Deep Seq2Seq Keyphrase Generation: Model Calibration and Uncertainty

Anonymous ACL submission

Abstract

Keyphrase generation aims to generate topical phrases from a given text either by copying from the original text (present keyphrases) or by producing new keyphrases (absent keyphrases) that capture the topical and salient aspects of the text. While many neural models have been proposed and analyzed for this task, there is limited analysis of the properties of their generative distributions at the decoding stage. Particularly, it remains to be known how well-calibrated or uncertain the confidence of different models is with empirical success rate and whether they can express their uncertainty. Here, we study the confidence scores, perplexity, and expected calibration errors of five strong keyphrase generation models with unique characteristics and designs based on seq2seq recurrent neural networks (ExHiRD), transformers with no pre-training (Transformer, Trans2Set), and transformers with pre-training (BART, and T5). We propose a novel strategy for keyphrase-level perplexity calculation and for normalizing sub-word-level perplexity to gauge model confidence.

1 Introduction

Keyphrase generation is the task of predicting a set of keyphrases from a given document that capture the core ideas and topics of the document. Among these keyphrases, some exist within the source document (present keyphrases), and some are absent from the document (absent keyphrases). Keyphrases are widely used in various applications, such as document indexing and retrieval (Jones and Staveley, 1999; Boudin et al., 2020), document clustering (Hulth and Megyesi, 2006), topic classification (Sadat and Caragea, 2022), and text summarization (Wang and Cardie, 2013; Abu-Jbara and Radev, 2011). Hence, keyphrase generation is of great interest to the scientific community.

In recent years, neural encoder-decoder (seq2seq) models have been adapted to generate both absent and present keyphrases (Meng et al., 2017). These approaches (Yuan et al., 2020; Chan et al., 2019a; Chen et al., 2020) to keyphrase generation aim at autoregressively decoding a sequence of concatenated keyphrases from a given source document. Typically, these models are equipped with cross-attention (Luong et al., 2015; Bahdanau et al., 2015) and a copy (or pointer) mechanism (Gu et al., 2016; See et al., 2017). Another emerging trend is to adapt pre-trained language models for keyphrase generation (Liu et al., 2020; Wu et al., 2021; Garg et al., 2022; Ray Chowdhury et al., 2022; Kulkarni et al., 2021; Madaan et al., 2022; Wu et al., 2022b,a; Kulkarni et al., 2022). However, although a number of such variants and extensions of seq2seq models have been proposed to enhance keyphrase generation, there have been limited attempts at analyzing the predictive distribution of neural seq2seq models in this task. Particularly, we are interested here in taking a closer look at the decoder of seq2seq models to understand model calibration and evaluate uncertainty estimation (Guo et al., 2017) of keyphrase predictions.

Model Calibration and Uncertainty: In practical applications, it is often desirable to accurately estimate *the confidence of a model prediction* to decide whether that prediction can be used or not (Guo et al., 2017; Rybkin et al., 2021; Zhao et al., 2021; Tian et al., 2023). Thus, the models must not only be accurate, but also must indicate when they are likely to get a wrong prediction (reflected in the model's confidence or uncertainty). This allows the decision-making to be routed as needed to a human or another more accurate, but possibly more expensive, model. Similarly, in keyphrase generation, in principle, calibrated model confidence could be used to make different decisions - for example, ranking keyphrases after *overgeneration*, or mixing predictions of different models based on their confidence, or even switching control to an expert for annotation. However, before we can rely on the confidence estimated by a model (based on its prediction probabilities), we need to determine how well calibrated the model is. A well-calibrated model should generally "know what it does not know", which can be reflected by a strong alignment between its empirical likelihood (accuracy) and its probability estimates (confidence). Thus, in this work, we measure and contrast calibration and performance of five key models for keyphrase generation: (1) ExHiRD; (2) Transformer; (3) Trans2Set; (4) BART; and (5) T5. Moreover, to be able to measure confidence calibration and uncertainty at the level of keyphrases, we propose a novel perplexity-based measure called *Keyphrase Perplexity* (KPP) which we use to analyze a model's own estimated confidence.

Overall, our contributions are as follows:

1. We introduce *keyphrase perplexity* (KPP) metric to gauge model confidence. Using KPP, we analyze the prediction confidence of multiple seq2seq models.
2. We explore the models' calibration for keyphrase generation to study confidence versus generation performance for five seq2seq models and evaluate their performance on standard F1-score and expected calibration error (ECE) using four benchmark datasets.
3. We examine the variance of model performance with that of the position of extracted present keyphrases in the source document.

2 Related Work

Keyphrase Generation: The current focus of research on keyphrase generation has been increasingly shifting towards seq2seq models particularly because of their capability to generate absent keyphrases (Meng et al., 2017). Multiple works built upon seq2seq architectures to address keyphrase generation (Meng et al., 2017; Chen et al., 2018; Chan et al., 2019a,b; Swaminathan et al., 2020; Chen et al., 2020; Ye et al., 2021b,a; Huang et al., 2021) (inter alia). Some recent works also explored the inclusion of pre-trained models for both absent and present keyphrase generation (Liu et al., 2020; Wu et al., 2021; Kulkarni et al., 2021; Wu et al., 2022b,a; Garg et al., 2022; Ray Chowdhury et al., 2022; Madaan et al., 2022; Wu et al., 2023). Our focus, however, is more on the analysis and evaluation rather than the development of a new architecture. In terms of analysis, Meng et al. (2021) showed the effects of different hyperparameters including the ordering format for concatenating target keyphrases on the task. Boudin et al. (2020) and Boudin and Gallina (2021) analyzed the contribution of present key phrases and different types of absent keyphrases for document retrieval. Do et al. (2023) and Shen et al. (2022) investigated unsupervised open-domain keyphrase generation using a transformer based seq2seq model to avoid human-supervision. Garg et al. (2022) analyzed additional information, e.g., an extractive summary of a document or citation sentences from its content, rather than simply using title and abstract for keyphrase generation. Garg et al. (2023) explored the impact of data augmentation strategies for keyphrase generation in resource-constrained domains. Meng et al. (2023) proposed a framework for keyphrase generation for domain adaptation. Keyphrase generation has also been studied in other works such as Liu et al. (2024); Zhang et al. (2022); Zhao et al. (2022); Chen and Iwaihara (2024); Choi et al. (2023).

Model Calibration: Calibration and uncertainty of modern deep neural models (Guo et al., 2017) have started to gain attention on several natural language processing tasks, including neural machine translation (Müller et al., 2019; Kumar and Sarawagi, 2019; Wang et al., 2020), natural language understanding (Park and Caragea, 2022a,b; Desai and Durrett, 2020), coreference resolution (Nguyen and O'Connor, 2015), and summarization Xu et al. (2020). For example, Wang et al. (2020) focused on the calibration of neural machine translation (NMT) models to understand the generative capability of the models at inference (decoding time) under the *exposure bias* (Ranzato et al., 2016), i.e., the discrepancy between training and inference due to teacher forcing in the training of auto-regressive models. Other recent studies on the calibration of pre-trained language models include (Chen et al., 2023; Zhu et al., 2023; Tian et al., 2023; Jiang et al., 2023). Chen et al. (2023) and Zhu et al. (2023) showcased studies on how pre-training and training affect the calibration of language models. Tian et al. (2023) explored calibration of recent language models pre-trained with reinforcement learning with human feedback (RLHF) pre-training objective. Jiang et al. (2023) proposed generative calibration with in-context predictive distributions adjusted by label marginal.

3 Our Models

For our analysis, we consider five models: (1) ExHiRD (Chen et al., 2020); (2) Transformer (Ye et al., 2021b); (3) Trans2Set (Ye et al., 2021b); (4) BART (Lewis et al., 2020); and (5) T5 (Raffel et al., 2020). We chose ExHiRD because it is one of the strongest performing Recurrent Neural Network-based keyphrase generation architectures without relying on reinforcement learning or GANs. We chose Transformer (Ye et al., 2021b) to show the effect of simply using a Transformer-based architecture over a specialized RNN-based one when both have no pre-training. We chose Transformer One2Set because it is one of the strongest performing Transformer-based architecture with no pre-training (Ye et al., 2021b). We chose T5 and BART because they are starting to become foundation models (Bommasani et al., 2021) for keyphrase generation with pre-training (Kulkarni et al., 2021; Ray Chowdhury et al., 2022; Wu et al., 2022b; Madaan et al., 2022; Wu et al., 2022a).

ExHiRD: ExHiRD (Chen et al., 2020) is an RNN-based seq2seq model with attention and copy-mechanism. It uses a hierarchical decoding strategy to address the hierarchical nature of a sequence of keyphrases, where each keyphrase is, in turn, a sub-sequence of words. ExHiRD also proposes exclusion mechanisms to improve the diversity of keyphrases generated and reduce duplication.

Transformer: Transformer One2Seq is simply the vanilla Transformer model (Vaswani et al., 2017) without prior pre-training that is trained on keyphrase generation in the seq2seq paradigm—the target sequence is a concatenation of keyphrases with some delimiter (Yuan et al., 2020). We use the same settings as Ye et al. (2021b).

Trans2Set: Transformer One2Set is similar to Transformer One2Seq but trains the Transformer model in a One2Set paradigm (Ye et al., 2021b) and does not require any prior pre-training. In this paradigm, the decoder uses a constant number of trainable embeddings as "control codes" to condition cross-attention to generate a single keyphrase (or alternatively some null token) per control code. In other words, keyphrases in Trans2Set are generated simultaneously without any influence of order—the generation of one keyphrase is not dependent on some generation of earlier keyphrase like in the seq2seq paradigm. We use the same settings for the model as Ye et al. (2021b).

T5: T5 (Raffel et al., 2020) is a large-scale pre-trained encoder-decoder Transformer-based model pre-trained on the C4 dataset, which was introduced in the paper along with T5. T5 is pre-trained using the BERT-style masked language modeling (MLM) objective and deshuffling. MLM objective in T5 includes spans of text are corrupted and masked using a single sentinel token whereas deshuffling consists of shuffling the input sequence in random order and trying to predict the original text. We use the `t5-base` model from the Transformers library (Wolf et al., 2020).

BART: BART (Lewis et al., 2020) is a large-scale pre-trained encoder-decoder Transformer-based model. BART has been pre-trained as a denoising autoencoder for seq2seq tasks with a bidirectional encoder similar to BERT (Devlin et al., 2019) and a GPT (Radford et al., 2018)-like autoregressive decoder. BART achieved state-of-the-art results over abstractive dialogue, summarization and question answering at the time of its release. Pre-training data used for BART is the same as for RoBERTa (Liu et al., 2019). We use the `BART-large` model in similar settings as Wu et al. (2022a). BART is fine-tuned similar to how Transformer is trained.

We provide further implementation details for our five models in Appendix B.

4 Model Calibration and Uncertainty

As we discussed before, it is important to check how well calibrated a given model is to determine how trustworthy and reliable the model is. In this section, we first present our novel *Keyphrase perplexity* (KPP) metric, to estimate a model's confidence at the level of keyphrases and then we describe how we use KPP to estimate calibration and uncertainty.

4.1 Keyphrase Perplexity

We propose *Keyphrase Perplexity* (KPP) to gauge model confidence on a particular predicted keyphrase. KPP is rooted in the general concept of perplexity, which is a widely used metric for evaluating language models. For a sequence of tokens $w_{1:n} = w_1, w_2, ..., w_n$ of length n, perplexity is the inverse normalized probability p of generating them and can be defined as: $PP(w_{1:n}) = p(w_1, w_2, ..., w_n)^{-1/n}$. For an auto-regressive decoder, the probability p of the sequence can be factorized and reformulated as:

$$PP(w_{1:n}) = \left(\prod_{i=1}^{n} p(w_i|w_1, w_2, \ldots w_{i-1}) \right)^{-1/n} \quad (1)$$

However, note that in the widely used seq2seq framework (Yuan et al., 2020), a generated/decoded sequence is a concatenation of keyphrases. The vanilla perplexity is only defined over the whole generated sequence and cannot be directly applied for subsequences (keyphrases) within the sequence. Thus, to get an estimate of the model confidence at the level of predicting individual keyphrases, we adapt the original perplexity and define keyphrase perplexity (KPP) as follows. Given a particular keyphrase represented as the sub-sequence $w_{j:k} = w_j, w_{j+1}, ..., w_k$ within the sequence $w_{1:n}$ $(1 \leq j \leq k \leq n)$ (representing a sequence of concatenated keyphrases), the KPP of that keyphrase ($w_{j:k}$) is defined as:

$$KPP(w_{j:k}) = \left(\prod_{i=j}^{k} p(w_i|w_1, w_2, \ldots w_{i-1}) \right)^{-1/m} \quad (2)$$

where $m = k - j + 1$ is the number of tokens in the keyphrase $w_{j:k}$. Essentially, for KPP, we simply use the conditional probabilities of tokens within the keyphrase $w_{j:k}$ under consideration. One limitation of this KPP formulation is that it does not negate the conditioning effect of previous keyphrases (included in sub-sequence w_1 to w_{j-1} while measuring the KPP of the keyphrase starting from w_j). However, removing this limitation is not straight-forward; so we take a naive assumption that the delimiters guide the overall probabilities of keyphrases to be independent of the earlier keyphrases. As such, our formulation is a form of "quasi-perplexity" measure. During our analysis, any probability of the form $p(w_i|w_1, w_2, \ldots w_{i-1})$ indicates the predicted model probability for token w_i given that tokens $w_1, w_2, \ldots w_{i-1}$ have been already generated. We do not consider special tokens (e.g., keyphrase delimiters or end of sequence markers) as part of any keyphrase subsequence for KPP. As in perplexity, a lower KPP indicates a higher confidence in the prediction, whereas a higher KPP indicates a lower confidence.

Trans2Set KPP In case of One2Set models (Transformer One2Set), we will get some k *independent* keyphrase span predictions in a set where none of the keyphrase prediction is autoregressively dependent on earlier or latter keyphrases. This is analogous to running a separate decoder for generating each keyphrase. In this case, we apply KPP individually to each keyphrase (span of words) in the set. In other words, w_1 in Equation 2 represents the special start token, and w_2 represents the first token of the keyphrase whose KPP is being calculated.

$$KPP(w_{1:k}) = \left(\prod_{i=1}^{k} p(w_i|w_1, w_2, \ldots w_{i-1}) \right)^{-1/m} \quad (3)$$

Subword to Word-level KPP One problem with our KPP formulation is that the non-pretrained models (ExHiRD, Transformer One2Seq, Transformer One2Set) are using word-level tokenization whereas the pre-trained models (T5, BART) are using subword-level tokenization. The prediction subwords are generally easier than whole words, and confidence per subwords can be generally higher - which can lead to an inherent bias towards lower perplexity/higher confidence simply as an artifact of tokenization choice. As an example, consider a prediction probability of a word 'geothermal' as $p(\text{geothermal}) = 0.5$. KPP of the word would be:

$$p(\text{geothermal})^{-1/1} = 2 \quad (4)$$

For the same word, a subtokenization could be a sequence ('geo','thermal'). Let us say the predicted probabilities are $p(\text{geo}) = 0.625$ and $p(\text{thermal}|\text{geo}) = 0.8$. In this case the overall word-level probability is the same:

$$p(\text{geothermal}) = p(\text{geo}) \cdot p(\text{thermal}|\text{geo}) = 0.5 \quad (5)$$

However now KPP(('geo','thermal')) is:

$$(p(\text{geo}) \cdot p(\text{thermal}|\text{geo}))^{-1/2} = 1.41 \quad (6)$$

Thus, the subword tokenization will have an inherent bias towards lower perplexity/higher confidence because of the difference in length normalization based on token numbers despite having the same probabilities at the word-level.

Given these circumstances, to level the playing field for comparison, we use a different KPP metric (KPP-s) for T5 and BART where we normalize the keyphrase subsequence based on number of *words* rather than the number of (subword) tokens. For a subsequence of tokens $w_{j:k}$, this can be expressed as:

$$KPP\text{-s}(w_{j:k}) = \left(\prod_{i=j}^{k} p(w_i|w_1, \dots w_{i-1})\right)^{\frac{-1}{wc(w_{j:k})}} \quad (7)$$

Here, $wc(w_{j:k})$ returns the number of words[1] in the subsequence of (subword) tokens $w_{j:k}$. Thus, for our example above, KPP from Eq. 6 becomes:

$$(p(\text{geo}) \cdot p(\text{thermal}|\text{geo}))^{-1/1} = 2 \quad (8)$$

which is the same as the KPP for the word 'geothermal' from Eq. 4.

Henceforth, we simply use KPP to refer to both KPP and KPP-s - we simply apply the former for word-level tokenization models (ExHiRD, Transformer, Trans2Set), and the latter for subword-level tokenization models (T5, BART).

4.2 Calibration

Model calibration reflects the accuracy of model predictions as a function of its generated posterior probabilities. A calibrated model has alignment between its empirical likelihood (accuracy) and its probability estimates (confidence). For example, a calibrated model that has a confidence of 90% while making predictions, would correctly predict 90 out of 100 possible samples. Formally, calibration models the joint distribution $P(Q, Y)$ over generated model probabilities $Q \in \mathbb{R}$ and labels Y. $P(Y = y|Q = q) = q$ signifies perfect calibration of a model (Guo et al., 2017).

Expected calibration error (ECE) is a popular measure of model miscalibration (Naeini et al., 2015). ECE is computed by partitioning the predictions according to their confidence estimates into k bins (we set k=10) and summing up the weighted average of the absolute value of the difference between the accuracy and the average confidence of keyphrases in each bin. This can be formalized as:

$$ECE = \sum_{i=1}^{k} \frac{|B_i|}{n} |acc(B_i) - confid(B_i)| \quad (9)$$

Here n is the number of total samples, $|B_i|$ is the number of samples in bin B_i, $1 \leq i \leq k$, of k bins. In our task, we compute $acc(B_i)$ as the fraction of accurately predicted keyphrases in bin B_i and $confid(B_i)$ as the average confidence in bin B_i. We define confidence of a particular generated keyphrase as the inverse of its KPP (KPP^{-1}) that is, roughly, the length normalized product of posterior probabilities for the tokens of that keyphrase.

In addition to ECE, reliability diagrams depict the accuracy of the model as a function of the probability across the k bins.

5 Experiments and Results

We share hyperparameter details in Appendix A.

5.1 Datasets

We select four widely used benchmarks for our experimentation: **KP20k** (Meng et al., 2017), **Krapivin** (Krapivin et al., 2009), **Inspec** (Hulth, 2003) and **SemEval** (Kim et al., 2010). We use the KP20k training set (~500,000 samples) to train our models. As test sets, we use the test sets available for each dataset for performance evaluation and analysis. The test sets of KP20k, Inspec, Krapivin, and SemEval have ~20,000, 500, 400, and 100 documents, respectively. All datasets have annotated present and absent keyphrases.

5.2 Models' Performance

We compare the results of our models using standard F_1 metrics (F_1@5 and F_1@M), similar to Chen et al. (2020), in Table 1 after training them on KP20k. For F_1 evaluation, we used similar post-processing as Chen et al. (2020). We share more concrete details in Appendix B. Interestingly, we find trained-from-scratch models (ExHiRD, Trans2Set, Transformers) to perform competitively or outperform pre-trained language models (PLMs) like T5/BART in several datasets, with Trans2Set generally coming out on top. This shows that domain-general pre-training may not be as effective for keyphrase generation. Similar results are also noted in (Ray Chowdhury et al., 2022; Wu et al., 2023, 2022a). However, there could be better ways to utilize PLMs by adapting them in a trans2set framework (Madaan et al., 2022).

PLMs can also be augmented by new decoding strategies (Wu et al., 2023; Zhao et al., 2022), reranking (Choi et al., 2023), task-specific training (Kulkarni et al., 2022; Wu et al., 2022a), or data-augmentation (Ray Chowdhury et al., 2022; Chen and Iwaihara, 2024) among others.

[1]The words can be counted by first turning the subword tokens into a string based on the respective tokenizer implementations for T5 and BART in Huggingface (Wolf et al., 2020) and then using space tokenizer for word tokenization. The length of the list of the tokenized words will then be the return value of $wc(w_{j:k})$.

Models	Inspec		Krapivin		SemEval		KP20k	
	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5
Present Keyphrase								
ExHiRD	0.291	0.253	0.347	0.286	0.335	0.284	0.374	0.311
Transformer	0.325	0.281	0.315	0.365	0.287	0.325	0.392	0.332
Trans2Set	0.324	0.285	0.364	0.326	0.357	0.331	0.392	0.358
BART	0.323	0.270	0.336	0.270	0.321	0.271	0.388	0.322
T5	0.340	0.287	0.328	0.271	0.306	0.275	0.387	0.335
Absent Keyphrase								
ExHiRD	0.022	0.011	0.043	0.022	0.025	0.017	0.032	0.016
Transformer	0.019	0.010	0.060	0.032	0.023	0.020	0.046	0.023
Trans2Set	0.034	0.021	0.073	0.047	0.034	0.026	0.058	0.036
BART	0.017	0.010	0.049	0.028	0.021	0.016	0.042	0.022
T5	0.025	0.014	0.053	0.028	0.023	0.016	0.036	0.018

Table 1: Keyphrase generation performance for different models. Transformer represents Transformer One2Seq; Trans2Set represents Transformer One2Set.

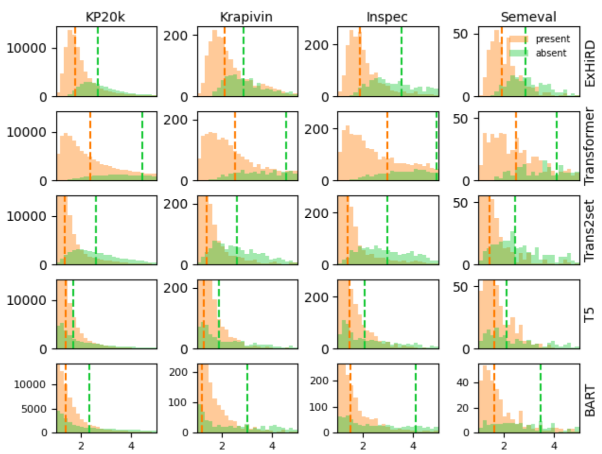

Figure 1: Histograms with Y axis depicting number of keyphrases and X axis indicating keyphrase perplexity values for both present and absent keyphrase generation. Dashed lines indicate the median of each distribution.

5.3 Keyphrase Perplexity Analysis

We compare keyphrase perplexities (KPP) of all the models - ExHiRD, Transformer, Trans2Set, T5, and BART (using histograms) in Figure 1. In this experiment, we compute the KPP of each keyphrase generated by the model and plot the number of present and absent keyphrases generated by each model on every dataset. We analyze the plots generated across a range of intervals. Unsurprisingly, we find that all models have lower KPP (thus, higher confidence) for present keyphrases than absent keyphrases (which are harder to learn to generate). However, T5, ExHiRD, and Trans2Set appear to be substantially more confident about their absent keyphrase predictions compared to others. There is also a degree of randomness in the KPP values generated for absent keyphrase distributions as the middle 80% is spread across the x-axis for all the models, which suggests higher entropy in the probability distributions generated by the model. The majority of the KPP values generated for present keyphrases are skewed towards the intervals between 0 and 2 in figure 1, showcasing the high degree of confidence the models have for extractive keyphrase generation. The results showcase how models generate predictive distributions and help us understand the weaknesses in the language models to work towards improving them.

In Figure 2, we show that the *conditional probabilities of tokens in a keyphrase* tend to be low at the boundaries (at the beginning of a keyphrase) but start to increase monotonically as the decoder moves towards the end of the keyphrase (with the exception of BART and T5 where the increment is not purely monotonic). Intuitively, it makes sense that a model will have less confidence predicting the start of a keyphrase because it requires settling on a specific keyphrase to generate out of many potential candidates. However, the first keyphrase token, once already generated, will condition and restrict the space of plausible candidates for the second token, thereby increasing its confidence. For the same reason, the probabilities of the second keyphrase token and later tend to be much higher. This shows how critically important generating the first token of a keyphrase is in terms of generative language models. The high entropy while generating the first token shows the fine margins in terms of how language models are generating incorrect predictions.

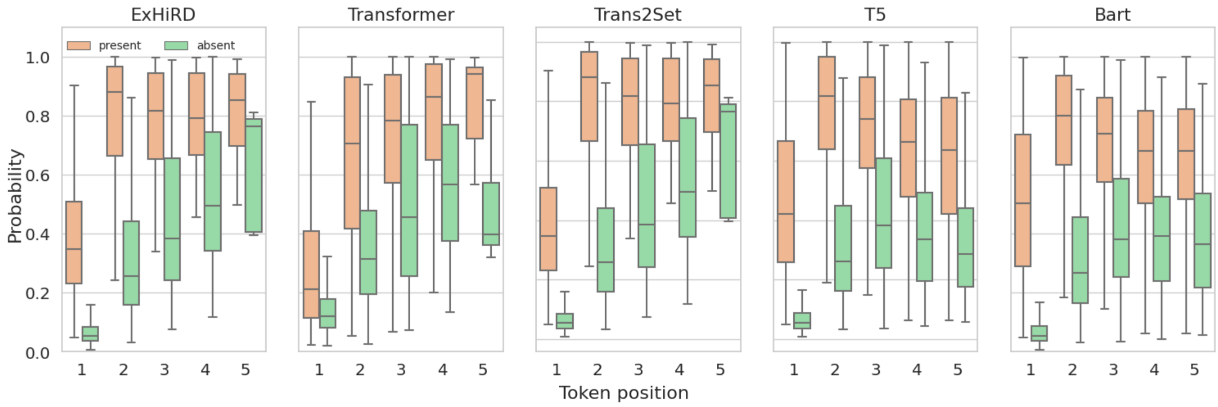

Figure 2: ExHiRD, Transformer, Trans2Set, T5, and BART's conditional probabilities for the first five word-level tokens of the keyphrases generated in a sequence on the KP20K test set.

Models	KP20k	Inspec	Krapivin	Semeval
Present keyphrases				
ExHiRD	19.06	15.66	11.69	17.51
Transformer	23.41	20.01	22.96	20.49
Trans2Set	25.46	26.38	24.38	22.44
BART	29.93	21.85	51.85	16.10
T5	33.06	23.04	61.78	18.75
Absent keyphrases				
ExHiRD	17.72	11.87	15.55	15.83
Transformer	89.86	78.65	75.16	85.12
Trans2Set	47.34	37.57	43.69	47.22
BART	8.18	11.14	9.27	11.30
T5	16.19	15.76	13.44	18.65

Table 2: Expected calibration error(ECE) (lower the better) for ExHiRD, Transformer, Trans2set, T5 and BART on various datasets.

5.4 Model Calibration Analysis

We saw that T5, Trans2Set, and ExHiRD generally predict keyphrases with higher confidence (lower KPP). But does the higher confidence actually translate into better predictions? Figure 3 shows the reliability diagrams of all the models. Here, we generate the probability values at the keyphrase level, computing them from the token probabilities. We map the keyphrase probabilities or confidence to the accuracy of correct predictions across various intervals. Interestingly, we can see that the calibration of ExHiRD or Transformer is better than the other models. T5's high-confidence keyphrase predictions do not translate into optimal accuracy values. In Table 2, we show the Expected Calibration Error (ECE) for the different models in our consideration across various datasets. Consistent with the reliability diagrams, here, we find that T5's ECE is much higher than ExHiRD. ExHiRD, in fact, achieves the best ECE. Transformer is generally lower in ECE compared to other models besides ExHiRD. The other three models - Trans2Set, T5, BART (despite having strong F1 performances) are on the higher end of ECE. We also observe in figure 3 that simpler models such as ExHiRD and Transformer models are better calibrated than models with specific decoding techniques (Trans2Set) and pre-training (T5 and Bart). Pre-trained models have allowed us to perform zero-shot or few-shot learning due to the retention of vast amounts of information. But the pre-training also introduces high entropy within the models which translates into the variability in predictive distributions as seen in Figure 3 and Table 2.

5.5 Robustness to Positional Variance

We analyze all models' present keyphrase predictions with respect to their position in the input text. First, we look at the distribution of gold present keyphrases in the input text. We divide the input text into five sections with 20% of characters in each, and bin the keyphrases appearing in each section accordingly. We compute the numbers of present keyphrases in each section in the source text for all the datasets and show them in Table 3. As we can see, the majority of gold present keyphrases are in the first section (bin) of the input sequence. In Figure 4, we compare the accuracy of our five models for present keyphrases in different sections of the text. We notice that all models have a similar accuracy at identifying keyphrases from the first section of the input, and they progressively fail to identify keyphrases in the later sections of the input text. Interestingly, T5 and BART not only perform well in identifying keyphrases present in the initial sections of the text, but they also perform better than the other models in predicting keyphrases from the later sections (bins). This pattern is par-

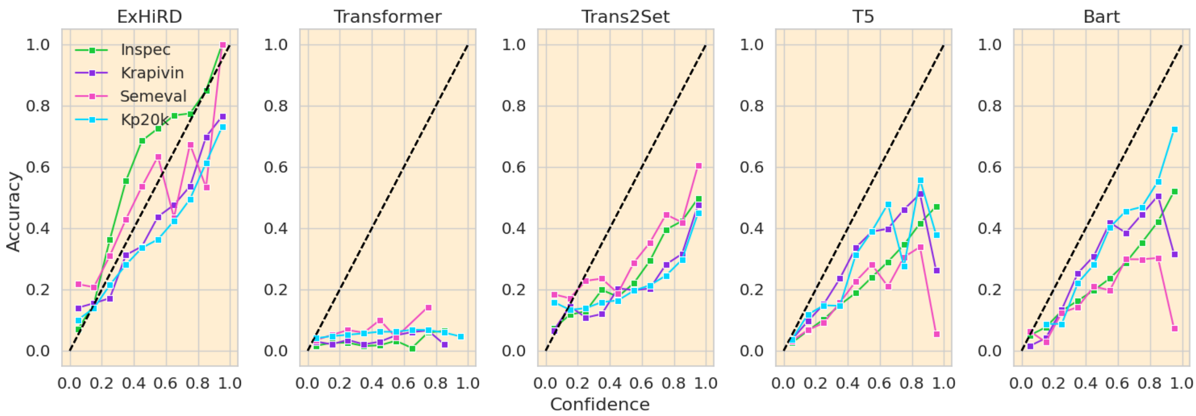

Figure 3: Reliability diagrams for model calibration of ExHiRD, Transformer, Trans2set, T5 and Bart respectively. Dotted black line depicts perfectly calibrated model.

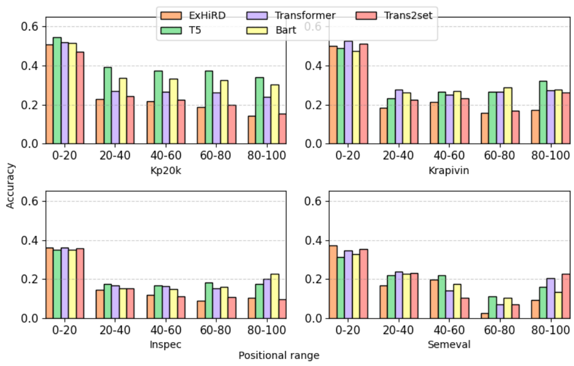

Figure 4: Accuracy of present keyphrase generation with respect to their position in the original text for our five models. The x-axis denotes the percent characters of the source text where present keyphrases are located.

Dataset	Positional range				
	0-20	20-40	40-60	60-80	80-100
Inspec	1,326	845	686	602	173
Krapivin	706	206	182	159	59
SemEval	346	126	103	54	20
KP20k	39,571	9,865	8,313	6,317	1,704

Table 3: Number of keyphrases present keyphrases in gold labels binned into five sections, each having 20% characters of the source document.

ticularly prominent on KP20k. The bias towards predicting earlier present keyphrases is, most likely, further compounded by the fact that the present keyphrases are ordered according to their position of first occurrence within the target sequence.

As such the models can be biased to be good at only predicting keyphrases that occur early in the source text. However, a potential main reason for the bias is simply that the majority of keyphrases exist in the earlier segments of a document as shown in Table 3. Nevertheless, T5 and BART appear more resistant to these biases, despite being exposed to the same data and similarly ordered target sequences. These results hint also to a "better understanding" of the overall semantics of the document by the T5 and BART models, and hence, their improved generation of short phrase document summaries (i.e., keyphrases).

6 Conclusion

Here, we discuss our main findings and motivate their use for future work. First, we find that the model confidences of absent keyphrase predictions are much lower than present keyphrase predictions for both models. Thus, the models know to be more uncertain with absent keyphrase generation (for which all models indeed have poor performance). However, upon checking for model calibrations, interestingly, we find that pre-trained Transformer models are more overconfident (poorly calibrated) compared to RNN (ExHiRD) and non-pre-trained transformer models.

Second, we find that the models are much less confident in predicting the starting words of a keyphrase. We believe deciding on the start of the keyphrase is much harder than predicting the follow-up tokens. Based on this finding, we may be able to make more efficient semi-autoregressive models that sequentially decode different keyphrases but simultaneously decode different tokens within a particular keyphrase.

Third, pre-trained models are poorly calibrated for the keyphrase generation task even though they have been trained on a large corpus of text. RNN and transformer models that have not been pre-trained are better calibrated. Better calibrated models are less erroneous when model confidence is high while generating keyphrases. Thus, there is potential for further work on models' calibration.

7 Limitations

Our analysis showcases key parameters of comparison between models in terms of KPP and calibration measures for the keyphrase generation task. This provides insights into intrinsic model behavior while generating keyphrases. As we discussed before, one limitation of our KPP measure as used in the study is that in a Transformer framework, it is difficult to negate the effect of previously generated keyphrases. However, the keyphrase delimiters may naturally, to an extent, reduce the effect of previous keyphrases. Thus, it still can be decent heuristics. Note that Non-exact (quasi-)perplexity measures (in different formulations) have been also proposed in other contexts (Wang et al., 2019) before.

8 Ethics Statement

We analyze various aspects of the keyphrase generation task. Keyphrase generation is a popular and established NLP task that is useful in information extraction. We do not forsee any ethical concern regarding our contribution to this domain

References

Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *ACL*, pages 500–509. ACL.

Rohan Anil, Vineet Gupta, Tomer Koren, and Yoram Singer. 2019. Memory efficient adaptive optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 9749–9758. Curran Associates, Inc.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*.

Florian Boudin and Ygor Gallina. 2021. Redefining absent keyphrases and their effect on retrieval effectiveness. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4185–4193, Online. Association for Computational Linguistics.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase generation for scientific document retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1126.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019a. 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.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019b. 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.

Bin Chen and Mizuho Iwaihara. 2024. Enhancing keyphrase generation by bart finetuning with splitting and shuffling. In *PRICAI 2023: Trends in Artificial Intelligence*, pages 305–310, Singapore. Springer Nature Singapore.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase generation with correlation constraints. In *EMNLP*, pages 4057–4066.

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.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2023. A close look into the calibration of pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1343–1367, Toronto, Canada. Association for Computational Linguistics.

Minseok Choi, Chaeheon Gwak, Seho Kim, Si Kim, and Jaegul Choo. 2023. SimCKP: Simple contrastive learning of keyphrase representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3003–3015, Singapore. Association for Computational Linguistics.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lam Do, Pritom Saha Akash, and Kevin Chen-Chuan Chang. 2023. Unsupervised open-domain keyphrase generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10614–10627, Toronto, Canada. Association for Computational Linguistics.

Krishna Garg, Jishnu Ray Chowdhury, and Cornelia Caragea. 2022. Keyphrase generation beyond the boundaries of title and abstract. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5809–5821, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Krishna Garg, Jishnu Ray Chowdhury, and Cornelia Caragea. 2023. Data augmentation for low-resource keyphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8442–8455, Toronto, Canada. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Xiaoli Huang, Tongge Xu, Lvan Jiao, Yueran Zu, and Youmin Zhang. 2021. Adaptive beam search decoding for discrete keyphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):13082–13089.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223.

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.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. 2023. Generative calibration for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2312–2333, Singapore. Association for Computational Linguistics.

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.

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.

Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large dataset for keyphrases extraction.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2021. Learning rich representation of keyphrases from text. *ArXiv*, abs/2112.08547.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 891–906, Seattle, United States. Association for Computational Linguistics.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *CoRR*, abs/1903.00802.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jie Liu, Yaguang Li, Shizhu He, Shun Wu, Kang Liu, Shenping Liu, Jiong Wang, and Qing Zhang. 2024. Seq2set2seq: A two-stage disentangled method for reply keyword generation in social media via multi-label prediction and determinantal point processes. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Rui Liu, Zheng Lin, and Weiping Wang. 2020. Keyphrase prediction with pre-trained language model. *ArXiv*, abs/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. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Aman Madaan, Dheeraj Rajagopal, Niket Tandon, Yiming Yang, and Antoine Bosselut. 2022. Conditional set generation using seq2seq models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4874–4896, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rui Meng, Tong Wang, Xingdi Yuan, Yingbo Zhou, and Daqing He. 2023. General-to-specific transfer labeling for domain adaptable keyphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1602–1618, Toronto, Canada. Association for Computational Linguistics.

Rui Meng, Xingdi Yuan, Tong Wang, Peter Brusilovsky, Adam Trischler, and Daqing He. 2019. Does order matter? an empirical study on generating multiple keyphrases as a sequence. *arXiv preprint arXiv:1909.03590*.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He. 2021. An empirical study on neural keyphrase generation.

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.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2901–2907. AAAI Press.

Khanh Nguyen and Brendan O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598, Lisbon, Portugal. Association for Computational Linguistics.

Seo Yeon Park and Cornelia Caragea. 2022a. A data cartography based MixUp for pre-trained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4244–4250, Seattle, United States. Association for Computational Linguistics.

Seo Yeon Park and Cornelia Caragea. 2022b. On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5364–5374, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks.

Jishnu Ray Chowdhury, Seo Yeon Park, Tuhin Kundu, and Cornelia Caragea. 2022. KPDROP: Improving absent keyphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4853–4870, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. 2021. Simple and effective vae training with calibrated decoders. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9179–9189. PMLR.

Mobashir Sadat and Cornelia Caragea. 2022. Hierarchical multi-label classification of scientific documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8923–8937, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Xianjie Shen, Yinghan Wang, Rui Meng, and Jingbo Shang. 2022. Unsupervised deep keyphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11303–11311.

Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Ratn 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, Online. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.

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.

Di Wu, Wasi Ahmad, and Kai-Wei Chang. 2023. Rethinking model selection and decoding for keyphrase generation with pre-trained sequence-to-sequence models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6642–6658, Singapore. Association for Computational Linguistics.

Di Wu, Wasi Uddin Ahmad, and Kai-Wei Chang. 2022a. Pre-trained language models for keyphrase generation: A thorough empirical study. *arXiv preprint arXiv:2212.10233*.

Di Wu, Wasi Uddin Ahmad, Sunipa Dev, and Kai-Wei Chang. 2022b. Representation learning for resource-constrained keyphrase generation. *ArXiv*, abs/2203.08118.

Huanqin Wu, Wei Liu, Lei Li, Dan Nie, Tao Chen, Feng Zhang, and Di Wang. 2021. UniKeyphrase: A unified extraction and generation framework for keyphrase prediction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 825–835, Online. Association for Computational Linguistics.

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding neural abstractive summarization models via uncertainty. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6275–6281.

Jiacheng Ye, Ruijian Cai, Tao Gui, and Qi Zhang. 2021a. Heterogeneous graph neural networks for keyphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2705–2715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021b. One2Set: Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4598–4608, Online. Association for Computational Linguistics.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One size does not fit all: Generating and evaluating variable number of keyphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975.

Yuxiang Zhang, Tianyu Yang, Tao Jiang, Xiaoli Li, and Suge Wang. 2022. Hyperbolic deep keyphrase generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 521–536. Springer.

Guangzhen Zhao, Guoshun Yin, Peng Yang, and Yu Yao. 2022. Keyphrase generation via soft and hard semantic corrections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7757–7768, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2023. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9778–9795, Singapore. Association for Computational Linguistics.

A Implementation Details

ExHiRD is trained from the publicly available code [2] using the original settings mentioned in the paper (Chen et al., 2020). Transformer One2Seq and One2Set are trained from the code[3] made publicly available by Ye et al. (2021b). BART was trained from the script in a publicly available code[4]. T5 was trained with SM3 optimizer (Anil et al., 2019) for its memory efficiency. We use a learning rate (lr) of 0.1 and a warm-up for 2000 steps with the following formulation: $lr = lr \cdot minimum\left(1, \left(\frac{steps}{warmup\_steps}\right)^2\right)$ The learning rate was tuned among the following choices: $[1.0, 0.1, 0.01, 0.001]$ (using grid search). We use an effective batch size of 64 based on gradient accumulation. We train T5 for 10 epochs with a maximum gradient norm of 5. All models were trained using teacher forcing. We use train, validation, and test splits from Meng et al. (2017) for kp20k. Following (Meng et al., 2019; Chen et al., 2020), the keyphrases in the target sequence are ordered according to their position of the first occurrence within the source text. The first occurring keyphrase in the source text appears first in the target sequence. The absent keyphrases were appended in the end according to their original order. Predictions for both models were generated through greedy decoding. We use a maximum length of 50 tokens for T5 during decoding. The models were trained in $1-2$ NVIDIA RTX A5000.

B Evaluation

For the F_1 evaluations, we first stemmed both target keyphrases and predicted keyphrases using Porter stemmer. We removed all duplicates from predictions after stemming. We determined whether a keyphrase is present by checking the stemmed version of the source document. As standard, we consider two F_1 based metrics - $F_1@M$ and $F_1@5$. Both are macro-F_1. For $F_1@M$, we select all the keyphrase predictions generated by the model. For $F_1@5$, following Chen et al. (2020), we select at most the top 5 keyphrase predictions. If there are less than 5 predictions, similar to Chen et al. (2020); Ye et al. (2021b), we append incorrect keyphrases to the predictions to make it exactly 5 (which is equivalent to always dividing by 5 for per sample precision computation).

[2] https://github.com/Chen-Wang-CUHK/ExHiRD-DKG
[3] https://github.com/jiacheng-ye/kg_one2set
[4] https://github.com/uclanlp/DeepKPG