Learning Rich Representation of Keyphrases from Text

Mayank Kulkarni
Bloomberg, USA
mkulkarni24@bloomberg.net

Debanjan Mahata[*]
Moody's Analytics, USA
Debanjan.Mahata@moodys.com

Ravneet Arora
Bloomberg, USA
rarora62@bloomberg.net

Rajarshi Bhowmik
Bloomberg, USA
rbhowmik6@bloomberg.net

Abstract

In this work, we explore how to train task-specific language models aimed towards learning rich representation of keyphrases from text documents. We experiment with different masking strategies for pre-training transformer language models (LMs) in discriminative as well as generative settings. In the discriminative setting, we introduce a new pre-training objective - **K**eyphrase **B**oundary **I**nfilling with **R**eplacement (**KBIR**), showing large gains in performance (upto 8.16 points in F1) over SOTA, when the LM pre-trained using KBIR is fine-tuned for the task of keyphrase extraction. In the generative setting, we introduce a new pre-training setup for BART - **KeyBART**, that reproduces the keyphrases related to the input text in the CatSeq format, instead of the denoised original input. This also led to gains in performance (upto 4.33 points in F1@M) over SOTA for keyphrase generation. Additionally, we also fine-tune the pre-trained language models on named entity recognition (NER), question answering (QA), relation extraction (RE), abstractive summarization and achieve comparable performance with that of the SOTA, showing that learning rich representation of keyphrases is indeed beneficial for many other fundamental NLP tasks.

1 Introduction and Background

Keyphrases capture the most salient topics of a document and facilitates extreme summarization. Identifying them in an automated way from a text document can be useful for several downstream tasks - classification (Hulth and Megyesi, 2006), clustering (Hammouda et al., 2005), summarization (Qazvinian et al., 2010; Zhang et al., 2004), reviewer and document recommendation (Augenstein et al., 2017), and many different information retrieval tasks such as enabling semantic and faceted search (Sanyal et al., 2019; Gutwin et al., 1999), query expansion (Song et al., 2006), and interactive document retrieval (Jones and Staveley, 1999).

Keyphrases could either be *extractive* (part of the document) or *abstractive* (not part of the document). Prior works have referred to them as present and absent keyphrases, respectively. Automatically identifying them entails the process of detecting the extractive (Hasan and Ng, 2014) and generating the abstractive keyphrases (Çano and Bojar, 2019a) from a given document. While extractive approaches have mostly dominated over the generative ones with higher accuracies (Çano and Bojar, 2019b), the task is far from solved and the performances of the present systems are worse in comparison to many other NLP tasks (Liu et al., 2010). Some of the major challenges are the varied length of the documents to be processed, their structural inconsistency and developing strategies that can perform well in different domains.

Most of the prior work on identifying keyphrases using deep learning techniques have concentrated on developing new architectures and frameworks based on different training paradigms such as seq2seq (Meng et al., 2017; Yuan et al., 2018; Zhang et al., 2017a; Chen et al., 2018; Ye and Wang, 2018; Chen et al., 2019; Ye et al., 2021), sequence tagging (Alzaidy et al., 2019), reinforcement learning (Chan et al., 2019), adversarial training (Swaminathan et al., 2020) and game theory (Saxena et al., 2020). Although, there has been tremendous progress in learning better semantic and syntactic representation of language at different levels - characters, words, phrases, sentences and documents (Liu et al., 2020b), there hasn't been any effort in learning rich pre-trained representations of keyphrases, which is the major focus of this work.

Transformer language models when pre-trained on large corpora with different pre-training objectives (Qiu et al., 2020) have shown great success

[*]This work was done by the author Debanjan Mahata as an employee of Bloomberg

in various downstream tasks on fine-tuning, including the tasks of keyphrase extraction (Sahrawat et al., 2019; Martinc et al., 2020; Santosh et al., 2020) and generation (Liu et al., 2020a). However, pre-training objectives tailored towards learning better representation of keyphrases that can result in improving the performance of identifying and generating keyphrases from text have not yet been explored. This motivated us to look into this specific problem and make an attempt to answer the following questions:

Q1 *- Can we formulate a pre-training objective for language models that can learn better representation of keyphrases?*

Previous work explored training language models for learning better representation of text spans (Joshi et al., 2020), summary sentences (Zhang et al., 2020), and tokens for named entity recognition (Yamada et al., 2020). To effectively learn rich representation of keyphrases in a BERT like discriminative setup, we propose a new pre-training objective - **Keyphrase Boundary Infilling with Replacement (KBIR)** (Section 2) which utilizes a multi-task learning setup for optimizing a combined loss of random token Masked Language Modeling (MLM) (Devlin et al., 2018), **Keyphrase Boundary Infilling (KBI)** (Section 2.1) and **Keyphrase Replacement Classification (KRC)** (Section 2.2).

We also propose a new setup for pre-training BART (Lewis et al., 2019) - **KeyBART** (Section 3), focused towards learning better representation of keyphrases in a generative setting. Instead of reproducing the denoised input text as proposed in the original setup, we produce the keyphrases associated with the input document in the CatSeq (Meng et al., 2017) format from a corrupted input.

Q2 *- Does learning rich representation of keyphrases in a language model lead to performance gains for the tasks of keyphrase extraction and generation?*

One of the key contributions of this work is the introduction of KBIR, which is the combination of the KBI and KRC objectives with MLM that helps to learn good representation of keyphrases. This is validated by obtaining SOTA performance for the task of keyphrase extraction on three benchmark datasets (Section 4.2.1), surpassing the existing SOTA (Duan et al., 2021) by at most 8.16 F1 points on the SemEval 2017 corpus (Augenstein et al., 2017).

We also evaluated the KeyBART approach across five benchmark datasets for the task of keyphrase generation and obtained SOTA performances for both present and absent keyphrases (Section 4.2.2). Our best model surpassed the SOTA ONE2SEQ model (Ye et al., 2021) by 4.33 F1@M points and 0.72 F1@M points on Inspec (Hulth, 2003a) for present and absent keyphrases respectively.

Q3 *- Do rich keyphrase representations aid other fundamental tasks in NLP such as NER, QA, RE and summarization?*

It is to be noted, that although we trained our models on a large corpus of 23 million scientific articles, we find that it performs reasonably well when fine-tuned on datasets that do not belong to the scientific domain for different NLP tasks as shown in Section 4.2.3, 4.2.4, 4.2.5 and 4.2.6. This also suggests that identifying keyphrases in the context of an input text is a fundamental NLP task and a language model trained to learn optimal representation of keyphrases can aid many other tasks.

To summarize the main contributions that we make in this work are:

- We make the first attempt to train task-specific language models in discriminative as well as generative settings geared towards learning rich representation of keyphrases from text.
- We introduce a novel pre-training objective **Keyphrase Boundary Infilling with Replacement (KBIR)** and train a new language model that achieves SOTA performance for the task of keyphrase extraction.
- We propose a new setup - **KeyBART** for pre-training a generative language model for learning better representation of keyphrases and achieve SOTA performance on the task of keyphrase generation.
- We also empirically show how learning rich keyphrase representations from text is also useful for other NLP tasks like NER, RE, QA and summarization by achieving near SOTA performances in all of them using our language models trained using KBIR objective and KeyBART settings.

We have made our models[1] [2] publicly avail-

[1]https://huggingface.co/bloomberg/KeyBART

[2]https://huggingface.co/bloomberg/KBIR

able. We also make our pre-training code[3] available. Next, we give a detailed description of the methods that we propose in this work.

2 Keyphrase Boundary Infilling with Replacement (KBIR)

In the previous section, we mentioned various methods that aim at learning representations of text spans. Unlike LMs like SpanBERT (Joshi et al., 2020) and PEGASUS (Zhang et al., 2020) whose primary objective is to learn representations of random or heuristically chosen spans of text, the intuition behind learning good keyphrase representation is to provide the LM the ability to learn spans as well as to identify important phrases (in this case keyphrases) in the context of an input text. This motivated us to devise a framework that can optimize both of these objectives. Towards this effort, we propose a new pre-training objective *Keyphrase Boundary Infilling with Replacement* (KBIR) which is composed of two individual tasks - *Keyphrase Boundary Infilling* (KBI) and *Keyphrase Replacement Classification* (KRC) jointly learnt in a multi-task learning setup as shown in Figure 1. We build our framework on top of RoBERTa which implements random token Masked Language Modeling (MLM), therefore making our LM essentially optimizing MLM along with KBI and KRC objectives. In the following section, we describe the individual components of our framework.

2.1 Keyphrase Boundary Infilling (KBI)

To effectively learn span representations of keyphrases, we propose a new pre-training objective that builds upon the Span Boundary Objective (SBO) from SpanBERT (Joshi et al., 2020) and the Text Infilling setup from BART (Lewis et al., 2019). Similar to BART, we replace the entire span, in this case a keyphrase, with a single [MASK] token as shown in Figure 2 and predict the original tokens using positional embeddings in conjunction with boundary tokens. Text Infilling is a more challenging task than SpanBERT's objective of individual masked token predictions as the model must predict how many tokens correspond to a span (Lewis et al., 2019). Different from SpanBERT, which does not penalize incorrect predictions of a sequence of tokens within a masked span, we propose a cumulative loss (Equation 2) across all tokens in the masked span to capture intra-span token relationships to learn better span representations. *Text infilling, to the best of our knowledge, has not been explored in a discriminative setup as done in this work.*

We denote the output of the transformer encoder for each token x_l in the sequence $x_1, \ldots, x_L$ as $\mathbf{x_l}$. However, since the entire span of tokens $(x_s, ..., x_e)$ of a keyphrase y_m is masked with a mask token x_m, it is represented with a single vector $\mathbf{x_m}$, where (s, e) indicates its start and end positions and m represents the index of a masked keyphrase span. We set a maximum possible number of tokens corresponding to a keyphrase span, T_{max} such that $i \in [1, T_{max}]$. We then predict the sequence of tokens to replace x_m using the output encodings of the external boundary tokens x_{s-1} and x_{e+1}, as well as the position embedding $\mathbf{p_i}$ of the target token as shown in Equation 1.

$$\mathbf{y_i} = f(\mathbf{x_{s-1}}, \mathbf{x_{e+1}}, \mathbf{p_i}) \tag{1}$$

where positional embeddings use relative positions of the masked tokens with respect to the left boundary token x_{s-1}. We use Layer Normalization (Ba et al., 2016) and GeLU (Hendrycks and Gimpel, 2016) activation function to represent $f(\Delta)$. We then use the vector representation $\mathbf{y_i}$ to predict the potential token x_i and compute the cumulative cross-entropy loss for each i present within the unmasked x_m as shown in Equation 2.

$$\mathcal{L}_{\text{Infill}}(\boldsymbol{\theta}) = \sum_{i=1}^{T_{max}} \log p\left(x_i|\mathbf{y_i}\right) \tag{2}$$

In addition to predicting the actual tokens, we use a classification head to predict the expected number of tokens corresponding to the [MASK] in anticipation of providing a stronger learning signal. Each possible length of the [MASK] is represented as a class and therefore, the number of such classes is equal to the maximum number of possible tokens (T_{max}). The architecture used for classifying the number of tokens is a single linear layer which is trained with cross-entropy loss $\mathcal{L}_{\text{LP}}(x_m, z_m)$ along with the infilled masked token x_m and the corresponding actual length of the span class z_m.

The Keyphrase Boundary Infilling (KBI) objective is formally represented as:

$$\mathcal{L}_{\text{KBI}}(\boldsymbol{\theta}) = \alpha\mathcal{L}_{\text{MLM}}(\boldsymbol{\theta}) + \gamma\mathcal{L}_{\text{Infill}}(\boldsymbol{\theta}) + \sigma\mathcal{L}_{\text{LP}}(\boldsymbol{\theta}) \tag{3}$$

[3] https://github.com/bloomberg/kbir_keybart

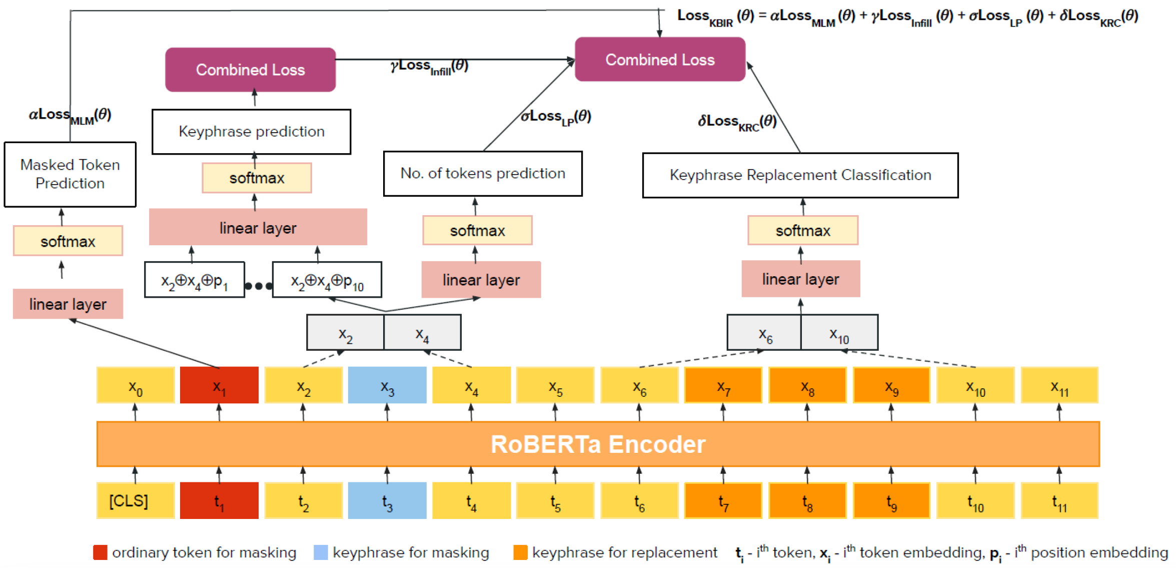

Figure 1: The KBIR model architecture for the training phase. The Random Token MASK is denoted in red, the Keyphrase MASK is denoted in blue and the Replaced Keyphrase in orange.

where α, γ and σ are co-efficients applied to each loss and are primarily used to normalize the losses across the tasks.

We propose this pre-training objective to be used with keyphrases, however the objective is generic enough to be applied to any spans of text, these could be keyphrases, entities or even random spans.

2.2 Keyphrase Replacement Classification (KRC)

Apart from learning representations of keyphrase spans, we wanted our framework to have the ability to identify them within the context of a text input. Motivated by WKLM (Xiong et al., 2019) that explores pretraining a language model through weak supervision by replacing entities with random entities of the same type that belongs to a knowledge base, we adapt it to replace keyphrases by randomly choosing another keyphrase of variable length from the universe of keyphrases identified in a tagged corpus. The KRC task is then modeled as a binary classification task to determine whether a keyphrase is replaced or retained.

To implement this strategy, we construct a keyphrase universe by identifying the set of unique keyphrases tagged across the entire dataset. We then randomly shuffle this keyphrase universe and restrict it to 500,000 keyphrases for computational complexity. We use the concatenated representation of boundary tokens of a keyphrase x_{s-1} and x_{e+1} as input to a linear classifier as shown in Figure 1. Given the label y_k representing whether a keyphrase was replaced or not, the objective here is to minimize the binary cross-entropy loss $\mathcal{L}_{\text{KRC}}((x_{s-1} + x_{e+1}), y_k)$.

Finally, in order to train a LM with an objective of learning good keyphrase representations we use the KBIR pre-training strategy in which we jointly optimize the KBI loss and the KRC loss along with the already existing MLM loss ($\mathcal{L}_{\text{MLM}}(\boldsymbol{\theta})$) in RoBERTa. This is formally shown in Equation 4.

$$\mathcal{L}_{\text{KBIR}}(\boldsymbol{\theta}) = \alpha\mathcal{L}_{\text{MLM}}(\boldsymbol{\theta}) + \gamma\mathcal{L}_{\text{Infill}}(\boldsymbol{\theta}) + \sigma\mathcal{L}_{\text{LP}}(\boldsymbol{\theta}) + \delta\mathcal{L}_{\text{KRC}}(\boldsymbol{\theta}) \quad (4)$$

3 KeyBART

We also explored learning a generative LM for the text generation tasks such as keyphrase generation and abstractive summarization. Our hypothesis behind the proposed setup is that masking and replacing task-specific spans, in this case keyphrases, that need to be re-generated should allow the generative model to learn a better representation of surrounding tokens and also the spans themselves.

BART (Lewis et al., 2019) generates sequences of different lengths from the input perturbed with [MASK] tokens along with token addition and deletion. On similar lines, we propose learning rich keyphrase representations by attempting to generate the *Original Present Keyphrases* in the Catseq format as proposed in (Meng et al., 2017) from an input perturbed with token masking, keyphrase masking, and keyphrase replacement as shown in

Figure 2. We call this method **KeyBART**. We maintain the order of occurrence of the keyphrases in the original document and remove duplicate occurrences. We also use the same method for finding keyphrase replacements as used in KRC (Section 2.2). We don't explicitly try to model the keyphrase replacement through a replacement classification head, but rely on learning this implicitly as part of the generation task. Similar to BART, we use a reconstruction loss objective during training which is a cross-entropy loss between the output and set of expected keyphrases.

4 Experiments and Results

4.1 Language Modeling

Dataset - We use the OAGKX (Çano and Bojar, 2020) dataset which consists of 23 million scientific documents across multiple domains sampled from the Open Academic Graph with keyphrases tagged by the authors of the articles. The OAGKX contains keyphrases that appear in the abstract and also those which don't appear in the abstract, making it similar to the keyphrase generation setting with present and absent keyphrases. *To the best of our knowledge, we are the first to explore OAGKX dataset for pre-training a large language model.*

During LM pre-training we restricted the length of the input text for each sample to 512 tokens.

Note that we do not explicitly tag the keyphrases and use the readily available author tagged keyphrases associated with each document, which is a common practice in the scientific domain. This setup is analogous to how the Wikipedia corpus is used to perform entity specific pre-training in LUKE (Yamada et al., 2020) and WKLM (Xiong et al., 2019) among others.

We conducted a preliminary study of using TextRank, a baseline unsupervised keyphrase tagging techniques, to create a weakly supervised dataset but observed only marginal gains. More details are provided in Section 6 that would serve as a potential direction for future work. Additionally, we address Limitations and Ethical Concerns in Appendix 8.1.

Pre-training Strategies - We train LMs in different settings with different hyperparameters as shown in Table 1 (refer Appendix- 8.2 for more details). We pre-train in a discriminative setting with the **KBIR** method using RoBERTa (Liu et al., 2019) pre-trained weights. We also pre-train in a generative setting with the **KeyBART** method using BART (Lewis et al., 2019) pre-trained weights.

Ablations - Considering computational costs and environmental impact, we conduct a limited set of ablation studies to demonstrate the effectiveness of our proposed methods and also to demonstrate that the gains in performance are not due to additional data.[4] We pre-train using basic random token masking strategy as **RoBERTa-extended** ablating both KBI and KRC from KBIR, using RoBERTa pre-trained weights. We also pre-train using the **KBI** method, ablating KRC from KBIR using RoBERTa pre-trained weights. We pre-train BART's original denoising autoencoder strategy to recreate the original document as **KeyBART-DOC** by using BART pre-trained weights.

4.2 Downstream Task Evaluation

All our downstream evaluations are performed using HuggingFace Transformer's (Wolf et al., 2020) RoBERTa or BART architectures to facilitate reproducibility. We also specify all hyperparameters in Table 2. We add no additional parameters over RoBERTa or BART for the corresponding downstream evaluation architecture, demonstrating the effectiveness of our updated pre-trained weights.

4.2.1 Keyphrase Extraction

We report performance of our models for Keyphrase Extraction (KE) on Inspec (Hulth, 2003b), SemEval-2010 (SE10) (Kim et al., 2010), and SemEval-2017 (SE17) (Augenstein et al., 2017). (Sahrawat et al., 2019) explored KE as a sequence tagging task with contextual embeddings and demonstrate the effectiveness of a CRF. We compare our performance with RoBERTa+BiLSTM-CRF (Sahrawat et al., 2019), RoBERTa+TG-CRF (Chen et al., 2019), previous state-of-the-art model RoBERTa+Hypertnet-CRF, and SciBERT+Hypernet-CRF (Duan et al., 2021). However, different from these architectures, we do not use a LSTM/BiLSTM layer between the contextualized embeddings and the CRF. We fine-tune all the pre-trained language models on B-I-O tagged datasets for KE[5]. We use hyperparameters specified in (Sahrawat et al., 2019) and F1-score is used as the evaluation metric.

Table 3 shows our pre-trained LMs outperform SOTA by significant margins across all

[4]We attempted whole word masking keyphrases for both SBO and MLM for BASE model pre-training and observed no significant gains.

[5]https://github.com/midas-research/keyphrase-extraction-as-sequence-labeling-data

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 using 8 Tesla V100 GPUs with the Adam (Kingma and Ba, 2015) optimizer and a learning rate of 1e-5. Difference in number of steps is to account for changes in batch size while seeing the same number of data points across training regimes. 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.

Parameter	KE	NER	RE	QA	KG	SUM
Learning Rate	5e-5	1e-5	4e-5	3e-5	5e-5	5e-5
Batch	4	8	32	48	32	8
Epochs	100	5	10	2	300k	20k
GPUs	2	1	2	1	4	2

Table 2: Hyperparameters for our downstream task evaluations. KG and SUM specifies steps instead of epochs.

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 3: F1 scores for Keyphrase Extraction on Inspec, SE10 and SE17 datasets (* LMs trained by us).

three datasets despite having fewer parameters. While RoBERTa-extended, shows gains over RoBERTa+BiLSTM-CRF, this is expected since the domain of the continued pre-training data is more in line for KE evaluation. However, the models that explicitly learn keyphrase representations such as KBI and KBIR significantly outperform RoBERTa-extended. We believe the slight gain for SemEval-2010 is because of the small size of the dataset (130 - train, 100 - test).

4.2.2 Keyphrase Generation

We evaluate keyphrase generation (KG) performance on Inspec (Hulth, 2003b), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin et al., 2009), SemEval (Kim et al., 2010) and KP20K (Meng et al., 2017). The task is to generate the *CatSeq* output of the present and absent keyphrases for a given concatenated title and abstract, as done in previous works (Meng et al., 2017; Chen et al., 2019; Yuan et al., 2018). We use the *PresAbs* ordering of the keyphrases as that was shown to be the most effective representation in (Meng et al., 2021). Further, we only train a single model by fine-tuning on the KP20K dataset and perform inference on all the test datasets. Similar to (Meng et al., 2021) we use a beam width of 50 for beam search and restrict our maximum generated sequence length to 40 tokens. For our evaluation we use macro-averaged F1@5 and F1@M as in (Chan et al., 2019) and (Chen et al., 2020) for both present and absent keyphrase generation. F1@M evaluates all the keyphrases predicted by the model with the ground-truth keyphrases. F1@5, as the name suggests evaluates only the first 5 keyphrases, however when there are fewer than five keyphrases, random incorrect keyphrases are appended till it reaches five predictions. (Chan et al., 2019) show that without this appending F1@M is the same as F1@5, when predictions are fewer than five. (Ye et al., 2021) also present a ONE2SET training paradigm and for a fair comparison we compare to their Transformer (ONE2SEQ) results, since we also train in the ONE2SEQ paradigm and not ONE2SET.

In Table 4 and Table 5 we see that KeyBART is the most effective pre-training method achieving SOTA on most datasets for F1@M in present and absent KG. We believe our choice of perturbation of the input during the pre-training setup makes this model robust and helps it identify and generate keyphrases more effectively. We also observe that our results for F1@5 aren't as competitive as F1@M and we believe this is because our model tends to favor predicting fewer than 5 keyphrases and thus tends to suffer from the random addition of keyphrases for F1@5. More concretely, the average predicted keyphrases per document for SemEval is

	Inspec		NUS		Krapivin		SemEval		KP20k	
Model	**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 4: Keyphrase Generation for Present Keyphrases. SOTA is marked in **Bold** and our best performing models as ***Bold-Italicized***.

	Inspec		NUS		Krapivin		SemEval		KP20k	
Model	**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	***1.80***	***2.75***	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	***3.09***	**6.08**	***1.96***	***2.65***	***2.03***	***4.26***
KeyBART* (no finetune)	**1.83**	**2.92**	1.46	2.19	1.29	2.09	1.12	1.45	0.70	1.14

Table 5: Keyphrase Generation for Absent Keyphrases. SOTA is marked in **Bold** and our best performing models as ***Bold-Italicized***.

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
(Baevski 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*	***92.97***

Table 6: Named Entity Recognition results on CONLL-2003. SOTA is marked in **Bold** and our best performing models as ***Bold-Italicized***.

2.51, NUS is 2.86, Krapivin is 2.86, Inspec is 3.09 and KP20k is 2.73. The Inspec dataset is anomalous where the non-finetuned model performs significantly better, demonstrating the effectiveness of the KeyBART training strategy.

4.2.3 Named Entity Recognition

We report the performance of different models for the task of NER by conducting experiments on CoNLL-2003 dataset (Sang and De Meulder, 2003).

Table 6 demonstrate that KBI and KBIR have performance gains over RoBERTa on CoNLL-2003. With RoBERTa-extended, we see that only continued pre-training with the MLM objective results in minor gains. However, when we inspect the results for KBI and KBIR, we see consistent jumps in performance showing how both these architectures contribute in learning richer representations that directly impact NER performance. We hypothesize that KBIR is more effective at NER than KBI because the additional keyphrase replacement classification task builds richer boundary token representations making entity identification potentially easier. The results are also fairly competitive with SOTA NER models in literature despite the fact that we did not attempt modeling entities explicitly like existing SOTA model (Yamada et al., 2020).

4.2.4 Relation Extraction

The relation extraction (RE) task predicts relations among pairs of entity mentions in a text. We fine-tuned our models for the sentence-level relation extraction task using the popular TACRED benchmark dataset (Zhang et al., 2017b). TACRED contains more than 100,000 sentences with entities that belong to 23 different fine-grained semantic types and with 42 different relations among entities.

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*	***89.04***	***94.75***

Table 7: Question Answering 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***.

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*	***71.0***

Table 8: Relation Extraction results on TACRED. State-of-the-art is marked in **Bold** and our best performing models as ***Bold-Italicized***.

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*	***43.10***	***20.26***	***39.90***

Table 9: Summarization results on CNN/DailyMail dataset. Our best performing models are marked as ***Bold-Italicized***.

To fine-tune our models, we modified the input sequences to mark the start and end of the subject entity with @ and the object entity with #. We use the final layer representation of the [CLS] token as the input to a multi-class classifier.

The results in the top half of Table 8 are reported from the respective papers that use various input formatting strategy. Similar to (Zhou and Chen, 2021), we also observe that a model's performance depends heavily on the formatting of the input sequence. All models in the bottom half of the table are trained with the same input format mentioned above. We observe that our KBIR model performs slightly worse than the original RoBERTa model. We also observe similar trends for KBI and RoBERTa-extended models. We conjecture that the domain shift of the pre-training corpus is responsible for the slight performance degradation.

4.2.5 Question Answering

The relation between question answering (QA) and KE has been explored to some extent in (Subramanian et al., 2018), which leverages keyphrase extraction for question generation. Motivated by their work, we evaluate our models on SQuAD v1.1 (Rajpurkar et al., 2016) dataset for the extractive question answering task. For all the models, we use a maximum sequence length of 512 with a sliding window of size 128.

Table 7 reports the F1 and Exact Match (EM) scores achieved by different model architectures on the DEV set. We observe improved performance with KBI and KBIR as compared to RoBERTa. We have an interesting observation where RoBERTa-extended performs worse than RoBERTa and we conjecture that it is because of the domain shift in the pre-training data which comprises scientific articles. On the other hand, the models trained with keyphrase pre-training objectives are fairly competitive with the SOTA QA models. We explicitly include LUKE w/o entity attention since that removes the entity-aware attention module, making it slightly more comparable to our setup. We observe that KBIR outperforms it by a slim margin in F1. However, the performance is slightly lower in the EM scores. Note that our model does not yield the similar performance in EM as it does in F1 when compared to SOTA. A potential reason for this is that our model is more likely to identify keyphrases as answers.

4.2.6 Summarization

We fine-tune BART (Lewis et al., 2019), KeyBART-DOC and KeyBART on the CNN DailyMail (Hermann et al., 2015) summarization dataset (SUM). Keyphrase Generation is also considered as an extreme form of summarization and therefore, we expect to see improved performance for the summarization task. Since we were unable to reproduce the original BART scores for R1, R2 and RLSum, we used the reported hyperparameters to reproduce the results to best of our ability, accounting for minor implementation differences in framework versions. We hope this provides a more fair comparison with our model results. We do not claim SOTA for summarization models, rather want to demonstrate that there are potential performance gains by training on a keyphrase specific objective. This is demonstrated in Table 9 where we see that the standard denoising autoencoder setup results in marginal losses. However, training with the keyphrase generation objective improves the ROUGE scores across the board when compared with BART trained on the same dataset.

5 Qualitative Analysis

We perform a qualitative analysis on the SemEval-2010 dataset as it is the only common dataset between KE and KG tasks by leveraging predictions from the best performing models. We present examples in Table 10, in the Appendix, which captures the ground truth, extracted keyphrases and generated keyphrases (present and absent) for a given document from the SemEval dataset using our best performing models on the respective tasks. We observe that when the model tends to generate more keyphrases, it typically relies on the copy mechanism and hence most of the generated keyphrases are present in the text itself (Example 1). We also observe that when absent keyphrases are generated accompanied by a large number of generated keyphrases, they are usually a combination of two or more words directly present in the text such as 'user study' (Example 3). The example discusses how the authors study user behavior, potentially making 'user study' a fair prediction, however the ground truth would penalize the model if this was in the training phase. Finally, we observe more generated keyphrases when the model isn't able to identify keyphrases in text and doesn't rely heavily on the copy mechanism, but on it's understanding of the text. This results in keyphrases such as 'natural language processing' (Example 2). Although the prediction is not in the ground-truth, it aligns with the mentions of 'question answering' and 'linguistics'. This demonstrates that the model is indeed able to generate meaningful absent keyphrases. However, we observe that the model is not able to learn or infer world knowledge required to produce the absent keyphrases in the ground-truth. For keyphrase extraction, we see that the model tends to tag phrases more frequently than previous models, improving recall. We hypothesize that it is due to the model having a better understanding of keyphrases in a document because of the keyphrase masking perturbation and also the KRC task.

6 Other Experiments

We also attempted to use a combination of the Wikipedia (English) dump and S2ORC (Lo et al., 2019) corpora for pre-training our models. In order to obtain keyphrase tags for data at a large scale, we employed TextRank (Mihalcea and Tarau, 2004) on each document in the corpora. We set the maximum number of keyphrases to 10 for the TextRank algorithm and considered all keyphrases tagged by TextRank. We created random splits for our dataset to generate a train and development (dev) set. However, we found that keyphrases tagged in this manner added a lot of noise to the dataset and resulted in only marginal overall gains.

To train a keyphrase specific language model, we also used a combined generative and discriminative approach as introduced in ELECTRA (Clark et al., 2020). The generative approach made the model predict the masked tokens that are part of a keyphrase. In the discriminative approach, the original sequence was perturbed by replacing the tokens of a keyphrase with another semantically unrelated keyphrase. We were unable to stabilize the training for such a setup and didn't get promising results.

7 Conclusion and Future Work

We explored LMs capable of learning rich representations of keyphrases that achieve SOTA performance across multiple datasets for keyphrase extraction and generation tasks. Towards this effort, we proposed a new pre-training objective KBIR and a new training setup KeyBART. The trained LMs demonstrate their effectiveness by achieving SOTA or near SOTA performance for various downstream NLP tasks when fine-tuned on benchmark datasets spanning across multiple domains. As a next step, we would like to probe our LMs to understand them more and also gauge their effectiveness for the tasks of cross-domain keyphrase extraction and generation. We would also like to explore scaling these approaches to more datasets by revisiting more sophisticated unsupervised keyphrase tagging methods.

References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.

Rabah Alzaidy, Cornelia Caragea, and C Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The world wide web conference*, pages 2551–2557.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5360–5369, Hong Kong, China. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Erion Çano and Ondřej Bojar. 2019a. Keyphrase generation: A multi-aspect survey. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 85–94. IEEE.

Erion Çano and Ondřej Bojar. 2019b. Keyphrase generation: A text summarization struggle. *arXiv preprint arXiv:1904.00110*.

Erion Çano and Ondřej Bojar. 2020. Two huge title and keyword generation corpora of research articles. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6663–6671, Marseille, France. European Language Resources Association.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural keyphrase generation via reinforcement learning with adaptive rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy. Association for Computational Linguistics.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase generation with correlation constraints. *arXiv preprint arXiv:1808.07185*.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1105, Online. Association for Computational Linguistics.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R Lyu. 2019. Title-guided encoding for keyphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6268–6275.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wenying Duan, Xiaoxi He, Zimu Zhou, Hong Rao, and Lothar Thiele. 2021. Injecting Descriptive Meta-Information into Pre-Trained Language Models with Hypernetworks. In *Proc. Interspeech 2021*, pages 3216–3220.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1-2):81–104.

Khaled M Hammouda, Diego N Matute, and Mohamed S Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *International workshop on machine learning and data mining in pattern recognition*, pages 265–274. Springer.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Anette Hulth. 2003a. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anette Hulth. 2003b. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, page 216–223, USA. Association for Computational Linguistics.

Anette Hulth and Beáta Megyesi. 2006. A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 537–544.

Steve Jones and Mark S Staveley. 1999. Phrasier: a system for interactive document retrieval using keyphrases. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large dataset for keyphrases extraction. Technical report, University of Trento.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Rui Liu, Zheng Lin, and Weiping Wang. 2020a. Keyphrase prediction with pre-trained language model. *arXiv preprint arXiv:2004.10462*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 366–376.

Zhiyuan Liu, Yankai Lin, and Maosong Sun. 2020b. Representation learning and nlp. In *Representation Learning for Natural Language Processing*, pages 1–11. Springer.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2019. GORC: A large contextual citation graph of academic papers. *CoRR*, abs/1911.02782.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2020. Tnt-kid: Transformer-based neural tagger for keyword identification. *arXiv preprint arXiv:2003.09166*.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He. 2021. An empirical study on neural keyphrase generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4985–5007, Online. Association for Computational Linguistics.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326, Berlin, Heidelberg. Springer Berlin Heidelberg.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Vahed Qazvinian, Dragomir Radev, and Arzucan Özgür. 2010. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics (COLING 2010)*, pages 895–903.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. *arXiv preprint arXiv:1910.08840*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Tyss Santosh, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. Sasake: Syntax and semantics aware keyphrase extraction from research papers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5372–5383.

Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, Partha Pratim Das, Samiran Chattopadhyay, and TYSS Santosh. 2019. Enhancing access to scholarly publications with surrogate resources. *Scientometrics*, 121(2):1129–1164.

Arnav Saxena, Mudit Mangal, and Goonjan Jain. 2020. Keygames: A game theoretic approach to automatic keyphrase extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2037–2048.

Il Yeol Song, Robert B Allen, Zoran Obradovic, and Min Song. 2006. Keyphrase extraction-based query expansion in digital libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06)*, pages 202–209. IEEE.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88, Melbourne, Australia. Association for Computational Linguistics.

Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Shah, and Amanda Stent. 2020. A preliminary exploration of gans for keyphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8021–8030.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *CoRR*, abs/1911.06136.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *CoRR*, abs/1912.09637.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Hai Ye and Lu Wang. 2018. Semi-supervised learning for neural keyphrase generation. *arXiv preprint arXiv:1808.06773*.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2set: Generating diverse keyphrases as a set. *arXiv preprint arXiv:2105.11134*.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2018. One size does not fit all: Generating and evaluating variable number of keyphrases. *arXiv preprint arXiv:1810.05241*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Yong Zhang, Yang Fang, and Xiao Weidong. 2017a. Deep keyphrase generation with a convolutional sequence to sequence model. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pages 1477–1485. IEEE.

Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. 2004. World wide web site summarization. *Web Intelligence and Agent Systems: An International Journal*, 2(1):39–53.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *CoRR*, abs/2102.01373.

8 Appendix

8.1 Limitations and Ethical Concerns

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 kgCO$_2$eq/kWh. A cumulative of 6,144 hours of computation was performed on hardware of type Tesla V100-SXM2-32GB (TDP of 300W).We calculate that the combined cost of training all these models is 796.26 KGs of CO_2 eq. Estimations were conducted using the Machine Learning Impact calculator presented in (Lacoste et al., 2019). Given the computational cost and environmental impact we restrict the number of experiment settings and ablation studies conducted in the pre-training stage - so there may be some hyperparameters that could be further optimized but have not been realized and also more robust results to further demonstrate the effectiveness our proposed approach.

The OAGKX dataset we train on is made publicly available with a Creative Commons License 4.0 and is primarily focused on scientific documents from the Open Academic Graph. This decreases the potential for the pre-trained model to imbibe offensive content and also in not generating the same.

We also acknowledge that the current setup uses a dataset that already contains pre-tagged keyphrases similar to how Wikipedia leverages entities and our attempt at using a basic unsupervised keyphrase tagging technique did not yield much success as seen in Section 6. We believe that further research in exploring more sophisticated techniques for unsupervised keyphrase tagging would help overcome this hurdle. Further our proposed approaches should work on entities from Wikipedia or even random spans from the BookCorpus, and we encourage exploration of the same.

8.2 Pre-training Strategies

Figure 2 provides a visual representation of the various masking strategies we deploy, a more detailed description of each stage is available in the subsections below.

We train LMs in different settings and hyperparameters as listed in Table 1.

Discriminative Setting - We pre-train three language models in the discriminative setting as described below. All of them use the pre-trained weights of RoBERTa-large[6] as the initial weights and are trained by continuing the learning of the parameters on the OAGKX dataset using our pre-training strategies.

- **RoBERTa-extended** - Previous work (Gururangan et al., 2020) has shown that adding more data to pre-training a language model typically results in better downstream performance. To verify that our performance gains stem from modeling improvements and the new pre-training objectives proposed by us rather than addition of data, we extend the training of RoBERTa-large on the OAGKX corpus. We call this model *RoBERTa-extended*. This also ensures fair comparison of the LMs trained by us using our pre-training objectives with that of RoBERTa.

- **KBI** - During the pre-training of the LM with the KBI objective, we employ both token masking and keyphrase masking strategies as shown in Figure 2 and explained in Section 2.1. We randomly mask 15% of the tokens that are not included in keyphrase spans. We additionally mask 20% of the keyphrase spans with a single [MASK] token. We restrict the maximum number of tokens for a keyphrase mask span to 10, based on the average keyphrase length reported in (Çano and Bojar, 2020).

- **KBIR** - While pre-training the LM with the KBIR objective we employ 5% token masking, in line with the findings reported in (Xiong et al., 2019) and 20% of keyphrases are masked through keyphrase masking, with a maximum possible span size of 10 as in the KBI LM. Additionally, we replace 40% of the non-masked keyphrases with randomly sampled keyphrases from the keyphrase universe as explained in Section 2.2. We restrict the maximum number of keyphrases to be replaced to no more than 20, restricted by the computational complexity of the problem. Figure 1 shows the final architecture, with a multi-task learning objective trained with a weighted combined loss.

Generative Setting - In the generative setting we pre-train two language models as described below. In both the models we continue the training of the

[6]https://huggingface.co/roberta-large

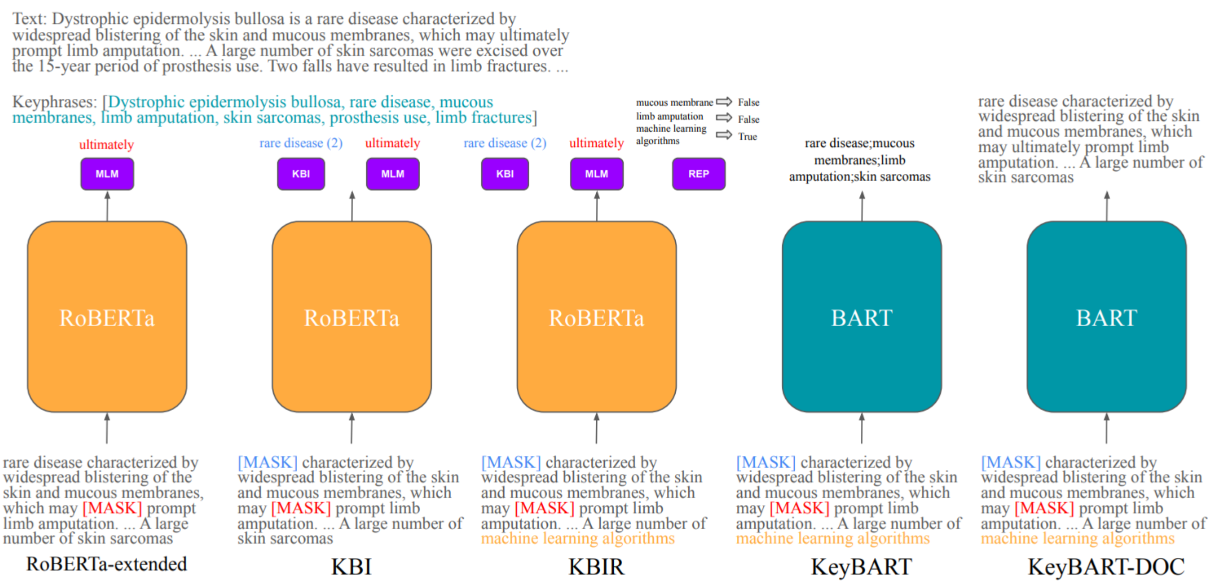
Text: Dystrophic epidermolysis bullosa is a rare disease characterized by widespread blistering of the skin and mucous membranes, which may ultimately prompt limb amputation. ... A large number of skin sarcomas were excised over the 15-year period of prosthesis use. Two falls have resulted in limb fractures. ...
Keyphrases: [Dystrophic epidermolysis bullosa, rare disease, mucous membranes, limb amputation, skin sarcomas, prosthesis use, limb fractures]
ultimately
MLM
RoBERTa
rare disease characterized by widespread blistering of the skin and mucous membranes, which may [MASK] prompt limb amputation. ... A large number of skin sarcomas
RoBERTa-extended
rare disease (2)
ultimately
KBI
MLM
RoBERTa
[MASK] characterized by widespread blistering of the skin and mucous membranes, which may [MASK] prompt limb amputation. ... A large number of skin sarcomas
KBI
rare disease (2)
ultimately
mucous membrane ⇨ False
limb amputation ⇨ False
machine learning algorithms ⇨ True
KBI
MLM
REP
RoBERTa
[MASK] characterized by widespread blistering of the skin and mucous membranes, which may [MASK] prompt limb amputation. ... A large number of machine learning algorithms
KBIR
rare disease;mucous membranes;limb amputation;skin sarcomas
BART
[MASK] characterized by widespread blistering of the skin and mucous membranes, which may [MASK] prompt limb amputation. ... A large number of machine learning algorithms
KeyBART
rare disease characterized by widespread blistering of the skin and mucous membranes, which may ultimately prompt limb amputation. ... A large number of skin sarcomas
BART
[MASK] characterized by widespread blistering of the skin and mucous membranes, which may [MASK] prompt limb amputation. ... A large number of machine learning algorithms
KeyBART-DOC

Figure 2: Pre-training Strategies, the keyphrase present in the text are highlighted in teal and every perturbation in the form of a random MASK is represented in red, keyphrase MASK is represented in blue and Keyphrase Replacement is represented in orange.

weights of BART-large[7] on our corpus using our pre-training strategies.

- **KeyBART** - We perform token masking, keyphrase masking and keyphrase replacement with same masking hyperparameters as KBIR on the input text and pre-train the model to predict the original keyphrases in Catseq format following the setup explained in Section 3.
- **KeyBART-DOC** - This setup uses the same input denoising settings as KeyBART, with the only difference in the output, where KeyBART generates the keyphrases associated with the document in Catseq format, whereas KeyBART-DOC similar to BART generates the original denoised input.

We use the exact same data in all the pre-training setups as explained above. We increase the number of steps while decreasing the batch size such that all the models see the data the same number of times (i.e., 2 epochs). *The batch size is only reduced to accommodate increases in memory usage in the model pre-training*.

[7]https://huggingface.co/facebook/bart-large

Input Text: On The Complexity of Combinatorial Auctions : Structured Item Graphs and Hypertree Decompositions. The winner determination problem in combinatorial auctions is the problem of determining the allocation of the items among the bidders that maximizes the sum of the accepted bid prices. While this problem is in general NPhard, it is known to be feasible in polynomial time on those instances whose associated item graphs have bounded treewidth called structured item graphs. Formally, an item graph is a graph whose nodes are in one-to-one correspondence with items, and edges are such that for any bid, the items occurring in it induce a connected subgraph. Note that many item graphs might be associated with a given combinatorial auction, depending on the edges selected for guaranteeing the connectedness. In fact, the tractability of determining whether a structured item graph of a fixed treewidth exists and if so, computing one was left as a crucial open problem. In this paper, we solve this problem by proving that the existence of a structured item graph is computationally intractable, even for treewidth 3. Motivated by this bad news, we investigate different kinds of structural requirements that can be used to isolate tractable classes of combinatorial auctions. We show that the notion of hypertree decomposition, a recently introduced measure of hypergraph cyclicity, turns out to be most useful here. Indeed, we show that the winner determination problem is solvable in polynomial time on instances whose bidder interactions can be represented with dual hypergraphs having bounded hypertree width. Even more surprisingly, we show that the class of tractable instances identified by means of our approach properly contains the class of instances having a structured item graph.
Extracted Keyphrases: [combinatorial auctions]; [structured item graphs]; [hypertree decompositions]; [item graphs]; [treewidth]; [bidders]; [hose nodes]; [hypertree cyclicity]; [polynomial time]
Generated Keyphrases: [combinatorial auctions]; [structured item graphs]; [treewidth]; [hypergraphs]; [hypertree decompositions]
Ground Truth: [hypergraph]; [structured item graph]; [polynomial time]; [combinatorial auction]; [fixed treewidth]; [accepted bid price]; [hypertree decomposition]; structured item graph complexity; simplification of the primal graph; hypertree based decomposition method; hypergraph hg; the primal graph simplification; well known mechanism for resource and task allocation; complexity of structured item graph;
Input Text: Interesting Nuggets and Their Impact on Definitional Question Answering. Current approaches to identifying definitional sentences in the context of Question Answering mainly involve the use of linguistic or syntactic patterns to identify informative nuggets. This is insufficient as they do not address the novelty factor that a definitional nugget must also possess. This paper proposes to address the deficiency by building a Human Interest Model from external knowledge. It is hoped that such a model will allow the computation of human interest in the sentence with respect to the topic. We compare and contrast our model with current definitional question answering models to show that interestingness plays an important factor in definitional question answering.
Extracted Keyphrases: [interesting nuggets]; [definitional sentences]; [question answering]; [nuggets]; [novelty]; [definitional nuggets]; [human interest model]; [human interest]
Generated Keyphrases: [definitional question answering]; natural language processing
Ground Truth: [human interest]; [use of linguistic]; [interesting nugget]; [definitional question answer]; [informative nugget]; [interest]; [computation of human interest]; sentence fragment; unique quality; manual labor; news corpus; baseline system; human interest computation; human reader; linguistic use; question topic; external knowledge; surprise factor; lexical pattern
Input Text: The Influence of Caption Features on Clickthrough Patterns in Web Search. Web search engines present lists of captions, comprising title, snippet, and URL, to help users decide which search results to visit. Understanding the influence of features of these captions on Web search behavior may help validate algorithms and guidelines for their improved generation. In this paper we develop a methodology to use clickthrough logs from a commercial search engine to study user behavior when interacting with search result captions. The findings of our study suggest that relatively simple caption features such as the presence of all terms query terms, the readability of the snippet, and the length of the URL shown in the caption, can significantly influence users ’ Web search behavior.
Extracted Keyphrases: [influence]; [caption features]; [clickthrough patterns]; [web search]; [snippet]; [methodology]; [clickthrough logs]
Generated Keyphrases: [web search]; [clickthrough]; [captions]; user study
Ground Truth: [clickthrough pattern]; [snippet]; [web search behavior]; [web search]; [caption feature]; summarization; extractive summarization; significant word; query log; human factor; clickthrough inversion; query term match; query re formulation

Table 10: Sample keyphrases extracted by KBI-REP-CRF and generated by KeyBART on the SemEval-2010 dataset. Present keyphrases are marked with square brackets.