19. Text Summarization

텍스트 요약

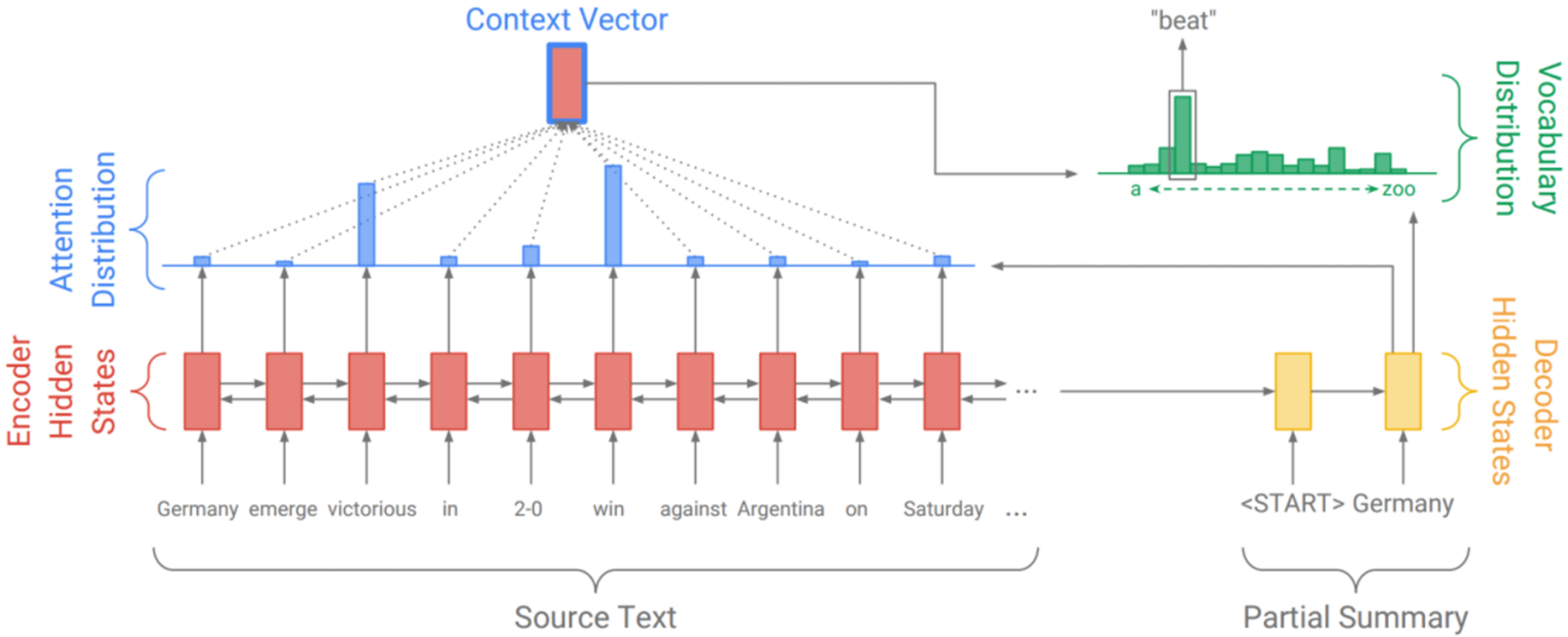
- 긴 원문을 짧은 요약문으로 변환하는 것
- **추출적 요약(Extractive summarization):** 원문에서 중요한 핵심 문장 또는 단어구를 뽑아서 요약문을 구성
  - 이미 존재하는 문장이나 단어구로만 구성
- **추상적 요약(Abstractive summarization):** 핵심 문맥을 반영한 새로운 문장을 생성
  - '원문'과 '실제 요약문'을 이용한 지도 학습을 사용

RNN을 이용한 추상적 요약

- "Summarization is also a mapping from input sequence to a (shorter) output sequence"
- 기계번역과 유사하게 attention을 가진 seq2seq model을 이용하여 추상적 요약을 훈련시킴
- 원 문장과 요약문을 함께 가진 훈련 문장들을 사용

seq2seq 기반 추상적 요약

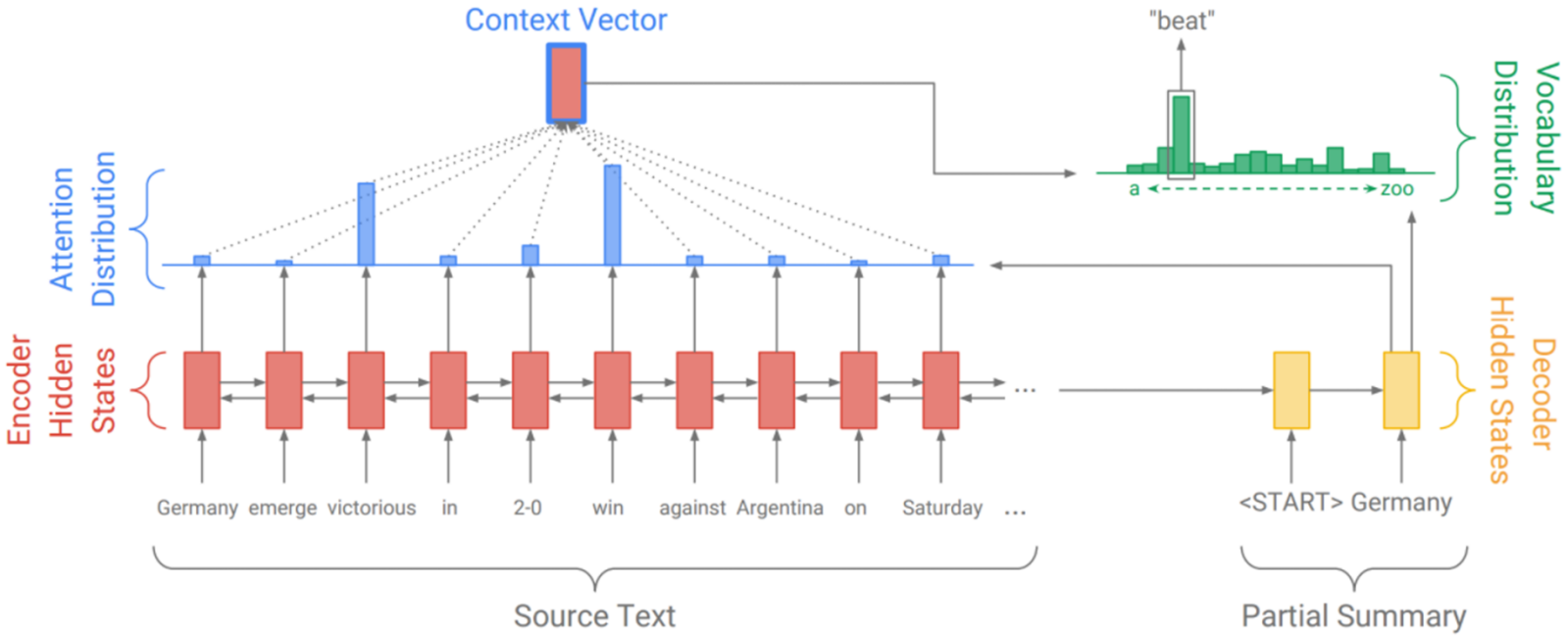
- A.M. Rush, S. Chopra, and J. Weston, A neural attention model for abstractive sentence summarization, 2015.



- **Source Text(원 문장):** *Germany emerge victorious in 2-0 win against Argentina on Saturday*
- **Summary:** *Germany beat Argentina 2-0*

seq2seq 요약 방식

- **Encoder:** <원 문장(source text)>을 입력하여 attention을 계산하고 Context Vector를 생성
- **Decoder:** <요약문>을 생성하도록 훈련시킴
- 기계번역기를 훈련시키는 것과 매우 유사함
- Context vector는 대용량의 고정된 사전에서 모든 단어에 대한 확률분포인 사전 분포(vocabulary distribution)를 계산하기 위해 사용됨
- 기존 방식에 비해 우수한 성과를 거두었음



Seq2seq 요약 방식의 문제점

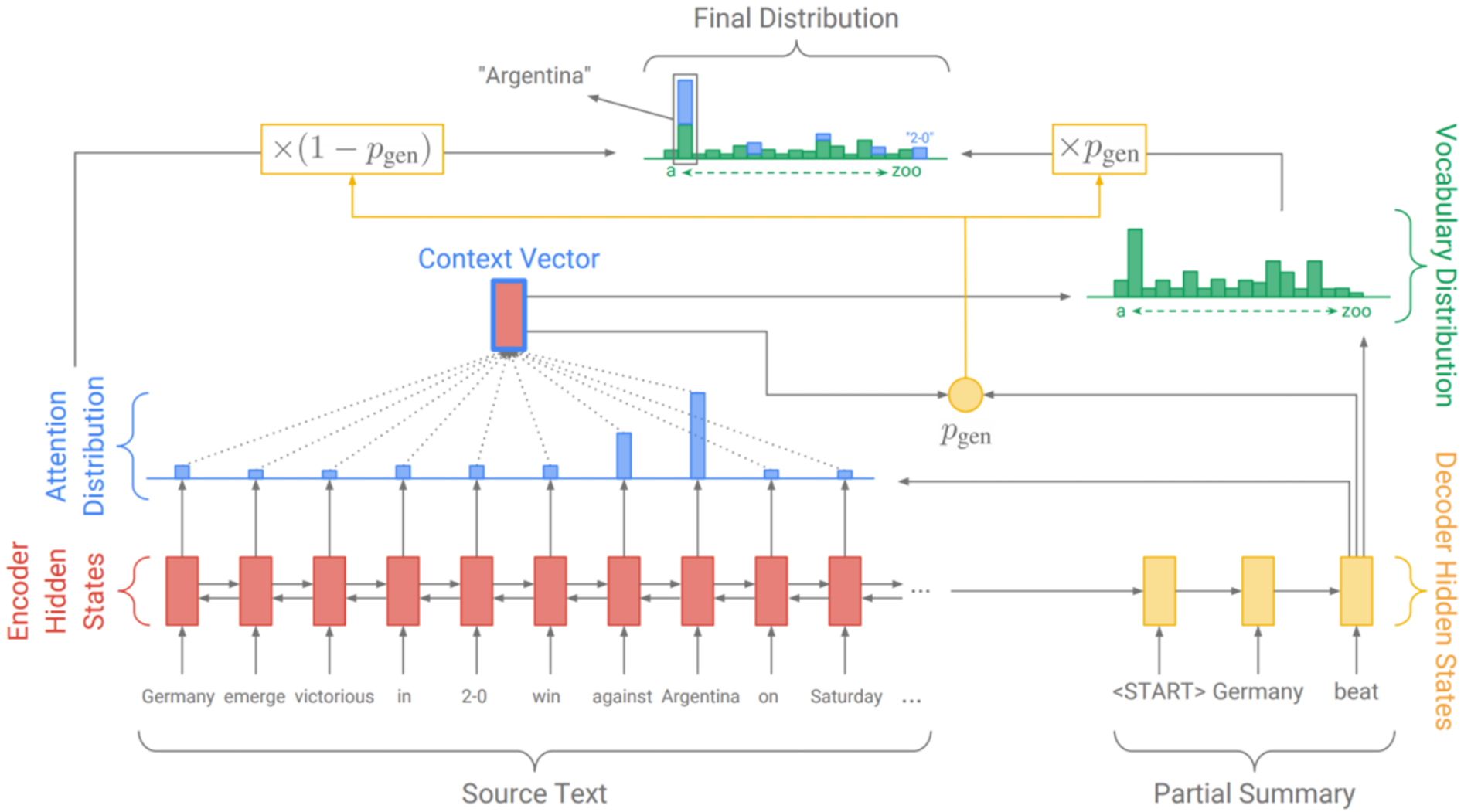
- **문제 1:** 요약문은 사실적인 세부사항을 부정확하게 재생산하는 경향이 있음. 사전에 없는 단어(out-of-vocabulary)이거나 희귀(rare) 단어인 경우 잘 발생
  - 예: *Germany beat Argentina 3-2* ('2-0'이 없는 단어라서 이런 결과를 생성)

- **문제 2:** 요약문은 때때로 같은 단어끼리 재반복해서 생산될 수 있음
  - 예: *Germany beat Germany beat Germany beat ….*

A. See, P.J. Liu, and C.D. Manning, Get to the point: Summarization with pointer-generator networks, 2017.

Pointer-generator network

- **문제 1**을 해결하기 위해 단어를 사전에서 가져오는 대신 <원 문장>에서 가져올 수 있는 pointing 방식을 도입



Pointer-generator network

- 요약문 단어 생성 함수:

$$P_{final}(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen})\sum_{i:w_i=w} a_i$$

  - p_{gen}은 단어를 사전에서 생성할지 <원 문장>에서 복사해올지를 결정
  - a는 attention 분포를 의미

Pointer-generator 모델의 특징

- 원 문장에서 단어들을 가져오기 쉬움
- 원 문장에서 OOV(out-of-vocabulary) 단어를 그대로 가져오는 것이 가능

- Pointer-generator 적용 후 성능 향상 사례

전	후
UNK UNK was expelled from the dubai open chess tournament	*galoz nigalidze* was expelled from the dubai open chess tournament
the *2015* rio Olympic games	the *2016* rio Olympic games

Coverage

- **문제 2**를 해결하기 위해 요약문에서 생성된 단어들에 대해 cover해온 기록들을 추적
  - 같은 부분을 다시 반복하면 penalty를 부과
- Decoder의 각 단계 t에서 coverage vector c^t를 다음과 같이 계산

$$c^t = \sum_{t'=0}^{t-1} a^{t'}$$

- Coverage와 attention a^t간의 중복성을 다음과 같이 계산

$$covloss_t = \sum_i min\left(a_i^t, c_i^t\right)$$

ROUGE

- **Recall-Oriented Understudy for Gisting Evaluation**
- 텍스트 예약 모델의 성능 평가 지표
- 모델이 생성한 요약본을 미리 만들어 놓은 시스템 요약과 대조해 성능 점수를 계산

ROUGE 사례 문장

- **시스템 요약(System summary):** *the cat was found under the bed*

- **참조 요약(Reference summary: 프로그램으로 생성한 요약):** *the cat was under the bed*

ROUGE에서의 Precision과 Recall

- ROUGE에서는 두 요약을 비교하여 Recall과 Precision을 계산
- **Recall:** 참조 요약에서 나타난 단어 중 몇 개가 시스템 요약과 겹치는지를 계산

$$Recall = \frac{Number\ of\ overlapped\ words}{Total\ words\ in\ reference\ summary}$$

- 앞의 사례의 경우

$$Recall = \frac{6}{6} = 1.0$$

ROUGE에서의 Precision

- **Precision:** 시스템 요약 단어 중 얼마나 참조 요약과 과 겹치는지를 계산

$$Precision = \frac{Number\ of\ overlapped\ words}{Total\ words\ in\ system\ summary}$$

- 앞의 사례의 경우

$$Precision = \frac{6}{7} = 0.86$$

- 보다 정확한 성능 평가를 위해 Precision과 Recall을 계산한 후 F-점수를 측정

F1 Score

- Precision과 Recall을 동시에 반영하기 위해 다음과 같이 $F1$ 점수를 정의

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

- Precision과 Recall의 범위는 $[0, 1]$이므로 $0 \leq F1 \leq 1$

ROUGE-N

- 두 요약문을 비교할 때 몇 개의 n-gram을 사용하는지 정의
- 단어를 비교한 앞의 ROUGE는 ROUGE-1 에 해당
- Bigram을 사용하는 경우
  - **시스템 요약:** *the cat was found under the bed*

    the cat, cat was, was found, found under, under the, the bed
  - **참조 요약:** *the cat was under the bed*

    the cat, cat was, was under, under the, the bed

$$ROUGE2_{recall} = \frac{4}{5} = 0.8$$

$$ROUGE2_{precision} = \frac{4}{6} = 0.67$$

ROUGE-L

- LCS(Longest common subsequence) 기법을 이용하여 최장 길이로 매칭되는 문자열을 측정. 보다 유연한 성능 비교가 가능
- 사례:
  - Reference: police killed the gunman
  - System-1: police kill the gunman
  - System-2: the gunman kill police
  - ROUGE-L:
    - System-1: 3/4 (“police the gunman”)
    - System-2: 2/4 (“the gunman”)

Pointer-generator 요약 시스템 성능 평가

- 논문에서 제시된 ROUGE F1 score

	ROUGE		
	1	2	L
abstractive model (Nallapati et al., 2016)*	35.46	13.30	32.65
seq-to-seq + attn baseline (150k vocab)	30.49	11.17	28.08
seq-to-seq + attn baseline (50k vocab)	31.33	11.81	28.83
pointer-generator	36.44	15.66	33.42
pointer-generator + coverage	**39.53**	**17.28**	**36.38**
lead-3 baseline (ours)	40.34	17.70	36.57
lead-3 baseline (Nallapati et al., 2017)*	39.2	15.7	35.5
extractive model (Nallapati et al., 2017)*	39.6	16.2	35.3