

16.3.1 ~ 16.3.2

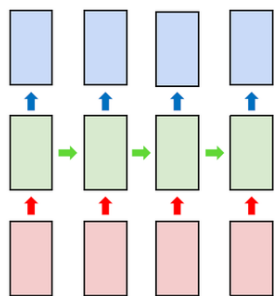
양방향 RNN ~ 빔 검색

2021.05.16 백관구

16.3.1. 양방향 RNN

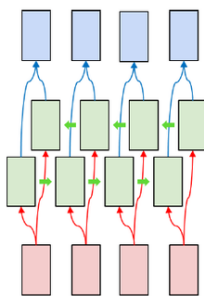
- 일반적인 RNN
 - 과거, 현재의 입력을 보고 출력을 생성
 - 시계열 예측에는 적합, NLP 작업에는 부족
 - the Queen of the UK / the queen of hearts
- ➔ queen의 정확한 의미를 알기 위해서는 뒤의 단어들을 알아야 함

What is “bidirectional”?



There is **imbalance** in the amount of information seen by the hidden states at different time steps.

VS



There is **balance** in the amount of information seen by the hidden states at different time steps.

- 양방향 RNN (Bidirectional RNN)
 - 두 개의 순환 층을 실행
 - 왼쪽 → 오른쪽 & 오른쪽 → 왼쪽
 - tensorflow.keras.layers.Bidirectional(순환 층)

매 타임스텝마다 10개의 값 출력

```
keras.layers.Bidirectional(keras.layers.GRU(10, return_sequences=True))
```

매 타임스텝마다 20개의 값 출력

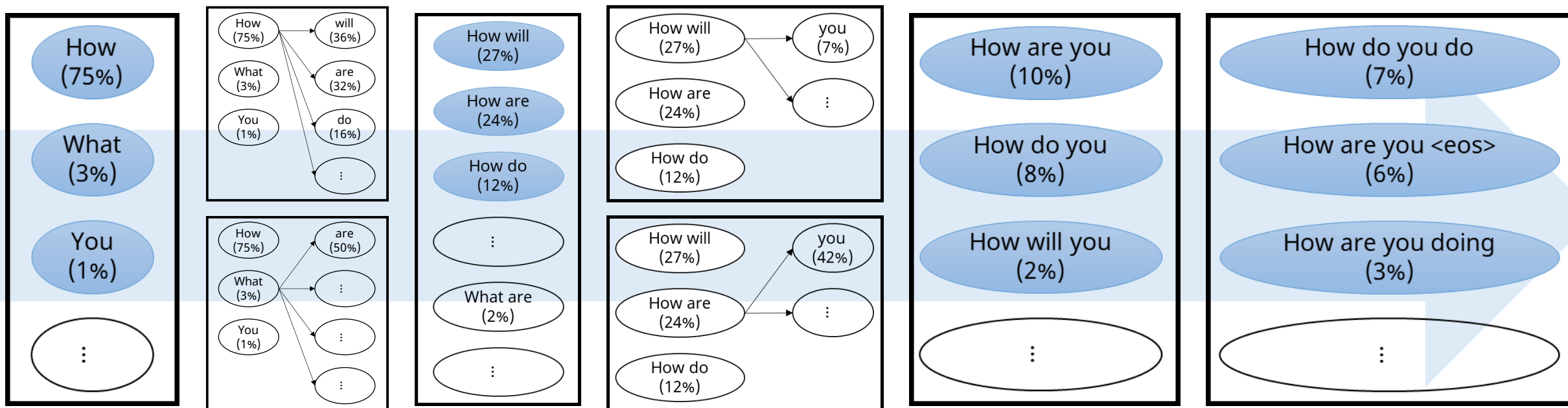
16.3.2. 빔 검색 (Beam search)

- 지금까지의 모델은 뒤로 돌아가 고칠 수 없음
 - 예시: (프랑스어) **Comment vas-tu?** 를 영어로 번역
 - 정답은 How are you? 지만, 모델은 **How will you?** 라고 출력함. Why?
 - 훈련 세트: **Comment vas-tu** jouer? → **How will you** play?



16.3.2. 빔 검색 (Beam search)

- 스텝마다 무조건 가능성이 가장 높은 단어를 출력해서는 앞의 사례와 같이 최적의 번역을 하지 못함
- 빔 검색: 상위 k개의 문장 리스트를 유지하고, 디코더에서 이 문장의 다음에 올 단어를 각각 생성하여 k개의 문장을 만듦
- 빔 너비(Beam width) = k
- 예시: Comment vas-tu? (k = 3)



16.3.2. 빔 검색 (Beam search)

- tensorflow_addons.seq2seq.BeamSearchDecoder 로 구현 (pip install tensorflow-addons)
 - 짧은 문장에 대해서는 좋은 성능, 하지만 긴 문장에서는 단기 기억 문제로 인해 성능 저하
- ➔ 어텐션 메커니즘으로 해결 가능