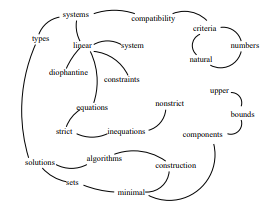
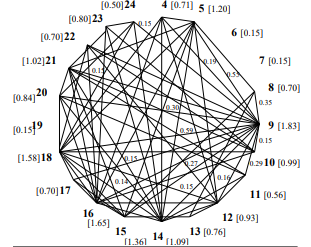
TextRank : Bringing Order into Text 리뷰

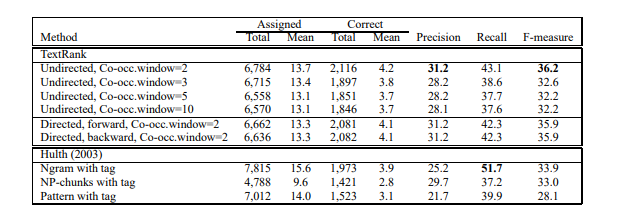
1. Introduction

* Graph
  + Vertex(node) 집합과 edge 집합으로 이루어진 자료구조
  + 위 vertex간 연결관계를 바탕으로 중요도 점수(score) 산출 -> 특정 vertex의 score는 연결된 다른 vertex들의 중요도 점수에 edge(weight)를 반영하여 산출
  + 이 중요도 점수를 반복적으로 연산하여 vertex마다 순위를 매기는 것 -> **그래프 기반 랭킹 알고리즘**
* History
  + 본래 웹페이지 관계 분석을 통한 페이지별 ranking이 목적
  + 각 페이지의 rank(중요도)는 연결된 다른 페이지의 중요도를 반영하여 계산
    - 특정 페이지가 중요 페이지들에 의해 연결된다면, 해당 페이지 점수를 높게 부여
  + 대표적인 방법
    - HITS(Hyperlink Induced Topic Search, Kleinberg, 1998)
    - PageRank(Brin and Page, 1998)
      * (directed, unweighted) graph 만 다룸(directed : 방향성 부여, unweighted : edge에 weight 미부여)
  + TextRank의 경우, Text에 특화된 그래프 기반 랭킹 알고리즘으로, Text unit(단어 ,문장 등)간 연결 관계를 찾아 그래프로 표현하고, 전체 Text에서 도출된 knowledge를 반영하여 각 unit의 중요도(ranking) 산출

2. TextRank

* 본 논문에서는 크게 키워드 추출, 문장 추출(문서 요약) 테스크에 대해 적용
* PageRank는 (directed, unweighted) 만 다룬 것과 다르게, TextRank에서는 (directed, weighted), (undirected, weighted), (directed, unweighted), (undirected, unweighted) 4가지 케이스에 모두 적용 가능
* 문서 내 주요 키워드 추출
  + 1. Preprocessing
       - Tokenizing
         * 문서/문장을 토큰 단위로 분리
         * 전치사, 관사 제외
         * 불용어 제외
       - POS
         * 토큰에 품사 태깅
    2. Syntactic Filtering
       - 모든 토큰들에 대해 Vertex를 생성하면 그래프가 너무 방대해지므로, 일부 품사/단어집합에 속하는 단어로만 Vertex 생성
       - 본 논문에서는 명사, 형용사 조합이 가장 좋은 성능을 보이는 것으로 확인
    3. Co-occurrence relation(윈도우 사이즈 내 동시 등장)에 따라 Edge 생성
       - 특정 토큰에 대해 Window size N개 이내 동시 등장하는 토큰을 vertex로 하여 두 토큰을 edge로 연결
       - 본 논문에서 N=2~10 사이를 권장하며 논문에서는 2를 사용(아래 예시)  
           
         
    4. Importance Score 반복 산출(iteration)
       - (directed, unweighted)
         * 중요도 산출 공식  
           
         * i : 현재 기준 vertex indx, j : 현재 기준 vertex와 연결된 vertex index  
           d : randomness를 부여하는 damping factor(0~1 사이, 논문은 0.85)  
           초기 score: 1 or 랜덤 지정  
           모든 vertex의 점수가 수렴할 때까지 반복  
           수렴조건 : (현재 iter 중요도) – (이전 iter 중요도) < threshold
         * 예를들어 ‘criteria’의 중요도는 ‘criteria’ vertex로 화살표 가 향하는 ‘natural’, ‘numbers’ vertex 중요도 점수로 계산이 됨  
           즉, 위 시그마에서 In(V ‘criteria’) = {V ‘natural’, V ‘numbers’} 가 됨  
           따라서, natural의 중요도에 natural의 out집합 원소 개수만큼 나누고, numbers의 중요도에 numbers의 out 집합 원소 개수만큼 나눈 후, 더하여 damping factor 적용하면 criteria 중요도가 산출됨
       - (undirected, unweighted)
         * directed와 대부분 동일하나, 방향성이 존재하지 않으므로, In(Vi) = Out(Vi)로 간주함
         * 예를 들어, 위에서 criteria의 중요도 계산시, ‘natural’, ‘numbers’ 만 고려했다면, 여기서는 ‘compatibility’도 고려함
       - (directed, weighted)
         * 중요도 산출 공식  
           
         * 각 edge마다 초기 random weights(0~10) 부여
         * unweighted와 크게 다르지 않으며, 각 in->out vertex간의 weight를 반영, 가중한 계산이 추가됨
       - (undirected, weighted)
         * directed와 대부분 동일하나, 방향성이 존재하지 않으므로, In(Vi) = Out(Vi)로 간주함
         * 예를 들어, 위에서 criteria의 중요도 계산시, ‘natural’, ‘numbers’ 만 고려했다면, 여기서는 ‘compatibility’도 고려함
    5. 최종 score를 바탕으로 Top N 단어 선정
    6. Top N 단어 중 서로 인접한 단어는 Multi-keyword로 묶음
* 문서 내 주요 문장 추출(문서 요약)
  + 1. Similarity Relation을 바탕으로 Vertex와 Edge 생성
       - 문서 내 모든 문장을 각 vertex로 지정
       - Co-occurrence는 문장-문장간 관계인 large context에 적합하지 않음
       - 따라서 문장 추출에서는 얼마나 많은 단어가 겹치는지 고려함  
           
         = (동시 등장 단어 개수) / 각 문장의 단어 개수 log 합  
         분모는 normalization factor 역할을 하여 긴 문장의 유사도만 지나치게 증가하지 않도록 제한함
       - 문서 내 모든 n개 문장에 대해 서로간 n by n 유사도 행렬이 생성됨(tfidf로 대체 가능)
       - 이후, min\_sim(최소 기준)을 넘는 문장 vertex 사이에만 edge를 생성(키워드 추출에서는 Co-occurrence relation를 고려하여 윈도우 내 동시 등장 사이 edge를 생성했었음)
       - 각 similarity에 비례하도록 edge에 가중치를 부여(weighted)하여, 아래와 같은 문장간 그래프생성(문장 추출은 weighted만 고려하며 문장 순서에 따라 directed graph도 구축 가능)   
         
    2. Weighted Importance Score 반복적으로 산출(Iteration)
       - 중요도 산출 공식(키워드와 동일)  
         
    3. 최종 score를 바탕으로 Top N 문장 선정한 후, 원래 순서로 sorting하여 요약문 생성

3. Evaluation

* 키워드 추출
  + ‘Computer science and Information Technology’ 저널에서 발췌한 500개 요약문 대상
  + 각 요약문마다 사람이 직접 생성한 2개 키워드셋 존재
    - Controlled keywords : 주어진 어휘집의 단어로 제한
    - Uncontrolled keywords : 제한 없이 자유롭게 생성
  + Metic
    - Precision, Recall, F1-score
  + Result
    - Undirected, window=2인 경우에 가장 높은 Pre, f1 기록
    - Window size를 키울수록 성능 하락 -> 멀리 위치한 단어일수록 연결성이 줄어들기 때문  
      
* 문장 추출
  + ‘Document Understanding Evaluations 2002’ 벤치마크의 567개 뉴스 기사 대상
    - 각 기사마다 100단어의 요약문 생성
  + Metric
    - Rouge score
  + Result
    - baseline : 첫 문장 포함 100단어 요약문
    - 비교 모델 : top5 performing systems among 15 different systems 상위 랭크 기록
    - stemmed, no-stopwords 버전에서 textrank의 요약문이 2위 기록  
      