Long Story Short: Using BERT for abstractive text summarization on a small, curated corpus

Limor Gultchin, ATI DL group, Jan 21, 2020

## Problem definition

- Summer internship project for industry research team
- Clients have many PDF documents. Many of them contain tables. While software can extract tables from documents, devoid of context they make far less sense

Concentrators	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Computer Science	86	86	99	143	198	253	263	306	363	394
Computer Science + another field	4	7	10	13	17	22	32	42	47	59
Another field + Computer Science	4	8	10	15	7	18	21	24	25	41

# Solution: table + page titles

Harvard College Handbook for Students *Handbook for Students 2019-2020* 

### Computer Science

### **ENROLLMENT STATISTICS**

**Number of Concentrators as of December** 

Concentrators	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Computer Science	86	86	99	143	198	253	263	306	363	394
Computer Science + another field	4	7	10	13	17	22	32	42	47	59
Another field + Computer Science	4	8	10	15	7	18	21	24	25	41

# Scientific papers as table-title-generation prototype

Table 8.1: The scalability parameters (using SimpleNews as an example)

	SimpleNews	Your APP
#features	3	
$\# { t actions}$	4 news topics	
Typical delay	5 sec	
Data rate	100 users/sec	
Rare event frequency	typical click prob. 2-5%	
Stationarity timescale	one week	

Scale and feasibility. To estimate the scale and feasibility of the learning problem in APP, one needs to estimate a few parameters: the number of features (#features), the number of feasible actions (#actions), a typical delay between making a decision and observing the corresponding outcome, and the data rate: the number of experimental units per a time unit. If the outcome includes a rare event whose frequency is crucial to estimate — e.g., clicks are typically rare compared to non-clicks, — then we also need a rough estimate of this frequency. Finally, we need the stationarity timescale: the time interval during which the environment does not change too much, namely the distribution of arriving contexts (where a context includes the set of feasible actions and, if applicable, their features), and the expected rewards for each context-action pair. In the SimpleNews example, this corresponds to the distribution of arriving user profiles and the click probability for the top headline (for a given news topic, when presented to a typical user with a given user profile). To summarize, one should be able to fill in Table 8.1.

## Titles are conceptually similar to summarization

- Text summarization is an area of much interest within the study of NLP
- Text summarization falls into two groups:

Extractive summarization

Abstractive summarization





## NLP approaches to summarization

Area of much research, usually tested on corpuses of short news articles with labeled three-sentence summaries.

- Extractive: Text-rank, Latent Semantic Analysis (LSA w\ SVD decomposition), Bayesian approaches, etc... Most based on frequency of appearance of word in sentence/document level.
- Abstractive: Seq-to-Seq, Pointer-Generator networks, Transformers (can be used for extractive as well sometimes, depends on optimization objective)

# Mixture of patterns and topics in our dataset

1.14 2.41

155 326

1.15 2.42

010 116

0.35 1.32

0.07 0.98

0.22 1.20

114

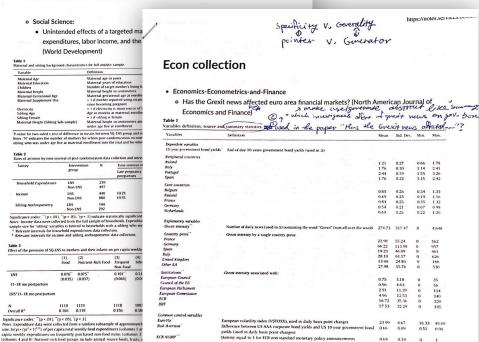
320

#### Social Science collection

mother-infant pair was randomized to receive SO-LNS and zero if the mother receive

mo postpartum' is an indicator variable for the interval (relative to the birth of it

postpartum'. Controls for interval of data collection, enumerator, year of maternal i each model (unreported). Standard errors (in parentheses) are clustered at the ho



Notes: Summary statistics are calculated for variables expressed in levels over the whole period. Information for additional country-specific control

variables (i.e. W1) are not reported in the table for the sake of space. Sources: Elaborations on Datastream; "Factiva;" Falaguarda et al. (2015),

Pattern 1:	lable	x pre	sents/snow	's/expiaii	ns_
	Table name		Ve	rb	Des
Pattern 2:	As we	show Verb	r in Table Y, Table name	Desc.	
Pattern 3:		see 7	Table Z		
••••	Desc.	Verb	Table name		

### Generating Titles for Web Tables (Hancock et al. 2019)



```
Flattened Input:
```

```
#page_title__nicole___eggert__ - wikipedia
#section_headings __nicole___eggert__
#section_headings filmography [ edit ]
#section_headings television [ edit ]
#column_headers year
#column_headers title
#column_headers role
#column_headers notes
```

### **Generated Output:**

nicole eggert television roles

Massive dataset: original 14.1 tables scraped from google (e.g. Wikipedia) + user study 10K tables, trained from scratch.

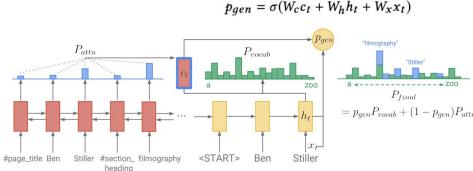
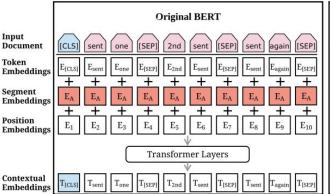


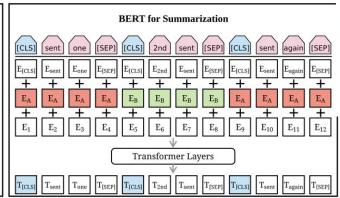
Figure 6: At each decode step, the pointer-generator network calculates the scalar value  $p_{qen}$ , which determines the relative contributions of the vocabulary distribution (tokens that can be generated) and the attention distribution (tokens that can be copied from the input sequence). Beam search is performed over the final distribution.

## BERT for text summarization (Liu et al. 2019)

### **Text Summarization with Pre-trained Encoders**

Yang Liu. Mirella Lapata (EMNLP 2019)





This code is for EMNLP 2019 paper Text Summarization with Pretrained Encoders

Results on CNN/DailyMail (20/8/2019):

ROUGE-1	ROUGE-2	ROUGE-L
40.90	18.02	37.17
43.23	20.24	39.63
43.85	20.34	39.90
40.21	17.76	37.09
41.72	19.39	38.76
42.13	19.60	39.18
	40.90 43.23 43.85 40.21 41.72	40.90 18.02 43.23 20.24 43.85 20.34 40.21 17.76 41.72 19.39

Figure 1: Architecture of the original BERT model (left) and BERTSUM (right). The sequence on top is the input document, followed by the summation of three kinds of embeddings for each token. The summed vectors are used as input embeddings to several bidirectional Transformer layers, generating contextual vectors for each token. BERTSUM extends BERT by inserting multiple [CLS] symbols to learn sentence representations and using interval segmentation embeddings (illustrated in red and green color) to distinguish multiple sentences.

## BERT vs. other pre-trainings

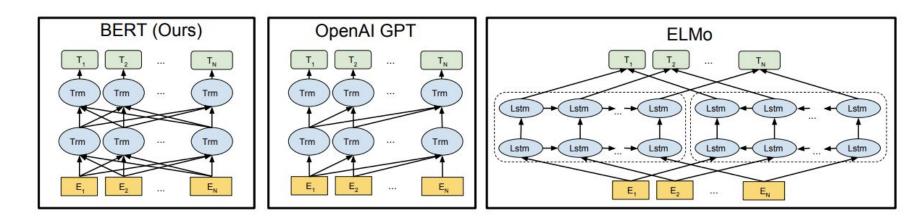


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

# BERT fine-tuning original tasks

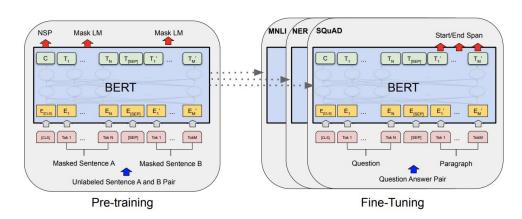
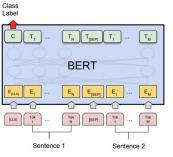
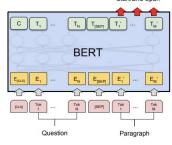


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

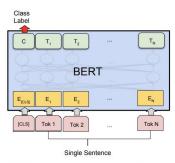


(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE. SWAG

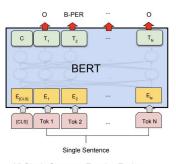
Start/End Span



(c) Question Answering Tasks: SQuAD v1.1



(b) Single Sentence Classification Tasks: SST-2. CoLA



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

### Liu et al., BERT fine-tuning for text summaries





State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

### **Extractive**

$$ilde{h}^l = ext{LN}(h^{l-1} + ext{MHAtt}(h^{l-1}))$$
 $h^l = ext{LN}( ilde{h}^l + ext{FFN}( ilde{h}^l))$ 
where
 $h^0 = ext{PosEmb}(T)$ 
output
 $\hat{y}_i = \sigma(W_o h_i^L + b_o)$ 
loss

 $L_i = H_b(\hat{y}_i)$   $H_p(q) = -\sum_{c} q(y_c) \cdot log(p(y_c))$ 

Decoder:
Intersentence
transformer
layers for
document
level
features

### **Abstractive**

Continue training from Extractive

Decoder: 6-layered Transformer initialized randomly+label smoothing+ Allow higher slack in KL divergence

Optimize encoder/decoder separately for Stability

Subword tokenizer handles OOV; Block trigram repetitions (no copy or coverage)

### What usual text summarization datasets look like

#### STORY HIGHLIGHTS

Trump will head to Texas on Tuesday

The White House has yet to say where Trump will travel **Washington (CNN)** — President Donald Trump struck a unifying tone Monday as he addressed the devastation in Texas wrought by Hurricane Harvey at the top of a joint news conference with Finland's president.

"We see neighbor helping neighbor, friend helping friend

and stranger helping stranger," Trump said. "We are one American family. We hurt together, we struggle together and believe me, we endure together."

Trump extended his "thoughts and prayers" to those affected by the hurricane and catastrophic flooding that ensued in Texas, and also promised Louisiana residents that the federal government is prepared to help as the tropical storm makes its way toward that state.

"To the people of Texas and Louisiana, we are 100% with you," Trump said from the East Room of the White House.

### What our dataset looked like

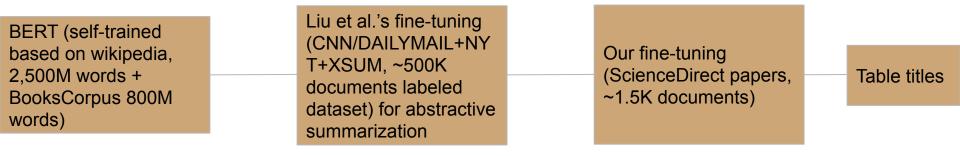
Model	R1	R2	RL
ORACLE	52.59	31.24	48.87
LEAD-3	40.42	17.62	36.67
Extractive			
SUMMARUNNER (Nallapati et al., 2017)	39.60	16.20	35.30
REFRESH (Narayan et al., 2018b)	40.00	18.20	36.60
LATENT (Zhang et al., 2018)	41.05	18.77	37.54
NEUSUM (Zhou et al., 2018)	41.59	19.01	37.98
SUMO (Liu et al., 2019)	41.00	18.40	37.20
TransformerEXT	40.90	18.02	37.17
Abstractive	*		
PTGEN (See et al., 2017)	36.44	15.66	33.42
PTGEN+Cov (See et al., 2017)	39.53	17.28	36.38
DRM (Paulus et al., 2018)	39.87	15.82	36.90
BOTTOMUP (Gehrmann et al., 2018)	41.22	18.68	38.34
DCA (Celikyilmaz et al., 2018)	41.69	19.47	37.92
TransformerABS	40.21	17.76	37.09
BERT-based			
BERTSUMEXT	43.25	20.24	39.63
BERTSUMEXT w/o interval embeddings	43.20	20.22	39.59
BERTSUMEXT (large)	43.85	20.34	39.90
BERTSUMABS	41.72	19.39	38.76
BERTSUMEXTABS	42.13	19.60	39.18

Table 2: ROUGE F1 results on CNN/DailyMail test set (R1 and R2 are shorthands for unigram and bigram overlap; RL is the longest common subsequence). Results for comparison systems are taken from the authors' respective papers or obtained on our data by running publicly released software.

Table 2 summarizes our results on the CNN/DailyMail dataset. The first block in the table includes the results of an extractive ORACLE system as an upper bound. We also present the LEAD-3 baseline (which simply selects the first three sentences in a document).

# BERT transfer learning to the rescue

With a little bit of trial (and error) it turns out continuing to train Liu et al.'s model with our dataset of 1500k was surprisingly successful!



## Experience and Insights

- Deep nets are excellent at dealing with noise. Do not spend too much time cleaning your data! Try it first.
- A small dataset can be used to fine tune a model that was first trained on a big, established corpus. It might perform better than you think.
- Be careful not to overtrain your dataset (too many steps, etc.) when moving from bigger dataset to smaller one. Early stopping is important.
- Summarization metrics (ROUGE, BLEU) are not as helpful for abstractive summaries, but can still be useful.