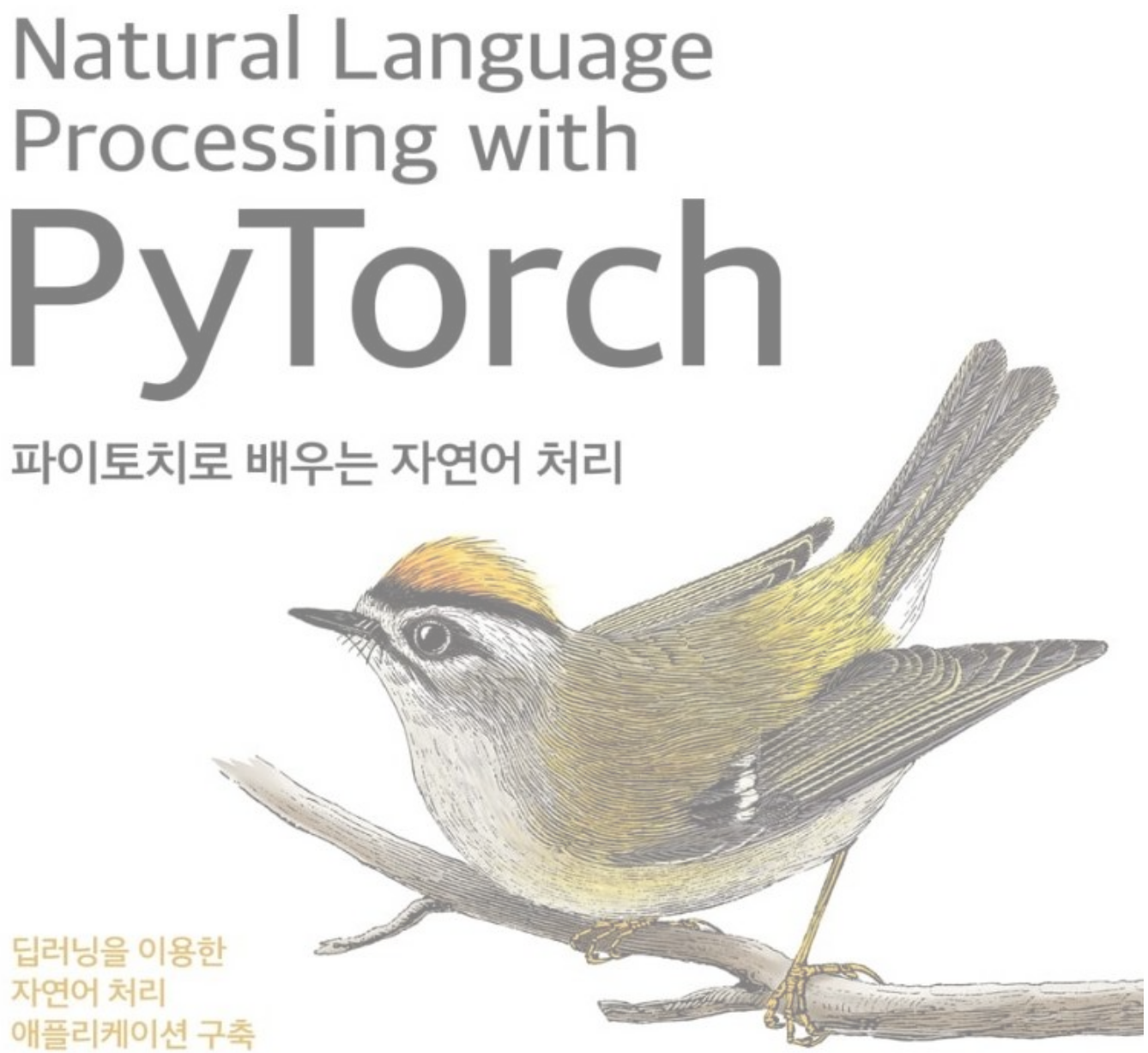


5 ~ 6장 발표

24.08.14
김태균

목차

1. 단어와 타입 임베딩

1.1 임베딩의 효율성

- **이산 타입** : 유한한 집합에서 얻은 모든 입력 특성
- 이산 타입을 **밀집 벡터**로 표현하는 것이 핵심
- **임베딩** : 이산 타입과 벡터 공간의 포인트 사이에 매핑을 학습하는 것

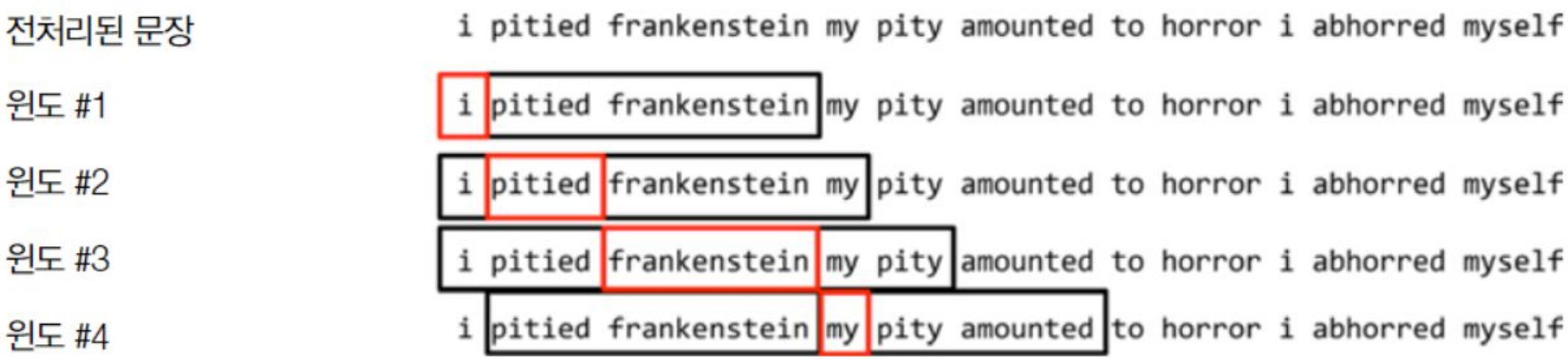
1.1 임베딩의 효율성

- **전체 어휘 사전의 크기 > 단어 임베딩의 크기**
- 단어를 낮은 차원의 밀집 벡터로 표현
  - **효율적인 계산**
  - **통계적 분석 가능**
- 단어 사용에서 규칙적으로 나타나는 의미 관계 파악 가능
  - man : woman :: he : _____

1.2 단어 임베딩 학습

- CBOW(Continuous Bag-of-Words)

: 주어진 context로부터 target 단어를 예측하는 방식으로 단어 임베딩을 학습



1.2 단어 임베딩 학습

- **CBOW 분류기**

1. Embedding 층을 사용해 단어를 벡터로 변환
2. 전반적인 문맥을 감지하도록 벡터를 결합
3. Linear 층에서 문맥 벡터를 사용해 예측 벡터를 계산

```
x_embedded_sum = F.dropout(self.embedding(x_in).sum(dim=1), 0.3)
y_out = self.fc1(x_embedded_sum)

if apply_softmax:
    y_out = F.softmax(y_out, dim=1)

return y_out
```

2. 시퀀스 모델링

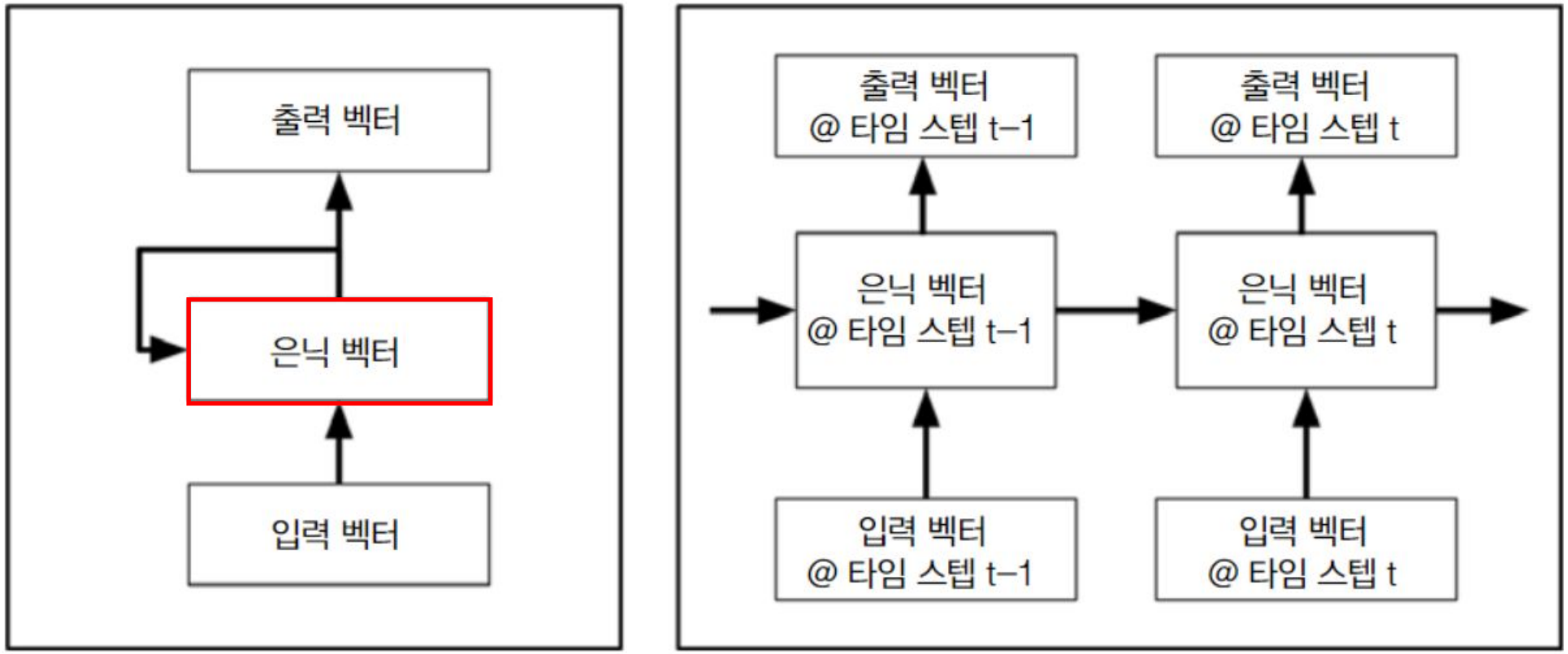
2. 시퀀스 모델링

- **시퀀스** : 순서가 있는 항목의 모임
- 시퀀스 데이터 : 한 데이터 항목이 앞뒤 항목에 의존하는 데이터
- 언어는 문장의 단어가 무작위로 나열되어 있지 않음

⇒ 언어를 이해하기 위해 시퀀스를 이해해야 함

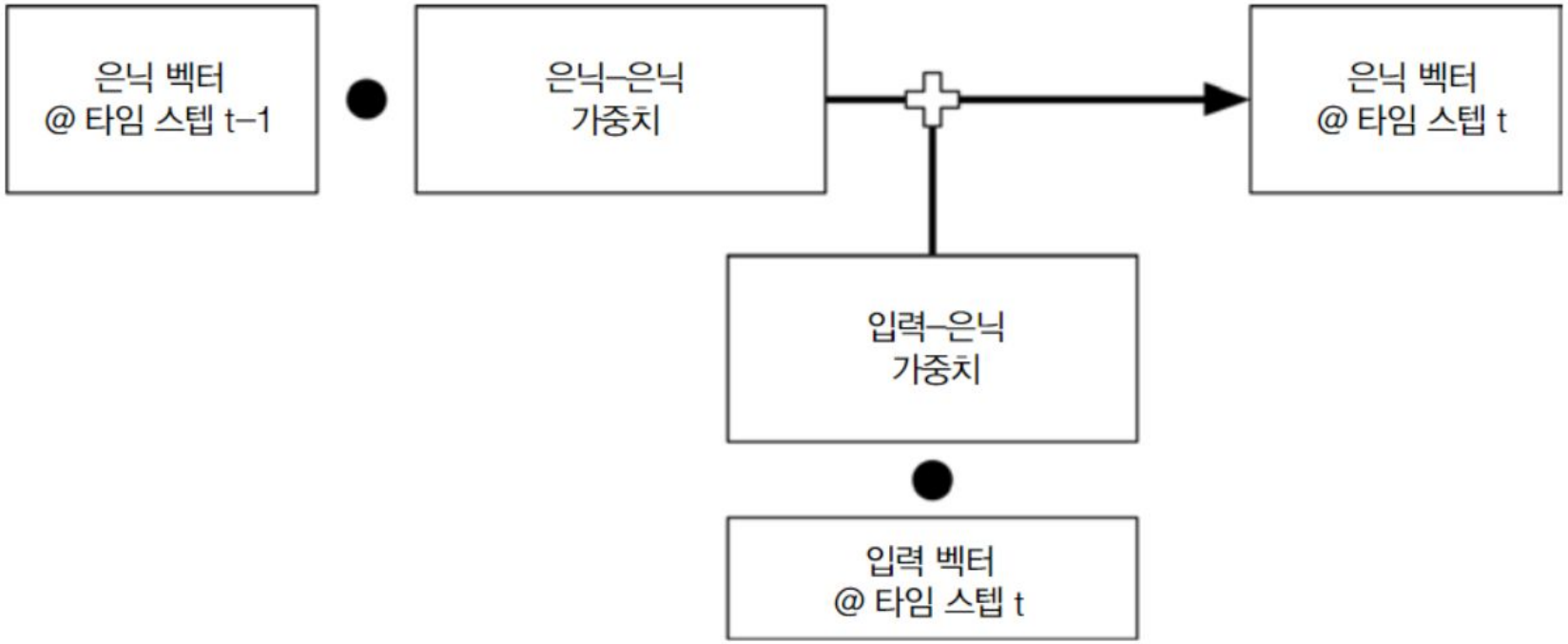
2. 시퀀스 모델링

- RNN(Recurrent Neural Network)
  - 엘만 RNN



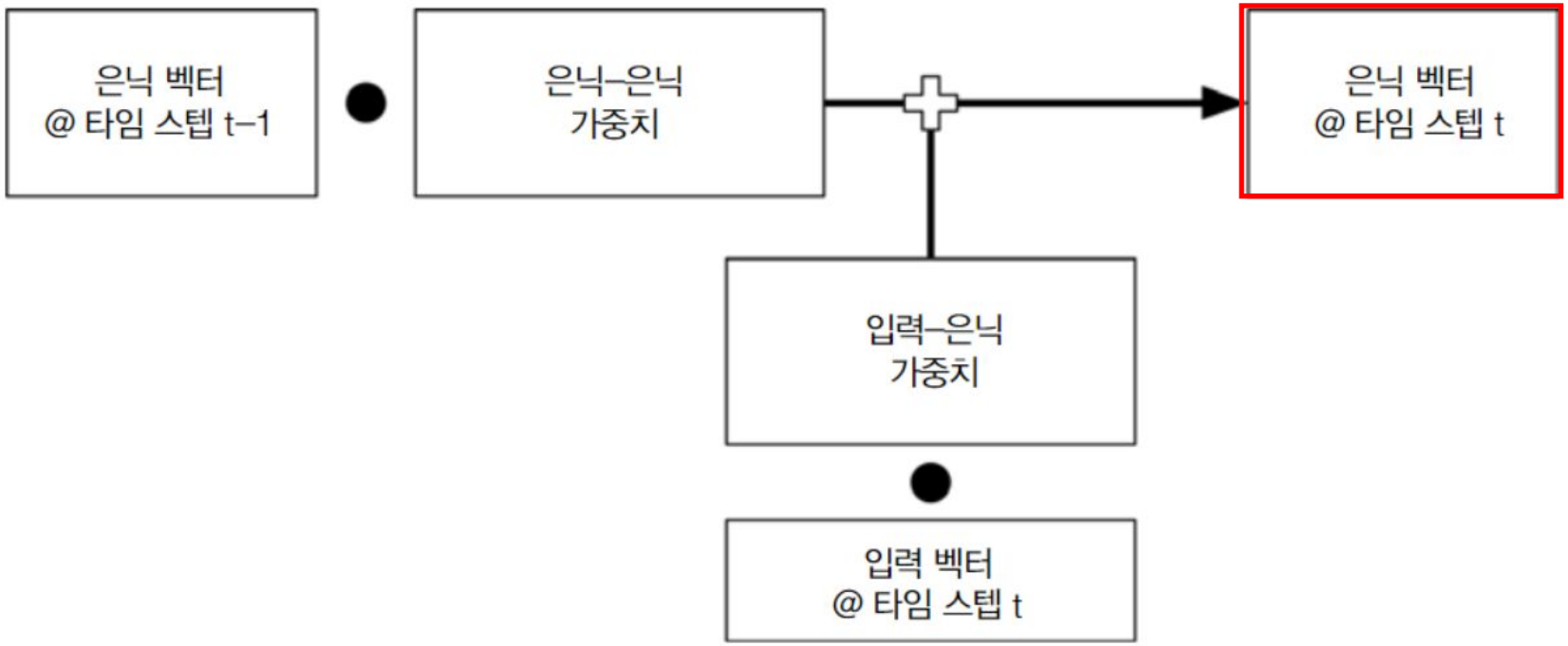
2. 시퀀스 모델링

- RNN(Recurrent Neural Network)
  - 은닉–은닉 가중치 : 이전 은닉 상태 벡터를 매핑
  - 입력–은닉 가중치 : 입력 벡터 → 새로운 은닉 벡터 계산



2. 시퀀스 모델링

- RNN(Recurrent Neural Network)
  - 두 가중치가 연속된 타임 스텝에 걸쳐 공유
  - **은닉 벡터**에 시퀀스 정보가 담겨져 있음



2. 시퀀스 모델링

- **Surname 분류기**

1. Embedding 층을 사용해 단어를 벡터로 변환
2. RNN 층을 사용해 시퀀스의 벡터 표현 계산
3. Linear 층에서 예측 벡터를 계산

```
x_embedded = self.emb(x_in)
y_out = self.rnn(x_embedded)

if x_lengths is not None:
    y_out = column_gather(y_out, x_lengths)
else:
    y_out = y_out[:, -1, :]

y_out = F.relu(self.fc1(F.dropout(y_out, 0.5)))
y_out = self.fc2(F.dropout(y_out, 0.5))

if apply_softmax:
    y_out = F.softmax(y_out, dim=1)

return y_out
```

정리

- 단어와 타입 임베딩
- 시퀀스 모델링

QnA