An Empirical Study on Extractive Summarization

冯家展

Sept. 2nd, 2019

Searching for Effective Neural Extractive Summarization: What Works and What's Next

Ming Zhong[*], **Pengfei Liu**[*], **Danqing Wang**, **Xipeng Qiu**[†], **Xuanjing Huang**
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{mzhong18,pfliu14,dqwang18,xpqiu,xjhuang}@fudan.edu.cn

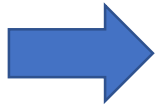
Motivation

However, there is no clear understanding of *why* they perform so well, or *how* they might be improved. (they == existing methods)

1. Analyzed different types of model architectures, transferable knowledge and learning schemas.
2. Got "SOTA" result on CNN/DailyMail (improved with a large margin)

Existing Framework

Extractive Summarization:

- Document $D = d_1, \cdots, d_n$  $R = r_1, \cdots, r_m$
- sentence encoder + document encoder + decoder

Training Environment

- Architectures:
  - Sentence Encoders: CNN
  - Document Encoders: LSTM, Transformer
  - Decoders: auto-regressive* (Pointer), non auto-regressive (SeqLab)
- External Transferable Knowledge:
  - Glove
  - BERT
  - Newsroom (Grusky et al., 2018)
- Learning Schemas:
  - Supervised learning
  - Reinforcement learning

*Auto-regressive indicates that the decoder can make current prediction **with knowledge of previous predictions.**

Datasets

Domains	Train	Valid	Test
CNN/DailyMail	287,227	13,368	11,490
NYTimes	152,981	16,490	16,624
WashingtonPost	96,775	10,103	10,196
FoxNews	78,795	8,428	8,397
TheGuardian	58,057	6,376	6,273
NYDailyNews	55,653	6,057	5,904
WSJ	49,968	5,449	5,462
USAToday	44,921	4,628	4,781

Table 2: Statistics of multi-domain datasets based on CNN/DailyMail and NEWSROOM.

Testing Environment

- Rouge: Rouge-1 Rouge-2 Rouge-L F_1 scores
- Cross-domain Evaluation (based on CNN/DM, Newsroom)
- Repetition (Diversity)

$$\mathrm{REP_n} = \frac{\mathrm{CountUniq}(ngram)}{\mathrm{Count}(ngram)}$$

- Positional Bias (Study on ground-truth distribution, effect on Arch.)
- Sentence Length (of summarized sentence)
- Sentence Shuffling (articles in training set) -> robustness

Analysis of Decoder

Pointer vs SeqLab

Analysis of Decoder

1. Domains:

- Models with Pointer-based decoder > SeqLab-based decoder
  - in six domain, and two comparable
  - especially in NYTimes, WashingtonPost, TheGuardian(>1.0 in R-1)

Model		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Dec.	**Enc.**	**CNN/DM (2/3)**			**NYTimes (2)**			**WashingtonPost (1)**			**Foxnews (1)**		
	Lead	40.11	17.64	36.32	28.75	16.10	25.16	22.21	11.40	19.41	54.20	46.60	51.89
	Oracle	55.24	31.14	50.96	52.17	36.10	47.68	42.91	27.11	39.42	73.54	65.50	71.46
SeqLab	LSTM	41.22	18.72	37.52	30.26	17.18	26.58	21.27	10.78	18.56	59.32	51.82	56.95
	Transformer	41.31	**18.85**	37.63	30.03	17.01	26.37	21.74	10.92	18.92	59.35	51.82	56.97
Pointer	LSTM	**41.56**	18.77	**37.83**	31.31	**17.28**	**27.23**	**24.16**	**11.84**	**20.67**	**59.53**	**51.89**	**57.08**
	Transformer	41.36	18.59	37.67	**31.34**	17.25	27.16	23.77	11.63	20.48	59.35	51.68	56.90
Dec.	**Enc.**	**TheGuardian (1)**			**NYDailyNews (1)**			**WSJ (1)**			**USAToday (1)**		
	Lead	22.51	7.69	17.78	45.26	35.53	42.70	39.63	27.72	36.10	29.44	18.92	26.65
	Oracle	41.08	21.49	35.80	73.99	64.80	72.09	57.15	43.06	53.27	47.17	33.40	44.02
SeqLab	LSTM	23.02	8.12	18.29	53.13	43.52	50.53	41.94	29.54	38.19	30.30	18.96	27.40
	Transformer	23.49	8.43	18.65	53.66	44.19	51.07	42.98	**30.22**	39.02	30.97	19.77	28.03
Pointer	LSTM	24.71	8.55	19.30	53.31	43.37	50.52	43.29	30.20	**39.12**	31.73	19.89	28.50
	Transformer	**24.86**	**8.66**	**19.45**	**54.30**	**44.70**	**51.67**	**43.30**	30.17	39.07	**31.95**	**20.11**	**28.78**

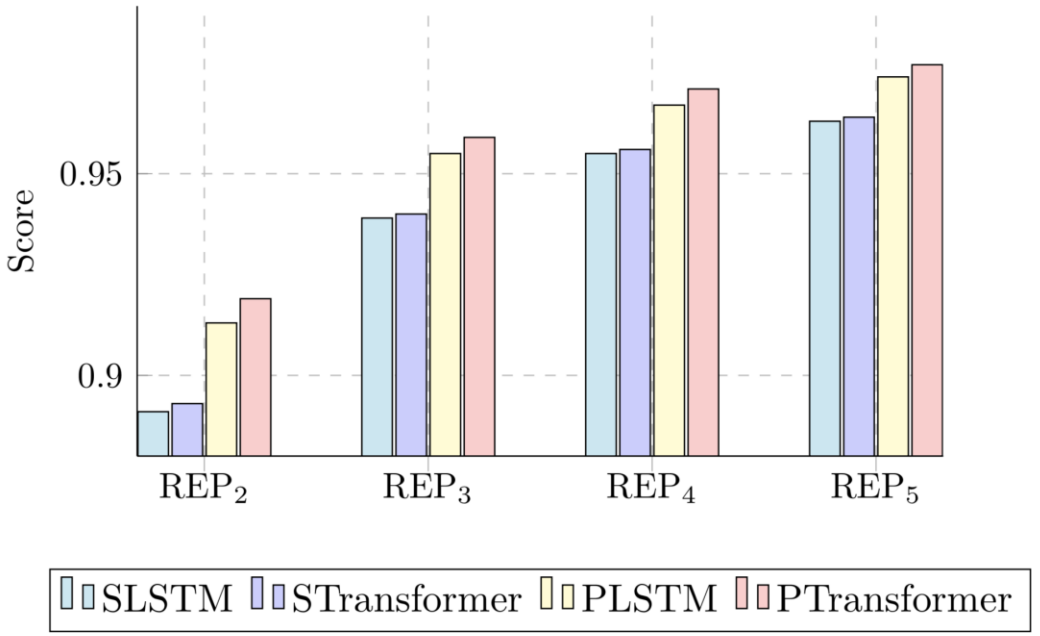
Table 3: Results of different architectures over different domains, where **Enc.** and **Dec.** represent document encoder and decoder respectively. Lead means to extract the first k sentences as the summary, usually as a competitive lower bound. Oracle represents the ground truth extracted by the greedy algorithm (Nallapati et al., 2017), usually as the upper bound. The number k in parentheses denotes k sentences are extracted during testing and choose lead-k as a lower bound for this domain. All the experiments use word2vec to obtain word representations.

Analysis of Decoder

2. Repetition: (Extractive multi-sent.):

- Pointer > SeqLab

$$\mathrm{REP_n} = \frac{\mathrm{CountUniq}(ngram)}{\mathrm{Count}(ngram)}$$



(a) Repetition score

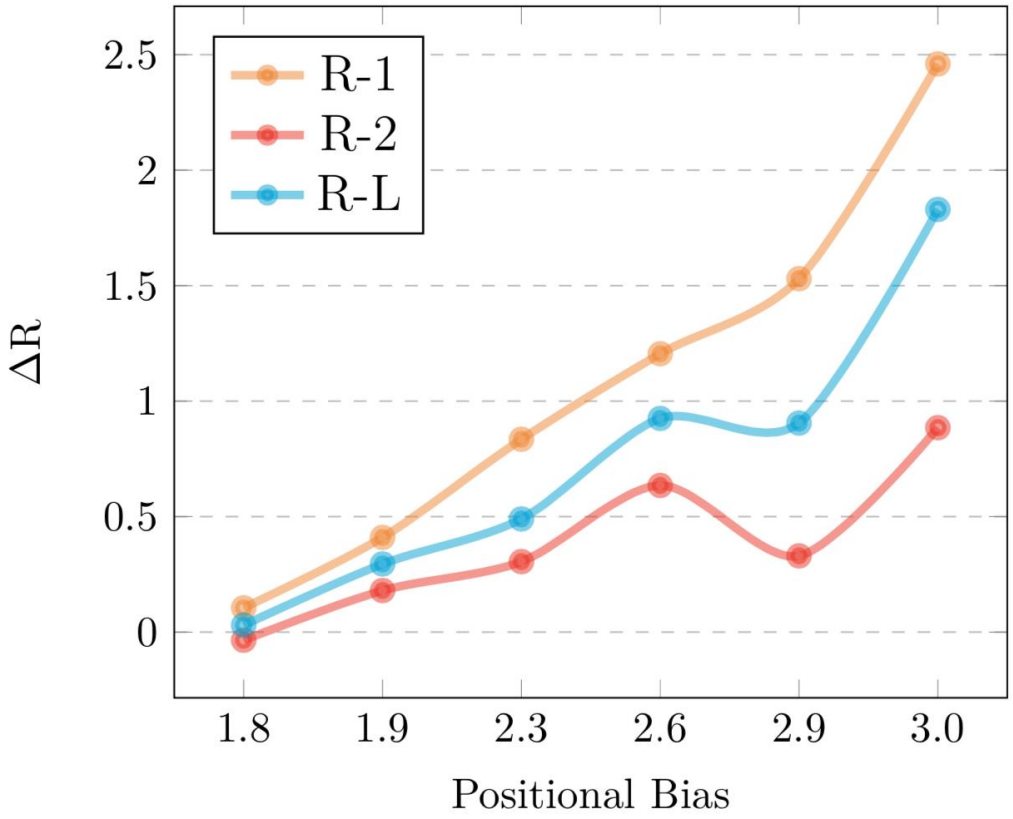
- Indicates that Pointer does capture word-level information from previous selected sentences and has positive effects on subsequent decisions.

Analysis of Decoder

3. Positional Bias (Extractive 1 sent.):

- Gap between Pointer and SeqLab

- Pointer > SeqLab



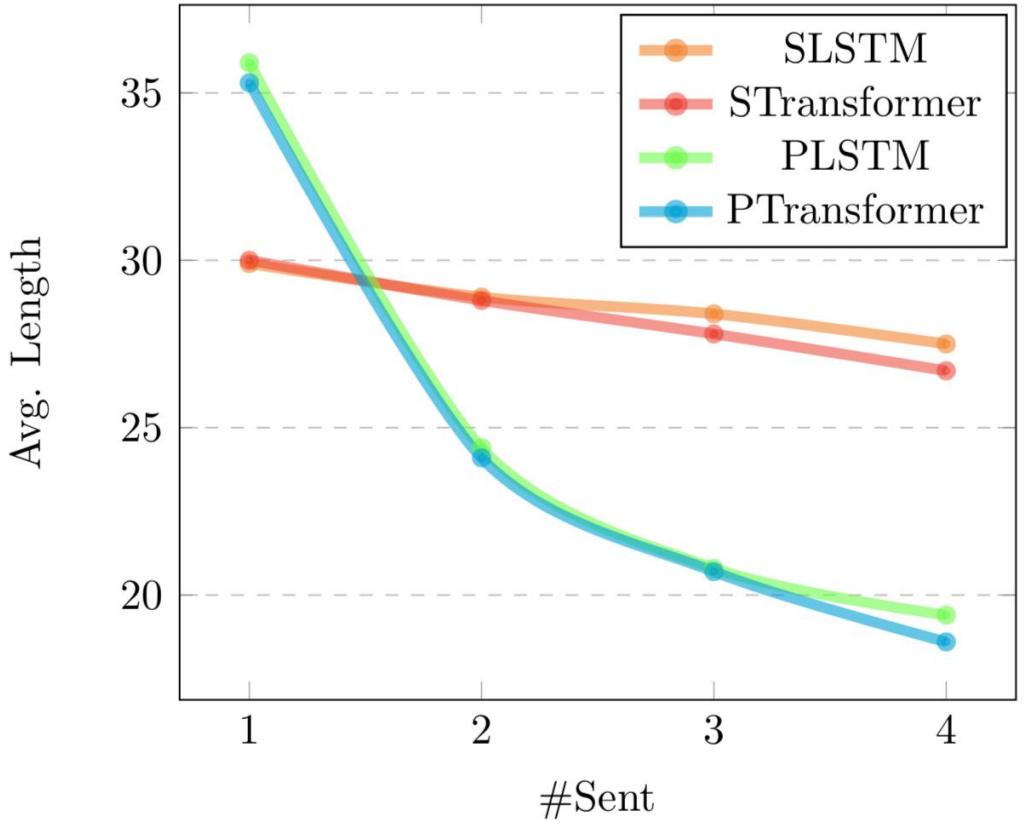
(b) Positional bias

- Indicates that SeqLab is more sensitive to positional bias, which impairs its performance on some datasets.

Analysis of Decoder

4. Sentence Length:

- Models with Pointer
  - choose longer sentences as the first sentence
  - greatly reduce the length of the sentence in the subsequent extractions



(c) Average length

- Indicates that Pointer can adaptively change the length of the extracted sentences, thereby achieving better performance regardless of whether one sentence or multiple sentences are required.

Analysis of Encoder

LSTM vs Transformer

Analysis of Encoder

1. Domains:
   - Transformer can outperform LSTM on some datasets “NYDailyNews” by a relatively large margin while LSTM beats Transformer on some domains with closed improvements.

   - Above phenomena suggest that LSTM easily suffers from the architecture overfitting problem compared with Transformer.

   - When equipped with SeqLab decoder, Transformer always obtains a better performance compared with LSTM (non-local bias (Wang et al., 2018) of Transformer).

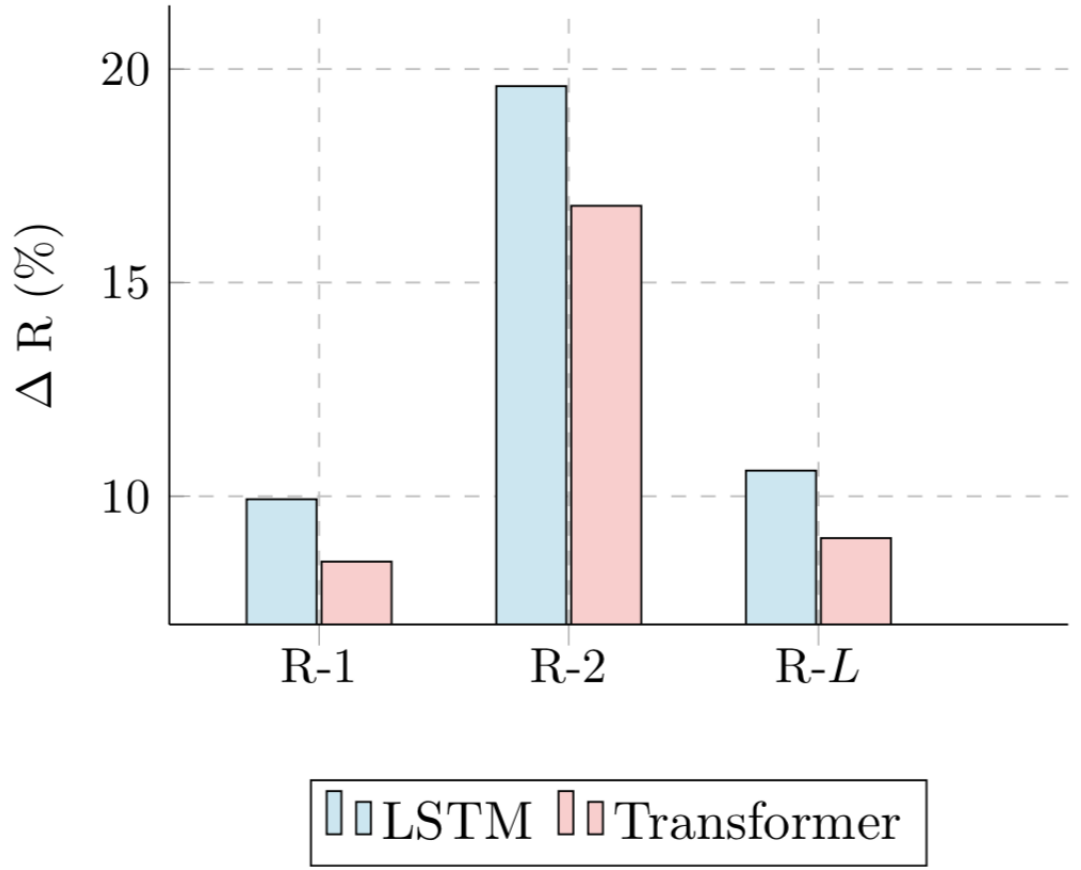
Model		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Dec.	**Enc.**	**CNN/DM (2/3)**			**NYTimes (2)**			**WashingtonPost (1)**			**Foxnews (1)**		
	Lead	40.11	17.64	36.32	28.75	16.10	25.16	22.21	11.40	19.41	54.20	46.60	51.89
	Oracle	55.24	31.14	50.96	52.17	36.10	47.68	42.91	27.11	39.42	73.54	65.50	71.46
SeqLab	LSTM	41.22	18.72	37.52	30.26	17.18	26.58	21.27	10.78	18.56	59.32	51.82	56.95
	Transformer	41.31	**18.85**	37.63	30.03	17.01	26.37	21.74	10.92	18.92	59.35	51.82	56.97
Pointer	LSTM	**41.56**	18.77	**37.83**	31.31	**17.28**	**27.23**	**24.16**	**11.84**	**20.67**	**59.53**	**51.89**	**57.08**
	Transformer	41.36	18.59	37.67	**31.34**	17.25	27.16	23.77	11.63	20.48	59.35	51.68	56.90
Dec.	**Enc.**	**TheGuardian (1)**			**NYDailyNews (1)**			**WSJ (1)**			**USAToday (1)**		
	Lead	22.51	7.69	17.78	45.26	35.53	42.70	39.63	27.72	36.10	29.44	18.92	26.65
	Oracle	41.08	21.49	35.80	73.99	64.80	72.09	57.15	43.06	53.27	47.17	33.40	44.02
SeqLab	LSTM	23.02	8.12	18.29	53.13	43.52	50.53	41.94	29.54	38.19	30.30	18.96	27.40
	Transformer	23.49	8.43	18.65	53.66	44.19	51.07	42.98	**30.22**	39.02	30.97	19.77	28.03
Pointer	LSTM	24.71	8.55	19.30	53.31	43.37	50.52	43.29	30.20	**39.12**	31.73	19.89	28.50
	Transformer	**24.86**	**8.66**	**19.45**	**54.30**	**44.70**	**51.67**	**43.30**	30.17	39.07	**31.95**	**20.11**	**28.78**

Table 3: Results of different architectures over different domains, where **Enc.** and **Dec.** represent document encoder and decoder respectively. Lead means to extract the first k sentences as the summary, usually as a competitive lower bound. Oracle represents the ground truth extracted by the greedy algorithm (Nallapati et al., 2017), usually as the upper bound. The number k in parentheses denotes k sentences are extracted during testing and choose lead-k as a lower bound for this domain. All the experiments use word2vec to obtain word representations.

Analysis of Encoder

2. Shuffled Testing:

- Only in training set
- Drop: LSTM > Transformer



- Transformer obtains lower decrease against LSTM.
- Indicates that Transformer are more robust.

Analysis of Encoder

2. Disentangling Testing:

- Investigating what role positional information plays.
- Only utilizing positional embedding (the model is only told how many sentences the document contains), our model can achieve 40.08 on R-1, which is comparable to many existing models.
- Once the positional information is removed, the performance dropped by a large margin.
- Success of such extractive summarization heavily relies on the ability of learning the positional information on CNN/DailyMail

Analysis of Transferable Knowledge

Context-independent (Glove, Newsroom)
vs
Context-dependent (BERT)

Analysis of Transferable Knowledge

1. Unsupervised Learning (baseline: word2vec):

- Context-independent word representation contributes less
- BERT (improved by a large margin, new SOTA scores 42.11 in R-1 in CNN-LSTM-Pointer)

Model		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Dec.	**Enc.**	**Baseline**			**+ GloVe**			**+ BERT**			**+ NEWSROOM**		
SeqLab	LSTM	41.22	18.72	37.52	**41.33**	**18.78**	**37.64**	42.18	19.64	38.53	41.48	**18.95**	37.78
	Transformer	41.31	**18.85**	37.63	40.19	18.67	37.51	42.28	**19.73**	38.59	41.32	18.83	37.63
Pointer	LSTM	**41.56**	18.77	**37.83**	41.15	18.38	37.43	**42.39**	19.51	**38.69**	41.35	18.59	37.61
	Transformer	41.36	18.59	37.67	41.10	18.38	37.41	42.09	19.31	38.41	**41.54**	18.73	**37.83**

Analysis of Transferable Knowledge

2. Supervised Learning:

- Pre-trained from Newsroom dataset
- In most cases, the performance increases.
- CNN-LSTM-Pointer fails and performance decreases. (Domain shift problem)

Analysis of Learning Schema

Supervised learning
vs
Reinforcement learning

Analysis of Learning Schema

Models	R-1	R-2	R-L
Chen and Bansal (2018)	41.47	18.72	37.76
Dong et al. (2018)	41.50	18.70	37.60
Zhou et al. (2018)	41.59	19.01	37.98
Jadhav and Rajan (2018)[8]	41.60	18.30	37.70
LSTM + PN	41.56	18.77	37.83
LSTM + PN + RL	41.85	18.93	38.13
LSTM + PN + BERT	42.39	19.51	38.69
LSTM + PN + BERT + RL	**42.69**	**19.60**	**38.85**

Table 6: Evaluation on CNN/DailyMail. The top half of the table is currently state-of-the-art models, and the lower half is our models.

More constraints??

Improved??

Only one experiment??

Conclusions

1. Auto-regressive decoder(Pointer) > non auto-regressive decoder (SeqLab)

2. LSTM tends to suffer from architecture overfitting while Transformer is more robust

3. CNN/DailyMail corpus heavily relies on the ability to learn positional information of the sentence

4. Unsupervised transferable knowledge > supervised one (domain shift problem)

5. New SOTA 42.39 R-1 score in CNN/DM and 42.69 with RL

Thanks!