KOREA UNIVERSITY

EMNLP 2022 Papers

Lab Seminar

Sugyeong Eo

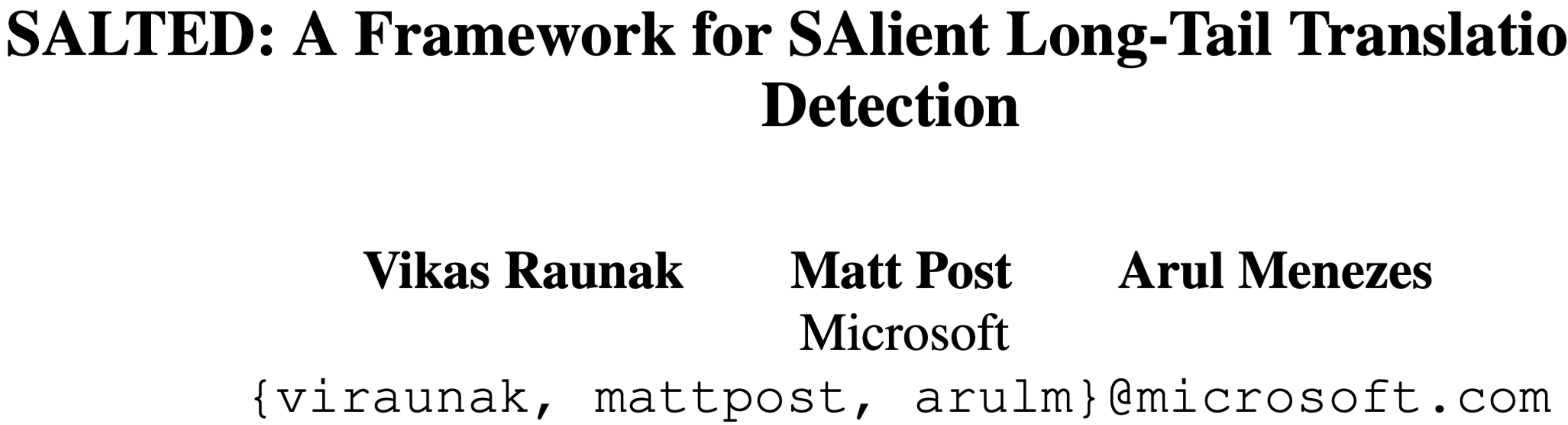
Three Papers

1. SALTED: A Framework for SAlient Long-tail Translation Error Detection
    - Oral presentation
    - Keywords: neural machine translation, quality estimation, error detection

2. CTRLsum: Towards Generic Controllable Text Summarization
    - Poster presentation
    - Keywords: summarization, keyword control, length control

3. SentBS: Sentence-level Beam Search for Controllable Summarization
    - Short paper
    - Keywords: summarization, decoding strategy

Paper 1



SALTED: A Framework for SAlient Long-Tail Translation Error Detection

Vikas Raunak **Matt Post** **Arul Menezes**
Microsoft
{viraunak, mattpost, arulm}@microsoft.com

1. Introduction

- Limitation in NMT:

1. Existing NMT metrics **do not provide fine-grained visibility** into rarer error categories

- Researchers **do not have a reliable way to gauge whether and to what extent a system may exhibit a wide range of negative behaviors** (e.g. hallucinations, dropped content, sporadic mistranslation of important content)

2. **These behaviors are rare** enough that they will not be observed on standard test sets

1. Introduction

- In this paper:

  1. They explore Behavioral Testing as a means to **provide find-grained measurements of salient long-tailed errors in MT**, while addressing the challenges of rarity and scalability in obtaining those measurements

  2. They propose an **iterative, specifications-based process for obtaining reliable measurements through high-precision detectors** and demonstrate their utility on seven MT error classes across both research and commercial systems

  3. They demonstrate that **detectors are amenable to multiple applications in MT**, including higher-recall training data filtering, system comparisons, metamorphic testing and fixing errors through fine-tuning on synthetic data

2. SALTED approach

- CHECKLIST(Ribeiro et al., 2020)
  - A process to construct test cases for evaluating different linguistic capabilities
  - Limitations:
    1. There could be **multiple valid translations** of the same input
    2. **Errors are highly contextual**: the same phenomenon may be translated accurately in one sentence, but inaccurately in another
    3. **Errors are unstable**: different model iterations may manifest errors in different sentences
    4. **Errors are rare**: a particular mistranslation may manifest itself only once in a million sentences

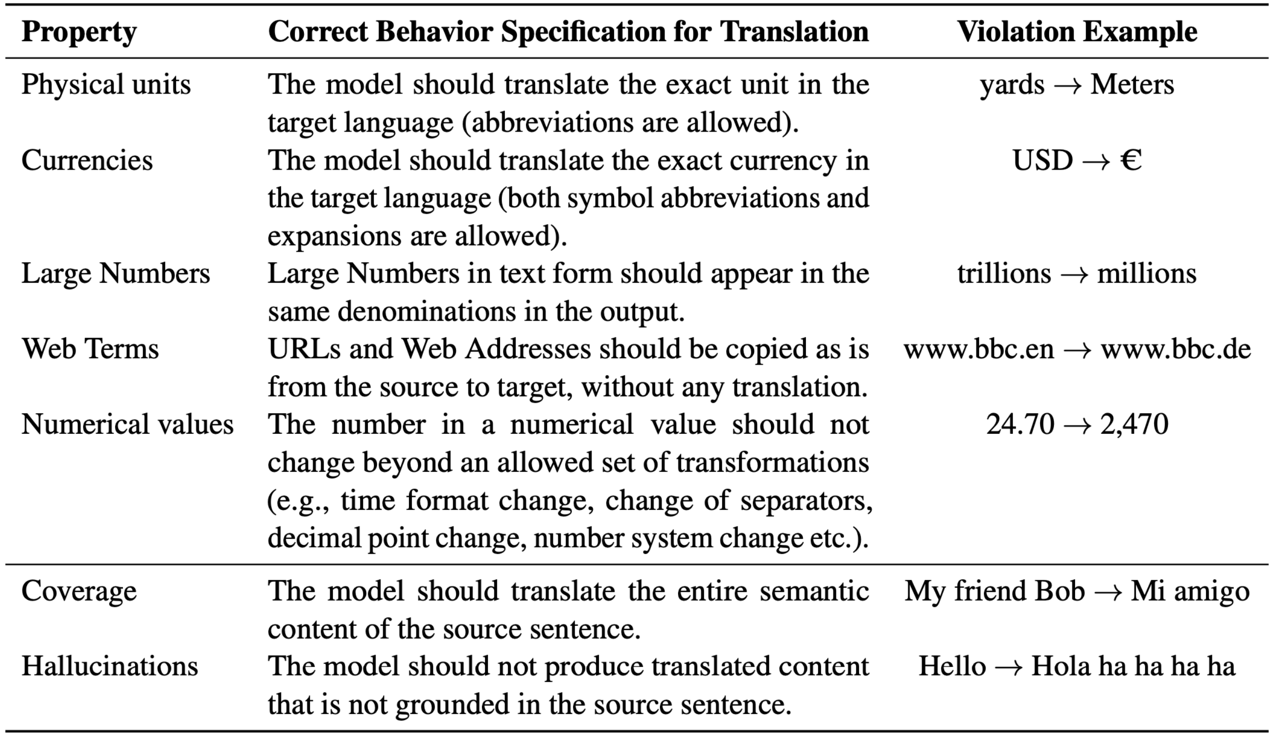
→ Test set을 만드는 것은 challenging

→ Test cases를 만드는 대신 *detectors*를 생성

→ *detectors*: an algorithm, which given an input-output sentence pair returns a **boolean value indicating the presence of an error condition with very high precision**

3. Behavior Specification for MT

- Detector를 만들기 위한 첫 번째 과정
- 특정한 salient content, property들의 번역에 대해서 모델이 반드시 잘 번역해야 하는 desired behaviors들을 구체화 (잘 번역되지 못했을 때 심각한 문제가 생기는 요소들로 한정)



Property	**Correct Behavior Specification for Translation**	**Violation Example**
Physical units	The model should translate the exact unit in the target language (abbreviations are allowed).	yards → Meters
Currencies	The model should translate the exact currency in the target language (both symbol abbreviations and expansions are allowed).	USD → €
Large Numbers	Large Numbers in text form should appear in the same denominations in the output.	trillions → millions
Web Terms	URLs and Web Addresses should be copied as is from the source to target, without any translation.	www.bbc.en → www.bbc.de
Numerical values	The number in a numerical value should not change beyond an allowed set of transformations (e.g., time format change, change of separators, decimal point change, number system change etc.).	24.70 → 2,470
Coverage	The model should translate the entire semantic content of the source sentence.	My friend Bob → Mi amigo
Hallucinations	The model should not produce translated content that is not grounded in the source sentence.	Hello → Hola ha ha ha ha

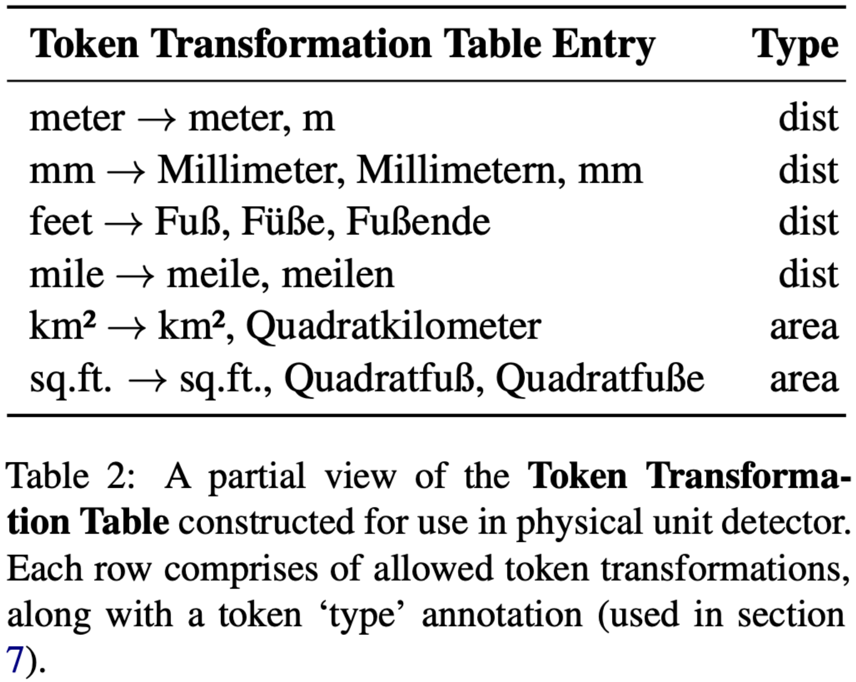
Table 1: **Behavior specification**: The first step in building a detector is to specify the correct expected behavior.

4. Designing Detector Algorithms

- Three detector construction process:
- 앞선 7가지 specifications 중 하나씩 차례로 다음 과정을 iteratively 거침
    1. Behavior specification
        - Desired behavior를 찾아냄 (10 feet -> 10 meters, 10 miles -> 16km는 모두 error)
    2. Resource construction
        - Transformation Table을 구축
        - relevant source tokens가 its set of potential translations에 잘 mapping되는지 확인
    3. Checking for specified behavior
        - Detector는 transformation table 내의 토큰이 소스에 등장할 때, 그에 매핑된 단어가 타겟이 되는지를 체크

→ 각 프로세스들은 human evaluation을 통한 error detection의 정밀도를 정량화하여 iterate됨

→ a large initial set of monolingual source sentence, MT에 대해 absolute precision을 얻을 때까지 위 과정을 iterate



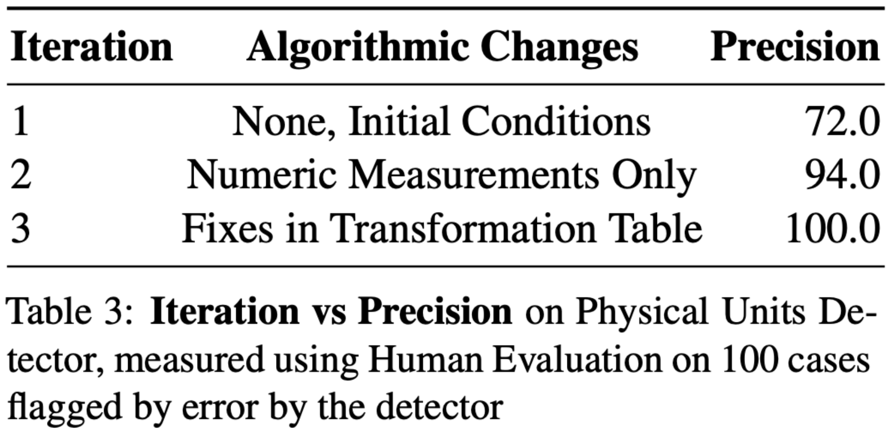
Token Transformation Table Entry	Type
meter → meter, m	dist
mm → Millimeter, Millimetern, mm	dist
feet → Fuß, Füße, Fußende	dist
mile → meile, meilen	dist
km^2 → km^2, Quadratkilometer	area
sq.ft. → sq.ft., Quadratfuß, Quadratfuße	area

Table 2: A partial view of the **Token Transformation Table** constructed for use in physical unit detector. Each row comprises of allowed token transformations, along with a token 'type' annotation (used in section 7).

4. Designing Detector Algorithms

- Iterations vs precision:
  - 매 iteration마다 detector를 1M random sample에 적용 -> 100개의 flagged cases에 대해 human eval 실시, precision 측정

- **Iteration 1:** A number of false positives를 발견 (e.g. idiomatic expressions “missed by a mile”, approximations “a few yards further”), 즉 detector가 맞게 번역된 애들을 잘못된 애들이라고 예측해버림
- **Iteration 2:** Error detection의 범위를 numeric measurements only로 좁힘 -> higher precision
- **Iteration 3:** Added/fixed entries to avoid FP



Iteration	Algorithmic Changes	Precision
1	None, Initial Conditions	72.0
2	Numeric Measurements Only	94.0
3	Fixes in Transformation Table	100.0

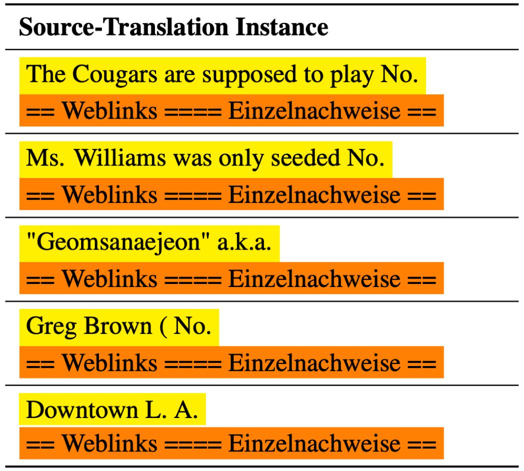
Table 3: **Iteration vs Precision** on Physical Units Detector, measured using Human Evaluation on 100 cases flagged by error by the detector

-> human eval에서 precision이 100이 되면 멈춤

-> high-precision을 지니는 detector 만들기 완료!

4. Designing Detector Algorithms

- Full Suite of Detectors
  - Token-level Detectors:
    - Physical units, Currencies, Large numbers, Web terms, Numerical values
    - Numerical values: fixed transformation table 대신 인스턴스마다의 transformation functions 활용. 숫자만 추출한 후 숫자 변환 format에 대해 변환해야 함 (time format change, change of separators, decimal point change, number system change etc.)
  - Sequence-level Detectors:
    - Coverage detector: src-tgt의 content words끼리 align으로 연결, unaligned content words가 threshold를 넘으면 error (소스 문장 길이의 k%를 넘으면 erroneous case로 간주)
    - Hallucinations: oscillatory + natural hallucinations
      - (oscillatory) 빈도수가 가장 높은 bigram이 소스에서 4번 이상 등장, 타겟에서 10번 이상 등장할 경우 error
      - (natural) 동일한 output에 대해 길이가 다른 source text가 5번 이상 나타나면 error



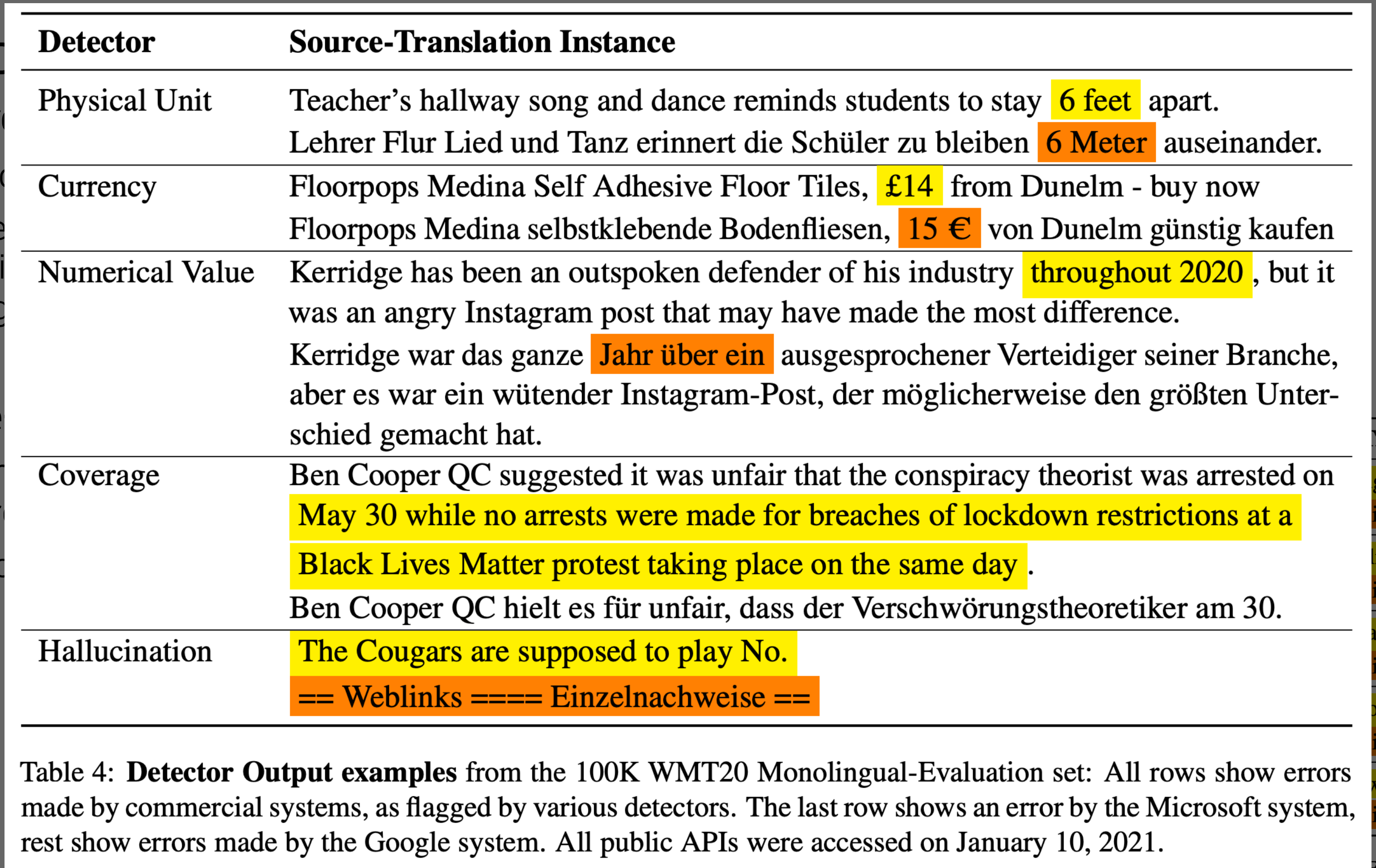
Source-Translation Instance
The Cougars are supposed to play No. == Weblinks ==== Einzelnachweise ==
Ms. Williams was only seeded No. == Weblinks ==== Einzelnachweise ==
"Geomsanaejeon" a.k.a. == Weblinks ==== Einzelnachweise ==
Greg Brown (No. == Weblinks ==== Einzelnachweise ==
Downtown L. A. == Weblinks ==== Einzelnachweise ==

Table 14: Examples of **Hallucinations** in one of the Commercial Translation Systems (Microsoft). The public API was accessed on January 10, 2021.

4. Designing Detector Algorithms

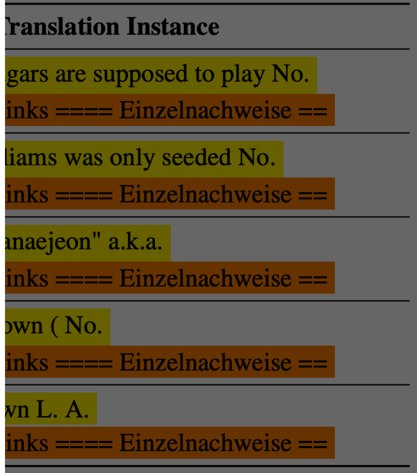
Detector Output Examples

- Full Suite of D
  - Token-lev
    - Physic
    - Nume functi chang
  - Sequence
    - Cover 가 thr
    - Halluc
      -
      -



Detector	Source-Translation Instance
Physical Unit	Teacher's hallway song and dance reminds students to stay 6 feet apart. Lehrer Flur Lied und Tanz erinnert die Schüler zu bleiben 6 Meter auseinander.
Currency	Floorpops Medina Self Adhesive Floor Tiles, £14 from Dunelm - buy now Floorpops Medina selbstklebende Bodenfliesen, 15 € von Dunelm günstig kaufen
Numerical Value	Kerridge has been an outspoken defender of his industry throughout 2020, but it was an angry Instagram post that may have made the most difference. Kerridge war das ganze Jahr über ein ausgesprochener Verteidiger seiner Branche, aber es war ein wütender Instagram-Post, der möglicherweise den größten Unterschied gemacht hat.
Coverage	Ben Cooper QC suggested it was unfair that the conspiracy theorist was arrested on May 30 while no arrests were made for breaches of lockdown restrictions at a Black Lives Matter protest taking place on the same day. Ben Cooper QC hielt es für unfair, dass der Verschwörungstheoretiker am 30.
Hallucination	The Cougars are supposed to play No. == Weblinks ==== Einzelnachweise ==

Table 4: **Detector Output examples** from the 100K WMT20 Monolingual-Evaluation set: All rows show errors made by commercial systems, as flagged by various detectors. The last row shows an error by the Microsoft system, rest show errors made by the Google system. All public APIs were accessed on January 10, 2021.

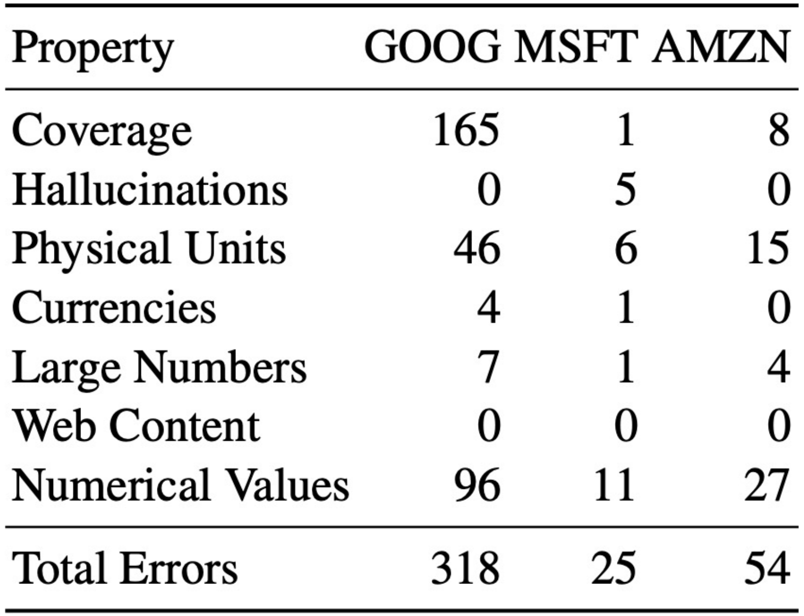


ranslation Instance
gars are supposed to play No. inks ==== Einzelnachweise ==
liams was only seeded No. inks ==== Einzelnachweise ==
naejeon" a.k.a. inks ==== Einzelnachweise ==
own (No. inks ==== Einzelnachweise ==
wn L. A. inks ==== Einzelnachweise ==

Table 14: Examples of **Hallucinations** in one of the Commercial Translation Systems (Microsoft). The public API was accessed on January 10, 2021.

5. Experiments

- Long-tailed error는 0.3% 등장할 정도로 rare하지만, NMT에서는 꽤나 만연하게 등장하는 유형들임

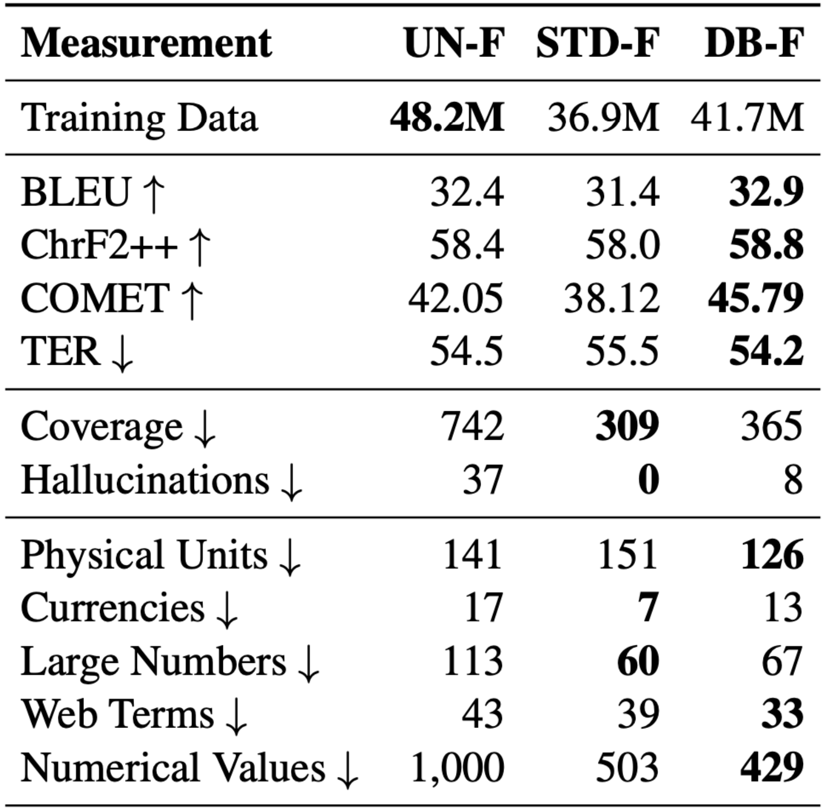


Property	GOOG	MSFT	AMZN
Coverage	165	1	8
Hallucinations	0	5	0
Physical Units	46	6	15
Currencies	4	1	0
Large Numbers	7	1	4
Web Content	0	0	0
Numerical Values	96	11	27
Total Errors	318	25	54

Table 5: Counts of **Erroneous Translations** found by Detectors in the 100K WMT20 Monolingual Eval Set.

5. Experiments

- Unfiltered (UN-F): EN-DE 로 학습시킨 일반 NMT 모델
- Standard (STD-F): Wu et al., (2020)에서 활용한 filtering method를 활용
- Detector-based (DB-F): 이 논문의 detectors를 통해 filtering



Measurement	**UN-F**	**STD-F**	**DB-F**
Training Data	**48.2M**	36.9M	41.7M
BLEU ↑	32.4	31.4	**32.9**
ChrF2++ ↑	58.4	58.0	**58.8**
COMET ↑	42.05	38.12	**45.79**
TER ↓	54.5	55.5	**54.2**
Coverage ↓	742	**309**	365
Hallucinations ↓	37	**0**	8
Physical Units ↓	141	151	**126**
Currencies ↓	17	**7**	13
Large Numbers ↓	113	**60**	67
Web Terms ↓	43	39	**33**
Numerical Values ↓	1,000	503	**429**

Table 6: **Metric Based** System Comparisons on the WMT20 Test set and **Detector Based** system comparisons on the 1M Mono-Eval Set for the three systems. Note that ↓ implies lower is better, ↑ implies otherwise.

5. Experiments

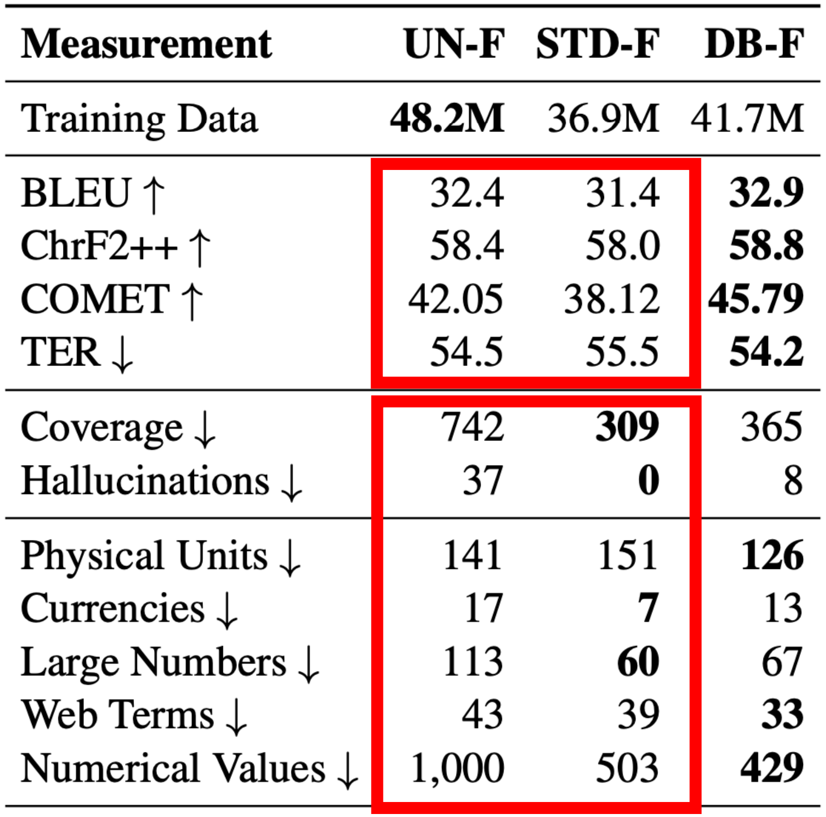
- Unfiltered (UN-F): EN-DE 로 학습시킨 일반 NMT 모델
- Standard (STD-F): Wu et al., (2020)에서 활용한 filtering method를 활용
- Detector-based (DB-F): 이 논문의 detectors를 통해 filtering

1. UN-F VS STD-F

→ BLEU, TER은 Unfiltered case가 가장 좋음

→ 그러나 Hallucination 등 전체적인 detectors에서는 filtering한 결과가 더 낮은 error를 report

→ standard metrics로는 이런 error들을 잘 찾아내지 못함을 발견



Measurement	**UN-F**	**STD-F**	**DB-F**
Training Data	**48.2M**	36.9M	41.7M
BLEU ↑	32.4	31.4	**32.9**
ChrF2++ ↑	58.4	58.0	**58.8**
COMET ↑	42.05	38.12	**45.79**
TER ↓	54.5	55.5	**54.2**
Coverage ↓	742	**309**	365
Hallucinations ↓	37	**0**	8
Physical Units ↓	141	151	**126**
Currencies ↓	17	**7**	13
Large Numbers ↓	113	**60**	67
Web Terms ↓	43	39	**33**
Numerical Values ↓	1,000	503	**429**

Table 6: **Metric Based** System Comparisons on the WMT20 Test set and **Detector Based** system comparisons on the 1M Mono-Eval Set for the three systems. Note that ↓ implies lower is better, ↑ implies otherwise.

5. Experiments

- Unfiltered (UN-F): EN-DE 로 학습시킨 일반 NMT 모델
- Standard (STD-F): Wu et al., (2020)에서 활용한 filtering method를 활용
- Detector-based (DB-F): 이 논문의 detectors를 통해 filtering

1. UN-F VS STD-F

→ BLEU, TER은 Unfiltered case가 가장 좋음

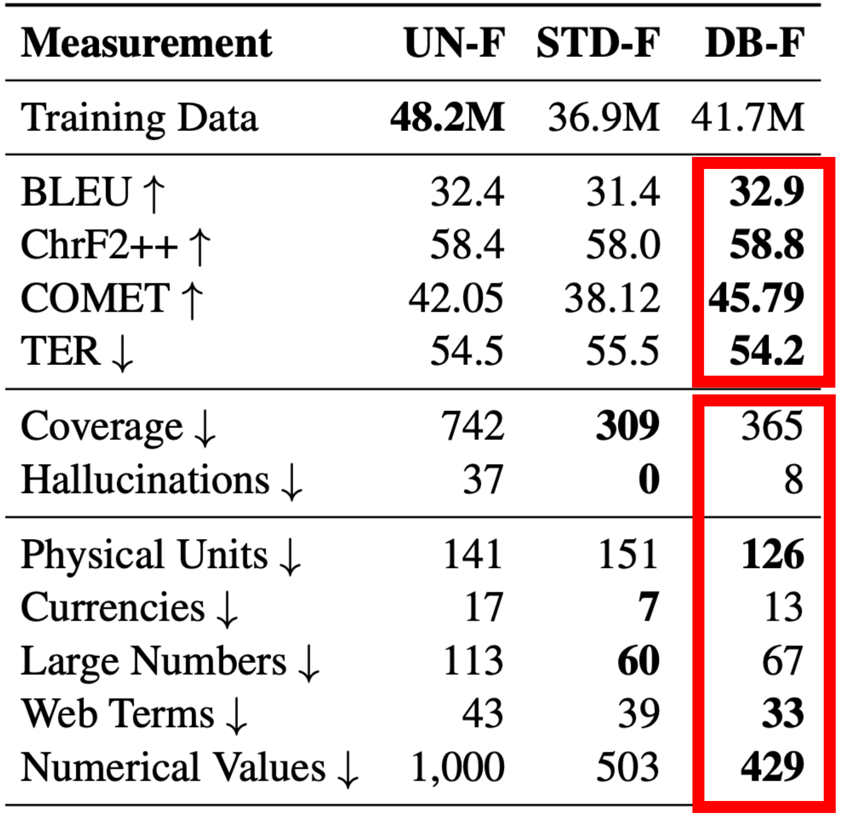
→ 그러나 Hallucination 등 전체적인 detectors에서는 filtering한 결과가 더 낮은 error를 report

→ standard metrics로는 이런 error들을 잘 찾아내지 못함을 발견

2. DB-F VS UN-F/STD-F

→ BLEU, TER 등에서 improvement를 보임

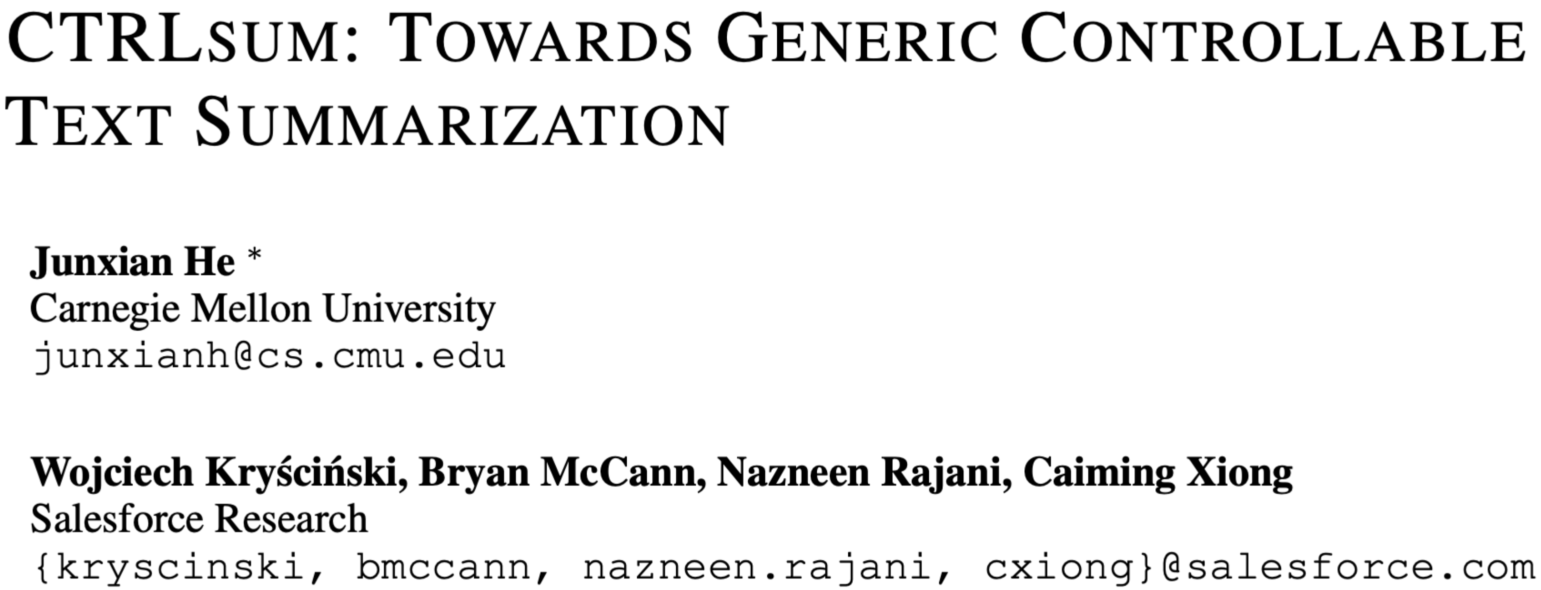
→ 동시에, STD-F와 유사하게 낮아진 error가 detect됨



Measurement	UN-F	STD-F	DB-F
Training Data	**48.2M**	36.9M	41.7M
BLEU ↑	32.4	31.4	**32.9**
ChrF2++ ↑	58.4	58.0	**58.8**
COMET ↑	42.05	38.12	**45.79**
TER ↓	54.5	55.5	**54.2**
Coverage ↓	742	**309**	365
Hallucinations ↓	37	**0**	8
Physical Units ↓	141	151	**126**
Currencies ↓	17	**7**	13
Large Numbers ↓	113	**60**	67
Web Terms ↓	43	39	**33**
Numerical Values ↓	1,000	503	**429**

Table 6: **Metric Based** System Comparisons on the WMT20 Test set and **Detector Based** system comparisons on the 1M Mono-Eval Set for the three systems. Note that ↓ implies lower is better, ↑ implies otherwise.

Paper 2



CTRLsum: Towards Generic Controllable Text Summarization

Junxian He *
Carnegie Mellon University
junxianh@cs.cmu.edu

Wojciech Kryściński, Bryan McCann, Nazneen Rajani, Caiming Xiong
Salesforce Research
{kryscinski, bmccann, nazneen.rajani, cxiong}@salesforce.com

1. Introduction

- Abstractive summarization: 주어진 document에 대해 summary를 생성하는 task
- Limitation: Summaries should select information with respect to **preferences of a user**
  - e.g. NBA 농구 경기에 대한 문서 → match result에 대한 summary를 생성, but users는 특정 농구 스타에 대한 정보를 알고 싶을 수 있음

- In this paper,
  1. **Controllable summarization** which allows the users to manipulate the summaries from the model
  2. Introducing **CTRLSum, a framework to control summaries** through control tokens in the form of a set of keywords or descriptive prompts

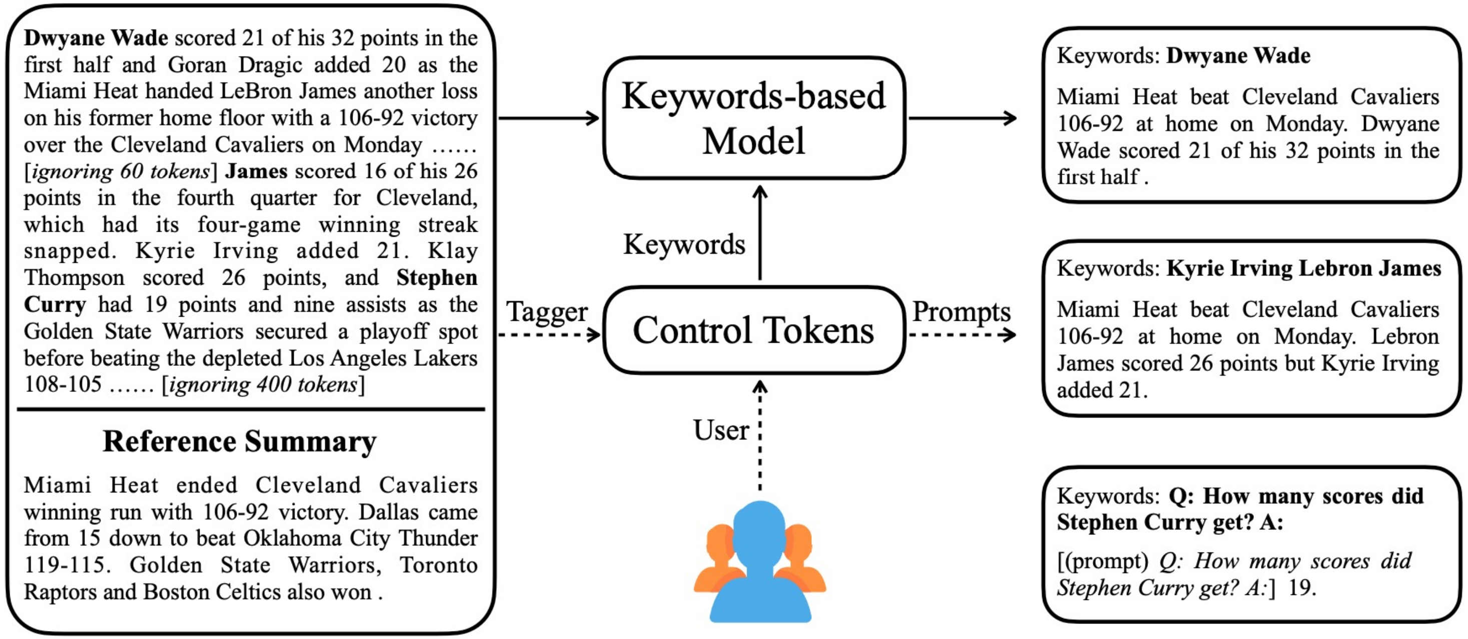
- Overview:

  Training → source document, keywords(external guidance)를 활용해 summary 생성

  Inference → with keywords and optimal prompts, summary 생성

2. CTRLsum

- CTRLSum



Dwyane Wade scored 21 of his 32 points in the first half and Goran Dragic added 20 as the Miami Heat handed LeBron James another loss on his former home floor with a 106-92 victory over the Cleveland Cavaliers on Monday …… [*ignoring 60 tokens*] **James** scored 16 of his 26 points in the fourth quarter for Cleveland, which had its four-game winning streak snapped. Kyrie Irving added 21. Klay Thompson scored 26 points, and **Stephen Curry** had 19 points and nine assists as the Golden State Warriors secured a playoff spot before beating the depleted Los Angeles Lakers 108-105 …… [*ignoring 400 tokens*]

Reference Summary

Miami Heat ended Cleveland Cavaliers winning run with 106-92 victory. Dallas came from 15 down to beat Oklahoma City Thunder 119-115. Golden State Warriors, Toronto Raptors and Boston Celtics also won .

Keywords: **Dwyane Wade**

Miami Heat beat Cleveland Cavaliers 106-92 at home on Monday. Dwyane Wade scored 21 of his 32 points in the first half .

Keywords: **Kyrie Irving Lebron James**

Miami Heat beat Cleveland Cavaliers 106-92 at home on Monday. Lebron James scored 26 points but Kyrie Irving added 21.

Keywords: **Q: How many scores did Stephen Curry get? A:**

[(prompt) *Q: How many scores did Stephen Curry get? A:*] 19.

Figure 1: Workflow of the CTRLsum framework at inference time. Users interact with summaries through textual control tokens in the form of keywords or prompts. Keywords are required as input during training and testing, while prompts are optionally used at test time. Dashed lines represent optional paths – control tokens can come from the source article, user, or both. The right portion of the figure shows actual outputs from CTRLsum.

2. CTRLsum

- Automatic keyword extraction
  - Source document에서 키워드를 찾기 위해 GT summary를 활용
  - (1) reference summary마다 rouge score가 가장 높은 문장들을 선택
  - (2) 추출된 문장들 중 reference summary와 매치되는 가장 긴 sub-sequences를 추출
  - (3) duplicate words and stop words 제거

  → few salient words를 추출하는 것보다 summary 내 most content words를 찾아낼 수 있음

  → User provided keywords가 summary 생성 시 무시되지 않도록 유도

  - BERT-based sequence tagger 학습: [input] document [output] keywords
  - Tagger는 token마다의 selection probability(q_j)를 계산 → 문장마다의 average token selection probability를 계산한 후, probability가 높은 순서대로 문장 n_s개 추출 → 이 문장들 내에서, maximum number $\mathrm{m}_{\max}$ 까지 $q_j > \varepsilon$ 인 경우를 추출 (n_s, q_j, ε는 하이퍼파라미터)

2. CTRLsum

- Summarization: Training details
  - Maximize $p(y|x,z)$ * x:document, z:keyword
  - Keyword 무시를 방지하기 위해 separator “|” 추가
  - Keyword에 대한 지나친 dependency로 인해 novel words 생성 어려움 → randomly drop keywords at training time (only at training time)

- Summarization: Inference with keywords
  - **Entity Control**: summaries that focus on entities of interest
  - **Length Control**: user-specified length parameter

    (1) Summary 길이에 따라 training data를 동일한 example 개수를 지닌 5 buckets로 분할

    (2) 각 bucket마다의 keywords 평균 개수를 구함

    → 사용자는 제시된 keyword를 최대 몇 개 포함할지를 선택함으로써 length를 조절

3. Experiments

- Oracle entity: GT summary에서부터 얻은 keywords로 summarization 수행
- The use of oracle entities helps boost the ROUGE-2 score by 3.6 points compared with using automatic keywords

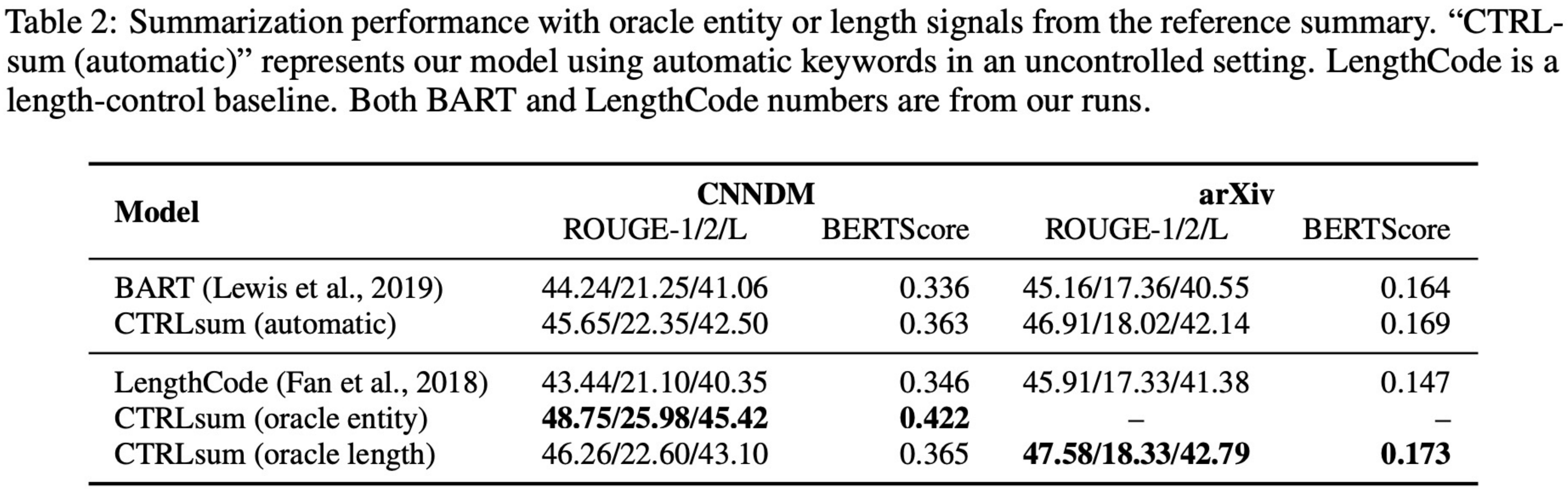


Table 2: Summarization performance with oracle entity or length signals from the reference summary. "CTRLsum (automatic)" represents our model using automatic keywords in an uncontrolled setting. LengthCode is a length-control baseline. Both BART and LengthCode numbers are from our runs.

Model	**CNNDM**		**arXiv**	
	ROUGE-1/2/L	BERTScore	ROUGE-1/2/L	BERTScore
BART (Lewis et al., 2019)	44.24/21.25/41.06	0.336	45.16/17.36/40.55	0.164
CTRLsum (automatic)	45.65/22.35/42.50	0.363	46.91/18.02/42.14	0.169
LengthCode (Fan et al., 2018)	43.44/21.10/40.35	0.346	45.91/17.33/41.38	0.147
CTRLsum (oracle entity)	**48.75/25.98/45.42**	**0.422**	–	–
CTRLsum (oracle length)	46.26/22.60/43.10	0.365	**47.58/18.33/42.79**	**0.173**

3. Experiments

- Success rate: the fraction of decoded summaries that actually mention the given entity
- Factual Correctness: s the fraction of summaries that are judged as factually correct by human annotators
- Entity 중요도 여부와 관계없이 사실적으로 일관된 요약을 생성함을 확인

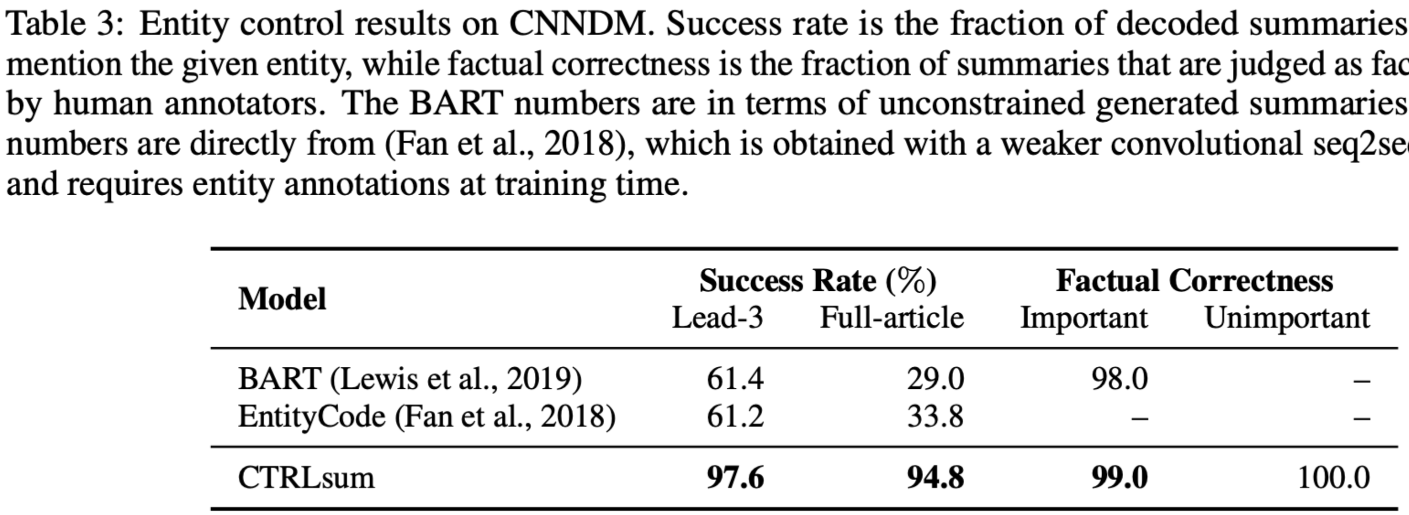
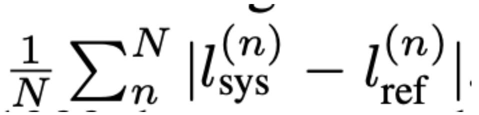


Table 3: Entity control results on CNNDM. Success rate is the fraction of decoded summaries that actually mention the given entity, while factual correctness is the fraction of summaries that are judged as factually correct by human annotators. The BART numbers are in terms of unconstrained generated summaries. EntityCode numbers are directly from (Fan et al., 2018), which is obtained with a weaker convolutional seq2seq architecture and requires entity annotations at training time.

Model	**Success Rate** (%)		**Factual Correctness**	
	Lead-3	Full-article	Important	Unimportant
BART (Lewis et al., 2019)	61.4	29.0	98.0	–
EntityCode (Fan et al., 2018)	61.2	33.8	–	–
CTRLsum	**97.6**	**94.8**	**99.0**	100.0

3. Experiments

- Mean of absolute deviation (MAD): deviation of output length from reference length
- Pearson Correlation Coefficient (PCC): correlation between given length signal and actual output length



$$\frac{1}{N}\sum_{n}^{N}|l_{\text{sys}}^{(n)} - l_{\text{ref}}^{(n)}|$$

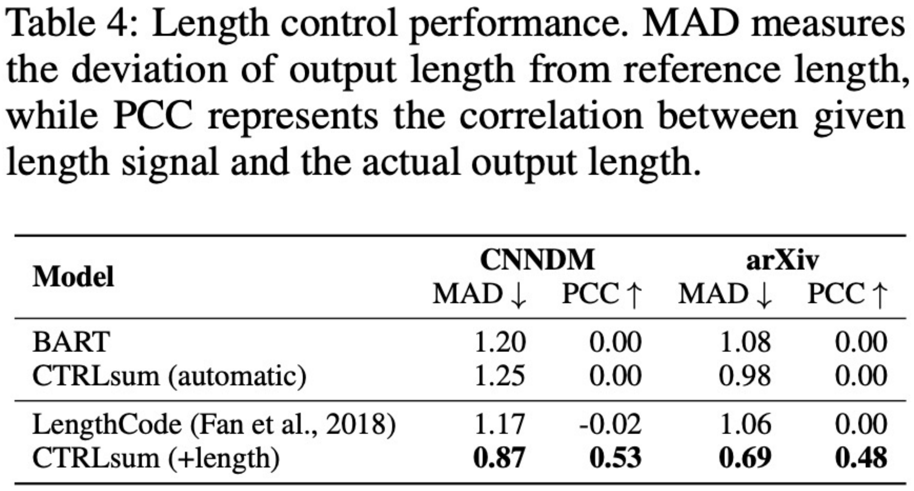
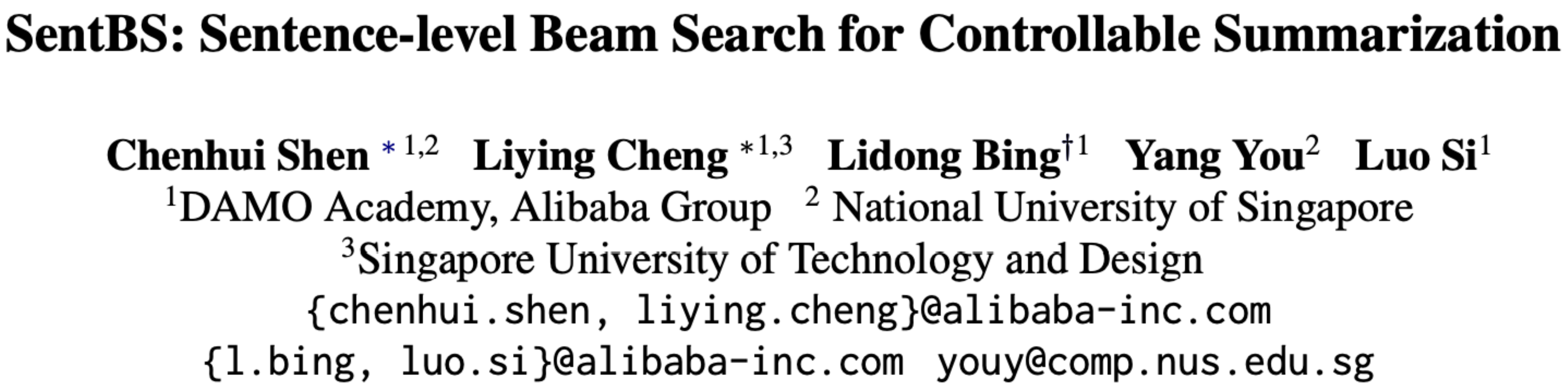


Table 4: Length control performance. MAD measures the deviation of output length from reference length, while PCC represents the correlation between given length signal and the actual output length.

Model	**CNNDM**		**arXiv**	
	MAD ↓	PCC ↑	MAD ↓	PCC ↑
BART	1.20	0.00	1.08	0.00
CTRLsum (automatic)	1.25	0.00	0.98	0.00
LengthCode (Fan et al., 2018)	1.17	-0.02	1.06	0.00
CTRLsum (+length)	**0.87**	**0.53**	**0.69**	**0.48**

Paper 3



SentBS: Sentence-level Beam Search for Controllable Summarization

Chenhui Shen *[1,2] **Liying Cheng** *[1,3] **Lidong Bing**†[1] **Yang You**[2] **Luo Si**[1]
[1]DAMO Academy, Alibaba Group [2] National University of Singapore
[3]Singapore University of Technology and Design
{chenhui.shen, liying.cheng}@alibaba-inc.com
{l.bing, luo.si}@alibaba-inc.com youy@comp.nus.edu.sg

1. Introduction

- **Controllable text generation** focuses on controlling the structure of the output summary
- Input: reviews on the research paper, control sequence: “abstract | strength | decision”

  -> summarizes the contents of the paper, followed by a sentence discussing the strengths, then the last sentence giving the final decision

- **Motivation:**
  - Previous works mainly focus on improving the summary’s similarity with the gold reference, **leaving room for further improvement on the controllability**
  - autoregressive models can **suffer from error propagation** in generation due to self-attention

  -> if the previous sequences are not well-controlled, subsequent generations may deviate further from the desired output

- **In this paper:**
  - They enhance the structure-controllability in summarization
  - Introducing Sentence-level Beam Search (SentBS) method

2. Method

Sequence likelihood: log-likelihood of each generated token from the decoder, and then calculate the normalized sequence log-likelihood by averaging the loglikelihood for all tokens in the sequence

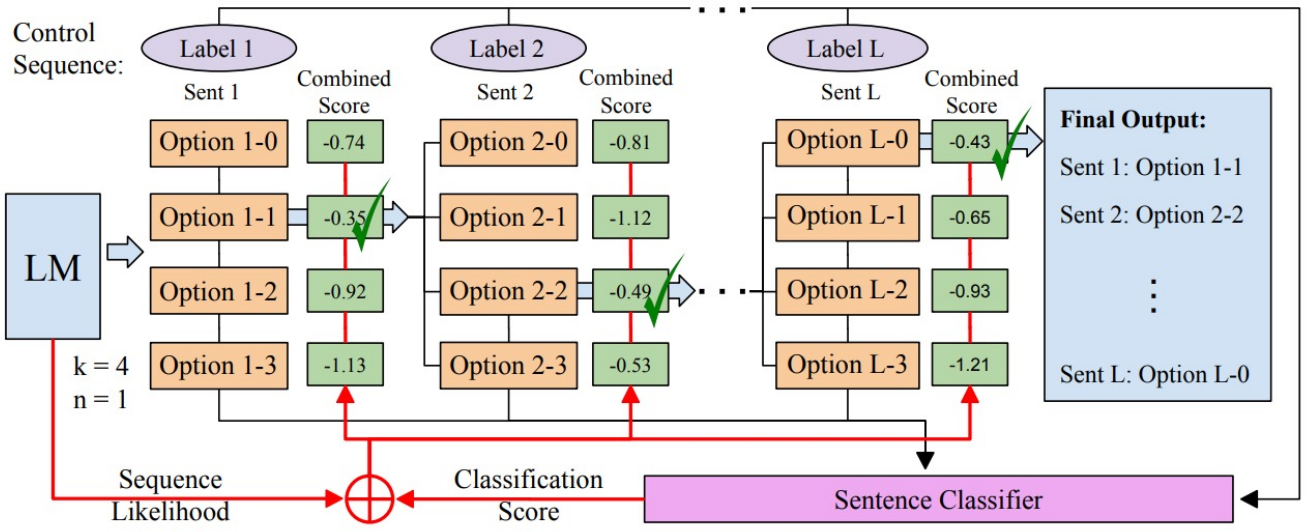


Figure 1: Illustration of SentBS. The score values are for illustration purposes only. For simplicity, we only illustrate for $k = 4$ and $n = 1$.

(1) SentBS **generates k sentence** options in parallel using multiple decoding strategies

Sequence likelihood: log-likelihood of each generated token from the decoder, and then calculate the normalized sequence log-likelihood by averaging the loglikelihood for all tokens in the sequence

2. Method

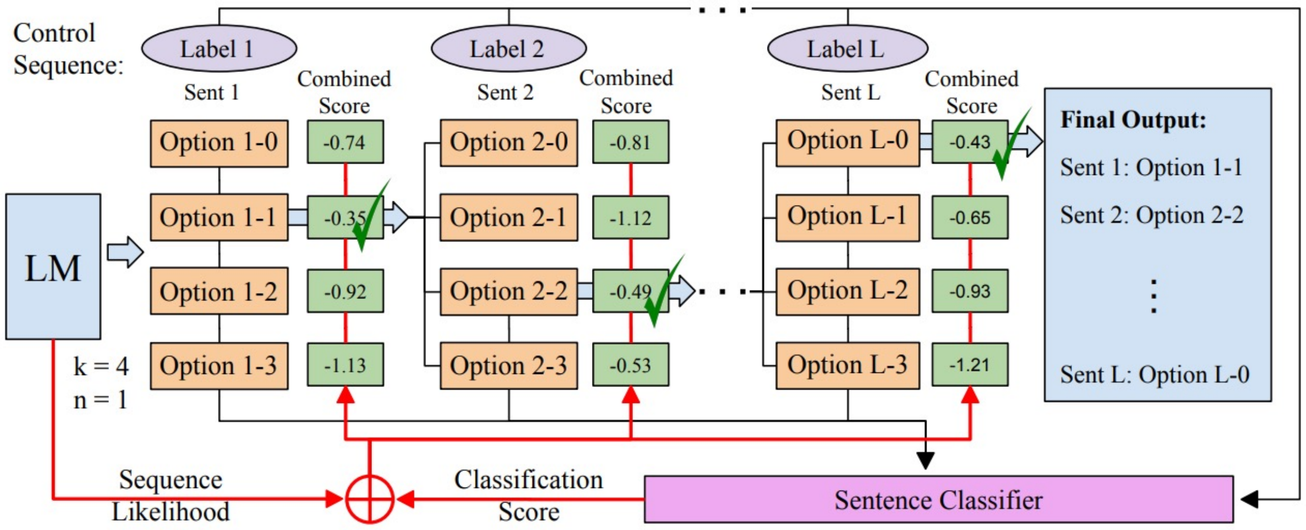


Figure 1: Illustration of SentBS. The score values are for illustration purposes only. For simplicity, we only illustrate for $k = 4$ and $n = 1$.

(1) SentBS **generates k sentence** options in parallel using multiple decoding strategies

(2) Calculating **combined score** for each sentence: normalized sequence likelihood + classifier-predicted probability score of the sentence belonging to the required category

2. Method

Sequence likelihood: log-likelihood of each generated token from the decoder, and then calculate the normalized sequence log-likelihood by averaging the loglikelihood for all tokens in the sequence

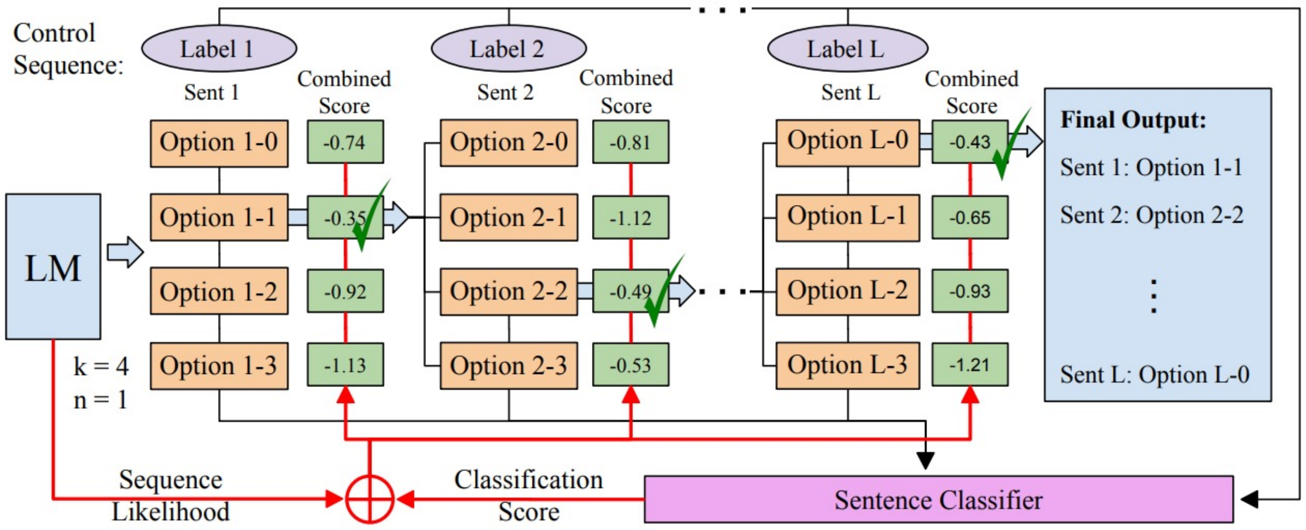


Figure 1: Illustration of SentBS. The score values are for illustration purposes only. For simplicity, we only illustrate for $k = 4$ and $n = 1$.

(1) SentBS **generates k sentence** options in parallel using multiple decoding strategies

(2) Calculating **combined score** for each sentence: normalized sequence likelihood + classifier-predicted probability score of the sentence belonging to the required category

(3) Top n sentences are selected and **feed them individually into the decoder as prompts** for generating the next sentence.

2. Method

Sequence likelihood: log-likelihood of each generated token from the decoder, and then calculate the normalized sequence log-likelihood by averaging the loglikelihood for all tokens in the sequence

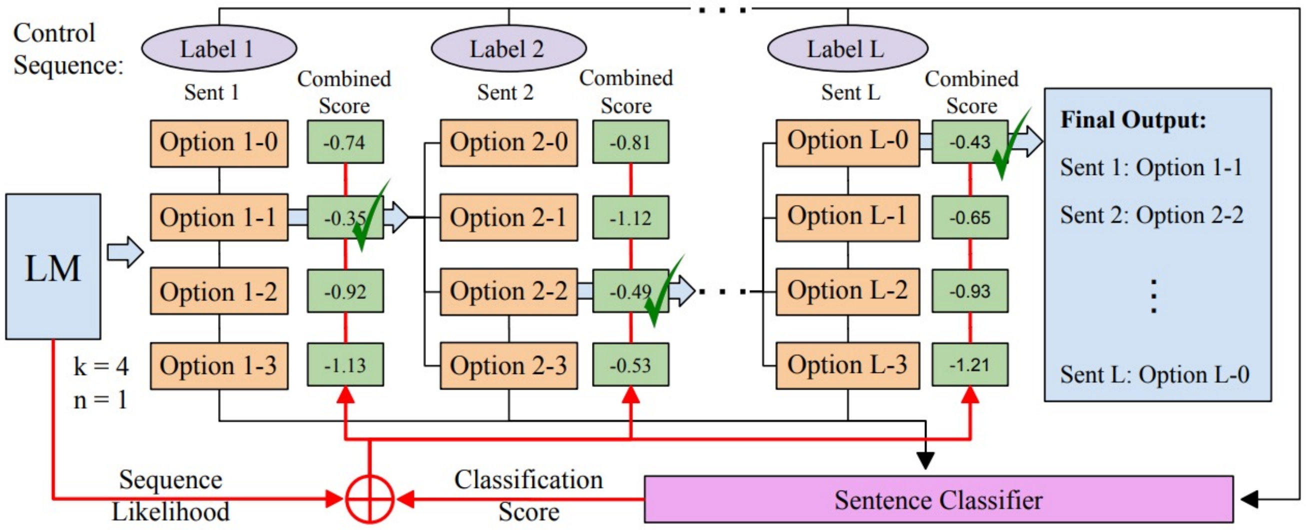


Figure 1: Illustration of SentBS. The score values are for illustration purposes only. For simplicity, we only illustrate for $k = 4$ and $n = 1$.

(1) SentBS **generates k sentence** options in parallel using multiple decoding strategies

(2) Calculating **combined score** for each sentence: normalized sequence likelihood + classifier-predicted probability score of the sentence belonging to the required category

(3) Top n sentences are selected and **feed them individually into the decoder as prompts** for generating the next sentence.

(4) The same generation process continues **until all sentences required in the control sequence are produced.**

→ k subsequent sentence options for each of the n prompts: k*n output

2. Method

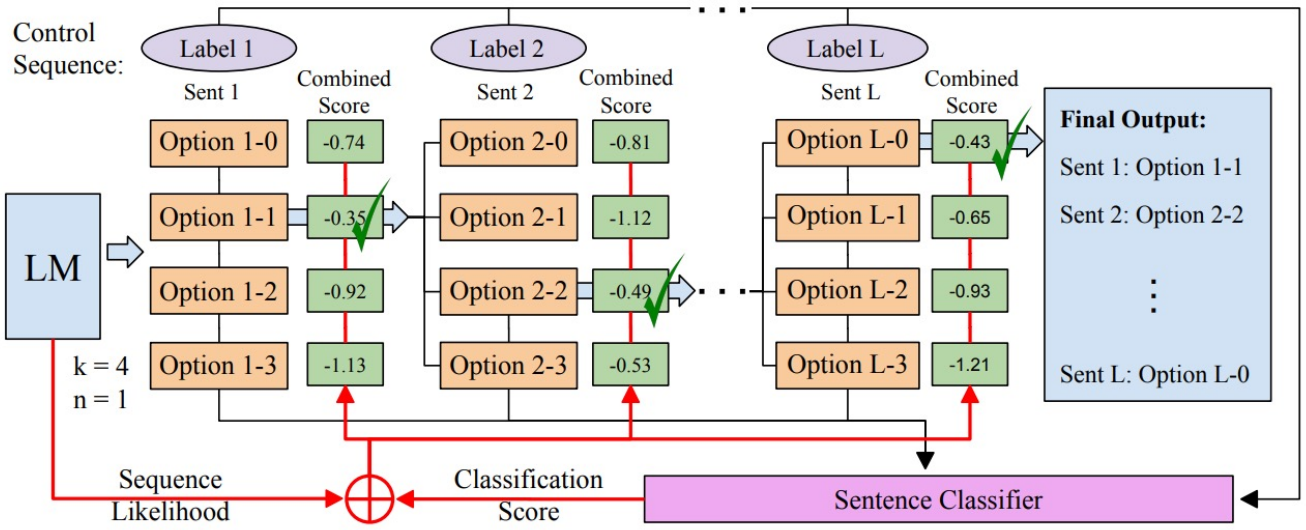


Figure 1: Illustration of SentBS. The score values are for illustration purposes only. For simplicity, we only illustrate for $k = 4$ and $n = 1$.

Control sequence: “abstract | strength | decision”

(1) Generate the first “abstract” sentence

(2) Get classification score on both “abstract” and “strength” for the second sentence

(3) Assign the label with higher probability to it

(4) second sentence is assigned a label of either “abstract” or “strength”, depending on the generated content

(5) subsequent sentences until the model gives the stop signal

3. Experiments

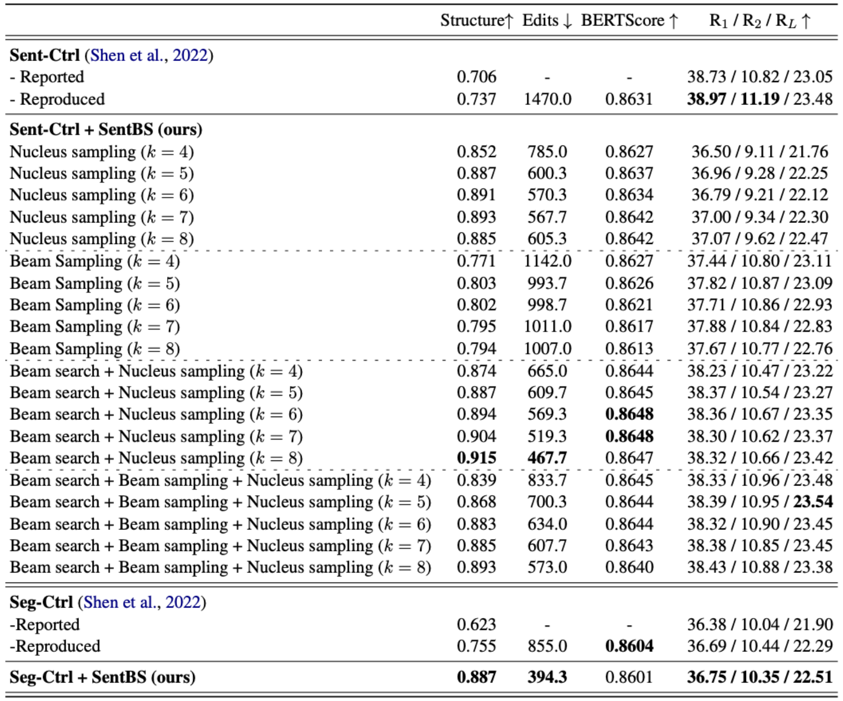
Dataset: MReD dataset (Shen et al., 2022)

-> Each sentence in the target meta-review is annotated with a category label (“abstract”, “strength”, “weakness”, “suggestion”, “rating summary”, “rebuttal process”, “ac disagreement”, “decision”, and “misc”)

Sent-Ctrl: 문장마다의 label을 활용하는 setting

Seg-Ctrl: segment마다의 label을 활용하는 setting

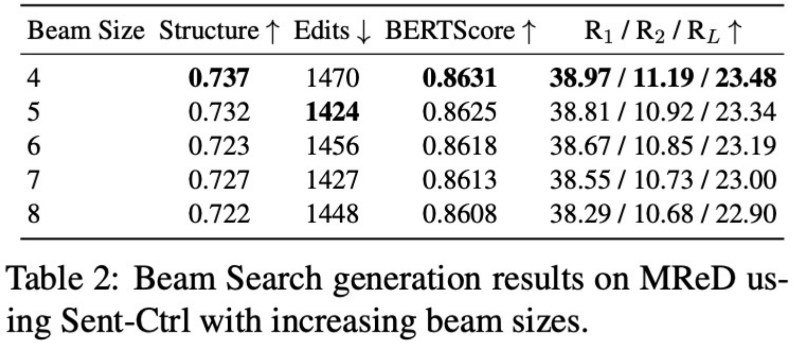
- Human eval: structure similarity
- Edit distance between summary structure and the golc structure



	Structure↑	Edits ↓	BERTScore ↑	R_1 / R_2 / R_L ↑
Sent-Ctrl (Shen et al., 2022)				
- Reported	0.706	-	-	38.73 / 10.82 / 23.05
- Reproduced	0.737	1470.0	0.8631	**38.97 / 11.19** / 23.48
Sent-Ctrl + SentBS (ours)				
Nucleus sampling ($k=4$)	0.852	785.0	0.8627	36.50 / 9.11 / 21.76
Nucleus sampling ($k=5$)	0.887	600.3	0.8637	36.96 / 9.28 / 22.25
Nucleus sampling ($k=6$)	0.891	570.3	0.8634	36.79 / 9.21 / 22.12
Nucleus sampling ($k=7$)	0.893	567.7	0.8642	37.00 / 9.34 / 22.30
Nucleus sampling ($k=8$)	0.885	605.3	0.8642	37.07 / 9.62 / 22.47
Beam Sampling ($k=4$)	0.771	1142.0	0.8627	37.44 / 10.80 / 23.11
Beam Sampling ($k=5$)	0.803	993.7	0.8626	37.82 / 10.87 / 23.09
Beam Sampling ($k=6$)	0.802	998.7	0.8621	37.71 / 10.86 / 22.93
Beam Sampling ($k=7$)	0.795	1011.0	0.8617	37.88 / 10.84 / 22.83
Beam Sampling ($k=8$)	0.794	1007.0	0.8613	37.67 / 10.77 / 22.76
Beam search + Nucleus sampling ($k=4$)	0.874	665.0	0.8644	38.23 / 10.47 / 23.22
Beam search + Nucleus sampling ($k=5$)	0.887	609.7	0.8645	38.37 / 10.54 / 23.27
Beam search + Nucleus sampling ($k=6$)	0.894	569.3	**0.8648**	38.36 / 10.67 / 23.35
Beam search + Nucleus sampling ($k=7$)	0.904	519.3	**0.8648**	38.30 / 10.62 / 23.37
Beam search + Nucleus sampling ($k=8$)	**0.915**	**467.7**	0.8647	38.32 / 10.66 / 23.42
Beam search + Beam sampling + Nucleus sampling ($k=4$)	0.839	833.7	0.8645	38.33 / 10.96 / 23.48
Beam search + Beam sampling + Nucleus sampling ($k=5$)	0.868	700.3	0.8644	38.39 / 10.95 / **23.54**
Beam search + Beam sampling + Nucleus sampling ($k=6$)	0.883	634.0	0.8644	38.32 / 10.90 / 23.45
Beam search + Beam sampling + Nucleus sampling ($k=7$)	0.885	607.7	0.8643	38.38 / 10.85 / 23.45
Beam search + Beam sampling + Nucleus sampling ($k=8$)	0.893	573.0	0.8640	38.43 / 10.88 / 23.38
Seg-Ctrl (Shen et al., 2022)				
-Reported	0.623	-	-	36.38 / 10.04 / 21.90
-Reproduced	0.755	855.0	**0.8604**	36.69 / 10.44 / 22.29
Seg-Ctrl + SentBS (ours)	**0.887**	**394.3**	0.8601	**36.75 / 10.35 / 22.51**

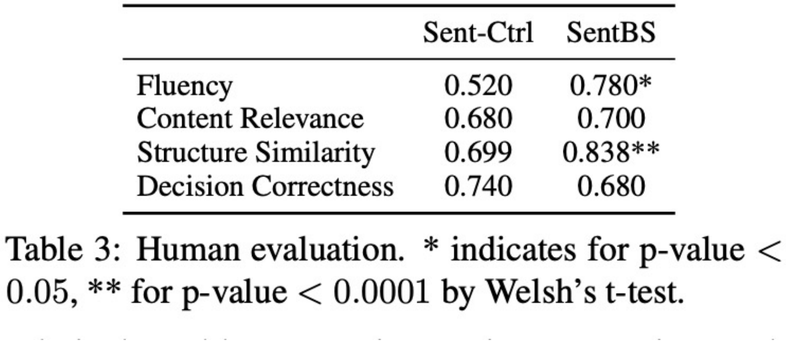
Table 1: Main results. We divide the table into 2 sections using double horizontal lines. The top section shows the Sent-Ctrl baseline and Sent-Ctrl with various settings of SentBS, and the bottom section shows the Seg-Ctrl baseline and Seg-Ctrl with SentBS. For the latter section, we use the “Beam search + Beam sampling + Nucleus sampling” setting and $k=8$ for SentBS.

3. Experiments



Beam Size	Structure ↑	Edits ↓	BERTScore ↑	R_1 / R_2 / R_L ↑
4	**0.737**	1470	**0.8631**	**38.97 / 11.19 / 23.48**
5	0.732	**1424**	0.8625	38.81 / 10.92 / 23.34
6	0.723	1456	0.8618	38.67 / 10.85 / 23.19
7	0.727	1427	0.8613	38.55 / 10.73 / 23.00
8	0.722	1448	0.8608	38.29 / 10.68 / 22.90

Table 2: Beam Search generation results on MReD using Sent-Ctrl with increasing beam sizes.



	Sent-Ctrl	SentBS
Fluency	0.520	0.780*
Content Relevance	0.680	0.700
Structure Similarity	0.699	0.838**
Decision Correctness	0.740	0.680

Table 3: Human evaluation. * indicates for p-value $<$ 0.05, ** for p-value $<$ 0.0001 by Welsh's t-test.

- Beam Search generation results on MReD using Sent-Ctrl with increasing beam sizes → 4에서 가장 좋음
- Human evaluation results

Conclusion

1. SALTED: A Framework for SAlient Long-tail Translation Error Detection

- Oral presentation
- Keywords: neural machine translation, quality estimation, error detection

→ long tail problem을 다루며, 실제 field에도 적용될 여지가 높음. 반드시 다루어야 할 critical errors를 이 논문에서 다룸

2. CTRLsum: Towards Generic Controllable Text Summarization

- Poster presentation
- Keywords: summarization, keyword control, length control

→ summarization의 practical application을 위한 논문, user의 preference를 고려

3. SentBS: Sentence-level Beam Search for Controllable Summarization

- Short paper
- Keywords: summarization, decoding strategy

→ 어떤 seq-to-seq 모델에도 추가로 활용할 수 있음. Retraining 없이도 성능을 significantly향상

Thank you!

Q&A