Shortest path for Word segmentation

Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit, github.com/lovit/shortestpath

Word segmentation

- Word segmentation 은 주어진 문장을 단어열로 분해하는 문제입니다.
  - '이것은예문입니다' → [이것, 은, 예문, 입니다]

- 단어의 활용이 이뤄지지 않는다면, tokenization 과 같은 개념입니다.
  - 앞서 살펴본 Max Score Tokenizer 는 word segmentation 알고리즘 입니다.

Word segmentation

- Word segmentation 은 중국어 / 일본어에서 많이 연구가 되었습니다.
  - 띄어쓰기를 이용하지 않습니다.
  - 영어나 한국어처럼 활용 (conjugation) 이 일어나지 않습니다.
  - 한자어를 이용하기 때문에 글자에 대한 모호성이 적습니다.
    - 중국어에서 자주 이용되는 한자의 수는 약 7,000 자 입니다.
    - 한국어에서는 약 300 글자가 전체 문서의 99% 를 차지합니다.
- 중국어 / 일본어의 품사 판별 과정은 문장에서 단어를 분리하는 것입니다.

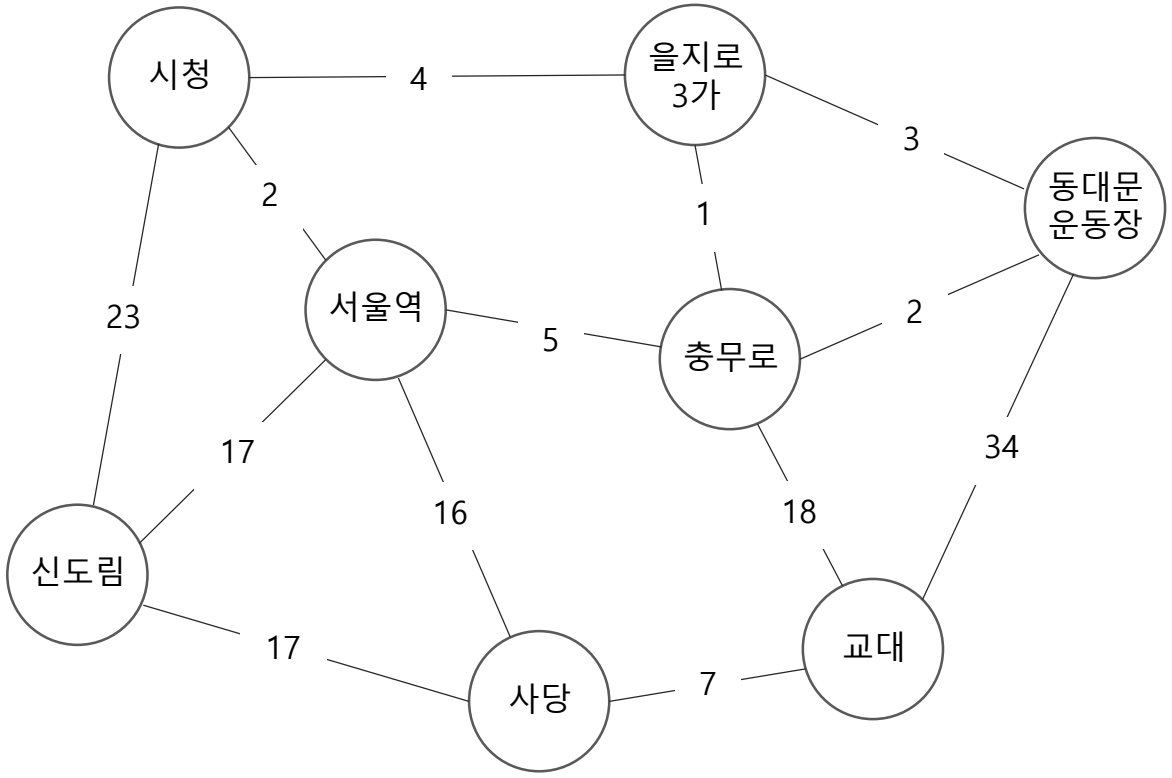
Shortest path in NLP

- 최단 경로 (Shortest path) 문제는 그래프에서 두 마디 사이를 연결하는 경로 중, 가장 가까운 경로를 찾는 문제입니다.
  - (예시) 출발 역에서 목적 역까지의 가장 빠른 지하철 노선을 찾는 문제

- Word segmentation (품사 판별 문제) 를 최단 경로 문제로 풀 수 있습니다.

Undirected graph

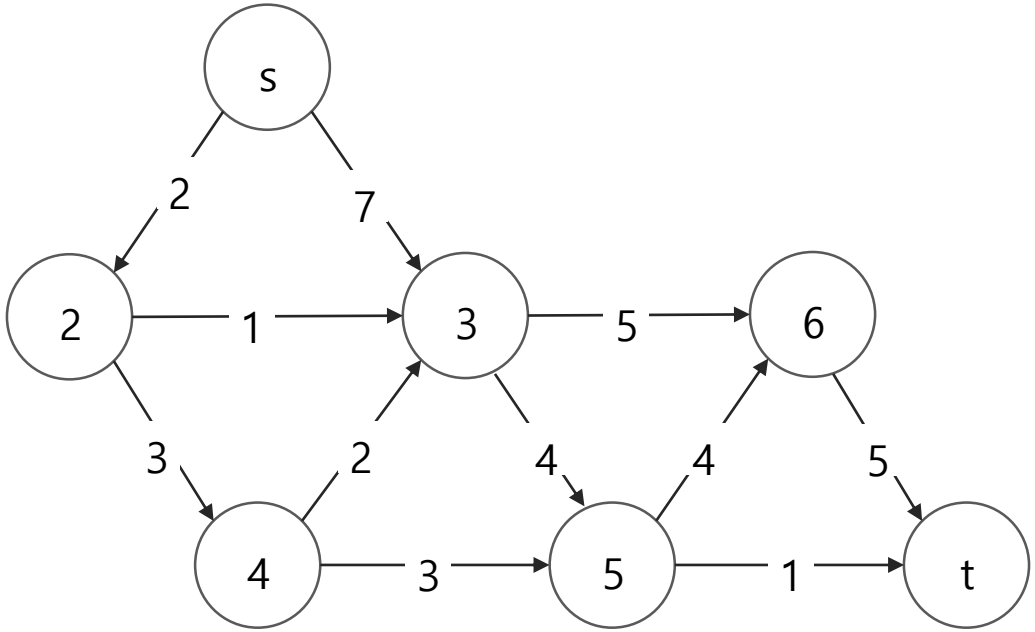
- 그래프의 두 마디가 서로 이동가능하며, 동일한 비용으로 연결되어 있다면 undirected graph 입니다.
- Undirected graph 에서는 edge 를 무방향 선으로 표현합니다.
- Edge 의 weight 는 두 마디를 연결하는 거리/비용 입니다.



(예시 출처) 경영과학: 기초부터 심화까지, 홍성필, 율곡출판사 (2014)

Directed graph

- Directed graph 는 두 마디가 서로 비대칭으로 연결되어 있는 그래프 입니다.
- 화살표로 이동 가능한 마디를 연결합니다.



(예시 출처) 경영과학: 기초부터 심화까지, 홍성필, 율곡출판사 (2014)

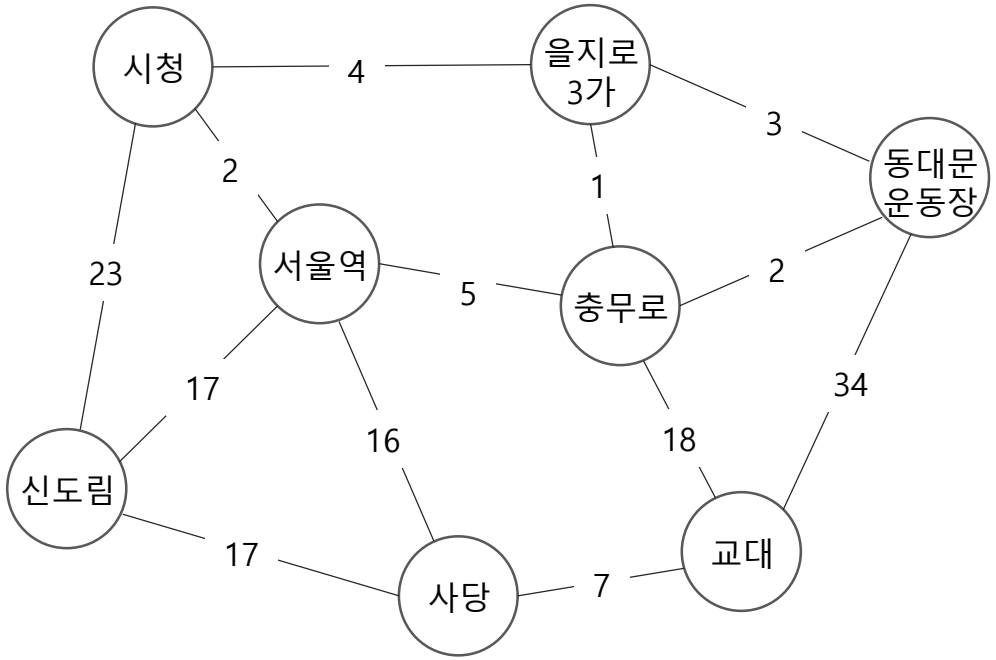
Ford algorithm

- Dijkstra 와 Ford 는 대표적인 최단 경로 탐색 알고리즘입니다.
  - Dijkstra 는 그래프의 edge weight 가 반드시 0 이상이어야 합니다.
  - Edge weight 가 음수가 될 수 있다면 Ford 를 이용하면 됩니다.

Ford algorithm

‘시청 → 교대‘ 의 최단 경로

- Initialize
  - 출발지로부터의 거리를 행렬 d 에 저장합니다.
  - 출발지를 제외한 모든 마디의
    거리를 최대값 (inf) 로 초기화 합니다.
  - 시청을 제외한 다른 역까지 가는 거리를
    모른다는 의미입니다.

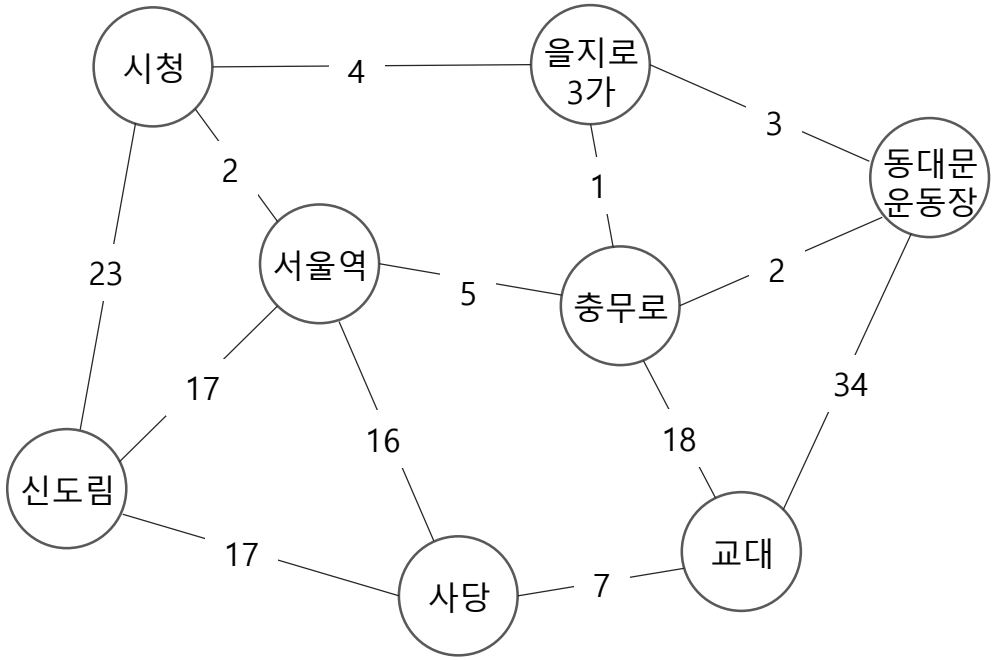


교대	동대문 운동장	사당	서울역	시청	신도림	을지로3가	충무로
-	-	-	-	0	-	-	-

Ford algorithm

‘시청 → 교대‘ 의 최단 경로

- Iteration
  - d[U] + w(U, V) < d[V] 이면 V 까지 가기 위해 U 를 거쳐가는 것이 더 빠르다는 의미입니다.
  - d[시청] + w(시청, 서울역) < d[서울역] 이므로 d[서울역] = d[시청] + w(시청, 서울역) 으로 업데이트합니다.
  - 모든 마디에 대하여 반복합니다.

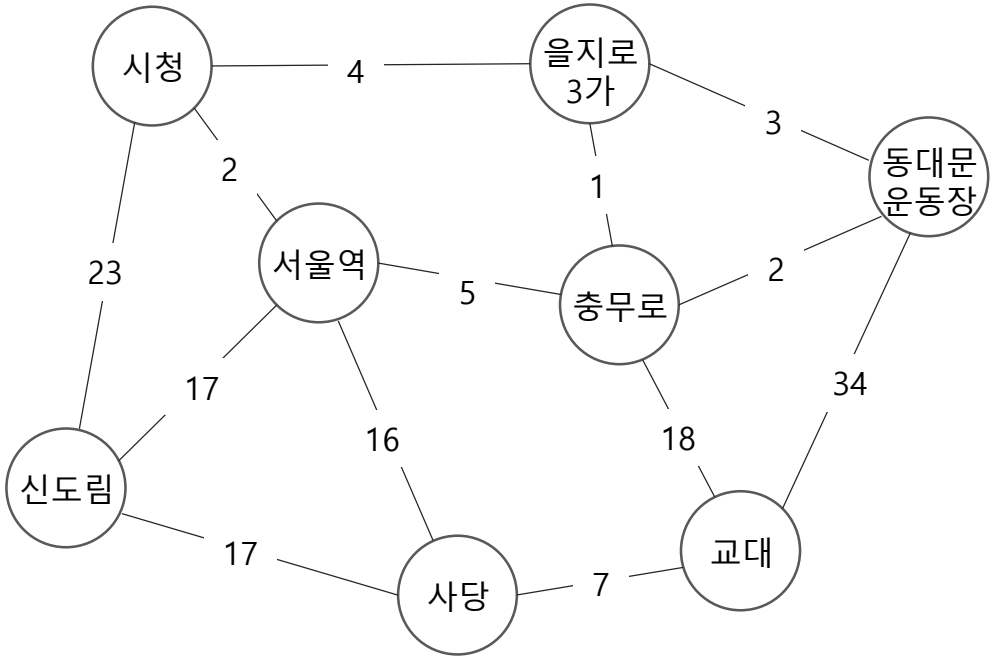


교대	동대문 운동장	사당	서울역	시청	신도림	을지로3가	충무로
-	-	-	-	0	-	-	-
-	-	-	2	0	23	4	-

Ford algorithm

‘시청 → 교대’ 의 최단 경로

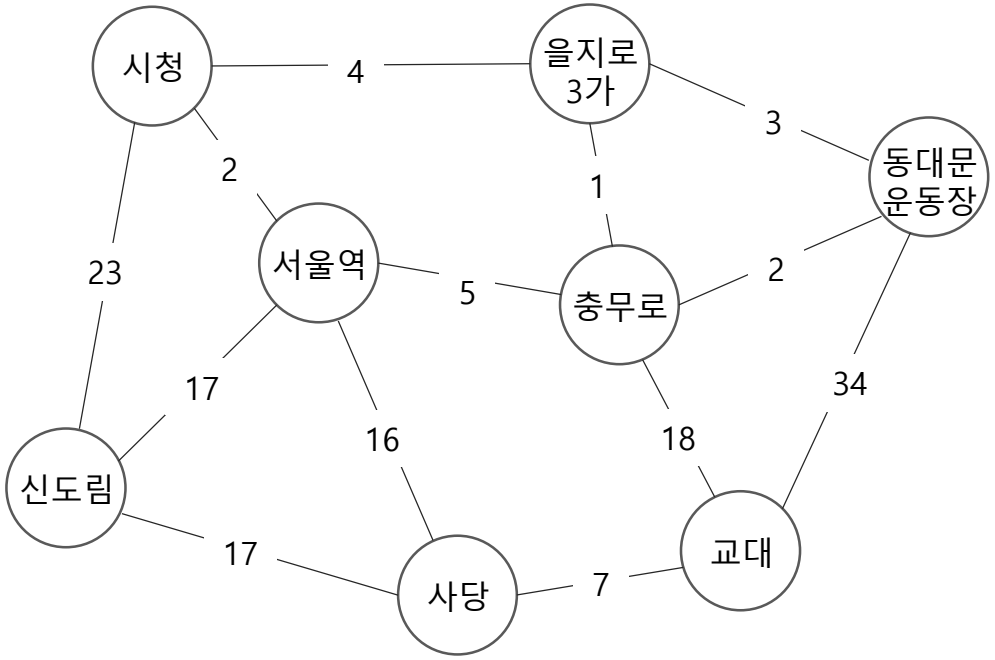
교대	동대문 운동장	사당	서울역	시청	신도림	을지로3가	충무로
-	-	-	-	0	-	-	-
-	-	-	2	0	23	4	-
-	7	18	2	0	19	4	7
-	7	18	2	0	19	4	5
23	7	18	2	0	19	4	5



Ford algorithm

‘시청 → 교대’ 의 최단 경로

교대	동대문 운동장	사당	서울역	시청	신도림	을지로3가	충무로
-	-	-	-	0	-	-	-
-	-	-	2	0	23	4	-
-	7	18	2	0	19	4	7
-	7	18	2	0	19	4	5
23	7	18	2	0	19	4	5

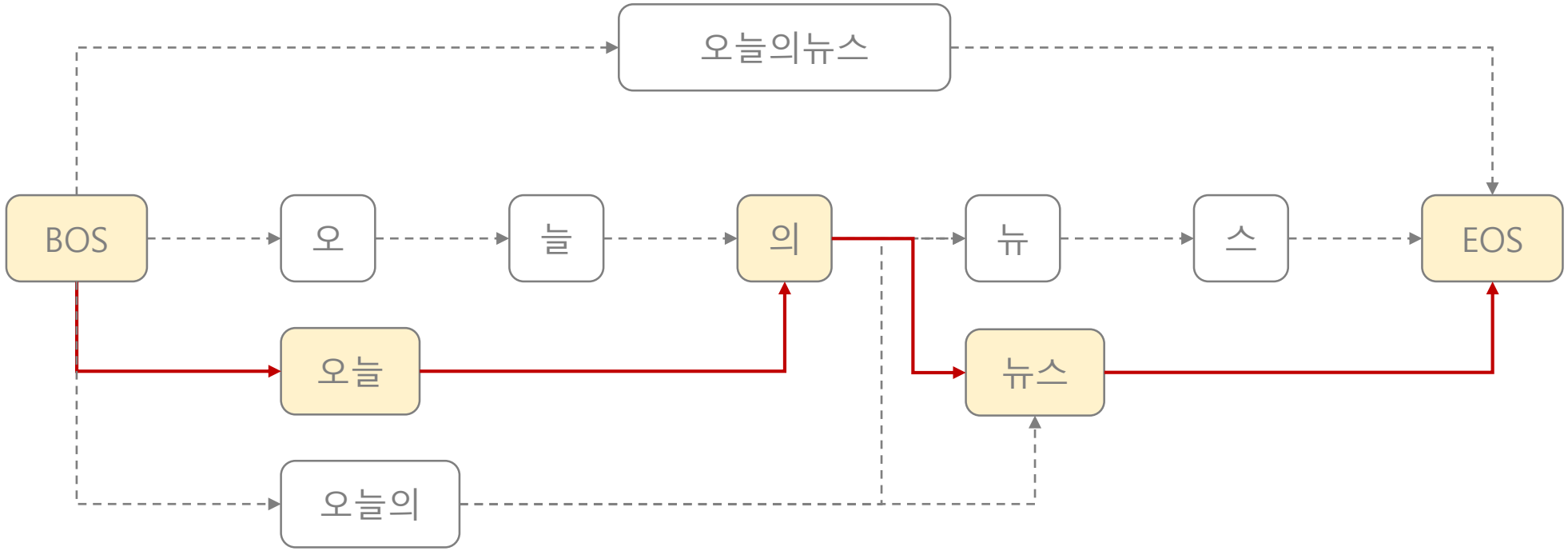


d[U] + w(U, V) == d[V] 인 마디를 연결

- [시청 – 서울역 – 사당 – 교대]

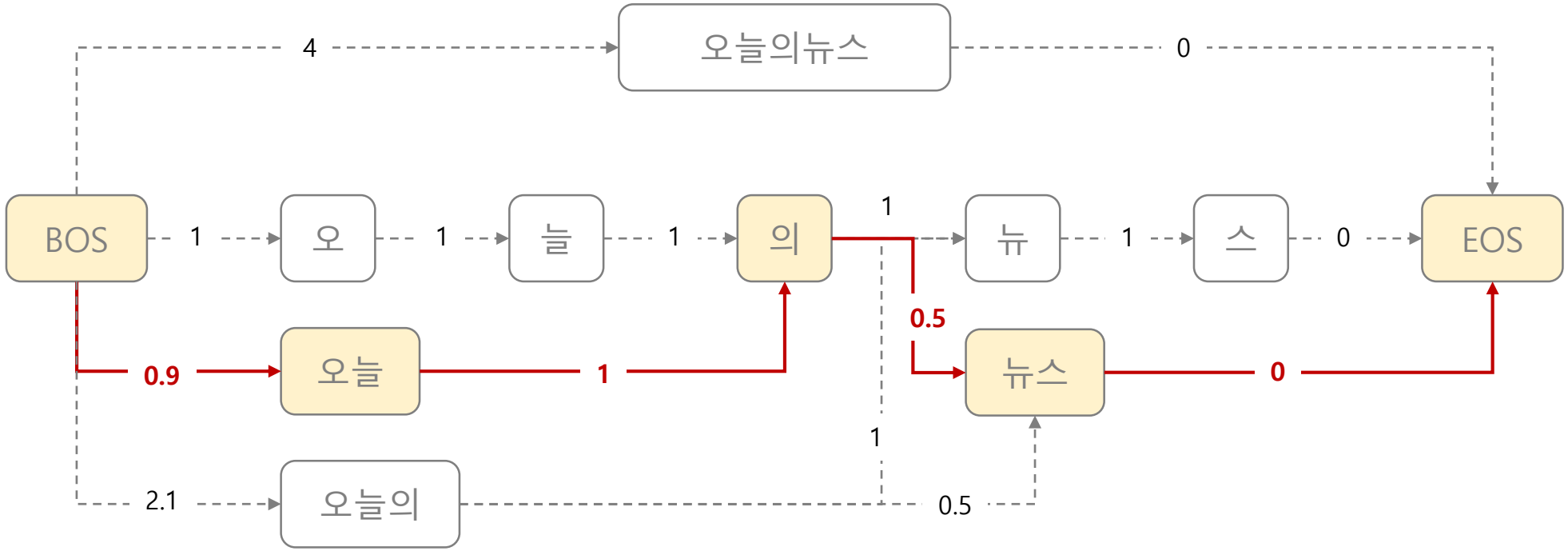
Word segmentation as Shortest path

- 단어를 마디로 생각하면, 토크나이징은 최단 경로 문제입니다.
  - (예문) 오늘의뉴스



Word segmentation as Shortest path

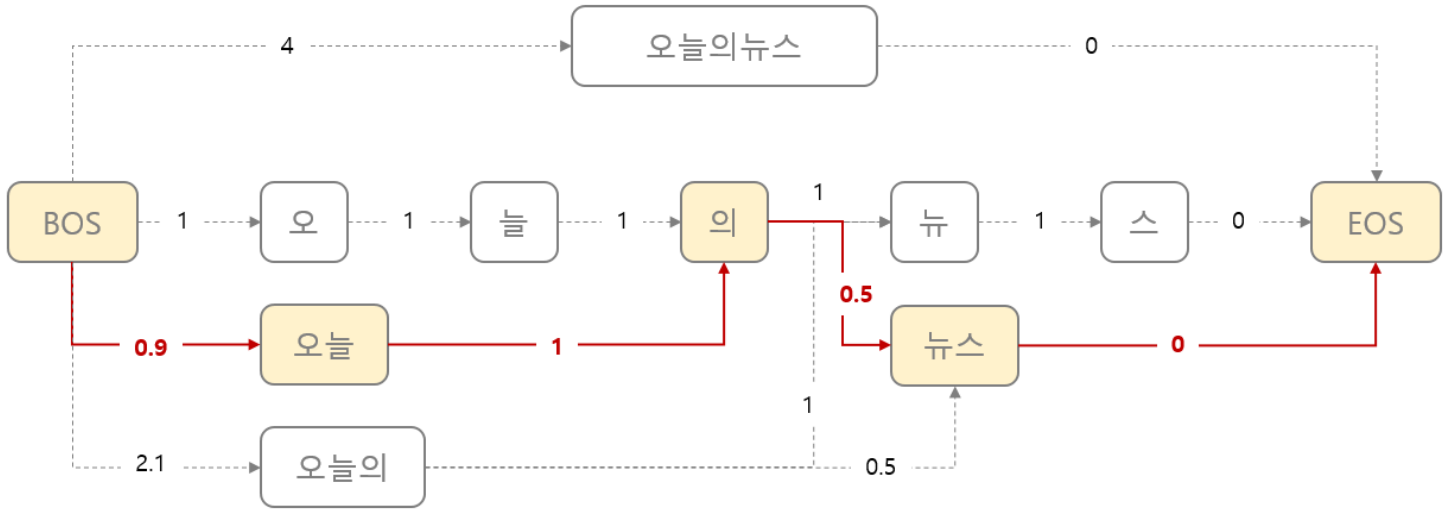
- Edge weight 를 잘 정의해야 합니다.
  - 원하는 단어가 선택되었을 때의 비용을 작게, 원하지 않는 segmentation 이 일어났을 때의 비용을 크게 설정합니다



Word segmentation as Shortest path

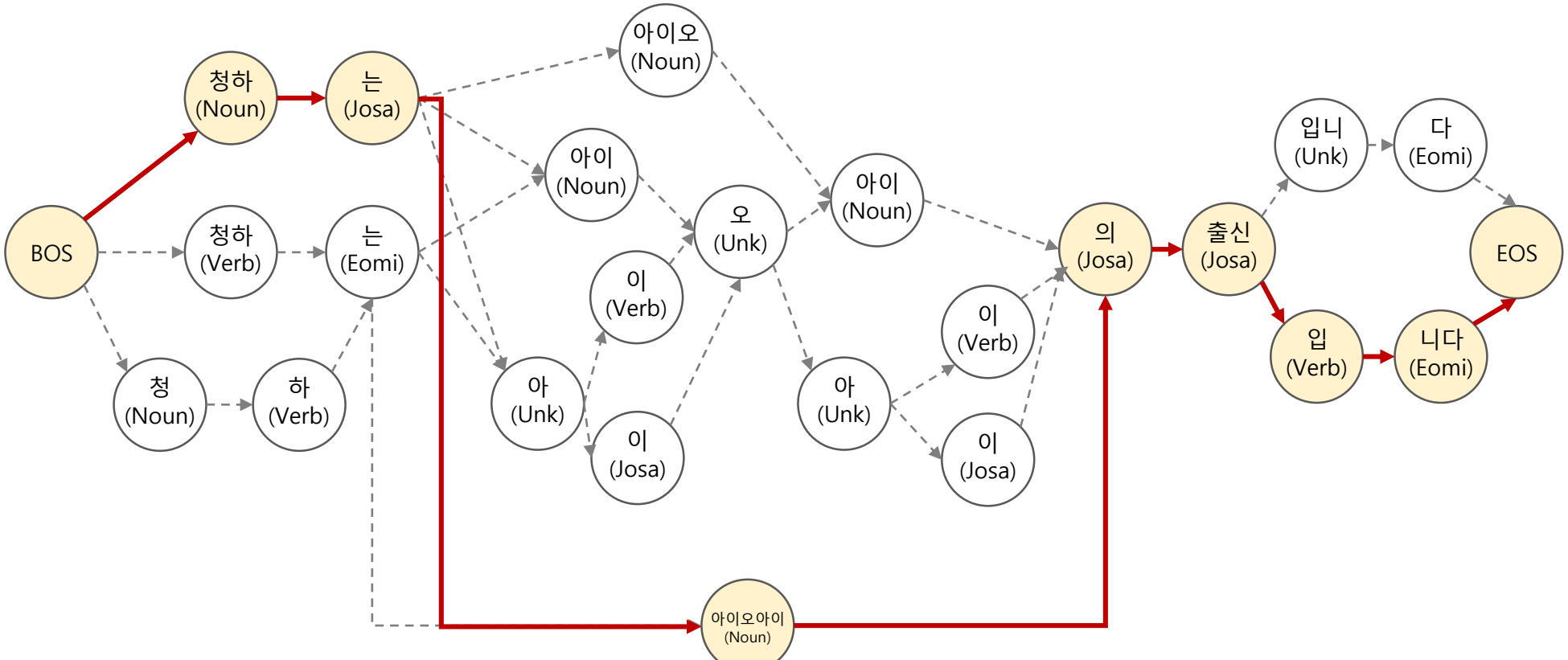
- Edge weight 를 잘 정의해야 합니다.
  - 도착 마디의 (단어 길이 – 단어 점수) 예시입니다

```
sent = '오늘의뉴스'
edges = [
  ('BOS', '오', 1),
  ('BOS', '오늘의뉴스', 4),
  ('오', '늘', 1),
  ('늘', '의', 1),
  ('의', '뉴', 1),
  ('뉴', '스', 1),
  ('스', 'EOS', 0),
  ('BOS', '오늘', 0.9),
  ('BOS', '오늘의', 2.1),
  ('오늘', '의', 1),
  ('의', '뉴스', 0.5),
  ('오늘의', '뉴', 1),
  ('오늘의', '뉴스', 0.5),
  ('의', '뉴스', 0.5),
  ('뉴스', 'EOS', 0),
  ('오늘의뉴스', 'EOS', 0)
]
```



POS tagging as Shortest path

- 품사 판별 문제 역시 최단 경로로 풀 수 있습니다.
  - (예문) 청하는 아이오아이의 출십입니다



POS tagging as Shortest path

- 사전에 등록되어 있는 단어를 lookup 하여 마디 후보를 만듭니다.
- 마디 (U, V) weight 는 $-\log(P(V_t|U_t) \times P(V_w|V_t)$ 를 이용하면 Hidden Markov Model (HMM) 을 이용하는 품사 판별기가 됩니다.

  - $w(BOS \rightarrow 청하/Noun) = -\log\left(P(Noun|BOS) \times P(청하|Noun)\right)$
  - $w(청하/Noun \rightarrow 는/Josa) = -\log(P(Josa|Noun) \times P(는|Josa))$

HMM vs Shortest path

- HMM 은 문장 s 에 대하여 아래 확률이 가장 큰 단어/품사열을 찾습니다
  - $argmax_{w,t}\ P(w_{1:m}, t_{1:m}|S) = \prod_{i=1}^{m} P(t_i|t_{i-1}) \times P(w_i|t_i)$

- 위 식은 아래와 같습니다.
  - $argmin_{w,t}\ -\log \prod_{i=1}^{m} P(t_i|t_{i-1}) \times P(w_i|t_i)$

    $= \sum_{i=1}^{m} -\log\left(P(w_i|\,t_i) \times P(t_i|\,t_{i-1})\right)$

    shortest path edge weight

Reference

- 경영과학: 기초부터 심화까지, 홍성필, 율곡출판사 (2014)