Text Summarization with Pretrained Encoder

Yang Liu, Mirella Lapata

집현전 논문 리뷰

20조 강민지, 강민구

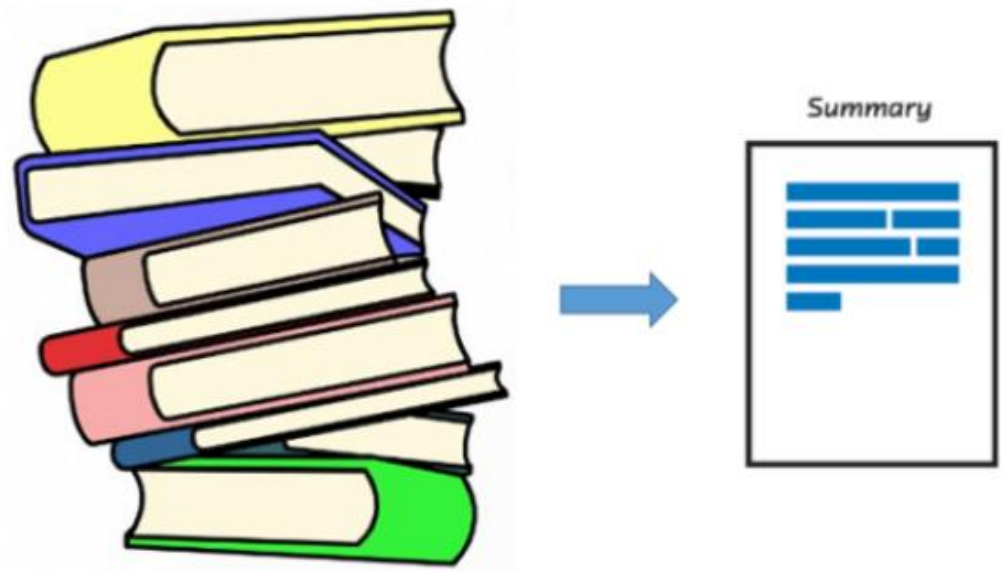
목차

1. Background

Background

https://youtu.be/moCNw4j2Fkw

Text Summarization

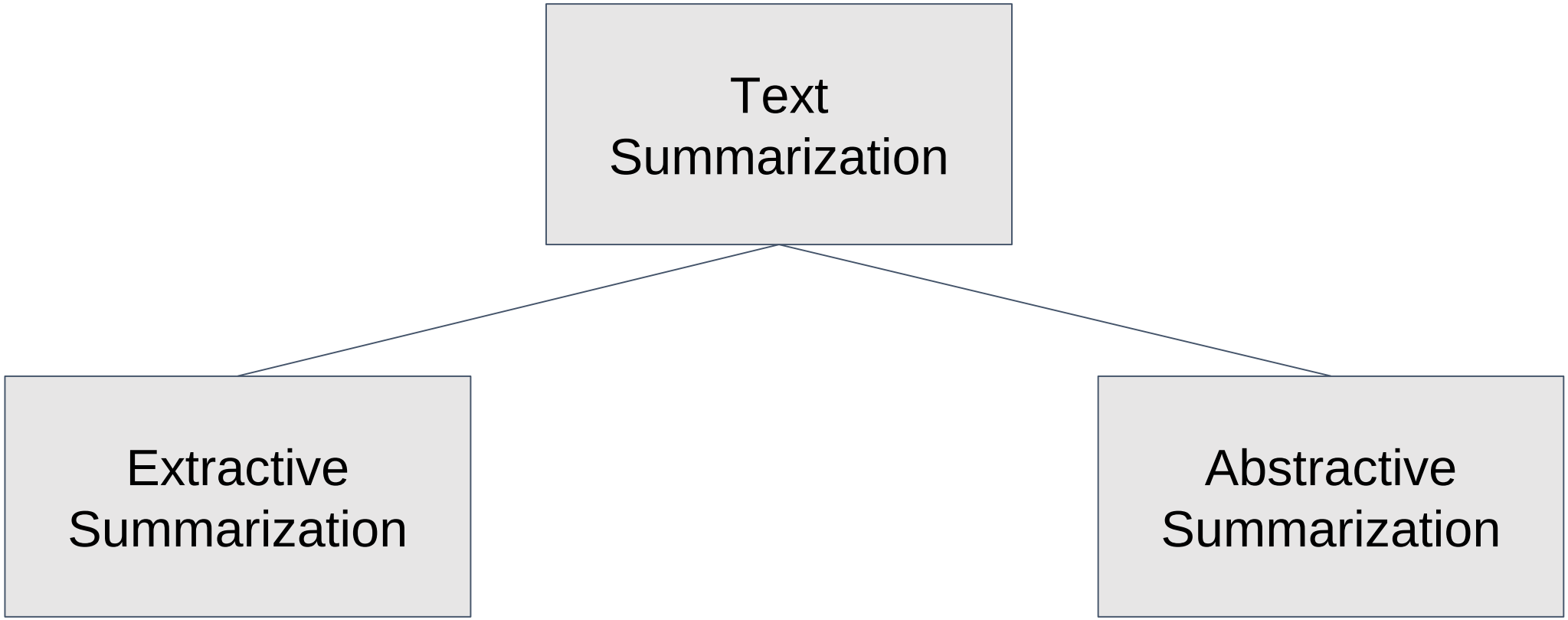


The task of producing a **concise** and **fluent summary**
while preserving **key information content** and **overall meaning**

-Text Summarization Techniques: A Brief Survey, 2017

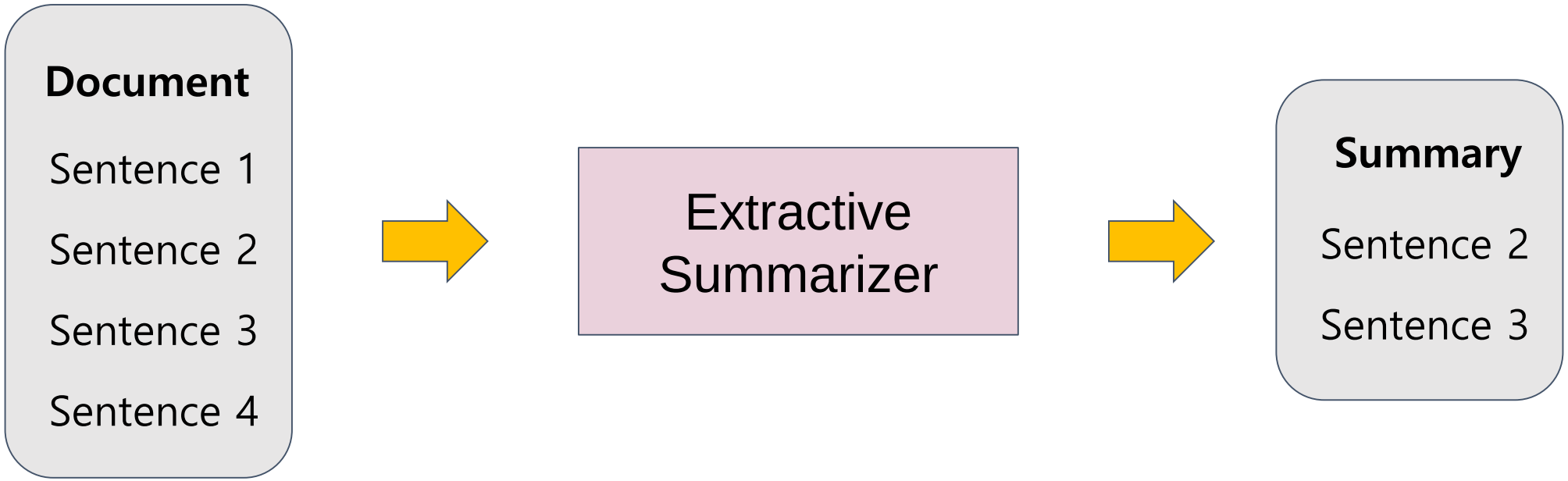
핵심 내용과 의미를 보존하면서
간결하고 유창한 요약을 만드는 작업

Text Summarization



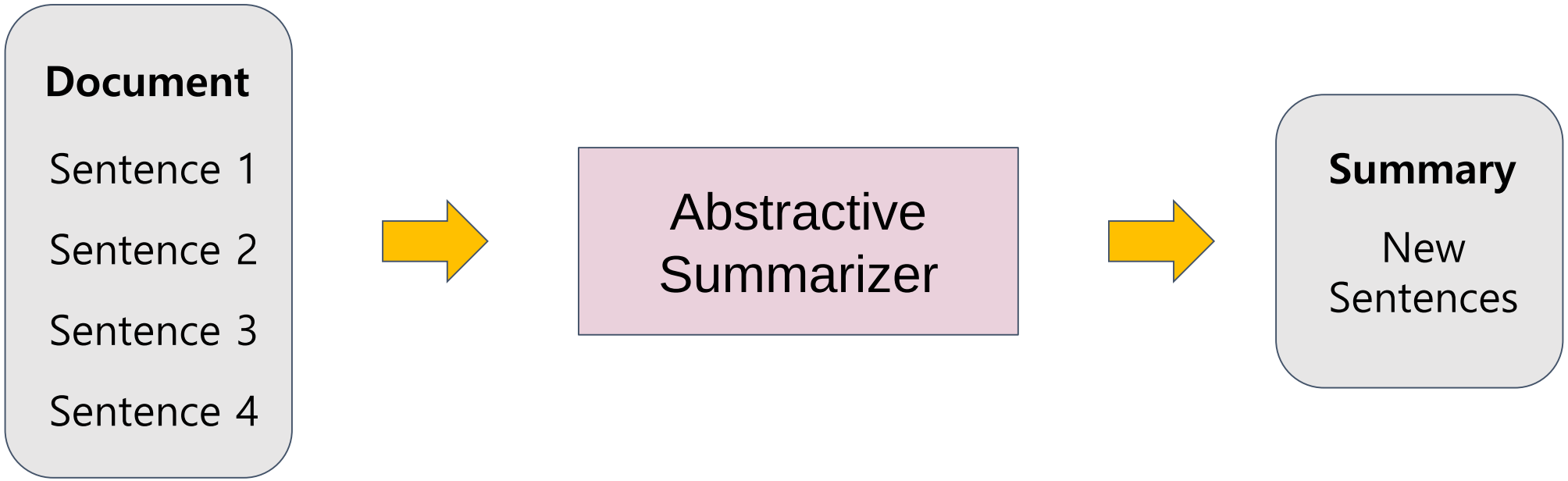
Text Summarization - Extractive

문서에서 중요한 문장이나 구를 찾아내고 발췌하는 작업



Text Summarization - Abstractive

주어진 문서를 이해하고 짧고 간결한 새로운 문장을 생성하는 작업



About paper

Text Summarization with Pretrained Encoders

Yang Liu and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
yang.liu2@ed.ac.uk, mlap@inf.ed.ac.uk

- Published in: EMNLP 2019
- # of citation: 541

2. Introduction

Introduction

- **Pretrained Encoders**
  - BERT와 같은 Pretrained Language Model 들은 많은 NLP 작업에서 좋은 성능을 보여줌
    - Text classification
    - Text entailment
    - Reading comprehension
    - ...

Introduction

- **요약에는 자연어에 대한 넓은 이해를 필요로 함**
  - 문장을 넘어 문서 수준의 이해 필요

    (BERT is trained on sentence-level)

Challenge 1: representation of multiple sentences
- **Extractive** summarization requires that sentences be included in the summary.

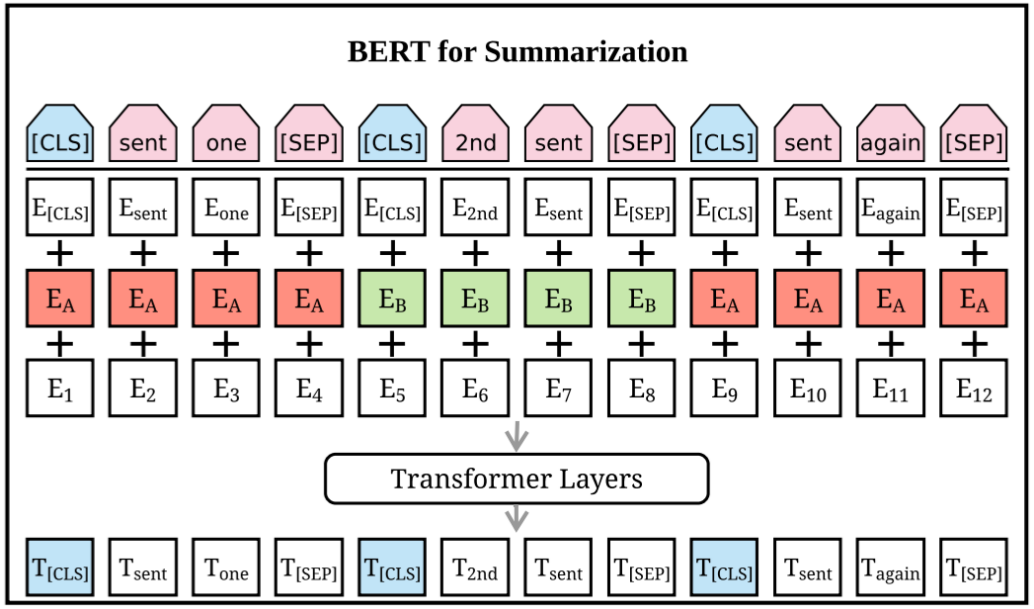
Challenge 2: mismatch between encoder and decoder
- **Abstract** summarization generates summaries containing new words and phrases

- 본 논문에서는, 위 문제를 해결할 수 있는 새로운 문서 수준의 요약 모델을 제안

3. Fine-tuning BERT for Summarization

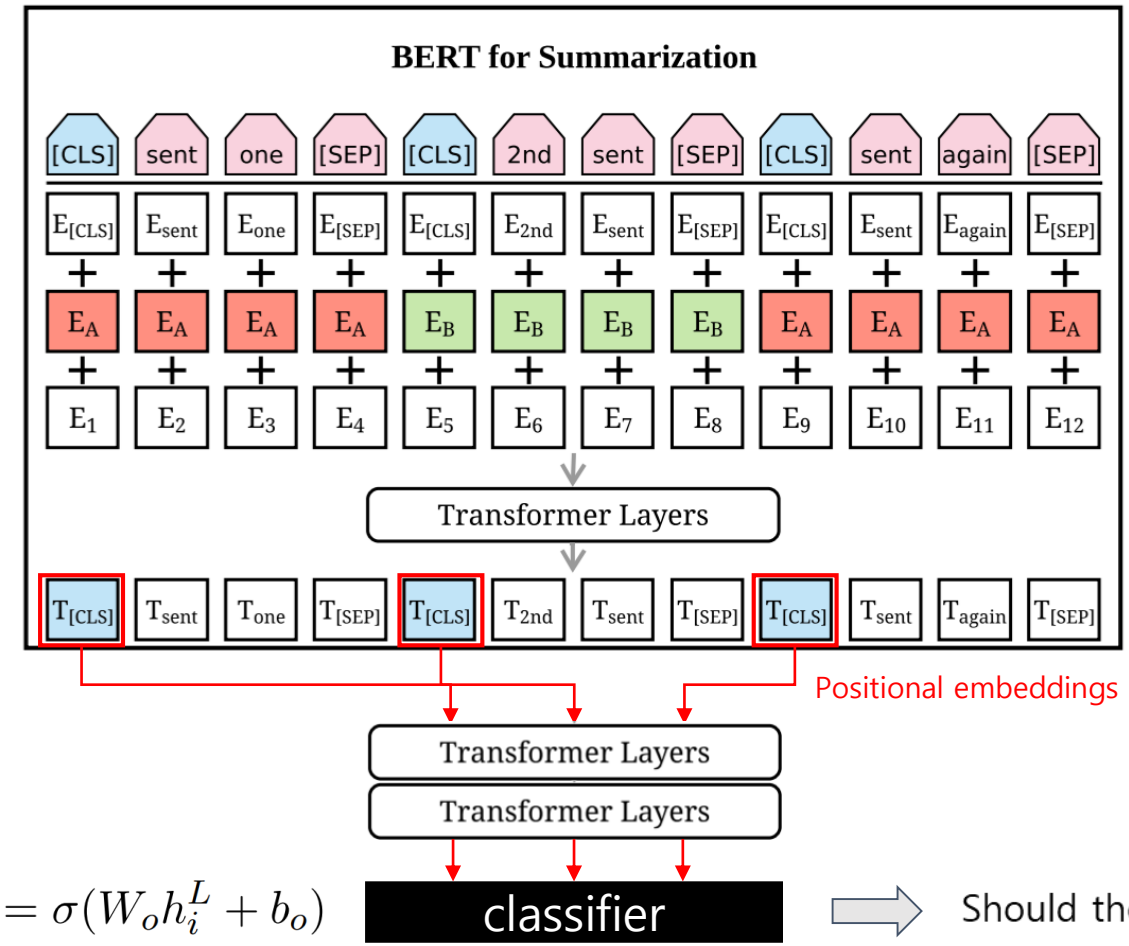
Architecture - BERTSUM

BERTSUM



- Challenge 1: representation of multiple sentences
  - 각 문장의 시작에 [CLS] 토큰 삽입
  - segmentation embedding

Extractive Summarization - 1) BERTSumExt

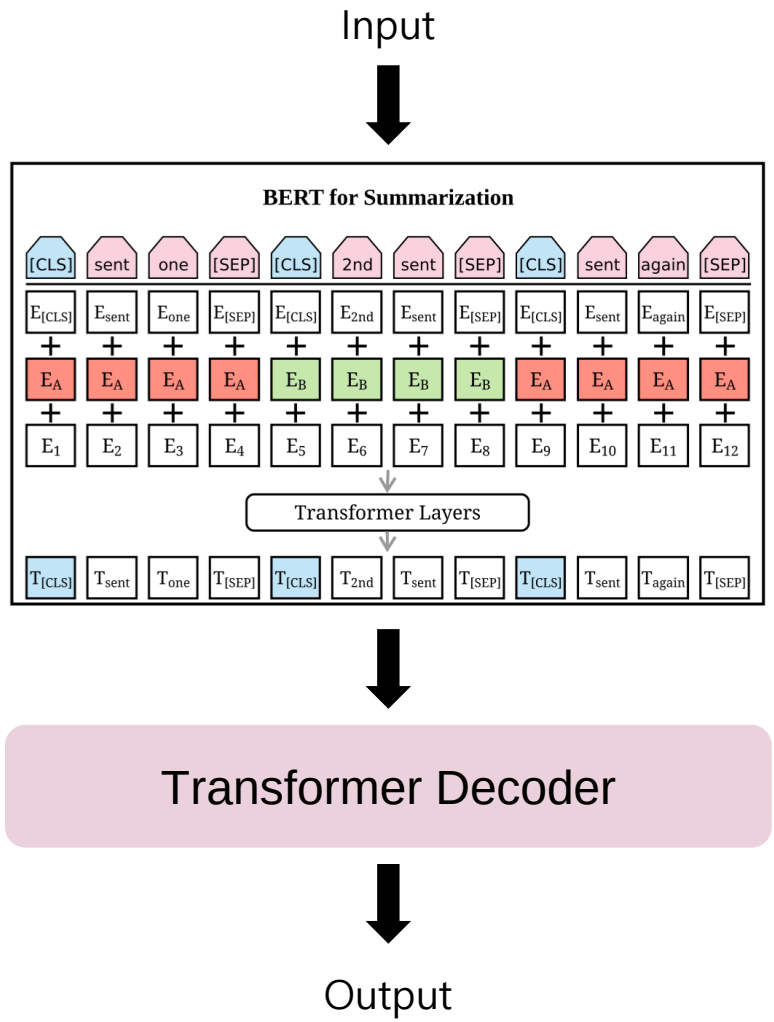


$$lr = 2e^{-3} \cdot \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5})$$
$$(\text{warmup} = 10,000):$$

$$\hat{y}_i = \sigma(W_o h_i^L + b_o)$$

⇨ Should the sentence (T_{CLS_i}) be included in the summary?

Abstractive Summarization - 2) BERTSumAbs

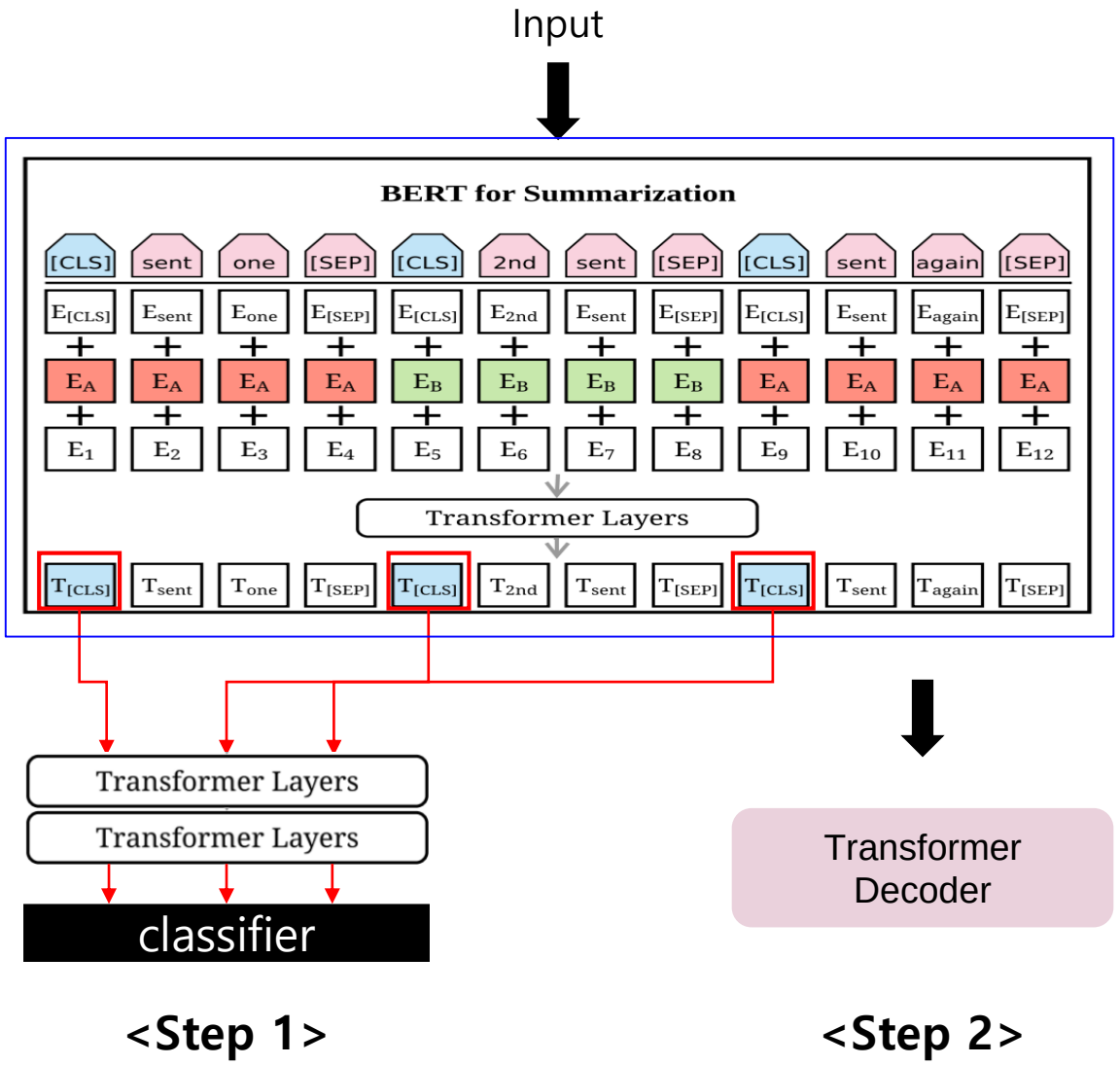


- BERTSUM은 인코더로, 랜덤 초기화된 decoder와 조합해야 함
- Challenge 2: mismatch between encoder and decoder
  Decoder initialized randomly; This can make fine-tuning unstable
  - **Separates the optimizers** of the encoder and the decoder

$$\tilde{lr}_{\mathcal{E}} \cdot \min(step^{-0.5}, \text{step} \cdot \text{warmup}_{\mathcal{E}}^{-1.5}) \qquad \tilde{lr}_{\mathcal{E}} = 2e^{-3}, \ \text{warmup}_{\mathcal{E}} = 20,000$$

$$\tilde{lr}_{\mathcal{D}} \cdot \min(step^{-0.5}, \text{step} \cdot \text{warmup}_{\mathcal{D}}^{-1.5}) \qquad \tilde{lr}_{\mathcal{D}} = 0.1, \ \text{warmup}_{\mathcal{D}} = 10,000$$

Abstractive Summarization - 3) BERTSumExtAbs



○ Propose a two-stage fine-tuning approach

1. fine-tune the encoder on the **extractive summarization task**
2. fine-tune it on the **abstractive summarization task**

↓

Boost the performance of abstractive summarization.

4. Experiments & Results

Summarization Datasets

- 기사

- 기사 요약문
- 사람이 요약(Gold Summary)
- 생성 요약문

	# docs(train/val/test)	avg. doc sentences	avg. summary sentences	% novel bi-grams in gold summary
CNN	90,266/1,220/1,093	33.98	3.59	52.90
DailyMail	196,961/12,148/10,397	29.33	3.86	52.16
NYT	96,834/4,000/3,452	35.55	2.44	54.70
XSum	204,045/11,332/11,334	19.77	1.00	84.31

- 3가지 벤치마크 데이터셋
- 데이터셋 요약문

  1)CNN/DailyMail - 몇 개의 bullet point

  2)NYT - 뉴욕타임즈 기사 요약문

  3)XSum - BBC 기사 한 줄 요약

- train/val/test 구분은 이전 논문과 동일하게 split(결과 비교를 위해)
- % novel bi-grams in gold summary

  = 요약문에 새로운 단어가 나타나는 비율

  = 비중이 클수록 BERTSumEXT보다

  BERTSumABS나 BERTSumEXTABS의 성능이 더 좋을 것으로 예상됨

Summarization Datasets

	# docs(train/val/test)	avg. doc sentences	avg. summary sentences	% novel bi-grams in gold summary
CNN	90,266/1,220/1,093	33.98	3.59	52.90
DailyMail	196,961/12,148/10,397	29.33	3.86	52.16
NYT	96,834/4,000/3,452	35.55	2.44	54.70
XSum	204,045/11,332/11,334	19.77	1.00	84.31

- % novel bi-grams in gold summary가 모두 50%를 초과
- 주어진 데이터셋의 요약문(gold summary)는 모두 생성 요약문 → 생성 요약 모델만 실험 가능
- 추출요약 모델 학습을 위한 정답 요약문을 생성해야 함.
  - gold summary와 가장 유사한 문장을 본문에서 greedy하게 3개 선택(ROUGE score 기준)
  - 중복 문장 추출 방지 : 이전 추출 문장과 tri-gram이 일치하면 추출X → 다음 문장 확인
  - 위와 같이 만들어낸 추출요약 정답 요약문 = **ORACLE summary**

	추출요약 모델 (BERTSumEXT)	생성요약 모델 (BERTSumABS)
모델 학습	ORACLE summary	Gold summary
모델 평가	Gold Summary	

Eval. Metric - ROUGE

● 정답 요약문(gold summary) y와 모델 요약문 y_hat 사이에 **겹치는 단어 or 구(phrase)의 수**로 판단

1) **Precision**

= 일치하는 단어 수/모델 요약문 단어 수

2) **Recall**

= 일치하는 단어 수/정답 요약문 단어 수

3) **F1-score**

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

[예시]

- 정답) the cat was under the chair
- 모델) the cat was found under the bed

1) Precision = 5/7
2) Recall = 5/6
3) F1-socre = 10/13

Eval. Metric - ROUGE(precision)

- 정답) the cat was under the chair
- 모델) the cat was found under the bed

→ Precision = 5/7

- **요약문에 오답 단어 1개 추가**
- 정답) the cat was under the chair
- 모델) the cat was found under the compact bed

→ Precision = 5/8

1. 모델 요약문에 쓸데없는 단어가 들어가면 precision **감소**
2. **간결한 문장을 선호**하는 지표

Eval. Metric - ROUGE(recall)

- 정답) the cat was under the chair
- 모델) the cat was found under the bed

→ Recall = 5/6

- **요약문에 오답 단어 1개 추가**
- 정답) the cat was under the chair
- 모델) the cat was found under the compact bed

→ Recall = 5/6

1. 모델 요약문에 오답 단어가 들어가도 **수치 변동 없음**
   → 모델이 만든 요약문 안에 정답 요약문의 단어가 최대한 많이 들어있어야 유리
2. **많은 정보량을 선호**하는 지표

Eval. Metric - ROUGE(f1-score)

- precision : 간결한 문장 선호
- recall : 많은 정보량을 선호

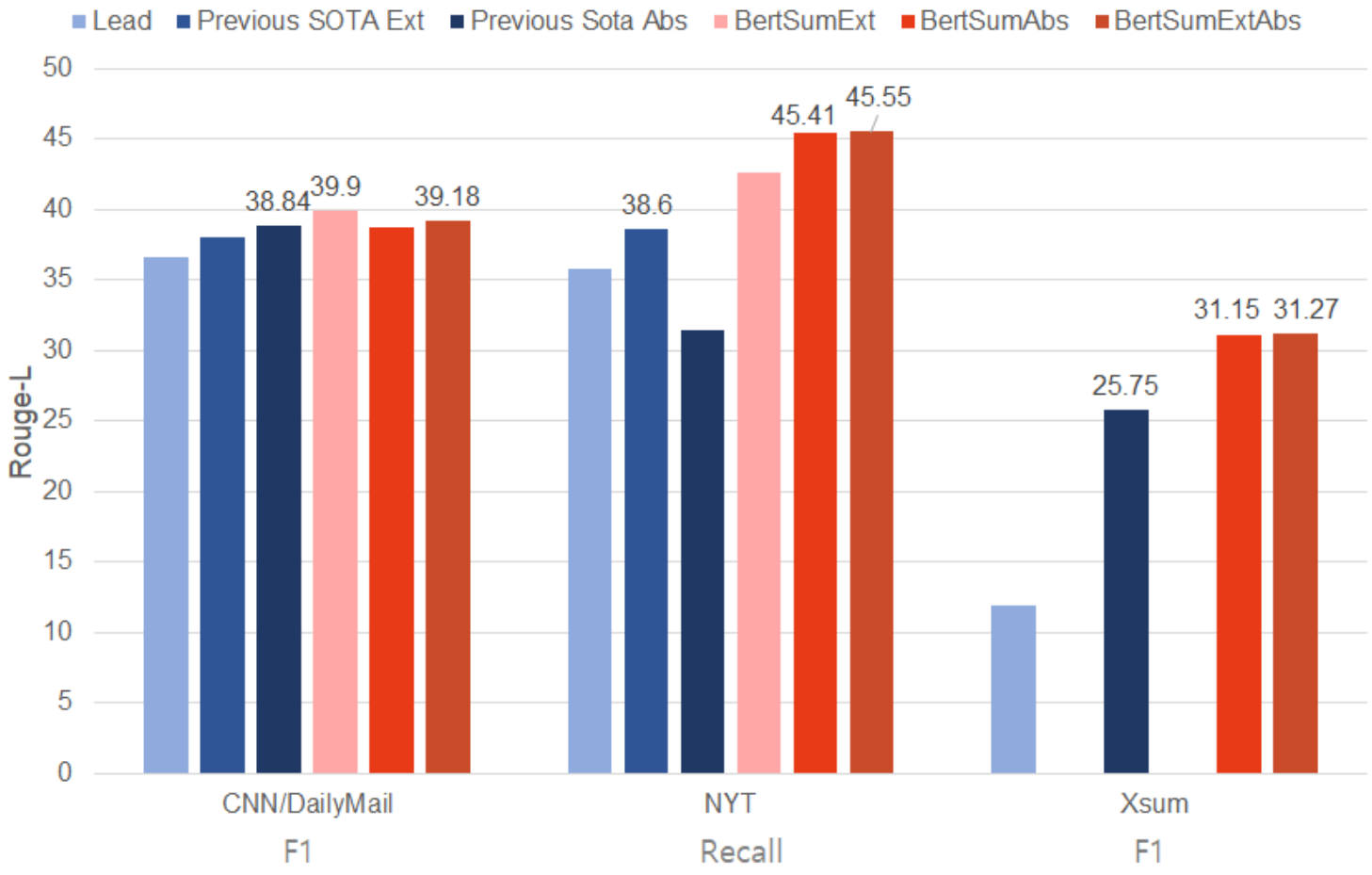
→

F1-score

들어가야하는 내용은 다 들어가면서 간결한 문장인지 평가

- 정답 요약문 y와 모델 요약문 y_hat 사이에 겹치는 단어 or 구(phrase)의 수로 판단
  - ROUGE-1 : 겹치는 1-gram
  - ROUGE-2 : 겹치는 2-gram
  - ROUGE-L : 겹치는 연속 sequence(LCS 기법)

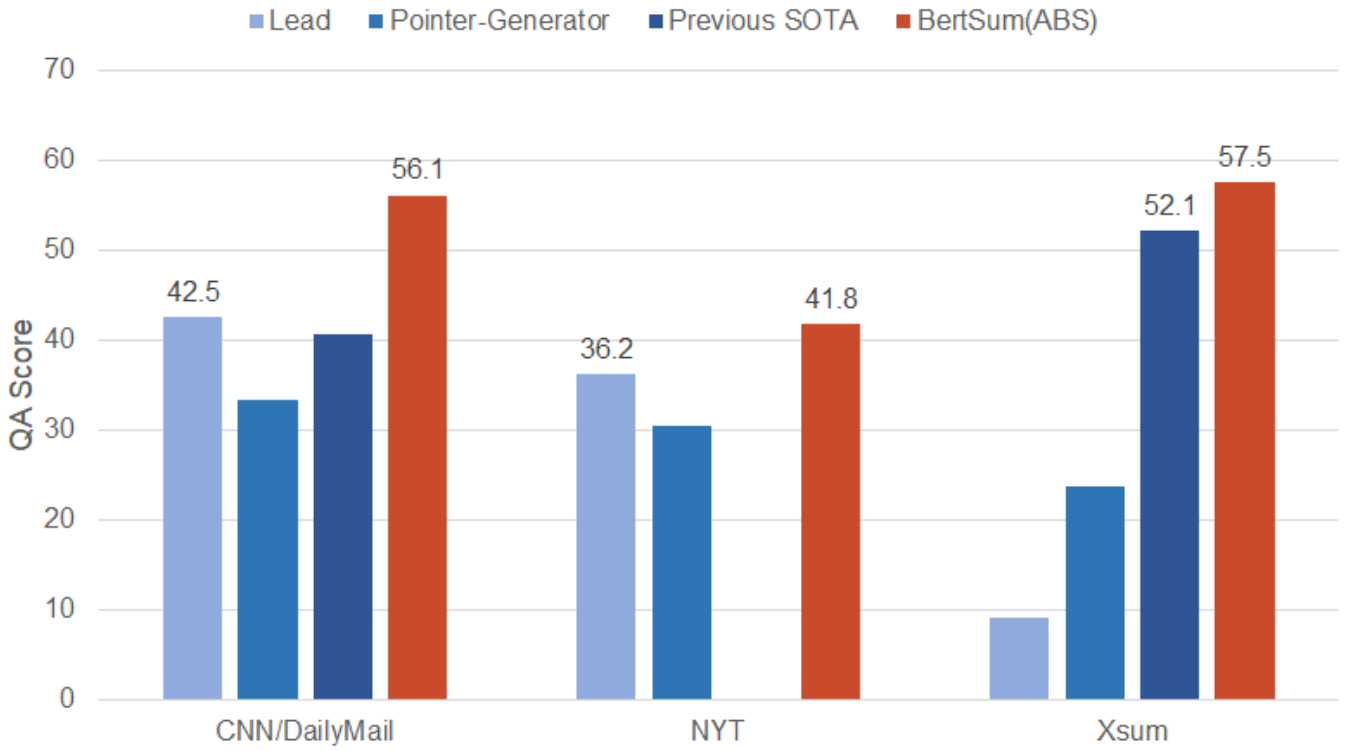
Result - Automatic Evaluation



Result - Human Evaluation1

Question Answering

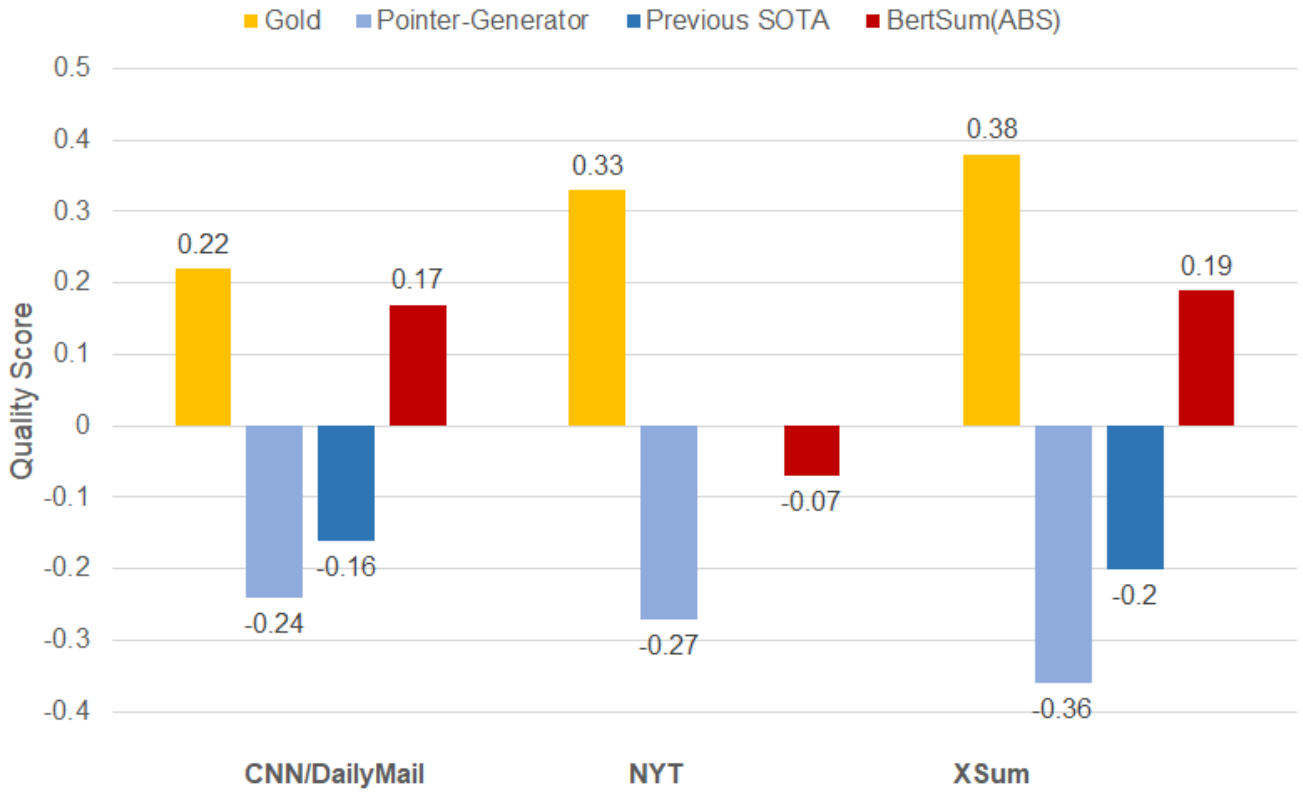
- 모델 요약문에 필요한 내용이 잘 들어갔는지 사람이 평가
- 정답 요약문으로 질문, 답변 생성 ▷ 사람이 모델 요약문만 보고 질문에 대한 답을 얼마나 할 수 있는지를 수치화



Result - Human Evaluation2

Quality Ranking

- 사람이 모델 요약문의 퀄리티를 평가(정보성, 유창성, 간결성)
- 참가자는 모델의 추출/생성 요약 결과와 원문을 비교해서 위의 기준을 토대로 best, worst를 지정하여 이를 -1~1로 정규화



5. Conclusion

Conclusion

1. BERT 모델을 요약 task에 적용
2. document-level encoder 소개
3. 추출요약과 생성요약 모두를 반영한 general framework를 제안(BERTSumEXTABS)
4. 실험 데이터셋에 대하여 SOTA를 달성(automatic, human-based 모두)

- We would like to take advantage the capabilities of BERT for language generation.

- 참고) https://github.com/uoneway/Text-Summarization-Repo