



From Fundamentals to Recent Advances A Tutorial on Keyphrasification

Part 3.3 Learning Better Keyphrase Representations

Rui Meng, Debanjan Mahata, Florian Boudin

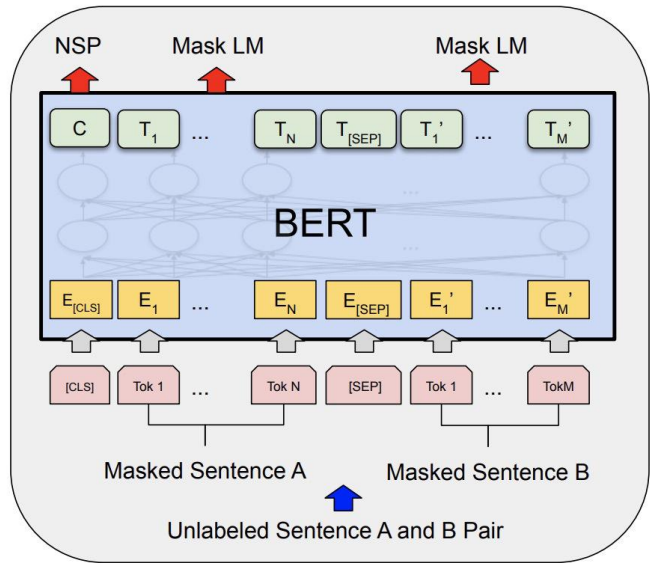
ECIR 2022



MOODY'S
ANALYTICS



Language Models

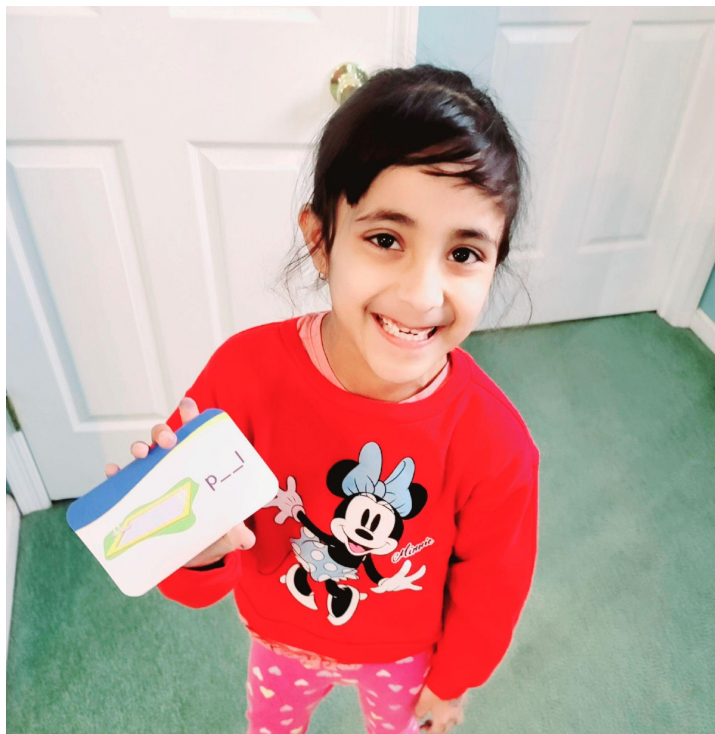


Why Language Models

Why does language modelling work so well?

The remarkable success of pretrained language models is surprising. One reason for the success of language modelling may be that it is a very difficult task, even for humans. To have any chance at solving this task, a model is required to learn about syntax, semantics, as well as certain facts about the world. Given enough data, a large number of parameters, and enough compute, a model can do a reasonable job. Empirically, language modelling works better than other pretraining tasks such as translation or autoencoding (Zhang et al. 2018; Wang et al., 2019).

Why not ... :)



Pre-trained Language Models

- ❖ Pre-training on a huge corpus can learn universal language representations and help with several downstream NLP tasks.
- ❖ Pre-trained language models provides better model initialization, leading to better generalization performance and speeds up convergence on target task.
- ❖ Pre-training can be regarded as regularization that helps in avoiding overfitting on small datasets.

Keyphrase Language Model


- ❖ Can we formulate a pre-training objective for language models that can learn better representation of keyphrases?
- ❖ Does learning rich representation of keyphrases in a language model lead to performance gains for the tasks of keyphrase extraction and generation?
- ❖ Do rich keyphrase representations aid other fundamental tasks in NLP such as NER, QA, RE and summarization?

Keyphrase Language Models


❖ KBIR - **K**eyphrase **B**oundary **I**nfling **T**ask

❖ KeyBART

```
from transformers import AutoModel
model = AutoModel.from_pretrained("bloomberg/KeyBART")
```

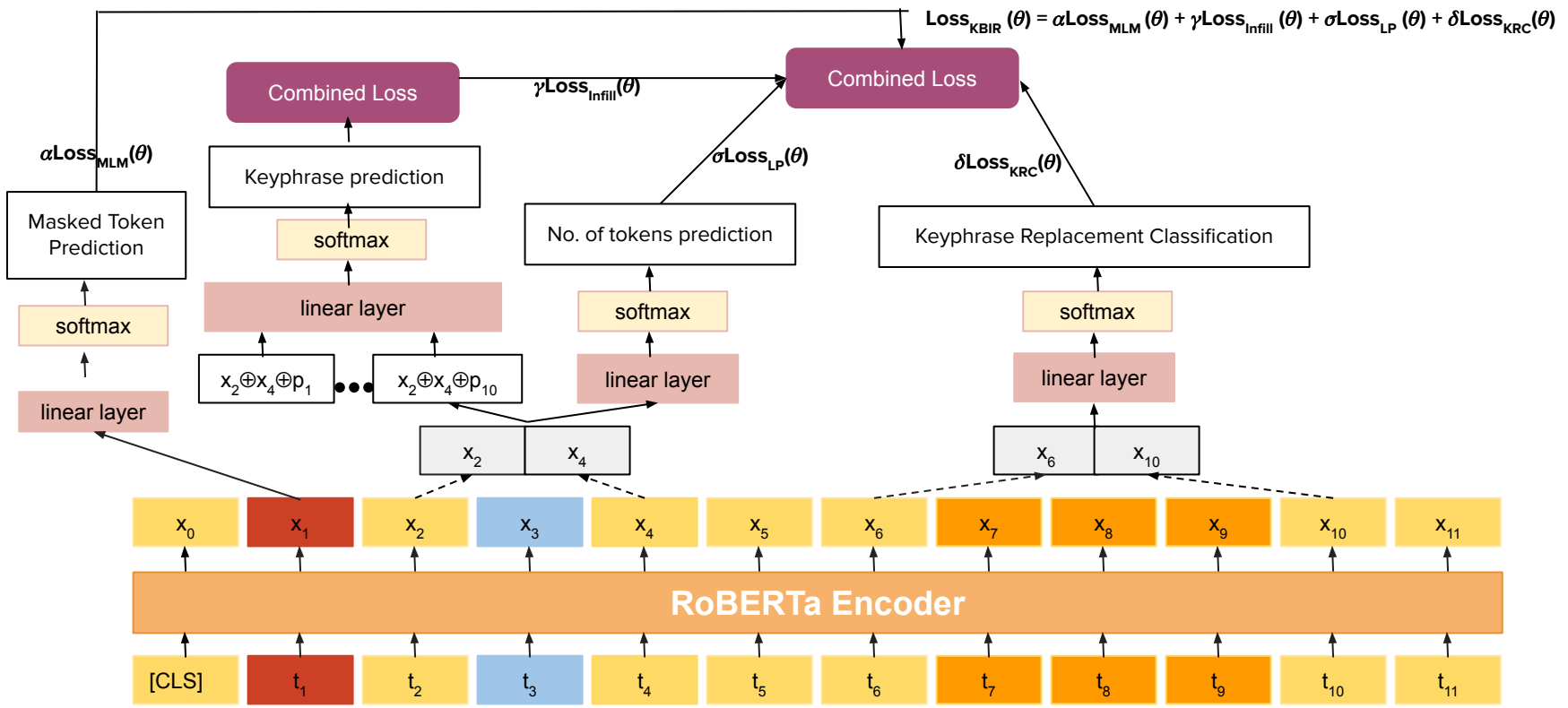


```
from transformers import AutoModel
model = AutoModel.from_pretrained("bloomberg/KBIR")
```



[Kulkarni, M., Mahata, D., Arora, R., & Bhowmik, R. \(2021\). Learning Rich Representation of Keyphrases from Text. Accepted at NAACL-HLT 2022 \(Findings\).](#)

Keyphrase Boundary Infilling and Replacement - KBIR



Pre-training Objectives - KBIR

- ❖ Masked Language Modeling (**MLM**) - token masking
- ❖ Keyphrase Boundary Infling (**KBI**) - keyphrase masking
- ❖ Keyphrase Replacement Classification (**KRC**) - contrastive learning

KBIR = MLM + KBI + KRC

RoBERTa

Masking Strategies

Input Text

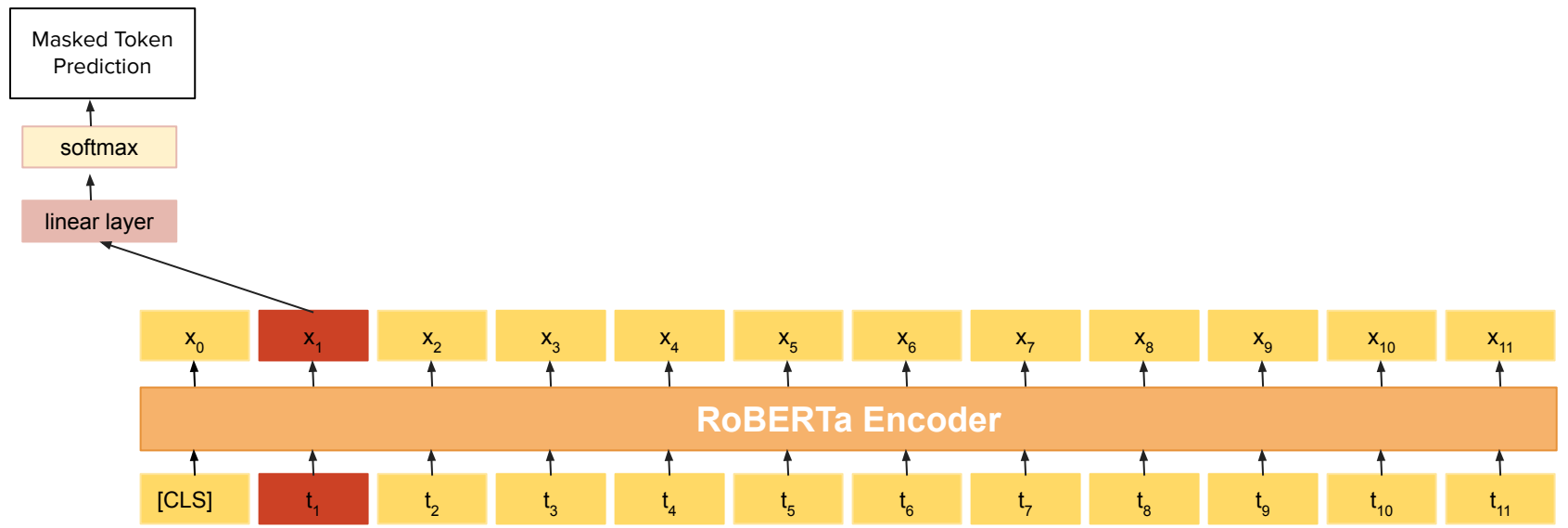
Keyphrases are an important means of document summarization, clustering, and topic search. Only a small minority of documents have author-assigned keyphrases, and manually assigning keyphrases to existing documents is very laborious. Therefore it is highly desirable to automate the keyphrase extraction process. This paper shows that a simple procedure for keyphrase extraction based on the naive Bayes learning scheme performs comparably to the state of the art. It goes on to explain how this procedure's performance can be boosted by automatically tailoring the extraction process to the particular document collection at hand. Results on a large collection of technical reports in computer science show that the quality of the extracted keyphrases improves significantly when domain-specific information is exploited.

Masking Strategies

Token Masking

Keyphrases are an important means of document summarization, clustering, and topic search. Only a small minority of documents have author-assigned keyphrases, and manually assigning keyphrases to existing documents is **[MASK]** laborious. Therefore it is highly desirable to automate the keyphrase extraction process. This paper shows that a simple procedure for keyphrase extraction based on the naive Bayes learning scheme performs comparably to the state of the art. It goes on to explain how this procedure's performance can be **[MASK]** by automatically tailoring the extraction process to the particular document collection at hand. Results on a large collection of technical reports in computer science show that the quality of the extracted keyphrases improves significantly when domain-specific information **[MASK]** exploited.

Token Masking



■ ordinary token for masking

t_i - i^{th} token, x_i - i^{th} token embedding, p_i - i^{th} position embedding

Masking Strategies

Keyphrase Masking

Keyphrases are an important means of [MASK], clustering, and topic search. Only a small minority of documents have author-assigned keyphrases, and manually assigning keyphrases to existing documents is [MASK] laborious. Therefore it is highly desirable to automate the [MASK]. This paper shows that a simple procedure for keyphrase extraction based on the [MASK] learning scheme performs comparably to the state of the art. It goes on to explain how this procedure's performance can be [MASK] by automatically tailoring the extraction process to the particular document collection at hand. Results on a large collection of technical reports in computer science show that the quality of the extracted keyphrases improves significantly when domain-specific information [MASK] exploited.

Learning Span Representations

$$\begin{aligned} \mathcal{L}(\text{football}) &= \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football}) \\ &= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3) \end{aligned}$$

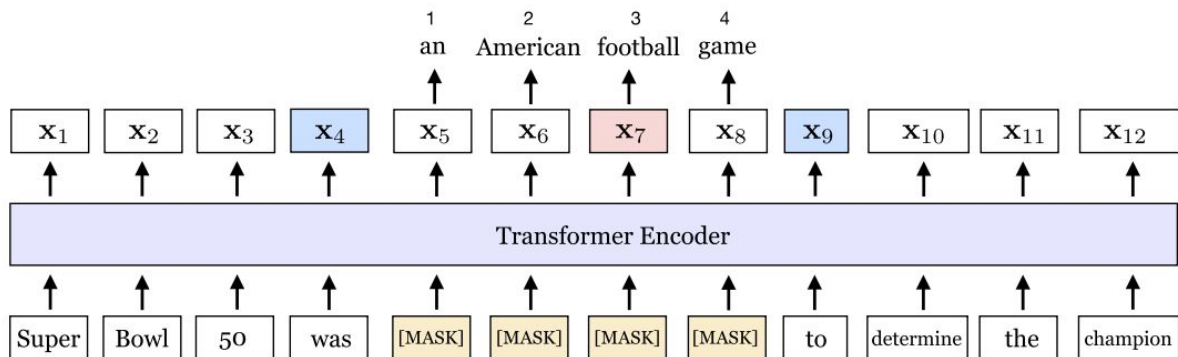
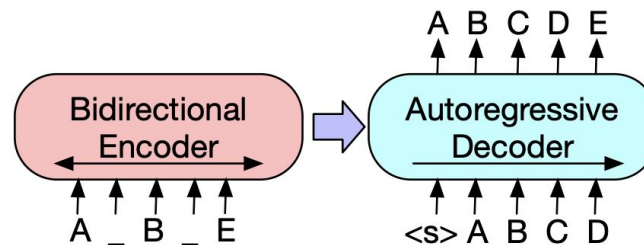
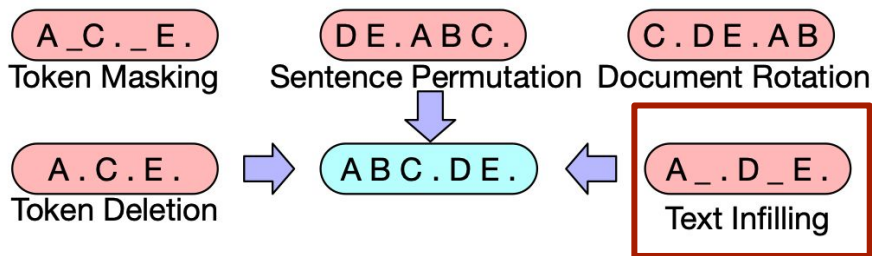


Figure 1: An illustration of SpanBERT training. The span *an American football game* is masked. The SBO uses the output representations of the boundary tokens, x_4 and x_9 (in blue), to predict each token in the masked span. The equation shows the MLM and SBO loss terms for predicting the token, *football* (in pink), which as marked by the position embedding p_3 , is the *third* token from x_4 .

Learning Span Representations

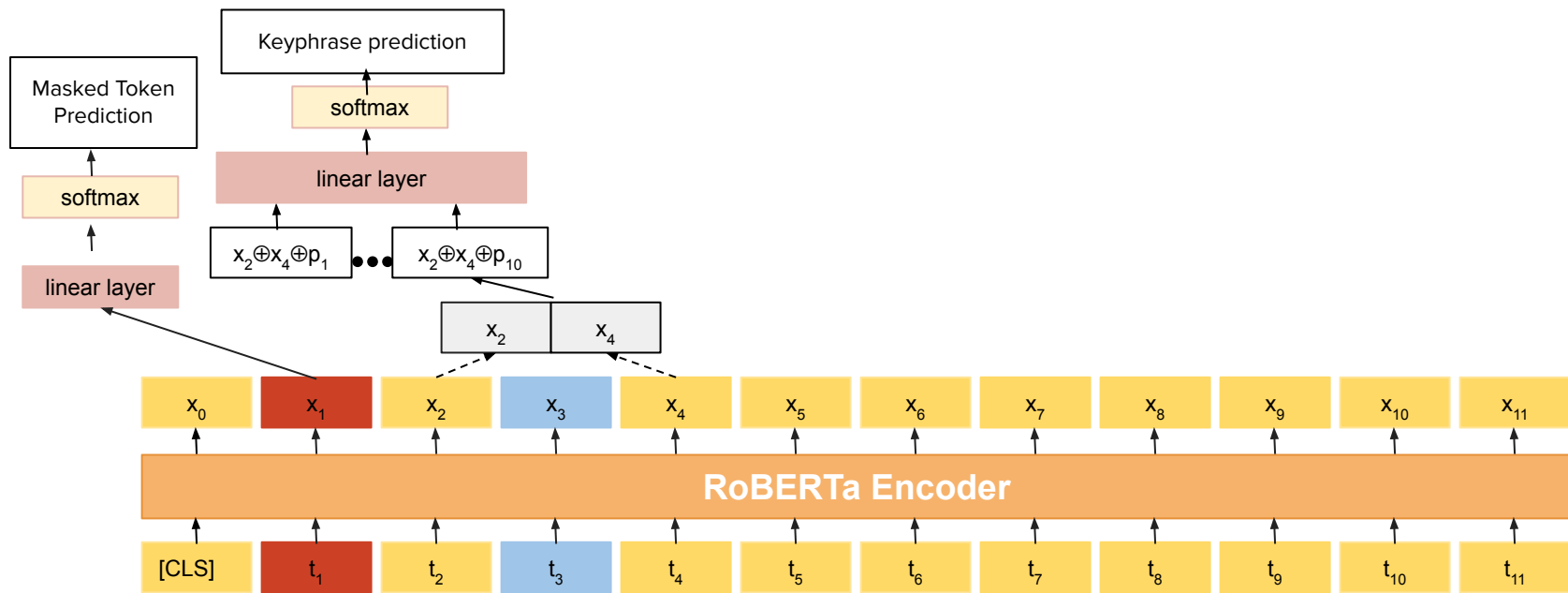


Text Infilling with Encoder

- ❖ Replace the entire span, in this case a keyphrase, with a single [MASK] token
- ❖ Predict the original tokens using positional embeddings in conjunction with boundary tokens
- ❖ More challenging task than SpanBERT's objective of individual masked token predictions as the model must predict how many tokens correspond to a span
- ❖ Different from SpanBERT, which does not penalize incorrect predictions of a sequence of tokens within a masked span, we propose a cumulative loss across all tokens in the masked span to capture intra-span token relationships to learn better span representations

$$\mathcal{L}_{\text{Infill}}(\theta) = \sum_{i=1}^{T_{max}} \log p(x_i | \mathbf{y}_i)$$

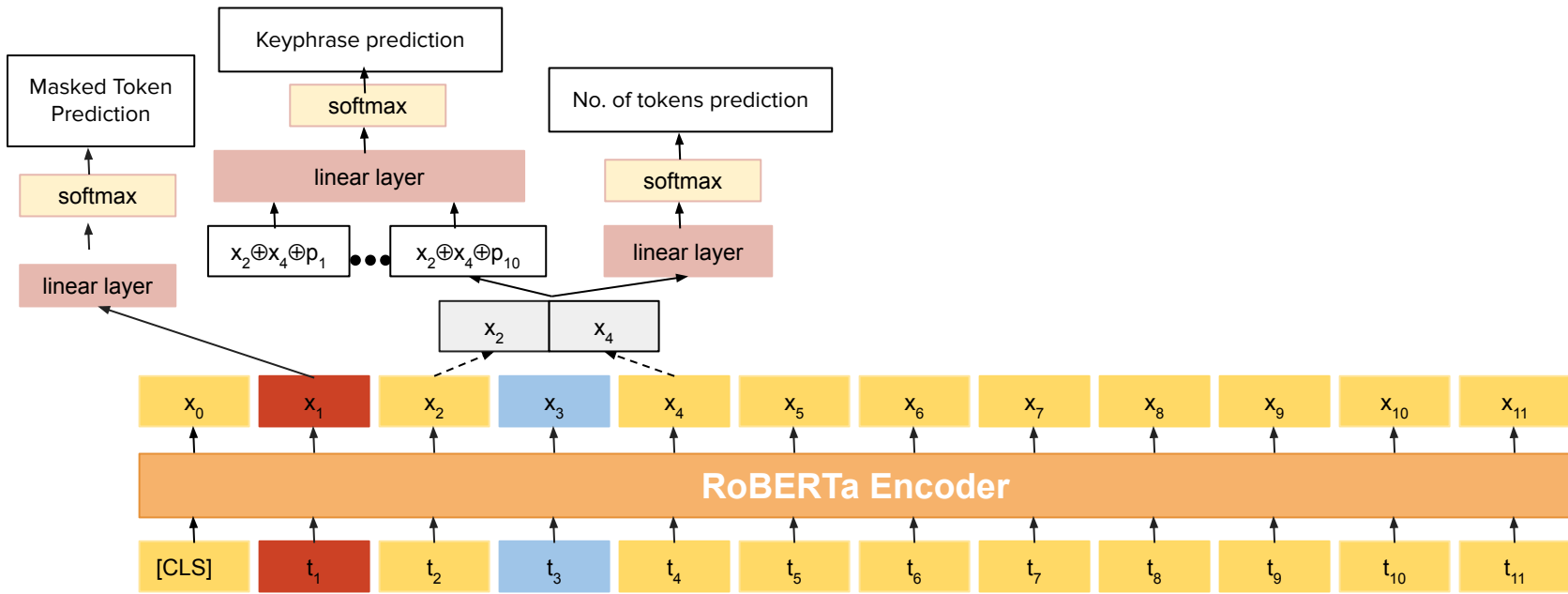
Keyphrase Masking - Keyphrase Boundary Infilling (KBI)



■ ordinary token for masking ■ keyphrase for masking

t_i - i^{th} token, x_i - i^{th} token embedding, p_i - i^{th} position embedding

Keyphrase Masking - Keyphrase Boundary Infilling (KBI)



■ ordinary token for masking
 ■ keyphrase for masking

t_i - i^{th} token, x_i - i^{th} token embedding, p_i - i^{th} position embedding

Keyphrase Replacement

Keyphrase Replacement

Keyphrases are an important means of [MASK], clustering, and topic search. Only a small minority of documents have author-assigned keyphrases, and manually assigning keyphrases to existing documents is [MASK] laborious. Therefore it is highly desirable to automate the [MASK]. This paper shows that a simple procedure for **keyphrase extraction** based on the [MASK] learning scheme performs comparably to the state of the art. It goes on to explain how this procedure's performance can be [MASK] by automatically tailoring the extraction process to the particular document collection at hand. Results on a large collection of technical reports in computer science show that the quality of the extracted keyphrases improves significantly when domain-specific information [MASK] exploited.

Keyphrase Replacement

Keyphrase Replacement

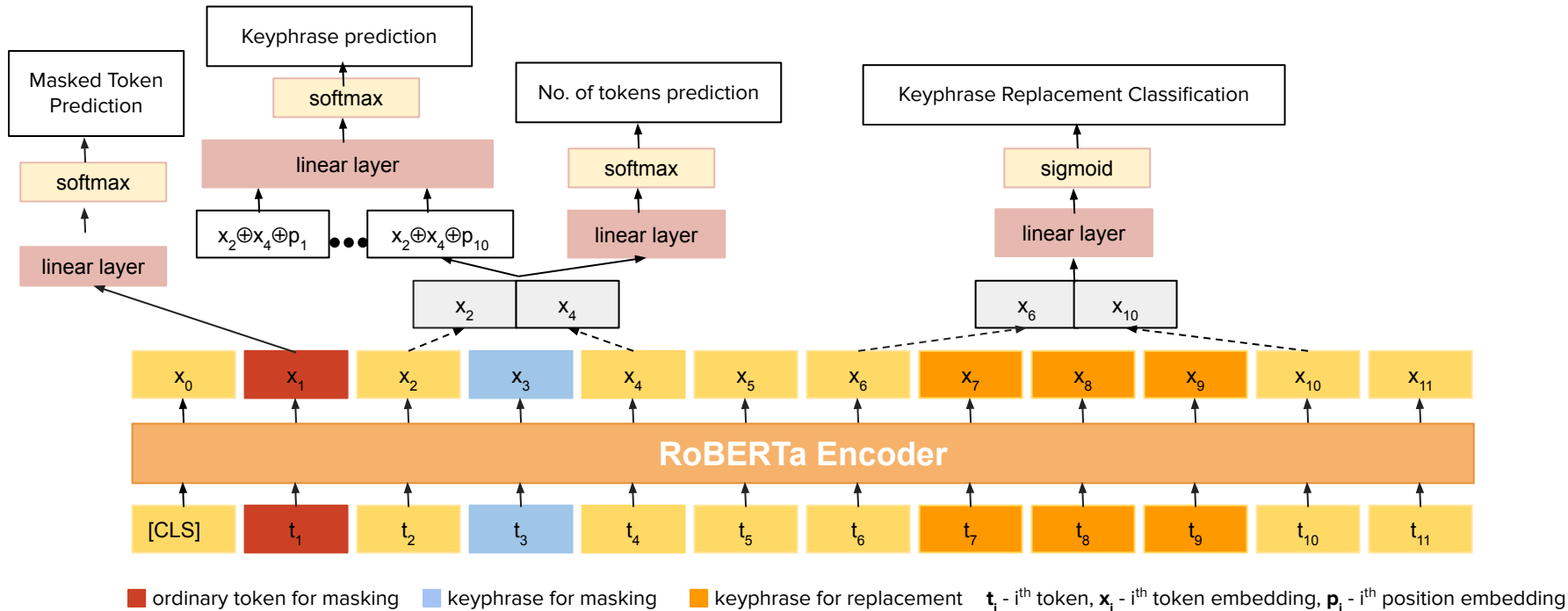


Keyphrase Vocabulary
500K

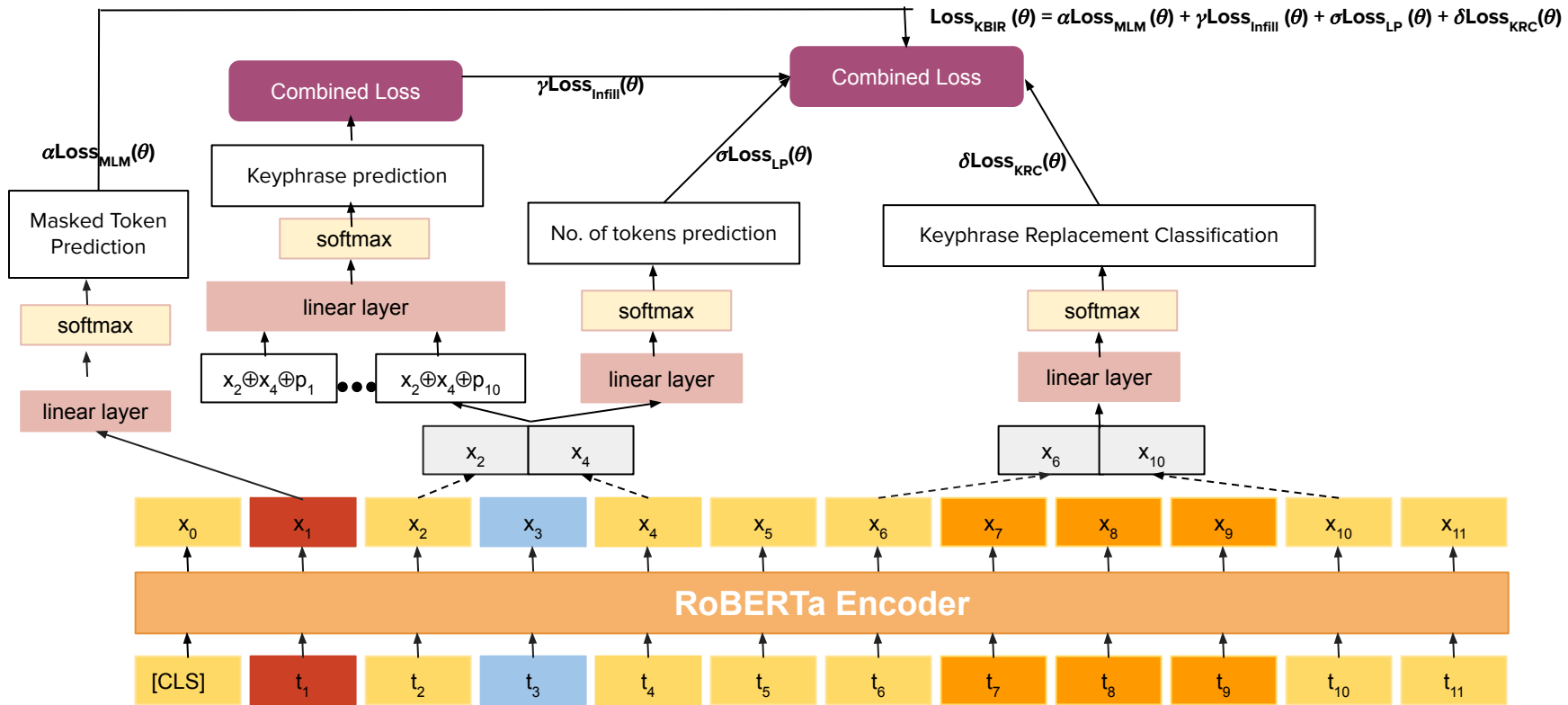
Keyphrases are an important means of [MASK], clustering, and topic search. Only a small minority of documents have author-assigned keyphrases, and manually assigning keyphrases to existing documents is [MASK] laborious. Therefore it is highly desirable to automate the [MASK]. This paper shows that a simple procedure for **dystropic epidermolysis bullosa** based on the [MASK] learning scheme performs comparably to the state of the art. It goes on to explain how this procedure's performance can be [MASK] by automatically tailoring the extraction process to the particular document collection at hand. Results on a large collection of technical reports in computer science show that the quality of the extracted keyphrases improves significantly when domain-specific information [MASK] exploited.

Contrastive Learning

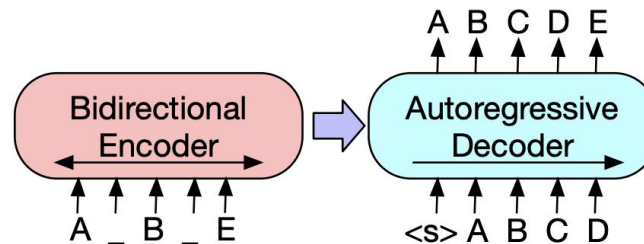
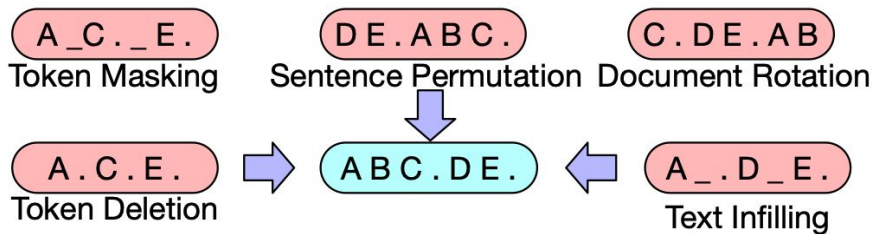
KBIR - **K**eyphrase **B**oundary **I**nfilling and **R**eplacement



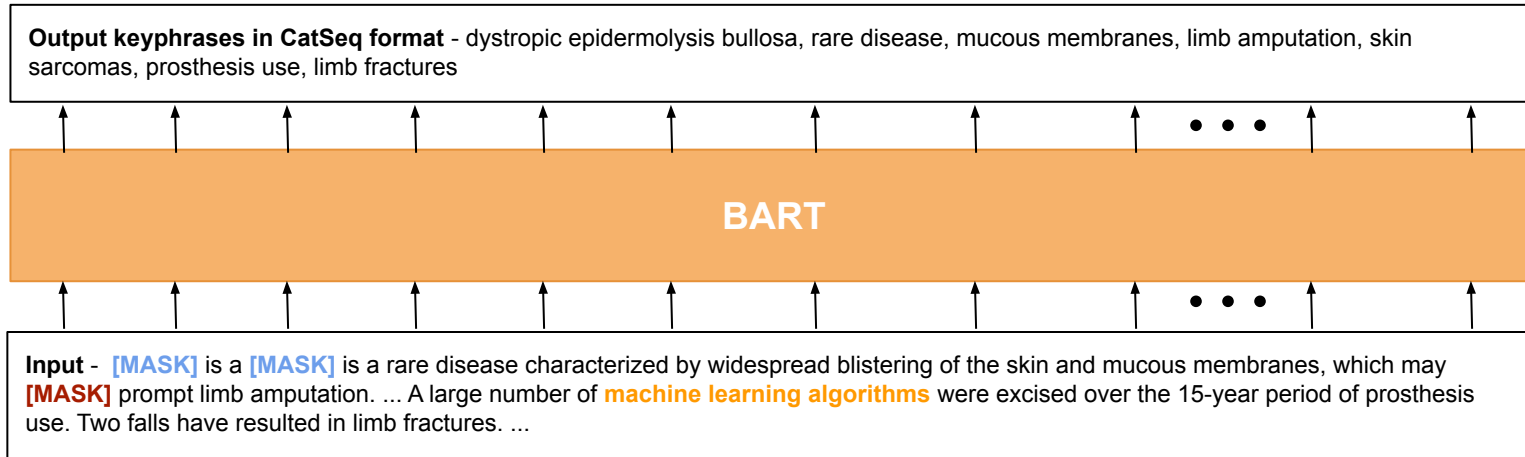
KBIR - Keyphrase Boundary Infilling and Replacement



BART



KeyBART

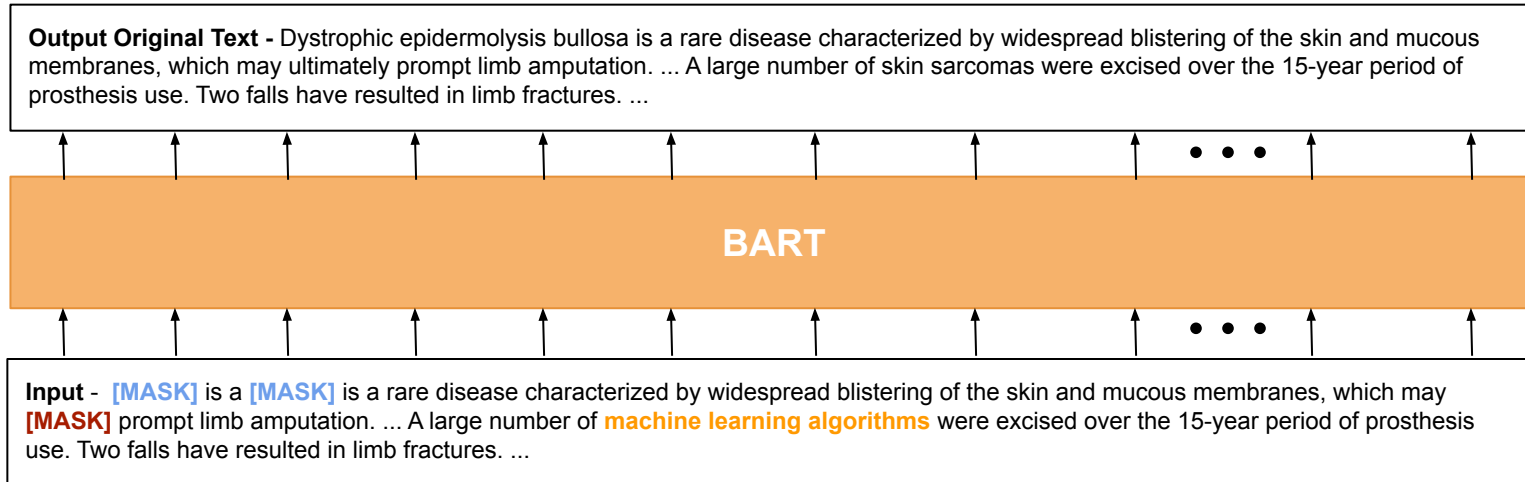


■ ordinary token for masking
 ■ keyphrase for masking
 ■ keyphrase for replacement

Original Text - **Dystrophic epidermolysis bullosa** is a **rare disease** characterized by widespread blistering of the skin and mucous membranes, which may **ultimately** prompt limb amputation. ... A large number of **skin sarcomas** were excised over the 15-year period of prosthesis use. Two falls have resulted in limb fractures. ...

Keyphrases - [dystropic epidermolysis bullosa, rare disease, mucous membranes, limb amputation, skin sarcomas, prosthesis use, limb fractures.]

KeyBART-DOC



■ ordinary token for masking
 ■ keyphrase for masking
 ■ keyphrase for replacement

Original Text - **Dystrophic epidermolysis bullosa** is a **rare disease** characterized by widespread blistering of the skin and mucous membranes, which may **ultimately** prompt limb amputation. ... A large number of **skin sarcomas** were excised over the 15-year period of prosthesis use. Two falls have resulted in limb fractures. ...

Keyphrases - [dystrophic epidermolysis bullosa, rare disease, mucous membranes, limb amputation, skin sarcomas, prosthesis use, limb fractures.]

Experiment Setup

OAGKx Corpus - 23 million scientific articles from various domains

Model	Batch	Steps	Warmup	α	γ	σ	δ	MLM	KI	KR	MISL	MKR
RoBERTa-extended	4	130k	2.5k	1.0	0.0	0.0	0.0	0.15	0.0	0.0	-	-
KBI	4	130k	2.5k	1.0	0.33	1.0	0.0	0.15	0.2	0.0	10	-
KBIR	2	260k	5k	1.0	0.33	1.0	2.0	0.05	0.2	0.4	10	20
KeyBART	4	130k	2.5k	-	-	-	-	0.05	0.2	0.4	10	20
KeyBART-DOC	2	260k	5k	-	-	-	-	0.05	0.2	0.4	10	20

Table 1: Hyperparameters for our pre-training strategies, all models were trained across 8 Tesla V100 GPUs with a learning rate of $1e-5$ using the Adam (Kingma and Ba, 2015) optimizer. Difference in number of steps is to account for changes in batch size. MLM, Keyphrase Infilling (KI) and Keyphrase Replacement (KR) show the probability of this perturbation occurring in the original text. MLM probability is reduced for KBIR in line with (Xiong et al., 2019). Maximum Infill Span Length (MISL) and Maximum Keyphrase Replacements (MKR), are based on averages from OAGKX and computational reasons. The coefficients for the loss are used to normalize the magnitude of loss across the different tasks.

Keyphrase Extraction using KBIR

Model	Inspec	SE10	SE17
RoBERTa+BiLSTM-CRF	59.5	27.8	50.8
RoBERTa+TG-CRF	60.4	29.7	52.1
SciBERT+Hypernet-CRF	62.1	36.7	54.4
RoBERTa+Hypernet-CRF	62.3	34.8	53.3
RoBERTa-extended-CRF*	62.09	40.61	52.32
KBI-CRF*	62.61	40.81	59.7
KBIR-CRF*	62.72	40.15	62.56

Table 2: F1 scores for keyphrase extraction on Inspec, SE10 and SE17 datasets (* LMs trained by us).

SE10 - SemEval 2010, SE17 - SemEval 2017

Present Keyphrase Generation using KeyBART

Model	Inspec		NUS		Krapivin		SemEval		KP20k	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
catSeq (Yuan et al., 2018)	22.5	26.2	32.3	39.7	26.9	35.4	24.2	28.3	29.1	36.7
catSeqTG (Chen et al., 2019)	22.9	27	32.5	39.3	28.2	36.6	24.6	29.0	29.2	36.6
catSeqTG-2RF1 (Chan et al., 2019)	25.3	30.1	37.5	43.3	30	36.9	28.7	32.9	32.1	38.6
GANMR (Swaminathan et al., 2020)	25.8	29.9	34.8	41.7	28.8	36.9	-	-	30.3	37.8
ExHiRD-h (Chen et al., 2020)	25.3	29.1	-	-	28.6	34.7	28.4	33.5	31.1	37.4
Transformer (Ye et al., 2021)	28.15	32.56	37.07	41.91	31.58	36.55	28.71	32.52	33.21	37.71
BART*	23.59	28.46	35.00	42.65	26.91	35.37	26.72	31.91	29.25	37.51
KeyBART-DOC*	24.42	29.57	31.37	39.24	24.21	32.60	24.69	30.50	28.82	37.59
KeyBART*	24.49	29.69	34.77	43.57	29.24	38.62	27.47	33.54	30.71	39.76
KeyBART* (no finetune)	30.72	36.89	18.86	21.67	18.35	20.46	20.25	25.82	12.57	15.41

Table 3: Keyphrase generation for present keyphrases. SOTA is marked in **Bold** and our best performing models as **Bold-Italicized** (* LMs trained by us).

Absent Keyphrase Generation using KeyBART

Model	Inspec		NUS		Krapivin		SemEval		KP20k	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
catSeq (Yuan et al., 2018)	0.4	0.8	1.6	2.8	1.8	3.6	1.6	2.8	1.5	3.2
catSeqTG (Chen et al., 2019)	0.5	1.1	1.1	1.8	1.8	3.4	1.1	1.8	1.5	3.2
catSeqTG-2RF1 (Chan et al., 2019)	1.2	2.1	1.9	3.1	3.0	5.3	2.1	3.0	2.7	5.0
GANMR (Swaminathan et al., 2020)	1.3	1.9	2.6	3.8	4.2	5.7	-	-	3.2	4.5
ExHiRD-h (Chen et al., 2020)	1.1	2.2	-	-	2.2	4.3	1.7	2.5	1.6	3.2
Transformer (Ye et al., 2021)	1.02	1.94	2.82	4.82	3.21	6.04	2.05	2.33	2.31	4.61
BART*	1.08	1.96	<i>1.80</i>	<i>2.75</i>	2.59	4.91	1.34	1.75	1.77	3.56
KeyBART-DOC*	0.99	2.03	1.39	2.74	2.40	4.58	1.07	1.39	1.69	3.38
KeyBART*	0.95	1.81	1.23	1.90	<i>3.09</i>	<i>6.08</i>	<i>1.96</i>	<i>2.65</i>	<i>2.03</i>	<i>4.26</i>
KeyBART* (no finetune)	1.83	2.92	1.46	2.19	1.29	2.09	1.12	1.45	0.70	1.14

Table 4: Keyphrase generation for absent keyphrases. SOTA is marked in **Bold** and our best performing models as ***Bold-Italicized*** (* LMs trained by us).

NER and RE using KBIR

NER - Named Entity Recognition

Model	F1
LSTM-CRF (Lample et al., 2016)	91.0
ELMo (Peters et al., 2018)	92.2
BERT (Devlin et al., 2018)	92.8
(Akbik et al., 2019)	93.1
(Baeovski et al., 2019)	93.5
LUKE (Yamada et al., 2020)	94.3
LUKE w/o entity attention	94.1
RoBERTa (Yamada et al., 2020)	92.4
RoBERTa-extended*	92.54
KBI*	92.73
KBIR*	<i>92.97</i>

Table 5: Named Entity Recognition (NER) results on CONLL-2003. SOTA is marked in **Bold** and our best performing models as ***Bold-Italicized***.

RE - Relation Extraction

Model	F1
BERT (Zhang et al., 2019)	66.0
C-GCN (Zhang et al., 2018)	66.4
ERNIE (Zhang et al., 2019)	68.0
SpanBERT (Joshi et al., 2020)	70.8
MTB (Baldini Soares et al., 2019)	71.5
KnowBERT (Peters et al., 2019)	71.5
KEPLER (Wang et al., 2019)	71.7
K-Adapter (Wang et al., 2021)	72.0
LUKE (Yamada et al., 2020)	72.7
LUKE w/o entity attention	72.2
RoBERTa (Wang et al., 2021)	71.3
RoBERTa-extended*	70.94
KBI*	70.71
KBIR*	<i>71.0</i>

Table 6: Relation Extraction (RE) results on TACRED. State-of-the-art is marked in **Bold** and our best performing models as ***Bold-Italicized***.

Question Answering using KBIR

Model	EM	F1
BERT (Devlin et al., 2018)	84.2	91.1
XLNet (Yang et al., 2019)	89.0	94.5
ALBERT (Lan et al., 2019)	89.3	94.8
LUKE (Yamada et al., 2020)	89.8	95.0
LUKE w/o entity attention	89.2	94.7
RoBERTa (Liu et al., 2019)	88.9	94.6
RoBERTa-extended*	88.88	94.55
KBI*	88.97	94.7
KBIR*	<i>89.04</i>	<i>94.75</i>

Table 7: Question Answering (QA) results on SQuAD v1.1 on the DEV set. State-of-the-art is marked in **Bold** and our best performing models as ***Bold-Italicized***.

Summarization using KeyBART and KeyBART-DOC

Model	R1	R2	RL
BART (Lewis et al., 2019)	44.16	21.28	40.9
BART*	42.93	20.12	39.72
KeyBART-DOC*	42.92	20.07	39.69
KeyBART*	<i>43.10</i>	<i>20.26</i>	<i>39.90</i>

Table 8: Summarization results on CNN/DailyMail dataset. Our best performing models are marked as ***Bold-Italicized***.

Resources

arXiv > cs > arXiv:2112.08547

Computer Science > Computation and Language

[Submitted on 16 Dec 2021]

Learning Rich Representation of Keyphrases from Text

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, Rajarshi Bhowmik

```
from transformers import AutoModel  
model = AutoModel.from_pretrained("bloomberg/KeyBART")
```



```
from transformers import AutoModel  
model = AutoModel.from_pretrained("bloomberg/KBIR")
```



Resource Constrained Keyphrase Generation

- ❖ Concurrent work to KeyBART
- ❖ Trains in a resource constrained setting
 - D_{kp} - used for finetuning
 - D_{aux} - used for pretraining
- ❖ Pretraining Objectives
 - **Salient Span Recovery**
 - Selects spans using TF-IDF and masks them
 - Recovers the masked spans during training
 - **Salient Span Prediction**
 - Selects spans using Tf-IDF and masks them
 - Generates the masked out spans during training

From Fundamentals to Recent Advances A Tutorial on Keyphrasification



All materials available at
<https://keyphrasification.github.io/>

